Oracle® Communications Service Broker

Policy Controller Implementation Guide

Release 6.1

E29455-01

February 2013

ORACLE®

Oracle Communications Service Broker Policy Controller Implementation Guide, Release 6.1

E29455-01

Contents

3 Configuring Subscriber Profile and Charging Information

4 Monitoring Policy Controller Using Runtime MBeans

8 Creating Policy Charging and Control Profiles

9 Creating Application Detection and Control Profiles

10 Strategies for Creating Rules

11 Creating Rules and Rulesets

12 Working With Service Data Records

A Policy Controller Reference

Preface

This document describes how to install, configure, and use the Oracle Communications Service Broker Policy Controller (Policy Controller) to set bandwidth service levels or limits for telecommunications subscribers.

Audience

This document is intended for IT professionals who install, configure, or use Policy Controller.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Communications Service Broker Release 6.1 documentation set:

- *Oracle Communications Policy Controller Release Notes*
- *Oracle Communications Service Broker Concepts Guide*
- *Oracle Communications Service Broker Installation Guide*
- *Oracle Communications Service Broker Administrator's Guide*
- *Oracle Communications Service Broker Signaling Server Units Configuration Guide*
- *Oracle Communications Service Broker Subscriber Store User's Guide*
- *Oracle Communications Service Broker Policy Controller Protocol Implementation Conformance Statement*

Downloading Oracle Communications Documentation

Oracle Communications Service Broker documentation is available from the Oracle software delivery Web site:

http://edelivery.oracle.com/

Additional Oracle Communications documentation is available from Oracle Technology Network:

http://www.oracle.com/technetwork/index.html

1

About Policy Controller

This chapter provides an overview of Oracle Communication Service Broker Policy Controller (Policy Controller) and explains its capabilities and components.

Policy Controller is a Policy and Charging Rules Function (PCRF) product that makes business policy decisions as defined by the 3GPP TS 23.203 v9.9.0 (2011-06) specification.

What Policy Controller Does

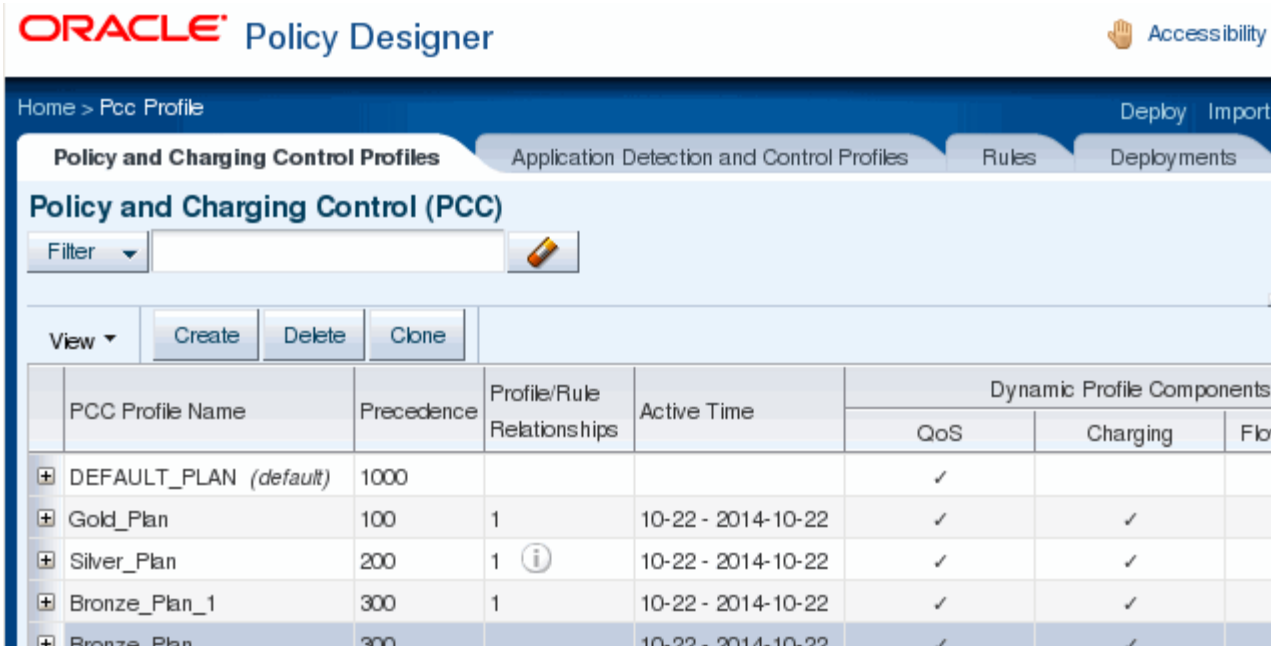
You use Oracle Communications Policy Controller to shape your subscriber's service data flow. It is a decision engine with a graphical interface that you use to specify minimum and maximum bandwidth limitations, charging information, and any application redirection instructions. Policy Controller is aware of each subscriber's services, devices, subscriber profile repository (SPR) information, and online charging system (OCS) information. Consequently, you can offer subscribers a highly personalized data experience.

Policy Controller enables you to closely control how you charge subscribers for data and bandwidth consumption based on combinations of subscriber, device, and customer profile data. You specify this traffic shaping data when you create *profiles* which set Quality of Service (QoS) limits, charging information, and/or application redirection information. These profiles are the equivalent of the "PCC rules" defined in the 3GPP 23.203 v9.9.0 (2011-06) specification. Your profiles define exact QoS limits and link them to charging information already set in your online charging system to influence subscriber data consumption. You can have any number of profiles, and a default profile is available in case no other profiles apply.

Once your traffic shaping profiles are defined, you create Policy Controller *rules* that dynamically decide which profiles to apply to a subscriber at the start of each session. Your rules can use input from a variety of sources, such as subscriber profile information, the applications themselves, and so on. Policy Controller can revaluate the profile selection decisions as subscriber and network information changes dynamically. These decisions are then passed on to your Policy Charging and Enforcement Function (PCEF) which enforces these decisions.

Policy Controller includes the Policy Designer graphical interface that you use to create these profiles and rules. Figure 1–1 shows the Policy Designer Policy and Charging Profiles tab with some example profiles defined.

Figure 1–1 Policy Designer Policy And Charging Control Profile Tab

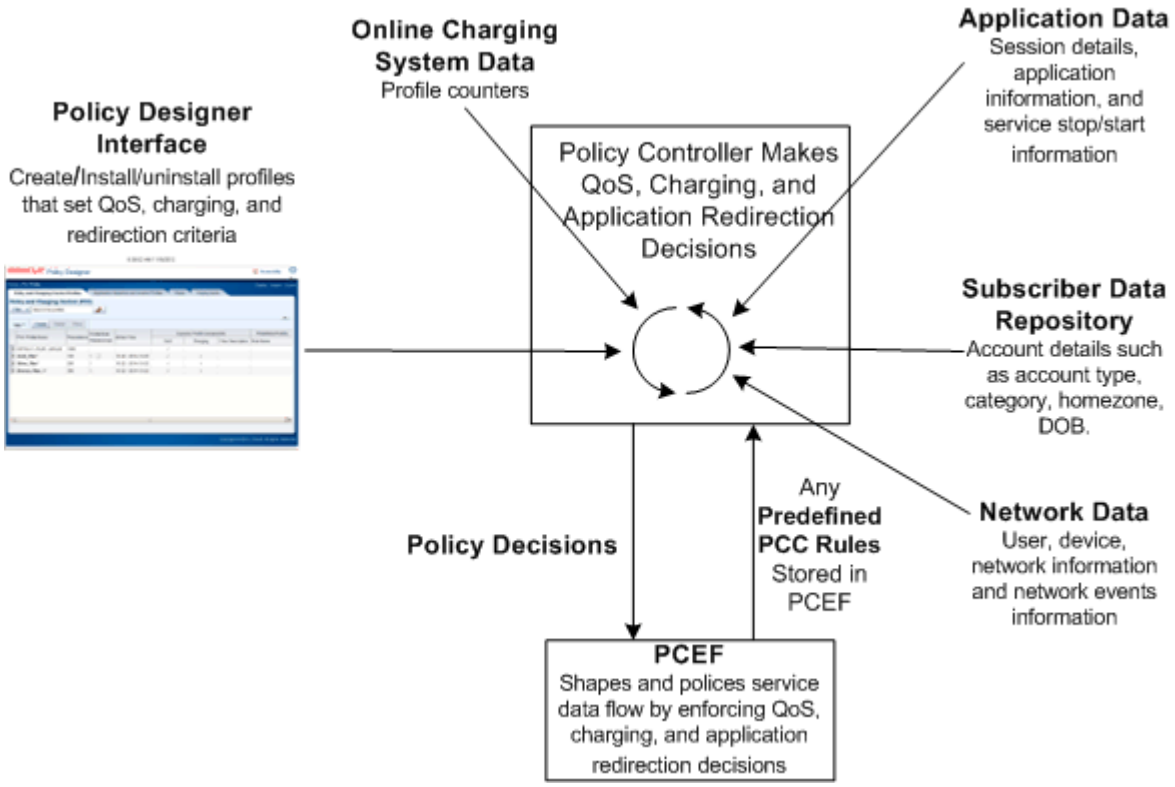


After you have created profiles and the rules that select them, Policy Controller applies the appropriate profiles to individual subscribers each time they start a session. Policy Controller does this by evaluating all of the rules you have created and using them to decide which profiles apply to the individual subscriber. As the session continues, Policy Controller reevaluates the rules based on new input from your other PCRF entities, such as your Subscriber Profile Repository (SPR), OCS, PCEF, or application functions (AFs). If the reevaluation causes a change in the services or bandwidth levels that a subscriber is entitled do, Policy Controller informs your PCEF for enforcement. A common reevaluation example is changing service QoS limits when a subscriber reaches a usage threshold.

As your product line changes, you can easily expand or change the profiles that specify the QoS and charging information in your profiles, and the rules that decide which subscribers are entitled to those profiles. This flexibility enables you to taylor your products to the specific usage habits, needs, and budgets of your subscribers.

Figure 1–2 shows how Policy Controller makes decisions about which profiles each subscribers is entitled to. This figure shows the various entities that offer input to the Policy Controller decision making process.

Figure 1–2 Overview of the Policy Controller Decision Making Process



Your PCEF (or Traffic Detection Function (TDF)) enforces the decisions that Policy Controllers makes. For example, the PCEF directs the AF to change a level of service, and directs the charging engine to charge for those services.

If your PCEF already contains PCC rules that you want to use, you can add them to Policy Controller as *predefined* PCC profiles. Predefined PCC profiles are abbreviated PCC profiles that reference the existing PCC rule by name.

Policy Controller supports these profiles (the equivalents of PCC rules) that shape your service data flows:

- PCC profiles: Policy Charging and Control profiles that specify network QoS limits, charging information, and a valid activation time limit. These profiles specify how broadband resources are allocated among your subscribers, and charged for. You can use a wide variety of subscriber profile, application, and network data to select PCC profiles.
- ADC profiles: Application Detection and Control profiles that specify QoS limits, traffic redirection instructions, and a valid time limit for specific applications. These profiles target individual applications in your PCRF implementation for the QoS limits and/or redirection. Redirection enables you to send all traffic for an application to a specific Web location, such as an "HTTP Error 403 Forbidden" message or a simple "Application Not Allowed" Web page that you create.

The Policy Controller design enables you to create highly-customized sets of decision criteria to apply to your subscribers. The simplest example is creating PCC profiles (say, gold silver, and bronze) that index QoS limits to the monthly charges your subscribers pay. Policy Controller enables you to create highly-customized pricing plans for your services that give you the flexibility to provide special deals, exceptions, and promotions that take effect and expire whenever you choose.

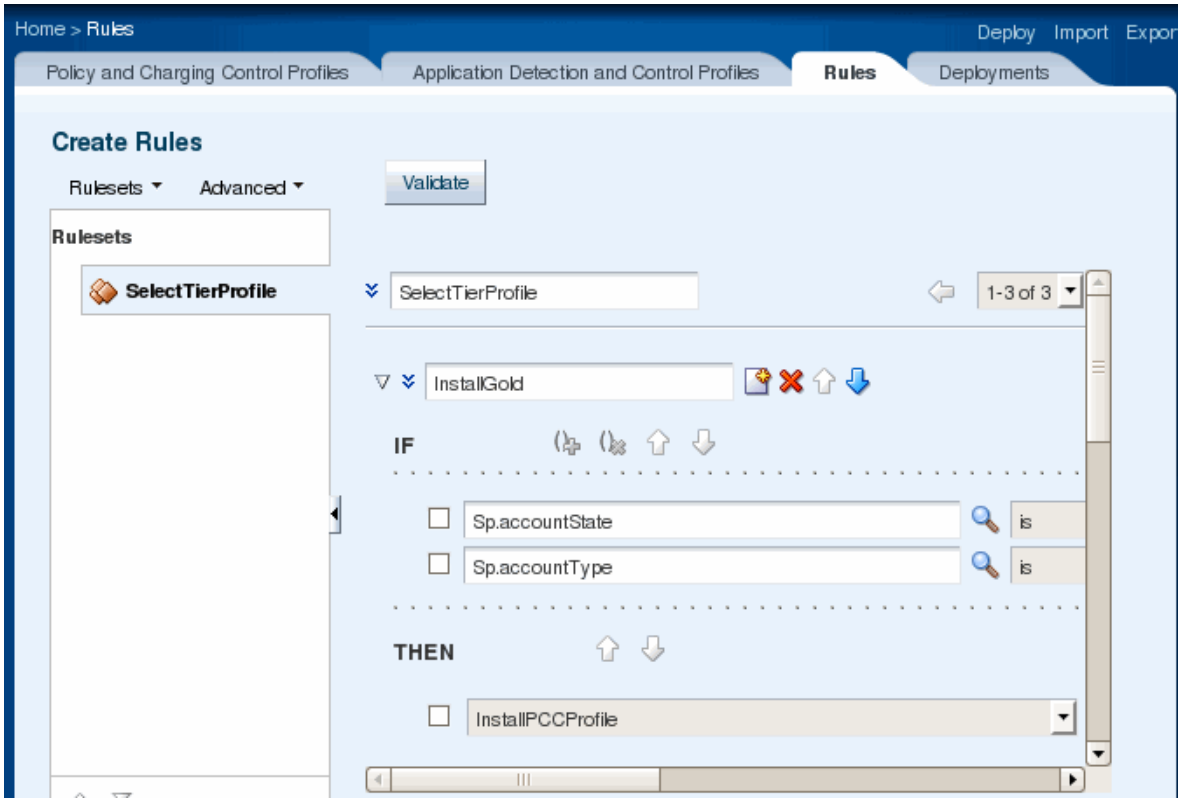
After creating PCC and ADC profiles, you create Policy Controller rules that select the profiles to apply to subscriber service data flow. Policy Controller rules are if-then statements that can use information (AVPs) from a variety of Diameter reference points, including:

- A PCEF (Diameter Gx reference point)
- An SPR (Diameter Sp reference point) that you configure as part of your Policy Controller implementation. See "Configuring Subscriber Profile and Charging Information" for details on the supported SPRs.
- Your application functions (Diameter Rx reference point)
- The Bearer Binding and Event Reporting Function (BBERF) (Diameter Gxx interface) that is part of your Policy Controller implementation.
- An OCS (Diameter Sy reference point) that you configure as part of your Policy Controller implementation. See "Configuring Subscriber Profile and Charging Information" for details on the supported OCSs.
- Between visitor and home PCRFs (Diameter S9 reference point).

Policy Controller takes information from these various sources, interprets the information using the rules you created, and decides which profiles apply to individual subscribers.

Figure 1–3 shows the Policy Designer **Rules** tab that you use to create rules, with some example rule information. See "About the Policy Designer Interface" for more information.

Figure 1–3 Policy Designer Rules tab



See "Strategies for Creating Rules" for example rules that illustrate the various approaches available for applying PCC and ADC profiles to your subscribers. See "Creating Rules and Rulesets" for instructions on how to create rules.

Policy Controller obtains subscriber profile and charging information from these supported SPR/OCSs:

- A Service Broker (local) SPR/OCS.
- Oracle Communications Billing and Revenue Management (BRM)
- Any third-party SPR/OCS using Diameter-based connectivity
- Any third-party SPR/OCS using Web-based connectivity

Policy Controller supports Offline charging through the PCEF, but the nature of offline charging prevents an offline charging system from updating Policy Controller dynamically. See "Configuring Subscriber Profile and Charging Information" for details on configuring an SPR/OCS.

Policy Controller includes the Policy Designer interface that you use to graphically:

- Create PCC profiles that shape your subscriber's service data flow.
- Create ADC profiles that gate your subscriber's service data flow.
- Create the rules that Policy Controller uses to select PCC and ADC profiles for individual subscribers, and gather those rules into rulesets.
- Specify the order in which rulesets are applied.
- Create deployments that include rulesets and profiles.
- Deploy the deployments to make them take effect.
- Export deployments to a file to work on them later.

Once you deploy a deployment, Policy Controller interprets the rules in the rulesets it contains, makes decisions regarding the QoS limits, charging information, and application redirection, and passed them on to your PCEF for enforcement.

Policy Controller also provides SMS messaging and redirection capabilities within a session. You can configure rules that send subscribers an SMS message, or redirect them to a URL of your choice when they meet the conditions you set. The SMS or target URL can contain any content that your implementation requires. These features are popular for "bill shock" prevention by using them to alert subscribers to diminishing balances and prompt them to add resources. You can also offer targeted promotions customized to individual subscribers (such as a deal on their birthday).

By default Policy Controller accepts all Diameter AVPs for the Diameter reference points it supports. You can also add support for any custom AVPs that your PCEF, AF, or other policy node uses, and use the AVPs in Policy Controller rules.

Policy Controller is based on the 3GPP standard and is Release 9 compliant. See Oracle *Communications Service Broker Policy Controller Protocol Implementation Conformance Statement* for details on the specifications supported.

About Policy Controller Scalability

Policy Controller uses a highly-available built-in coherence infrastructure based on the *Oracle Coherence* technology that you use to easily and quickly add capability. You configure coherence each time you create a domain, and your data is automatically replicated, and sessions remain persistent. For details, see the discussions about

scaling the deployment in *Service Broker Installation Guide* and the discussion about clusters in *Service Broker Administrator's Guide*.

About Policy Controller Terms

You must understand the following terms when using Policy Controller:

Application Detection and Control (ADC) Profile
A Policy Controller entity that specifies QoS limits and any redirection instructions for service data flow originating with a specific application. You define activation and deactivation time limits for each ADC profile.

Application Function (AF)
AFs are the applications, or services that you provide to your subscribers and (generally) charge for. Policy Controller communicates with your AFs using the Diameter Rx protocol.

Bearer Binding and Event Reporting Function (BBERF)
Provides user plane traffic handling as defined in the 3GPP TS 23.203 v9.9.0 (2011-06) specification. Your BBERF is responsible for: bearer binding, uplink bearer binding verification, event reporting to the PCRF, and service date flow detection.

Deployment
A collection of profiles, rules, and lists of values that work together as a logical unit. Policy Controller enables you to work on these entities as a set and then save them as a **.rap** file for later work. You can also send deployment files to someone else to work on, and then import them back in to your Policy Designer. The *currently active* deployment is the set of profiles, rules, and supporting data that is open and being edited in Policy Designer.

Dynamic Profiles
You specify the details of dynamic profiles using Policy Designer and store these profiles in Policy Controller. If a profile is not dynamic, then it is *predefined*.

Policy and Charging Rules Function (PCRF)
PCRF is a policy control decision engine defined in the 3GPP TS 23.203 v9.9.0 (2011-06) specification. Policy Controller is the Oracle Communications PCRF product.

Facts
Objects that rules reason on. These facts are predefined, however you can define your own local facts to use within a deployment.

List of Values
Originally called bucketsets, list of values are frequently used data values that you use in rules, such as days of the week. You can create your own lists of values as needed.

PCC Rule
See PCC profile.

Policy Control Enforcement Function (PCEF)
PCEF is the policy enforcement engine that you set up to accept the policy decisions from Policy Controller. Your PCEF accepts policy decisions from Policy Controller and enforces those decisions. Policy Controller communicates with your PCEF by using the Diameter Gx protocol.

PCC Profile

PCC profiles are the Policy Controller implementations of PCC rules as defined in the 3GPP TS 23.203 v9.9.0 (2011-06) specification. They specify Quality of Service (QoS) limits and charging information. You define activation and deactivation time limits for each PCC profile.

Predefined Profiles

Predefined profiles reference PCC rules that are already defined and stored in your PCEF. Policy Controller references these rules by naming them in abbreviated PCC or ADC profiles called predefined profiles.

