SRC PE Software

Application Services Gateway Configuration Guide

Release
4.10.x

Modified: 2016-05-26

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

SRC PE Software Application Services Gateway Configuration Guide
Release 4.10.x

Revision History
June 2016—Revision 1

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

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

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

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

END USER LICENSE AGREEMENT

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

Table of Contents

List of Figures

List of Tables

About the Documentation

SRC Documentation and Release Notes

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

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

Audience

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

Documentation Conventions

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

Table 1: Notice Icons

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

Table 2: Text Conventions

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

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

Regular sans serif typeface	• Represents configuration statements. • Indicates SRC CLI commands and options in text. • Represents examples in procedures. • Represents URLs.	• **system ldap server{ stand-alone;** • Use the **request sae modify device failover** command with the **force** option • **user@host# . . .** • **http://www.juniper.net/techpubs/software/management/src/api-index.html**

Table 2: Text Conventions *(continued)*

Italic sans serif typeface	Represents variables in SRC CLI commands.	**user@host# set local-address *local-address***
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Italic typeface	• Emphasizes words. • Identifies book names. • Identifies distinguished names. • Identifies files, directories, and paths in text but not in command examples.	• There are two levels of access: *user* and *privileged*. • *SRC PE Getting Started Guide* • *o=Users, o=UMC* • The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class=\ net.juniper.smgt.sae.plugin\ RadiusTrackingPluginEvent
Words separated by the | symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic | line

Documentation Feedback

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

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Accessing SRC Software in a Business-to-Business Environment

CHAPTER 1

Overview of the Web Services Gateway

Web Services Gateway Overview

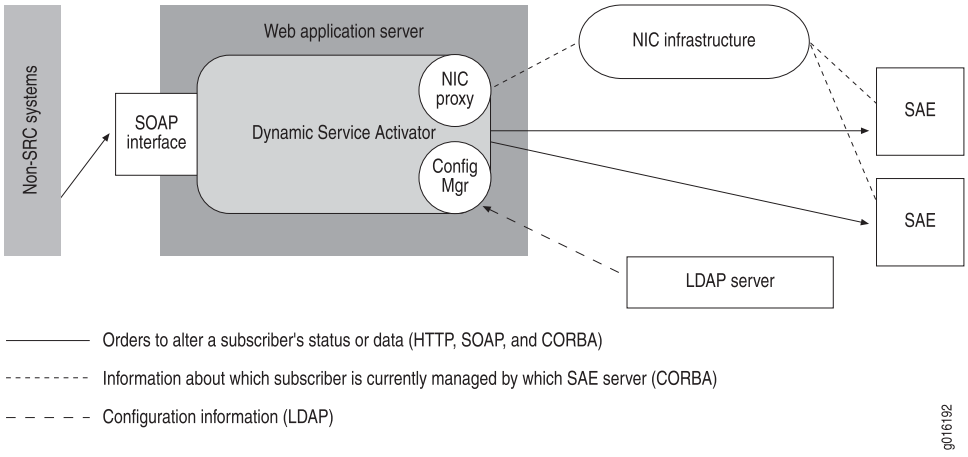
The Web Services Gateway allows a *gateway client*—an application that is not part of the SRC network—to interact with SRC components through a Simple Object Access Protocol (SOAP) interface. This feature is useful for business-to-business (B2B) situations, such as wholesaler-retailer environments. Typically, the wholesaler owns and administers the SRC components, whereas the retailer maintains a database of subscribers. Retailers purchase services from one or more wholesalers and sell the services to their subscribers.

The Web Services Gateway supports Web applications that allow gateway clients to interact with the SRC network. The SRC owner installs, configures, and administers the Web applications. Using information provided by the SRC owner, the business partner creates gateway clients to communicate with the SRC components.

The Web Services Gateway offers the Dynamic Service Activation Web application (subsequently known as Dynamic Service Activator). The Dynamic Service Activator allows a gateway client to dynamically activate and deactivate services for subscribers and to run scripts that manage the SAE.

Figure 1 on page 4 shows the architecture for the Web Services Gateway.

Figure 1: Web Services Gateway Architecture



Related Documentation

- Dynamic Service Activator Overview on page 7
- Dynamic Service Activator in a Redundant Environment on page 9
- Starting Dynamic Service Activator (SRC CLI) on page 33
- Terminology on page 4

Terminology

Table 3 on page 4 provides a list of terms and corresponding definitions that are used in the gateway documentation.

Table 3: Web Services Gateway Terms

Term	Definition
Argument	Value that a gateway client passes to a Web application; the application uses the value to perform an action on behalf of the gateway client
Business partner	Organization that manages a database of subscribers and uses an SRC owner's equipment to provision services
Dynamic properties	Properties that the software changes in the directory
Gateway client	Software application that submits requests to a gateway Web application
Gateway extension	Servlet deployed on the Web Services Gateway Web application server
Gateway Web application	Software application that belongs to the SRC owner and allows business partners to interact with the SRC network
Configuration namespace	Entry in the directory that defines a list of properties and inherits properties from parent objects in the hierarchy

Table 3: Web Services Gateway Terms *(continued)*

Term	Definition
SRC owner	Organization that owns the SRC components and software and offers the use of this equipment to business partners who manage a database of subscribers
Static properties	Properties that you change in the directory
Web application server	Software application that supports Web applications and serves Web pages to browsers through HTTP

Related Documentation

- Web Services Gateway Overview on page 3
- Dynamic Service Activator Overview on page 7
- Dynamic Service Activator in a Redundant Environment on page 9

CHAPTER 2

Activating Services with SOAP

Dynamic Service Activator Overview

Dynamic Service Activator enables business partners or their subscribers to dynamically activate services or run scripts on an SRC owner's SAE through the SAE's CORBA remote interface.

For managing services, Dynamic Service Activator supports a fixed set of methods and uses the SAE access interface module to access the SAE core API. For invoking scripts, Dynamic Service Activator uses the remote Java scripts interface module. These scripts can perform any function offered by the SAE's core Java APIs.

For access control, Dynamic Service Activator requires the Juniper Networks database to be running on the same host.

The SRC owner is responsible for:

- Deciding how to control clients' access to methods and scripts. You can allow clients to access all methods and scripts in the directory or restrict clients' access to specific methods and scripts.
- Configuring Dynamic Service Activator. If you restrict clients' access to specific methods and scripts, this task involves configuring a set of access controls between a client and each method or script that the client can use.
- Creating Java scripts that Dynamic Service Activator will invoke on an SAE (see the SAE CORBA Remote API documentation on the Juniper Networks website at **http://www.juniper.net/techpubs/software/management/src/api-index.html**).

The business partner is responsible for:

- Creating the gateway clients that communicate with the gateway.
- Optionally, providing a way for subscribers to activate services; for example, through a portal.

Dynamic Service Activator Operation

The following steps explain how Dynamic Service Activator interacts with other components to enable the gateway client to execute a method or script on a particular SAE. Figure 2 on page 9 illustrates the processes.

1. The gateway client sends a SOAP message to the Web application server through HTTP.

   The request includes:

   - Name of the method or script that the gateway client wants to activate.
   - Arguments that the gateway client wants to pass to the method or script.
   - Type-value arguments that the gateway client passes to the method or script for one of the following:
     - Subscriber's DN
     - Name with which the subscriber logs in
     - Name of the interface and name of the virtual router to which the subscriber connects
     - SNMP index of the interface and name of the virtual router to which the subscriber connects
     - Subscriber's IP address, name of the managed interface, and name of the virtual router to which the subscriber connects
     - Subscriber's primary username
     - Subscriber's session handle

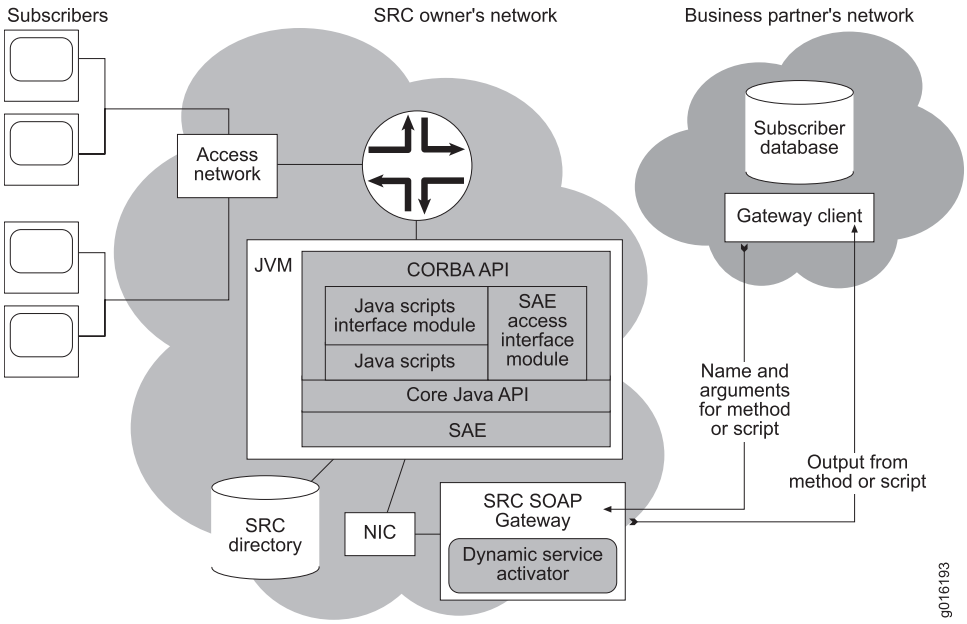
2. The Web application server authenticates the gateway client's identity.
3. The Web application passes the SOAP request to Dynamic Service Activator.
4. Dynamic Service Activator checks that:
   a. The Web application server has authenticated the gateway client and refused any requests from an unauthenticated gateway client.
   b. The gateway client is allowed to access the specified method or script.
   c. The arguments supplied by the gateway client satisfy any restrictions specified in the Dynamic Service Activator configuration that apply to the gateway client for the requested method or script.
5. If the gateway client satisfies these requirements, Dynamic Service Activator passes an argument, such as a subscriber's IP address specified in the Dynamic Service Activator configuration, to the network information collector (NIC).
6. The NIC uses the argument to determine the SAE on which Dynamic Service Activator should execute the method or script.

7. Dynamic Service Activator passes the name of the method or script and the associated arguments to the SAE through CORBA.
8. The SAE executes the method or script and returns the expected output or SOAP fault codes through CORBA to Dynamic Service Activator.

   The expected output from the method or script depends on the values that the method or script is programmed to return. Some methods and scripts return no values; others may return a short indicator of the success or failure of the operation, an HTML page, or a complex data structure in a format the gateway client understands.

   For information about the SOAP fault codes that the methods and scripts return, see "SOAP Fault Codes for Dynamic Service Activator" on page 74.
9. Dynamic Service Activator returns an output from the method or script to the gateway client through a SOAP response.

Figure 2: Dynamic Service Activator Operation



Related Documentation

- API for Dynamic Service Activator on page 65
- Dynamic Service Activator in a Redundant Environment on page 9
- Before You Use Dynamic Service Activator on page 14
- Starting Dynamic Service Activator (SRC CLI) on page 33
- Configuration Statements for Dynamic Service Activator on page 11

Dynamic Service Activator in a Redundant Environment

Based on the availability requirements for the Dynamic Service Activator, you can set up Dynamic Service Activator to run in a redundant environment in the following ways:

- Use load-balancing software to manage load for Dynamic Service Activator between two or more instances of it that run on different systems.

  The Web application server on each system may already be installed in an environment that uses load-balancing software.

- Install and activate an instance of Dynamic Service Activator on two systems, and configure services to send SOAP requests from client applications to both instances of the Dynamic Service Activators at the same time.

  Each SOAP request tries to activate or deactivate the same service session at the same time. This scenario gives better subscriber response time than spaced requests from the service.

  For a configuration in which services send SOAP requests to two Dynamic Service Activators, you can optimize the code to wait only for the first call. As soon as one of the SOAP requests is completed, the service application can continue and not wait for the second call to be completed. To detect a failure of the Web Services Gateway or the client's network connection, the application needs to wait for both calls to time out before the request fails.

  Duplicate requests and responses places a higher load on the system that runs the service application and on the system running the SAE.

Related Documentation

- Web Services Gateway Overview on page 3
- Dynamic Service Activator Overview on page 7
- Terminology on page 4

CHAPTER 3

Configuring Dynamic Service Activator (SRC CLI)

Configuration Statements for Dynamic Service Activator

Use the following configuration statements to configure the operating properties for Dynamic Service Activator at the [edit] hierarchy level.

```
slot number dsa {
   shared shared;
}

slot number dsa deploy {
   virtual-host virtual-host;
}

slot number dsa initial {
   base-dn base-dn;
   static-dn static-dn;
   dynamic-dn dynamic-dn;
}

slot number dsa initial directory-connection {
   url url;
   backup-urls [backup-urls...];
   principal principal;
   credentials credentials;
   protocol protocol;
```

```
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}

slot number dsa initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}
```

Use the following configuration statements to configure Dynamic Service Activator at the [edit] hierarchy level.

```
shared dsa group name

shared dsa group name configuration {
    disable-access-control-mechanism;
    disable-soap-client-authentication;
}

shared dsa group name configuration client name {
    restricted;
}

shared dsa group name configuration client name application application-id {
    disabled;
    listener-url listener-url;
    http-id http-id;
    http-password http-password;
    jms-queue-size jms-queue-size;
}

shared dsa group name configuration client name application application-id
    event-subscription event-subscription-name {
    disabled;
    subject-id subject-id;
    public-interface-id public-interface-id;
    event-type-filter [(user-start | user-interim | user-stop | service-start | service-interim |
        service-stop | interface-start | interface-interim | interface-stop)...];
    service-name-filter [service-name-filter...];
    event-filter event-filter;
    attribute-names [attribute-names...];
}

shared dsa group name configuration client name permissions attributes {
    service [service...];
    subscription [subscription...];
    subscriber [subscriber...];
}

shared dsa group name configuration client name permissions method name

shared dsa group name configuration client name permissions method name constraints
    argument-index

shared dsa group name configuration client name permissions script name
```


shared dsa group *name* configuration client *name* permissions script *name* constraints *argument-index*

shared dsa group *name* configuration logger *name* file {
filter *filter*;
filename *filename*;
rollover-filename *rollover-filename*;
maximum-file-size *maximum-file-size*;
}

shared dsa group *name* configuration logger *name* syslog {
filter *filter*;
host *host*;
facility *facility*;
format *format*;
}

shared dsa group *name* configuration method (commit-resources | invoke-gateway-extension | invoke-script | query-available-services | query-contexts | release-resources | subscriber-activate-service | subscriber-deactivate-service | subscriber-login | subscriber-logout | subscriber-modify-service | subscriber-read-subscription) constraints *argument-index*

shared dsa group *name* configuration nic-proxy-configuration *name*

shared dsa group *name* configuration nic-proxy-configuration *name* cache {
cache-size *cache-size*;
cache-cleanup-interval *cache-cleanup-interval*;
cache-entry-age *cache-entry-age*;
}

shared dsa group *name* configuration nic-proxy-configuration *name* nic-host-selection {
groups [*groups*...];
selection-criteria (roundRobin | randomPick | priorityList);
}

shared dsa group *name* configuration nic-proxy-configuration *name* nic-host-selection blacklisting {
try-next-system-on-error;
number-of-retries-before-blacklisting *number-of-retries-before-blacklisting*;
blacklist-retry-interval *blacklist-retry-interval*;
}

shared dsa group *name* configuration nic-proxy-configuration *name* resolution {
resolver-name *resolver-name*;
key-type *key-type*;
value-type *value-type*;
expect-multiple-values;
constraints *constraints*;
}

shared dsa group *name* configuration nic-proxy-configuration *name* test-nic-bindings {
use-test-bindings;
}

shared dsa group *name* configuration nic-proxy-configuration *name* test-nic-bindings key-values *name*

shared dsa group *name* configuration script *name*

shared dsa group *name* configuration script *name* constraints *argument-index*

```
shared dsa group name configuration session-handle {
  strong-encoding;
  encoding-key encoding-key;
}

shared dsa group name configuration subscriber-types name {
  subscriber-id-type (address | dn | login-name | interface-name | interface-index |
    address-interface-name | primary-user-name | session-handle |tunnel-session |
    global-address | global-login-name);
  nic-proxy nic-proxy;
}
```

For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

Related Documentation

- Before You Use Dynamic Service Activator on page 14
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19

Before You Use Dynamic Service Activator

Before you use Dynamic Service Activator, you must:

- Deploy a working SRC network.
- Configure and start the application server.

  See *Configuring the Web Application Server (SRC CLI)*.
- Configure a NIC that identifies the SAE reference for each subscriber type.

  See *Configuring the NIC (SRC CLI)*.
- Create the Java scripts that Dynamic Service Activator invokes (see the SAE CORBA Remote API documentation on the Juniper Networks website at **http://www.juniper.net/techpubs/software/management/src/api-index.html**).
- Load the sample data with the **request system ldap load dsa-configuration** command.

  See "Sample Data for Dynamic Service Activator" on page 34.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Dynamic Service Activator in a Redundant Environment on page 9
- Starting Dynamic Service Activator (SRC CLI) on page 33
- SRC CLI Commands to Monitor Dynamic Service Activator on page 47

Enabling Dynamic Service Activator on a Web Application Server (SRC CLI)

Deploy one copy of Dynamic Service Activator on a Web application server to support all business partners. Before you enable Dynamic Service Activator, you must configure and start the application server.

To enable Dynamic Service Activator, perform the following tasks:

Creating Grouped Configurations for Dynamic Service Activator (SRC CLI)

You must configure Dynamic Service Activator within a group. Dynamic Service Activator configurations are organized into a hierarchy of groups. Subordinate groups inherit configuration from superior groups.

Configuration groups allow you to share the Dynamic Service Activator configuration with different Dynamic Service Activator instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped Dynamic Service Activator configuration that is shared with some Dynamic Service Activator instances. For example, if you create two different Dynamic Service Activator groups called config1 and config2 within the shared Dynamic Service Activator configuration, you could select the Dynamic Service Activator configuration that should be associated with a particular Dynamic Service Activator instance.

Use the **shared** option of the **slot** ***number*** **dsa** statement to select the group for a Dynamic Service Activator instance as part of the local configuration. Use the **shared dsa group** ***name*** **configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for a Dynamic Service Activator instance. For example, to select a group called config1 in the root group:

   ```
   [edit]
   user@host# set slot 0 dsa shared /config1
   ```

2. Commit the configuration.

   ```
   [edit]
   user@host# commit
   commit complete.
   ```

3. From configuration mode, configure a group. For example, to configure a group called config1, specify the group as part of the Dynamic Service Activator configuration.

   ```
   [edit]
   user@host# edit shared dsa group config1 ?
   Possible completions:
   <[Enter]> Execute this command
   > configuration
   > group Group of configuration properties
   ```


| Pipe through a command

For more information, see "Configuring Dynamic Service Activator Properties (SRC CLI)" on page 19.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
- Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
- Starting Dynamic Service Activator (SRC CLI) on page 33
- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Configuring Dynamic Service Activator (SRC CLI)

To use Dynamic Service Activator in the SRC network, you configure basic and initial properties, including directory connection and directory eventing properties. You also configure Dynamic Service Activator properties.

For information about these configuration procedures, see:

1. Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
2. Configuring Dynamic Service Activator Properties (SRC CLI) on page 19

Configuring Local Properties for Dynamic Service Activator (SRC CLI)

Configure basic and initial properties, including directory connection and directory eventing properties. Tasks to configure the local properties for Dynamic Service Activator are:

- Configuring Basic Local Properties for Dynamic Service Activator on page 16
- Configuring Initial Properties for Dynamic Service Activator on page 17
- Configuring Initial Directory Connection Properties for Dynamic Service Activator on page 17
- Configuring Initial Directory Eventing Properties for Dynamic Service Activator on page 18

Configuring Basic Local Properties for Dynamic Service Activator

Use the following configuration statements to configure basic local properties for Dynamic Service Activator:

```
slot number dsa {
   shared shared;
}
```

To configure basic local properties:

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

   [edit]

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

2. Specify the configuration namespace for Dynamic Service Activator as the path, relative to the root of the static configuration properties, that defines the object for the namespace.

```
[edit slot 0 dsa]
user@host# set shared shared
```

For example:

```
[edit slot 0 dsa]
user@host# set shared /sample
```

3. (Optional) Verify your configuration.

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

Configuring Initial Properties for Dynamic Service Activator

Use the following configuration statements to configure initial properties for Dynamic Service Activator:

```
slot number dsa initial {
  base-dn base-dn;
  static-dn static-dn;
  dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

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

```
[edit]
user@host# edit slot 0 dsa initial
```

2. Specify the properties for Dynamic Service Activator.

```
[edit slot 0 dsa initial]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see "Changing the Location of Data in the Directory" on page 149.

Configuring Initial Directory Connection Properties for Dynamic Service Activator

Use the following configuration statements to configure directory connection properties for Dynamic Service Activator:

```
slot number dsa initial directory-connection {
  url url;
  backup-urls [backup-urls...];
  principal principal;
```

```
    credentials credentials;
    protocol protocol;
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
  }
```

To configure directory connection properties:

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

   ```
   [edit]
   user@host# edit slot 0 dsa initial directory-connection
   ```

2. Specify the properties for Dynamic Service Activator.

   ```
   [edit slot 0 dsa initial directory-connection]
   user@host# set ?
   ```

Configuring Initial Directory Eventing Properties for Dynamic Service Activator

Use the following configuration statements to configure directory eventing properties for Dynamic Service Activator:

```
slot number dsa initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory eventing properties:

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

   ```
   [edit]
   user@host# edit slot 0 dsa initial directory-eventing
   ```

2. Specify the initial directory eventing properties for Dynamic Service Activator.

   ```
   [edit slot 0 dsa initial directory-eventing]
   user@host# set ?
   ```

   For more information about configuring local properties for the SRC components, see *Configuring Initial Directory Eventing Properties for SRC Components*.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
- Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39

- Configuring the Test Environment for Dynamic Service Activator Services on page 54
- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Configuring Dynamic Service Activator Properties (SRC CLI)

You can configure properties for the Dynamic Service Activator (DSA) that include general properties, types of subscribers, encoding key and encoding algorithm for session handles, network information collector (NIC) proxy for each subscriber type, access to methods and scripts, and access to attributes. You can also restrict access to service sessions, send service activation engine (SAE) tracking events, publish events to Simple Object Access Protocol (SOAP) applications, and log destinations.

Tasks to configure the Dynamic Service Activator are:

- Configuring General Properties for the Dynamic Service Activator on page 19
- Configuring Subscriber Types for the Dynamic Service Activator on page 20
- Configuring Session Handles for Dynamic Service Activator on page 21
- Configuring the NIC Proxies for Dynamic Service Activator on page 22
- Configuring Access to Methods and Scripts for Dynamic Service Activator on page 22
- Configuring Access to Methods for Dynamic Service Activator on page 23
- Configuring Access to Scripts for Dynamic Service Activator on page 26
- Restricting Access to Service Sessions for Dynamic Service Activator on page 27
- Configuring Access to Attributes for Dynamic Service Activator on page 27
- Configuring the SAE to Send Tracking Events to Dynamic Service Activator on page 28
- Configuring Dynamic Service Activator to Publish Events to SOAP Applications on page 29
- Configuring the Logging Destinations for Dynamic Service Activator on page 31

Configuring General Properties for the Dynamic Service Activator

The general properties for the Dynamic Service Activator determine the behavior of the application rather than the relationship between a gateway client and the application.

Use the following configuration statements to configure an operation script action:

```
shared dsa group name configuration {
  disable-soap-client-authentication;
  disable-access-control-mechanism;
}
```

To configure general properties for the Dynamic Service Activator:

1. From configuration mode, access the statement that configures the general properties. In this sample procedure, the properties are configured in the trial group.

```
[edit]
user@host# edit shared dsa group trial configuration
```

2. (Optional) Specify the type of access that gateway clients have to methods and scripts.

   ```
   [edit shared dsa group trial configuration]
   user@host# set disable-access-control-mechanism
   ```

   Set this value only if you want gateway clients to have unrestricted access to all methods and scripts. The client must still provide a valid client name and password, and the client name must be configured to access at least one method (for Dynamic Service Activator or PacketCable Multimedia) to access methods of the same type. By default, gateway clients have access only to methods and scripts that you specify in the configuration. Access control should be disabled only for troubleshooting purposes.

3. (Optional) Disable the security control for SOAP clients.

   ```
   [edit shared dsa group trial configuration]
   user@host# set disable-soap-client-authentication
   ```

NOTE: You must configure the **disable-access-control-mechanism** option when you disable the security control for SOAP clients by using the **disable-soap-client-authentication** option. Otherwise, a message indicating that the configuration is invalid is displayed when you commit the changes.

Configuring Subscriber Types for the Dynamic Service Activator

You configure which types of information identify subscribers to the service activation engine (SAE). The subscriber types that you can configure are the same subscriber types that you can use in applications created with the SAE Common Object Request Broker Architecture (CORBA) remote application programming interface (API).

To configure subscriber types:

1. From configuration mode, access the statement that configures the subscriber types. The specified name is used to construct the subscriber's URI. In this sample procedure, the properties are configured in the trial group.

   ```
   [edit]
   user@host# edit shared dsa group trial configuration subscriber-types name
   ```

2. Specify the type of information used to identify a subscriber.

   ```
   [edit shared dsa group trial configuration subscriber-types name]
   user@host# set subscriber-id-type (address | dn | login-name | interface-name |
     interface-index | address-interface-name | primary-user-name | session-handle
     |tunnel-session | global-address | global-login-name)
   ```

   where:

- **address**—Subscriber's IP address
- **dn**—Distinguished name of the subscriber profile
- **login-name**—Subscriber's login name
- **interface-name**—Name of the interface and name of the virtual router to which the subscriber connects
- **interface-index**—SNMP index of the interface and name of the virtual router to which the subscriber connects
- **address-interface-name**—Subscriber's IP address, name of the managed interface, and name of the virtual router to which the subscriber connects
- **primary-user-name**—Primary username
- **session-handle**—Subscriber's session handle used to reference an existing subscriber session
- **tunnel-session**—Subscriber's Layer 2 Tunneling Protocol (L2TP) tunnel session identifier, tunnel identifier, and IP address of the L2TP access concentrator (LAC)
- **global-address**—Subscriber's IP address and VPN identifier
- **global-login-name**—Subscriber's login name and VPN identifier

3. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy.

   [edit shared dsa group trial configuration subscriber-types *name*]
   user@host# **set nic-proxy** *nic-proxy*

   For example:

   [edit shared dsa group trial configuration subscriber-types *name*]
   user@host# **set nic-proxy ip**

Configuring Session Handles for Dynamic Service Activator

You configure the encoding key and encoding algorithm for the session handles to determine how the session handle URI is constructed. Session handles are encoded when returned by SOAP calls for the service provider's privacy and to prevent service provider partners who operate SOAP clients from managing subscribers with whom they do not have a relationship.

To configure encoding for session handles:

1. From configuration mode, access the statement that configures the session handles.

   [edit]
   user@host# **edit shared dsa group trial configuration session-handle**

2. Specify the private key to use for encoding a session handle.

   [edit shared dsa group trial configuration session-handle]
   user@host# **set encoding-key** *encoding-key*

3. (Optional) Specify that the DES algorithm with MD5 hash digested key be used to encode the session handle. If you do not set this value, an exclusive OR algorithm is used.

   ```
   [edit shared dsa group trial configuration session-handle]
   user@host# set strong-encoding
   ```

Configuring the NIC Proxies for Dynamic Service Activator

You create a NIC proxy for each subscriber type to be configured. The name of the NIC proxy must match the name configured for the NIC proxy namespace.

Subscriber types that have different subscriber ID types can use the same NIC proxy. For example, a subscriber type configured as SubscriberType1 that has a subscriber ID type of interface-name, and a subscriber type configured as subscriberType2 that has a subscriber ID type of interface-index can both use the same NIC proxy. Likewise, a subscriber type configured as SubscriberType1 and a subscriber type configured as subscriberType2 that both have a subscriber ID type of address can use the same NIC proxy.

To configure NIC proxies:

1. From configuration mode, access the statement that configures the NIC proxy. In this sample procedure, the NIC proxy called ip is configured in the trial group.

   ```
   [edit]
   user@host# edit shared dsa group trial configuration nic-proxy-configuration ip
   ```

2. Specify the properties for the NIC proxy.

   ```
   [edit shared dsa group trial configuration nic-proxy-configuration ip]
   user@host# set ?
   ```

   For information about configuring NIC proxies, see *Configuration Statements for NIC Proxies*.

Configuring Access to Methods and Scripts for Dynamic Service Activator

Configuring access to methods and scripts involves adding methods, scripts, and clients to the configuration and configuring access properties between each client and each method or script.

NOTE: Client profiles are cached by Dynamic Service Activator for 30 minutes. If you change the password or role of a client that has been used within the last 30 minutes, it can take up to 30 minutes before these changes take effect.

When permissions are configured, roles are assigned to application server user objects automatically. The first time you add a method or script for a client, the DSA role is added to the corresponding application server user, and when the last method or script is deleted, the DSA role is removed from the corresponding user. Only role and password changes take up to 30 minutes to take effect.

If you do not want to wait 30 minutes for the changes to take effect, restart the Web application server.

Dynamic Service Activator interacts with the Web application server to determine whether a gateway client has access to a method or script. The name and credentials, such as a password, that are used to authenticate the gateway client are configured on the Web application server as user accounts.

Access constraints are regular expressions that the arguments for the method or script in the SOAP request must match. If the arguments for the method or script in a particular SOAP request do not match these regular expressions, then Dynamic Service Activator rejects the request.

Configuring Access to Methods for Dynamic Service Activator

Use the following configuration statements to configure methods and access properties between each client and each method:

shared dsa group *name* configuration client *name*

shared dsa group *name* configuration client *name* permissions method *name*

shared dsa group *name* configuration client *name* permissions method *name* constraints *argument-index*

shared dsa group *name* configuration method (commit-resources | invoke-gateway-extension | invoke-script | query-available-services | query-contexts | release-resources | subscriber-activate-service | subscriber-deactivate-service | subscriber-login | subscriber-logout | subscriber-modify-service | subscriber-read-subscription) constraints *argument-index*

Tasks to configure access to methods are:

- Configuring Methods on page 24
- Configuring Access to Methods on page 25

Configuring Methods

To configure methods for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the method to activate on the SAE. Use the text string that exactly matches the name of the method.

   ```
   [edit]
   user@host# edit shared dsa group name configuration method (commit-resources |
     invoke-gateway-extension | invoke-script | query-available-services | query-contexts
     | release-resources | subscriber-activate-service | subscriber-deactivate-service |
     subscriber-login | subscriber-logout | subscriber-modify-service |
     subscriber-read-subscription)
   ```

   where:

   - **commit-resources**—Specifies the resources that are being requested in the CommitResource message.
   - **invoke-gateway-extension**—Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server. The servlet can be a standalone application, or it can be part of a WAR or EAR file. When deployed, servlets invoked with this method should be accessible only from the local host.
   - **invoke-script**—Manages all operations involved with invoking scripts: retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client.
   - **query-available-services**—Searches for the services that are available to the calling application.
   - **query-contexts**—Searches for the context ID and context status for a subscriber.
   - **release-resources**—Specifies the resources that are being requested to be released in the ReleaseResources message.
   - **subscriber-activate-service**—Activates subscribers' subscriptions to services.
   - **subscriber-deactivate-service**—Deactivates subscribers' subscriptions to services.
   - **subscriber-login**—Logs in subscribers. This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNs.
   - **subscriber-logout**—Logs out subscribers. This method supports only subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNs.
   - **subscriber-modify-service**—Modifies subscriptions.
   - **subscriber-read-subscription**—Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions.

   For example:

```
user@host# edit shared dsa group trial configuration method
  subscriber-read-subscription
```

2. Specify the access constraints applied to the method for all clients.

```
[edit shared dsa group trial configuration method subscriber-read-subscription]
user@host# set constraints argument-index value
```

where:

- ***argument-index***—Zero-based index of the argument used to locate the SAE on which to activate the method
- ***value***—Regular expression

  For information about the regular expression syntax, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html.

For example:

```
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Configuring Access to Methods

To configure access to methods for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the gateway client's access to a method. You must use the same name for the gateway client that is configured on the Web application server.