Rule

Policy Controller rules are if-then statements that Policy Controller uses to decide which PCC or ADC profiles to apply to a subscriber. You create rules by using the Policy Designer **Rules** tab.

Ruleset

Rulesets contain a set of rules. Rulesets are intended for you to group logically related rules for convenient management.

Traffic Detection Function (TDF)

A PCEF enhanced with ADC. A software entity that sorts or filters traffic based on its packet or datagram content (service data flow). A TDF can redirect, gate, block, or shape traffic *by application*, or permit it to pass it unrestricted. See "Application Detection and Control (ADC) Profile" for details on ADC profiles..

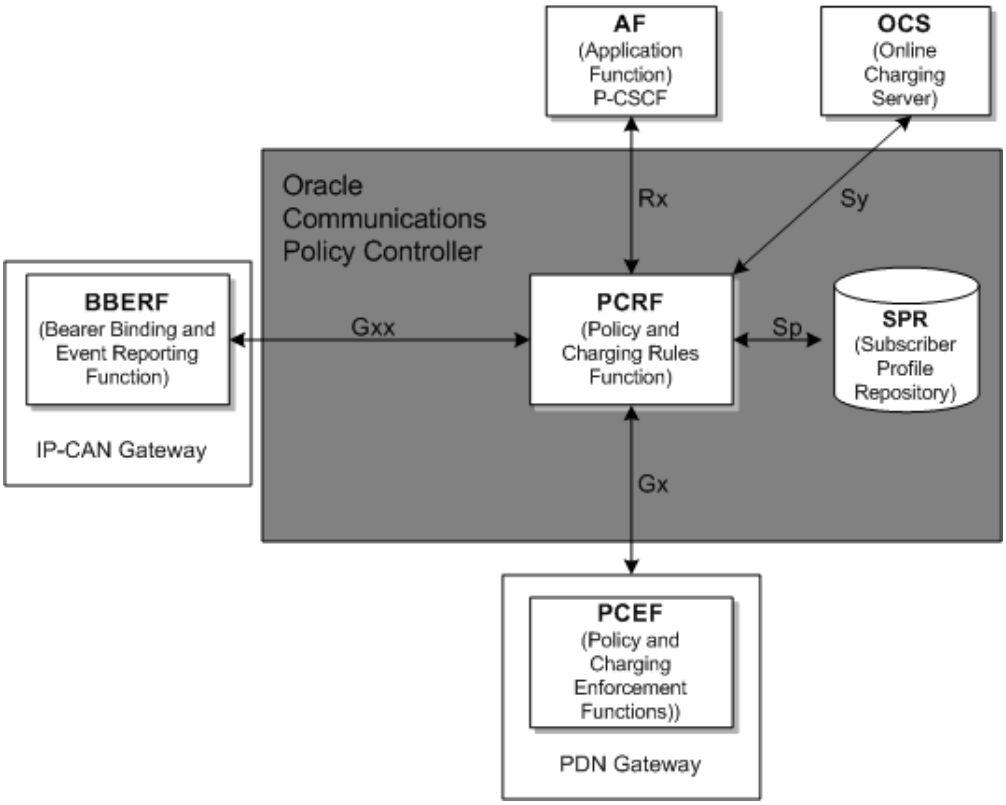
About the Policy Controller Architecture

This section explains how Policy Controller relates to the other entities in your PCEF implementation and introduces you to the Policy Controller back end components.

Architecture Overview

Figure 1–4 shows the software components that interact with Policy Controller.

Figure 1–4 Policy Controller Implementation Components



About the Policy Controller Back End Components

The Policy Controller back end entities are listed in Table 1–1. You configure these entities as part of setting up Policy Controller and Policy Designer.

Table 1–1 Service Controller Entities that Policy Controller Uses

Service Broker Entity	Policy Controller Usage
Domains	Policy Controller requires that you create a domain for the processing and signaling servers to run in. Your domains may be clustered. You create a domain or domains as part of installing and configuring Policy Controller. See *Service Broker Installation Guide* for information on domains.
Processing Server	Policy Controller is executed one or more processing servers which are part of a domain. See *Service Broker Concepts Guide* for background information on processing servers and domains.

Table 1–1 (Cont.) Service Controller Entities that Policy Controller Uses

Service Broker Entity	Policy Controller Usage
Signaling Servers	Policy Controllers uses signaling servers (SSUs) to connect to external entities such as: ■ The Diameter SSU to configure charging, and route traffic to your PCEF and AFs. ■ The BRM SSU, if you use BRM as an SPR/OCS. ■ The SMPP SSU to add SMS capability to your rules. You create these SSUs as part of installing and configuring Policy Controller. See *Service Broker Concepts Guide* for background information on signaling servers and SSUs, and see the *Service Broker Signaling Server Units Configuration Guide* for instructions on configuring individual SSUs.
Administration Console	You use the Administration Console user interface for a a variety of Policy Controller configuration tasks. See *Service Broker Administrator's Guide* for details on using the Administration Console.
Subscriber Profile Repository (SPR) and online charging system (OCS)	You have these options for using an SPR/OCS to store and retrieve subscriber profile and charging information: ■ The local Policy Controller SPR. ■ The Oracle Communications Billing and Revenue Management SPR/OCS capabilities. ■ A third-party SPR/OCS using Web-based connectivity. ■ A third-party SPR/OCS using Diameter-based connectivity. You will select and configure an SPR while installing and configuring Policy Controller. See "Configuring Subscriber Profile and Charging Information" for details on configuring connections to these SRPS.

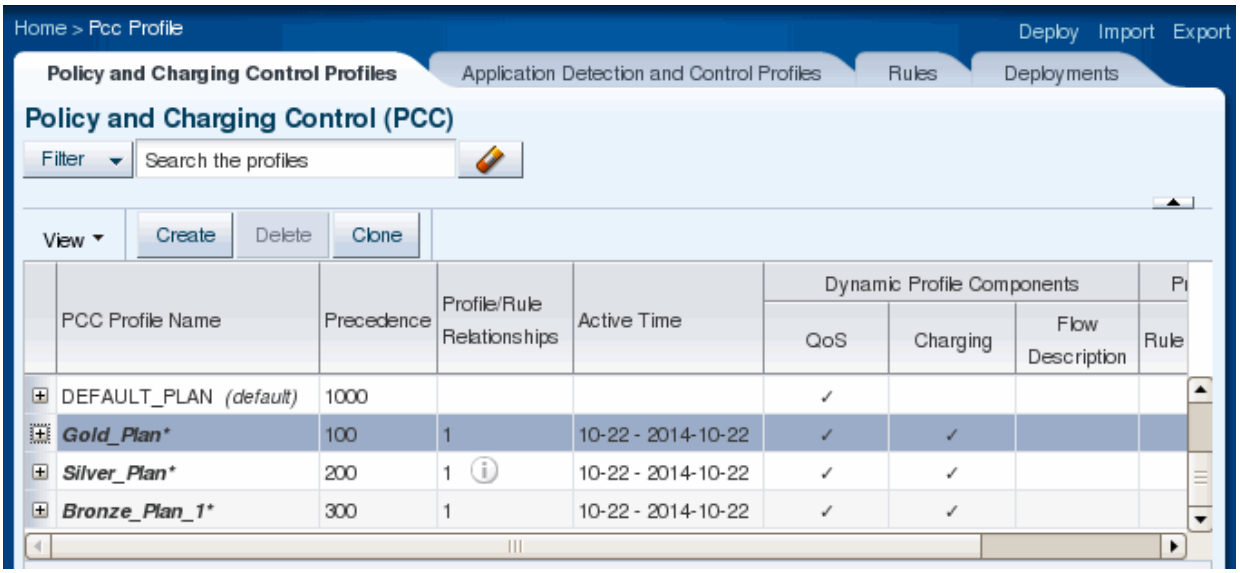
About the Policy Designer Interface

Policy Controller includes the Policy Designer interface that you use to:

- Create PCC profiles that specify QoS limits and charging information.
- Create ADC profiles that specify QoS limits, and redirection instructions for service data flow for individual applications.
- Create rules that determine which PCC and ADC profiles to apply to which subscribers.
- Collect your rules into logical collections called rulesets, and specify how to apply them.

Figure 1–5 shows the **Policy Charging and Control Profile** tab of the Policy Designer that you use to create PCC profiles. You then create rulesets to apply them to subscriber traffic. The Policy Designer shows these tabs: **Policy and Charging Control Profiles**, **Application and Detection and Control Profiles**, **Rules**, and **Deployments**. The **Policy Charging and Control Profile** subtab is shown displayed with a few PCC profiles, one per row.

Figure 1–5 The Policy Designer User Interface



Using Policy Designer your personnel can change business rules from any Web browser without stopping business processes.

About Creating PCC Profiles to Set Bandwidth and Charging Levels

You use the Policy Designer **Policy and Charging Control (PCC)** tab to create PCC profiles that specify the bandwidth limit and charging aspects of products that you sell to customers. A simple example is creating "Gold," "Silver," and "Bronze" PCC profiles to specify tiered levels of bandwidth that each have a different fee. Gold would be the fastest, and you charge the most for it. Silver would have slower speeds, but also cost less, and Bronze is the economy option that allows the slowest speeds, but costs the least. You can specify QoS minimum or maximum limits any way that your implementation requires. Policy Controller rules probe subscriber data when a session is started and dynamically decides which PCC profiles to apply. The PCC profile selection can change as the subscriber data changes during the session.

PCC profiles generally contain information about a specific service, but they are flexible and you can create one that applies to all services. If the PCC profile does not contain QoS or charging specifics then your PCEF must provide that information.

Keep rule flexibility in mind when planning your PCC profiles. Once you have created profiles that correspond to your products, you can create rules to decide which subscribers are entitled to those profiles. Rules can be simple or complex; they act on information from these Diameter reference points:

- Application and media information from the AF
- Internal rule configurations (other PCC profiles)
- Subscriber information from the SPR
- Information such as event triggers from the PCEF
- Charging information from an OCS
- Information from other PCRFs
- Information from the BBERF

There are two ways Policy Controller can send information to the PCEF:

- Pull (solicited provisioning). Sending PCC profiles in response to a CCR request message from the PCEF. The request is answered in a CCA message.
- Push (unsolicited provisioning). Sending PCC profiles to the PCEF based on new data. To do this the Policy Controller includes the profiles in an RA-request message instead of using CCR/CCA messages.

Your rules can also use the event triggers specified in 3GPP TS 23.203 for pull provisioning. Event triggers are convenient shortcuts that your rules can use to apply profiles based on a subscriber's current location, credit status, Connectivity Access Network changes, and other common events. Policy Controller includes some default triggers and you can define additional triggers that your implementation requires.

About Creating ADC Profiles to Manage Application Traffic

You use the Policy Designer **Application Detection and Control Profiles** tab to create ADC profiles to set bandwidth limits for, and optionally redirect specific applications. ADC takes advantage for the Deep Packet Inspection (DPI) capabilities that TDFs offer to target specific applications for bandwidth limits or redirection.

ADC profiles use the same bandwidth limiting features that PCC profiles do. See "Creating an ADC Profile" for details.

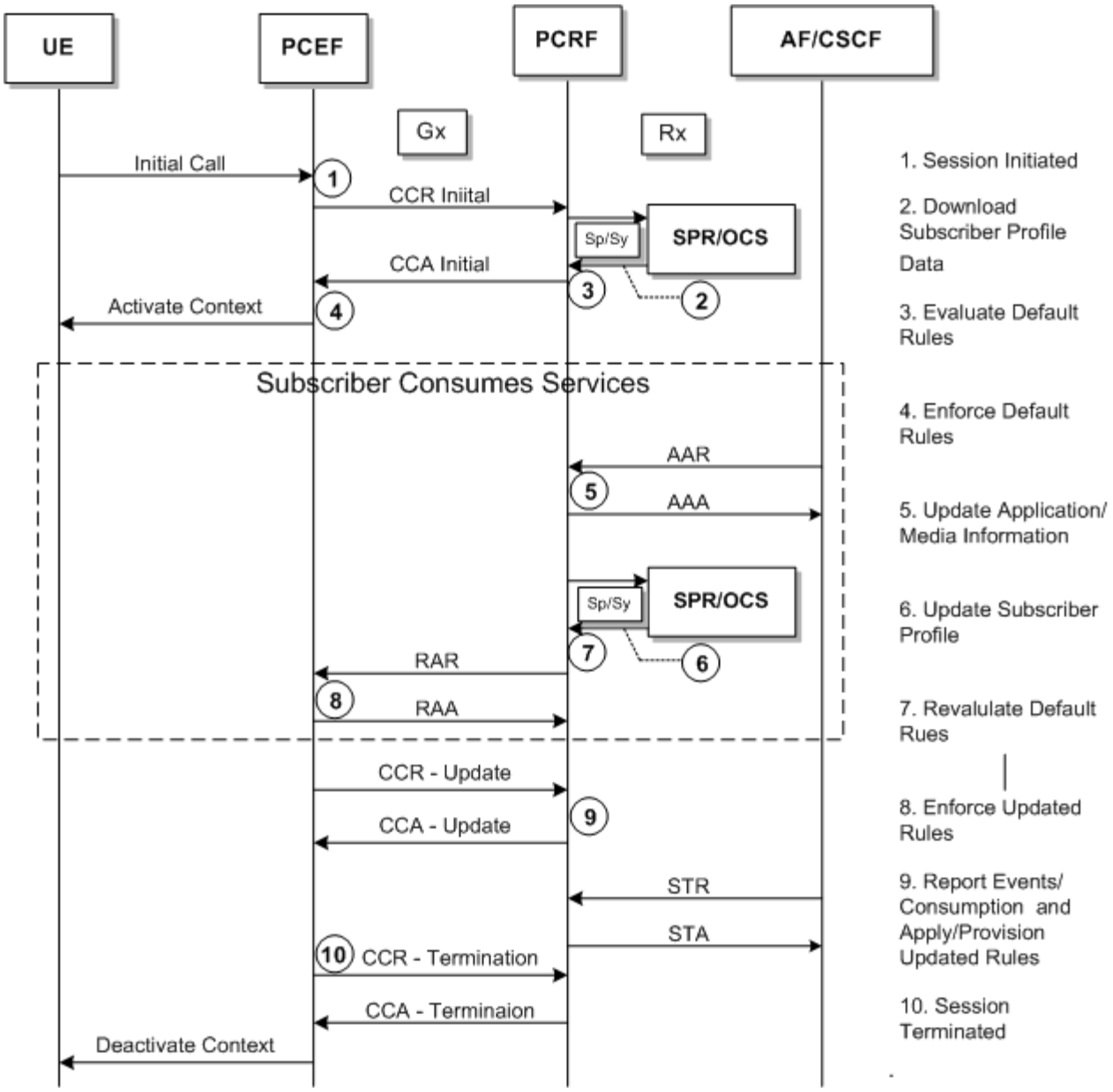
You can redirect subscriber service data flow to an IP address or URL You can redirect traffic for any reason that your implementation requires, such as "parental control" type restrictions. This setting also overrules any PCEF settings.

See "Creating an ADC Profile" for details on redirecting application traffic.

About the Policy Controller Session Call Flow

Figure 1–6 shows the call flow of a typical Policy Controller session. The call flow helps you understand the Policy Controller components and its features.

Figure 1–6 Typical Policy Controller Call Flow



2

Installing and Configuring Policy Controller

This chapter explains how to install and configure an Oracle Communications Service Broker Policy Controller (Policy Controller) implementation.

Policy Controller Hardware and Software Requirements

This section lists the hardware and software components that your Policy Controller implementation requires.

The list of hardware and software components that you need depends on the extent of your Policy Controller Implementation. Policy Controller itself is a Policy and Charging Rules Function (PCRF) software node, and you need to also obtain a Policy and Charging Enforcement Function (PCEF) software node. The remaining components depend on the services you intend to deliver to your subscribers. Depending on the extent of your implementation you may also need to obtain:

- A BBERF to connect to your IP-CAN, and handle policy signal flow.
- A Diameter charging server to charge your subscribers for services, if not using Oracle Communications Billing and Revenue Management (BRM).
- A Short Message Service Center (SMSC) if you take advantage of the Policy Controller SMS message sending feature. This feature enables you to send Short Message Service (SMS) messages either to your subscribers or to your Policy Controller administration personnel based on changing subscriber information.

Any additional hardware and software components depend on your implementation's needs.

Installing and Deinstalling Policy Controller

You install Policy Controller using the Oracle Communications Service Broker Installation program. See *Oracle Communications Service Broker Installation Guide* for instructions on installing and deinstalling Policy Controller. During installation:

- Select **Policy Controller 6.1.0.0** from the installer options.
- A JDK is required to run Policy Controller, and you have these options:
  - Java JDK (recommended): Oracle recommends that you use this JDK for Policy Controller production implementations because its "HotSpot" garbage collector is efficient for cleaning up session information from the heap. See "Configuring Java JVM Parameters" for suggestions on how to configure this JDK.

– JRockit JDK (the default): The JRockit JDK's is appropriate if your production implementation of Policy Controller generally has very short sessions. Cleaning up sessions with very short response times are most easily achieved by using JRockit's deterministic garbage collector. See "Configuring Java JVM Parameters" for suggestions on how to configure this JDK.

Creating and Starting a Domain and Managed Server for Policy Controller

For each Policy Controller installation create one or more managed servers and a domain. See the discussions on setting up a domain in *Oracle Communication Service Broker Installation Guide* for information about creating the domain, adding managed servers, and managed server starting those components.

Understanding Policy Controller Memory Requirements

Policy Controller requires a lot of memory to store subscriber data. It caches session and subscriber profile information to achieve a nearly-realtime response, and you need to reserve significant resources for it. Policy Controller's memory requirements are most effected by:

- The expected number of subscribers per server or blade.
- The required transactions per second rate.
- The expected session duration time.
- The service availability option you choose. The service continuity option requires significantly more memory that service availability. For details, see the discussion on about setting the service availability mode in *Service Broker Installation Guide*.

A good general rule is to reserve 4 KB of memory for each active subscriber session running any given time. The 4 KB is divided between in-memory space required for subscriber profile information, and Java heap memory for session information. The in-memory requirements are generally affected more by the number of active subscribers caching data, and the Java heap is generally affected more by the length of their sessions. As session length increases, so do Java heap requirements, and they can require 15-20 GB of space per blade to maintain high performance response. The large Java heap memory requirement are a challenge for the Java heap garbage collector. See "Configuring Java JVM Parameters" for guidelines for configuring your JDK heap size.

Configuring Java JVM Parameters

Tune your Java JDK garbage collection performance to reduce the time required to perform a full garbage collection cycle. Do not attempt to tune the JVM to minimize the frequency of full garbage collections, because this generally results in an eventual forced garbage collection cycle that may take up to several full seconds to complete.

The following examples list domain configuration settings for different Policy Controller configurations. Implementations that use service availability require more memory because of the additional data replication. See *Oracle Communications Service Broker Installation Guide* for details on the service availability and continuity options.

Example Configuration for a Unified Domain with Service Continuity

Table 2–1 lists Java JDK tuning settings for a domain using service continuity, in a clustered deployment (2 Sun Blade X6270 blades, each running two JVMs) with this expected load:

- Total subscribers: 1 million/blade
- Average Policy Controller session length: 30 minutes
- Expected traffic throughput: 1,500 sessions/second/blade
- JVM heap size: 14 GB

These settings generally work well in this environment. Adjust the following settings for your hardware and software configuration.

Table 2–1 JDK Parameters for a Unified Domain with Service Continuity

Category	Setting
Memory and Garbage Collection Options	AXIA_OPTS="-d64 -XX:MaxPermSize=512m -XX:PermSize=512m" -Xms14g -Xmx14g -Xmn3g -XX:+UseCompressedOops -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly"
Application Settings	AXIA_OPTS="${AXIA_OPTS} -Dsubscriber.core.cache.mem.initial=2047" AXIA_OPTS="${AXIA_OPTS} -Doracle.ocsb.app.rcc.feature.pcrf.max.session.actors=7000" AXIA_OPTS="${AXIA_OPTS} -Dsubscriber.core.wb.high-units=10000 -Dsubscriber.core.wb.low-units=5000" AXIA_OPTS="${AXIA_OPTS} -Doracle.axia.util.executor.max_clean_time=1200000"

Example Configuration for Service Availability

Table 2–2 lists Java JDK tuning settings for a unified domain in a clustered deployment (2 Sun Blade X6270 blades, each running 3 JVMs) with this expected load:

- Total subscribers 1 M per blade
- Average Policy Controller session length: 30 minutes
- Expected traffic throughput: 3,500 sessions/second/blade
- JVM heap size: 17 GB

These settings work well in this environment. Adjust the following settings for your hardware and software configuration.