   If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string "anonymous client" as the first argument to the SAE's Java scripts interface module.

```
[edit]
user@host# edit shared dsa group name configuration client name permissions method
  name
```

For example:

```
user@host# edit shared dsa group trial configuration client name permissions method
  subscriber-read-subscription
```

2. Specify the regular expressions that the method arguments must match for the gateway client.

```
[edit shared dsa group trial configuration client name permissions method
  subscriber-read-subscription]
user@host# set constraints argument-index value
```

For example:

```
[edit shared dsa group trial configuration client name permissions method
  subscriber-read-subscription]
```

```
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Configuring Access to Scripts for Dynamic Service Activator

Use the following configuration statements to configure scripts and access properties between each client and each script:

shared dsa group *name* configuration client *name*

shared dsa group *name* configuration client *name* permissions script *name*

shared dsa group *name* configuration client *name* permissions script *name* constraints *argument-index*

shared dsa group *name* configuration script *name*

shared dsa group *name* configuration script *name* constraints *argument-index*

Tasks to configure access to scripts are:

- Configuring Scripts on page 26
- Configuring Access to Scripts on page 27

Configuring Scripts

To configure scripts for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the script to activate on the SAE. Use the text string that exactly matches the name of the script.

   ```
   [edit]
   user@host# edit shared dsa group name configuration script name
   ```

2. Specify the zero-based index of the script argument used to locate the SAE on which to invoke the script.

   ```
   [edit shared dsa group name configuration script name]
   user@host# set sae-locator-index sae-locator-index
   ```

3. Specify the access constraints applied to the script for all clients.

   ```
   [edit shared dsa group name configuration script name]
   user@host# set constraints argument-index value
   ```

   where:

   - ***argument-index***—Zero-based index of the argument used to locate the SAE on which to activate the method
   - ***value***—Regular expression

     For information about the regular expression syntax, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html.

   For example:

   ```
   user@host# set constraints 1 Audio-[a-zA-Z]*
   ```

Configuring Access to Scripts

To configure access to scripts:

1. From configuration mode, access the statement that configures the gateway client's access to a script. You must use the same name for the gateway client that is configured on the Web application server.

   If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string "anonymous client" as the first argument to the SAE's Java scripts interface module.

   ```
   [edit]
   user@host# edit shared dsa group name configuration client name permissions script
     name
   ```

2. Specify the regular expressions that the script arguments must match for the gateway client.

   ```
   [edit shared dsa group trial configuration client name permissions script name]
   user@host# set constraints argument-index value
   ```

   For example:

   ```
   [edit shared dsa group trial configuration client name permissions script name]
   user@host# set constraints 1 Audio-[a-zA-Z]*
   ```

Restricting Access to Service Sessions for Dynamic Service Activator

You can restrict the service sessions to which a gateway client has access. If you do not restrict access, the client has access to all service sessions.

To control the service sessions to which a gateway client has access:

1. From configuration mode, access the statement that configures the gateway client. You must use the same name for the gateway client that is configured on the Web application server.

   ```
   [edit]
   user@host# edit shared dsa group name configuration client name
   ```

2. (Optional) Specify that access is restricted to the client's own service session.

   ```
   [edit shared dsa group trial configuration client name]
   user@host# set restricted
   ```

Configuring Access to Attributes for Dynamic Service Activator

You can control the configured attributes to which a gateway client has access. If you do not configure the client's access to attributes, all configured attributes are allowed.

To control the attributes to which a gateway client has access:

1. From configuration mode, access the statement that configures the gateway client's access to attributes. You must use the same name for the gateway client that is configured on the Web application server.

   If you disable the access control mechanism, then the client has no restrictions on access to the configured attributes.

   ```
   [edit]
   user@host# edit shared dsa group name configuration client name permissions
     attributes
   ```

2. (Optional) Specify the service attributes to which the gateway client has access.

   ```
   [edit shared dsa group trial configuration client name permissions attributes]
   user@host# set service [service...]
   ```

   Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

3. (Optional) Specify the subscription attributes to which the gateway client has access.

   ```
   [edit shared dsa group trial configuration client name permissions attributes]
   user@host# set subscription [subscription...]
   ```

   Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

4. (Optional) Specify the subscriber attributes to which the gateway client has access.

   ```
   [edit shared dsa group trial configuration client name permissions attributes]
   user@host# set subscriber [subscriber...]
   ```

   Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

Configuring the SAE to Send Tracking Events to Dynamic Service Activator

The SAE communicates with Dynamic Service Activator through the Java Message Service (JMS) adapter plug-in. This SAE plug-in sends SAE tracking events to Dynamic Service Activator.

To configure the JMS adapter plug-in:

1. From configuration mode, access the JMS adapter plug-in configuration. In this sample procedure, the JMS adapter plug-in called soapapps is configured in the nw-area SAE group.

   ```
   [edit]
   user@host# edit shared sae group nw-area configuration plug-ins name soapapps
     jms-adaptor
   ```

2. Configure the grouped configuration used by all Dynamic Service Activator instances to which this plug-in forwards SAE events.

   [edit shared sae group nw-area configuration plug-ins name soapapps jms-adaptor]
   user@host# **set shared-dsa-configuration** ***shared-dsa-configuration***

3. Specify the Dynamic Server Activator application servers to which the SAE events are published. The URLs reference the JNDI name servers on the application servers.

   [edit shared dsa group nw-area configuration plug-ins name soapapps jms-adaptor]
   user@host# **set dsa-application-server-urls [*dsa-application-server-urls*...]**

4. (Optional) Specify the SAE plug-in event attributes. The attribute values are the event's subject ID, and they specify a subscriber or interface. The values can be set by the SAE's subscriber classification script. If any of the event attributes contain a value that matches the subject ID in a Dynamic Service Activator event subscription, then the plug-in forwards the event to a Dynamic Service Activator instance.

   [edit shared sae group nw-area configuration plug-ins name soapapps jms-adaptor]
   user@host# **set subject-id-attribute-name [*subject-id-attribute-name*...]**

If you want to configure JMS adapter plug-in features not available at the basic editing level, set the editing level to advanced or expert and use the CLI Help to obtain information about statement options.

Configuring Dynamic Service Activator to Publish Events to SOAP Applications

When the SAE sends tracking events to Dynamic Service Activator, Dynamic Service Activator can publish events to external SOAP applications used by content service providers. Events are published according to the configured event subscription. Tasks to configure event subscriptions are:

- Configuring External SOAP Applications on page 29
- Configuring Event Subscriptions on page 30

Configuring External SOAP Applications

Dynamic Service Activator can publish subscriber, service session, and interface events to external SOAP applications.

To configure the external SOAP application to which Dynamic Service Activator can publish events:

1. From configuration mode, access the statement that configures the application to which events are published. You must use the same name for the gateway client that is configured on the Web application server.

   [edit]
   user@host# **edit shared dsa group trial configuration client** *name* **application** *application-id*

   where ***application-id*** identifies the external SOAP application.

2. (Optional) Specify that sending events to this external SOAP application is disabled.

   [edit shared dsa group trial configuration client *name* application*application-id*]
   user@host# **set disabled**

3. Specify the URL of the external SOAP application.

   [edit shared dsa group trial configuration client *name* application *application-id*]
   user@host# **set listener-url** ***listener-url***

4. (Optional) If HTTP authentication is required, specify the username Dynamic Service Activator provides to the external SOAP application.

   [edit shared dsa group trial configuration client *name* application *application-id*]
   user@host# **set http-id** ***http-id***

5. (Optional) If HTTP authentication is required, specify the password Dynamic Service Activator provides to the external SOAP application.

   [edit shared dsa group trial configuration client *name* application *application-id*]
   user@host# **set http-password** ***http-password***

6. (Optional) Specify the size of the queue that holds received SAE events that have not been published yet.

   [edit shared dsa group trial configuration client *name* application *application-id*]
   user@host# **set jms-queue-size** ***jms-queue-size***

Configuring Event Subscriptions

You can configure the event subscriptions owned by external SOAP applications. The event subscription defines a set of events, and the attributes in those events, that are published to the external SOAP application.

To configure event subscriptions:

1. From configuration mode, access the statement that configures the event subscription owned by an application.

   [edit]
   user@host# **edit shared dsa group trial configuration client** *name* **application** *application-id* **event-subscription** *event-subscription-name*

   where ***event-subscription-name*** is the arbitrary identifier of the event subscription.

2. (Optional) Specify that this event subscription is disabled.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# **set disabled**

3. Configure the persistent identifier that specifies the subscriber or interface for which events are published. Only those events associated with the specified subscriber or interface are forwarded to the external SOAP application.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# **set subject-id** ***subject-id***

4. (Optional) For interface events, configure the identifier for the interface that is published to the external SOAP application instead of the persistent identifier.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# set public-interface-id *public-interface-id*

5. (Optional) Specify the types of events that Dynamic Service Activator forwards to the external SOAP application. If no event types are specified, all event types are allowed. Only subscriptions for subscriber and service session events can be created by calls to the Dynamic Service Activator's SOAP interface.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# set event-type-filter [(user-start | user-interim | user-stop | service-start | service-interim | service-stop | interface-start | interface-interim | interface-stop)...]

6. (Optional) Specify the names of services for which Dynamic Service Activator can send service session events to the external SOAP application. If no service names are specified, events for all services are allowed.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# set service-name-filter [*service-name-filter*...]

7. (Optional) Specify the SAE plug-in events that Dynamic Service Activator can forward to the external SOAP application. This filter allows constraints to be placed on the event attributes. If event attributes do not satisfy the specified constraints, Dynamic Service Activator cannot forward the event to the external SOAP application. If no events are specified, no constraints are applied.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# set event-filter *event-filter*

8. (Optional) Specify the names of SAE plug-in event attributes that Dynamic Service Activator can forward to the external SOAP application. If no attribute names are specified, all attributes are forwarded.

   [edit shared dsa group trial configuration client *name* application *application-id* event-subscription *event-subscription-name*]
   user@host# set attribute-names [*attribute-names*...]

Configuring the Logging Destinations for Dynamic Service Activator

Use the following configuration statements to configure logging destinations for Dynamic Service Activator:

```
shared dsa group name configuration logger name

shared dsa group name configuration logger name file {
  filter filter;
  filename filename;
```

```
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
  }

  shared dsa group name configuration logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
  }
```

Tasks to configure the logging destinations are:

- Configuring Logging Destinations to Store Messages in a File on page 32
- Configuring Logging Destinations to Send Messages to the System Logging Facility on page 32

Configuring Logging Destinations to Store Messages in a File

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

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

   ```
   [edit]
   user@host# edit shared dsa group trial configuration logger file-1 file
   ```

2. Specify the properties for the logging destination.

   ```
   [edit shared dsa group trial configuration logger file-1 file]
   user@host# set ?
   ```

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

Configuring Logging Destinations to Send Messages to the System Logging Facility

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

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

   ```
   [edit]
   user@host# edit shared dsa group trial configuration logger syslog-1 syslog
   ```

2. Specify the properties for the logging destination.

   ```
   [edit shared dsa group trial configuration logger syslog-1 syslog]
   user@host# set ?
   ```

   For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- Viewing Information About NIC Proxies (SRC CLI) on page 49
- Monitoring NIC Proxies (SRC CLI) on page 49

Deploying Dynamic Service Activator on the Virtual Host (SRC CLI)

Use the following configuration statements to deploy Dynamic Service Activator on the virtual host:

```
slot number dsa deploy {
  virtual-host virtual-host;
}
```

To deploy the application on the virtual host:

1. From configuration mode, access the statement that configures the virtual host.

   ```
   [edit]
   user@host# edit slot 0 dsa deploy
   ```

2. Specify the virtual host on which to deploy Dynamic Service Activator. Virtual hosts for Web applications are configured in the application server.

   ```
   [edit slot 0 dsa initial deploy]
   user@host# set virtual-host virtual-host
   ```

Related Documentation

- *Web Application Server on C Series Controllers Overview*
- Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) on page 44
- Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
- *Configuring Virtual Hosts for the Web Applications (SRC CLI)*

Starting Dynamic Service Activator (SRC CLI)

You must configure a group before you enable Dynamic Service Activator. To deploy Dynamic Service Activator, you must specify the virtual host, and the Web application server must be running.

To start Dynamic Service Activator:

- From operational mode, enable Dynamic Service Activator.

  ```
  user@host> enable component dsa
  ```

Related Documentation

- Dynamic Service Activator Overview on page 7
- Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
- Starting Dynamic Service Activator (C-Web Interface) on page 45
- Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19

Sample Data for Dynamic Service Activator

The SRC software includes sample data that you load with the **request system ldap load dsa-configuration** command. Loading the sample data does not update roles of existing clients. If clients with the names Bob, Joe, or Fred already exist in your configuration, you should use the **request system ldap load dsa-configuration replace** command to replace the existing client configurations with the sample data configuration. The sample Dynamic Service Activator is configured as follows:

- NIC proxy properties—For each subscriber type.
- Permissions—Specifically for each method and script.

Methods, Scripts, and Clients

The sample data shows several methods and one script. The methods are configured as follows:

- Clients Fred and Joe have access with no constraints to the Subscriber_readSubscription method.
- Client Joe has access to the Subscriber_activateService and the Subscriber_deactivateService methods with no constraints. Client Fred, however, can use these methods only to manage services with names that start with “Audio.”
- Client Fred can also use the Subscriber_modifyService method for services with names that start with “Audio.” Client Joe cannot use this method, but he can see the Subscriber_modifyService method for all services.
- Client Joe is the only client who can use the Subscriber_login and Subscriber_logout methods.
- Both clients Joe and Fred can use the invokeGwExtension method.

The script is configured as follows:

- Client Bob can use the Echo method. The script requires that the fourth argument be an IP address that starts with 10 (the IP address has the form 10.x.x.x).
- Client Joe can use the Echo method without restriction.

PCMM Available Services

The sample data has two global PCMM services and two PCMM clients configured:

- The two global available services are News and Video-Silver.
- Client Joe is allowed the Video-Silver and PCMM-Down service.
- Client Fred is allowed the Video-Gold service.
- Client Bob is not allowed any services (except for the global services).

Related Documentation

- *Loading Sample Data into a Juniper Networks Database (SRC CLI)*
- Dynamic Service Activator Overview on page 7
- Starting Dynamic Service Activator (SRC CLI) on page 33
- Methods for the Dynamic Service Activator Web Service Interface on page 66
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 56
- Running Methods for PCMM Services (SRC CLI) on page 61

CHAPTER 4

Configuring Dynamic Service Activator (C-Web Interface)

Enabling Dynamic Service Activator on a Web Application Server (C-Web Interface)

Deploy one copy of Dynamic Service Activator on a Web application server to support all business partners. Before you enable Dynamic Service Activator, you must configure and start the application server.

To enable Dynamic Service Activator:

1. Configure grouped configurations.

   See "Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface)" on page 38.

2. Configure local properties.

   See "Configuring Local Properties for Dynamic Service Activator (C-Web Interface)" on page 39.

3. Configure Dynamic Service Activator properties.

   See "Configuring Dynamic Service Activator Properties (C-Web Interface)" on page 40.

4. Deploy Dynamic Service Activator.

See "Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface)" on page 44.

5. Start Dynamic Service Activator.

   See "Starting Dynamic Service Activator (C-Web Interface)" on page 45.

Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface)

You must configure Dynamic Service Activator within a group. Dynamic Service Activator configurations are organized into a hierarchy of groups. Subordinate groups inherit configuration from superior groups. When you create a configuration group, the software creates a configuration with default values filled in.

Configuration groups allow you to share the Dynamic Service Activator configuration with different Dynamic Service Activator instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped Dynamic Service Activator configuration that is shared with some Dynamic Service Activator instances. For example, if you create two different Dynamic Service Activator groups called config1 and config2 within the shared Dynamic Service Activator configuration, you could select the Dynamic Service Activator configuration that should be associated with a particular Dynamic Service Activator instance.

To select and configure a group:

1. Click **Configure**, expand **Slot**, expand the slot for which you want to configure the group, and then click **DSA**.

   The DSA pane appears.

2. From the Shared list, select a group for a Dynamic Service Activator instance.
3. (Optional) Type a name for the new group in the box below the Shared list using the **/<path>** format, and click **Add**.
4. Click **Apply**.

   The group appears in the side pane when you are configuring Dynamic Service Activator properties.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- Starting Dynamic Service Activator (C-Web Interface) on page 45
- Viewing Statistics for Dynamic Service Activator (C-Web Interface) on page 51

Configuring Dynamic Service Activator (C-Web Interface)

To use Dynamic Service Activator in the SRC network, you configure basic and initial properties, including directory connection and directory eventing properties. You also configure Dynamic Service Activator properties.

For information about these configuration procedures, see:

1. Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
2. Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40

Configuring Local Properties for Dynamic Service Activator (C-Web Interface)

Configure basic and initial properties, including directory connection and directory eventing properties. Tasks to configure the local properties for Dynamic Service Activator are:

- Configuring Basic Local Properties for Dynamic Service Activator on page 39
- Configuring Initial Properties for Dynamic Service Activator on page 39
- Configuring Initial Directory Connection Properties for Dynamic Service Activator on page 39
- Configuring Initial Directory Eventing Properties for Dynamic Service Activator on page 40

Configuring Basic Local Properties for Dynamic Service Activator

To configure basic local properties:

1. Click **Configure>Slot>Slot:0>DSA**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Properties for Dynamic Service Activator

To configure initial local properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Directory Connection Properties for Dynamic Service Activator

To configure directory connection properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial>Directory Connection**.

   The Directory Connection pane appears.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Directory Eventing Properties for Dynamic Service Activator

To configure initial directory eventing properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial> Directory Eventing**.

   The Directory Eventing pane appears.

2. Enter information as described in the Help text in the main pane, and click **Apply**.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- Starting Dynamic Service Activator (C-Web Interface) on page 45
- Viewing Statistics for Dynamic Service Activator (C-Web Interface) on page 51

Configuring Dynamic Service Activator Properties (C-Web Interface)

Tasks to configure the Dynamic Service Activator are:

- Configuring General Properties for Dynamic Service Activator on page 40
- Configuring Subscriber Types for Dynamic Service Activator on page 41
- Configuring the NIC Proxies for Dynamic Service Activator on page 41
- Configuring Access to Methods and Scripts for Dynamic Service Activator on page 42
- Configuring Access to Methods for Dynamic Service Activator on page 42
- Configuring Access to Scripts for Dynamic Service Activator on page 43
- Configuring the Logging Destinations to Store Log Messages in a File on page 43
- Configuring the Logging Destinations to Send Log Messages to the System Logging Facility on page 44

Configuring General Properties for Dynamic Service Activator

The general properties for Dynamic Service Activator determine the behavior of the application rather than the relationship between a gateway client and the application.

To configure general properties for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure general properties, and click **Configuration**.
2. (Optional) To allow gateway clients unrestricted access to all methods and scripts, select the **Disable Access Control Mechanism** check box.

   The client still must provide a valid client name and password, and the client name must be configured to access at least one method (for Dynamic Service Activator or PCMM) to access methods of that type. By default, gateway clients have access only

to methods and scripts that you specify in the configuration. Access control should be disabled only for troubleshooting purposes.

Configuring Subscriber Types for Dynamic Service Activator

You configure which types of information identify subscribers to the SAE. The subscriber types that you can configure are the same subscriber types that you can use in applications created with the SAE CORBA remote API.

To configure subscriber types:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>Subscriber Types**.
2. (Optional) In the Create new list, select **Subscriber Types**.

   Type a name for the new configuration in the dialog box, and click **OK**.

   The configuration appears in the side pane.
3. Select the Subscriber Types configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring the NIC Proxies for Dynamic Service Activator

You create a NIC proxy for each subscriber type to be configured. The name of the NIC proxy must match the name configured for the NIC proxy namespace.

Subscriber types that have different subscriber ID types can use the same NIC proxy. For example, a subscriber type configured as SubscriberType1 that has a subscriber ID type of interface-name, and a subscriber type configured as subscriberType2 that has a subscriber ID type of interface-index can both use the same NIC proxy. Likewise, a subscriber type configured as SubscriberType1 and a subscriber type configured as subscriberType2 that both have a subscriber ID type of address can use the same NIC proxy.

To configure NIC proxies:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>NIC Proxy Configuration**.
2. (Optional) In the Create new list, select **NIC Proxies**.

   Type a name for the new configuration in the dialog box, and click **OK**.

   The configuration appears in the side pane.
3. Expand the NIC Proxies configuration, select the NIC proxy properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Access to Methods and Scripts for Dynamic Service Activator

Configuring access to methods and scripts involves adding methods, scripts, and clients to the configuration and configuring access properties between each client and each method or script.

NOTE: Client profiles are cached by Dynamic Service Activator for 30 minutes. If you change the password or role of a client that has been used within the last 30 minutes, it can take up to 30 minutes before these changes take effect.

When permissions are configured, roles are assigned to application server user objects automatically. The first time you add a method or script for a client, the DSA role is added to the corresponding application server user; and when the last method or script is deleted, the DSA role is removed from the corresponding user. Only role and password changes take up to 30 minutes to take effect.

If you do not want to wait 30 minutes for the changes to take effect, restart the Web application server.

Dynamic Service Activator interacts with the Web application server to determine whether a gateway client has access to a method or script. The name and credentials, such as a password, that are used to authenticate the gateway client are configured on the Web application server as user accounts.

Access constraints are regular expressions that the arguments for the method or script in the SOAP request must match. If the arguments for the method or script in a particular SOAP request do not match these regular expressions, then Dynamic Service Activator rejects the request.

Configuring Access to Methods for Dynamic Service Activator

To configure methods for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>Method**.
2. Expand the Method configuration, select the properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

To configure access to methods for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Client**.

   Type a name for the new configuration in the dialog box, and click **OK**. You must use the same name for the gateway client that is configured on the Web application server.

The configuration appears in the side pane.

3. Expand **Client>Permissions**.
4. In the Create new list, select the method you want to configure. Expand the Method configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Access to Scripts for Dynamic Service Activator

To configure scripts for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Script**.

   Type a name for the new configuration in the dialog box, and click **OK**. Use the text string that exactly matches the name of the script.

   The configuration appears in the side pane.
3. Expand the Script configuration, select the properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

To configure access to scripts:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Client**.

   Type a name for the new configuration in the dialog box, and click **OK**. You must use the same name for the gateway client that is configured on the Web application server.

   The configuration appears in the side pane.
3. Expand **Client>Permissions**.
4. In the Create new list, select **Script**.

   Type a name for the new configuration in the dialog box, and click **OK**. Use the text string that exactly matches the name of the script.

   The configuration appears in the side pane.
5. Expand the Script configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring the Logging Destinations to Store Log Messages in a File

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

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. Expand the Logger:file configuration, click **File**, enter information as described in the Help text in the main pane, and click **Apply**.

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

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. In the Create new list, select **Logger**.
3. Type a name for the new logger in the dialog box, and click **OK**.

   The logger appears in the side pane.
4. Expand the Logger configuration, click **File**, enter information as described in the Help text in the main pane, and click **Create**.

Configuring the Logging Destinations to Send Log Messages to the System Logging Facility

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

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. Expand the Logger configuration, click **Syslog**, enter information as described in the Help text in the main pane, and click **Apply**.

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

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. In the Create new list, select **Logger**.
3. Type a name for the new logger in the dialog box, and click **OK**.

   The logger appears in the side pane.
4. Expand the Logger configuration, click **Syslog**, enter information as described in the Help text in the main pane, and click **Create**.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
- Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
- Viewing Information About NIC Proxies (C-Web Interface) on page 52

Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface)

Virtual hosts for Web applications are configured in the application server.

To deploy the application on the virtual host:

1. Click **Configure>Slot>Slot:0>DSA>Deploy**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Related Documentation

- Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
- Starting Dynamic Service Activator (C-Web Interface) on page 45
- Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
- Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- *Web Application Server on C Series Controllers Overview*
- *Configuring Virtual Hosts for the Web Applications (SRC CLI)*

Starting Dynamic Service Activator (C-Web Interface)

You must configure a group before you enable Dynamic Service Activator. To deploy Dynamic Service Activator, you must specify the virtual host, and the Web application server must be running.

To start Dynamic Service Activator:

1. Click **Manage>Enable**.

   The Enable pane appears.

2. From the Component list, select **dsa**, and click **OK**.

Related Documentation

- Before You Use Dynamic Service Activator on page 14
- Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) on page 44
- Starting Dynamic Service Activator (SRC CLI) on page 33
- Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40

CHAPTER 5

Monitoring Dynamic Service Activator (SRC CLI)

SRC CLI Commands to Monitor Dynamic Service Activator

You can view statistics for Dynamic Service Activator and information about SOAP operations and NIC proxies. Table 4 on page 47 lists the commands you use to monitor Dynamic Service Activator.

Table 4: Commands to Monitor Dynamic Service Activator

Command	Output Displayed
show dsa statistics general	General statistics for Dynamic Service Activator.
show dsa statistics soap-operation	Information about SOAP operations for Dynamic Service Activator.
monitor dsa soap-operation ***operation-name***	Real-time statistics about SOAP operations for Dynamic Service Activator.
show dsa statistics nic-proxy	Information about NIC proxies for Dynamic Service Activator.
monitor dsa nic-proxy ***proxy-name***	Real-time statistics about NIC proxies for Dynamic Service Activator.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Starting Dynamic Service Activator (SRC CLI) on page 33
- *Output Control Keys for monitor Command*

- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Viewing Statistics for Dynamic Service Activator (SRC CLI)

Purpose View general statistics for Dynamic Service Activator with the SRC CLI.

Action

```
user@host> show dsa statistics general
```

Related Documentation

- Dynamic Service Activator Overview on page 7
- Starting Dynamic Service Activator (SRC CLI) on page 33

Viewing Information About SOAP Operations (SRC CLI)

Purpose View information about SOAP operations for Dynamic Service Activator.

Action To display information about a specific SOAP operation:

1. Click **Monitor>DSA>Statistics>SOAP Operation**.

   The Statistics/SOAP Operation pane appears.
2. From the Operation Name list, select the operation for which you want to display information.
3. Click **OK**.

   The Statistics/SOAP Operation pane displays information about the SOAP operation.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Web Services Gateway Overview on page 3
- SOAP Fault Codes for Dynamic Service Activator on page 74
- Monitoring SOAP Operations (SRC CLI) on page 48

Monitoring SOAP Operations (SRC CLI)

Purpose Display real-time statistics about SOAP operations for Dynamic Service Activator. You can monitor SOAP operations.

Action To display real-time statistics about SOAP operations for Dynamic Service Activator:

```
user@host> monitor dsa soap-operation operation-name operation-name
```

Related Documentation

- Dynamic Service Activator Overview on page 7
- Web Services Gateway Overview on page 3
- Viewing Information About SOAP Operations (SRC CLI) on page 48
- *Output Control Keys for monitor Command*

Viewing Information About NIC Proxies (SRC CLI)

Purpose View information about NIC proxies for Dynamic Service Activator.

Action To display information about a specific NIC proxy:

```
user@host> show dsa statistics nic-proxy proxy-name proxy-name
```

For more information about NIC proxies, see the *SRC PE Network Guide*.

Related Documentation

- Monitoring NIC Proxies (SRC CLI) on page 49

Monitoring NIC Proxies (SRC CLI)

Purpose Display real-time statistics about NIC proxies for Dynamic Service Activator.

Action To display real-time statistics about NIC proxies for Dynamic Service Activator:

```
user@host> monitor dsa nic-proxy proxy-name proxy-name
```

For more information about NIC proxies, see the *SRC PE Network Guide*.

Related Documentation

- Viewing Information About NIC Proxies (SRC CLI) on page 49
- *Output Control Keys for monitor Command*

CHAPTER 6

Monitoring Dynamic Service Activator (C-Web Interface)

Viewing Statistics for Dynamic Service Activator (C-Web Interface)

Purpose View general statistics for Dynamic Service Activator.

Action Click **Monitor>DSA>Statistics>General**.

The Statistics/General pane appears.

Related Documentation

- Dynamic Service Activator Overview on page 7
- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48
- Starting Dynamic Service Activator (C-Web Interface) on page 45

Viewing Information About SOAP Operations (C-Web Interface)

Purpose View information about SOAP operations for Dynamic Service Activator.

Action To display information about a specific SOAP operation:

1. Click **Monitor>DSA>Statistics>SOAP Operation**.

   The Statistics/SOAP Operation pane appears.
2. From the Operation Name list, select the operation for which you want to display information.
3. Click **OK**.

   The Statistics/SOAP Operation pane displays information about the SOAP operation.

Related Documentation

- Dynamic Service Activator Overview on page 7

- Web Services Gateway Overview on page 3
- Viewing Information About SOAP Operations (SRC CLI) on page 48

Viewing Information About NIC Proxies (C-Web Interface)

Purpose View information about NIC proxies for Dynamic Service Activator.

Action To display information about a specific NIC proxy:

1. Click **Monitor>DSA>Statistics>NIC Proxy**.

   The Statistics/NIC Proxy pane appears.

2. From the Proxy Name list, select the NIC proxy for which you want to display information.

3. Click **OK**.

   The Statistics/NIC Proxy pane displays information about the NIC proxy.

Related Documentation

- Viewing Information About NIC Proxies (SRC CLI) on page 49
- Monitoring NIC Proxies (SRC CLI) on page 49

CHAPTER 7

Testing Dynamic Service Activator (SRC CLI)

Testing the Web Application Gateway Client

The SRC software includes a Web application gateway client that you can use to test Dynamic Service Activator's ability to invoke methods and scripts for:

- Dynamic Service Activator services

  For more information about the methods for these services, see "Methods for the Dynamic Service Activator Web Service Interface" on page 66.

- PCMM services

  For more information about the methods for these services, see "SRC PCMM Web Service Interface Methods" on page 79.

Related Documentation

- API for Dynamic Service Activator on page 65
- Configuring the Test Environment for Dynamic Service Activator Services on page 54
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 56
- Configuring the Test Environment for PCMM Services on page 59
- Running Methods for PCMM Services (SRC CLI) on page 61

Configuring the Test Environment for Dynamic Service Activator Services

Configuring the settings for your test environment is optional. You can choose to enter the settings each time you test the Web client or to configure the test settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

Configuring Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to configure the testing environment for the Dynamic Service Activator Web service interface:

test dsa dsa-service environment set <client-id ***client-id***> <client-password ***client-password***> <subscriber-id ***subscriber-id***> <subscriber-password ***subscriber-password***> <host ***host***> <port ***port***>

To configure the settings for the test environment:

1. Issue the **test dsa dsa-service environment set** command.

   To configure the settings for the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment set** command.

2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the username that the SAE uses to authenticate this subscriber, use the **subscriber-id** option.
5. (Optional) To specify the password that the SAE uses to authenticate this subscriber, use the **subscriber-password** option.
6. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
7. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Configuring the Subscriber URI for Dynamic Service Activator Services (SRC CLI)

Use the following command to configure the subscriber Uniform Resource Identifier (URI) in the testing environment for the Dynamic Service Activator Web service interface:

test dsa dsa-service environment set subscriber-uri <***subscriber-uri***> <subscriber-type ***subscriber-type***> <subscriber-address ***subscriber-address***> <login-name ***login-name***> <dn ***dn***> <virtual-router ***virtual-router***> <interface-name ***interface-name***> <interface-index ***interface-index***> <primary-user-name ***primary-user-name***> <session-handle ***session-handle***> <subscriber-constraints ***subscriber-constraints***>

To configure the settings for the test environment:

1. Issue the **test dsa dsa-service environment set subscriber-uri** command.

   To configure the settings for the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment set subscriber-uri** command.

2. (Optional) To specify the subscriber URI value in the same format as the syntax for the subURI argument, use ***subscriber-uri***. See "Format of the Subscriber's URI" on page 70.

   For example: **test dsa dsa-service environment set subscriber-uri ip:ipAddress=1.2.3.4**

3. (Optional) To specify the name of the configured subscriber type used to construct the subscriber URI, use the **subscriber-type** option.

4. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option. This value is mandatory when the subscriber ID type is **address** or **address-interface-name**.

5. (Optional) To specify the name with which the subscriber logs in, use the **login-name** option. This value is mandatory when the subscriber ID type is **login-name**.

6. (Optional) To specify the DN of the subscriber profile, use the **dn** option. This value is mandatory when the subscriber ID type is **dn**.

7. (Optional) To specify the name of the virtual router associated with the subscriber, use the **virtual-router** option. This value is mandatory when the subscriber ID type is **interface-name** or **interface-index**, optional when the subscriber ID type is **address-interface-name**.

8. (Optional) To specify the name of the interface on which the subscriber logs in, use the **interface-name** option. This value is mandatory when the subscriber ID type is **interface-name** and is optional when the subscriber ID type is **address-interface-name**.

9. (Optional) To specify the SNMP index of the interface on which the subscriber logs in, use the **interface-index** option. This value is mandatory when the subscriber ID type is **interface-index**.

10. (Optional) To specify the primary username, use the **primary-user-name** option. This value is mandatory when the subscriber ID type is **primary-user-name**.

11. (Optional) To specify the subscriber session handle, use the **session-handle** option. This value is mandatory when the subscriber ID type is **session-handle**.

12. (Optional) To specify the constraints for the NIC key, use the **subscriber-constraints** option.

Verifying Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to verify the settings for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment show
```

Use the following commands to verify specific settings:

```
test dsa dsa-service environment show client-id
test dsa dsa-service environment show client-password
test dsa dsa-service environment show host
test dsa dsa-service environment show port
test dsa dsa-service environment show subscriber-id
test dsa dsa-service environment show subscriber-password
test dsa dsa-service environment show subscriber-uri
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment show** commands to verify the settings.

Deleting Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to delete the settings for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa dsa-service environment clear client-id
test dsa dsa-service environment clear client-password
test dsa dsa-service environment clear host
test dsa dsa-service environment clear port
test dsa dsa-service environment clear subscriber-id
test dsa dsa-service environment clear subscriber-password
test dsa dsa-service environment clear subscriber-uri
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment clear** commands to delete the settings.

Related Documentation

- Testing the Web Application Gateway Client on page 53
- Sample Data for Dynamic Service Activator on page 34
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 56

Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI)

To run a method or script for the Dynamic Service Activator Web service interface:

- Run the **test dsa dsa-service** command.