Table 2–2 JDK Parameters for a Unified Domain with Service Availability

Category	Setting
Memory and Garbage Collection Options	AXIA_OPTS="-d64 -XX:MaxPermSize=512m -XX:PermSize=512m -Xms14g -Xmx14g -Xmn3g -XX:+UseCompressedOops -XX:+UseConcMarkSweepGC-XX:+UseCMSInitiatingOccupancyOnly"
Application Settings	AXIA_OPTS="${AXIA_OPTS} -Dsubscriber.core.cache.mem.initial=2047" AXIA_OPTS="${AXIA_OPTS} -Doracle.ocsb.app.rcc.feature.pcrf.max.session.actors=7000" AXIA_OPTS="${AXIA_OPTS} -Dsubscriber.core.wb.high-units=10000 -Dsubscriber.core.wb.low-units=5000" AXIA_OPTS="${AXIA_OPTS} -Doracle.axia.util.executor.max_clean_time=1200000"

Starting, and Configuring Policy Designer

This section explains how to start and configure Policy Designer. This section assumes that you have created a domain and managed server, as described in *Oracle Communication Service Broker Installation Guide*.

Starting Policy Designer

Use the following steps to start Policy Designer:

1. Log on to the Service Broker system on which you created the Policy Controller domain.
2. Start the domain (Administration Server) (this also starts the Policy Controller JETTY server):

   ```
   cd Oracle_home/ocsb61/admin_server
   ./start.sh domain_configuration_directory
   ```

   Where:

   Oracle_home is the Oracle home directory you defined when you installed Service Broker.

   domain_configuration_directory is the path to the domain configuration directory.

   Note: If you use basic authorization (**axia.digest.auth=true** in *Oracle_home*/**ocsb61/admin_server/properties/admin.properties**) you will be prompted for two sets of credentials to use when starting the Administration Console. The first set is required to access the Administration Console, and the second set is required to access the Policy Designer user interface.

   Note: The Policy Controller does not work with the standalone version of the Administration Console.

3. Start the managed server you created during installation with these commands:

   ```
   cd Oracle_home/ocsb61/managed_server
   ./start.sh managed_server_name file:///domain_configuration_
   directory/initial.zip
   ```

   Where:

   Oracle_home is the Oracle home directory you defined when you installed Service Broker.

   managed_server_name is the name of the managed server file you created.

   domain_configuration_directory is the path to the domain configuration directory.
4. Open a Web browser.
5. Enter one of these Policy Designer URLs:

   (SSL off) **https: //** [**localhost** | *IP_address*] **:** *port_number* **/policydesigner**

   (SSL on) **http: //** [**localhost** | *IP_address*] **:** *port_number* **/policydesigner**

   Where:

- *IP_address* is the IP address of the Service Broker server running Policy Controller.
- *port_number* is the server port number to use. The defaults are 8090 (SSL off) and 8091 (SSL on). You can change the default port numbers. See "Setting the Policy Designer Port Number" for details.

If you enabled basic authorization (**axia.digest.auth=true** in *Oracle_home*/**ocsb61/admin_server/properties/admin.properties**) you are prompted for the username and password that you entered when you created the Policy Controller domain.

This example starts Policy Designer with SSL off on your local system using the default port number:

```
http://localhost:8090/policydesigner
```

Note: If SSL is off, use **http**, not **https** to avoid **ssl_error_rx_record_too_long** errors.

For information on changing the default port number, see "Setting the Policy Designer Port Number".

Setting the Policy Designer Port Number

The default Policy Designer uses port 8090 using HTTP, and 8091 using HTTPS. You can change this by following these steps:

1. Open the common.properties file located in *Oracle_home*/**admin_server/common.properties**.
2. Add one of the following entries:
   - If you do not use SSL:

     ```
     oracle.ocsb.app.rcc.pcrf.gui.port=port_number
     ```

   - If use SSL:

     ```
     pcrf.gui.http.port.secure=port_number
     ```

   Where *port_number* is the new port number to use.
3. Restart the Administration Console.

   For details, see *Oracle Communications Service Broker System Administrator's Guide*.

Using the Policy Designer Accessibility Features

All Policy Designer interface windows include an **Accessibility Preferences** link that you use to access the interface's accessibility capabilities.

The accessibility options include these alternatives to the default browser preferences:

- Screen reader - Puts the browser in screen reader mode.
- High contrast - Adds a background color to make text easier to see.
- Large Fonts - Increases the font size of all Policy Controller interface text.

Selecting Accessibility Features

To enable Policy Designer accessibility features:

1. Start the Policy Design interface. See "Starting Policy Designer" for details.
2. Click the **Accessibility** link in the upper right of your browser page with a mouse, or use the Ctrl-Alt-A keyboard shortcut.

   The **Accessibility Preferences** page appears.
3. Click any combination of these accessibility check boxes to select the feature:
   - **Screen reader**
   - **High contrast**
   - **Large fonts**
4. Click **OK** to make your changes take effect.
5. Click **Home** to return to the Policy Designer interface.

Disabling the Policy Designer Automatic Start

By default, the Policy Designer process is started when you start the Service Broker domain that contains it. To prevent this automatic startup, use the following steps:

1. Add this entry to the *Oracle_home*/**admin_server/common.properties** file:

   ```
   oracle.ocsb.app.rcc.pcrf.gui.disable=true
   ```

   A value of **true** disables the automatic startup. If this entry is **false** or missing, the Policy Designer starts automatically.
2. Restart the Administration Console.

   For details, see *Oracle Communications Service Broker System Administrator's Guide*.

Configuring Policy Controller

This section explains the tasks necessary to configure Policy Controller for your implementation. The "Policy Controller Configuration Checklist" section servers as a guide to walk you through the process, and the sections that follow have details on the configuration tasks.

Policy Controller Configuration Checklist

This section explains how to configure a Policy Controller implementation. This section assumed that you have installed Policy Controller as described in *Service Broker Installation Guide* and created a domain for it as described in "Creating and Starting a Domain and Managed Server for Policy Controller".

Configuring a Policy Controller implementation involves these tasks:

- Configuring Data Storage for Policy Controller

  Use this task to configure the Profile Store to store subscriber data for Policy Controller to use.
- Configuring the SSU Diameter and Specify AF and PCEF Servers as Peers

  Use this task to connect Policy Controller to AFs and PCEFs. If your OCS is Diameter-based there is some duplication between these tasks and configuring the OCS.

- Configuring the SSU Diameter to Route PCEF and AF Service Data Flow
- Configuring Policy Controller System Parameters

  Use this task to configure the Policy Controller global parameters, timers, and any event triggers that your implementation requires.
- Configuring Data Storage for Policy Controller

  Use this step to configure the BDB or JDBC databases.
- Configuring Your PCEF Server

  Use this task to configure your PCEF Server to work with Policy Controller.
- Configuring Your AF Server

  Use this task to configure your AF server to work with a Policy Controller.
- Starting, and Configuring Policy Designer

There are additional configuration procedures in these chapters in this manual:

- "Configuring Subscriber Profile and Charging Information" describes how to configure online charging and subscriber profile databases.
- "Implementing Overload Protection" explains overload protection and describes how to set it up.
- "Adding Custom Diameter AVPs" explains how to add your own custom AVPs for use with Policy Controller.

Configuring Your Diameter Traffic

To configure Diameter traffic for Policy Controller, complete these tasks which are explained in the following sections:

- Define your AF servers and PCEF servers as Diameter peers. See "Configuring the SSU Diameter and Specify AF and PCEF Servers as Peers" for details.
- Diameter Routing Rules that direct all Gx service data flow from your PCEF to Policy Controller, and all Rx service data flow from Policy Controller to your AFs. See "Configuring the SSU Diameter to Route PCEF and AF Service Data Flow" for details.

Configuring the SSU Diameter and Specify AF and PCEF Servers as Peers

This section assumes that you know details for of each of your PCEF and AF servers (host name, address, port no, protocol). The *Service Broker Signaling Server Units Configuration Guide* has details on the configuring the SSU Diameter. This section provides basic instructions.

Note: You will also configure your online charging system and Service Profile Repository as Diameter peers when you configure them. See "Configuring Subscriber Profile and Charging Information" for details.

To configure the SSU Diameter for Policy Controller:

1. Start the Administration Console.

   For details, see *Oracle Communications Service Broker System Administrator's Guide*.

2. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **SSU Diameter, DIAMETER**, then **Diameter Configuration**.

   The **General** subtab appears.

3. Click Lock and Edit
4. Configure these fields as explained in the discussion on configuring the Diameter Signaling Server Unit in the *Service Broker Signaling Server Units Configuration Guide*.
5. Click the **Peers** subtab.
6. Click **Add** to your AF or PCEF server as a peer.

   The **New** popup screen appears.

7. Enter a name for the new SSU and click **Apply**.
8. Select the new peer to configure it.
9. Select **True** from the **Allow Dynamic Peers** menu.
10. Enter a value for **Peer Retry Delay** (in seconds) and click **Apply**.
11. Scroll down in the **Peers** pane and click **New**.

    The **New** popup screen appears.

12. In the **New** popup screen, enter your AF or PCEF server identifying data:
    - **Host** - The host name of the server.
    - **Address** - the IP address of the server.
    - **Port** - A server port number to use on your PCEF.
    - **Protocol** - A service data flow protocol to use.
    - **Watchdog Enabled** - the **True**/**False** choices enable/disable the watchdog timer.
13. Click **Apply**.
14. Repeat steps 6 through 13 for each of your PCEF and AF servers.
15. Click **Commit** to save your changes.

Configuring the SSU Diameter to Route PCEF and AF Service Data Flow

The Diameter SSU routing rules specify how the PCEF and AF server service data flow is routed through Policy Controller. Use the following steps to create these routing rules:

1. Start the Administration Console.

   For details, see *Oracle Communications Service Broker System Administrator's Guide*.

2. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **SSU Diameter, SSU Diameter, Routing**, then **Incoming Routing Rules**.
3. Click **Lock and Edit**.
4. Click **Add** to create an Incoming Routing Rule for Gx traffic:

   The **New** popup window appears.

5. Enter a name for the new routing rule and click **Apply**.

   The new routing rule appears and is highlighted.

6. Edit these **Incoming Routing Rules** fields as necessary:
   - **Name:** Change if necessary.
   - **Priority:** Leave the **0** default priority.
   - **Module Instance:** Enter **ssu:ocsb/pcrf**.
7. Click **Apply**.
8. Click **Add** to create the Gx traffic routing criteria.

   The **New** popup window appears.
9. Enter a name for the new routing rule and click **Apply**.

   The new routing rule appears and is highlighted.
10. Enter the routing rule criteria for Gx traffic:
    - **Name:** Enter an informal name for the criteria.
    - **Attribute:** Select **APPLICATION_ID** from the dropdown list.
    - **Value:** Enter **16777238** to specify Gx service data flow.

      This value is the default for Gx traffic. Other values are also supported.
11. Click **Apply**.
12. Click **Add** to create an Incoming Routing Rule for Rx traffic.

    The **New** popup window appears.
13. Enter a name for the new routing rule and click **Apply**.

    The new routing rule appears and is highlighted.
14. Enter the routing rule criteria for Rx traffic.
    - **Name:** Change if necessary.
    - **Priority:** Leave the **0** default priority.
    - **Module Instance:** Enter **ssu:ocsb/pcrf**.
15. Click the **Incoming Routing Rule** subtab to add Rx traffic routing criteria.
16. Click **New**.

    The **New** routing popup window appears.
17. Enter the routing rule criteria for the Rx traffic:
    - **Name:** Change if necessary.
    - **Attribute:** Select **APPLICATION_ID** from the dropdown list.
    - **Value:** Enter **16777236** to specify Rx service data flow.
18. Click **Apply** to save your changes.
19. Click **Commit** to save your changes.

Configuring Policy Controller System Parameters

This section lists the Policy Controller global parameters that you must approve or change before using Policy Controller.

This section assumes that you have acquired and configured a rating engine and PCEF, and know the values you use to indicate rating groups and PCC service identifiers.

Configuring Policy Controller Global Parameters

This section lists Policy Controller parameters that you set by using the Service Broker Administration Console.

The default Policy Controller global parameters are listed in Table 2–3.

Table 2–3 Default Policy Controller Global Parameter Settings

Global Parameter	Default Value	Data Type	Description
Revalidation Time	14400000 (4 hours)	Milliseconds	The maximum time before your PCEF should trigger Policy Controller. This parameter also subscribes the **Revalidation_Timeout** trigger (See "About Event Triggers" for details). A value of **0** specifies no time limit.
Events to subscribe	See *Oracle Communications Service Broker Policy Controller Protocol Implementation Conformance Statement* for details for the list of default values.	Comma-separated list of integers	Specifies the list of 3GPP 29.211 event triggers to use. This field accepts the integer values that represent the Event-Trigger AVP. See the 3GPP 29.211 specification for a complete list. See "Using an Event Trigger to Change a PCC Profile" for a discussion of event triggers and "Add any Vendor-specific Values for Event-Trigger to Use as Event Triggers" for instructions on how add your own.
Primary Online Charging System address	None	URL	The primary online charging system address. Your PCEF attempts to send charging traffic to this server for charging first.
Secondary Online Charging System address	None	URL	The secondary online charging server address. Used if the primary server is unavailable.
Primary Offline Charging System Address	None	URI	The primary online collection server address. Policy Controller attempts to send collection traffic to this server for charging first.
Secondary Offline Charging System Address	None	URL	The secondary online collection server address. Used if the primary collection server is unavailable.
Default Online Charging Method	None	Boolean	Select **TRUE** to allow online charging; **FALSE** to disallow it. No entry directs Policy Controller to use your PCEF's default charging method.
Default Offline Charging method	None	Boolean	Select **TRUE** to allow offline charging; **FALSE** to disallow it. No entry directs Policy Controller to use your PCEF's default charging method
Install Default Plan	True	Boolean	Whether to automatically install and use the default PCC profile if no other plan applies.

Change these settings as necessary using the PCRF **Global Parameters** tab of the Service Broker Administration Console:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Navigate to **PCRF**, **OCSB**, **Execution Blocks**, then **System Parameters**.
3. Click the **PCRF System Parameters** tab, then the **Global Parameters** subtab.
4. Click **Edit**.
5. Enter the required new values in the fields displayed. See Table 2–3 for details on the Global parameter settings.
6. Click **Apply**.
7. Click **Commit**.

Add any Vendor-specific Values for Event-Trigger to Use as Event Triggers

Event-Trigger is a Gx AVPs that directs Policy Controller to reinterpret PCC and ADC profiles when received in a session. See "Using Vendor-specific and Default Gx Event Triggers to Reinterpret Rules" for details on how to use Event-Trigger in Policy Controller rules.

See "Supported Gx Event-Trigger Event Values" for the list of the default event triggers supported by Policy Controller. The instructions in this section explain how to add up to ten additional vendor-specific Gx AVPs to use as event triggers. For example, if your PCEF implementation uses custom values for the Event-Trigger AVP you could add them as event triggers here.

To add vendor-specific AVPs to the event trigger list the Policy Controller uses:

1. Start the domain and managed server.
2. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

3. Navigate to **PCRF**, **OCSB**, **Execution Blocks**, then **System Parameters**.
4. Click the **PCRF System Parameters** tab, then the **Oracle Event Triggers** subtab.
5. Click **Edit**.
6. Enter values in the unused OCPC_GENERIC_EVENT fields.
7. Click **Apply**.
8. Click **OK** to make your changes take effect.

Configuring the Policy Controller Session Guard Timers

Policy Controller uses Gx and Rx session guard timers (Tcc) and closing guard timers (Tsc) protect Policy Controller service data flow session validity by avoiding dangling sessions. In the event that any part of the PCEF implementation is not working correctly, the Gx and Rx timers ensure that a session is correctly terminated. The default settings work for a test and evaluation system and may also work for a production implementation. Configure them to fit your implementation's requirements.

You configure these timers independently to prevent problems caused by missing messages expected from network nodes, which orphan/dangling sessions. The Gx and Rx Session Guard timers set a time limit for expected session messages. If the Gx or Rx Session Guard timer limit runs out, the appropriate Gx or Rx Session Closing Guard

Timer starts and attempts to close the session cleanly. If it cannot close the session cleanly it closes out the session after the time limit expires.

Table 2–4 lists the timers that control and protect Rx and Gx traffic.

To change the Policy Controller Rx and Gx default timer values:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Navigate to **PCRF**, **OCSB**, **Execution Blocks**, then **System Parameters**.

3. Click the **PCRF System Parameters** tab, then the **Timers** subtab.

4. Click **Lock & Edit**.

5. Enter new parameters in the parameter text fields. See Table 2–4 for details on the individual timers.

Table 2–4 RX and Gx Timer Names, Default Values, and Descriptions.

Timer	Alternate Timer Name	Default Value (ms)	Description
Gx Session Guard Timer	Gx Tcc	**36000000**	Guards each Gx session by setting a time limit for pending Gx messages. This timer is started each time a Gx request (CCA-Initial) arrives at the Policy Controller, and is canceled when the last response (CCA-Terminal) is sent. Each CCA-Update resets this timer.
Gx Session Closing Guard Timer	Gx Tsc	**5000**	Sets a time limit for the Gx session termination process. If the Gx Session Guard Timer expires, Policy Control starts this timer and sends an RAR with a Session-Release-Cause of **UNSPECIFIED_REASON**. If an answering RAA arrives, Policy Controller waits for a CCR-Terminate, and the responds with CCA-Terminate. In any case, the session is closed when the time limit is reached.
Rx Session Guard Timer	Rx Tcc	**36000000**	Guards each Rx session by setting a time limit for pending Rx messages. This timer starts when the first AAA message is sent and canceled when the STA is sent. This timer is reset each time an AAA/AAR combination is sent and received.
Rx Session Closing Guard Timer	Rx Tsc	**5000**	Sets a time limit for the Rx session termination process. If the Rx Session Guard Timer expires, Policy Controller send an ASR message with an Abort Cause of **INSUFFICIENT_SERVER_RESOURCES**. If an ASA message is returned, Policy Controller waits for the STR message from the network node, and then responds with an STA and cleans up the session. If no ASA is received before this timer expires the session is closed.

6. Click **Apply**.

7. Click **Commit** to make your changes take effect.

About Execution Blocks

The **Default Execution Blocks** tab (under the **PCRF** tab of the Administration Console) is reserved for Oracle use.

Configuring Data Storage for Policy Controller

Policy Controller requires that you set up persistent data storage using the Data Storage feature. For information on setting up data storage, see the discussion on configuring data storage in *Oracle Communications Service Broker Installation Guide*.

Configuring Your PCEF Server

This section explains the steps necessary to make your PCEF server work with Policy Controller.

- Set up a Diameter link between PCEF and Policy Controller. See "Configuring the SSU Diameter to Route PCEF and AF Service Data Flow" for details if you haven't already.
- Configure your PCEF server to use the Diameter Gx messages listed in the *Policy Controller PICS* document. See your PCEF product documentation for details.
- If your PCEF uses AVPs that are not in one of the supported Diameter interfaces, you must add them to the Service Broker stack manually. See "Adding Custom Diameter AVPs" for details on adding custom Diameter AVPs to Service Broker.

Configuring Your AF Server

You must set up a Diameter link between Policy Controller and any AFs that you use. You set up the Policy Controller end of the link using the SSU Diameter. See "Configuring the SSU Diameter and Specify AF and PCEF Servers as Peers" for details if you haven't already. See your AF product documentation for details on setting up the AF end of the link, and any other setup steps necessary.

(Optional) Configuring Policy Controller to Send SMS Messages

Configuring Policy Controller to sent SMS messages includes these configuration tasks:

- Add an SMSC. Sending SMS messages from rules requires that you obtain a Short Message Service Center (SMSC) and configure the Service Broker SMPP SSU to use it. For details see the discussion on configuring an SMPP Signaling Server Unit in the *Service Broker Signaling Server Units Configuration Guide*.
- If necessary, change the SMPP data coding (language) to one appropriate for your locale. The SMPP data coding specifies the list of characters allowed in an SMS message. The allowed values are listed in the data_coding section of the Short Message Peer-to-Peer Protocol Specification 3.4.

  The default Policy Controller data coding is "SMSC Default Alphabet" (GSM 03.38 7-bit ASCII alphabet; integer value: 0). This data coding value applies to all SMSs sent from your Policy Controller implementation.

  To change the SMPP encoding use in SMS messages;

  1. Start the Administration Console.

     For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Click **Lock and Edit.**
3. Navigate to **PCRF**, **OCSB**, **Execution blocks**, **System Parameters**, then **PCRF System Parameters**.
4. Select the **Global Parameters** Tab.
5. Change the value in the **SMPP Data Coding** field to a different ISO/IEC 8859 value.
6. Click **Apply**.

(Optional) Configuring Custom Event Triggers

If your implementation requires support for nonstandard Diameter AVPs, add them now. For example if your PCEF or AF use nonstandard AVPs, you must at least add them to the Service Broker Diameter stack so that they do not cause error messages when they arrive. You can also use them as triggers for actions in Policy Designer rules. See "Adding Custom Diameter AVPs" details on adding custom AVPs.

(Optional) Configuring the Operation Timeout Setting

This timer sets the amount of time (in milliseconds) that Policy Controller waits for synchronous traffic responses sent by Policy Controller and returned by the SPR you use. The default value of 500 milliseconds is sufficiently large that only the slowest network connections, or a pressing need to receive responses from absolutely *all* messages may require a higher value.

Indications that you must change this setting are frequent error messages like these:

```
2012-11-08 06:07:45,201 ERROR wm-diameter_inbound-1
SynchronousUserProfileServiceImpl - Operation timed out.1352347664698
2012-11-08 06:07:45,202 ERROR wm-diameter_inbound-1 PcrfHash - Failed to retrieve
subscriber profile for subscriber: 622|END_USER_E164
```

Changing the Operation Timeout Settings

To change the operation timeout settings:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.
2. Navigate to **PCRF**, **OCSB**, **Domain Management**, **Subscriber Store**, then **Core Service**.

   The **Core Service** subtab appears:
3. Enter a new value in the **Operation Timeout** box.

   The default setting is 500 milliseconds.
4. Click **Apply**.

(Optional) Supporting Explicit Rule Removal

In the Policy Controller 6.0 release this product did not use the implicit profile removal strategy that it does now. The 6.0 release used *explicit* rule removal. that is, your rules needed to specifically remove any profiles that you did not want applied using **Remove_PCCProfile** actions. Explicit rule removal is generally seen as less efficient

and more error prone than the implicit rule removal, but you can use it if your implementation requires it.

See "Understanding How Policy Controller Makes Policies Take Effect" for details on the explicit rule removal strategy that Policy Controller now uses.

> **Note:** Only the 6.0 rule *strategy* (explicit rule removal) is supported. Using 6.0 *rules* with this release is not supported.

To use rules with the explicit rule removal strategy:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Click **Lock & Edit**.
3. Navigate to **OCSB**, **Execution Blocks**, **System Parameters**, **PCRF System Parameters**, then **Global Parameters**.
4. Set the **Disable Implicit Rule Actions** parameter to **True**.

   > **Note:** Do not make any other changes in the Execution Blocks area. The other settings are reserved for Oracle use.

5. Click **Apply**.
6. Click **Commit**.

3

Configuring Subscriber Profile and Charging Information

This chapter explains how to configure Oracle Communications Service Broker Policy Controller (Policy Controller) to use Subscriber Profile Repository (SPR) and online charging system (OCS) repositories to supply subscriber information for your Policy Controller rules to use.

About the SPR/OCSs Available for Policy Controller

Policy Controller is designed to use individual subscriber data to decide which services and QoS limits a subscriber is entitled to. This can be as simple as probing for the level of service a subscriber has purchased and applying the correct QoS limits. However, you can use any combination of SPR/OCS information in your rules as criteria to select services, applications, and QoS limits for a subscriber. More importantly, you can take advantage of the dynamic nature of the supported SPRs/OCSs and have Policy Controller reinterpret rules and change QoS limits and/or redirect service data flow based on updated information.

Policy Controller does not provision or maintain SPR/OCS data; the SPR/OCS you choose is responsible for doing that. Policy Controller only reads and uses it, and then deletes the local copy the session ends. Policy Controller obtains information from SPRs/OCSs in two ways. First it requests subscriber profile information from the SPR/OCS as soon as a subscriber opens a session. Second, Policy Controller also accepts notifications from your OCS/SPR updating the subscriber's profile during the session, and act on it if necessary. The local Subscriber store is the only SPR/OCS that does not support notifications. Your SPR/OCS product documentation explains how to configure and provision it with subscriber information, and set information on how to set up, provision, and use notifications to send subscriber data to Policy Controller.

Policy Controller can connect to and use subscriber data from any one of these combination SPR/OCSs. Select one and configure it using the instructions in the section listed:

- The Local Subscriber Store provided with Policy Controller. Use this option if you do not already have an SPR/OCS, and your implementation does not require one of the other options. See "Using the Local Subscriber Store as an SPR/OCS" for details.

  The local Subscriber Store contains both SPR and OCS data, and is available for Policy Controller information requests. However, the way it causes Policy Controller to reinterpret rules is different. The other SPR/OCSs use a notification mechanism. The local Subscriber Store reinterprets rules each time you use the **updateSubscriber** API operation to update a subscriber profile.

- The Oracle Communications Billing and Revenue Management (BRM) SPR/OCS features. See "Using Oracle Billing and Revenue Management as an SPR/OCS" for details.
- A third-party SPR/OCS that uses Diameter-based communication. See "Using a Third-Party SPR with Diameter Connectivity" for details.
- A third-party SPR/OCS that uses Web-based (SOAP) communication. See "Using a Third-Party SPR/OCS with Web-Based Connectivity" for details.

How Policy Controller Obtains Subscriber Information from Your SPR/OCS

The instructions for configuring your SPR/OCS to connect to and work with Policy Controller are in this chapter. Once configured, Policy Controller automatically probes for subscriber profile information from your SPR/OCS when a session is started. Once the session is started Policy Controller can accept notifications of new subscriber information from the SPR and reinterpret Policy Controller rules as needed. Once connected with a Policy Controller SSU, your SPR/OCS and Policy Controller can pass information back and forth freely. using the Diameter Sy/Sp reference points.

Note: Policy Controller uses the Diameter SP specification to communicate with your SPR. However details of an Sp implementation is left to the implementor. Policy Controller uses a modified form of the Diameter Sh commands and AVPs for its Sp implementation. The supported Sh specification is *Policy and Charging Control: Spending Limit Reporting over Sy reference point 3GPP TS 29.219 V11.0.0 (2012-03)*.

See *Oracle Communications Service Broker Policy Controller Protocol Implementation Conformance Statement* for details on the commands and AVPs supported.

About the Subscriber Store Data Model

All of the SPR/OCSs that Policy Controller supports use the data model explained in "Subscriber Profile Data Model". The local subscriber store uses this data model, so no further mapping is required. BRM uses the mapping shown in "PCP Profile Provider Data Mapping" to use this data model with its database. The Diameter and Web-based SPR/OCSs also require mapping to their internal structures which are defined by the WSDL files provided with Policy Controller. You connect to them during the configuration procedures for these SPR/OCSs.

Using the Local Subscriber Store as an SPR/OCS

The local Policy Controller SPR is ready to configure after Policy Controller is installed. See *Service Broker Subscriber Store User's Guide* for instructions on how to configure the local Subscriber Store, provision it with data for the Policy Controller to use, and configure an Sp reference point to communicate with.

The *Service Broker Subscriber Store User's Guide* also includes information on the Subscriber Store provisioning API that you use to provision the load SPR.

You can extend subscriber profiles with any number of key-value pairs that your Policy Controller implementation requires. For details on using the Subscriber Store API to add key/value pairs, see *Service Broker Subscriber Store User's Guide*.

Once configured, the local Subscriber Store causes Policy Controller to reinterpret its rules each time you use the **updateScriber** API operation to change a subscriber profile.

Using Oracle Billing and Revenue Management as an SPR/OCS

If you implement both Policy Controller and BRM you will probably also use BRM as your SPR/OCS.

You provision and maintain accurate subscriber profiles using the BRM tools, and configure BRM to send subscriber updates Policy Controller as necessary.

About BRM Subscriber Profiles

Subscriber profiles in the SPR/OCS include:

- The service profile.
- An extensible subscriber lifecycle that you use to define subscriber states and state transitions.
- Usage counter and thresholds information.

Using the BRM as an SPR/OCS

The Policy Controller **PCP Profile Provider** reads BRM subscriber data by using the PCP Signaling Server Unit (SSU) and populates the Subscriber Store with this information. BRM remains the system of record for subscriber data when using the PCP Profile Adapter.

The Subscriber Store updates BRM subscriber data by listening for Oracle Advanced Queuing (AQ) notification messages generated by the BRM database. When Policy Controller receives a message it checks if any of the changes apply to users with active sessions. If any such users are found the PCP Profile Provider initiates a refresh of their Subscriber Store data for affected users.

See "Subscriber Profile Data Model", for additional information on the attributes retrieved and mapped from BRM into the Subscriber Store.

Note: BRM only supports the **active** and **inactive** subscriber account lifecycles. If you use BRM as an SPR/OCS, you should only use those values.

See "Configuring the BRM Subscriber Store", for information on configuring the BRM Subscriber Store.

See "PCP Profile Provider Data Mapping", for information on field mapping between BRM and the Subscriber Store.

Configuring the BRM Subscriber Store

To configure Policy Controller to use the BRM database as the Subscriber Store source:

1. Verify that the PCP persistence bundle is installed and active. See "Configuring the PCP Profile Provider", for more information.

2. Create a connection pool for the BRM database. See the information on creating database connections in the *Service Broker Installation Guide* for more information on setting up the BRM JDBC connection.
3. Configure the Signal Server Unit (SSU) connection to BRM. See the chapter on configuring a PCP Signaling Service Unit in *Service Broker Signaling Server Units Configuration Guide*, for more information.
4. Configure the Subscriber Store Provider containing the queue from where BRM-generated notifications for updates to subscriber profile data are published. See "Configuring a Subscriber Store Provider", for more information.
5. Populate and maintain the Subscriber Store with user information.

Configuring the PCP Profile Provider

To configure the Subscriber Store to use the PCP Profile Provider:

1. Remove the existing local Subscriber Store persistence bundle from the domain. By default, it is:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.store.provider
2. Install and start the PCP Profile Provider bundle with a level of 260:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.pcp.jar
3. Make sure the run level for the package matches that of the following package:

   oracle.ocsb.app.rcc.service.subscriber_store.core
4. Configure the database schema for the Subscriber Store using the following SQL script:

   *Oracle_home***/ocsb/admin_console/scripts/database/subscriber_store.sql**

   To use the script to configure your database schema, run the script using a SQL client tool, such as sqlplus, or interface provided by your database management system.
5. If the managed servers in the domain are not already configured to connect to the database, configure the connections in the **Data Store** node under **Domain Management**.

   For the Subscriber Store database, the connection pool name value in the JDBC configuration should be the name of your BRM database that you set in the "Configuring a Subscriber Store Provider" section. For example, **oracle_driver**.

   See *Service Broker Installation Guide* for more information about configuring data storage.
6. Configure the PCP Signaling Server Units (SSU) to connect to BRM.

   See the chapter on configuring PCP signal server units in *Service Broker Signaling Server Units Configuration Guide*.

Configuring a Subscriber Store Provider

Configure the Subscriber Store Provider indicating the connection pool and queue on which to listen for BRM generated Oracle AQ notifications.

1. Start the Administration Console.

   For details see *Service Broker System Administrator's Guide*

2. If the Administration Console is not already in offline mode, click the **Switch to OFFLINE mode** icon at the top of the page.
3. Click **Lock & Edit**.
4. Expand the **OCSB** node and then **Domain Management**.
5. Expand the **Subscriber Store** node.
6. Click the **Providers** node.
7. In the **BRM** tab, add the Subscriber Store Provider information as a new **Notifications** entry by clicking **New** and providing the following information:
   a. Enter a unique **Name** for the provider.
   b. In the **Connection Pool Reference** field, enter the name of the JDBC connection pool created for the BRM database. This value must be identical to the connection pool value.
   c. In the **Queue Owner** field enter the BRM database user with access to the notification queue.
   d. In the **Queue Name** field, enter the name of the notification queue as defined in the BRM database.
   e. In the **Number of Sessions field**, enter the number of connections to open to the Provider. This value cannot be larger than the connection pool size.
8. Click **OK** to save the configuration.

   The new provider instance appears in the list of Notifications.
9. Click the **Commit** icon.

Subscriber Profile Data Model

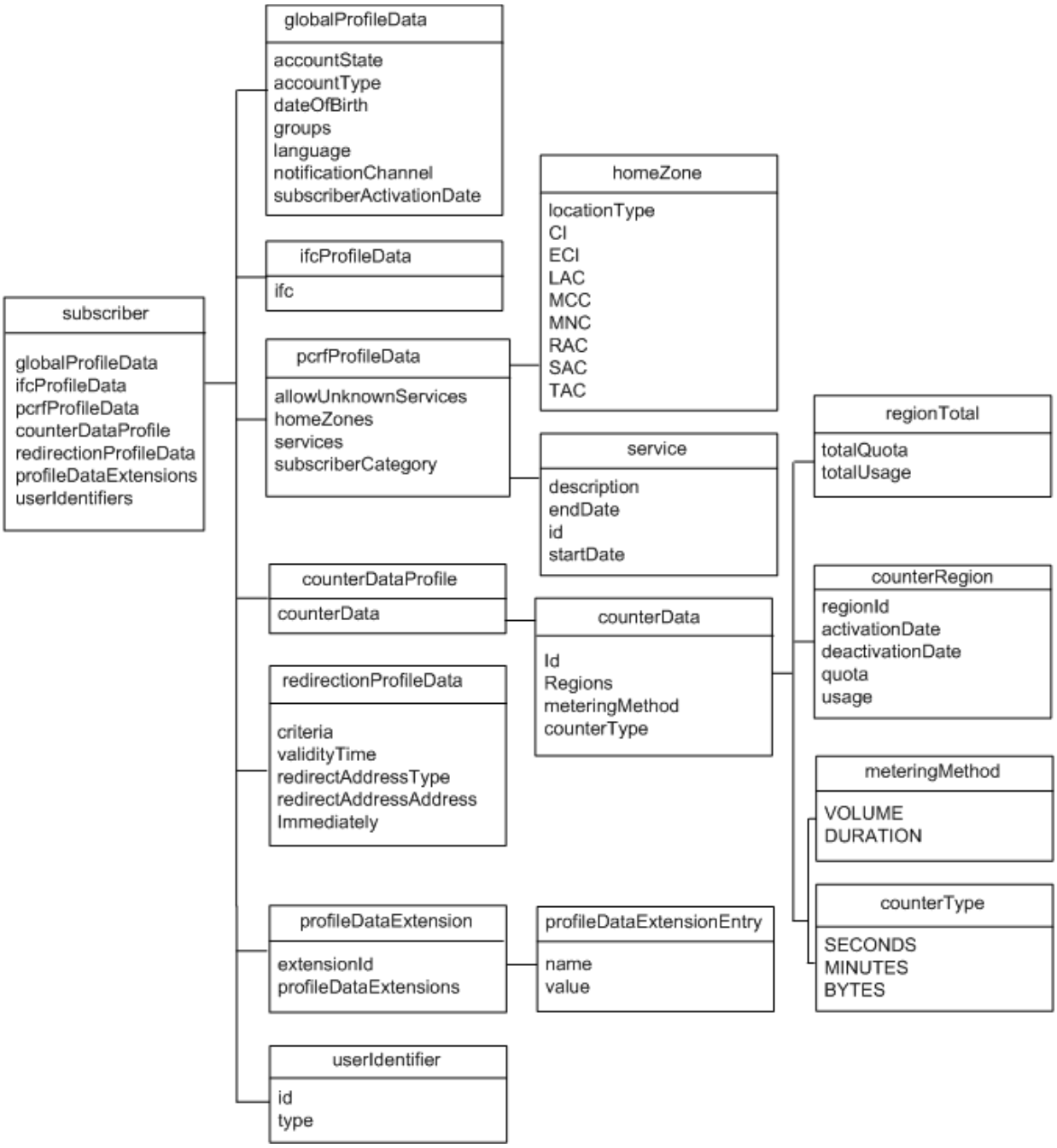
The subscriber profile data model defines the elements of a subscriber profile. The model includes general information for the subscriber along with feature-specific information, such as Policy Controller data. Figure 3–1 illustrates the subscriber profile data model.

The subscriber profile data model is defined by an XML Schema file. You can access the schema at the following default location:

http://*host***:***port***/soap/SubscriberProvisioning?xsd=1**

where *host* is the host name or IP address and *port* is the server port number you specified as the server address.

Figure 3–1 Subscriber Profile Data Model



As shown in Figure 3–1, the top-level data elements that compose the subscriber profile are:

- globalProfileData Element
- pcrfProfileData Element
- counterProfileData Element
- profileDataExtension Element
- userIdentifier Element

The following sections provide information on the data elements of the subscriber profile.

globalProfileData Element

The **globalProfileData** element defines general properties for a subscriber. This element contains the following elements:

- **AccountState**: The account lifecycle state that indicates the status of the account, such as active or inactive.
- **AccountType**: The account type from the options prepaid, postpaid, or hybrid. Hybrid accounts can use both prepaid and postpaid charging methods.
- **DateOfBirth**: The birth date of the subscriber. The date is in *YYYYMMDD* format, **19910128**.
- **Groups**: Group identifier for the subscriber.
- **Language**: The language to use for the subscriber. AFs that interact with subscribers can use this attribute to select a response or adapt a message for the subscriber.
- **NotificationChannel**: The notification channel used to send notifications to the user.
- **SubscriberActivationDate**: The date, in *YYYYMMDD* format, when the subscriber account was first activated, such as **20111231**.

pcrfProfileData Element

The **PcrfProfileData** element contains subscriber data used by the Policy Controller. **PcrfProfileData** includes the following elements:

- **AllowUnknownServices**: Whether the subscriber is permitted to access services that are not specifically allocated to that subscriber, as indicated by the **services** field.
- **HomeZone**: The home, non-roaming network zone for the subscriber. Policy Controller administrators can apply charging rates or service access decisions based on whether the subscriber is using the network from their home zone.

  The **HomeZone** element is made up of a **LocationType**, which identifies the protocol type in which the location information is represented, such as IEEE-802.11a or 3GPP-GERAN. Depending on the location type, it also includes elements that identify the location as specified for that type.

  Possible values for **LocationType** can be: **CGI**, **SAI**, **RAI**, **TAI**, **ECGI**, **TAI_ECGI**. Depending on the value of **LocationType**, the following additional elements may be present:

  - **CI**
  - **ECI**
  - **LAC**
  - **MCC**
  - **MNC**
  - **RAC**
  - **SAC**
  - **TAC**

For example, if the location type is **CGI**, then the location-type specific parameters that should be set are: **CI**, **LAC**, **MCC**, and **MNC**.

For more information on the location identification protocol, see the 3GPP specifications: TS 29.060, TS 29.061, and TS 29.274.

- **Services** contains one or more **Service** elements. The **Service** defines the services this subscriber can access. Each service has the following fields:
  - **description**: A description of the service.
  - **endDate**: The date, in YYYYMMDD format, on which the service availability ends for an individual subscriber. Along with the **startDate** value, this value enables you to make services available to subscribers within a specific date range.
  - **id**: The identity of the Policy Controller application described by this **service** element. See *Service Broker Policy Controller Implementation Guide* for more information.
  - **startDate**: The date, in YYYYMMDD format, on which service availability begins. Along with the **endDate** value, this value enables you to make services available to subscribers within a specific date range.
- **SubscriberCategory**: An application-defined service level for a given subscriber, such as gold, silver, and bronze. The Policy Controller rules determine how a category dictates the level of service access for the subscriber.

counterProfileData Element

The **counterProfileData** element contains subscriber usage data represented as one or more **counterData** elements. Each **counterData** element identifies a resource or usage balance and its associated **regions** defined by a set of consecutive value ranges. See "About Counter Regions", for an example of how **regions** can be used in the Subscriber Store.

CounterData contains the following:

- **Id**: A unique string identifier for the counter.
- **Region(s)**: One or more consecutive **CounterRegion** resource balance value ranges used to categorize a subscriber's usage status. The value of a subscriber's resource counter determines the region that the subscriber is currently in.

  The **CounterRegion** element consists of the following:
  - **regionID**: A unique string identifier for the region.
  - **activationDate**: The date on which the region became active.
  - **deactivationDate**: The date on which the region will become inactive.
  - **usage**: The currently used amount for the region if supported by the underlying billing system. A value of **-1** indicates that the usage for this region is not available.
  - **quota**: The available quota for the region if this region has a quota. A null value indicates an unlimited quota. A value of **-1** indicates that the quota for this region is not available
- **MeteringMethod**: Indicates a how counter tracks subscriber usage. The following methods are supported:
  - **VOLUME**: The amount of resource usage. For example, bytes.