```
user@host> test dsa dsa-service (invoke-gateway-extension | invoke-script |
  subscriber-activate-service | subscriber-deactivate-service | subscriber-login |
  subscriber-logout | subscriber-modify-service | subscriber-read-subscription)
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** command.

```
user@host> test dsa dsa2-service (invoke-gateway-extension | invoke-script |
  subscriber-activate-service | subscriber-deactivate-service | subscriber-login |
  subscriber-logout | subscriber-modify-service | subscriber-read-subscription)
```

where:

- **invoke-gateway-extension**—Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server.
- **invoke-script**—Manages all operations involved with invoking scripts.
- **subscriber-activate-service**—Activates subscribers' subscriptions to services.
- **subscriber-deactivate-service**—Deactivates subscribers' subscriptions to services.
- **subscriber-login**—Logs in subscribers.
- **subscriber-logout**—Logs out subscribers.
- **subscriber-modify-service**—Modifies subscriptions.
- **subscriber-read-subscription**—Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions.

Testing Subscriber Logins and Logouts (SRC CLI)

These methods support only subscribers who are identified by their IP addresses. These methods do not support subscribers who are identified by the names they use to log in or by their DNs.

Use the following commands to test the methods that log in and log out subscribers:

```
test dsa dsa-service subscriber-login <subscriber-uri subscriber-uri> <subscriber-id
  subscriber-id> <subscriber-password subscriber-password> <client-id client-id>
  <client-password client-password> <host host> <port port>
```

```
test dsa dsa-service subscriber-logout <subscriber-uri subscriber-uri> <client-id client-id>
  <client-password client-password> <host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscriber Access to Subscriptions (SRC CLI)

Use the following command to test the method that determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions:

test dsa dsa-service subscriber-read-subscription <subscriber-uri *subscriber-uri*> <attributes *attributes*> <filter *filter*> <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscription Activations and Deactivations (SRC CLI)

Use the following commands to test the methods that activate and deactivate subscribers' subscriptions to services:

test dsa dsa-service subscriber-activate-service <subscriber-uri *subscriber-uri*> service-name *service-name* <service-session *service-session*> <accounting-tag *accounting-tag*> <downstream-bandwidth *downstream-bandwidth*> <upstream-bandwidth *upstream-bandwidth*> <session-timeout *session-timeout*> <subscription-user *subscription-user*> <subscription-password *subscription-password*> <substitutions *substitutions*> <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

test dsa dsa-service subscriber-deactivate-service <subscriber-uri *subscriber-uri*> service-name *service-name* <service-session *service-session*> <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscription Modifications (SRC CLI)

Use the following command to test the method that modifies subscribers' subscriptions to services:

test dsa dsa-service subscriber-modify-service <subscriber-uri *subscriber-uri*> service-name *service-name* <service-session *service-session*> <accounting-tag *accounting-tag*> <downstream-bandwidth *downstream-bandwidth*> <upstream-bandwidth *upstream-bandwidth*> <session-timeout *session-timeout*> <subscription-user *subscription-user*> <subscription-password *subscription-password*> <substitutions *substitutions*> <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Script Invocations (SRC CLI)

Use the following command to test the method that manages all operations involved with invoking scripts:

test dsa dsa-service invoke-script sae-script-name *sae-script-name* sae-script-arguments *sae-script-arguments* <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

This method retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client.

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service invoke-script** command.

Testing Gateway Extension Invocations (SRC CLI)

Use the following command to test the method that invokes a servlet that has been created and deployed in the Web Services Gateway Web application server:

test dsa dsa-service invoke-gateway-extension gateway-extension-name *gateway-extension-name* gateway-extension-arguments *gateway-extension-arguments* <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

The servlet can be a standalone application, or it can be part of a WAR or EAR file. When deployed, servlets invoked with this method should be accessible only from the local host.

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service invoke-gateway-extension** command.

Example: Testing Subscriber Access to Subscriptions

To view a list of the Dynamic Service Activator client's subscriptions:

1. Issue the **test dsa dsa-service subscriber-read-subscription** command.
2. Enter the required information (such as client ID, client password, and subscriber address). The entered data must match the data you configured.

   For example, this information is provided for Fred:

```
user@host> test dsa dsa-service subscriber-read-subscription client-id Fred
  client-password secret subscriber-address 10.19.1.6 subscriber-type ip attributes
  "serviceName"
  The Subscriber_readSubscription method was successfully performed for
  subURI: ip:ipAddress=10.19.1.6&timestamp=1216221624715
  serviceName: COAService
```

Related Documentation

- Testing the Web Application Gateway Client on page 53
- Sample Data for Dynamic Service Activator on page 34
- Configuring the Test Environment for Dynamic Service Activator Services on page 54

Configuring the Test Environment for PCMM Services

Configuring the settings for your test environment is optional. You can choose to enter the settings in the DSA SOAP Client window each time you use the Web client or to

configure the default settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

Configuring Settings for PCMM Services (SRC CLI)

Use the following command to configure the testing environment for the PCMM Web service interface:

```
test dsa pcmm-service environment set <client-id client-id> <client-password
  client-password> <subscriber-address subscriber-address> <subscriber-uri subscriber-uri>
```

To configure the settings for the test environment:

1. Issue the **test dsa pcmm-service environment set** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
5. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.

Verifying Settings for PCMM Services (SRC CLI)

Use the following command to verify the settings for the PCMM Web service interface:

```
test dsa pcmm-service environment show
```

Use the following commands to verify specific settings:

```
test dsa pcmm-service environment show client-id
test dsa pcmm-service environment show client-password
test dsa pcmm-service environment show subscriber-address
test dsa pcmm-service environment show subscriber-uri
```

Deleting Settings for PCMM Services (SRC CLI)

Use the following commands to delete the settings for the PCMM Web service interface:

```
test dsa pcmm-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa pcmm-service environment clear client-id
test dsa pcmm-service environment clear client-password
test dsa pcmm-service environment clear subscriber-address
test dsa pcmm-service environment clear subscriber-uri
```

Related Documentation

- Testing the Web Application Gateway Client on page 53
- Sample Data for Dynamic Service Activator on page 34
- Configuring the Test Environment for Dynamic Service Activator Services on page 54
- Running Methods for PCMM Services (SRC CLI) on page 61

Running Methods for PCMM Services (SRC CLI)

To run a method for the PCMM Web service interface:

- Run the **test dsa pcmm-service** command.

  ```
  user@host> test dsa pcmm-service (commit-resources | release-resources |
    query-contexts | query-available-services)
  ```

  where:

  - **commit-resources**—Specifies the resources that are being requested in the CommitResource message.
  - **release-resources**—Specifies the resources that are being requested to be released in the ReleaseResources message.
  - **query-contexts**—Searches for the context ID and context status for a subscriber.
  - **query-available-services**—Searches for the services that are available to the calling application.

- Testing Resource Requests (SRC CLI) on page 61
- Testing Resource Release Requests (SRC CLI) on page 62
- Testing Queries for Subscriber Contexts (SRC CLI) on page 63
- Testing Queries for Available Services (SRC CLI) on page 63

Testing Resource Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested in the CommitResource message:

```
test dsa pcmm-service commit-resources <subscriber-address subscriber-address>
  <subscriber-uri subscriber-uri> service-name service-name <context-id context-id>
  <time-usage-limit time-usage-limit> <classifier classifier> <traffic-profile traffic-profile>
  <flow-spec flow-spec> <client-id client-id> <client-password client-password>
```

To test the method:

1. Issue the **test dsa pcmm-service commit-resources** command.
2. To specify the name of the SRC service, use the **service-name** option.
3. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
4. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.

5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify a limit on the lifetime of a context, use the **time-usage-limit** option. An application server may specify multiple time usage limits to request different limits in the upstream and downstream directions. If the application server does not specify a time usage limit, the application manager determines the time usage limit.
7. (Optional) To specify the object that identifies the traffic flow for which the application server is requesting services, use the **classifier** option.
8. (Optional) To specify information about the bandwidth and QoS characteristics desired for a request, use the **traffic-profile** option. You express the traffic profile by configuring the SRC policies with FlowSpec, service class name, or DOCSIS actions.
9. (Optional) To specify a FlowSpec action, use the **flow-spec** option.
10. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
11. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Resource Release Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested to be released in the ReleaseResources message:

test dsa pcmm-service release-resources <subscriber-address *subscriber-address*> <subscriber-uri *subscriber-uri*> <service-name *service-name*> <context-id *context-id*> <client-id *client-id*> <client-password *client-password*>

To test the method:

1. Issue the **test dsa pcmm-service release-resources** command.
2. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
3. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.
4. (Optional) To specify the name of the SRC service, use the **service-name** option.
5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
7. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Queries for Subscriber Contexts (SRC CLI)

Use the following command to test the method that searches for the context ID and context status for a subscriber:

test dsa pcmm-service query-contexts <subscriber-address ***subscriber-address***> <subscriber-uri ***subscriber-uri***> <service-name ***service-name***> <context-id ***context-id***> <client-id ***client-id***> <client-password ***client-password***>

To test the method:

1. Issue the **test dsa pcmm-service query-contexts** command.
2. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
3. (Optional) To specify the subscriber's URI, use the **subscriber-uri** option.
4. (Optional) To specify the name of the SRC service, use the **service-name** option.
5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
7. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Queries for Available Services (SRC CLI)

Use the following command to test the method that searches for the services that are available to the calling application:

test dsa pcmm-service query-available-services <client-id ***client-id***> <client-password ***client-password***>

To test the method:

1. Issue the **test dsa pcmm-service query-available-services** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Related Documentation

- Testing the Web Application Gateway Client on page 53
- Sample Data for Dynamic Service Activator on page 34
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 56
- Configuring the Test Environment for PCMM Services on page 59

CHAPTER 8

Developing Gateway Clients

API for Dynamic Service Activator

This topic contains information that developers need to create gateway clients and that administrators need to manage gateway clients and their interactions with the gateway.

Public SOAP Interfaces of Web Applications

When you have installed Dynamic Service Activator, you can access a Web Services Description Language (WSDL) file for the application. The WSDL file defines the SOAP properties that you or your customers can use to develop a gateway client. The URL for Dynamic Service Activator is:

http://<host>:<portNumber>:/dsa/services/DynamicServiceActivation?wsdl

- **<host>**—IP address or name of the host that supports Dynamic Service Activator
- **<portNumber>**—Number of the TCP port

There is an implementation of Dynamic Service Activator that uses the document literal encoding style (DSA2). The URL for this implementation is:

http://<host>:<portNumber>:/dsa/services/DSA2?wsdl

- **<host>**—IP address or name of the host that supports Dynamic Service Activator
- **<portNumber>**—Number of the TCP port

Related Documentation

- Dynamic Service Activator Overview on page 7
- Web Services Gateway Overview on page 3
- Sample Data for Dynamic Service Activator on page 34
- Methods for the Dynamic Service Activator Web Service Interface on page 66

- SOAP Fault Codes for Dynamic Service Activator on page 74

Methods for the Dynamic Service Activator Web Service Interface

This topic describes the methods associated with Dynamic Service Activator, and provides information additional to that in the WSDL file.

invokeGwExtension

- Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server. The servlet can be a standalone application, or it can be part of a WAR or EAR file.

  When deployed, servlets invoked with invokeGwExtension should be accessible only from the local host.

- Arguments
  - extensionName—String that contains the name of the servlet that the gateway client invokes
  - extensionArguments—String array of arguments that the gateway client passes to the servlet
- Guidelines—The names in the following components and properties must be the same as the name of the extensionName argument:
  - The name of the WAR file that is the gateway extension servlet
  - In the *WEB-INF/web.xml* file in the servlet section, the servlet name
  - In the *WEB-INF/web.xml* file in the servlet-mapping section, the URL pattern in the format */servlet/<extensionName>*
- Expected output—String returned by the extension
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

invokeScript

- Manages all operations involved with invoking scripts: retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client. For a complete description of Dynamic Service Activator's interactions with the gateway client and other components, see "Dynamic Service Activator Overview" on page 7.
- Arguments
  - scriptName—String that contains the name of the script that the gateway client wants to invoke
  - scriptArgs—String array of arguments that the gateway client passes to the script
- Expected output—String returned by the script
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_readSubscription

- Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions; returns the result in a two-dimensional array.
- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
  - select—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

    For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks website at http://www.juniper.net/techpubs/software/management/sdx/api-index.html (the sae.Select structure).
- Expected output—Multidimensional array of Attr objects that contain the subscriptions for the subscriber
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_readSubscription_retAttrSeq

- Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions; returns the result in a one-dimensional array for SOAP clients that do not support two-dimensional arrays.
- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
  - select—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

    For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks website at http://www.juniper.net/techpubs/software/management/sdx/api-index.html (the sae.Select structure).
- Expected output—One-dimensional array of AttrSeq objects that contain the subscriptions for the subscriber
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_activateService

- Activates subscribers' subscriptions to services.
- Arguments

- subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
- subscriptionName—String that contains the name of the subscription
- sessionName—String that contains the name of the service session; default string is *default*
- activationAttributes—Array of one or more of the following attributes that can be specified for the subscription:
  - sessionTimeout
  - downStreamBandwidth
  - upStreamBandwidth
  - sessionTag
  - subscriptionUsername
  - subscriptionPassword
  - substitutions
  - serviceIdentifier
  - ratingGroup
  - grantTotalOctets
  - grantOutOctets
  - grantInOctets

NOTE: **The serviceIdentifier, ratingGroup, grantTotalOctets, grantOutOctets, and grantInOctets attributes are specific for Gx services.**

For information about these attributes, see "Subscription Attributes" on page 72.

- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_deactivateService

- Deactivates subscribers' subscriptions to services.
- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
  - subscriptionName—String that contains the name of the subscription
  - sessionName—String that contains the name of the service session; default string is *default*

- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_modifyService

- Modifies subscriptions.
- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
  - subscriptionName—String that contains the name of the subscription
  - sessionName—String that contains the name of the service session; default string is *default*
  - optionalAttributes—Array of one or more of the following attributes that can be modified for the subscription:
    - sessionTimeout
    - downStreamBandwidth
    - upStreamBandwidth
    - sessionTag
    - substitutions
    - serviceIdentifier
    - ratingGroup
    - grantTotalOctets
    - grantOutOctets
    - grantInOctets

    NOTE: **The serviceIdentifier, ratingGroup, grantTotalOctets, grantOutOctets, and grantInOctets attributes are specific for Gx services.**

    For information about these attributes, see "Subscription Attributes" on page 72.
- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_login

- Logs in subscribers

NOTE: This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNs.

- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
- Expected output—Boolean operator that indicates success
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Subscriber_logout

- Logs out subscribers. This method supports only subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNs.
- Arguments
  - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 70)
- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 74

Format of the Subscriber's URI

Many of Dynamic Service Activator's methods require the argument subURI, the subscriber's Uniform Resource Identifier (URI). This argument comprises two portions: the type of subscriber (subscriber-type) and a list of the subscriber's attributes (subscriber-comp). The syntax for the subURI argument is:

<subscriber-type>:<subscriber-comp> [& <subscriber-comp>]*

The <subscriber-type> variable is the name of the Subscriber Type instance that is defined in the directory during the Dynamic Service Activator configuration. For example, the sample data provides these Subscriber Type instances for the <subscriber-type> value:

- ip—The sidType is SIT_ADDRESS
- dn—The sidType is SIT_DN
- login—The sidType is SIT_LOGIN_NAME
- assignedIp—The sidType is SIT_ADDR_IF_NAME
- intName—The sidType is SIT_IF_NAME
- intIndex—The sidType is SIT_IF_INDEX

The <subscriber-comp> variable has the format <type>=<value>.

- <type> can be one of the following:
  - ipAddress—Subscriber's IP address; mandatory when the sidType is SIT_ADDRESS or SIT_ADDR_IF_NAME
  - timestamp—Time at which the request was sent; optional when the sidType is SIT_ADDRESS
  - dn—Subscriber's DN; mandatory when the sidType is SIT_DN
  - intfName—Name of the interface on which the subscriber logs in; mandatory when the sidType is SIT_IF_NAME, optional when the sidType is SIT_ADDR_IF_NAME
  - vrfName—Name of the VRF associated with the subscriber; mandatory when the sidType is SIT_IF_NAME or SIT_IF_INDEX, optional when the sidType is SIT_ADDR_IF_NAME
  - login_name—Name with which the subscriber logs in; mandatory when the sidType is SIT_LOGIN_NAME
  - intfIndex—SNMP index of the interface on which the subscriber logs in; mandatory when the sidType is SIT_IF_INDEX
  - primary_user_name—Primary username; mandatory when the sidType is SIT_PRIMARY_USER_NAME
- <value> can be a combination of the following:
  - <operator>—One of the following:

    ; | / | ? | : | @ | & | = | + | $ | ,
  - <textString>—Text string that uses the following characters:
    - Lowercase letters
    - Uppercase letters
    - Arabic numerals
    - - | _ | . | ! | ~ | * | ' | (|)
  - % <hexNumber>—2-character hexadecimal number

For example, you can use the following subscriber's URI to look up a subscriber by IP address as specified by the sample data:

ip:ipAddress=192.168.1.10

The sample data defines a subscriber type named ip, whose sidType is SIT_ADDRESS and nicProxyNamespace is /nicProxies/ip. As a result, ip is the value of the <subscriber-type> variable. Because the sidType is SIT_ADDRESS, ipAddress is mandatory for the <type> component, and the subscriber's IP address is the <value> component of the <subscriber-comp> variable.

Related Documentation

- API for Dynamic Service Activator on page 65
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
- Methods for the Dynamic Service Activator Web Service Interface on page 66
- Subscription Attributes on page 72
- SOAP Fault Codes for Dynamic Service Activator on page 74

Subscription Attributes

Some methods take attributes for subscriptions as follows.

sessionTimeout

- Timeout for the service.
- Value—Number of seconds in the range –1 through 2147483647
- Guideline— –1 indicates no timeout.
- Example—600

downStreamBandwidth

- Traffic rate between the subscriber and the network.
- Value—Number of bits per second in the range 0–2147483647
- Example—10000

upStreamBandwidth

- Traffic rate between the network and subscriber.
- Value—Number of bits per second in the range 0–2147483647
- Example—5000

sessionTag

- Tag that the software uses to track a session for accounting purposes.
- Value—Text string
- Example—News:Joe

subscriptionUsername

- Name of the subscriber to the service.
- Value—Text string
- Example—Joe

subscriptionPassword

- Password for the service.
- Value—Text string
- Example—Secret

substitutions

- Attributes and values that the method should substitute for existing settings.
- Value—Array of strings; each array has the format <substitutionType>=<substitutionValue>
- Example—Port=9999

serviceIdentifier

- Name of the service identifier to the service.
- Value—The value ranges from –0 through 4,294,967,295
- Example—600

ratingGroup

- Rating Group associated with the service.
- Value—The value ranges from –0 through 4,294,967,295
- Example—600

grantTotalOctets

- Total number of input and output octet threshold value for the granted service unit.
- Value—Number of octets in the range –0 through 9,223,372,036,854,775,807
- Example—600

grantOutOctets

- Total number of output octet threshold value for the granted service unit.
- Value—Number of octets in the range –0 through 9,223,372,036,854,775,807
- Example—600

grantInOctets

- Total number of input octet threshold value for the granted service unit.
- Value—Number of octets in the range –0 through 9,223,372,036,854,775,807
- Example—600

NOTE: You can configure the granted service unit attributes by using the input-octets, output-octets, or total-octets or by using both input-octets and output-octets. The **serviceIdentifier**, **ratingGroup**, **grantTotalOctets**, **grantOutOctets**, and **grantInOctets** attributes are specific for Gx service subscriptions.

SOAP Fault Codes for Dynamic Service Activator

When Dynamic Service Activator receives a SOAP request that it cannot handle, it returns a SOAP fault message to the gateway client. This message contains a text string that specifies a SOAP fault code and a text string that provides additional information about the fault.

Dynamic Service Activator returns the following SOAP fault codes. If the SOAP fault code has the format client.<variables>, you must correct the gateway client. However, if the SOAP fault code has the format server.<variables>, record the error code and notify the gateway administrator.

Client.InvalidArguments

- Specifies that one or more of the arguments for the script or method is an empty string or has a null value.
- Classification of call—Denied
- Action—Check that all arguments are correct.

Client.InvalidSubscriberFormat

- Specifies that the subURI argument, which the client passes to all methods except invokeScript, is invalid.
- Classification of call—Denied
- Action—Check that subURI arguments are correct.

Server.AccessDenied

- Specifies that the client does not have permission to make the request because of one or more of the following errors:
  - Web Services Gateway does not recognize the client.
  - Supplied name of the method or script is incorrect.

- Number or value of the arguments supplied for the script or method is incorrect.

- Classification of call—Denied
- Action—Check that the values are correct. If they are, contact the gateway administrator.

Server.AccessControlMisconfiguration

- Specifies that the gateway administrator did not configure Dynamic Service Activator correctly, in one of the following ways:
  - The saeLocatorArg is not an integer.
  - The number of arguments is less than the value of the saeLocatorArg parameter.
  - The regular expressions are not correctly defined.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.Misconfiguration

- Specifies that the gateway administrator did not configure the Web application to authenticate SOAP clients, and an unauthenticated client has passed a request to Dynamic Service Activator, which refuses unauthenticated requests.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAEUnreachable

- Specifies that the gateway client either cannot identify the SAE server or cannot reach it through CORBA. This SAE can be one on which the subscriber has logged in or one on which the client wants to invoke a script.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.UnknownSAEUser

- Specifies that the SAE on which Dynamic Service Activator tried to run a method or script cannot identify the subscriber. Dynamic Service Activator may have retained in its cache an SAE that is no longer current for the subscriber.
- Classification of call—Failed
- Action—Try this operation again. If it fails again, contact the gateway administrator.

Server.SAE.UserNotUniqueToSAE

- Specifies that the subscriber is already active on the SAE on which Dynamic Service Activator tried to run a method or script. Dynamic Service Activator may have retained in its cache an SAE that is no longer current.
- Classification of call—Failed
- Action—Try this operation again. If it fails again, contact the gateway administrator.

Server.SAE.UnknownService

- Specifies that the SAE cannot identify the service that a subscription specifies.
- Classification of call—Failed
- Action—Check the service parameter for the following methods:
  - Subscriber_activateService
  - Subscriber_deactivateService
  - Subscriber_modifyService

Server.SAE.UnknownSubscription

- Specifies that the SAE cannot identify the subscription that the subscriber wants to activate.
- Classification of call—Failed
- Action—Check the service parameter for the following methods:
  - Subscriber_activateService
  - Subscriber_deactivateService
  - Subscriber_modifyService

Server.SAE.ServiceAuthenticationError

- Specifies that the service the subscriber wants to activate requires authentication.
- Classification of call—Failed
- Action—Check the activation attributes. If they are correct, contact the gateway administrator.

Server.SAE.UnknownServiceSession

- Specifies that the SAE cannot identify the service session that the subscriber wants to modify.
- Classification of call—Failed
- Action—Check the serviceId and sessionName parameters.

Server.SAE.LoginError

- Specifies that the subscriber could not log in successfully.
- Classification of call—Failed
- Action—Check the subURI, username, and userPassword arguments.

Server.SAE.Exception

- Specifies that the SAE raised an exception after Dynamic Service Activator tried to run a method or script on the SAE.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.Overload

- Specifies that the SAE raised an overload exception after Dynamic Service Activator tried to run a method or script on the SAE.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.ScriptProcessorError

- Specifies that the SAE's script processor failed to invoke the script. This error could occur for many reasons, such as:
  - Gateway administrator did not deploy the script on the SAE host.
  - Gateway administrator did not configure the script processor correctly.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.ScriptExecutionError

- Specifies that the script processor invoked the script, but the script was not completed successfully.
- Classification of call—Failed
- Action—Contact the gateway administrator.

CHAPTER 9

Activating PCMM Services with SOAP

Web Service Interface for PCMM Overview

PCMM Web Service Interface Specification (PKT-SP-MM-WS-101-051221) defines a common Web service SOAP/XML interface between a generic application server and a PCMM application manager. The interface allows an application server to dynamically request resources on the cable operator's access network. It provides operations for requesting, releasing, and querying network resources on the cable network. The SRC PCMM Web service interface supports the following operations, which are specified in the PCMM Web service interface:

- CommitResources
- ReleaseResources
- QueryAvailableServices
- QueryContexts

Each of these operations is implemented as a method in the SRC PCMM Web service interface.

Related Documentation

- Configuring Services That Are Available for PCMM Clients (SRC CLI) on page 84
- Configuring PCMM Policies and Parameter Substitutions (SRC CLI) on page 82
- SRC PCMM Web Service Interface Methods on page 79

SRC PCMM Web Service Interface Methods

This section describes the methods in the SRC PCMM Web service interface.

CommitResources

- Specifies the resources that are being requested in the CommitResource message.
- Argument—CommitResourcesReq
  - classifier—Object that identifies the traffic flow that the application server is requesting services for. The object contains:
    - protocol
    - sourceIpAddress
    - sourceIpMask
    - sourcePortStart
    - sourcePortEnd
    - destinationIpAddress
    - destinationIpMask
    - destinationPortStart
    - destinationPortEnd

    For more information, see "Configuring Classify-Traffic Conditions for Dynamic Service Activator" on page 82.
  - trafficProfile—Provides information about the bandwidth and QoS characteristics desired for a request. You express the traffic profile through SRC policies. You can configure the policies one of the following ways:
    - FlowSpec action—See "Configuring FlowSpec Actions for Dynamic Service Activator" on page 83.
    - Specifying a traffic class in the service class name action—See "Configuring Service Class Name Actions for Dynamic Service Activator" on page 83.
    - Bandwidth parameter—See "Configuring DOCSIS Actions for Dynamic Service Activator" on page 83.
  - TimeUsageLimit—Specifies, in seconds, a limit on the lifetime of a context. An application server may specify multiple TimeUsageLimit elements to request different limits in the upstream and downstream directions. If the application server does not specify a TimeUsageLimit, the application manager determines the TimeUsageLimit.
  - SubscriberID—IPv4 address or SubscriberURI
    - If a subscriber URI and an IPv4 address are provided, the IPv4 address is ignored.
    - If the IPv4 address is supplied and the subscriber URI is not supplied, then a subscriber URI is constructed with the given IPv4 address as an assignedIp subscriber type (sidType is SIT_ADDR_IF_NAME).
  - ServiceName—Name of SRC service
  - contextID—Globally unique identifier that the application manager must use if it is included in the message

The context ID is used as the SRC sessionName when the PCMM gateway activates a service.

- Expected output—CommitResourcesRsp. The CommitResourcesRsp object contains the ContextID of the activated service.

ReleaseResources

- Specifies the resources that are being requested to be released in the ReleaseResources message.
- Argument—ReleaseResourcesReq
  - Subscriber URI—IP address of the subscriber
  - ServiceName—Name of SRC service
  - contextID—Globally unique identifier that the application manager must use if it is included in the message

    The context ID is used as the SRC sessionName when the PCMM gateway activates a service.
- Expected output—ReleaseResourcesRsp

QueryAvailableServices

- Searches for the services that are available to the calling application.
- Expected output—QueryAvailableServicesRsp

QueryContexts

- Searches for the context ID and context status for a subscriber.
- Argument—QueryContextsReq
  - SubscriberID—IPv4 address or SubscriberURI
    - If a subscriber URI and an IPv4 address is provided, the IPv4 address is ignored.
    - If the IPv4 address is supplied and the subscriber URI is not supplied, then a subscriber URI is constructed with the given IPv4 address as an assignedIp subscriber type (sidType is SIT_ADDR_IF_NAME).
  - ServiceName—Name of SRC service
  - contextID—Globally unique identifier that the application manager must use if it is included in the message

The context ID is used as the SRC sessionName when the PCMM gateway activates a service.

- Expected output—QueryContextsRsp

Configuring PCMM Policies and Parameter Substitutions (SRC CLI)

This topic describes how to define your SRC policies and parameter substitutions so that they comply with the PCMM Web service interface. The sample data shows the types of policies and parameters that you can configure. View the sample data with the following command:

```
user@host> show configuration policies folder sample folder pcmm folder pcmm-ws
```

If you use parameter substitutions, the PCMM Web service interface provides the values for the substitutions. You must name your parameters as specified in this topic.

- Configuring Classify-Traffic Conditions for Dynamic Service Activator on page 82
- Configuring FlowSpec Actions for Dynamic Service Activator on page 83
- Configuring Service Class Name Actions for Dynamic Service Activator on page 83
- Configuring DOCSIS Actions for Dynamic Service Activator on page 83

Configuring Classify-Traffic Conditions for Dynamic Service Activator

Table 5 on page 82 lists the classify-traffic condition fields that you can configure for PCMM classifiers that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the classifier.

Table 5: Parameter Names for Classify-Traffic Conditions

Field	Parameter Name
Protocol	protocol
Source IP Address	sourceIpAddress
Source IP Mask	sourceIpMask
Port	sourcePortStart and sourcePortEnd destinationPortStart and destinationPortEnd
Destination IP Address	destinationIpAddress
Destination IP Mask	destinationIpMask

Configuring FlowSpec Actions for Dynamic Service Activator

You can use a FlowSpec action to specify the traffic profile in CommitResource messages. Table 6 on page 83 lists the fields that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the FlowSpec action.

Table 6: Parameter Names for FlowSpec Actions

Field	Parameter Name
Service Number	serviceNumber
Token Bucket Rate	bucketRate
Token Bucket Size	bucketDepth
Peak Data Rate	peakRate
Minimum Policed Unit	minPolicedUnit
Maximum Packet Size	maxDatagramSize
Rate	reservedRate
Slack Term	slackTerm

Configuring Service Class Name Actions for Dynamic Service Activator

You can use a service class name action to specify the traffic profile in CommitResource messages. Table 7 on page 83 lists the field that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for a service class name action.

Table 7: Parameter Names for Service Class Actions

Field	Parameter Name
Service Class	TrafficClass

Configuring DOCSIS Actions for Dynamic Service Activator

You can use a DOCSIS action to specify the priority and bandwidth in the traffic profile in CommitResource messages. Table 8 on page 84 lists the fields that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the DOCSIS action.

Table 8: Parameter Names for DOCSIS Actions

Field	Parameter Name
Traffic Priority	priority
Maximum Sustained Traffic Rate	bandwidth
Maximum Traffic Burst	bandwidth
Minimum Reserved Traffic Rate	bandwidth

Related Documentation

- Web Service Interface for PCMM Overview on page 79
- Configuring Services That Are Available for PCMM Clients (SRC CLI) on page 84
- *Configuring Classify-Traffic Conditions (SRC CLI)*
- *Configuring FlowSpec Actions (SRC CLI)*
- *Configuring Service Class Name Actions (SRC CLI)*
- *Configuring DOCSIS Actions (SRC CLI)*

Configuring Services That Are Available for PCMM Clients (SRC CLI)

Application managers require that usernames be specified from the Web Services Security: Username Token Profile 1,0, OASIS Standard.

Use the following configuration statements to configure available services for PCMM clients:

```
shared dsa group name configuration client name

shared dsa group name configuration client name permissions {
   pcmm-service [pcmm-service...];
}
```

To configure PCMM clients and the services that are available to the clients:

1. From configuration mode, access the statement that configures the PCMM client. You must use the same name for the PCMM client that is configured on the Web application server.

   If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string “anonymous client” as the first argument to the SAE’s Java scripts interface module.

   ```
   [edit]
   user@host# edit shared dsa group name configuration client name permissions
   ```

2. Specify the services available to the PCMM client.

   ```
   [edit shared dsa group name configuration client name permissions]
   ```

user@host# **set pcmm-service** [*pcmm-service*...]

Related Documentation

- Web Service Interface for PCMM Overview on page 79
- Sample Data for Dynamic Service Activator on page 34
- Configuring PCMM Policies and Parameter Substitutions (SRC CLI) on page 82
- *Configuring User Accounts for Web Applications (SRC CLI)*
- SRC PCMM Web Service Interface Methods on page 79

PART 2

Providing Services in IMS Networks

CHAPTER 10

Providing Services in IMS Networks

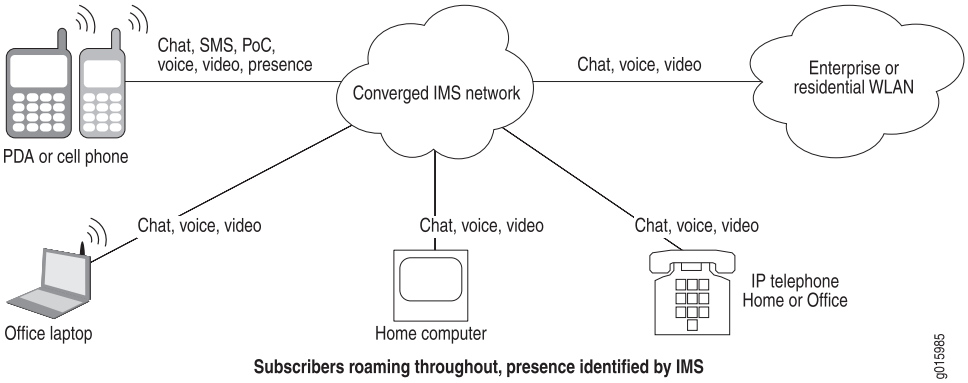
IMS Environment Overview

IP multimedia subsystem (IMS) is a flexible network architecture that allows providers to introduce rich multimedia services across both next-generation packet-switched and traditional circuit-switched networks. It uses open interfaces and functional components that can be assembled flexibly to support real-time interactive services and applications.

Third Generation Partnership Project (3GPP) developed IMS to provide a standards-based architecture for mobile carriers to migrate to their next-generation networks that will support applications that combine voice, video, and data functionality. The European Telecommunications Standards Institute (ETSI) created Telecommunications and Internet Converged Services and Protocols for Advanced Networks (TISPAN) to extend IMS support to fixed-line carriers. This extension is commonly called fixed mobile convergence (FMC). IMS/FMC allows subscribers to access any network (wireless or fixed) from any device (computer, PDA, or cell phone) and be able to move seamlessly from one network to another.

Figure 3 on page 90 shows, at a high level, a converged IMS network that manages and controls the movement of subscribers between fixed and wireless networks.

Figure 3: A Simplified IMS Converged Network (Service Focus)



By itself, IMS does not specify new services; rather, it provides a framework for network operators to build and launch their services regardless of access method. The IMS architecture simplifies network operations and allows providers to focus on service introduction and business opportunities. For example, an IMS architecture could allow fixed and mobile users to communicate using voice, video, chat, and online gaming, and to take advantage of functionality such as Push-to-Talk over Cellular (PoC; the ability to quickly arrange meetings through a walkie-talkie mechanism), instant messaging, and presence (whether and how a subscriber is available, and how the subscriber wants to be contacted).

Related Documentation

- IMS and ETSI References on page 90
- IMS Layers on page 92
- ETSI-TISPAN Architecture on page 93
- SRC Software in the ETSI-TISPAN Architecture on page 95
- SRC Software in the IMS Environment on page 95
- Configuring the IMS Software (SRC CLI) on page 101

IMS and ETSI References

For more information about IMS and TISPAN, consult the following specifications:

- ETSI ES 283 026 V0.0.7 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control; Protocol for QoS reservation information exchange between the Service Policy Decision Function (SPDF) and the Access-Resource and Admission Control Function (A-RACF) in the Resource and Protocol specification.*
- ETSI TS 183 017 V.0.0.8 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control: DIAMETER protocol for session based policy set-up information exchange between the Application Function (AF) and the Service Policy Decision Function (SPDF); Protocol specification.*

- ETSI ES 283 034 V0.0.5 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networks (TISPAN); Network Attachment Sub-System (NASS); e4 interface based on the DIAMETER protocol*.

Abbreviations

Table 9 on page 91 identifies abbreviations used in the IMS and ETSI-TISPAN environments.

Table 9: Abbreviations in the IMS and ETSI-TISPAN Environments

Abbreviation	Description
3GPP	Third-Generation Partnership Project, which developed the IMS specifications.
A-RACF	Access-resource and admission control function. Provides admission control and network policy assembly.
AVP	Attribute value pair
BGF	Border gateway function
ETSI	European Telecommunications Standards Institute
FMC	Fixed mobile convergence
IMS	IP multimedia subsystem
NGN	Next-generation network
RACS	Resource and admission control subsystem. Consists of the A-RACF and the SPDF.
RCEF	Resource control enforcement function
SPDF	Service policy decision function. The SPDF coordinates the resource reservations requests that it receives from the application function.
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networks

Related Documentation

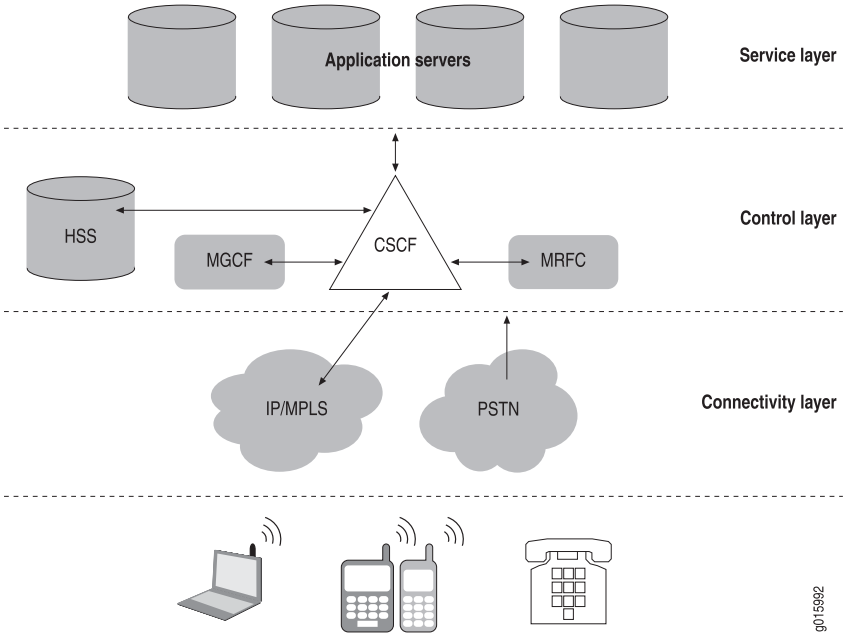
- IMS Environment Overview on page 89
- IMS Layers on page 92
- ETSI-TISPAN Architecture on page 93
- SRC Software in the ETSI-TISPAN Architecture on page 95
- Configuring the IMS Software (SRC CLI) on page 101

IMS Layers

The IMS specifications define functions to handle the signaling and subscriber traffic for multimedia applications. The functions are separated into logical layers, and many of the specified functions often reside in a single platform. Vendors have the flexibility to implement IMS functions in consolidated ways, and it is natural that platforms such as softswitches will combine many logically separate IMS call-processing functions, and that routers will take on some of the session-enforcement and gateway functionality in IMS.

The three layers are the service layer, the control layer, and the connectivity layer. Figure 4 on page 92 shows a high-level view of the IMS architecture.

Figure 4: High-Level View of the IMS Architecture



- Service layer—Hosts application and content services, including application servers and Web servers. It also includes generic service enablers that manage service elements such as user groups and presence. These service elements connect to subscribers through the control plane. The application layer supports most of the multimedia applications or application enablers, such as presence and location of the subscriber.
- Control layer—Makes the policy decisions that are enforced in the connectivity layer. This layer provides session control and management, and is responsible for setting up and taking down packet sessions. It also contains information about subscriber authentication, service authorization, and location.
- Connectivity layer—Supports the core network architecture of the General Packet Radio Service (GPRS), which consists of support nodes for data services. This layer is where routers, switches, firewalls, and optical transport reside, along with gateways that translate protocols between packet- and circuit-based traffic.

Signaling Protocol

Session Initiation Protocol (SIP) is the main signaling protocol in IMS. SIP is the proposed standard for multimedia communication between subscribers interacting with voice, video, and instant messaging. In IMS, the use of SIP facilitates interconnectivity between fixed and mobile networks.

Related Documentation

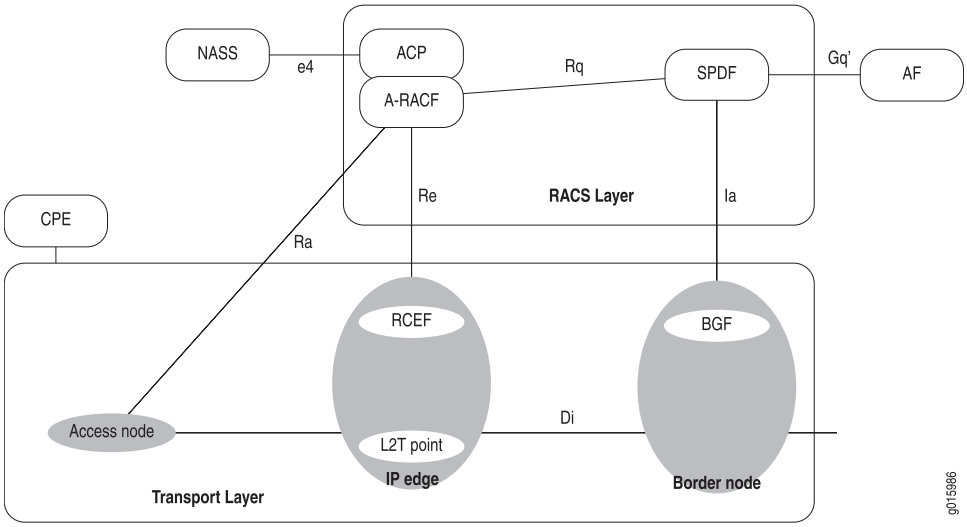
- IMS Environment Overview on page 89
- IMS and ETSI References on page 90
- SRC Software in the IMS Environment on page 95
- Configuring the IMS Software (SRC CLI) on page 101

ETSI-TISPAN Architecture

TISPAN is an extension to the IMS architecture developed by ETSI to fit the specific requirements of fixed-line providers.

Figure 5 on page 93 shows a high-level view of the TISPAN architecture.

Figure 5: High-Level View of the ETSI-TISPAN Architecture



RACS Layer

The RACS layer is the TISPAN next-generation network subsystem that is responsible for elements of policing control, including resource reservation and admission control in the access and aggregation networks. The RACS layer also includes support for NAT in the access, aggregation, and core networks required to support end-to-end application-initiated sessions.

The RACS provides policy-based transport control services to applications. These services enable applications to request and reserve transport resources from the transport networks within the scope of the RACS.

Rq Interface

The Rq interface is the interface between the SPDF and the A-RACF. The SPDF issues requests for resources in the access network through the Rq interface. These requests indicate IP QoS characteristics. The A-RACF uses the IP QoS information to perform admission control and indicates to the SPDF through the Rq interface its admission control decisions.

SPDF

The SPDF is a functional element that coordinates the resource reservation requests that it receives from the application function (the application-level controller, such as a SIP server). The SPDF performs the following functions:

- Determines whether the request information received from the application function is consistent with the policy rules defined in the SPDF.
- Authorizes the requested resources for the application function session. The SPDF uses the request information received from the application function to calculate the proper authorization (that is, to authorize certain media components).
- Provides the location of the BGF and/or the A-RACF device, in accordance with the required transport capabilities.
- Requests resources of the A-RACF.
- Requests services from the BGF.
- Hides the details of the RACS and the core transport layer from the control architecture.
- Provides resource mediation by mapping requests from application functions toward an appropriate A-RACF and/or BGF.

A-RACF

The A-RACF is a functional element that provides admission control and network policy assembly.

For admission control, the A-RACF receives requests for QoS resources from the SPDF and uses the QoS information received to perform admission control. It then indicates to the SPDF whether or not a request for resources is granted.

Access network policies are a set of rules that specify the policies that should be applied to an access line. For network policy assembly, the A-RACF:

- Ensures that requests from the SPDF match the access policies because multiple SPDFs can request resources from the A-RACF.
- Combines the requests from the SPDFs that have requested resources and ensures that the total of the requests match the capabilities of the access line.

Related Documentation

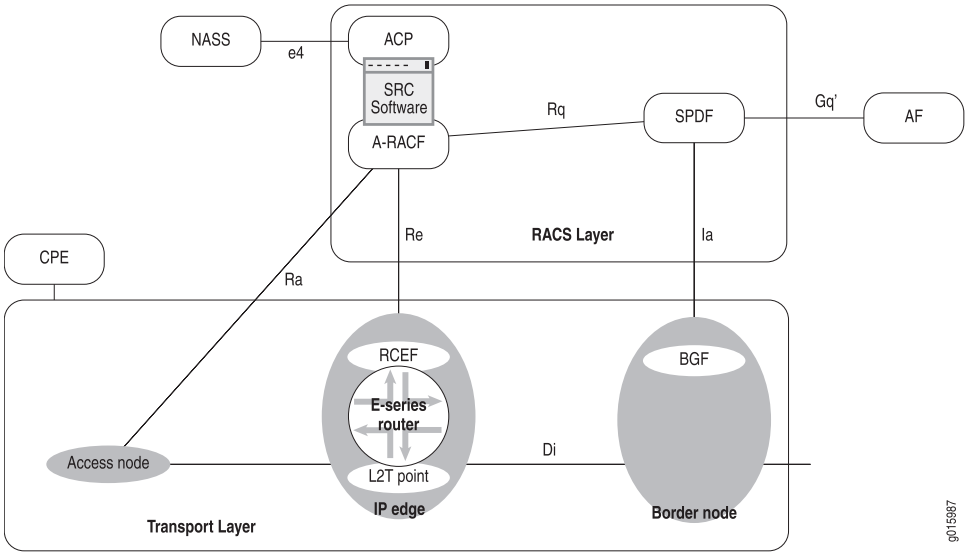
- IMS Environment Overview on page 89
- IMS and ETSI References on page 90

- SRC Software in the ETSI-TISPAN Architecture on page 95

SRC Software in the ETSI-TISPAN Architecture

Figure 6 on page 95 shows the SRC software in the ETSI-TISPAN architecture.

Figure 6: SRC Software in the ETSI-TISPAN Architecture



The SAE provides the A-RACF functionality, and the SRC software provides a northbound Rq interface from the A-RACF to the SPDF. This interface is equivalent to the Rq interface defined in the ETSI-TISPAN release 1 architecture. It is a DIAMETER protocol–based interface that allows the SRC software to integrate with services found on the application layer of IMS.

The SRC software uses its COPS and BEEP interfaces as the Re interface to Juniper Networks routers.

Related Documentation

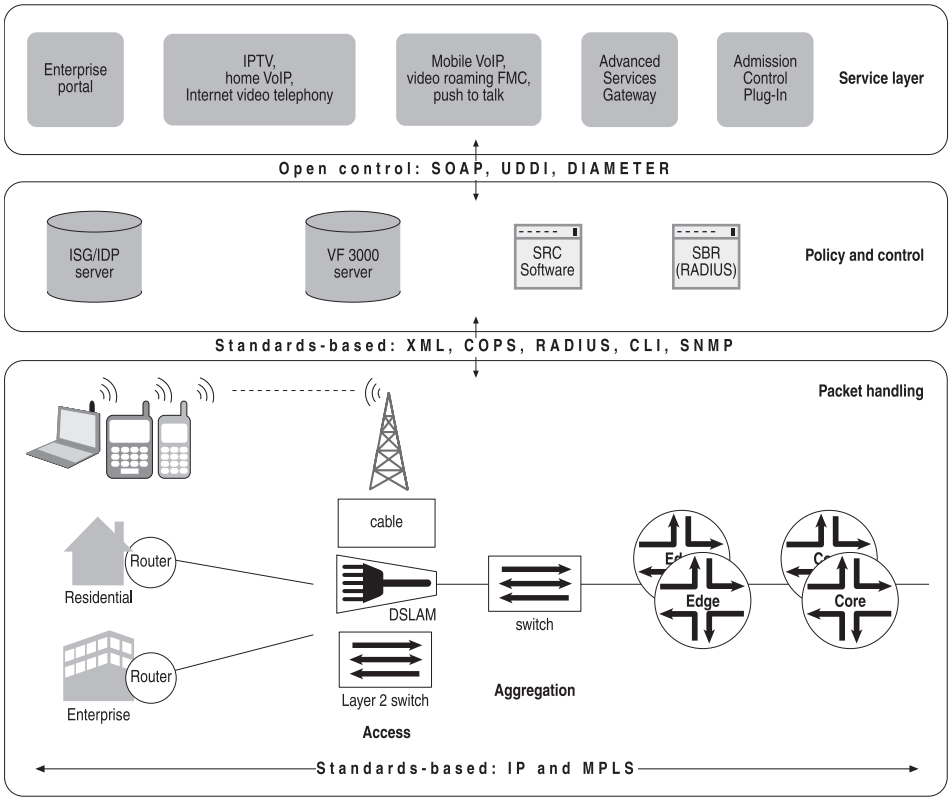
- IMS Environment Overview on page 89
- IMS and ETSI References on page 90
- ETSI-TISPAN Architecture on page 93
- Configuring the IMS Software (SRC CLI) on page 101

SRC Software in the IMS Environment

Figure 7 on page 96 shows the Juniper Networks layered IMS architecture.

The northbound Rq interface of the policy and control layer allows integration with SRC applications, such as the portals, the Application Services Gateway, and the Admission Control Plug-In (ACP).

Figure 7: Juniper Networks IMS Architecture



State Synchronization

When the SRC IMS Gateway fails over or is restarted, it needs to use the session ID that is acquired during service activation to deactivate the service. This information can be passed to the SAE if the IMS Gateway synchronizes states with the SAE. If state synchronization is enabled, the current session information can be transferred so that the IMS Gateway does not have to keep a local and persistent copy of the data. The IMS Gateway registers its interoperable object reference (IOR) with the SAE so that the SAE can communicate with the IMS Gateway.

Redundancy

If two SRC IMS Gateways synchronize states with the SAE, one IMS Gateway can provide redundancy for the other IMS Gateway. The first IMS Gateway connected to the SPDF is the primary IMS Gateway, and the other IMS Gateway is the redundant (standby) IMS Gateway. After the IMS Gateway asks the SAE to activate services for the session, the session information is passed to the SAE as activation attributes that are stored in the SAE's session store. The SAE synchronizes states with the standby IMS Gateway, so that the standby IMS Gateway can become the primary IMS Gateway if the primary IMS Gateway becomes unavailable.

Related Documentation

- IMS Environment Overview on page 89
- IMS and ETSI References on page 90

- IMS Layers on page 92
- Configuring the IMS Software (SRC CLI) on page 101

CHAPTER 11

Providing Services in IMS Networks (SRC CLI)

Configuration Statements for IMS Support

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

```
slot number ims {
   shared shared;
}
slot number ims aracf-rq {
   protocol protocol;
```

```
    port port;
    address address;
    origin-host origin-host;
    origin-realm origin-realm;
  }
  slot number ims aracf-rq peer name {
    address address;
    port port;
    origin-host origin-host;
    watchdog-timeout watchdog-timeout;
    incoming-queue-limit incoming-queue-limit;
  }
  slot number ims initial {
    static-dn static-dn;
    dynamic-dn dynamic-dn;
  }
  slot number ims initial directory-connection {
    url url;
    backup-urls [backup-urls...];
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
  }
  slot number ims initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
  }
  slot number ims logger name ...
  slot number ims logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
  }
  slot number ims logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
  }
```

For more information about the configuration statements, see the *SRC PE CLI Command Reference*.

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- IMS Environment Overview on page 89
- Example: Configuring JunosE Policies for IMS (SRC CLI) on page 120

Configuring the IMS Software (SRC CLI)

To configure the IMS software:

1. Configure initial properties, including the connection to the directory and directory monitoring properties.

   See "Configuring Initial Properties for IMS (SRC CLI)" on page 101.

   See "Configuring Directory Connection Properties for IMS (SRC CLI)" on page 102.

   See "Configuring Initial Directory Eventing Properties for IMS (SRC CLI)" on page 103.

2. Configure the local and remote Diameter peers.

   See "Configuring the Local Diameter Peer (SRC CLI)" on page 104.

   See "Configuring the Remote Diameter Peer (SRC CLI)" on page 105.

3. Configure logging destinations.

   See "Configuring Logging Destinations to Store Messages in a File (SRC CLI)" on page 106.

   See "Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)" on page 107.

4. Configure subscriber types.

   See "Configuring the Subscriber Type (SRC CLI)" on page 109.

5. Configure the NIC proxies.

   See "Configuring a NIC Proxy for IMS (SRC CLI)" on page 110.

6. Start the IMS process to provide the A-RACF Rq interface.

   See "Starting the IMS Process (SRC CLI)" on page 117.

You must restart the IMS process after you commit a configuration change. To restart IMS, see "Restarting the IMS Process (SRC CLI)" on page 117.

Related Documentation

- IMS Environment Overview on page 89
- Monitoring IMS (SRC CLI) on page 118
- Monitoring IMS (C-Web Interface) on page 119
- Configuration Statements for IMS Support on page 99
- SRC Software in the IMS Environment on page 95

Configuring Initial Properties for IMS (SRC CLI)

Use the following configuration statements to configure initial properties for IMS:

```
slot number ims initial {
  static-dn static-dn;
```

```
    dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

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

   ```
   user@host# edit slot 0 ims initial
   ```

2. Specify the properties for IMS.

   ```
   [edit slot 0 ims initial]
   user@host# set ?
   ```

   For more information about configuring local properties for SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 ims initial]
   user@host# show
   ```

Related Documentation

- IMS Environment Overview on page 89
- Configuring the IMS Software (SRC CLI) on page 101
- Configuring Initial Directory Eventing Properties for IMS (SRC CLI) on page 103
- Configuration Statements for IMS Support on page 99

Configuring Directory Connection Properties for IMS (SRC CLI)

Use the following configuration statements to configure directory connection properties for IMS:

```
slot number ims initial directory-connection {
    url url;
    backup-urls [backup-urls...];
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}
```

To configure directory connection properties:

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

   ```
   user@host# edit slot 0 ims initial directory-connection
   ```

2. Specify the properties for IMS.

```
[edit slot 0 ims initial directory-connection]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 ims initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=Operators,<base>;
credentials ********;
```

Related Documentation

- IMS Environment Overview on page 89
- Configuring the IMS Software (SRC CLI) on page 101
- Configuring Initial Properties for IMS (SRC CLI) on page 101
- Configuring Initial Directory Eventing Properties for IMS (SRC CLI) on page 103
- Configuration Statements for IMS Support on page 99

Configuring Initial Directory Eventing Properties for IMS (SRC CLI)

Use the following configuration statements to configure directory eventing properties for IMS:

```
slot number ims initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory eventing properties:

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

```
user@host# edit slot 0 ims initial eventing
```

2. Specify the initial directory eventing properties for IMS.

```
[edit slot 0 ims initial directory-eventing]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 ims initial directory-eventing]
user@host# show
eventing;
polling-interval 30;
```

Related Documentation

- IMS Environment Overview on page 89
- Configuring the IMS Software (SRC CLI) on page 101
- Configuring Initial Properties for IMS (SRC CLI) on page 101
- Configuring Directory Connection Properties for IMS (SRC CLI) on page 102
- Configuration Statements for IMS Support on page 99

Configuring the Local Diameter Peer (SRC CLI)

Use the following configuration statements to configure the local Diameter peer:

```
slot number ims aracf-rq {
    protocol protocol;
    port port;
    address address;
    origin-host origin-host;
    origin-realm origin-realm;
}
```

To configure the local Diameter peer:

1. From configuration mode, access the statement that configures the Diameter peer.

   ```
   user@host# edit slot 0 ims aracf-rq
   ```

2. (Optional) Specify the protocol used for the transport layer.

   ```
   [edit slot 0 ims aracf-rq]
   user@host# set protocol protocol
   ```

3. (Optional) Specify the port used for incoming connections.

   ```
   [edit slot 0 ims aracf-rq]
   user@host# set port port
   ```

4. (Optional) Specify the IP address of the local peer.

   ```
   [edit slot 0 ims aracf-rq]
   user@host# set address address
   ```

5. (Optional) Specify the Diameter identifier for the local endpoint that is the originator of the Diameter message.

   ```
   [edit slot 0 ims aracf-rq]
   user@host# set origin-host origin-host
   ```

6. (Optional) Specify the Diameter identifier for the realm of the local endpoint that is the originator of the Diameter message.

   ```
   [edit slot 0 ims aracf-rq]
   user@host# set origin-realm origin-realm
   ```

7. (Optional) Verify your configuration.

```
[edit slot 0 ims aracf-rq]
user@host# show
protocol tcp;
port 3868;
address 127.0.0.1;
origin-host testserver;
origin-realm testrealm;
peer 1 {
  address 127.0.0.1;
  origin-host testclient;
}
```

Related Documentation

- IMS Environment Overview on page 89
- Configuring the IMS Software (SRC CLI) on page 101
- Configuring the Remote Diameter Peer (SRC CLI) on page 105
- Monitoring IMS (SRC CLI) on page 118

Configuring the Remote Diameter Peer (SRC CLI)

Use the following configuration statements to configure the remote Diameter peer:

```
slot number ims aracf-rq peer name {
  address address;
  port port;
  origin-host origin-host;
  watchdog-timeout watchdog-timeout;
  incoming-queue-limit incoming-queue-limit;
}
```

To configure the remote Diameter peer:

1. From configuration mode, access the statement that configures the Diameter peer. In this sample procedure, the remote SPDF peer called primary-spdf is configured.

   ```
   user@host# edit slot 0 ims aracf-rq peer primary-spdf
   ```

2. (Optional) Specify the IP address of the remote peer.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# set address address
   ```

3. (Optional) Specify the port of the remote peer.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# set port port
   ```

4. (Optional) Specify the Diameter identifier for the remote endpoint that is the originator of the Diameter message.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# set origin-host origin-host
   ```

5. (Optional) Specify the watchdog timeout of the connection to the remote peer.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# set watchdog-timeout watchdog-timeout
   ```

6. (Optional) Specify the size of the incoming message queue before the system rejects messages.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# set incoming-queue-limit incoming-queue-limit
   ```

7. (Optional) Verify your configuration.

   ```
   [edit slot 0 ims aracf-rq peer primary-spdf]
   user@host# show
   address 127.0.0.1;
   origin-host testclient;
   ```

Related Documentation

- IMS Environment Overview on page 89
- Configuring the IMS Software (SRC CLI) on page 101
- Configuring the Local Diameter Peer (SRC CLI) on page 104
- Monitoring IMS (SRC CLI) on page 118

Configuring Logging Destinations to Store Messages in a File (SRC CLI)

Use the following configuration statements to configure file logging for IMS:

```
slot number ims logger name ...
slot number ims logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}
```

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

1. From configuration mode, access the statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log1 is configured.

   ```
   user@host# edit slot 0 ims logger log1 file
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 ims logger log1 file]
   user@host# set ?
   ```

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

3. (Optional) Verify your configuration.

```
[edit slot 0 ims logger log1 file]
user@host# show
filter /info-;
filename var/log/ims-a-racf-rq-info.log;
rollover-filename var/log/ims-a-racf-rq-info.alt;
maximum-file-size 2000000;
```

Related Documentation

- Configuring Initial Properties for IMS (SRC CLI) on page 101
- Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI) on page 107

Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)

Use the following configuration statements to configure system logging for IMS:

```
slot number ims logger name ...
slot number ims logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
```

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

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

   ```
   user@host# edit slot 0 ims logger log2 syslog
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 ims logger log2 syslog]
   user@host# set ?
   ```

   For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 ims logger log2 syslog]
   user@host# show
   ```

Related Documentation

- Configuring Initial Properties for IMS (SRC CLI) on page 101
- Configuring Logging Destinations to Store Messages in a File (SRC CLI) on page 106

Creating Grouped Configurations for IMS (SRC CLI)

Configuration groups allow you to share the IMS configuration with different IMS instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped IMS configuration that is shared with some IMS instances. For example, if you create two different IMS groups called config1 and config2 within the shared IMS configuration, you could select the IMS configuration that should be associated with a particular IMS instance.

Use the **shared** option of the **slot** ***number*** **ims** statement to select the group for an IMS instance as part of the local configuration. Use the **shared ims group** ***name*** **configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an IMS instance. For example, to select a group called config1 in the root group:

   ```
   [edit]
   user@host# set slot 0 ims shared /config1
   ```

2. Commit the configuration.

   ```
   [edit]
   user@host# commit
   commit complete.
   ```

3. From configuration mode, configure a group. For example, to configure a group called config1, specify the group as part of the IMS configuration.

   ```
   [edit]
   user@host# edit shared ims group config1 ?
   Possible completions:
   <[Enter]> Execute this command
   > configuration
   | Pipe through a command
   ```

   For more information, see "Configuring the IMS Software (SRC CLI)" on page 101.

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- Configuring the Subscriber Type (SRC CLI) on page 109
- Configuring a NIC Proxy for IMS (SRC CLI) on page 110
- Configuring IMS for Failover (SRC CLI) on page 115
- IMS Environment Overview on page 89

Configuring the Subscriber Type (SRC CLI)

Use the following configuration statements to configure the subscriber type:

```
shared ims configuration subscriber-types (ip | login-name) {
  nic-proxy nic-proxy;
  subscriber-id-type (address | login-name | primary-user-name);
}
```

To configure the subscriber type:

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

   user@host# **edit shared ims configuration subscriber-types (ip | login-name)**

   where:

   - **ip**—Subscriber type of IP address
   - **login-name**—Subscriber type of login ID

   For example, the subscriber type called ip is configured in this sample procedure.

   user@host# **edit shared ims configuration subscriber-types ip**

2. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy. In this sample procedure, the namespace for the NIC proxy called ip is configured.

   [edit shared ims configuration subscriber-types ip]
   user@host# **set nic-proxy ip**

3. (Optional) Specify the type of information used to identify the subscriber.

   [edit shared ims configuration subscriber-types ip]
   user@host# **set subscriber-id-type** (address | login-name | primary-user-name)

   where:

   - **address**—Subscriber's IP address
   - **login-name**—Subscriber's login name
   - **primary-user-name**—Primary username

   In this sample procedure, the subscriber ID type is specified as the subscriber IP address.

   [edit shared ims configuration subscriber-types ip]
   user@host# **set subscriber-id-type address**

4. (Optional) Verify your configuration.

   ```
   [edit shared configuration subscriber-types ip]
   user@host# show
   ```

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- Configuring Initial Properties for IMS (SRC CLI) on page 101
- Configuring a NIC Proxy for IMS (SRC CLI) on page 110

Configuring a NIC Proxy for IMS (SRC CLI)

Tasks to configure the NIC proxy are:

- Configuring Resolution Information for a NIC Proxy on page 110
- Changing the Configuration for the NIC Proxy Cache on page 111
- Configuring a NIC Proxy for NIC Replication on page 112
- Configuring NIC Test Data on page 114

Configuring Resolution Information for a NIC Proxy

You create a NIC proxy for each subscriber type to be configured. Subscriber types that have different subscriber ID types can use the same NIC proxy.

Use the following configuration statements to configure the NIC proxy:

```
shared ims configuration nic-proxy-configuration name
shared ims configuration nic-proxy-configuration name resolution {
   resolver-name resolver-name;
   key-type key-type;
   value-type value-type;
   expect-multiple-values;
   constraints constraints;
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

   ```
   user@host# edit shared ims configuration nic-proxy-configuration ip resolution
   ```

2. Specify the NIC resolver that this NIC proxy uses. This resolver must be the same as one that is configured on the NIC host.

   ```
   [edit shared ims configuration nic-proxy-configuration ip resolution]
   user@host# set resolver-name resolver-name
   ```

3. Specify the NIC data type that the key provides for the NIC resolution.

   ```
   [edit shared ims configuration nic-proxy-configuration ip resolution]
   user@host# set key-type key-type
   ```

   To qualify data types, enter a qualifier within parentheses after the data type; for example, to specify username as a qualifier for the key LoginName:

   ```
   [edit shared ims configuration nic-proxy-configuration ip resolution]
   ```

```
user@host# set key-type LoginName (username)
```

4. Specify the type of value to be returned in the resolution for the application that uses the NIC proxy.

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# set value-type value-type
```

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# set expect-multiple-values
```

6. (Optional. Available at the Advanced editing level.) If the application provides a constraint in the resolution request, specify the data type for the constraint. The constraint represents a condition that must or may be satisfied before the next stage of the resolution process can proceed.

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# set constraints constraints
```

7. (Optional) Verify your configuration.

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# show
resolver-name /realms/ip/A1;
key-type Ip;
value-type SaeId;
```

Changing the Configuration for the NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
shared ims configuration nic-proxy-configuration name cache {
  cache-size cache-size;
  cache-cleanup-interval cache-cleanup-interval;
  cache-entry-age cache-entry-age;
  }
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims configuration nic-proxy-configuration ip cache
```

2. (Optional) Specify the maximum number of keys for which the NIC proxy retains data.

```
[edit shared ims configuration nic-proxy-configuration ip cache]
```

```
user@host# set cache-size cache-size
```

If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.