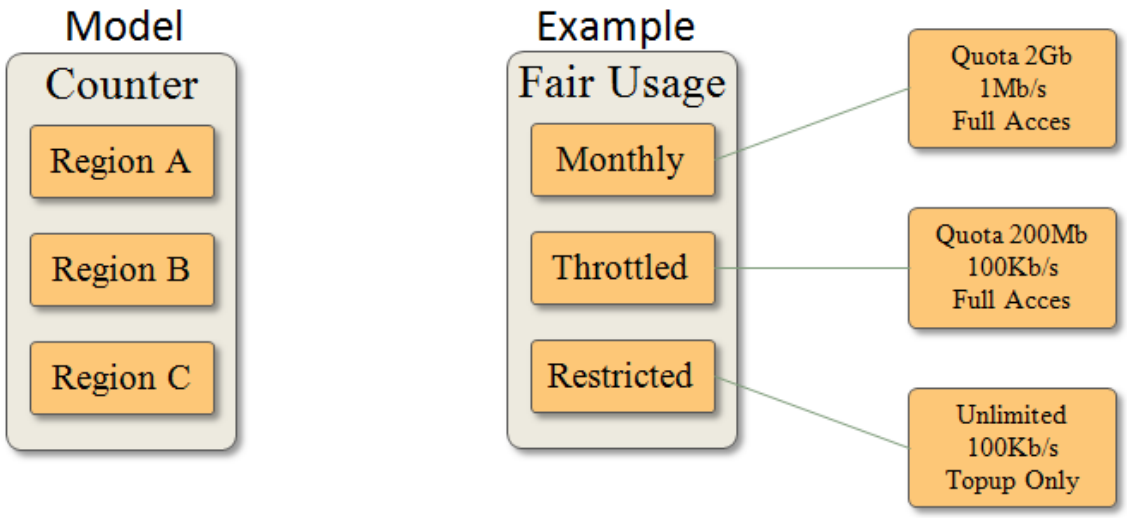
  - **DURATION**: The length of time a resource is used. For, example, seconds or minutes.

- **CounterType**: Indicates the type of metric used to track subscriber usage. The following types are supported:
  - **SECONDS**
  - **MINUTES**
  - **BYTES**
- **RegionTotal**:
  - **totalQuota**: The total amount of resource quota available in a region.
  - **totalUsage**: A subscriber's total resource usage within a region.

About Counter Regions The following sample scenario provides an example of how the Subscriber Store uses counter regions are used.

In this example, counter regions model a fair usage scenario for a data subscriber. The subscriber's service provider implements a data plan including reduced (throttled) data speeds triggered by the subscriber's total data usage across the thresholds as shown in Figure 3–2.

Figure 3–2 Counter Regions Example



As the subscriber's usage accumulates in the fair usage counter, he moves through the "Monthly", "Throttled", and "Restricted" counter regions. The subscriber exists in only one region at a time.

Policy Controller can use the subscriber's current region to enforce the correct data throttling and access limitations. The bandwidth and access values in the example are not part of the counter data model, but rather part of the rules enforced based on the data model.

profileDataExtension Element

Policy Controller and AFs use the **ProfileDataExtension** element to store custom data elements in the profile. For example, the Policy Controller itself can make policy decisions based on dynamically defined subscriber attributes.

The **ExtensionId** value identifies the component using the data extension. For the Policy Controller, for example, the **ExtensionId** value is **pcrf**. Each data extension entry consists of one or more name-value pairs, as specified by the component that uses it.

userIdentifier Element

The **UserIdentifier** element contains the unique identifier for an end user on a particular network. An end user can belong to multiple networks, for instance, if they own multiple devices used to access different networks. Therefore, a subscriber can have multiple **UserIdentifier** elements.

Each **UserIdentifier** element consists of the identifier value and the type of the identifier. The type represents the system in which the ID belongs. It may have one of the following values:

- **END_USER_E164**: The subscriber's identity in ITU-T E.164 format, as defined in recommendations E164 and CE164.
- **END_USER_IMSI**: The subscriber's identity in International Mobile Subscriber Identity format, as defined in ITU-T recommendations E212 and CE212.
- **END_USER_SIP_URI**: The subscriber's identity in a SIP network.
- **END_USER_NAI**: The subscriber's identity in a mobile IP Network Address Identifier format.
- **END_USER_PRIVATE**: The subscriber's private identity in a credit-control server.
- **END_USER_GLOBAL_UID**: A unique global user ID generated by Policy Controller to identify subscribers internally. Policy Controller generates this ID automatically when you create a subscriber profile.

The value for each identifier type varies by the network type. For example, the identifier for the URI type should be in the form of a SIP URL, such as **sip:username@example**.

PCP Profile Provider Data Mapping

Policy Controller maps BRM subscriber profile data to Subscriber Store attributes used in determining a subscriber's state or service usage eligibility. For more information about BRM to Subscriber Store mapping, see the chapter on using the Subscriber Provisioning API in *Subscriber Store User's Guide*.

The following tables describe the mapping of Subscriber Store profile data fields to the BRM flist fields used in the BRM database.

Table 3–1 describes the **GlobalProfileData** fields mapping between the Subscriber Store and BRM database.

Table 3–1 GlobalProfileData Fields Mapping

GlobalProfileData Field	Type	Flist Field Name	Flist Field Type
AccountState	String	globalProfileData.accountState	String
NotificationChannel	String	globalProfileData.notificationChannel	String
Groups	List	globalProfileData.groups.[n]	N/A
AccountType	String	globalProfileData.accountType	String
SubscriberActivationDate	String	globalProfileData.subscriberActivationDate	String
DateOfBirth	String	globalProfileData.dateOfBirth	String

Table 3–2 describes the **PcrfProfileData** field mapping between the Subscriber Store and BRM database.

Table 3–2 PcrfProfileData Fields Mapping

PcrfProfileData Field	Type	Flist Field Name	Flist Field Type
SubscriberCategory	String	pcrfProfileData.subscriberCategory	String
HomeZones	List	pcrfProfileData.homeZone.[n]	N/A
Home.GeographicLocationType	Enum	pcrfProfileData.homeZone.[n].geographicalLocationType	String
Home.LocationData	Map	pcrfProfileData.homeZone.[n].locationData	String
Home.LocationField	Enum	pcrfProfileData.homeZone.[n].locationField	String
Services	List	pcrfProfileData.services.[n]	N/A
Service.id	String	pcrfProfileData.services.[n].id	String
Service.description	String	pcrfProfileData.services.[n].description	String
Service.startDate	String	pcrfProfileData.services.[n].startDate	String
Service.endDate	String	pcrfProfileData.services.[n].endDate	String
AllowUnknownServices	Boolean	pcrfProfileData.allowUnknownServices	String

Table 3–3 describes the **CounterProfileData** field mapping between the Subscriber Store and BRM database

Table 3–3 CounterProfileDat Fields Mapping

CounterProfileData Field	Type	Flist Field Name	Flist Field Type
Id	String	counterProfileData.id	String
CounterType	Enum	Mapped from FldGroupInfo	FldRumName
MeteringMethod	Enum	Derived from CounterType	FldRumName
Regions	List	counterProfileData.region.[n]	N/A
CounterRegion.regiodId	String	counterProfileData.region.[n].regionId	String
CounterRegion.value	Long	counterProfileData.region.[n].value	String
CounterRegion.activationDate	String	counterProfileData.region.[n].activationDate	String
CounterRegion.deactivationDate	String	counterProfileData.region.[n].deactivationDate	String

Table 3–4 describes the **ProfileDataExtension** field mapping between the Subscriber Store and BRM database

Table 3–4 ProfileDataExtension Fields Mapping

ProfileData Extension Field	Type	Flist Field Name	Flist Field Type
Id	String	profileDataExtension.[n].id	String
Data	Map	profileDataExtension.[n].data.[n]	N/A
Data.key	String	profileDataExtension.[n].data.[n].key	String
Data.value	String	profileDataExtension.[n].data.[n].value	String

Using a Third-Party SPR with Diameter Connectivity

If your implementation requires it, you can use a third-party SPR/OCS that uses Diameter-based connectivity and populate it with subscriber data for Policy Controller to use. The **Sy/Sp Profile Provider** reads subscriber data by using the Diameter Server Signaling Unit (SSU). The Diameter-based SPR/OCS remains the system of record for subscriber data when using the Sy/SP Profile Adapter.

Configuring a Diameter-based SPR/OCS requires a basic knowledge of the Diameter protocol and its connection and routing requirements. It will be helpful to you to read through the configuring a Diameter signaling server unit section in *Service Broker Signaling Server Units Configuration Guide* first.

To configure an OCS using Diameter-based connectivity:

1. Start the Administration Console.

   For details see *Service Broker System Administrator's Guide*

2. Click **Lock and Edit**.
3. Navigate to **OCSB**, **Domain Management**, then **Packages**.

   The **Bundles** tab appears.

4. Select existing local persistence bundle from the domain. By default, it is:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.store.provider

5. Click **Uninstall**.
6. Select the Web-based Sy Profile Provider bundle:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.spy.common.jar

7. Select **Start Leve**l and set the start level to **190**.
8. Click **Start**
9. Select the Web-based Sy Profile Provider bundle:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.spy.diameter

10. Select **Start Level** and set the start level to **190**.
11. Click **Start**
12. Click **Apply**.
13. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **SSU Web Services**, then **SSU Diameter.**
14. Select the **SSU Diameter**. node

    The SSU Diameter window appears. See configuring a Diameter signaling server unit section of *Service Broker Signaling Server Units Configuration Guide* for details on configuring the Diameter SSU. You must configure at least these items in the Diameter SSU:

    - An incoming routing rule from the **SSU Diameter, Routing**, **Incoming Routing Rules** window.
    - Diameter nodes from the **Diameter**, **Diameter Configuration**, **General** window.
    - A Diameter default route from the **Diameter**, **Diameter Configuration**, **Default Route** window.

- Set the OCS as a Diameter peer from the **Diameter**, **Diameter Configuration**, **Default Route** window.
- Set the other Diameter routes from the **Diameter**, **Diameter Configuration**, **Routes** window

15. Navigate to **Platform**, **OCSB**, **Processing Tier**, **Subscriber Store**, then **Provider**.

    The **Diameter Sp/Sy** window appears.

16. Enter the URL of your Diameter-based SPR/OCS server in the **Sp DestinationRealm** field.
17. Enter the host name of your Diameter-based SPR/OCS server in the **Sp DestinationHost** field.
18. Enter the URL of your Diameter-based SPR/OCS server in the **Sy DestinationRealm** field
19. Enter the host name of your Diameter-based SPR/OCS server in the **Sy DestinationHost** field.
20. Click **Apply**.
21. Navigate to **Platform**, **Domain Management**, **Data Store**, then **Persistent Store**.

    The **Persistent Store**s widow appears.

22. Create a new Persistent Store to cache SPR/OCS information. Create either a JDBC Store, or Berkeley DB Store, depending on the database you chose during installation.
23. Navigate to **PCRF**, **OCSB**, **Execution Blocks**, then **System Parameters**, PCRF System Parameters.
24. In the Global Parameters window, fill in these items:
    - **Primary Online Charging System address**. Enter the URL of your primary OCS.
    - **Secondary Online Charging System address**. The URL of an OCS to use if the primary OCS is offline.
    - **Default Online Charging Method**. Set this to **True** if online charging is your default charging method. Use **False** otherwise.
25. Click **Commit**.
26. Configure the Diameter-based SPR/OCS to send notifications. See your SPR/OCS product documentation for details.

Using a Third-Party SPR/OCS with Web-Based Connectivity

If your implementation requires it, you can use a third-party SPR/OCS that uses Web-based connectivity and populate it with subscriber data for Policy Controller to use. The **Sy/Sp Profile Provider** reads subscriber data by using the Web Services Signaling Unit (SSU) and populates the SPR/OCS with this information. The Web-based SPR/OCS remains the system of record for subscriber data when using the Sy/SP Profile Adapter.

To configure an OCS using Web-based connectivity:

1. Start the Administration Console.

   For details see *Service Broker System Administrator's Guide*

2. Click Lock and Edit.
3. Navigate to **OCSB**, **Domain Management**, then **Packages**.

   The **Bundles** tab appears.
4. Select the existing local persistence bundle from the domain. By default, it is:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.store.provider
5. Click **Uninstall**.
6. Select this Web-based Sy Profile Provider bundle:

   oracle.ocsb.app.rcc.service.subscriber_store.providers.spy.soap.provider
7. Select **Start Level** and set the start level to **190**.
8. Click **Start**
9. Select this Web-based Sy Profile Provider bundle:
   oracle.ocsb.app.rcc.service.subscriber_store.providers.spy.soap.ws
10. Select **Start Level** and set the start level to **190**.
11. Click **Start**
12. Click **Apply**.
13. Expand to **Platform**, **OCSB**, **Signaling Tier**, then **SSU Web Services**,.

    The SSU Web Services window appears. See configuring the Web services signaling server unit section of *Service Broker Signaling Server Units Configuration Guide* for details on configuring the Web Services SSU. You must configure at least these items in the Diameter SSU:

    - Incoming routing rules from the **General**, **Incoming Routing Rules** tab.
    - An outgoing routing rule from the **General**, **Outgoing Routing Rules** tab
    - HTTP server settings from the **HTTP** tab.
    - HTTP server access settings from the **HTTP**, **Network Access** tab.
    - HTTP client settings from the HTTP, **Client** tab.
    - SOAP access settings from the **SOAP** tab.
    - The SOAP URI path from the **Subscriber Provisioning** or **Balance Manager** tab.
    - The SOAP client parameters from the **Client** tab.
14. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **Subscriber Store**, then **Provider**.

    The **Web Service Provider** window appears.
15. In the **Configuration** tab edit the Sy Service Endpoint URL field

    Enter the URL of the Web server serving the OCS.
16. Click **Apply**.
17. Click **Commit**.
18. Configure the Web-based SPR/OCS to send notifications. See your SPR/OCS product documentation for details.

Confirming that Your Web-Based SPR/OCS is Connected

Once configured and started, your Web-based SPR/OCS can connect to the WSDL file using this syntax: *IP_address*:*port_number*/**SyClient**.

4

Monitoring Policy Controller Using Runtime MBeans

This chapter explains how to monitor Oracle Communications Service Broker Policy Controller (Policy Controller) using runtime MBeans.

About Monitoring Policy Controller

You can monitor how Policy Controller operates by receiving the following information:

- Statistics on messages and sessions that Policy Controller handles. See "Monitoring the Processing Domain" for more information.
- Status of the network entities with which signaling server units (SSUs) communicate. See "Monitoring the Signaling Domain" for more information.

Monitoring the Processing Domain

Policy Controller generates and collects a set of statistics for each processing managed server. You can use the statistics to monitor subscriber traffic. The statistics are derived from a variety of counters and gauges collected by MBean attributes. These statistics are available for you to view using a JConsole or another third-party JMX-compliant monitoring tool.

By default:

- Policy Controller MBean statistics counters and gauges are active.
- Statistics notification is enabled.
- The counters are reset to zero every 900 seconds.
- Monitoring notifications are sampled every 30 seconds to check whether the threshold has been crossed.

Each unified domain contains one set of statistics MBeans. In dual processing and signaling domains the statistics MBeans reside on the processing managed servers.

Monitoring the Policy Controller Interfaces

Using **PcrfMBean**, you can get counters of messages that Policy Controller sends and receives through the Gx and Rx interfaces.

The object name of this MBean is **oracle:Class=oracle.axia.api.platform.runtimembean.RuntimeStandardMBean,Type**

=PcrfCounter,Version=*MBean_Version*,**Location=***Managed_Server_Name*,**Name=pcrf,CountingMethod=***Counting_Method*

The following sections describe the counters and gauges that you can use to get statistics on messages that Policy Controller sends and receives through each of these interfaces.

Getting Statistics on the Gx Interface

Table 4–1 describes the counters and the gauge that PcrfMBean provides for the Gx interface.

Table 4–1 PcrfMBean Counters and Gauge for the Gx Interface

To Get Total Number of…	Use…
Successfully completed sessions	GxSuccessfullyCompletedSessions
Failed sessions	GxFailedSessions
CCAs sent with the Result-Code AVP less than 3000	GxCcaInitialSentSuccess
CCA-Initial messages sent with the Result-Code AVP equals to, or is greater than 3000	GxCcaInitialSentFailure
CCA-Update messages sent with the Result-Code AVP less than 3000	GxCcaUpdateSentSuccess
CCA-Update messages sent with the Result-Code AVP equals to, or is greater than 3000	GxCcaUpdateSentFailure
CCA-Terminate messages sent with the Result-Code AVP less than 3000	GxCcaTerminateSentSuccess
CCA-Terminate messages sent with the Result-Code AVP equals to, or is greater than 3000	GxCcaTerminateSentFailure
Incremented each time a Gx session timer (Tcc) fires.	GxTimedOutSessionsTcc
Incremented each time a Gx session closing timer (Tsc) fires.	GxTimedOutSessionsTsc
CCR-Initial message received	GxCcrInitialReceived
CCR-Update messages received	GxCcrUpdateReceived
CCR-Terminate messages received	GxCcrTerminateReceived
RAR messages sent	GxRarSent
RAA messages received with the Result-Code AVP less than 3000	GxRaaReceivedSuccess
RAA messages received with the Result-Code AVP equals to, or is greater than 3000	GxRaaReceivedFailure
CCR-Initial messages received and rejected due to overload	GxThrottledSessions
Currently active sessions (gauge)	GxActiveSession

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

Getting Statistics on the Rx Interface

Table 4–2 describes the counters and the gauge that **PcrfMBean** provides for the Rx interface.

Table 4–2 PcrfMBean Counters and Gauge for the Rx Interface

To Get Total Number of…	Use...
Initial AAR messages received and rejected due to overload	RxThrottledSessions
STA messages received with the Result-Code AVP less than 3000	RxSuccessfullyCompletedSessions
AAA or ASA messages sent with the Result-Code AVP equal to, or greater than 3000	RxFailedSessions
Timed Out Sessions (Tcc) messages received	RxTimedOutSessions
Initial AAR messages received	RxAarInitialReceived
Initial AAA messages sent with the Result-Code AVP less than 3000	RxAaaInitialSentSuccess
Initial AAA messages sent with the Result-Code AVP equals to, or greater than 3000	RxAaaInitialSentFailure
Subsequent AAA messages sent with the Result-Code AVP less than 3000	RxAaaSubsequentSentSuccess
Subsequent AAA messages sent with the Result-Code AVP equals to, or greater than 3000	RxAaaSubsequentSentFailure
ASR messages sent	RxAsrSent
ASA messages received with the Result-Code AVP less than 3000	RxAsaReceivedSuccess
RAA messages received with the Result-Code AVP equals to, or greater than 3000	RxAsaReceivedFailure
STR messages received	RxStrReceived
STA messages sent with the Result-Code AVP less than 3000	RxStaSentSuccess
STA messages sent with the Result-Code AVP equals to, or greater than 3000	RxStaSentFailure
Incremented each time an Rx session timer (Tcc) fires.	RxTimedOutSessionsTcc
Incremented each time an Rx session closing timer (Tsc) fires.	RxTimedOutSessionsTsc
Subsequent AAR messages received	RxAarSubsequentReceived
STR messages received	RxStrReceived

Table 4–2 (Cont.) PcrfMBean Counters and Gauge for the Rx Interface

To Get Total Number of…	Use...
ASR messages sent	RxAsrSent
Currently active sessions	RxActiveSession

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

Monitoring the Signaling Domain

Policy Controller might require the following SSUs:

- Diameter SSU. See "Checking the Status of Diameter Network Entities" for more information.
- PCP SSU. See "Checking the Status of PCP Network Entities" for more information.
- WS SSU. See "Checking the Status of Web Service Network Entities" for more information.
- SMPP SSU. See "Checking the Status of SMPP Network Entities" for more information.

Using runtime MBeans, you can check whether the network entity with which the SSU communicates is active.

Checking the Status of Diameter Network Entities

Using **NetworkEntityRuntimeMBean**, you can get the status of the Diameter network entity. Table 4–3 describes the attributes that **NetworkEntityRuntimeMBean** provides.

Table 4–3 NetworkEntityRuntimeMBean Attributes

Attribute	Description
getValue()	The attribute contains the address of the network entity in the URI format where with the colon character (:) is replaced with the underscore.
getStatus()	Specifies a network entity status: ■ 0 - Network entity is unavailable ■ 1 - Network entity is available ■ 2 - Status of the network entity is unknown

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

Checking the Status of PCP Network Entities

Using **PcpPaStatisticsMBean**, you can get the status of the BRM application. Table 4–4 describes the attribute that **PcpPaStatisticsMBean** provides.

Table 4–4 PCP SSU Monitoring Attribute

Attribute	Description
getPcpPaStatus()	Specifies the status of the BRM applicatikon. Possible values: ■ 0 - Inactive ■ 1 - Active

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

Checking the Status of Web Service Network Entities

Using **NetworkEntityRuntimeMBean**, you can get the status of the Web Services entity. Table 4–5 describes the attributes that **NetworkEntityRuntimeMBean** provides.