3. Specify the time interval at which the NIC proxy removes expired entries from its cache.

```
[edit shared ims configuration nic-proxy-configuration ip cache]
user@host# set cache-cleanup-interval cache-cleanup-interval
```

4. (Optional) Specify how long an entry remains in the cache.

```
[edit shared ims configuration nic-proxy-configuration ip cache]
user@host# set cache-entry-age cache-entry-age
```

5. (Optional) Verify your configuration.

```
[edit shared configuration nic-proxy-configuration ip cache]
user@host# show
cache-size 10000;
cache-cleanup-interval 15;
```

Configuring a NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Normal editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```
shared ims configuration nic-proxy-configuration name nic-host-selection {
  groups groups ;
  selection-criteria (roundRobin | randomPick | priorityList);
}
shared ims configuration nic-proxy-configuration name nic-host-selection blacklisting {
  try-next-system-on-error;
  number-of-retries-before-blacklisting number-of-retries-before-blacklisting ;
  blacklist-retry-interval blacklist-retry-interval ;
}
```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims configuration nic-proxy-configuration ip nic-host-selection
```

2. (Optional) Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests.

```
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups groups
```

3. (Optional) If you configure more than one group, specify the selection criteria that the NIC proxy uses to determine which NIC host to contact.

   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection]
   user@host# **set selection-criteria** (roundRobin | randomPick | priorityList)

   where:

   - roundRobin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
   - randomPick—NIC proxy selects NIC hosts randomly from the list.
   - priorityList—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.

     Priorities are defined by the order in which you specify the groups. You can change the order of NIC hosts in the list by using the **insert** command.

4. (Optional) Verify your configuration.

   ```
   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection]

   user@host# show
   groups ;
   selection-criteria round-;
   ```

5. Access the statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection]
   user@host# **edit blacklisting**
   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection blacklisting]

6. (Optional) Specify whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.

   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection blacklisting]
   user@host# **set try-next-system-on-error**

7. (Optional) Change the number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time. The default is 3.

   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection blacklisting]
   user@host# **set number-of-retries-before-blacklisting**
     number-of-retries-before-blacklisting

8. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

   [edit shared ims configuration nic-proxy-configuration ip nic-host-selection blacklisting]
   user@host# **set blacklist-retry-interval** blacklist-retry-interval

9. (Optional) Verify your configuration.

```
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# show
try-next-system-on-error;
number-of-retries-before-blacklisting 3;
blacklist-retry-interval 15;
```

Configuring NIC Test Data

To test a resolution without the NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the NIC proxy configuration. When the SRC component configured to use a NIC proxy stub passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value. When you use a NIC proxy stub, no NIC infrastructure is required.

Use the following configuration statements to configure a NIC proxy stub from the **[edit]** hierarchy level.

```
shared ims configuration nic-proxy-configuration name test-nic-bindings {
  use-test-bindings;
}
shared ims configuration nic-proxy-configuration name test-nic-bindings key-values name
  {
  value;
}
```

To use the NIC proxy stub for IMS:

1. In configuration mode, navigate to the NIC proxy configuration and specify the data type of the key you want to map to a value. In this sample procedure, the key ip is specified for the NIC proxy called ip.

   ```
   [edit shared ims configuration nic-proxy-configuration ip]
   user@host# set resolution key-type ip
   ```

2. Enable a NIC proxy stub for a resolution.

   ```
   [edit shared ims configuration nic-proxy-configuration ip]
   user@host# set test-nic-bindings use-test-bindings
   ```

3. Specify the values of the keys for testing. These statements are available at the Advanced CLI editing level.

   ```
   [edit shared ims configuration nic-proxy-configuration ip]
   user@host# set test-nic-bindings key-values name value
   ```

   where:

   - *name*—Indicates the NIC data value for the proxy.
   - *value*—Specifies a value for the NIC data type.

   For example, to set up a login name to IP mapping for login name jane@virneo.com to the IP address 192.0.2.30:

   ```
   [edit shared ims configuration nic-proxy-configuration ip]
   user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
   ```

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- *Configuring NIC Test Data (SRC CLI)*
- *Configuring a NIC Proxy for NIC Replication (SRC CLI)*
- Configuration Statements for IMS Support on page 99

Configuring IMS for Failover (SRC CLI)

Use the following configuration statements to configure IMS for failover:

```
shared ims configuration redundancy {
  state-synchronization;
  state-sync-bulk-size state-sync-bulk-size;
  state-synchronization-timeout state-synchronization-timeout ;
}
```

To configure state synchronization with the SAE:

1. From configuration mode, access the statement that configures redundancy for IMS.

   ```
   user@host# edit shared ims configuration redundancy
   ```

2. (Optional) Enable state synchronization from the SAE.

   ```
   [edit shared ims configuration redundancy]
   user@host# set state-synchronization
   ```

3. (Optional) Specify the number of events the SAE sends to IMS at one time during state synchronization.

   ```
   [edit shared ims configuration redundancy]
   user@host# set state-sync-bulk-size state-sync-bulk-size
   ```

4. (Optional) Specify the time to wait for the first full synchronization request from the SAE after starting or restarting IMS.

   ```
   [edit shared ims configuration redundancy]
   user@host# set state-synchronization-timeout state-synchronization-timeout
   ```

Related Documentation

- Configuring the SAE for IMS on page 115
- Configuring the IMS Software (SRC CLI) on page 101
- Configuration Statements for IMS Support on page 99

Configuring the SAE for IMS

You must configure the SAE to recognize IMS by adding information about IMS to the SAE properties. Tasks for configuring the SAE for IMS are:

- Configuring IMS as an External Plug-In on page 116
- Configuring Event Publishers on page 116

Configuring IMS as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the statement that configures the external plug-ins.

   user@host# edit shared sae configuration plug-ins name *name* external

2. Configure the object reference of the external plug-in that is exported to the SAE.

   [edit shared sae configuration plug-ins name *name* external]
   user@host# set corba-object-reference *corba-object-reference*

   where ***corba-object-reference*** is one of the following references:

   - Path to the interoperable object reference (IOR) file in the format file:///opt/UMC/ims/var/run/ims.ior
   - The corbaloc URL in the format corbaloc::*host*:9801/ASGIMS where *host* is the IP address of the C Series Controller or 127.0.0.1
   - Common Object Services (COS) in the format corbaname::*host*:2809#ASGIMS/statesync/*hostname* or corbaname::*host*:2809/NameService#ASGIMS/statesync/*hostname* where
     - *host* is the IP address of the C Series Controller
     - *hostname* is the hostname of the C Series Controller

3. Specify the plug-in attributes.

   [edit shared sae configuration plug-ins name *name* external]
   user@host# set attributes ?

   Attributes for IMS are service-name, service-session-name, router-name, login-name, terminate-cause, property.

   For more information about configuring plug-in attributes, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

Configuring Event Publishers

You must configure the SAE to publish the global service tracking events to the IMS. Any other events are ignored.

For information about configuring event publishers, see *Special Types of Event Publishers*.

Related Documentation

- *Configuring the SAE for External Plug-Ins (SRC CLI)*
- *Special Types of Event Publishers*

Managing IMS (SRC CLI)

After you have configured IMS, you can perform these tasks:

- Starting the IMS Process (SRC CLI) on page 117
- Restarting the IMS Process (SRC CLI) on page 117
- Stopping the IMS Process (SRC CLI) on page 117
- Displaying IMS Status (SRC CLI) on page 118

Starting the IMS Process (SRC CLI)

To start the IMS process:

```
user@host> enable component ims
```

The system responds with a start message. If IMS is already running, the system responds with a warning message.

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- Restarting the IMS Process (SRC CLI) on page 117
- Stopping the IMS Process (SRC CLI) on page 117
- Displaying IMS Status (SRC CLI) on page 118
- IMS Environment Overview on page 89

Restarting the IMS Process (SRC CLI)

You must restart the IMS process after you commit a configuration change.

To restart IMS:

```
user@host> restart component ims
```

The system responds with a start message. If IMS is already running, the system responds with a shutdown message and then a start message.

Related Documentation

- Starting the IMS Process (SRC CLI) on page 117
- Stopping the IMS Process (SRC CLI) on page 117
- Displaying IMS Status (SRC CLI) on page 118
- IMS Environment Overview on page 89

Stopping the IMS Process (SRC CLI)

To stop the IMS process:

```
user@host> disable component ims
```

The system responds with a shutdown message. If IMS is not running when you issue the command, the system responds with the command prompt.

Related Documentation

- Starting the IMS Process (SRC CLI) on page 117
- Restarting the IMS Process (SRC CLI) on page 117
- Displaying IMS Status (SRC CLI) on page 118
- IMS Environment Overview on page 89

Displaying IMS Status (SRC CLI)

Purpose Display IMS status.

Action

```
user@host> show component
```

The system responds with a status message.

Related Documentation

- Configuring the IMS Software (SRC CLI) on page 101
- Stopping the IMS Process (SRC CLI) on page 117
- Monitoring IMS (SRC CLI) on page 118
- Monitoring IMS (C-Web Interface) on page 119
- IMS Environment Overview on page 89

Monitoring IMS (SRC CLI)

Monitoring tasks are:

- Viewing Server Process Information on page 118
- Viewing Statistics for the Rq Interface on page 118
- Viewing Information About Peers on page 119

Viewing Server Process Information

Purpose View information about the IMS server process.

Action

```
user@host> show ims statistics aracf rq process
Rq Server Process
Rq server up time (seconds) 692942
Rq server up since          2007-03-13T15:30:48EDT
Rq server threads           93
Heap used (bytes)           16383752 (8%)
Heap limit (bytes)          200000000
```

Viewing Statistics for the Rq Interface

Purpose Monitor the current state of the A-RACF Rq interface.

Action

```
user@host> show ims statistics aracf rq
ims aracf rq Statistics
 Rq Server Process
  Rq server up time (seconds) 692920
  Rq server up since          2007-03-13T15:30:48EDT
  Rq server threads           93
  Heap used (bytes)           16332120 (8%)
  Heap limit (bytes)          200000000
```

Viewing Information About Peers

Purpose View information about the peers.

Action To view information about all configured peers:

```
user@host> show ims aracf-rq peers
```

To view information about a specific peer:

```
user@host> show ims aracf-rq peers peer-name peer-name
```

To view the name and status of configured peers:

```
user@host> show ims aracf-rq peers brief
```

```
user@host> show ims aracf-rq peers peer-name peer-name brief
```

Related Documentation

- Monitoring IMS (C-Web Interface) on page 119
- Configuring the IMS Software (SRC CLI) on page 101
- Displaying IMS Status (SRC CLI) on page 118

Monitoring IMS (C-Web Interface)

You can monitor statistics for the server process and the A-RACF Rq interface with the C-Web interface by:

- Viewing Statistics for the Server Process on page 119
- Viewing Statistics for the A-RACF Rq Interface on page 120

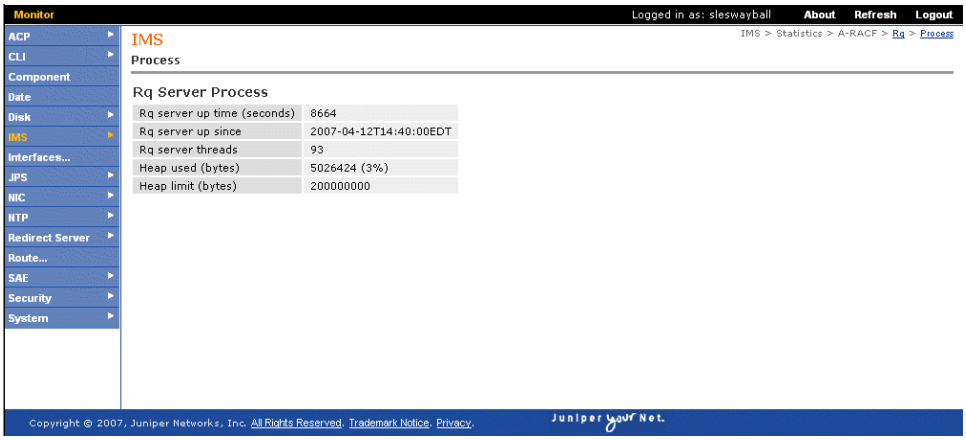
Viewing Statistics for the Server Process

Purpose View statistics for the server process.

Action Click **Monitor>IMS>Statistics>A-RACF>Rq>Process**.

The Process pane displays statistics for the server process.

Figure 8: C-Web Interface for Monitoring Server Process Statistics



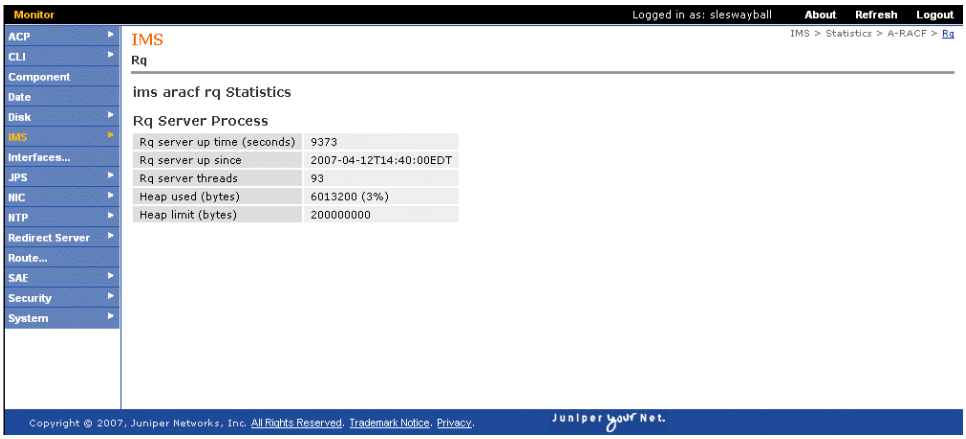
Viewing Statistics for the A-RACF Rq Interface

Purpose View statistics for the A-RACF Rq interface.

Action Click **Monitor>IMS>Statistics>A-RACF>Rq**.

The Rq pane displays statistics for the A-RACF Rq interface.

Figure 9: C-Web Interface for Monitoring A-RACF Rq Interface Statistics



Related Documentation

- Monitoring IMS (SRC CLI) on page 118
- Configuring the IMS Software (SRC CLI) on page 101
- Displaying IMS Status (SRC CLI) on page 118

Example: Configuring JunosE Policies for IMS (SRC CLI)

For IMS environments, you can configure JunosE policies. When you configure classify-traffic conditions, you can set up the software so that the SAE expands into multiple classifiers before it installs the policy on the router. If you enter a 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.

Enabling Expansion of JunosE Classify-Traffic Conditions

To enable the expansion of JunosE classify-traffic conditions:

1. From configuration mode, access the 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

For more information about expanded classifiers, see *Policy Information Model*.

Related Documentation

- *Configuring Classify-Traffic Conditions (SRC CLI)*
- *Policy Management Overview*

CHAPTER 12

Testing IMS Service Sessions (SRC CLI)

Testing Service Sessions for IMS

You can test service sessions by sending requests for:

- Activation
- Modification
- Deactivation

You can configure the settings for your test environment to easily test service sessions.

Related Documentation

- Configuring the Test Environment for IMS Services (SRC CLI) on page 123
- Testing Service Sessions (SRC CLI) on page 127

Configuring the Test Environment for IMS Services (SRC CLI)

Configuring the settings for your test environment is optional. You can choose to configure the test settings and specify changes to the test settings.

Configuring Settings for AAR Messages (SRC CLI)

Use the following command to configure the AA-Request (AAR) test message:

```
slot number ims aracf-rq test templates aar name {
  origin-host origin-host;
  origin-realm origin-realm;
  af-charging-identifier af-charging-identifier;
```

```
    authorization-lifetime authorization-lifetime;
    user-name user-name;
    specific-action (indication-of-bearer-release | indication-of-subscriber-detachment);
}
```

To configure the AAR message for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings.

   ```
   user@host# edit slot number ims aracf-rq test templates aar name
   ```

2. Specify the Diameter identifier for the endpoint that is the originator of the Diameter message.

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set origin-host origin-host
   ```

3. Specify the Diameter identifier for the realm of the endpoint that is the originator of the Diameter message.

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set origin-realm origin-realm
   ```

4. (Optional) Specify the charging identifier for the application function (AF).

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set af-charging-identifier af-charging-identifier
   ```

5. (Optional) Specify the timeout for the authorization.

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set authorization-lifetime authorization-lifetime
   ```

6. (Optional) Specify the username.

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set user-name user-name
   ```

7. (Optional) Specify the events for which notification is requested. If you do not configure this test setting, you must specify a value when testing service activations.

   ```
   [edit slot number ims aracf-rq test templates aar name]
   user@host# set specific-action (indication-of-bearer-release |
     indication-of-subscriber-detachment)
   ```

   where

   - **indication-of-bearer-release**—Provides notification of a bearer's removal
   - **indication-of-subscriber-detachment**—Provides notification of the subscriber detachment

Configuring the Globally Unique Address (SRC CLI)

Use the following command to configure the globally unique address for the AAR test message:

```
slot number ims aracf-rq test templates aar name globally-unique-address {
  framed-ip-address framed-ip-address;
}
```

To configure the globally unique address for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings.

   user@host# **edit slot** *number* **ims aracf-rq test templates aar** *name* **globally-unique-address**

2. (Optional) Specify the IPv4 address or the fully qualified domain name for the endpoint that is the originator of the Diameter message. If you do not configure this test setting, you must specify a value when testing service activations.

   [edit slot *number* ims aracf-rq test templates aar *name* globally-unique-address]
   user@host# **set framed-ip-address** *framed-ip-address*

Configuring Service Information for Media Types (SRC CLI)

Use the following command to configure the service information that is used to determine QoS requirements for the media type:

```
slot number ims aracf-rq test templates aar name media-component-description
  media-component-number {
  af-application-identifier af-application-identifier;
  media-type (audio | video | data | application | control | text | message | other);
  max-requested-download-bandwidth max-requested-download-bandwidth;
  max-requested-upload-bandwidth max-requested-upload-bandwidth;
  flow-status (enabled | removed);
}
```

To configure the media component for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings. Specify the appropriate media component number.

   user@host# **edit slot** *number* **ims aracf-rq test templates aar** *name* media-component-description *media-component-number*

2. Specify the service name.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number*]
   user@host# **set af-application-identifier** *af-application-identifier*

3. (Optional) Specify the media type.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number*]
   user@host# **set media-type (audio | video | data | application | control | text | message | other)**

4. (Optional) Specify the maximum download bandwidth requested.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number*]
   user@host# **set max-requested-download-bandwidth** *max-requested-download-bandwidth*

5. (Optional) Specify the maximum upload bandwidth requested.

[edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number*]
user@host# **set max-requested-upload-bandwidth** *max-requested-upload-bandwidth*

6. (Optional) Specify the action taken for the AAR.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number*]
   user@host# **set flow-status (enabled | removed)**

   where

   - **enabled**—Commits resource reservation in both directions
   - **removed**—Releases all resources associated with the corresponding resource reservation

Configuring IP Flows for Media Types (SRC CLI)

Use the following command to configure the QoS and filters for the IP flows:

```
slot number ims aracf-rq test templates aar name media-component-description
  media-component-number media-sub-component flow-number {
  flow-description [flow-description...];
  max-requested-download-bandwidth max-requested-download-bandwidth;
  max-requested-upload-bandwidth max-requested-upload-bandwidth;
}
```

To configure the media subcomponent for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings. Specify the appropriate flow number. These configuration settings override the media type settings.

   user@host# **edit slot** *number* **ims aracf-rq test templates aar** *name* media-component-description *media-component-number* media-sub-component *flow-number*

2. Define the packet filter for the flow. The flow description AVP contains the classifier (or filter) information.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number* media-sub-component *flow-number*]
   user@host# **set flow-description** [*flow-description*...]

   The syntax of this AVP has the following restrictions:

   - Only permit action should be used as action.
   - No options should be used.

   A subcomponent may include up to two flow descriptions (uplink and downlink), including:

   - Direction (in—uplink, or out—downlink)
   - Source IP address
   - Destination IP address

- Source port
- Destination port
- Protocol

3. (Optional) Specify the maximum download bandwidth requested.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number* media-sub-component *flow-number*]
   user@host# **set max-requested-download-bandwidth** *max-requested-download-bandwidth*

4. (Optional) Specify the maximum upload bandwidth requested.

   [edit slot *number* ims aracf-rq test templates aar *name* media-component-description *media-component-number* media-sub-component *flow-number*]
   user@host# **set max-requested-upload-bandwidth** *max-requested-upload-bandwidth*

Related Documentation

- Testing Service Sessions for IMS on page 123
- Testing Service Sessions (SRC CLI) on page 127

Testing Service Sessions (SRC CLI)

Tasks to test service sessions are:

- Testing Session Activations (SRC CLI) on page 127
- Testing Session Modifications (SRC CLI) on page 128
- Testing Session Deactivations (SRC CLI) on page 128

Testing Session Activations (SRC CLI)

Use the following command to test the activation of service sessions:

test ims aracf-rq aar session-start aar-name ***aar-name*** <framed-ip-address ***framed-ip-address***> <user-name ***user-name***> <origin-host ***origin-host***> <origin-realm ***origin-realm***>

To test service session activations:

1. Issue the **test ims aracf-rq aar session-start** command.
2. To specify the name of the AAR message settings, use the **aar-name** option.
3. (Optional) To specify the subscriber's IP address, use the **framed-ip-address** option. If you specify a value, it will overwrite the configured test setting. If you did not configure this test setting, you must specify a value.
4. (Optional) To specify the subscriber name, use the **user-name** option. If you specify a value, it will overwrite the configured test setting. If you did not configure this test setting, you must specify a value.

5. (Optional) To specify the origin host for the simulator that generates the message, use the **origin-host** option. If you specify a value, it will overwrite the configured test setting.
6. (Optional) To specify the origin realm for the simulator that generates the message, use the **origin-realm** option. If you specify a value, it will overwrite the configured test setting.

Testing Session Modifications (SRC CLI)

Use the following command to test service session modifications:

```
test ims aracf-rq aar session-modify session-id session-id aar-name aar-name
```

To test service session modifications:

1. Issue the **test ims aracf-rq aar session-modify** command.
2. To specify the session ID used to uniquely identify a user session, use the **session-id** option.
3. To specify the name of the AAR message settings, use the **aar-name** option.

Testing Session Deactivations (SRC CLI)

Use the following command to test the deactivation of service sessions:

```
test ims aracf-rq str session-id session-id
```

To deactivate a particular session, use the **session-id** option.

Related Documentation

- Testing Service Sessions for IMS on page 123
- Configuring the Test Environment for IMS Services (SRC CLI) on page 123

PART 3

Providing SRC 3GPP Gateway Services

CHAPTER 13

SRC 3GPP Gateway Overview

SRC 3GPP Gateway Overview

The SRC Third-Generation Partnership Project (3GPP) gateway is a Diameter-based component in the SRC software, which provides integration with 3GPP Policy and Charging Control environments, to provide fixed-mobile convergence (FMC).

The SRC 3GPP gateway provides Gx-based integration with the Policy and Charging Rules Function (PCRF). The SRC 3GPP gateway uses the Gx interface to mediate between the PCRF and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. Figure 10 on page 132 shows an example network configuration where the SRC 3GPP gateway acts as a mediator between the PCRF and an MX Series router. The Gx interface on the SRC 3GPP gateway communicates with the PCRF using the Diameter protocol.

Figure 10: SRC 3GPP Gateway as a Mediator Between the PCRF and an MX Series Router

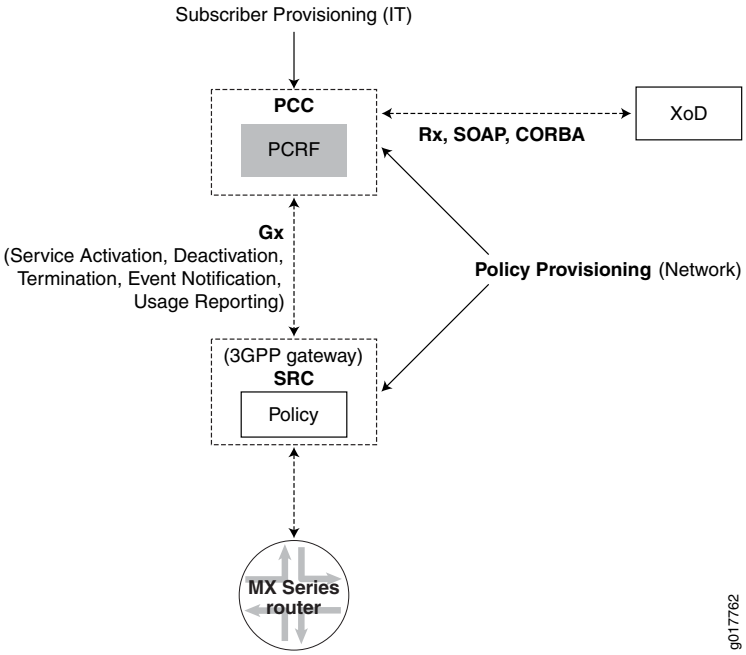
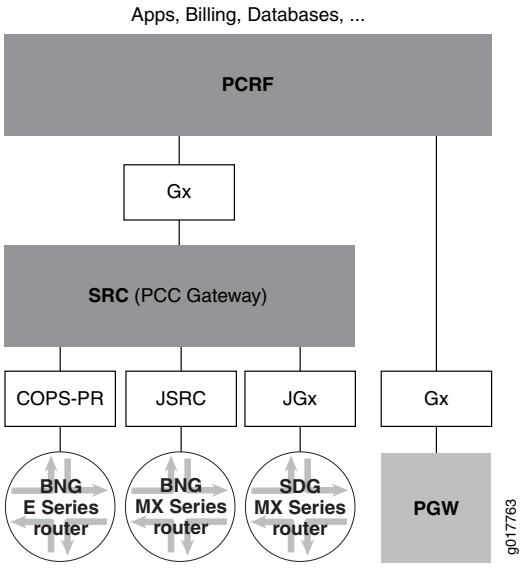


Figure 11 on page 132 represents the different policy references interfaces you can use, when using the SRC 3GPP gateway with other Juniper Networks products.

Figure 11: Various Interfaces on Juniper Networks Products Communicating with a PCRF Through the SRC 3GPP Gateway



The SRC 3GPP gateway supports the following scenarios:

- Subscriber Login Sequence on page 133
- Subscriber Logout Sequence on page 134

The following sections describe the sequences for each of these scenarios based on using an MX Series router.

Subscriber Login Sequence

The subscriber login uses the following sequence:

1. A subscriber session logs in and notifies the JSRC within the MX Series router.
2. The MX Series router sends a Diameter AA-Request (AAR) for the new subscriber session.
3. The service activation engine (SAE) performs necessary tasks such as creating the user session, provisioning any policy objects, and others.
4. When the login is complete, the SAE sends back a Diameter AA-Answer (AAA) to the router.
5. The SAE notifies the SRC 3GPP gateway that a subscriber was logged in successfully. This notification includes information such as the IP Address of the subscriber.
6. The SRC 3GPP gateway sends a Diameter Credit-Control-Request (CCR) with the following information to the PCRF:
   - CC-Request-Type AVP: INITIAL_REQUEST
   - Framed-IP-Address AVP: user equipment IPv4 address or Framed-IPv6-Prefix AVP: user equipment IPv6 address
   - Subscription-Id AVP: the login-name SAE plug-in attribute (Subscription-Id-Type set to END_USER_E164(0)) and, if available, the interface description SAE plug-in attribute (Subscription-Id-Type set to END_USER_PRIVATE(4))
   - CalledStationId AVP: Set to virtual router name.

   The SRC 3GPP gateway selects the PCRF according to the referenced Diameter peers under the **[edit slot 0 gw-3gpp gx diameter-peer]** hierarchy.
7. The SRC 3GPP gateway receives the Credit-Control-Answer (CCA) message from the PCRF.
   - If the PCRF rejected the CCR-I (initial) by providing an error code in the CCA message, the SRC 3GPP gateway logs the subscriber out using the SAE Common Object Request Broker Architecture (CORBA) interface.
   - If the PCRF accepted the CCR-I, the SRC 3GPP gateway iterates over the provided policy and charging control (PCC) rules (if any) from the CCA message and issues service activate calls for those services.

8. The SRC 3GPP gateway iterates over PPC Rules from the CCA message and issues service activate calls for those services, if applicable.
9. For each service to be activated, the SAE issues a Diameter Push-Profile-Request (PPR) to the MX Series router.
10. The MX Series router sends an acknowledgment to the SAE in a Push-Profile-Answer (PPA) message.
11. The SAE sends an acknowledgement to the SRC 3GPP gateway.

    The SRC 3GPP gateway collects all the failed service activations and sends a CCR-U (update) to the PCRF to indicate the failure. The SRC 3GPP gateway sets the experimental result code in the CCR-U to DIAMETER_PCC_RULE_EVENT(5142) and includes a charging-rule-report for each failed activation including the rule-name and the rule-failure-code.

Subscriber Logout Sequence

The subscriber logout uses the following sequence:

1. A subscriber session logs out and notifies the JSRC within the MX Series router.
2. The router sends a Diameter Session-Termination-Request (STR) to the SAE to indicate that the subscriber has logged out.
3. The SAE notifies the SRC 3GPP gateway that the service session has stopped.

   For each service stop event, the SRC 3GPP gateway issues a CC-Request to the PCRF that includes:

   - CC-Request-Type AVP: UPDATE_REQUEST
   - Usage-Monitoring-Information AVP includes:
     - Final accounting data
4. The SAE notifies the SRC 3GPP gateway that the subscriber was logged out successfully. The notification includes one of the following addresses:
   - Framed-IP-Address AVP: user equipment IPv4 address
   - Framed-IPv6-Prefix AVP: user equipment IPv6 address
5. The SRC 3GPP gateway sends a Diameter CCR message with the following information to the PCRF:
   - CC-Request-Type AVP: TERMINATION_REQUEST
   - Subscriber-Id AVP includes one of the following addresses:
     - Framed-IP-Address AVP: user equipment IPv4 address
     - Framed-IPv6-Prefix AVP: user equipment IPv6 address
6. The SRC 3GPP gateway receives the CCA message from the PCRF.
7. The SAE acknowledges the STR sent by the router by sending a Session-Termination-Answer (STA) to the router.

Subscriber Session Termination Sequence (Initiated by PCRF)

When the PCRF initiates the subscriber session termination, the following sequence occurs:

1. The application function requests the termination of the session from the PCRF.
2. The PCRF issues a Re-Auth-Request (RAR) message, which includes the Session-Release-Cause AVP.
3. The SRC 3GPP gateway issues a subscriber disconnect CORBA Call to the SAE. This method is not supported for JunosE Point-to-Point Protocol (PPP) sessions; in that case, the SRC 3GPP gateway only logs the subscriber out.
4. The SAE performs the necessary steps to disconnect the subscriber. For JunosE PPP sessions, the subscriber is logged out using the subscriber logout procedure.
5. The user equipment is disconnected.
6. The router sends an acknowledgement to the SAE.
7. The SAE acknowledges the CORBA Call by sending an acknowledgement to the SRC 3GPP gateway.
8. The SRC 3GPP gateway acknowledges the RAR message sent by the PCRF by sending a Re-Auth-Answer (RAA) message to the PCRF.
9. The SAE sends a Subscriber Stop Event message to the SRC 3GPP gateway.
10. The SRC 3GPP gateway sends a CCR with the following information to the PCRF:
    - CC-Request-Type AVP: TERMINATION_REQUEST
    - Subscriber-Id AVP includes one of the following addresses:
        - Framed-IP-Address AVP: user equipment IPv4 address
        - Framed-IPv6-Prefix AVP: user equipment IPv6 address
11. The SRC 3GPP gateway receives the CCA message.

Service Activation Sequence (Initiated by Application Function)

When the application function (AF) initiates a service activation, the following sequence occurs:

1. The AF receives any kind of trigger to set up a new AF session and issues an AAR message to the PCRF.
2. The PCRF stores all received service information, retrieves profiles, and sends an AA-Answer (AAA) message back to the AF.
3. The PCRF issues an RAR message, with the Charging-Rule-Install AVP set.
4. The SRC 3GPP gateway issues a subscriber activate service session CORBA Call to the SAE, by using the Bearer-Identity AVP (subscriber's IP address) and service name (Charging-Rule-Name AVP).

5. The SAE performs the necessary steps to activate the specified service for the given subscriber, including sending a Push-Profile-Request (PPR) message to the router.
6. The router sends an acknowledgement (PPA) to the SAE.
7. The SAE acknowledges the CORBA Call by sending an acknowledgement to the SRC 3GPP gateway.
8. The SRC 3GPP gateway sends an Re-Auth-Answer (RAA) message to the PCRF.
9. The SAE sends Service Start Event to the SRC 3GPP gateway.

Service Modification Sequence (Initiated by Application Function)

When the application function (AF) initiates a service modification, the following sequence occurs:

1. The AF receives any kind of trigger to update an existing AF session and issues an AAR message to the PCRF.
2. The PCRF stores all received service information, retrieves profiles, and sends an AA-Answer (AAA) message back to the AF.
3. The PCRF issues an RAR message, with the Charging-Rule-Install AVP set.
4. The SRC 3GPP gateway issues a subscriber-modify service CORBA Call to the SAE, which includes the subscriber IP address and the service name (Charging-Rule-Name AVP).
5. The SAE performs the necessary steps to modify the specified service for the given subscriber, including sending a Push-Profile-Request (PPR) message to the router.
6. The router sends an acknowledgement (PPA) to the SAE.
7. The SAE acknowledges the CORBA Call by sending an acknowledgement to the SRC 3GPP gateway.
8. The SRC 3GPP gateway sends a Re-Auth-Answer (RAA) message to the PCRF.

Service Deactivation Sequence (Initiated by Application Function)

When the application function (AF) initiates a service deactivation, the following sequence occurs:

1. The AF receives any kind of trigger to terminate or deactivate an existing session and issues an AAR message to the PCRF.
2. The PCRF stores all received service information, retrieves profiles, and sends an AA-Answer (AAA) message back to the AF.
3. The PCRF issues an RAR message with the Charging-Rule-Remove AVP set.
4. The SRC 3GPP gateway issues a subscriber-deactivate service CORBA Call to the SAE.
5. The SAE performs the necessary steps to deactivate the specified service for the given subscriber, including sending a Push-Profile-Request (PPR) message to the router.

6. The router sends an acknowledgement Push-Profile-Answer (PPA) to the SAE.
7. The SAE acknowledges the CORBA Call by sending an acknowledgement to the SRC 3GPP gateway.
8. The SRC 3GPP gateway sends a Re-Auth-Answer (RAA) message to the PCRF.
9. The SAE sends a Service Stop Event to the SRC 3GPP gateway. The SRC 3GPP gateway issues a CCR to the PCRF including:
   - CC-Request-Type AVP: UPDATE_REQUEST
   - Usage-Monitoring-Information AVP includes:
     - Final accounting data

Related Documentation

- Mapping Between SRC Software, Junos OS, and PCC Concepts on page 137
- SRC 3GPP Gateway Peer Communication and Redundancy on page 143
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147

Mapping Between SRC Software, Junos OS, and PCC Concepts

This section describes the mapping between the SRC software and Junos OS concepts and the policy and charging control (PCC) concepts. During these discussions, refer to Table 10 on page 137, which describes the mapping between SRC software and Junos OS terminology and the policy and charging control (PCC) function terminology.

Table 10: SRC Software and Junos OS Terminology Versus PCC Terminology

SRC Software and Junos OS Terminology	PCC Terminology
Subscriber session	IP CAN Session
Service with associated policies	PCC Rule
Service activation	Rule-Install
Service deactivation	Rule-Uninstall
Service accounting	Usage-Monitoring
Service templates, defined by SRC	Predefined Rule
Policies, defined by the PCRF	Dynamic Rule

Charging Rule Installation (Service Activation)

A PCRF can activate any number of non-parameterized and parameterized services (predefined PCC rules) in the same CCA or RAR message by providing a Charging-Rule-Install AVP. The Charging-Rule-Install AVP can contain multiple Charging-Rule-Name AVPs, one for each non-parameterized service to be activated.

The Charging-Rule-Install AVP can also contain multiple Charging-Rule-Definition AVPs, one for each parameterized service session that is to be activated.

NOTE: The names appearing in the Charging-Rule-Name AVPs must be unique; the same name must not appear multiple times in the same Gx message.

The SRC 3GPP gateway expects to receive the following AVPs from the PCRF in CCA and RAR messages:

```
Charging-Rule-Install ::= < AVP Header: 1001 >
       *[ Charging-Rule-Definition ]
       *[ Charging-Rule-Name ]
       *[ Charging-Rule-Base-Name ]
       *[ AVP ]
```

Where:

```
Charging-Rule-Definition ::= < AVP Header: 1003 >
   {Charging-Rule-Name}
   [Juniper-Substitution]

   *[Juniper-Substitution]

Juniper-Substitution::= < AVP Header: 2024>
     {Juniper-Substitution-Name}
     {Juniper-Substitution-Value"}
```

Table 11 on page 138 describes these AVPs.

Table 11: AVP Definitions

AVP	Code	Type	VendorID	Description
Charging-Rule-Name	1005	UTF8String	VID_3GPP	To activate a parameterized default service session, specify the serviceName in the format: ruleName@00010001 and specify the ruleName in numeric format.
Juniper-Substitution-Name	2025	UTF8String	VID_JNPR	Name of parameter as defined in the SRC policy definition.
Juniper-Substitution-Value	2026	UTF8String	VID_JNPR	Value to assign to the parameter.

Installing Non-Parameterized Predefined Charging Rules

Non-parameterized predefined charging rules are equivalent to activating an SAE service with no parameters.

The Charging-Rule-Install AVP provides the list of Charging-Rule-Names. The Charging-Rule-Name AVP sent by the PCRF must correspond to the SRC service name.

NOTE: The Charging-Rule-Base-Name AVP is not supported and is ignored by the SRC 3GPP gateway.

Installing Parameterized Predefined Charging Rules

Parameterized predefined charging rules are equivalent to activating an SAE service with parameters.

The Charging-Rule-Definition AVP must be provided with the list of parameters in a list of Juniper-Substitution-Name AVPs.

- For activating a default service session, the Charging-Rule-Name AVP must be set to the SRC service name.
- Specify the serviceName in the format ruleName@dynamicfixedPart. For example, “ruleName@00010001” where ruleName is in numeric format and the dynamicFixedPart is a fixed length (size is 8) identifier generated by the PCRF.

NOTE: Only default sessions are supported.

Example of Charging-Rule Installation

The following example Charging-Rule-Install AVP, sent by the PCRF, activates the services “foo1” and “foo2” with no parameters. It also activates the service “123” with two parameters and the service “456” with one parameter.

```
AVP: Charging-Rule-Install(1001)
        AVP: Charging-Rule-Name(1005) val=foo1  <- Activate service foo1
        AVP: Charging-Rule-Name(1005) val=foo2  <- Activate service foo2
    AVP: Charging-Rule-Definition(1003) vnd=VID_3GPP  <- 3GPP AVP for activating
 parameterized service “123” with 2 parameters
            AVP: Charging-Rule-Name (1005) vnd=3GPP val=123@00010000
      AVP: Juniper-Substitution (2024) vnd=JNPR
                AVP: Juniper-Substitution-Name(2025) vnd=JNPR val=rate
           AVP: Juniper-Substitution-Value(2026) vnd=JNPR val=5
              AVP: Juniper-Substitution (2024) vnd=JNPR
           AVP: Juniper-Substitution-Name(2025) vnd=JNPR val=color
                 AVP: Juniper-Substitution-Value(2026) vnd=JNPR val=red
AVP: Charging-Rule-Definition(1003) vnd=VID_3GPP  <- 3GPP AVP for activating
parameterized service “456” with 1 parameter
            AVP: Charging-Rule-Name (1005) vnd=3GPP val=456@00010001
      AVP: Juniper-Substitution (2024) vnd=JNPR
           AVP: Juniper-Substitution-Name(2025) vnd=JNPR val=rate
           AVP: Juniper-Substitution-Value(2026) vnd=JNPR val=10
```

In this example, foo1, foo2, 123 (rate, color), and “456” (rate) are configured services in the SRC software.

Charging Rule Removal (Service Deactivation)

A PCRF can deactivate any number of non-parameterized and parameterized services (predefined PCC rules) in the same CCA or RAR message by providing a Charging-Rule-Remove AVP. The Charging-Rule-Remove AVP can contain multiple Charging-Rule-Name AVPs, one for each non-parameterized or parameterized service to be deactivated.

The following AVPs are expected by the SRC (PCEF) from the PCRF in CCA and RAR messages:

```
Charging-Rule-Remove ::= < AVP Header: 1002 >
       *[ Charging-Rule-Definition ]
       *[ Charging-Rule-Name ]
       *[ Charging-Rule-Base-Name ]
       *[ AVP ]
```

- For non-parameterized charging rules, the Charging-Rule-Name AVP must correspond to the one provided in the Charging-Rule-Install AVP.
- For parameterized charging rules, the Charging-Rule-Name AVP must correspond to the one provided in the Charging-Rule-Definition AVP. If this format is not specified, the value in the Charging-Rule-Name AVP is treated as the serviceName.

NOTE: Charging-Rule-Base-Name AVP is not supported and is ignored by the SRC 3GPP gateway.

Example of Charging-Rule Removal

The following example Charging-Rule-Removal AVP, sent by the PCRF, deactivates the SRC services called “foo1”, “foo2”, and “123”.

```
AVP: Charging-Rule-Remove(1002)
             AVP: Charging-Rule-Name(1005) val=foo1 <- Deactivate service foo1
             AVP: Charging-Rule-Name(1005) val=foo2 <- Deactivate service foo2
           AVP: Charging-Rule-Name(1005) val=123@00010001  <- Deactivate service
 123
```

Charging Rule Report

The SRC 3GPP gateway can send charging rule reports for any number of non-parameterized and parameterized services in the same Credit Control Update (CCR-U) request or RAA message. This is achieved by providing a Charging-Rule-Report AVP for each failed service. The Charging-Rule-Report AVP contains a single Charging-Rule-Name AVP (for a non-parameterized or a parameterized service).

```
Charging-Rule-Report ::= < AVP Header: 1018 >
       *[ Charging-Rule-Name ]
       *[ Charging-Rule-Base-Name ]
     [ PCC-Rule-Status ]
       [ Rule-Failure-Code ]
    *[ AVP ]
```

- For non-parameterized charging rules, the Charging-Rule-Name AVP corresponds to the one provided in the Charging-Rule-Install AVP.
- For parameterized charging rules, the Charging-Rule-Name AVP must correspond to the one provided in the Charging-Rule-Definition AVP. If this format is not specified, the value in the Charging-Rule-Name AVP is treated as the serviceName.

NOTE: The Charging-Rule-Base-Name AVP is not supported and is never sent by the SRC 3GPP gateway.

Service Accounting

You can perform service accounting for one or more PCC rules.

When a PCRF requests service accounting, it needs to include an Event-Trigger AVP, set to “USAGE_REPORT”. This setting must be set in either the RAR message (if the PRCF initiates the PCC rule changes), or the CCA message (if the user equipment initiates the rule changes).

The PCRF may also provide usage threshold levels to the SRC 3GPP gateway at session establishment or modification time (CCA or RAR message). This is done, by setting those thresholds in the grouped Grant-Service-Unit AVP per Monitoring-Key in the Usage-Monitoring-Information AVP. The threshold level may be defined for:

- Total volume only (CC-Total-Octets AVP within Granted-Service-Unit hold threshold for total volume)
- Uplink volume only (CC-Input-Octets AVP within Granted-Service-Unit hold threshold for uplink volume)
- Downlink volume only (CC-Output-Octets AVP within Granted-Service-Unit hold threshold for downlink volume)

The Monitoring-Key AVP format is similar to the format of the Charging-Rule-Removal and Charging-Rule-Report AVPs:

- For non-parameterized charging rules, the Monitoring-Key AVP must be set to the Charging-Rule-Name AVP provided in the Charging-Rule-Install AVP.
- For parameterized charging rules, the Monitoring-Key AVP must be set to the Charging-Rule-Name AVP provided in the Charging-Rule-Install AVP (from the Charging-Rule-Definition AVP).

The SRC 3GPP gateway does not support SESSION_LEVEL monitoring. This means that the only supported value for the Usage-Monitoring-Level AVP is PCC_RULE_LEVEL.

The SRC 3GPP gateway sends accounting updates when it receives interim updates from the SAE for the service session. This is done by setting the usage counters in the Used-Service-Unit AVP within the Usage-Monitoring-Information AVP. Like the Granted-Service-Unit AVP (for setting the threshold), the Used-Service-Unit AVP is a grouped AVP and the SRC 3GPP gateway uses the CC-Total-Octet, CC-Input-Octets,

and CC-Output-Octets AVP within the Used-Service-Unit AVP to report the usage to the PCRF. The SRC 3GPP gateway sends this only in CCR messages (not in RAA messages). The reporting is done when any of the following conditions are met:

- When a usage threshold is reached.
  - The SRC 3GPP gateway stores the threshold information (provided by the PCRF) in the SAE session during service activation or modification.
  - After the SRC 3GPP gateway receives an interim update for a service session, it checks whether any of the thresholds were reached.
  - If a threshold is reached, the SRC 3GPP gateway generates a CCR-U to the PCRF and includes the accumulated usage volume in the Usage-Monitoring-Information AVP.
  - The Event-Trigger AVP is set to USAGE_REPORT.
- When the service is deactivated (the PCC rule, for which service accounting is enabled, is removed).
  - This is done after the SRC 3GPP gateway deactivates a given service (in response to an RAR or CCA message).
  - The SRC 3GPP gateway generates a CCR-U to the PCRF and includes the accumulated usage volume in the Usage-Monitoring-Information AVP.
  - The Event-Trigger AVP is set to USAGE_REPORT.
- When service accounting is explicitly disabled by the PCRF for a specific PCC rule.
  - This is done if the SRC 3GPP gateway receives a CCA or RAR message with the Usage-Monitoring-Support AVP set to USAGE_MONITORING_DISABLED for a specific Monitoring-Key within the Usage-Monitoring-Information AVP.
  - The SRC 3GPP gateway generates a CCR-U to the PCRF and includes the accumulated usage volume in the Usage-Monitoring-Information AVP.
  - The Event-Trigger AVP is set to USAGE_REPORT.
- When an IP-CAN session is terminated (either by the PCRF through an RAR message, or initiated by the user equipment).
  - For each service (the PCC rule, for which service accounting is enabled):
    - The SRC 3GPP gateway generates a CCR-U to the PCRF and includes the accumulated usage volume in the Usage-Monitoring-Information AVP for the specific service.
    - The Event-Trigger AVP is set to USAGE_REPORT.
  - The SRC 3GPP gateway also generates a termination request (CCR-T) message to the PCRF but does not include any usage report in it.
- When requested by the PCRF.

- This is done if the SRC 3GPP gateway receives an RAR message with the Usage-Monitoring-Report AVP set to USAGE_MONITORING_REPORT-REQUIRED within the Usage-Monitoring-Information AVP.
- The SRC 3GPP gateway generates a CCR-U to the PCRF for each requested Monitoring-Key AVP and includes the accumulated usage volume in the Usage-Monitoring-Information AVP.
- The Event-Trigger AVP is set to USAGE_REPORT

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- SRC 3GPP Gateway Peer Communication and Redundancy on page 143
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147

SRC 3GPP Gateway Peer Communication and Redundancy

The SRC 3GPP gateway configuration allows the definition of multiple Diameter peers to act as PCRFs. The SRC 3GPP gateway picks one peer to become the “active” peer, and forwards all CCR messages to it. If the SRC software detects a failure of the active peer, the SRC 3GPP gateway selects another peer to become the active peer. The SRC 3GPP gateway does not switch back to the original active peer when it comes back online.

For redundancy purposes, you can configure multiple SRC 3GPP gateways to handle the communication between the same group of SAEs and PCRFs. To achieve this redundancy, you configure multiple SRC 3GPP gateway instances to point to the same namespace in the shared configuration.

All SRC 3GPP gateway instances pointing to the same namespace use a Community Manager to elect the active member. The active member registers itself with the naming server and is the only member receiving subscriber-tracking events from the SAE and forwarding them to the PCRF.

NOTE: **All the other SRC 3GPP gateway community members can still receive and process RAR messages from PCRF peers.**

If the active SRC 3GPP gateway member fails, the Community Manager detects the failure and elects another member to become the active member. The new active member registers itself with the naming server thus overwriting the old member’s endpoints. During the failure period, the SAE keeps a fail queue for the SRC 3GPP plug-in and replays the queue to the new active member.

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Mapping Between SRC Software, Junos OS, and PCC Concepts on page 137
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147

CHAPTER 14

Configuring and Managing the SRC 3GPP Gateway (SRC CLI)

Configuration Statements for the SRC 3GPP Gateway

Use the following configuration statements to configure the SRC 3GPP gateway at the **[edit]** hierarchy level.

slot *number* gw-3gpp {

```
    shared shared;
}
slot number gw-3gpp gx {
    destination-host destination-host;
    destination-realm destination-realm;
    diameter-peer [diameter-peer.....];
    protocol (tcp | sctp);
    port port;
    address address;
    origin-host origin-host;
    origin-realm origin-realm;
}
slot number gw-3gpp initial {
    static-dn static-dn;
    dynamic-dn dynamic-dn;
}
slot number gw-3gpp initial directory-connection {
    url url;
    backup-urls [backup-urls...];
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}
slot number gw-3gpp initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}
slot number gw-3gpp java-orb object-adapter {
    address address;
}
slot number gw-3gpp logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number gw-3gpp logger name syslog {
    facility facility
    filter filter;
    format format;
    host host;
    port port;
}
shared gw-3gpp group name
shared gw-3gpp configuration subscriber-types (session-handle) {
    subscriber-id-type (session-handle);
    nic-proxy nic-proxy;
}
shared gw-3gpp configuration nic-proxy-configuration name {
```

```
}
shared gw-3gpp configuration nic-proxy-configuration name resolution {
   resolver-name resolver-name;
   key-type key-type;
   value-type value-type;
   expect-multiple-values;
   constraints constraints;
}
shared gw-3gpp configuration nic-proxy-configuration name cache {
   cache-size cache-size;
   cache-cleanup-interval cache-cleanup-interval;
   cache-entry-age cache-entry-age;
}
shared gw-3gpp configuration nic-proxy-configuration name nic-host-selection {
   groups [groups...];
   selection-criteria (roundRobin | randomPick | priorityList);
}
shared gw-3gpp configuration nic-proxy-configuration name nic-host-selection blacklisting
   {
   try-next-system-on-error;
   number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
   blacklist-retry-interval blacklist-retry-interval;
}
shared gw-3gpp configuration nic-proxy-configuration name test-nic-bindings {
   use-test-bindings;
}
shared gw-3gpp configuration nic-proxy-configuration name test-nic-bindings key-values
   name {
   value;
}
```

Related Documentation

- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- SRC 3GPP Gateway Overview on page 131
- Mapping Between SRC Software, Junos OS, and PCC Concepts on page 137
- Configuring the SRC 3GPP Gateway Gx Interface (SRC CLI) on page 155

Configuring the SRC 3GPP Gateway (SRC CLI)

To configure the SRC 3GPP gateway:

1. Configure initial properties, including the connection to the directory and directory-monitoring properties.

   See "Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI)" on page 149.

   See "Changing the Location of Data in the Directory" on page 149.

   See "Configuring Directory-Connection Properties for the SRC 3GPP Gateway (SRC CLI)" on page 150.

   See "Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gateway (SRC CLI)" on page 151.

2. Configure the connection to the PCRF (remote Diameter peer).

   See "Configuring Diameter Peers (SRC CLI)" on page 152.

3. Configure the SRC 3GPP gateway Gx interface.

   See "Configuring the SRC 3GPP Gateway Gx Interface (SRC CLI)" on page 155.

4. Configure the basic local properties for the SRC 3GPP gateway.

   See "Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI)" on page 156.

5. Configure the object adapter Internet address.

   "Configuring the Object Adapter Internet Address for the SRC 3GPP Gateway (SRC CLI)" on page 157.

6. Configure logging destinations.

   See "Configuring Logging Destinations to Store Messages in a File (SRC CLI)" on page 158.

   See "Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)" on page 159.

7. Create an SRC 3GPP gateway grouped configuration.

   See "Creating Grouped Configurations for the SRC 3GPP Gateway (SRC CLI)" on page 159.

8. Configure subscriber types.

   See "Configuring the Subscriber Type (SRC CLI)" on page 160.

9. Configure the NIC proxies.

   See "Configuring a NIC Proxy for the SRC 3GPP Gateway (SRC CLI)" on page 161.

10. Start the SRC 3GPP gateway.

    See "Starting the SRC 3GPP Gateway (SRC CLI)" on page 176.

11. Configure the SAE for the SRC 3GPP gateway.

    See "Configuring the SAE for the SRC 3GPP Gateway" on page 166.

Related Documentation

- SRC 3GPP Gateway Overview on page 131

- Configuration Statements for the SRC 3GPP Gateway on page 145
- Mapping Between SRC Software, Junos OS, and PCC Concepts on page 137

Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI)

Use the following configuration statements to configure initial properties for the SRC 3GPP gateway:

```
slot number gw-3gpp initial {
   static-dn static-dn;
   dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

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

   ```
   user@host# edit slot 0 gw-3gpp initial
   ```

2. Specify the properties for the SRC 3GPP gateway.

   ```
   [edit slot 0 gw-3gpp initial]
   user@host# set ?
   ```

   For more information about configuring local properties for SRC components, see "Changing the Location of Data in the Directory" on page 149.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp initial]
   user@host# show
   ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuration Statements for the SRC 3GPP Gateway on page 145
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- Configuring Directory-Connection Properties for the SRC 3GPP Gateway (SRC CLI) on page 150

Changing the Location of Data in the Directory

In most cases, you use the default configuration for the location of SRC data in the directory:

- Administrator-defined configuration data—*ou=staticConfiguration,ou=Configuration,o=Management,o=umc*
- Programmatically defined configuration data—*ou=dynamicConfiguration,ou=Configuration,o=Management, o=umc*

You can specify the full distinguished name (DN), or a DN relative to a base DN, identified as *<base>*.

You can change the location of data in the directory at the Expert CLI editing level.

Use the following configuration statements to change the location of data for a component in the directory:

```
slot number component-name initial {
  static-dn static-dn ;
  dynamic-dn dynamic-dn ;
}
```

To change the location of data in the directory:

1. From configuration mode, access the configuration statement that specifies the configuration for a component on a slot.

   ```
   [edit]
   user@host# edit slot number nic initial
   ```

   For example:

   ```
   [edit]
   user@host# edit slot 0 nic initial
   ```

2. (Optional) Change the location of administrator-defined configuration data in the directory.

   ```
   [edit slot 0 nic initial]
   user@host# set static-dn static-dn
   ```

3. (Optional) Change the location of programmatically defined configuration data in the directory.

   ```
   [edit slot 0 nic initial]
   user@host# set dynamic-dn dynamic-dn
   ```

Related Documentation

- *Configuring Initial Directory Eventing Properties for SRC Components*
- *Configuring Basic Local Properties*
- *Configuration Statements for Local Configuration*
- *Managing Directory Communication*

Configuring Directory-Connection Properties for the SRC 3GPP Gateway (SRC CLI)

Use the following configuration statements to configure directory-connection properties for the SRC 3GPP gateway:

```
slot number gw-3gpp initial directory-connection {
  url url;
  backup-urls [backup-urls...];
```

```
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}
```

To configure directory-connection properties:

1. From configuration mode, access the statement that configures the directory-connection properties.

   ```
   user@host# edit slot 0 gw-3gpp initial directory-connection
   ```

2. Specify the properties for the SRC 3GPP gateway.

   ```
   [edit slot 0 gw-3gpp initial directory-connection]
   user@host# set ?
   ```

   For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp initial directory-connection]
   user@host# show
   url ldap://127.0.0.1:389/;
   principal cn=conf,o=Operators,<base>;
   credentials ********;
   ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149
- Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gateway (SRC CLI) on page 151
- Configuration Statements for the SRC 3GPP Gateway on page 145

Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gateway (SRC CLI)

Use the following configuration statements to configure initial directory-eventing properties for the SRC 3GPP gateway:

```
slot number gw-3gpp initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory-eventing properties:

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

   ```
   user@host# edit slot 0 gw-3gpp initial eventing
   ```

2. Specify the initial directory-eventing properties for SRC 3GPP gateway.

   ```
   [edit slot 0 gw-3gpp initial directory-eventing]
   user@host# set ?
   ```

   For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp initial directory-eventing]
   user@host# show
   eventing;
   polling-interval 30;
   ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149
- Configuring Directory-Connection Properties for the SRC 3GPP Gateway (SRC CLI) on page 150
- Configuration Statements for the SRC 3GPP Gateway on page 145

Configuring Diameter Peers (SRC CLI)

Use the following configuration statements to configure the Diameter peers:

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

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

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

To configure the Diameter peer:

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

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

   The peer name must be unique.

2. Specify the protocol for the transport connection.

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

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

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

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

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

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

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

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

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

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

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

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

   [edit shared network diameter peer *name*]

```
user@host# set state-machine-timeout state-machine-timeout
```

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

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

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

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

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

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

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

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

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

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

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

Related Documentation

- *Configuring the Diameter Application (SRC CLI)*
- *Viewing SRC Diameter Server State (SRC CLI)*

Configuring the SRC 3GPP Gateway Gx Interface (SRC CLI)

Use the following configuration statements to configure the SRC 3GPP gateway Gx interface:

```
slot number gw-3gpp gx {
    destination-host destination-host;
    destination-realm destination-realm;
    diameter-peer [diameter-peer.....];
    protocol (tcp | sctp);
    port port;
    address address;
    origin-host origin-host;
    origin-realm origin-realm;
}
```

NOTE: The SRC 3GPP gateway uses its own Diameter stack, which is configured under the **[edit slot 0 gw-3gpp gx]** hierarchy. It does not use the Diameter stack configured under the **[edit system diameter]** hierarchy; this Diameter stack is used for SAE and router communication.

To configure the SRC 3GPP gateway Gx interface:

1. From configuration mode, access the statement that configures the SRC 3GPP gateway Gx interface.

   user@host# **edit slot 0 gw-3gpp gx**

2. (Optional) Specify the Diameter identifier for the remote endpoint, which is the destination of the Diameter message. The Destination-Host AVP (AVP Code 293) is of the DiameterIdentity type and is present in all Diameter messages.

   [edit slot 0 gw-3gpp gx]
   user@host# **set destination-host *destination-host***

3. Specify the Diameter identifier for the realm of the remote endpoint, which is the destination of the Diameter message. The Destination-Realm AVP (AVP Code 283) is of the DiameterIdentity type and is present in all Diameter messages.

   [edit slot 0 gw-3gpp gx]
   user@host# **set destination-realm *destination-realm***

4. Specify the list of remote Diameter peers (PCRFs) that connect to the SRC 3GPP gateway over the Gx interface.

   [edit slot 0 gw-3gpp gx]
   user@host# **set diameter-peer [*diameter-peer*....]**

   Each Diameter peer you specify must be previously configured under the **[edit shared network diameter peer]** hierarchy.

5. (Optional) Specify the protocol for the transport connection.

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set protocol [(tcp | sctp)...]
   ```

6. (Optional) Specify the port to use for incoming connections.

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set port port
   ```

7. Specify the local address of the peer.

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set address address
   ```

8. Specify the Diameter identifier for the local endpoint that is the originator of the Diameter message.

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set origin-host origin-host
   ```

9. Specify the Diameter identifier for the realm of the local endpoint that is the originator of the Diameter message.

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set origin-realm origin-realm
   ```

10. (Optional) Verify your configuration.

    ```
    [edit slot 0 gw-3gpp gx]
    user@host# show
    address 10.10.10.10;
    destination-host testpcrf;
    destination-realm englab.juniper.net;
    diameter-peer [ primary-pcrf secondary-pcrf ];
    origin-host duke;
    origin-realm example;
    port 3868;
    protocol tcp;
    }
    ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuration Statements for the SRC 3GPP Gateway on page 145
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147

Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI)

The local configuration for the SRC 3GPP gateway component defines the directory where the shared configuration is and points to the component namespace.

Use the following configuration statements to configure basic local properties for the SRC 3GPP gateway:

```
slot number gw-3gpp {
  shared shared;
}
```

To configure basic local properties:

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

   ```
   [edit]
   user@host# edit slot 0 gw-3gpp
   ```

2. Specify the configuration namespace for the SRC 3GPP gateway as the path, relative to the root of the static configuration properties, that defines the object for the namespace.

   ```
   [edit slot 0 gw-3gpp]
   user@host# set shared shared
   ```

   For example:

   ```
   [edit slot 0 gw-3gpp]
   user@host# set shared /sample
   ```

   NOTE: All SRC 3GPP gateway instances pointing to the same shared namespace run in the same redundant community.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp]
   user@host# show
   ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuration Statements for the SRC 3GPP Gateway on page 145
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149

Configuring the Object Adapter Internet Address for the SRC 3GPP Gateway (SRC CLI)

Use the following configuration statements to configure the object adapter Internet address:

```
slot number gw-3gpp java-orb object-adapter {
  address address;
}
```

To configure the object adapter Internet address:

1. From configuration mode, access the statement that configures the object adapter Internet address.

   ```
   user@host# edit slot number gw-3gpp java-orb object-adapter
   ```

2. Configure the address of the object adapter.

   ```
   [edit slot number gw-3gpp java-orb object-adapter]
   user@host# set address address
   ```

Related Documentation

- SRC 3GPP Gateway Overview on page 131
- Configuration Statements for the SRC 3GPP Gateway on page 145
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147

Configuring Logging Destinations to Store Messages in a File (SRC CLI)

Use the following configuration statements to configure file logging for the SRC 3GPP gateway:

```
slot number gw-3gpp logger name ...
slot number gw-3gpp logger name file {
   filter filter;
   filename filename;
   rollover-filename rollover-filename;
   maximum-file-size maximum-file-size;
}
```

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

1. From configuration mode, access the statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log1 is configured.

   ```
   user@host# edit slot 0 gw-3gpp logger log1 file
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 gw-3gpp logger log1 file]
   user@host# set ?
   ```

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

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp logger log1 file]
   user@host# show
   filter /info-;
   filename var/log/gw-3gpp-info.log;
   rollover-filename var/log/gw-3gpp-info.alt;
   maximum-file-size 2000000;
   ```

Related Documentation

- Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI) on page 156
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149
- Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI) on page 159

Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)

Use the following configuration statements to configure system logging for the SRC 3GPP gateway:

```
slot number gw-3gpp logger name ...
slot number gw-3gpp logger name syslog {
    facility facility
    filter filter;
    format format;
    host host;
    port port;
}
```

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

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

   ```
   user@host# edit slot 0 gw-3gpp logger log2 syslog
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 gw-3gpp logger log2 syslog]
   user@host# set ?
   ```

   For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gw-3gpp logger log2 syslog]
   user@host# show
   ```

Related Documentation

- Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI) on page 156
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149
- Configuring Logging Destinations to Store Messages in a File (SRC CLI) on page 158

Creating Grouped Configurations for the SRC 3GPP Gateway (SRC CLI)

Configuration groups allow you to share the SRC 3GPP gateway configuration with different SRC 3GPP gateway instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped SRC 3GPP gateway configuration that is shared among multiple SRC 3GPP gateway instances. For example, if you create two different SRC 3GPP gateway groups called config1 and config2 within the shared SRC 3GPP gateway configuration, you could select the SRC 3GPP gateway configuration that should be associated with a particular SRC 3GPP gateway instance.

Use the **shared** option of the **slot *number* gw-3gpp** statement to select the group for an SRC 3GPP gateway instance as part of the local configuration. Use the **shared gw-3gpp group *name* configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an SRC 3GPP gateway instance. For example, to select a group called config1 in the root group:

   ```
   [edit]
   user@host# set slot 0 gw-3gpp shared /config1
   ```

2. Commit the configuration.

   ```
   [edit]
   user@host# commit
   commit complete.
   ```

3. From configuration mode, configure a group. For example, to configure a group called config1, specify the group as part of the SRC 3GPP gateway configuration.

   ```
   [edit]
   user@host# edit shared gw-3gpp group config1
   ```

Related Documentation

- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI) on page 156
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149

Configuring the Subscriber Type (SRC CLI)

Use the following configuration statements to configure the subscriber type:

```
shared gw-3gpp configuration subscriber-types (session-handle) {
   subscriber-id-type (session-handle);
   nic-proxy nic-proxy;
}
```

To configure the subscriber type:

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

   ```
   user@host# edit shared gw-3gpp configuration subscriber-types session-handle)
   ```

2. Specify the subscriber ID type.

   NOTE: The only subscriber ID type supported is session-handle.

   ```
   edit shared gw-3gpp configuration subscriber-types session-handle
   user@host# set subscriber-id-type session-handle)
   ```

3. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy. In this sample procedure, the namespace for the NIC proxy called nic2 is configured.

   ```
   edit shared gw-3gpp configuration subscriber-types session-handle
   user@host# set nic-proxy nic2
   ```

4. (Optional) Verify your configuration.