Table 4–5 NetworkEntityRuntimeMBean

Attribute	Description
getValue()	The attribute contains the address of the network entity in the URI format where with the colon character (:) is replaced with the underscore.
getStatus()	Specifies a network entity status: ■ 0 - Network entity is unavailable ■ 1 - Network entity is available ■ 2 - Status of the network entity is unknown

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

Checking the Status of SMPP Network Entities

Using **SmppAdapterMBean**, you can get the status of all SMPP connections. Table 4–6 describes the attribute that **SmppAdapterMBean** provides.

Table 4–6 SmppAdapterMBean Status Attribute

Attribute	Description
getlistSmscConnectionStatus[]	Specifies the status of all SMSC connections. Possible values: ■ 0 - Inactive ■ 1 - Active

Using **NetworkEntityRuntimeMBean**, you can get the status of the network entity. Table 4–7 describes the attributes that **NetworkEntityRuntimeMBean** provides.

Table 4–7 NetworkEntityRuntimeMBean Status Attributes

Attribute	Description
getValue()	Specifies the address of the network entity in the URI format where with the colon character (:) is replaced with the underscore.

Table 4–7 (Cont.) NetworkEntityRuntimeMBean Status Attributes

Attribute	Description
getStatus()	Specifies a network entity status. Possible values: ■ 0 - Network entity is unavailable ■ 1 - Network entity is available ■ 2 - Status of the network entity is unknown

For more information on how to access runtime MBeans, see the discussion on monitoring Service Broker using runtime MBeans in *Oracle Communications Service Broker System Administrator's Guide*.

5

Implementing Overload Protection

This chapter explains how to protect Oracle Communications Service Broker Policy Controller (Policy Controller) from system overload. For more information, see the *Oracle Communications Service Broker System Administrator's Guide*.

About Overload Protection

In some cases, such as unanticipated traffic peaks or failure of a network hardware or software component, the load on Policy Controller can increase significantly. This can cause a situation known as system overload in which Policy Controller has insufficient resources to handle new sessions. If overload is not handled correctly, the system can fail and lose critical data.

To handle increased amounts of traffic without damaging operations of the entire system, Service Broker provides an overload protection mechanism. This mechanism operates in processing domains where you can define criteria for overload detection.

By default, an overload condition is triggered by too many active sessions or initial requests. The system rejects initial requests while the overload lasts. In addition to rejecting initial requests, Service Broker provides the capability to customize how the system behaves if overload occurs.

See the discussion on implementing overload protection in *Oracle Communications Service Broker System Administrator's Guide*, for more information.

Using Gauges and Counters as Key Overload Indicators

When you configure gauges and counters as key overload indicators, Service Broker triggers overload protection if threshold values are crossed as measured by those indicators. You can select any of the counters and gauges provided by Service Broker to serve as key overload indicators.

> **Note:** Consult with Oracle Technical Support if you have questions about which runtime MBeans are best suited to implement overload protection in your network environment.

When you configure overload protection, your settings are applied uniformly across all managed servers in the domain. Usually, server load balancing allocates traffic "fairly" across servers. However, it is possible for a single managed server in a domain to enter an overload condition while the other servers are functioning normally.

About System and Module Levels of Overload Protection

You can configure Service Broker overload protection at the system and module levels:

- System counters and gauges: These two indicators provide system level overload protection. They can detect and trigger an overload condition that might occur across any number of clustered managed servers or when deploying any combination of Service Broker products.
  - **`SystemCountersRuntimeMBean.SessionGauge`**
  - **`SystemCountersRuntimeMBean.InitialRequestCount`**

  These counters and gauges monitor sessions that occur per managed server and per product. The indicators are systemwide because they monitor all sessions across the domain for all managed servers. Note there is no global overload protection status.

  Using JConsole in a live environment, the platform system indicators are located under this node: **oracle.axia.platform.runtimembean.pn.**
- Module level counters and gauges:

  Any of the indicators described in Table 4–1 and Table 4–2 can provide module-level protection based on overload conditions that might occur when using Policy Controller.

  The Policy Controller **GxThrottledSessions** and **RxThrottledSessions** counters described in Table 4–1 and Table 4–2, are related to monitoring overload protection because they hold the number of initial requests that were throttled due to an overload condition. However, they should not be used to implement threshold-crossed overload protection.
- **Note:** Policy Controller should primarily use the platform-level **SessionGauge** and **InitialRequestCount** indicators for overload protection described in Table 5–1.

Table 5–1 Counters and Gauges For Implementing System Overload Protection

Attribute	Description	Type
InitialRequestCount	Incremented each time the SessionGauge is incremented within the current counter interval.	System-level Counter
SessionGauge	In a co-deployed domain, other applications such as the Online Mediation Controller can also update this gauge.	System-level Gauge

Understanding the Essential Steps for Configuring Overload Protection

These steps must be followed to configure overload protection.

1. By default, the following system level gauge and counter are defined as your key overload indicators:
   - **`SystemCountersRuntimeMBean.SessionGauge`**: A gauge that measures the number of active sessions currently handled by Service Broker.
   - **`SystemCountersRuntimeMBean.InitialRequestCount`**: A counter that measures the rate at which Service Broker receives new sessions.

   You can configure certain parameters such as threshold crossed and threshold ceased values but usually these indicators work well with their default settings.

2. Identify the Policy Controller level counters and gauges you want to use as key overload indicators.
3. Configure Threshold Crossed Notifications details for your Policy Controller counters and gauges.

   Define an upper threshold and ceased value threshold. For example, if the upper threshold value is 100 and the ceased value is 90 then if 100 is crossed the system remains overloaded until the value goes below 90.

   Specify a threshold name for each counter or gauge. If you use either sessionGauge or initialRequestCount as the threshold name value you do not have to add these indicators to the Key Overload Indicators pane.
4. Configure Key Overload Indicators.

   After you have configured the Policy Controller counters and gauges you want to use for overload protection you must specify that they are to be used by Service Broker as Key Overload Indicators.

   You do this in the Administration Console by selecting Tier Management, and then Overload Pane. In the Key Overload Indicators pane, you can set your key overload indicators.

   Note: Service Broker activates overload protection when any of your key overload indicator crosses its upper threshold during a sampling interval.
5. Customize overload protection behavior.

   The behavior of Service Broker is that if an overload condition occurs, the system continues to handle all active sessions but rejects initial requests until the overload condition ceases.

   You can customize the overload protection error code by modifying the value of the result code. In the Administration Console select the PCRF tab, then OCSB, then System Parameters, then Global Parameters. Configure the Overload Rejection Result Code.

Configuring Threshold Crossed Notifications Rules

This section describes how to create Threshold Crossed Notifications rules for overload protection. The components of these rules specify MBean type, threshold name, crossed and ceased threshold values, and other fields.

The following steps are applicable to both systemwide and module-level counters and gauges. There are only two systemwide overload indicators: sessionGauge and InitialRequestCount.

To transform the counter or gauge you configure in this section to be a key overload indicator you must match the threshold name value you set under the Monitoring tab with the threshold name value in the Key Overload indicators pane.

For example, the default key overload indicator threshold name **`sessionGauge`** matches the threshold name value in the default threshold crossed notifications rule also sessionGauge.

To configure Threshold Crossed Notification Rules do the following:

1. In the navigation tree, expand the **OCSB** node.

2. Expand **Processing Tier**.
3. Do any of the following:
   - To configure System-level Counters and Gauges: Expand Tier Management, then Monitoring, and then Monitoring. **Note**: You cannot add more counters and gauges in addition to the default sessionGauge and InitialRequestCount indicators. However, if required you can modify details such as crossed and ceased threshold values.
   - To configure Module-level Counters and Gauges: Expand PCRF, then OCSB, then System Parameters, and then Statistics.
4. Select **Threshold Crossed Notifications**.
5. Be sure you have selected **Lock & Edit** and then click **New**.
6. In the **Threshold Name** field, enter a string that names the threshold. This value is referenced by the key overload indicators.
7. For the **Enable threshold** field, select **True** or **False**. Only enabled thresholds are considered for overload protection.
8. In the **MBean Type** field, enter the type of MBean.
9. For the **Counting Type** field, select the Counting method. For gauges use CurrentGeneralValue and for counters use CurrentIntervalDeltaValue.
10. In the **MBean Attribute** field, enter an MBean attribute. For SystemGaugeRuntime use SessionGauge and for SystemCountRuntime use InitialRequestCount.
11. In the **Threshold class** field, enter **High**. Crossing a low threshold does not cause an overload state.
12. In the **Threshold Value** field, enter an integer value which when crossed triggers an overload state.
13. In the **Threshold ceased value** field, enter an integer value which when crossed the triggered threshold ceases. This value is applicable only to gauges.
14. In the **Threshold crossed message** field, enter a message included in the threshold notification.
15. In the **Threshold ceased message** field, enter a message included in the threshold ceased notification.
16. In the **Server filter** field, leave it empty or use a regular expression to filter on a managed server. For example “managed_1” or “server.”
17. In the **Resource filter** field, enter a unique name for the indicator.
18. Click **Apply**.

Specifying Your Key Overload Indicators

Identify the counters and gauges you want to use for overload protection. Use the Administration Console to list the names and threshold names for these indicators.

The Overload Protection pane is pre-populated with these two systemwide indicators:

- **`SystemCountersRuntimeMBean.SessionGauge`**
- **`SystemCountersRuntimeMBean.InitialRequestCount`**

To specify module level Key Overload Indicators do the following:

1. In the navigation tree, expand the **OCSB** node.

2. Expand **Processing Tier**.
3. Expand **Tier Management**.
4. Select **Overload Protection**.
5. Be sure you have selected **Lock & Edit** and then click **New**.
6. In the **Name** field, enter a unique name for the indicator.
7. In the **Threshold Name** field, enter a unique string that references the threshold.

   Multiple indicators at the system or module level can use the same Threshold Name. In this situation, all matching crossed thresholds will be considered to indicate an overload state.

   Example: If any counter or gauge uses either sessionGauge or initialRequestCount as the threshold name value, you do not have to add these indicators to the Key Overload Indicators pane. However, the module-level settings (for example, crossed threshold value) will override the platform-level settings for that individual software bundle only.
8. Click **Apply**.

Configuring General Monitoring Parameters For Policy Controller

This section describes how to configure general attributes for overload protection notifications.

To configure general Policy Controller statistics settings:

1. Start the Policy Controller domain and managed server.
2. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.
3. Expand PCRF, then OCSB, then System Parameters, then Statistics, and then General.
4. Be sure you have selected **Lock & Edit**.
5. In the **General** pane, configure the desired parameters according to the descriptions in Table 5–2.
6. When you have finished, click **Apply**.

Table 5–2 General Overload Configuration Parameters for Policy Controller

Name	Description
Enable runtime MBeans	Disables the runtime MBeans so you can neither poll them for values or get notifications.
Enable Notifications	Disables only notifications, so you can still poll values from the MBean.
Counter Interval (sec)	This parameter specifies the length of the interval in seconds.
Notification trigger interval (sec)	Sampling interval in seconds for checking notifications. For example, if the Counter Interval is set to 10 seconds and the Notification trigger interval is set to 2 seconds for each counter interval the system will determine 5 times whether the threshold value has been crossed.

Configuring the Overload Protection Behavior

When system overload occurs, Service Broker rejects new sessions and sends response messages to the network entities that attempted to establish the new sessions.

At the Policy Controller level you can configure an Overload Rejection Result Code.

In the Administration Console: PCRF, then OCSB, then Execution Block, and then Global Parameters. You can change the value in this field: Overload Rejection Result Code.

This Diameter result code is used in the answer of a rejected initial request and the default value is **5012 DIAMETER_UNABLE_TO_COMPLY**.

6

Adding Custom Diameter AVPs

The Oracle Communications Service Broker Policy Controller (Policy Controller) Diameter stack supports a specific list of Diameter AVPs that Policy Controller can use to pass information from Diameter reference points. If your Policy Controller implementation requires additional AVPs, follow the instructions in this chapter to add them to the Diameter stack using the Administration Console. Policy Controller rejects any messages with unsupported AVPs and returns an error message to the sender.

About Adding Custom Diameter AVPs

Policy Controller entities such as Policy Control Enforcement Functions (PCEFs), Application Functions (AFs), Online Charging Systems (OCSs) may use custom (non Diameter-standard) AVPs to transfer information among Diameter nodes. If your Policy Controller implementation requires it, you can add these custom AVPs to the Diameter stack. Policy Controller rejects any messages with unsupported AVPs and returns an error message to the sender.

You are free to add as many custom AVPs to the Policy Controller Diameter stack as necessary, and this chapter explains how. Once custom AVPs are added to the stack, Policy Controller accepts messages with these AVPs, but does not by default use any of the information they contain. To make use of the information they contain you must create a Policy Designer rule that uses the AVP information.

Note: If your Policy Controller implementation uses split domains, you must add the same list of AVPs to both domains manually.

The "About Diameter AVP Data Types" section lists how various Service Broker and Policy Controller entities represent the different AVP data types, and "Functions Available for Each Data Type" lists the functions that they can use.

Adding Custom AVPs to Use with Policy Controller

To add custom AVPs to Policy Controller:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Navigate to **Platform**, **OCSB**, **Domain Management**, **Diameter AVPs, AVP Definitions**.

   The **AVP Definitions** screen appears.

3. Click **Lock and Edit**.
4. Click **New**.

   The **New AVP Definitions/Attributes** pane appears.
5. Enter a **Code** for the new AVP.

   Ensure that this code is not used by any existing Diameter AVP.
6. Enter the **Vendor-Id** of the AVP.

   This is the IANA SMI Network Management Private Enterprise Codes for the vendor. For details see `http://www.iana.org/assignments/enterprise-numbers`. A value of **0** indicates that there is no vendor code.
7. Enter a **Name** for the AVP.

   Enter a unique name for the new AVP.
8. Select a data **Type** for the new AVP.

   See Table 6–1 for a list of the available types.
9. Select a **Mandatory Flag** option to control support for the new AVP. See RFC 3588 for details. The options are:
   - Default - No behavior is specified if the AVP is unsupported by a Diameter client.
   - **MUST** - The AVP must be rejected by a Diameter client if unsupported.
   - **MAY** - The AVP may be rejected by a Diameter client if unsupported
   - **MUST_NOT** - The AVP can never be rejected by the Diameter client if unsupported.
10. Select a level of security support in the **End-to-end encryption** setting. The options are:
    - Blank (no value - Does not specify a security behavior.
    - **TRUE** - The AVP must not be sent unless end-to-end encryption is supported by both parties of the connection.
    - **FALSE** - The AVP may be sent even if end-to-end encryption is not supported by both parties of the connection.
11. Click **Apply**.

    The new AVP appears in the **AVP Attributes** table.
12. Repeat steps 4 through 11 for each custom AVP you are adding.

About Diameter AVP Data Types

Table 6–1 lists the various AVP data types and how they are represented in Policy Controller entities.

Table 6–1 Diameter AVP Data Types and the Policy Controller/Policy Designer Equivalents

Policy Controller (Administration Console) Representation	Diameter AVP Type (from RFC 3588)	Derived from type (from RFC3588)	Policy Designer Representation	Policy Designer Representation for Custom AVPs
bytes	OctetString	N/A	java.lang.String	java.lang.String
integer32	Integer32	N/A	java.lang.Integer	java.lang.Integer
integer64	Integer64	N/A	java.lang.Long	java.lang.Long
unsigned32	Unsigned32	N/A	java.lang.Long	java.lang.Long
unsigned64	Unsigned64	N/A	java.math.BigInteger	java.math.BigInteger
float32	Float32	N/A	java.lang.Float	java.lang.Float
float64	Float64	N/A	java.lang.Double	java.lang.Double
grouped	Grouped	N/A	N/A	N/A
addressbytes	Address	OctetString	InetAddress Address	Address
time	Time	OctetString	java.sql.Time (**oracle.ocsb.app.rcc.feature.pcrf.api.type.Time**)	java.sql.Time
string	UTF8String	OctetString	java.lang.String	java.lang.String
bytes	DiameterIdentity	OctetString	java.lang.String	java.lang.String
bytes	Diameter URI	N/A	java.lang.String	java.lang.String
integer32	Enumerated	Integer32	Known types will have a new class, for example: EventTrigger (**oracle.ocsb.app.rcc.feature.pcrf.api.type.EventTrigger**) IPCanType (**oracle.ocsb.app.rcc.feature.pcrf.api.type.IPCanType**)	java.lang.Integer
bytes	IPFilterRule	OctetString	IPFilterRule (**oracle.ocsb.app.rcc.feature.pcrf.api.type.IPFilterRule**)	IPFilterRule (**oracle.ocsb.app.rcc.feature.pcrf.api.type.IPFilterRule**)
bytes	QoSFilterRule	OctetString	IPFilterRule (**oracle.ocsb.app.rcc.feature.pcrf.api.type.IPFilterRule**)	IPFilterRule (**oracle.ocsb.app.rcc.feature.pcrf.api.type.IPFilterRule**)

Functions Available for Each Data Type

Table 6–2 lists the Policy Designer rule functions available for custom AVPs by data type.

Table 6–2 Supported Functions for Custom AVPs

Data Type	Supported Function	Description
IntegerList	contains(Integer *intValue*)	Check whether *intValue* is in the list.
IntegerList	`hasGreaterThan(Integer` *`intValue`*`)`	Check whether the list contains a value greater than *intValue*.
IntegerList	`hasLesserThan(Integer` *intValue*)	Check whether the list contains a value lesser than *intValue*.
LongList	`contains(String` *longValue*)	Check whether *longValue* is in the list.
LongList	`hasGreaterThan(String` *longValue*)	Check whether the list contains a value greater than *longValue*.
LongList	`hasLesserThan(String` *longValue*)	Check whether the list contains a value lesser than *longValue*.
BigIntegerList	`contains(String` *bigIntegerValue*)	Check whether the list contains *bigIntegerValue*.
BigIntegerList	`hasGreaterThan(String` *bigIntegerValue*)	Check whether the list contains a value greater than *bigIntegerValue*.
BigIntegerList	`hasLesserThan(String` *bigIntegerValue*)	Check whether the list contains a value less than *bigIntegerValue*.
StringList	`contains(String` *stringValue*)	Check whether *stringValue* is in the list.
FloatList	`contains(String` *floatValue*)	Check whether *floatValue* is in the list.
FloatList	`hasGreaterThan(String` *floatValue*)	Check whether the list contains a value greater than *floatValue*.
FloatList	`hasLesserThan(String` *floatValue*)	Check whether the list contains a value less than *floatValue*.
DoubleList	`contains(String` *doubleValue*)	Check whether *doubleValue* is in the list.
DoubleList	`hasGreaterThan(String` *doubleValue*)	Check whether the list contains a value greater than *doubleValue*.
DoubleList	`hasLesserThan(String` *doubleValue*)	Check whether the list contains a value less than *doubleValue*.
AddressList	`contains(Integer` *addressType*, `String` *addressData*)	Check whether values for *addressType*, *addressData* are in the list.
IPFilterRuleList	Not used by Policy Controller	Not used by Policy Controller.
TimeList	`contains(Calendar` *c*)	Check whether the list contains *c*.
TimeList	`hasBefore(Calendar` *c*)	Checkwhether the list contains a value less than *c*.
TimeList	`hasAfter(Calendar` *c*)	Check whether the list contains a value greater than *c*.

7

Working With the Policy Designer Interface

This chapter explains information that is important to know when using the Policy Designer interface to Oracle Communications Service Broker Policy Controller (Policy Controller).

Working with Deployments

Each deployment includes all of the currently active rules, rulesets, and their supporting data (lists of values, complex lists, local facts, and so on), and all PCC and ADC profiles. You can:

- Save iterative copies of deployments inside Policy Controller, and revert to these previous deployment versions as necessary.
- Export deployments as a file out side Policy Controller. You can then later import the deployment, or allow other deployment developers to work on it.
- Import deployments from outside Policy Controller that other deployment developers have worked on.

You can revert to an older version of an individual deployment as necessary.

> **Note:** Each time you deploy a new version of a deployment it overwrites the entire deployment contents. Ideally then only one person should modify a deployment at a time. If your deployment requires that more than one person to modify a deployment, they need to manually coordinate their efforts so their changes do not collide.

The last 30 deployments are displayed by default. Clicking the **Show All** button displays all deployments.