   ```
   [edit shared configuration subscriber-types session-handle]
   user@host# show
   nic-proxy nic2;
   subscriber-id-type session-handle;
   ```

Related Documentation

- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- Configuring Basic Local Properties for the SRC 3GPP Gateway (SRC CLI) on page 156
- Configuring Initial Properties for the SRC 3GPP Gateway (SRC CLI) on page 149
- Configuring a NIC Proxy for the SRC 3GPP Gateway (SRC CLI) on page 161

Configuring a NIC Proxy for the SRC 3GPP Gateway (SRC CLI)

Tasks to configure the NIC proxy are:

- Configuring Resolution Information for a NIC Proxy on page 161
- Changing the Configuration for the NIC Proxy Cache on page 162
- Configuring a NIC Proxy for NIC Replication on page 163
- Configuring NIC Test Data on page 165

Configuring Resolution Information for a NIC Proxy

You create a NIC proxy for each subscriber type to be configured. Subscriber types that have different subscriber ID types can use the same NIC proxy.

Use the following configuration statements to configure the NIC proxy:

```
shared gw-3gpp configuration nic-proxy-configuration name {
}
shared gw-3gpp configuration nic-proxy-configuration name resolution {
  resolver-name resolver-name;
  key-type key-type;
  value-type value-type;
  expect-multiple-values;
  constraints constraints;
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called nic2 is configured.

   ```
   user@host# edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution
   ```

2. Specify the path to the NIC resolver that this NIC proxy uses. This resolver must be the same as the one that is configured on the NIC host.

   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
   user@host# set resolver-name *resolver-name*

3. Specify the NIC data type that the key provides for the NIC resolution. You can provide a qualifier to a data type to distinguish between different instances of a data type in a resolution scenario, or to provide information about a data type to clarify the use of that data type in a resolution.

   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
   user@host# set key-type *key-type*

   NOTE: The only valid **key-type** for the SRC 3GPP gateway is **SessionHandle**.

4. Specify the type of value to be returned in the resolution for the application that uses the NIC proxy.

   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
   user@host# set value-type *value-type*

   For the SRC 3GPP gateway, you must set **value-type** to **SaeId**.

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
   user@host# set expect-multiple-values

6. (Optional. Available at the Advanced editing level.) If the application provides a constraint in the resolution request, specify the data type for the constraint. The constraint represents a condition that must or may be satisfied before the next stage of the resolution process can proceed.

   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
   user@host# set constraints *constraints*

   Configure a constraint only if the constraint will be provided by the application in the resolution request. Typically, you do not need to configure constraints.

7. (Optional) Verify your configuration.

```
[edit shared gw-3gpp configuration nic-proxy-configuration nic2 resolution]
user@host# show
resolver-name /realms/nic2/A1;
key-type SessionHandle;
value-type SaeId;
```

Changing the Configuration for the NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default

settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
shared gw-3gpp configuration nic-proxy-configuration name cache {
  cache-size cache-size;
  cache-cleanup-interval cache-cleanup-interval;
  cache-entry-age cache-entry-age;
}
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called nic2 is configured.

   ```
   user@host# edit shared gw-3gpp configuration nic-proxy-configuration nic2 cache
   ```

2. (Optional) Specify the maximum number of keys for which the NIC proxy retains data.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 cache]
   user@host# set cache-size cache-size
   ```

   If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.

3. Specify the time interval at which the NIC proxy removes expired entries from its cache.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 cache]
   user@host# set cache-cleanup-interval cache-cleanup-interval
   ```

4. (Optional) Specify how long an entry remains in the cache.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 cache]
   user@host# set cache-entry-age cache-entry-age
   ```

5. (Optional) Verify your configuration.

   ```
   [edit shared configuration nic-proxy-configuration nic2 cache]
   user@host# show
   cache-size 10000;
   cache-cleanup-interval 15;
   ```

Configuring a NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Normal editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```
shared gw-3gpp configuration nic-proxy-configuration name nic-host-selection {
  groups [groups...];
```

```
    selection-criteria (roundRobin | randomPick | priorityList);
  }
  shared gw-3gpp configuration nic-proxy-configuration name nic-host-selection blacklisting
    {
    try-next-system-on-error;
    number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
    blacklist-retry-interval blacklist-retry-interval;
  }
```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called nic2 is configured.

   ```
   user@host# edit shared gw-3gpp configuration nic-proxy-configuration nic2
     nic-host-selection
   ```

2. (Optional) Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection]
   user@host# set groups groups
   ```

3. (Optional) If you configure more than one group, specify the selection criteria that the NIC proxy uses to determine which NIC host to contact.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection]
   user@host# set selection-criteria (roundRobin | randomPick | priorityList)
   ```

   where:

   - roundRobin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
   - randomPick—NIC proxy selects NIC hosts randomly from the list.
   - priorityList—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.

     Priorities are defined by the order in which you specify the groups. You can change the order of NIC hosts in the list by using the **insert** command.

4. (Optional) Verify your configuration.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2
   nic-host-selection]
   user@host# show
   groups ;
   selection-criteria round-;
   ```

5. Access the statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection]
   user@host# edit blacklisting
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection
     blacklisting]
   ```

6. (Optional) Specify whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection
     blacklisting]
   user@host# set try-next-system-on-error
   ```

7. (Optional) Change the number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time. The default is 3.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection
     blacklisting]
   user@host# set number-of-retries-before-blacklisting
     number-of-retries-before-blacklisting
   ```

8. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2 nic-host-selection
     blacklisting]
   user@host# set blacklist-retry-interval blacklist-retry-interval
   ```

9. (Optional) Verify your configuration.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2
   nic-host-selection blacklisting]
   user@host# show
   try-next-system-on-error;
   number-of-retries-before-blacklisting 3;
   blacklist-retry-interval 15;
   ```

Configuring NIC Test Data

To test a resolution without the NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the NIC proxy configuration. When the SRC component configured to use a NIC proxy stub passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value. When you use a NIC proxy stub, no NIC infrastructure is required.

Use the following configuration statements to configure a NIC proxy stub from the **[edit]** hierarchy level.

```
shared gw-3gpp configuration nic-proxy-configuration name test-nic-bindings {
  use-test-bindings;
}
shared gw-3gpp configuration nic-proxy-configuration name test-nic-bindings key-values
  name {
  value;
}
```

To use the NIC proxy stub for the SRC 3GPP gateway:

1. In configuration mode, navigate to the NIC proxy configuration and specify the data type of the key you want to map to a value. In this sample procedure, the key nic2 is specified for the NIC proxy called nic2.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2]
   user@host# set resolution key-type nic2
   ```

2. Enable a NIC proxy stub for a resolution.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2]
   user@host# set test-nic-bindings use-test-bindings
   ```

3. Specify the values of the keys for testing. These statements are available at the Advanced CLI editing level.

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2]
   user@host# set test-nic-bindings key-values name value
   ```

   where:

   - *name*—Indicates the NIC data value for the proxy.
   - *value*—Specifies a value for the NIC data type.

   For example, to set up a login name to IP mapping for login name jane@virneo.com to the IP address 192.0.2.30:

   ```
   [edit shared gw-3gpp configuration nic-proxy-configuration nic2]
   user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
   ```

Related Documentation

- *Configuring NIC Test Data (SRC CLI)*
- *Configuring a NIC Proxy for NIC Replication (SRC CLI)*
- Configuration Statements for the SRC 3GPP Gateway on page 145

Configuring the SAE for the SRC 3GPP Gateway

You must configure the SAE to recognize the SRC 3GPP gateway by adding information about it to the SAE properties. Tasks for configuring the SAE for the SRC 3GPP gateway are:

- Configuring the SRC 3GPP Gateway as an External Plug-In on page 166
- Configuring Event Publishers on page 167

Configuring the SRC 3GPP Gateway as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the statement that configures the external plug-ins.

   ```
   user@host# edit shared sae configuration plug-ins name name external
   ```

2. Configure the object reference of the external plug-in that is exported to the SAE.

   ```
   [edit shared sae configuration plug-ins name name external]
   user@host# set corba-object-reference corba-object-reference
   ```

where ***corba-object-reference*** is one of the following references:

- Path to the interoperable object reference (IOR) file in the format file:///opt/UMC/3gpp/var/run/3gpp-gw.ior
- The corbaloc URL in the format corbaloc::*host*:9801/ASGGx/statesync/<namespace> where *host* is the IP address of the C Series Controller or 127.0.0.1
- Common Object Services (COS) in the format corbaname::*host*:2809/NameService#ASGGx/statesync/<namespace>

3. Specify the plug-in attributes.

   [edit shared sae configuration plug-ins name *name* external]
   user@host# set attributes attribute

   Plug-in attributes for the SRC 3GPP gateway include:

   USER-SESSION-HANDLE

   LOGIN-NAME

   INTERFACE-DESCR

   USER-INET-ADDRESS

   PROPERTY

   SERVICE-SESSION-NAME

   SERVICE-NAME

   IN-OCTETS

   OUT-OCTETS

   TERMINATE-CAUSE

   ROUTER-NAME

   For more information about configuring plug-in attributes, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

Configuring Event Publishers

You must configure the SAE to publish the global service-tracking events to the SRC 3GPP gateway. Any other events are ignored.

For information about configuring event publishers, see *Special Types of Event Publishers*.

Related Documentation

- *Configuring the SAE for External Plug-Ins (SRC CLI)*
- *Special Types of Event Publishers*

Example: Configuring the SRC 3GPP Gateway

This example describes how to configure the SRC 3GPP gateway.

Requirements

This example uses the following hardware and software components:

- One or more C Series Controllers running the Juniper Networks Session and Resource Control (SRC) software
- One or more MX Series 3D Universal Edge Routers or E Series Broadband Services Routers
- One PCRF
- Junos OS Release 11.4 or later

No special configuration beyond device initialization is required before you can configure this feature.

Before you configure and apply the configurations in this example, be sure you have an understanding of the following:

- SRC 3GPP gateway
- Diameter
- SAE

Overview

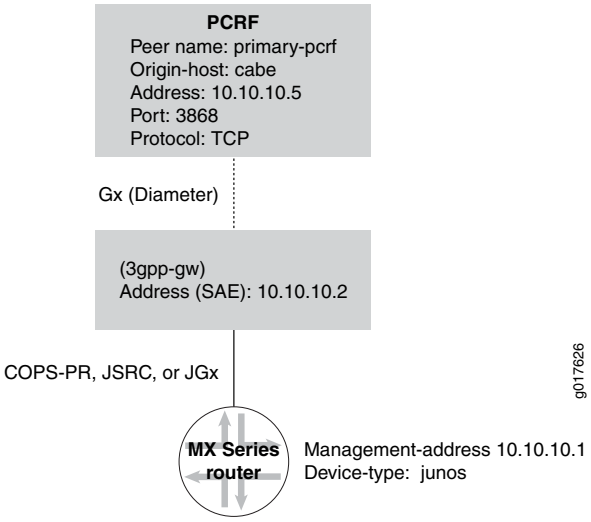
This example does not show all possible configuration choices.

Topology

This example shows how to configure the SRC software for the SRC 3GPP gateway component. The SRC 3GPP gateway is connected to a single PCRF over the Gx interface running the Diameter application. The C Series Controller is also connected to an MX Series router called mx1. In this example, the connection to the mx1 router is using the Junos OS device driver.

This example does not show all possible configuration choices.

Figure 12: SRC 3GPP Gateway Example Topology



Configuration

To configure the SRC 3GPP gateway, perform these tasks:

- Configure the Local Properties for the SRC 3GPP Gateway on page 170
- Configure the Shared Properties for the SRC 3GPP Gateway on page 173
- Adding the PCRF as a Diameter Peer on page 174
- Add the MX Series Router as a Shared Network Device. on page 174
- Configuring the SRC 3GPP Gateway as an External Plug-In for the SAE on page 175

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set slot 0 gw-3gpp initial directory-connection credentials testing1
set slot 0 gw-3gpp initial directory-connection principal cn=conf,o=operators,o=umc
set slot 0 gw-3gpp initial directory-connection url ldap://127.0.0.1:389
set slot 0 gw-3gpp initial directory-eventing eventing
set slot 0 gw-3gpp gx address 10.10.10.2
set slot 0 gw-3gpp gx diameter-peer primary-pcrf
set slot 0 gw-3gpp gx origin-host duke
set slot 0 gw-3gpp gx origin-realm example.com
set slot 0 gw-3gpp gx port 3868
set slot 0 gw-3gpp gx protocol tcp
set slot 0 gw-3gpp logger debug file filename var/log/3gpp-gx-debug.log filter /debug
set slot 0 gw-3gpp logger debug file maximum-file-size 1000000
set slot 0 gw-3gpp logger error file filename var/log/3gpp-gx-error.log filter /error
set slot 0 gw-3gpp logger error file maximum-file-size 2000000
set slot 0 gw-3gpp logger error file rollover-filename var/log/3gpp-gx-error.alt
set slot 0 gw-3gpp logger info file filename var/log/3gpp-gx-info.log /info
set slot 0 gw-3gpp logger info file maximum-file-size 2000000
set slot 0 gw-3gpp logger info file rollover-filename var/log/3gpp-gx-info.alt
set slot 0 gw-3gpp shared /3gpp-gw-share
```

```
set slot 0 nic hostname DemoHost scenario-name OnePopSessionHandle snmp-agent
  initial directory-connection credentials testing1 principal
  cn=nic,ou=Components,o=Operators url ldap://127.0.0.1:389/
set slot 0 nic hostname DemoHost scenario-name OnePopSessionHandle snmp-agent
  initial directory-eventing eventing
set shared gw-3gpp group 3gpp-gw-share
set shared gw-3gpp group 3gpp-gw-share configuration nic-proxy-configuration
  sessionHdl resolution key-type SessionHandle resolver-name /realms/sessionHandle/A1
  value-type SessionHandle
set shared gw-3gpp group 3gpp-gw-share configuration subscriber-types session-handle
  subscriber-id-type session-handle
set shared network diameter peer primary-pcrf address [ 10.10.10.5 ]
set shared network diameter peer primary-pcrf origin-host cabe
set shared network diameter peer primary-pcrf port 3868
set shared network diameter peer primary-pcrf protocol tcp
set shared network device mx1 device-type junos
set shared network device mx1 management-address 10.10.10.1
set shared network device mx1 virtual-router * sae-connection 10.10.10.2
set shared sae configuration driver junos beep-server-port 3333 tls-beep-server-port
  3434 sdx-group-name sdx sdx-session-group-name sdx-sessions
set shared sae configuration plug-ins name 3gppgw-test external
set corba-object-reference file:///opt/UMC/3gpp/var/run/3gpp-gw.ior
set attributes service-session-name
commit
Exit
enable component gw-3gpp
```

Configure the Local Properties for the SRC 3GPP Gateway

Step-by-Step Procedure

Local properties for the SRC 3GPP gateway include the Gx interface, the connection to the Juniper Networks database directory, the directory-monitoring properties, logging, specifying which group configuration to use for the SRC 3GPP gateway, and configuring the NIC operating properties.

1. Configure the directory-connection properties for the SRC 3GPP gateway.

   ```
   [edit slot 0 gw-3gpp initial directory-connection]
   user@host# set credentials credentials
   user@host# set principal principal
   user@host# set url ldap://127.0.0.1:389
   ```

2. Configure directory-eventing.

   ```
   [edit slot 0 gw-3gpp initial directory-eventing]
   user@host# set eventing
   ```

3. Configure the SRC 3GPP gateway Gx interface:

   ```
   [edit slot 0 gw-3gpp gx]
   user@host# set address 10.10.10.2
   user@host# set diameter-peer primary-pcrf
   user@host# set origin-host duke
   user@host# set origin-realm example.com
   user@host# set port 3868
   user@host# set protocol tcp
   ```

4. Configure logging.

```
[edit slot 0 gw-3gpp logger debug]
user@host# set file filename var/log/3gpp-gx-debug.log
user@host# set filter /debug-
user@host# set maximum-file-size 1000000
user@host# up
user@host# set logger error file filename var/log/3gpp-gx-error.log
user@host# set filter /error-
user@host# set maximum-file-size 2000000
user@host# set rollover-filename var/log/3gpp-gx-error.alt
user@host# up
user@host# set logger info file filename var/log/3gpp-gx-info.log
user@host# set filter /info-
user@host# set maximum-file-size 2000000
user@host# set rollover-filename var/log/3gpp-gx-info.alt
user@host# up
```

5. Specify the shared group for the local configuration.

```
[edit slot 0 gw-3gpp]
user@host# set shared /3gpp-gw-share
user@host# commit
```

6. Configure the NIC operating properties:

```
[edit slot 0 nic]
user@host# set hostname DemoHost
user@host# set scenario-name OnePopSessionHandle
user@host# set snmp-agent
user@host# edit initial directory-connection
[edit slot 0 nic initial directory-connection]
user@host# set credentials testing1
user@host# set principal cn=nic,ou=Components,o=Operators
user@host# set url ldap://127.0.0.1:389/
user@host# up
user@host# edit initial directory-eventing
[edit slot 0 nic initial directory-eventing]
user@host# set eventing
user@host# up
user@host# up
user@host# commit
```

Results Confirm the local configuration properties. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit slot 0 gw-3gpp]
user@duke# show
gx {
  address 10.10.10.2;
  diameter-peer primary-pcrf;
  origin-host duke;
  origin-realm example.com;
  port 3868;
  protocol tcp;
}
```

```
initial {
  directory-connection {
    backup-urls '';
    credentials ********;
    principal cn=conf,o=operators,o=umc;
    url ldap://127.0.0.1:389/;
  }
  directory-eventing {
    eventing;
    polling-interval 30;
  }
}
logger debug {
  file {
    filename var/log/3gpp-gx-debug.log;
    filter /debug-;
    maximum-file-size 1000000;
  }
}
logger error {
  file {
    filename var/log/3gpp-gx-error.log;
    filter /error-;
    maximum-file-size 2000000;
    rollover-filename var/log/3gpp-gx-error.alt;
  }
}
logger info {
  file {
    filename var/log/3gpp-gx-info.log;
    filter /info-;
    maximum-file-size 2000000;
    rollover-filename var/log/3gpp-gx-info.alt;
  }
}
shared /3gpp-gw-share;

[edit slot 0 gw-3gpp]
user@duke#

[edit slot 0 nic]
user@duke# show
base-dn o=umc;
hostname DemoHost;
initial {
  directory-connection {
    backup-urls '';
    credentials ********;
    principal cn=nic,ou=Components,o=Operators,<base>;
url ldap://10.10.10.10:389/;
  }
  directory-eventing {
    eventing;
    polling-interval 15;
  }
}
scenario-name OnePopSessionHandle;
snmp-agent;

[edit slot 0 nic]
user@duke#
```

Configure the Shared Properties for the SRC 3GPP Gateway

Step-by-Step Procedure

Configure the shared properties for the SRC 3GPP gateway including the shared group configuration, NIC proxy configuration, and subscriber type.

1. Specify the group to configure:

```
[edit shared gw-3gpp]
user@host# edit group 3gpp-gw-share
```

2. Configure the NIC proxy configuration:

```
[edit shared gw-3gpp group 3gpp-gw-share configuration]
user@host# edit nic-proxy-configuration sessionHdl resolution
[edit shared gw-3gpp group 3gpp-gw-share configuration nic-proxy-configuration
  sessionHdl resolution]
user@host# set key-type SessionHandle
user@host# set resolver-name /realms/sessionHandle/A1
user@host# set value-type SessionHandle
```

3. Configure the subscriber type:

```
[edit shared gw-3gpp]
user@host# edit configuration subscriber-types session-handle
[edit shared gw-3gpp group 3gpp-gw-share configuration subscriber-types
  session-handle]
user@host# set subscriber-id-type session-handle
```

Results

Confirm the shared configuration properties. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit shared gw-3gpp group 3gpp-gw-share]
user@duke show
configuration {
  nic-proxy-configuration {
    sessionHdl {
      resolution {
        key-type SessionHandle;
        resolver-name /realms/sessionHandle/A1;
        value-type SaeId;
      }
    }
  }
  subscriber-types {
    session-handle {
      nic-proxy sessionHdl;
      subscriber-id-type session-handle;
    }
  }
}

[edit shared gw-3gpp group 3gpp-gw-share]
user@duke#
```

Adding the PCRF as a Diameter Peer

Step-by-Step Procedure

You must specify the PCRF as a Diameter peer. This example uses a single PCRF, which connects to the C Series Controller over the SRC 3GPP gateway Gx interface using the Diameter application.

1. To add the PCRF as a Diameter peer:

```
[edit shared network diameter peer]
user@host# edit primary-pcrf
[edit shared network diameter peer primary-pcrf]
user@host# set address [ 10.10.10.5 ]
user@host# set origin-host cabe
user@host# set port 3868
user@host# set protocol tcp
```

Results

From configuration mode, confirm the configuration for the PCRFs. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit shared network diameter peer primary-pcrf]
user@duke# show
address [ 10.10.10.5 ];
connect-timeout 10;
origin-host cabe;
port 3868;
protocol tcp;

[edit shared network diameter peer primary-pcrf]
user@duke#
```

Add the MX Series Router as a Shared Network Device.

Step-by-Step Procedure

Add the MX Series router as a shared network device using the Junos OS device driver.

1. Add the MX Series router as a shared network device.

```
[edit shared network device mx1]
user@host# set management-address 10.10.10.1
user@host# set device-type junos
user@host# set virtual-router * sae-connection 10.10.10.2
user@host# commit
```

2. Configure the SAE to manage the MX Series router.

```
[edit shared network device mx1]
user@host# top
user@host# edit shared sae configuration driver junos
user@host# set beep-server-port 3333
user@host# set tls-beep-server-port 3434
user@host# set sdx-group-name sdx
user@host# set sdx-session-group-name sdx-sessions
user@host# commit
```

Results

```
[edit shared network device mx1]
user@duke# show
device-type junos;
management-address 10.10.10.1;
virtual-router * {
  sae-connection 10.10.10.2;
}

[edit shared network device mx1]
user@duke#

[edit shared sae configuration driver junos]
user@host# show
beep-server-port 3333;
tls-beep-server-port 3434;
connection-attempts 50;
keepalive-interval 45;
message-timeout 30000;
batch-size 10;
transaction-batch-time 2000;
sdx-group-name sdx;
sdx-session-group-name sdx-sessions;
send-commit-check true;

[edit shared sae configuration driver junos]
user@duke#
```

Configuring the SRC 3GPP Gateway as an External Plug-In for the SAE

Step-by-Step Procedure

1. Configure the SRC 3GPP gateway as an external plug-in for the SAE.

   ```
   user@host# edit shared sae configuration plug-ins name 3gppgw-test external
   user@host# set corba-object-reference file:///opt/UMC/3gpp/var/run/3gpp-gw.ior
   user@host# set attributes service-session-name
   user@host# commit
   ```

Results

```
[edit shared sae configuration plug-ins name 3gppgw-test external]
user@duke# show
attributes service-session-name;
corba-object-reference file:///opt/UMC/3gpp/var/run/3gpp-gw.ior;

[edit shared sae configuration plug-ins name 3gppgw-test external]
user@duke#
```

Related Documentation

- Configuration Statements for the SRC 3GPP Gateway on page 145
- Configuring the SRC 3GPP Gateway (SRC CLI) on page 147
- SRC 3GPP Gateway Overview on page 131

Managing the SRC 3GPP Gateway (SRC CLI)

After you have configured the SRC 3GPP gateway, you can perform these tasks:

- Starting the SRC 3GPP Gateway (SRC CLI) on page 176
- Restarting the SRC 3GPP Gateway (SRC CLI) on page 176
- Stopping the SRC 3GPP Gateway (SRC CLI) on page 176

Starting the SRC 3GPP Gateway (SRC CLI)

To start the SRC 3GPP gateway:

```
user@host> enable component gw-3gpp
```

The system responds with a start message. If the SRC 3GPP gateway is already running, the system responds with a warning message.

Restarting the SRC 3GPP Gateway (SRC CLI)

You must restart the SRC 3GPP gateway after you commit a configuration change.

To restart the SRC 3GPP gateway:

```
user@host> restart component gw-3gpp
```

The system responds with a start message. If the SRC 3GPP gateway is already running, the system responds with a shutdown message and then a start message.

Stopping the SRC 3GPP Gateway (SRC CLI)

To stop the SRC 3GPP gateway:

```
user@host> disable component gw-3gpp
```

The system responds with a shutdown message. If SRC 3GPP gateway is not running when you issue the command, the system responds with the command prompt.

PART 4

SRC 3GPP Northbound Gy Interface

CHAPTER 15

SRC 3GPP Gy Overview

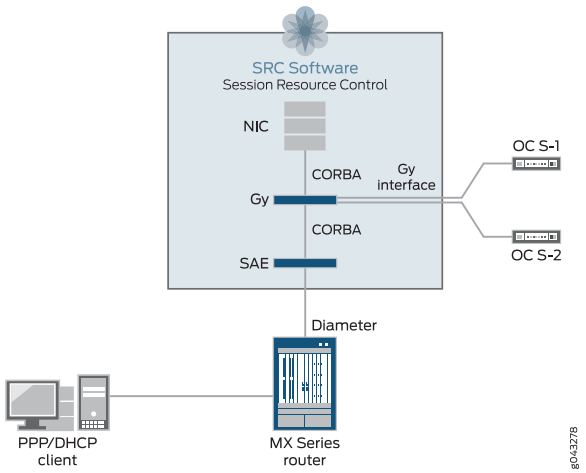
SRC 3GPP Gy Overview

The SRC 3GPP Gy feature is a Diameter-based component in the SRC software, which provides integration with the Online Charging System (OCS) to support fixed-mobile convergence (FMC).

SRC's Gy uses the northbound Gy interface to handle charging-related information between the OCS and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. Figure 13 on page 179 shows an SRC 3GPP Gy deployment scenario. The northbound Gy interface on SRC's Gy communicates with the OCS using the Diameter protocol.

NOTE: SRC's Gy supports volume-based quota only.

Figure 13: SRC 3GPP Gy Deployment Scenario



The following sections describe the sequences for each of SRC's Gy supported scenarios based on an MX Series router:

Subscriber Login Sequence

During subscriber login, the SAE notifies a service authorization event to SRC's Gy for each service to be activated. This event includes information such as login name, service name, session handle, service identifier, and other SAE plug-in attributes.

SRC's Gy performs the following actions for service activation received through activate-on-login (AOL) or SRC 3GPP gateway:

1. Sends a Diameter Credit-Control-Request initial (CCR-I) message with the following information to the OCS:
   - CC-Request-Type AVP: INITIAL_REQUEST
   - Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
   - Service-Identifier AVP: Identifier of the service. This AVP is available in the message only if the identifier is configured for the service.

     NOTE: **Gy functionalities require service identifier information, so you must ensure that the identifier is configured for the service.**
   - Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
     - Requested-Service-Unit AVP: Grouped AVP that contains:
       - CC-Input-Octets AVP: Value is always set to zero
       - CC-Output-Octets AVP: Value is always set to zero
       - CC-Total-Octets AVP: Value is always set to zero
2. Receives the Credit-Control-Answer initial (CCA-I) message from the OCS.
   - If the OCS rejected the CCR-I message by providing an error code in the CCA-I message, SRC's Gy requests the SAE not to activate the service for the subscriber.
   - If the OCS accepted the CCR-I message, SRC's Gy requests the SAE to activate the service for the subscriber and saves the received quota information in the service session properties. SAE interacts with the router to activate the service for the subscriber.

   The CCA-I message contains the following information:

- CC-Request-Type AVP: INITIAL_REQUEST
- Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
- Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
  - Granted-Service-Unit AVP: Grouped AVP that contains:
    - CC-Total-Octets AVP: The number of octets that can be received and sent by the router
    - CC-Input-Octets AVP: The number of octets that can be received by the router from the subscriber
    - CC-Output-Octets AVP: The number of octets that can be sent from the router to the subscriber
    - Final-Unit-Indication AVP: Indicates that the Granted-Service-Unit AVP has final units for the service
    - Volume-Quota-Threshold AVP: Threshold value in octets

Subscriber Logout Sequence

During subscriber logout, the SAE notifies a service stop event to SRC's Gy for each active service. The event contains accounting statistics received from the router.

For each service stop event, SRC's Gy issues a CCR termination (CCR-T) message to the OCS that includes:

- CC-Request-Type AVP: TERMINATION_REQUEST
- Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
- Service-Identifier AVP: Identifier of the service. This AVP is available in the message only if the identifier is configured for the service.

NOTE: **Gy functionalities require service identifier information, so you must ensure that the identifier is configured for the service.**

- Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
  - Used-Service-Unit AVP: Grouped AVP that contains:
    - CC-Input-Octets AVP: The number of octets received by the router from the subscriber
    - CC-Output-Octets AVP: The number of octets sent from the router to the subscriber

- CC-Total-Octets AVP: The number of octets received and sent by the router
- Terminate-Cause AVP: Reason for terminating the service. The value is set to USER_REQUEST.

Service Activation Sequence

When the Application Function (AF), Dynamic Service Activator (DSA), AOL, scheduler, or Volume Tracking Application (VTA) initiates a service activation, the SAE triggers a service authorization event to SRC's Gy. The event includes information such as login name, service name, session handle, service identifier, and other SAE plug-in attributes. SRC's Gy performs the following actions:

1. Sends a Diameter CCR-I message with the following information to the OCS:
   - CC-Request-Type AVP: INITIAL_REQUEST
   - Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
   - Service-Identifier AVP: Identifier of the service. This AVP is available in the message only if the identifier is configured for the service.

     NOTE: **Gy functionalities require service identifier information, so you must ensure that the identifier is configured for the service.**

   - Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
     - Requested-Service-Unit AVP: Grouped AVP that contains:
       - CC-Input-Octets AVP: Value is always set to zero
       - CC-Output-Octets AVP: Value is always set to zero
       - CC-Total-Octets AVP: Value is always set to zero

2. Receives the CCA message from the OCS.
   - If the OCS rejected the CCR-I message by providing an error code in the CCA message, SRC's Gy requests the SAE not to activate the service for the subscriber.
   - If the OCS accepted the CCR-I message, SRC's Gy requests the SAE to activate the service for the subscriber and saves the received quota information in the service session properties. SAE interacts with the router to activate the service for the subscriber.

   The CCA-I message contains the following information:
   - CC-Request-Type AVP: INITIAL_REQUEST
   - Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
   - Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
     - Granted-Service-Unit AVP: Grouped AVP that contains:

- CC-Total-Octets AVP: The number of octets that can be received and sent by the router
- CC-Input-Octets AVP: The number of octets that can be received by the router from the subscriber
- CC-Output-Octets AVP: The number of octets that can be sent from the router to the subscriber
- Final-Unit-Indication AVP: Indicates that the Granted-Service-Unit AVP has final units for the service
- Volume-Quota-Threshold AVP: Threshold value in octets

Service Deactivation Sequence

When the AF, DSA, AOL, scheduler, or VTA initiates a service deactivation, the SAE triggers a service stop event to SRC's Gy. The event contains accounting statistics received from the router.

SRC's Gy issues a CCR-T message to the OCS that includes:

- CC-Request-Type AVP: TERMINATION_REQUEST
- Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
- Service-Identifier AVP: Identifier of the service. This AVP is available in the message only if the identifier is configured for the service.

NOTE: **Gy functionalities require service identifier information, so you must ensure that the identifier is configured for the service.**

- Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
  - Used-Service-Unit AVP: Grouped AVP that contains:
    - CC-Input-Octets AVP: The number of octets received by the router from the subscriber
    - CC-Output-Octets AVP: The number of octets sent from the router to the subscriber
    - CC-Total-Octets AVP: The number of octets received and sent by the router
    - Terminate-Cause AVP: Reason for terminating the service. The value is set to USER_REQUEST.

Service Accounting Sequence

After the service is activated by the SAE, the following sequences occur:

1. The SAE updates SRC's Gy with the used service units received from the router through accounting messages.
2. SRC's Gy tracks the quota or threshold usage with the information received from the OCS.
3. If the quota or threshold is exhausted, SRC's Gy sends a CCR update (CCR-U) message to the OCS with the following information to request for a new quota:
   - CC-Request-Type AVP: UPDATE_REQUEST
   - Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
   - Service-Identifier AVP: Identifier of the service. This AVP is available in the message only if the identifier is configured for the service.

     NOTE: **Gy functionalities require service identifier information, so you must ensure that the identifier is configured for the service.**

   - Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
     - Requested-Service-Unit AVP: Grouped AVP that contains:
       - CC-Input-Octets AVP: Value is always set to zero
       - CC-Output-Octets AVP: Value is always set to zero
       - CC-Total-Octets AVP: Value is always set to zero
     - Used-Service-Unit AVP: Grouped AVP that contains:
       - CC-Input-Octets AVP: The number of octets received by the router from the subscriber
       - CC-Output-Octets AVP: The number of octets sent from the router to the subscriber
       - CC-Total-Octets AVP: The number of octets received and sent by the router
       - Reporting-Reason AVP: Reason for usage reporting. The value is set to QUOTA_EXHAUSTED, if the granted quota in the Granted-Service-Unit AVP is exhausted. The value is set to THRESHOLD, if the granted quota in Volume-Quota-Threshold AVP is exhausted.
4. The OCS sends the CCA update (CCA-U) message to SRC's Gy with the following information:
   - CC-Request-Type AVP: UPDATE_REQUEST
   - Subscription-ID AVP: Login name received from the router as an SAE plug-in attribute
   - Multiple-Service-Credit-Control AVP: Grouped AVP that contains:
     - Granted-Service-Unit AVP: Grouped AVP that contains:

- CC-Total-Octets AVP: The number of octets that can be received and sent by the router
- CC-Input-Octets AVP: The number of octets that can be received by the router from the subscriber
- CC-Output-Octets AVP: The number of octets that can be sent from the router to the subscriber
- Final-Unit-Indication AVP: Indicates that the Granted-Service-Unit AVP has final units for the service
- Volume-Quota-Threshold AVP: Threshold value in octets

5. SRC's Gy stores the quota in the service session properties.
6. The SAE updates SRC's Gy with the used service units received from the router through accounting messages.
7. SRC's Gy checks whether the Final-Unit-Indication AVP is stored in the service session properties. If the Final-Unit-Indication AVP is available, SRC's Gy requests SAE to deactivate the service. For more information about the communication between SRC's Gy and OCS during service deactivation, see "Service Deactivation Sequence" on page 183.
8. If the Final-Unit-Indication AVP is not available, SRC's Gy iterates sending of CCR-U messages to the OCS to request for a new quota.

Related Documentation

- SRC 3GPP Gy Peer Communication and Redundancy on page 185
- Configuring the SRC 3GPP Gy (SRC CLI) on page 189

SRC 3GPP Gy Peer Communication and Redundancy

The SRC 3GPP Gy configuration enables the definition of multiple Diameter peers to act as OCSes. SRC's Gy selects one peer to become the active peer, and forwards all CCR messages to it. If the SRC software detects a failure of the active peer, SRC's Gy selects another peer to become the active peer. SRC's Gy does not switch back to the original active peer when it comes back online.

For redundancy purposes, you can configure multiple SRC 3GPP Gys to handle the communication between the same group of SAEs and OCSes. To achieve this redundancy, you configure multiple SRC's Gy instances to point to the same namespace in the shared configuration.

All SRC's Gy instances pointing to the same namespace use a community manager to elect the active member. The active member registers itself with the naming server and is the only member receiving subscriber-tracking events from the SAE and forwarding them to the OCS.

If the active SRC's Gy member fails, the community manager detects the failure and elects another member to become the active member. The new active member registers itself with the naming server thus overwriting the old member's endpoints. During the

failure period, the SAE keeps a fail queue for the SRC 3GPP plug-in and replays the queue to the new active member.

Related Documentation

- SRC 3GPP Gy Overview on page 179
- Configuring the SRC 3GPP Gy (SRC CLI) on page 189

CHAPTER 16

Configuring and Managing the SRC 3GPP Gy (SRC CLI)

Configuration Statements for the SRC 3GPP Gy

Use the following configuration statements to configure SRC's Gy at the **[edit]** hierarchy level.

```
slot number gy-3gpp {
   shared shared;
}
```

```
slot number gy-3gpp gy {
  destination-host destination-host;
  destination-realm destination-realm;
  diameter-peer [diameter-peer.....];
  protocol (tcp | sctp);
  port port;
  address address;
  origin-host origin-host;
  origin-realm origin-realm;
}
slot number gy-3gpp initial {
  static-dn static-dn;
  dynamic-dn dynamic-dn;
}
slot number gy-3gpp initial directory-connection {
  url url;
  backup-urls [backup-urls...];
  principal principal;
  credentials credentials;
  protocol (ldaps);
  timeout timeout;
  check-interval check-interval;
  blacklist;
  snmp-agent;
}
slot number gy-3gpp initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
slot number gy-3gpp java-orb object-adapter {
  address address;
}
slot number gy-3gpp logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}
slot number gy-3gpp logger name syslog {
  facility facility
  filter filter;
  format format;
  host host;
  port port;
}
shared gy-3gpp group name
shared gy-3gpp configuration subscriber-types (session-handle) {
  subscriber-id-type (session-handle);
  nic-proxy nic-proxy;
}
shared gy-3gpp configuration nic-proxy-configuration name {
}
shared gy-3gpp configuration nic-proxy-configuration name resolution {
```

```
    resolver-name resolver-name;
    key-type key-type;
    value-type value-type;
    expect-multiple-values;
    constraints constraints;
  }
  shared gy-3gpp configuration nic-proxy-configuration name cache {
    cache-size cache-size;
    cache-cleanup-interval cache-cleanup-interval;
    cache-entry-age cache-entry-age;
  }
  shared gy-3gpp configuration nic-proxy-configuration name nic-host-selection {
    groups [groups...];
    selection-criteria (roundRobin | randomPick | priorityList);
  }
  shared gy-3gpp configuration nic-proxy-configuration name nic-host-selection blacklisting
    {
    try-next-system-on-error;
    number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
    blacklist-retry-interval blacklist-retry-interval;
  }
  shared gy-3gpp configuration nic-proxy-configuration name test-nic-bindings {
    use-test-bindings;
  }
  shared gy-3gpp configuration nic-proxy-configuration name test-nic-bindings key-values
    name {
    value;
  }
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- SRC 3GPP Gy Overview on page 179

Configuring the SRC 3GPP Gy (SRC CLI)

To configure SRC's Gy:

1. Configure initial properties, including the connection to the directory and directory-monitoring properties.

   See "Configuring Initial Properties for the SRC 3GPP Gy (SRC CLI)" on page 190.

   See "Changing the Location of Data in the Directory" on page 149.

   See "Configuring Directory-Connection Properties for the SRC 3GPP Gy (SRC CLI)" on page 192.

   See "Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gy (SRC CLI)" on page 193.

2. Configure the connection to the OCS (remote Diameter peer).

   See "Configuring Diameter Peers (SRC CLI)" on page 152.

3. Configure SRC's Gy interface.

   See "Configuring the SRC 3GPP Gy Interface (SRC CLI)" on page 197.

4. Configure the basic local properties for SRC's Gy.

   See "Configuring Basic Local Properties for the SRC 3GPP Gy (SRC CLI)" on page 198.

5. Configure the object adapter Internet address.

   See "Configuring the Object Adapter Internet Address for the SRC 3GPP Gy (SRC CLI)" on page 200.

6. Configure logging destinations.

   See "Configuring Logging Destinations for SRC 3GPP Gy to Store Messages in a File (SRC CLI)" on page 200.

   See "Configuring Logging Destinations for SRC 3GPP Gy to Send Messages to the System Logging Facility (SRC CLI)" on page 201.

7. Create an SRC's Gy grouped configuration.

   See "Creating Grouped Configurations for the SRC 3GPP Gy (SRC CLI)" on page 202.

8. Configure subscriber types.

   See "Configuring the Subscriber Type for SRC 3GPP Gy (SRC CLI)" on page 203.

9. Configure the NIC proxies.

   See "Configuring a NIC Proxy for the SRC 3GPP Gy (SRC CLI)" on page 204.

10. Start SRC's Gy.

    See "Starting the SRC 3GPP Gy (SRC CLI)" on page 211.

11. Configure the SAE for SRC's Gy.

    See "Configuring the SAE for the SRC 3GPP Gy" on page 209.

Related Documentation

- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179
- SRC 3GPP Gy Peer Communication and Redundancy on page 185

Configuring Initial Properties for the SRC 3GPP Gy (SRC CLI)

Use the following configuration statements to configure initial properties for SRC's Gy:

```
slot number gy-3gpp initial {
   static-dn static-dn;
   dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

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

   user@host# edit slot 0 gy-3gpp initial

2. Specify the properties for SRC's Gy.

   [edit slot 0 gy-3gpp initial]

```
user@host# set ?
```

For more information about configuring local properties for SRC components, see "Changing the Location of Data in the Directory" on page 149.

3. (Optional) Verify your configuration.

```
[edit slot 0 gy-3gpp initial]
user@host# show
directory-connection {
  backup-urls '';
  check-interval 60;
  credentials ********;
  principal cn=conf,o=Operators,o=UMC;
  timeout 10;
  url ldap://127.0.0.1:389/;
}
directory-eventing {
  dispatcher-pool-size 1;
  event-base-dn o=umc;
  eventing;
  polling-interval 30;
  signature-dn o=umc;
}
dynamic-dn ou=dynamicConfiguration,ou=Configuration,o=Management,o=umc;
static-dn 'l=3GPP-Gy,
ou=staticConfiguration,ou=Configuration,o=Management,o=umc';
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring Directory-Connection Properties for the SRC 3GPP Gy (SRC CLI) on page 192
- Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gy (SRC CLI) on page 193
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Changing the Location of Data in the Directory

In most cases, you use the default configuration for the location of SRC data in the directory:

- Administrator-defined configuration data—*ou=staticConfiguration,ou=Configuration,o=Management,o=umc*
- Programmatically defined configuration data—*ou=dynamicConfiguration,ou=Configuration,o=Management, o=umc*

You can specify the full distinguished name (DN), or a DN relative to a base DN, identified as <*base*>.

You can change the location of data in the directory at the Expert CLI editing level.

Use the following configuration statements to change the location of data for a component in the directory:

```
slot number component-name initial {
  static-dn static-dn ;
  dynamic-dn dynamic-dn ;
}
```

To change the location of data in the directory:

1. From configuration mode, access the configuration statement that specifies the configuration for a component on a slot.

   ```
   [edit]
   user@host# edit slot number nic initial
   ```

   For example:

   ```
   [edit]
   user@host# edit slot 0 nic initial
   ```

2. (Optional) Change the location of administrator-defined configuration data in the directory.

   ```
   [edit slot 0 nic initial]
   user@host# set static-dn static-dn
   ```

3. (Optional) Change the location of programmatically defined configuration data in the directory.

   ```
   [edit slot 0 nic initial]
   user@host# set dynamic-dn dynamic-dn
   ```

Related Documentation

- *Configuring Initial Directory Eventing Properties for SRC Components*
- *Configuring Basic Local Properties*
- *Configuration Statements for Local Configuration*
- *Managing Directory Communication*

Configuring Directory-Connection Properties for the SRC 3GPP Gy (SRC CLI)

Use the following configuration statements to configure directory-connection properties for SRC's Gy:

```
slot number gy-3gpp initial directory-connection {
  url url;
  backup-urls [backup-urls...];
  principal principal;
  credentials credentials;
  protocol (ldaps);
  timeout timeout;
  check-interval check-interval;
  blacklist;
  snmp-agent;
}
```

To configure directory-connection properties:

1. From configuration mode, access the statement that configures the directory-connection properties.

   user@host# edit slot 0 gy-3gpp initial directory-connection

2. Specify the properties for SRC's Gy.

   [edit slot 0 gy-3gpp initial directory-connection]
   user@host# set ?

   For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 gy-3gpp initial directory-connection]
user@host# show
backup-urls '';
check-interval 60;
credentials ********;
principal cn=conf,o=Operators,o=UMC;
timeout 10;
url ldap://127.0.0.1:389/;
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring Initial Properties for the SRC 3GPP Gy (SRC CLI) on page 190
- Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gy (SRC CLI) on page 193
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring Initial Directory-Eventing Properties for the SRC 3GPP Gy (SRC CLI)

Use the following configuration statements to configure initial directory-eventing properties for SRC's Gy:

```
slot number gy-3gpp initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory-eventing properties:

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

   user@host# edit slot 0 gy-3gpp initial eventing

2. Specify the initial directory-eventing properties for SRC's Gy.

   [edit slot 0 gy-3gpp initial directory-eventing]

```
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 gy-3gpp initial directory-eventing]
user@host# show
dispatcher-pool-size 1;
event-base-dn o=umc;
eventing;
polling-interval 30;
signature-dn o=umc;
}
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring Initial Properties for the SRC 3GPP Gy (SRC CLI) on page 190
- Configuring Directory-Connection Properties for the SRC 3GPP Gy (SRC CLI) on page 192
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring Diameter Peers (SRC CLI)

Use the following configuration statements to configure the Diameter peers:

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

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

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

To configure the Diameter peer:

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

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

   The peer name must be unique.

2. Specify the protocol for the transport connection.

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

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

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

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

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

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

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

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

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

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

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

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

   [edit shared network diameter peer *name*]

```
user@host# set state-machine-timeout state-machine-timeout
```

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

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

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

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

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

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

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

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

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

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

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

Related Documentation

- *Configuring the Diameter Application (SRC CLI)*
- *Viewing SRC Diameter Server State (SRC CLI)*

Configuring the SRC 3GPP Gy Interface (SRC CLI)

Use the following configuration statements to configure SRC's Gy interface:

```
slot number gy-3gpp gy {
  destination-host destination-host;
  destination-realm destination-realm;
  diameter-peer [diameter-peer.....];
  protocol (tcp | sctp);
  port port;
  address address;
  origin-host origin-host;
  origin-realm origin-realm;
}
```

NOTE: SRC's Gy uses its own Diameter stack, which is configured under the **[edit slot 0 gy-3gpp gy]** hierarchy. It does not use the Diameter stack configured under the **[edit system diameter]** hierarchy; this Diameter stack is used for SAE and router communication.

To configure SRC's Gy interface:

1. From configuration mode, access the statement that configures SRC's Gy interface.

   user@host# edit slot 0 gy-3gpp gy

2. (Optional) Specify the Diameter identifier for the remote endpoint, which is the destination of the Diameter message. The Destination-Host AVP (AVP Code 293) is of the DiameterIdentity type and is present in all Diameter messages.

   [edit slot 0 gy-3gpp gy]
   user@host# set destination-host *destination-host*

3. Specify the Diameter identifier for the realm of the remote endpoint, which is the destination of the Diameter message. The Destination-Realm AVP (AVP Code 283) is of the DiameterIdentity type and is present in all Diameter messages.

   [edit slot 0 gy-3gpp gy]
   user@host# set destination-realm *destination-realm*

4. Specify the list of remote Diameter peers (OCSes) that connect to SRC's Gy over the Gy interface.

   [edit slot 0 gy-3gpp gy]
   user@host# set diameter-peer *[diameter-peer....]*

   Each Diameter peer you specify must be previously configured under the **[edit shared network diameter peer]** hierarchy.

5. (Optional) Specify the protocol for the transport connection.

   [edit slot 0 gy-3gpp gy]
   user@host# set protocol [(tcp | sctp)...]

6. (Optional) Specify the port to use for incoming connections.

```
[edit slot 0 gy-3gpp gy]
user@host# set port port
```

7. Specify the local address of the peer.

```
[edit slot 0 gy-3gpp gy]
user@host# set address address
```

8. Specify the Diameter identifier for the local endpoint that is the originator of the Diameter message.

```
[edit slot 0 gy-3gpp gy]
user@host# set origin-host origin-host
```

9. Specify the Diameter identifier for the realm of the local endpoint that is the originator of the Diameter message.

```
[edit slot 0 gy-3gpp gy]
user@host# set origin-realm origin-realm
```

10. (Optional) Verify your configuration.

```
[edit slot 0 gy-3gpp gy]
user@host# show
address 10.212.10.10;
destination-realm englab.juniper.net;
diameter-peer bng-kelpie;
origin-host c3bng-src10.englab.juniper.net;
origin-realm englab.juniper.net;
port 53868;
protocol tcp;
}
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring Basic Local Properties for the SRC 3GPP Gy (SRC CLI)

The local configuration for SRC's Gy component defines the directory where the shared configuration is and points to the component namespace.

Use the following configuration statements to configure basic local properties for SRC's Gy:

```
slot number gy-3gpp {
  shared shared;
}
```

To configure basic local properties:

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

```
[edit]
user@host# edit slot 0 gy-3gpp
```

2. Specify the configuration namespace for SRC's Gy as the path, relative to the root of the static configuration properties, that defines the object for the namespace.

   ```
   [edit slot 0 gy-3gpp]
   user@host# set shared shared
   ```

   For example:

   ```
   [edit slot 0 gy-3gpp]
   user@host# set shared /3gpp-gy-share
   ```

   NOTE: All SRC's Gy instances pointing to the same shared namespace run in the same redundant community.

3. (Optional) Verify your configuration.

```
[edit slot 0 gy-3gpp]
user@host# show
gy {
  address 10.212.10.10;
  destination-realm englab.juniper.net;
  diameter-peer bng-kelpie;
  origin-host c3bng-src10.englab.juniper.net;
  origin-realm englab.juniper.net;
  port 53868;
  protocol tcp;
}
initial {
  directory-connection {
    backup-urls '';
    check-interval 60;
    credentials ********;
    principal cn=conf,o=Operators,o=UMC;
    timeout 10;
    url ldap://127.0.0.1:389/;
  }
directory-eventing {
    dispatcher-pool-size 1;
    event-base-dn o=umc;
    eventing;
    polling-interval 30;
    signature-dn o=umc;
  }
  dynamic-dn ou=dynamicConfiguration,ou=Configuration,o=Management,o=umc;
  static-dn 'l=3GPP-Gy,
ou=staticConfiguration,ou=Configuration,o=Management,o=umc';
}
logger error {
  file {
    filename var/log/3gpp-gy-error.log;
    filter /error-;
    maximum-file-size 2000000;
    rollover-filename var/log/3gpp-gy-error.alt;
  }
}
logger info {
  file {
    filename var/log/3gpp-gy-info.log;
    filter /debug-;
```

```
        maximum-file-size 2000000;
        rollover-filename var/log/3gpp-gy-info.alt;
      }
    }
    shared /3gpp-gy-share;
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring the Object Adapter Internet Address for the SRC 3GPP Gy (SRC CLI)

Use the following configuration statements to configure the object adapter Internet address for SRC's Gy:

```
slot number gy-3gpp java-orb object-adapter {
  address address;
}
```

To configure the object adapter Internet address for SRC's Gy:

1. From configuration mode, access the statement that configures the object adapter Internet address for SRC's Gy.

   user@host# **edit slot *number* gy-3gpp java-orb object-adapter**

2. Configure the address of the object adapter.

   [edit slot *number* gy-3gpp java-orb object-adapter]
   user@host# **set address *address***

3. (Optional) Verify you configuration.

```
[edit slot 0 gy-3gpp java-orb object-adapter]
user@host# show
address 10.212.10.10;
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring Logging Destinations for SRC 3GPP Gy to Store Messages in a File (SRC CLI)

Use the following configuration statements to configure file logging for SRC's Gy:

```
slot number gy-3gpp logger name ...
slot number gy-3gpp logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
```

```
}
```

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

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

   ```
   user@host# edit slot 0 gy-3gpp logger error file
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 gy-3gpp logger error file]
   user@host# set ?
   ```

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

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gy-3gpp logger error file]
   user@host# show
   filename var/log/3gpp-gy-error.log;
   filter /error-;
   maximum-file-size 2000000;
   rollover-filename var/log/3gpp-gy-error.alt;
   ```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring Logging Destinations for SRC 3GPP Gy to Send Messages to the System Logging Facility (SRC CLI) on page 201
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring Logging Destinations for SRC 3GPP Gy to Send Messages to the System Logging Facility (SRC CLI)

Use the following configuration statements to configure system logging for SRC's Gy:

```
slot number gy-3gpp logger name ...
slot number gy-3gpp logger name syslog {
   facility facility;
   filter filter;
   format format;
   host host;
   port port;
}
```

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

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

   ```
   user@host# edit slot 0 gy-3gpp logger info syslog
   ```

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 gy-3gpp logger info syslog]
   user@host# set ?
   ```

   For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 gy-3gpp logger info syslog]
   user@host# show
   facility 3;
   filter /error-;
   host loghost;
   port 514;
   ```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring Logging Destinations for SRC 3GPP Gy to Store Messages in a File (SRC CLI) on page 200
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Creating Grouped Configurations for the SRC 3GPP Gy (SRC CLI)

Configuration groups enable you to share SRC's Gy configuration with different SRC's Gy instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped SRC's Gy configuration that is shared among multiple SRC's Gy instances. For example, if you create two different SRC's Gy groups called 3gpp-gy-share and 3gpp-gy-share1 within the shared SRC's Gy configuration, you could select SRC's Gy configuration that should be associated with a particular SRC's Gy instance.

Use the **shared** option of the **slot *number* gy-3gpp** statement to select the group for an SRC's Gy instance as part of the local configuration. Use the **shared gy-3gpp group *name* configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an SRC's Gy instance. For example, to select a group called 3gpp-gy-share in the root group:

   ```
   [edit]
   user@host# set slot 0 gy-3gpp shared /3gpp-gy-share
   ```

2. Commit the configuration.

   ```
   [edit]
   user@host# commit
   commit complete.
   ```

3. From configuration mode, configure a group. For example, to configure a group called 3gpp-gy-share, specify the group as part of SRC's Gy configuration.

   ```
   [edit]
   user@host# edit shared gy-3gpp group 3gpp-gy-share
   ```

4. (Optional) Verify your configuration.

```
[edit shared gy-3gpp group 3gpp-gy-share]
user@host# show
configuration {
  nic-proxy-configuration {
    sessionHdl {
      resolution {
        key-type SessionHandle;
        resolver-name /realms/sessionHandle/A1;
        value-type SaeId;
      }
      test-nic-bindings {
        key-values {
          ANY_KEY corbaloc::127.0.0.1:8801/SAE;
        }
        use-test-bindings;
      }
    }
  }
  subscriber-types {
    session-handle {
      nic-proxy sessionHdl;
      subscriber-id-type session-handle;
    }
  }
}
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring the Subscriber Type for SRC 3GPP Gy (SRC CLI)

Use the following configuration statements to configure the subscriber type:

```
shared gy-3gpp configuration subscriber-types (session-handle) {
  subscriber-id-type (session-handle);
  nic-proxy nic-proxy;
}
```

To configure the subscriber type:

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

   ```
   user@host# edit shared gy-3gpp configuration subscriber-types session-handle)
   ```

2. Specify the subscriber ID type.

NOTE: The only subscriber ID type supported is session-handle.

```
edit shared gy-3gpp configuration subscriber-types session-handle
user@host# set subscriber-id-type session-handle
```

3. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy. In this sample procedure, the namespace for the NIC proxy called sessionHdl is configured.

```
edit shared gy-3gpp configuration subscriber-types session-handle
user@host# set nic-proxy sessionHdl
```

4. (Optional) Verify your configuration.

```
[edit shared configuration subscriber-types session-handle]
user@host# show
nic-proxy sessionHdl;
subscriber-id-type session-handle;
```

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Configuring a NIC Proxy for the SRC 3GPP Gy (SRC CLI) on page 204
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring a NIC Proxy for the SRC 3GPP Gy (SRC CLI)

Tasks to configure the NIC proxy are:

- Configuring Resolution Information for a 3GPP Gy NIC Proxy on page 204
- Changing the Configuration for the 3GPP Gy NIC Proxy Cache on page 206
- Configuring a 3GPP Gy NIC Proxy for NIC Replication on page 206
- Configuring NIC Test Data for 3GPP Gy on page 208

Configuring Resolution Information for a 3GPP Gy NIC Proxy

You create a NIC proxy for each subscriber type to be configured. Subscriber types that have different subscriber ID types can use the same NIC proxy.

Use the following configuration statements to configure the NIC proxy for 3GPP Gy:

```
shared gy-3gpp configuration nic-proxy-configuration name {
}
shared gy-3gpp configuration nic-proxy-configuration name resolution {
   resolver-name resolver-name;
   key-type key-type;
   value-type value-type;
   expect-multiple-values;
   constraints constraints;
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called sessionHdl is configured.

   user@host# edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl resolution

2. Specify the path to the NIC resolver that this NIC proxy uses. This resolver must be the same as the one that is configured on the NIC host.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl resolution]
   user@host# set resolver-name *resolver-name*

3. Specify the NIC data type that the key provides for the NIC resolution. You can provide a qualifier to a data type to distinguish between different instances of a data type in a resolution scenario, or to provide information about a data type to clarify the use of that data type in a resolution.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl resolution]
   user@host# set key-type *key-type*

   NOTE: The only valid **key-type** for SRC's Gy is **SessionHandle**.

4. Specify the type of value to be returned in the resolution for the application that uses the NIC proxy.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl resolution]
   user@host# set value-type *value-type*

   For SRC's Gy, you must set **value-type** to **SaeId**.

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

   [edit shared gy-3gpp configuration nic-proxy-configuration gy- nic2 resolution]
   user@host# set expect-multiple-values

6. (Optional. Available at the Advanced editing level.) If the application provides a constraint in the resolution request, specify the data type for the constraint. The constraint represents a condition that must or may be satisfied before the next stage of the resolution process can proceed.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl resolution]
   user@host# set constraints *constraints*

   Configure a constraint only if the constraint will be provided by the application in the resolution request. Typically, you do not need to configure constraints.

7. (Optional) Verify your configuration.

```
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
resolution]
user@host# show
key-type SessionHandle;
resolver-name /realms/sessionHandle/A1;
value-type SaeId;
```

Changing the Configuration for the 3GPP Gy NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
shared gy-3gpp configuration nic-proxy-configuration name cache {
  cache-size cache-size;
  cache-cleanup-interval cache-cleanup-interval;
  cache-entry-age cache-entry-age;
}
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called sessionHdl is configured.

   user@host# edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl cache

2. (Optional) Specify the maximum number of keys for which the NIC proxy retains data.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl cache]
   user@host# set cache-size *cache-size*

   If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.

3. Specify the time interval at which the NIC proxy removes expired entries from its cache.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl cache]
   user@host# set cache-cleanup-interval *cache-cleanup-interval*

4. (Optional) Specify how long an entry remains in the cache.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl cache]
   user@host# set cache-entry-age *cache-entry-age*

5. (Optional) Verify your configuration.

```
[edit shared configuration nic-proxy-configuration sessionHdl cache]
user@host# show
cache-size 10000;
cache-cleanup-interval 15;
```

Configuring a 3GPP Gy NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Normal editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```
shared gy-3gpp configuration nic-proxy-configuration name nic-host-selection {
  groups [groups...];
  selection-criteria (roundRobin | randomPick | priorityList);
}
shared gy-3gpp configuration nic-proxy-configuration name nic-host-selection blacklisting
  {
  try-next-system-on-error;
  number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
  blacklist-retry-interval blacklist-retry-interval;
}
```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called sessionHdl is configured.

   ```
   user@host# edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
     nic-host-selection
   ```

2. (Optional) Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests.

   ```
   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
     nic-host-selection]
   user@host# set groups groups
   ```

3. (Optional) If you configure more than one group, specify the selection criteria that the NIC proxy uses to determine which NIC host to contact.

   ```
   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
     nic-host-selection]
   user@host# set selection-criteria (roundRobin | randomPick | priorityList)
   ```

   where:

   - roundRobin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
   - randomPick—NIC proxy selects NIC hosts randomly from the list.
   - priorityList—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.

     Priorities are defined by the order in which you specify the groups. You can change the order of NIC hosts in the list by using the **insert** command.

4. (Optional) Verify your configuration.

   ```
   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
   nic-host-selection]
   user@host# show
   groups ;
   selection-criteria round-;
   ```

5. Access the statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

   ```
   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
     nic-host-selection]
   ```

```
user@host# edit blacklisting
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl nic-host-selection
  blacklisting]
```

6. (Optional) Specify whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.

```
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl nic-host-selection
  blacklisting]
user@host# set try-next-system-on-error
```

7. (Optional) Change the number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time. The default is 3.

```
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl nic-host-selection
  blacklisting]
user@host# set number-of-retries-before-blacklisting
  number-of-retries-before-blacklisting
```

8. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

```
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl nic-host-selection
  blacklisting]
user@host# set blacklist-retry-interval blacklist-retry-interval
```

9. (Optional) Verify your configuration.

```
[edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl
nic-host-selection blacklisting]
user@host# show
try-next-system-on-error;
number-of-retries-before-blacklisting 3;
blacklist-retry-interval 15;
```

Configuring NIC Test Data for 3GPP Gy

To test a resolution without the NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the NIC proxy configuration. When the SRC component configured to use a NIC proxy stub passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value. When you use a NIC proxy stub, no NIC infrastructure is required.

Use the following configuration statements to configure a NIC proxy stub from the **[edit]** hierarchy level.

```
shared gy-3gpp configuration nic-proxy-configuration name test-nic-bindings {
  use-test-bindings;
}
shared gy-3gpp configuration nic-proxy-configuration name test-nic-bindings key-values
  name {
  value;
}
```

To use the NIC proxy stub for SRC's Gy:

1. In configuration mode, navigate to the NIC proxy configuration and specify the data type of the key you want to map to a value. In this sample procedure, the key sessionHdl is specified for the NIC proxy called sessionHdl.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl]
   user@host# set resolution key-type sessionHdl

2. Enable a NIC proxy stub for a resolution.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl]
   user@host# set test-nic-bindings use-test-bindings

3. Specify the values of the keys for testing. These statements are available at the Advanced CLI editing level.

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl]
   user@host# set test-nic-bindings key-values ***name value***

   where:

   - *name*—Indicates the NIC data value for the proxy.
   - *value*—Specifies a value for the NIC data type.

   For example, to set up a login name to IP mapping for login name jane@virneo.com to the IP address 192.0.2.30:

   [edit shared gy-3gpp configuration nic-proxy-configuration sessionHdl]
   user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- *Configuring NIC Test Data (SRC CLI)*
- *Configuring a NIC Proxy for NIC Replication (SRC CLI)*
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Configuring the SAE for the SRC 3GPP Gy

You must configure the SAE to recognize SRC's Gy by adding information about SRC's Gy to the SAE properties. Tasks for configuring the SAE for SRC's Gy are:

- Configuring the SRC 3GPP Gy as an External Plug-In on page 209
- Configuring Event Publishers on page 210

Configuring the SRC 3GPP Gy as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the statement that configures the external plug-ins.

   user@host# edit shared sae configuration plug-ins name ***name*** **external**

2. Configure the object reference of the external plug-in that is exported to the SAE.

   [edit shared sae configuration plug-ins name ***name*** **external]**

user@host# **set corba-object-reference** *corba-object-reference*

where *corba-object-reference* is one of the following references:

- Path to the interoperable object reference (IOR) file in the format file:///opt/UMC/3gpp/var/run/3gpp-gy.ior
- The corbaloc URL in the format corbaloc::*host*:9889/ASGGy/statesync/<namespace> where *host* is the IP address of the C Series Controller or 127.0.0.1
- Common Object Services (COS) in the format corbaname::*host*:2809/NameService#ASGGy/statesync/<namespace>

3. Specify the plug-in attributes.

   [edit shared sae configuration plug-ins name *name* external]
   user@host# **set attributes** attribute

   Plug-in attributes for SRC's Gy include:

   USER-SESSION-HANDLE

   LOGIN-NAME

   INTERFACE-DESCR

   USER-INET-ADDRESS

   PROPERTY

   SERVICE-SESSION-NAME

   SERVICE-NAME

   IN-OCTETS

   OUT-OCTETS

   SERVICE-IDENTIFIER

   ROUTER-NAME

   For more information about configuring plug-in attributes, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

Configuring Event Publishers

You must configure the SAE to publish service-authorization events and service-tracking events to SRC's Gy at global and subscriber level. Any other events are ignored.

For information about configuring event publishers, see *Special Types of Event Publishers*.

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- *Configuring the SAE for External Plug-Ins (SRC CLI)*
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

- *Special Types of Event Publishers*

Managing the SRC 3GPP Gy (SRC CLI)

After you have configured SRC's Gy, you can perform these tasks:

- Starting the SRC 3GPP Gy (SRC CLI) on page 211
- Restarting the SRC 3GPP Gy (SRC CLI) on page 211
- Stopping the SRC 3GPP Gy (SRC CLI) on page 211

Starting the SRC 3GPP Gy (SRC CLI)

To start SRC's Gy:

```
user@host> enable component gy-3gpp
```

The system responds with a start message. If SRC's Gy is already running, the system responds with a warning message.

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Restarting the SRC 3GPP Gy (SRC CLI) on page 211
- Stopping the SRC 3GPP Gy (SRC CLI) on page 211
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Restarting the SRC 3GPP Gy (SRC CLI)

You must restart SRC's Gy after you commit a configuration change.

To restart SRC's Gy:

```
user@host> restart component gy-3gpp
```

The system responds with a start message. If SRC's Gy is already running, the system responds with a shutdown message and then a start message.

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Restarting the SRC 3GPP Gy (SRC CLI) on page 211
- Stopping the SRC 3GPP Gy (SRC CLI) on page 211
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

Stopping the SRC 3GPP Gy (SRC CLI)

To stop SRC's Gy:

```
user@host> disable component gy-3gpp
```

The system responds with a shutdown message. If SRC's Gy is not running when you issue the command, the system responds with the command prompt.

Related Documentation

- Configuring the SRC 3GPP Gy (SRC CLI) on page 189
- Starting the SRC 3GPP Gy (SRC CLI) on page 211
- Restarting the SRC 3GPP Gy (SRC CLI) on page 211
- Configuration Statements for the SRC 3GPP Gy on page 187
- SRC 3GPP Gy Overview on page 179

PART 5

Index

Index

Symbols