Managing Deployments

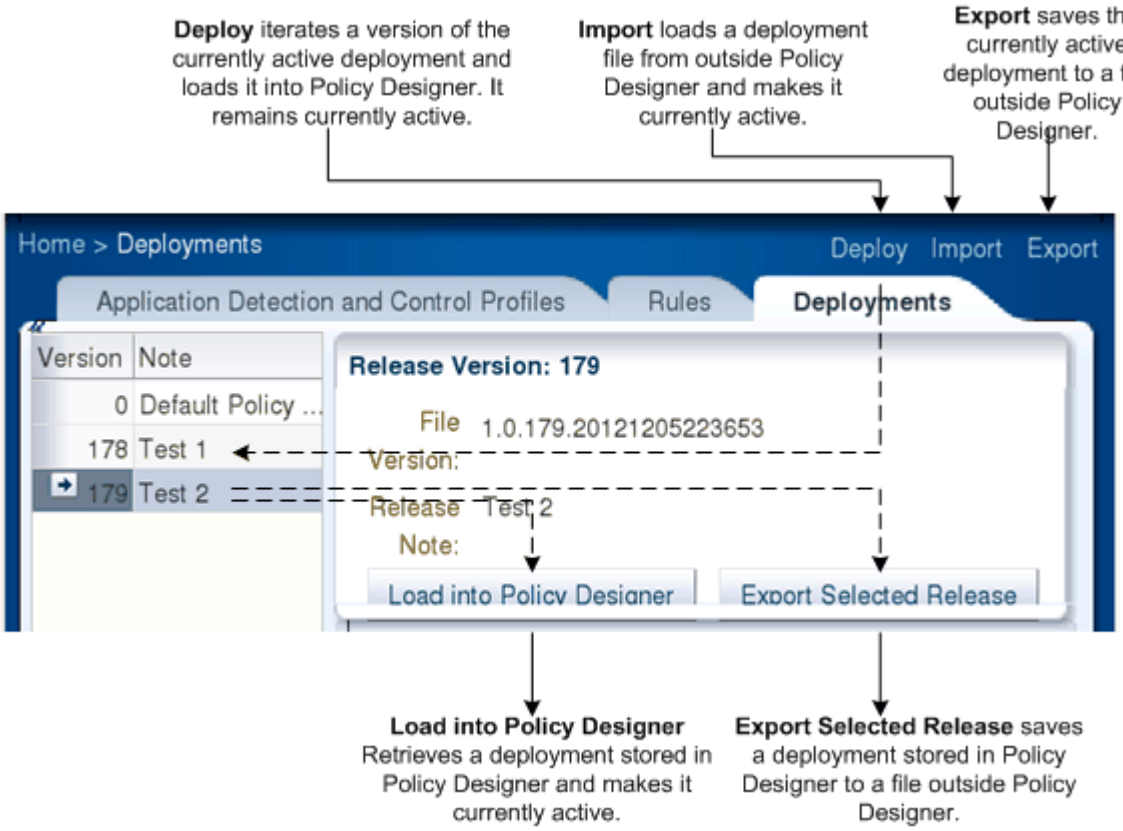
You can save incremental versions o f the currently active deployment using the Deploy button. Deployed deployments are listed on the left side of the Policy Designer Deployments tab under the **Version** column as shown in Figure 7–1.

Figure 7–1 also shows the controls you use to manage deployments:

- The **Deploy** button at the top right of the Policy Designer window makes a copy of the current deployment and saves it inside Policy Designer.

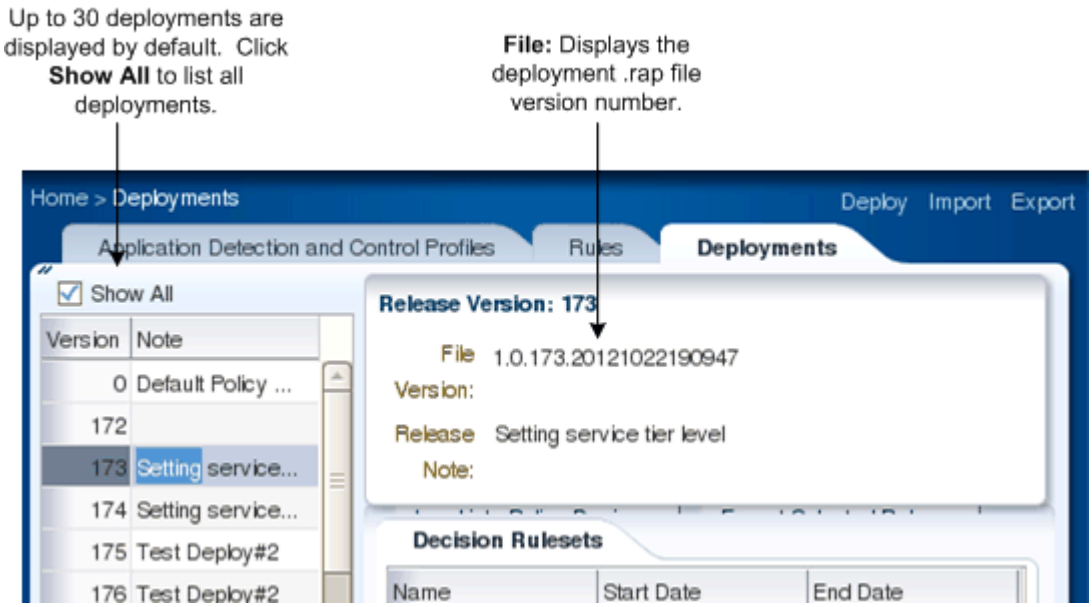
- The **Import** button at the top right of the Policy Designer window take a deployment file saved in the file outside Policy Designer and makes it the currently active deployment
- The **Export** button at the top right of the Policy Designer window save the currently active deployment to a file outside of Policy Designer
- The **Load into Policy Designer** button in the **Deployments** tab takes a deployment saved in Policy Designer and make it the currently active deployment.
- The **Export Selected Release** button in the **Deployments** Tab takes a deployment saved inside Policy Designer saves it to a file outside of Policy Designer.

Figure 7–1 Managing Deployments in the Deployments Tab



By default, Policy Controller stores your deployments in the *domain_home*/archive directory in files with this naming syntax: oracle.axia.cm.config.*release_version*.rap. Only the *release_version* is shown in the Deployments **File:** field as shown Figure 7–2.

Figure 7–2 Managing Deployments



After you deploy a deployment, select the **Deployments** tab to see whether it appears in the list of **Decision Rulesets** list.

To save an incomplete set of rulesets and their list of values and decision functions to work on later, export the deployment that contains them to an external **.rap** file using the **Export** button shown in Figure 7–2. When you are ready to resume work, import the **.rap** file containing the deployment back into the rule editor.

Sharing Deployment Files

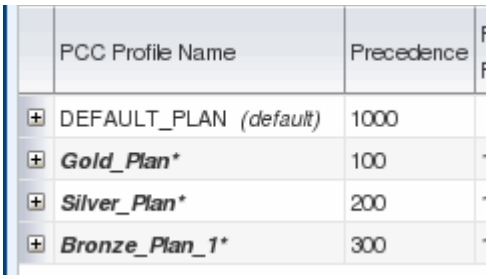
If your implementation requires that more than one person make changes to the same deployment, click **Export** from the upper left side of the Policy Designer interface to save the file to a location where the other person has access. Enter a location that the other person can access when the export window appears, and save the deployment to that location. Then let the other person know where you exported the file. When you are ready take the deployment back, click **Import** on the upper left side of the Policy Designer interface and select the deployment to import.

Working with Profiles

This section includes information that is common to the tasks of creating Policy Charging and Control (PCC) and Application Detection and Control (ADC) profiles. See the "Creating Policy Charging and Control Profiles" and "Creating Application Detection and Control Profiles" chapters for details about creating individual profiles.

The PCC and ADC profile names are shown in the ADC or PCC Profile Name columns indicate their status. Figure 7–3 shows newly-created PCC profiles. After they are created, but before they are stored in the Policy Controller managed server (deployed) they appear in bold italic font with an asterisk (*) symbol next to them.

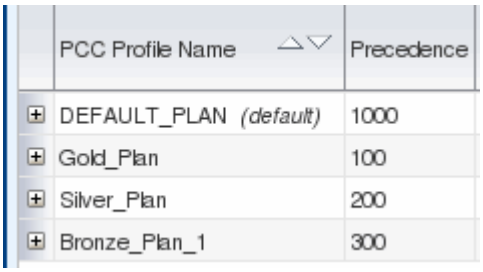
Figure 7–3 Created Profiles display with Bold-Italic Font



PCC Profile Name	Precedence
DEFAULT_PLAN *(default)*	1000
Gold_Plan*	100
Silver_Plan*	200
Bronze_Plan_1*	300

Figure 7–4 shows the same profiles after they have been deployed. Deployed profiles are displayed with a "normal" font. That is, neither bold nor italic, and the asterisk is removed.

Figure 7–4 Deployed Profiles Display with Normal Font



PCC Profile Name	Precedence
DEFAULT_PLAN *(default)*	1000
Gold_Plan	100
Silver_Plan	200
Bronze_Plan_1	300

If you change a deployed profile, the display font reverts to the bold italic font with the asterisk symbol (*) as for newly created profiles.

About Rules

The heart of the Policy Designer is the Policy Designer **Rules** tab that you use to create rules that select profiles to apply to subscribers. These rules are tests to decide whether a subscriber is entitled to the bandwidth capabilities and/or limits in a specific PCC or ADC profile.

See "Working with Rules" for details on the Policy Designer tools you use to create rules.

Example Rule Strategies

"Strategies for Creating Rules" contains example rules that illustrate the flexibility you have in creating rules.

About Redirecting Subscriber Service or Session Traffic

Redirecting subscriber traffic is the only non-3GPP specified feature that Policy Controller supports. Redirecting traffic is not supported by the 3GPP Gx Release 9 specification that Policy controller uses, so Oracle created this capability independently.

You can create rules that redirect a subscriber's service data flow to a Web address that you specify. *Global* redirection redirects all traffic for a single session; *service level* redirection redirects the service data flow for just a specific service that a PCC profile applies to. Redirection is often used to send subscribers to a website to top-up

accounts, or offer enhanced services for an additional fee, but you can use redirection for any purpose your implementation requires.

Global redirection blocks all subscriber traffic for a given session. Effectively then you are blocking all traffic for a subscriber unless their profile information changes, because the same rule tests that are performed each time they start a session. For example, if a subscriber has used up their bandwidth allotment for the month, you can redirect all traffic away from their services and to the address of a website that they can use to top-up their account. Once the account is topped-up, the next session they start allows them to access services again.

Service level redirection blocks specific services within a session. Examples of service-level redirection include parental controls to block adult services from children, or businesses to block gambling services from a professional network.

See "Redirecting Service Data Flow for a Session or Individual Service" for details on creating rules that redirect sessions.

About Using Policy Controller to Send SMS Messages

You can create rules that send SMS messages to your subscribers based on any of the same subscriber or other PCRF implementation information that your other rules can use. For more information on creating rules that send SMS messages, see "Using Rules to Send SMS Messages"

Using Multi-service Products

Policy Controller can create rules that are valid for all services, or per-service. Managing bandwidth per service enables you to offer multi-service plans that subscribers can use simultaneously. If services compete for bandwidth, you can control their behavior by setting different priorities for those services and selecting use options for keeping them within the bandwidth thresholds you set. For example, if two services compete for the same bandwidth you can terminate one, or throttle its bandwidth back, or simply offer to provide the same bandwidth for both at a higher cost.

Policy Controller can also create profiles that are valid for all services. You specify your preferences using rules and rulesets that you create using Policy Designer.

You have the following options for modifying services that exceed their thresholds:

- Remove access to the service
- Change the cost of service
- Throttle one or more services of a plan
- Block a specific UE device

Dynamically Changing Service Offerings

Policy Controller uses subscriber information from your SPR/OCS to select the appropriate PCC profiles. You can use any combination of the SPR/OCS AVPs in your rules to create custom service offerings for individual subscribers. For example you could charge different rates for different application functions. Or change rates when a subscriber uses a specific combination of application functions simultaneously.

A popular scenario is to create PCC profiles that specify bandwidth limitations based on a subscriber profile information. You use the subscriber profile fields in your rules to create custom service offerings for each individual subscriber. You can extend the

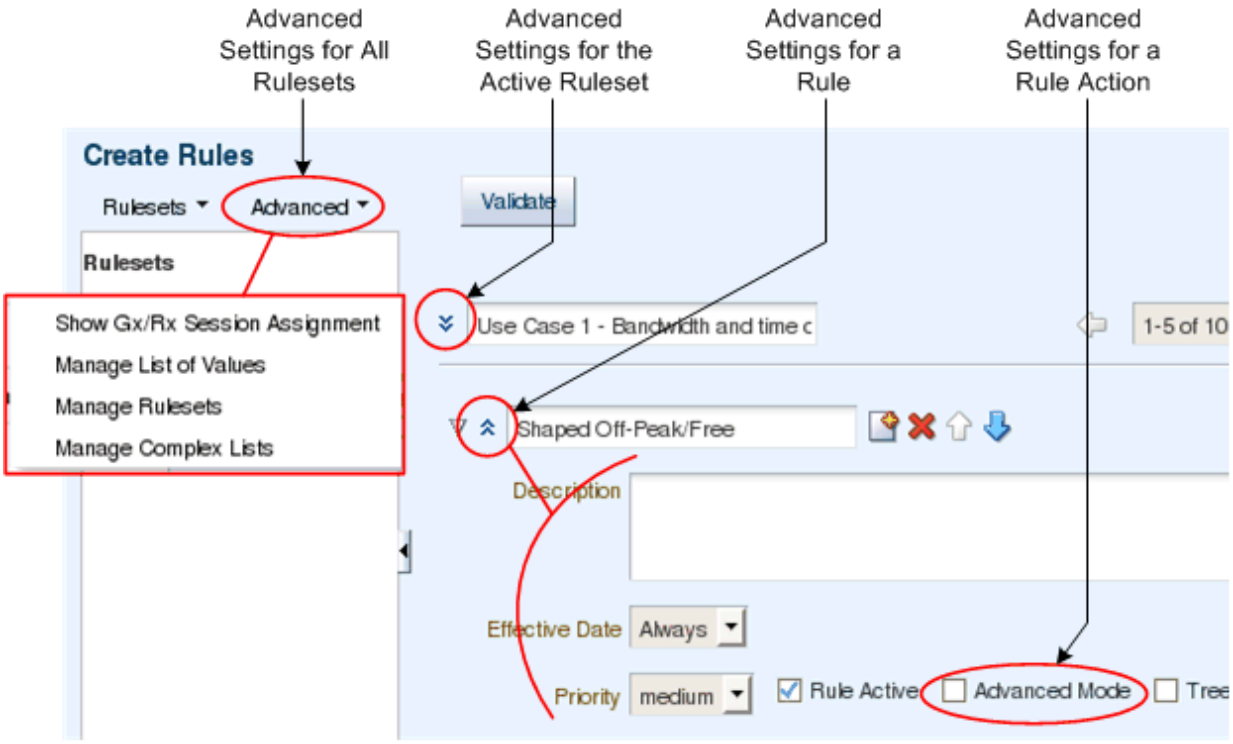
default subscriber profile as needed to store and obtain the information your services require.

For more information on the Subscriber Store see *Oracle Communications Service Broker Subscriber Store User's Guide*

About the Advanced Settings for Rulesets, Rules, and Rule Actions

The most common ruleset, rule, and rule action options are displayed in the Policy Designer **Rules** tab by default. If your implementation requires the settings that Policy Designer reserves for advanced features you can access them using the advanced setting icons shown in Figure 7–5.

Figure 7–5 Advanced Settings in the Rules Tab



Using the Advanced Menu Features for All Rulesets

When advanced settings are hidden, you see options that most users use. When advanced settings are shown, you are offered additional options to configure.

The Advanced menu on the upper left side of the Policy Designer interface displays these menu items:

- Show Rx/Gx Session Assignments - Used to select and reference nondefault Gx session in rules. See "Selecting Among Subscriber Sessions" for details.
- Manage List of Values - Defines a group of constants to use in your rules. The predefined lists include the names of months and result code types. See "Managing Lists of Values" for details on creating your own constants.
- Manage Rulesets - Used to activate/inactivate rules, and change the order in which they are applied. See "Managing Rulesets" for details and instructions.

- Manage Complex Lists - Shows/hides individual Rx and Gx grouped AVPs. See "Using Grouped AVPs in the Rules" for details.

Using the Advanced Settings for the Active Ruleset

Click the rulesets advance mode icon to display these settings for rulesets:

- **Description** - An informal description of the ruleset.
- **Effective Date**: - Enables you to enter Always to make the ruleset perpetually effective, or enter a range of dates for the ruleset to take effect. See "Setting the Effective Date for a Rule or Ruleset" for details.
- **Active** - Check this box to make the ruleset active.

Using the Advanced Settings for a Rule

Click the rule advanced settings icon to display these settings for rules:

1. In the **Description** field, you can enter an option textual description of the rule. [Advanced Setting]
2. From the **Effective Date** menu, select a configuration for the effective date of the rule. See "Setting the Effective Date for a Rule or Ruleset" for details. The default is **Always**. [Advanced Setting]
3. Select a priority level from the **Priority** menu relative to the priority of the other rules in the ruleset. The options are: **highest**, **higher**, **high**, **medium** (the default), **now**, **lower**, and **lowest**. [Advanced Setting] Higher priority rules are interpreted before lower priority rules. The default priority is **medium**.
4. Check the **Rule Active** check box to activate the rule or clear the check box to deactivate it. The default is Rule Active. [Advanced Setting]
5. To enable advanced mode, check the **Advanced Mode** check box. The default view hides a lot of the complexity and flexibility that Oracle Business Rules offers because most users do not need them. If your implementation does, then check this box.

   Advanced mode allows additional pattern-matching options for creating conditions and actions. Advanced mode also enables you to test the rule with specific data values. See the discussion of advanced mode rules in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for information about advanced mode. [Advanced Setting]
6. To enable tree mode, check the **Tree Mode** check box. Tree mode is used for master detail rule hierarchies. See the discussion of tree mode rules in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for information about tree mode. [Advanced Setting]

Working with Rulesets

A *ruleset* is a collection of Policy Controller rules. Rulesets are run in the order of the priorities assigned to them.

You create and delete a ruleset from the Rulesets menu in the upper left section of the rule editor.

The order in which rulesets are applied is set in the predefined **DEFAULT_PLAN** decision function. A decision function provides a contract for executing rulesets.

See these sections for details on working with rulesets:

- Creating and Deleting Rules
- Setting the Effective Date for a Rule or Ruleset
- Managing Rulesets

Managing Lists of Values

You use a list of values, also known as a list of value ranges, to define a group of constant values to use in rules. You can also create a shorter, more informal name for each value, called an *alias*.

WARNING: Do not delete or modify the default lists of values directly as this could cause unpredictable behavior in the rules engine. Instead, allow these list of values to be populated automatically by the Policy Controller interface.

For example, assume you have a set of account types with long names. You could create a list of values called **accountValues** with a String data type in which the true values are the value properties and the aliases are something shorter:

Value	Alias
acct1234567890462_A	acctA
acct7777770000443_B	acctB

Then you could create a rule that references the alias instead of the long name:

```
IF In.subscriberProfile.accountType is accountValues.acctA
```

See the discussion of working with lists in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for more information.

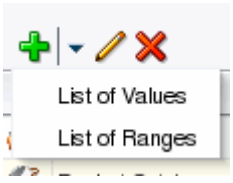
Creating a List of Values

To create a list of values:

1. Select **Manage List of Values** from the **Advanced** menu.

   The list of values list appears.

2. Click the Add List of Values menu and select whether the list of values will define a list of values or a list of ranges (LOV).



   The new list of values appears in the list with a default name.

3. In the list of values, select the list of values that you just created.

4. Click the pencil icon.

The **List of Values Editor** appears.

5. In the **Name** field, overwrite the default name with the name you want to assign to the list of values. Do not use the same name for the list of values as the alias of a fact, as this will cause a validation error.
6. In the **Description** field, optionally enter a textual description of the list of values.
7. From the **Data Type** menu, select the data type of the values in the list of values.

   All the values in a list of values must be of the same type.
8. If you want to allow invalid values in tests, check the **Include Disallowed Buckets in Tests** check box. This is useful for testing Policy Controller's reaction to invalid values.
9. Do one of the following:

   If the list of values defines a list of values:

   a. Click the add bucket icon above the **Bucket Values** list to add a value to check.

   b. In the **Value** field enter the name of the value to check.
   c. In the **Alias** field enter an alias for the value. This can provide a more meaningful name than the real value name.
   d. If the value is allowed in the actions (THEN) portion of a rule check the **Allowed in Actions** box. Otherwise the value is only allowed in the conditions (IF) portion of a rule.

      For more information, see the discussion of the list of values allowed in actions option in *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.
   e. Optionally add a description of the value in the **Description** field.
   f. Repeat steps a through e for every value that you want to add to the list of values.
   g. Click **OK**.

   If the list of values defines a range of values:

   a. Click the add bucket icon above the Range Bucket Values list to add a range to check.

   b. In the End Point field, enter the highest value in the range.
   c. Check the **Included Endpoint** check box if the endpoint is included in the acceptable range. Clear it if the endpoint is outside the range.
   d. In the Range field, enter the range of valid values.
   e. If the value is allowed in the actions (THEN) portion of a rule check the **Allowed in Actions** box. Otherwise the value is only allowed in the conditions (IF) portion of a rule.

For more information, see the discussion of the list of values allowed in actions option in *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.

f. In the Alias field enter an alias for the value. This can provide a more meaningful name than the real value name, which is in the Range field. The range field is read-only.

g. Optionally add a description of the value in the Description field.

h. Repeat steps a through g for every range that you want to add to the list of values.

10. Click **OK** to save your changes.

11. Deploy your changes to make them take effect. See "Deploying Rulesets to a Deployment" for details on deploying ruleset and list of values changes.

Editing a List of Values

To edit a list of values:

1. In the left panel, click **List of Values**.

   The list of values list appears.

2. Select a list of values to edit.

3. Click the pencil icon.

   The List of Values Editor appears.

4. Make your changes in the List of Values Editor following the guidance for creating a new list of values.

5. Click **OK** to save your changes.

6. Deploy your changes to make them take effect. See "Deploying Rulesets to a Deployment" for details on deploying ruleset and list of values changes.

Deleting an Item in a List of Values

To delete an item in a list of values:

1. In the List of Values Editor, select the item in the list that you want to delete.

2. Click the delete icon above the Bucket Values list.

Deleting a List of Values

To delete a list of values:

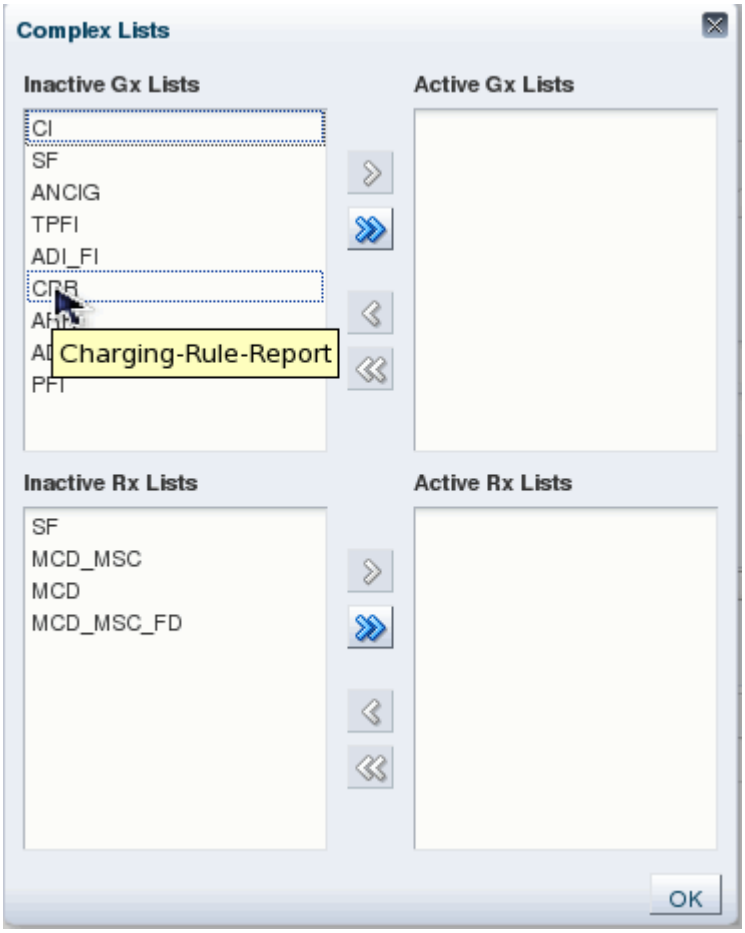
1. In the list of values list, select the list of values that you want to delete.

2. Click the delete icon above the list of values list.

Using Grouped AVPs in the Rules

To keep the Policy Designer rule **Condition Browser** window from becoming overly cluttered, not all Gx and Rx reference point AVPs are displayed. By default, the grouped Gx and Rx AVPs and their sub-grouped AVPs are not available to use so they are not displayed. However, you can choose to display some or all of these AVPs using the **Manage Complex Lists** item on the **Advanced** menu. You control the display on a per-AVP basis. Figure 7–6 shows the **Complex Lists** window that you use to make grouped AVPs available to use.

Figure 7–6 The Complex Lists Window



By default all grouped AVPs are set as inactive. That is, they are not available to use in Policy Controller rules unless you move them to the Active lists.

In the **Complex Lists** window, the acronyms represent AVPs. Hover your cursor over an acronym to find out what AVP it represents. Figure 7–6 for example, shows the cursor hovering over the "CRR" acronym, and the full AVP name "Charging-Rule-Report" is displayed. The underbar (_) between acronyms indicates that the grouped AVP contains another grouped AVP. MCD_MSC represents the Media-Sub-Component AVP within the Media-Component-Description AVP, and so on.

Adding Grouped AVPs to the Rule Condition Browser

To make grouped AVPs available to your rules:

1. Start the Policy Designer interface.
2. Select the **Rules** tab.
3. Select **Manage Complex Lists** from the **Advanced** menu.

   The **Complex Lists** window appears.
4. Select an AVP to make active.

   The AVP is highlighted.
5. Use the arrow (>) button to move the grouped AVP from the **Inactive** list to the **Active** list.

   Use the double arrow (>>) button to move all grouped AVPs to the **Active** list.
6. Click **OK**.

Removing Grouped AVPs from the Rules Condition Browser

To remove grouped AVPs from the Condition Browser:

1. Start the Policy Designer interface.
2. Select the **Rules** tab.
3. Select **Manage Complex Lists** from the **Advanced** menu.

   The **Complex Lists** window appears.
4. Select an AVP to make active.

   The AVP is highlighted.
5. Use the arrow (<) button to move the grouped AVP from the **Active** list to the **Inactive** list.

   Use the double arrow (<<) button to move all grouped AVPs to the **Inactive** list.
6. Click **OK**.

8

Creating Policy Charging and Control Profiles

This chapter explains how to create Oracle Communications Service Broker Policy Controller (Policy Controller) Policy Charging and Control (PCC) profiles that specify Quality of Service (QoS) limits and charging information for services.

This chapter assumes that you know the names of any predefined (PCEF-based) PCC rules that you must reference, including the Charging-Rule-Name AVPs to use.

About PCC Profiles

PCC profiles set the QoS limits and specify charging information for your subscribers. You must create PCC and ADC profiles first, then create rules that select them based on the criteria in those rules.

PCC and ADC profiles are the Policy Controller implementations of PCC rules as defined in the 3GPP TS 23.203 v9.9.0 specification. The PCC and ADC profiles can be either *dynamic* profiles that you create, configure, and store using Policy Designer, or *predefined* profiles that are created and stored in your PCEF. See your PCEF documentation for details any static PCC profiles it contains.

PCC profiles specify specific QoS limit details, and charging information, but not charging information details. You specify the Rating Group and Service IDs in PCC profiles that correspond to billing and charging details in your charging engine and other Policy Controller implementation entities.

There are two types of PCC profiles:

- Dynamic PCC profiles that your create and store in Policy Controller. The "Creating a Dynamic PCC Profile" section explains the process of creating these PCC profiles.
- Predefined PCC profiles that already exist and are stored in your PCEF.

Each PCC profile contains:

- A *precedence level* that specifies the order in which each PCC profile is interpreted.

  Precedence levels are positive integers with lower integers having higher priority. Profiles with the same priority are evaluated in a random order so it is best to avoid creating profiles with overlapping criteria.

  You can only set precedence levels for dynamic PCC profiles. The Policy Controller tools do not show, or allow you to set precedence levels for predefined profiles. You must use your PCEF tools and product documentation to find, and if necessary change the precedence level of predefined profiles.

  Consequently, you must create a precedence strategy that works for both dynamic and predefined profiles and keep it synchronized manually.

The default PCC profile has a precedence level of 1000 because you will probably give all other PCC profiles a higher precedence (lower integer). The default profile should always be evaluated last so it can specify bandwidth settings if no other profiles apply.

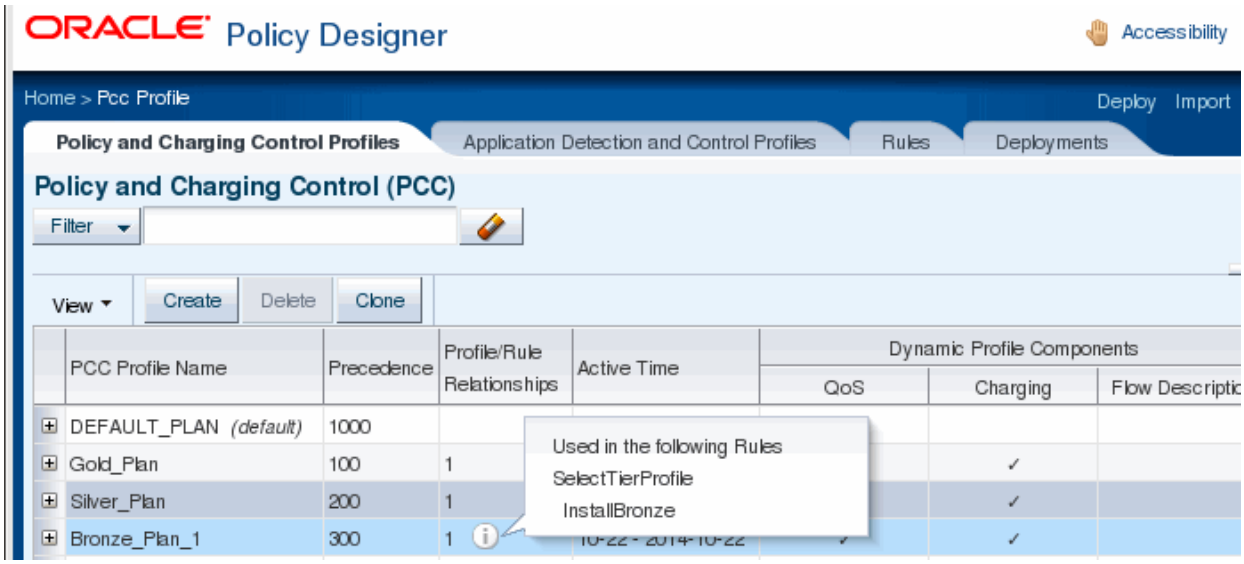
- Activation/deactivation dates and times that specify how long the PCC profile is in force.
- QoS limits, including:
  - A QoS Class Identifier (QCI) that specifies the type of traffic. There are nine specified options, or you can create your own.
  - The actual upper and lower bandwidth limits in bps, kpbs, Mbps, or Gbps.
  - And uplink status that controls whether uplink or downlink or both are allowed.
- Charging information, including:
  - The Rating Group ID
  - The Service ID
  - A metering method (duration, volume or both).
  - Options for enabling online charging, offline charging or both.

None of these options are required for a PCC profile to function; you can select the set of options that best suit your implementation. You create PCC profiles using the Policy Designer **Policy Charging and Control Profile** tab.

You also use the **Policy and Charging Control Profiles** tab to manage PCC profiles. Figure 8–1 shows the Policy Designer with the **Policy and Charging Control Profile**s tab selected. The **Policy and Charging Control Profiles** tab shows the PCC profiles listed (by default) in the order in which they were created in the **PCC Profile Name** column on the left. You can also list them alphabetically. The columns to the right of **PCC Profile Name** list the **Precedence** level, **Profile/Rule Relationships** status, **Active Time** limits, and whether **QoS**, **Charging**, or **Flow Description** profiles exit. An additional column (not shown) indicates whether the PCC profile is dynamic (created in Policy Designer) or predefined (stored in your PCEF).

Figure 8–1 also shows a cursor hovering over the **Profile/Rule Relationships** column, which displays a list of the rulesets and rules that contain the selected PCC profile. In this case **Bronze_Plan_1** PCC profile is selected, and it is used in the **InstallBronze** rule of the **SelectTierProfile** ruleset.

Figure 8–1 Policy Designer Profiles Tab



Policy Designer requires that you always have exactly one default (fallback) profile to charge subscribers if no other profiles apply. The default plan, **DEFAULT_PLAN** is provided for this purpose. You must define charging and Quality of Service details for the DEFAULT_PLAN to actually charge subscribers.

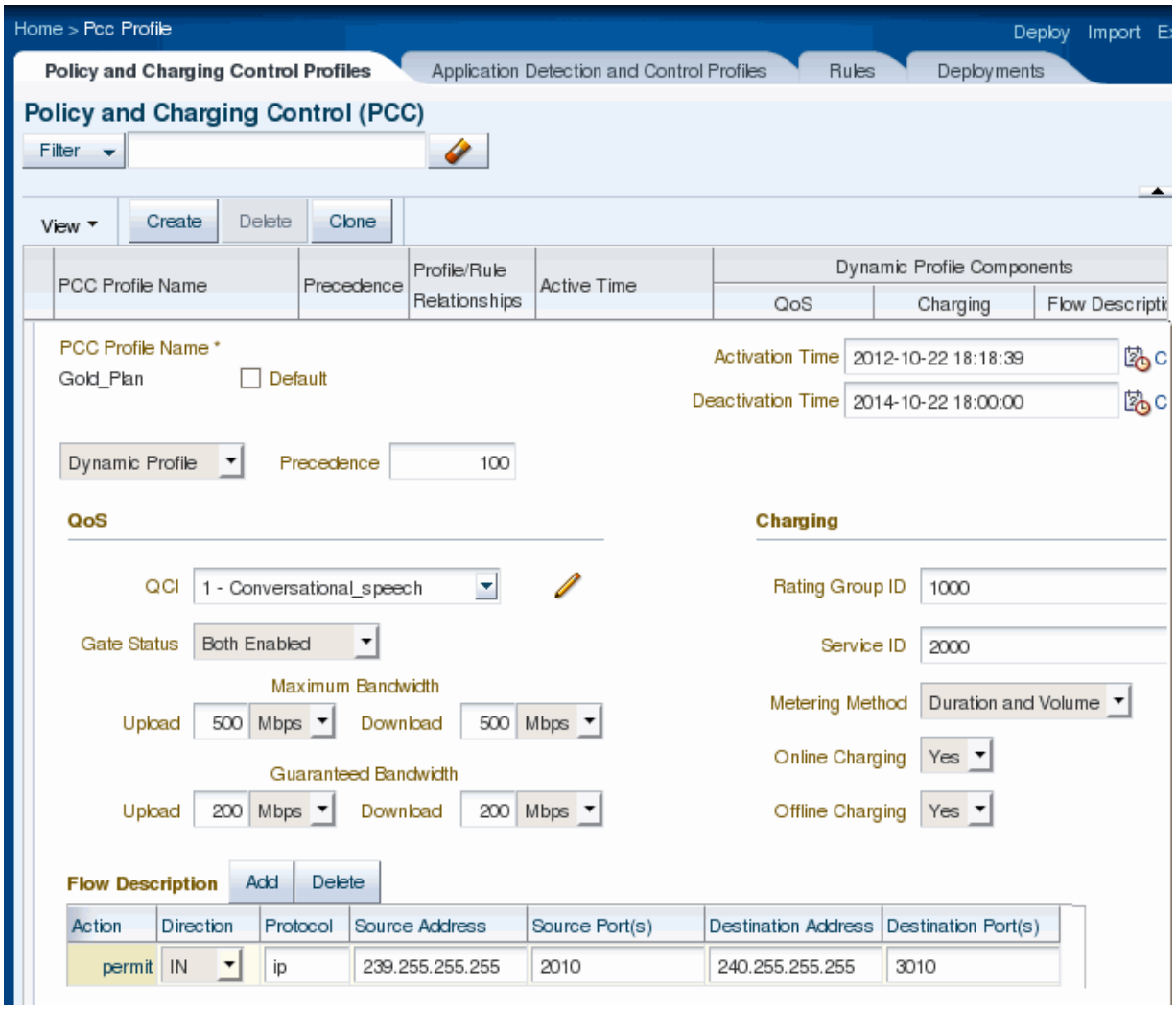
You cannot delete the default profile. However you can open another PCC profile and set it as the default plan. Whatever PCC profile you select as the default plan always has a precedence level of 1000 to ensure that it always has the lowest priority of all profiles. Remember that when you set a new PCC profile as the default plan, the old default plan keeps its status (probably active) and its precedence level of 1000. You will probably want to change these details so that it's activity does not collide with the new default plan's activities.

You may also use any PCC profiles stored in your PCEF with the Policy Designer (predefined PCC rules). To use a predefined PCC Rule in Policy Designer, simply create a new PCC profile with just the predefined PCC rule's name as it appears in the PCEF and select **Predefined Profile** from the Predefined Profile/Dynamic Profile menu box. Predefined PCC profiles are marked with a check mark in the **Predefined Profiles** column of the **Policy Charging and Control** tab.

Figure 8–2 shows the **Policy Charging and Control Profile** tab with details for a new PCC profile displayed. You use this window to specify the details of your PCC profiles:

- Charging details. This section references billing and charging details specified in your charging engine and your other Policy Congress implementation entities. Policy Controller does not actually rate or charge for service data flow. Instead, your OCS does the rating and charging based on its configuration and the values it receives from PCEF messages.
- A flow description (if any) to limit the service data flow by protocol and source (the **Flow-Description** AVP from 3GPP 29.214).

Figure 8–2 PCC Profile Details Window



Planning Your PCC Profiles

The procedures in "Creating a Dynamic PCC Profile" assume that you already know the details of the PCC profiles you will create, including:

- The bandwidth limits to set.
- A precedence level for each profile.
- Knowing the rating group IDs that your PCC profiles require, and adding them to Policy Controller. Follow the instructions in "Creating Rating Group IDs to Use in PCC Profiles" to add these rating group IDs to Policy Controller to use in PCC profiles.
- Knowing the service IDs that your PCC profiles require and adding them to Policy Controller. Follow the instructions in "Creating Service IDs to Use in PCC Profiles" to add these service IDs to Policy controller for use in PCC profiles.
- Deciding whether to use the PCC profile to filter service data flow by using the **Flow-Description** AVP parameters. For details on these parameters see the 3GPP TX 29.214 v9.*x* specification. A Flow Description dialog box gives you graphical interface access to these fields from the **Policy Charging and Control Profile**s tab.

Creating Rating Group IDs to Use in PCC Profiles

No rating group IDS are defined by default because each Online Charging System (OCS) uses a different list (also called "charging keys"). Obtain a list of the rating group IDs from your OCS documentation and follow the instructions in this section to add them to Policy Controller to use in PCC profiles.

To create rating group IDs:

1. Start Policy Designer.
2. Click the **Policy Charging and Control Profiles** tab.
3. Click **Create**.
4. Click the pencil icon next to the **Rating Group ID** box.

   The **Rating Group** window appears.
5. Click the **Create New** (paper and plus) icon.

   The **Rating Group ID** and **Rating Description** fields appear.
6. Enter the integer that represents the rating group ID in the **Rating Group ID** field.
7. Enter an informal description in the **Rating Description** field.
8. Repeat steps 4 through 7 for each rating group you will use.
9. Click **OK** to save the changes and return to the **Policy Charging and Control Profiles** tab.

Creating Service IDs to Use in PCC Profiles

Obtain a list of these IDs from your OCS documentation and follow the instructions in this section to add them to Policy Controller to use in PCC profiles. No service IDs are defined in Policy Controller by default because each Online Charging System (OCS) uses a different set of integers to identify services or service components.

To create service IDs:

1. Start Policy Designer.
2. Click the **Policy Charging and Control Profiles** tab.
3. Click **Create**.
4. Click the pencil icon next to the **Service ID** box.

   The **Service** window appears.
5. Click the **Create New** (paper and plus) icon.

   The **Service ID** and **Service Description** fields appear.
6. Enter the integer that represents the rating group ID in the **Service ID** field.
7. Enter an informal description in the **Service Description** field.
8. Repeat steps 4 through 7 for each rating group you will use.
9. Click **OK** to save the changes and return to the **Policy Charging and Control Profiles** tab.

Creating a Dynamic PCC Profile

This section explains how to create dynamic PCC profiles that are called by your rules to specify quality of service levels and charging information.

To create a dynamic PCC Profile:

1. Start Policy Designer.
2. Navigate to **Policy Charging and Control Profiles**.
3. Click **Create**.

   The PCC profile detail fields appear.
4. Enter a unique name in the **PCC Profile Name** box.
5. Enter an integer in the **Precedence** box (mandatory for dynamic profiles; not included in predefined profiles) to set the PCC profile evaluation order. See "Planning Your PCC Profiles" for details on the precedence levels.
6. Confirm that **Dynamic Profile** is selected in the dialog box.
7. Set a time limit for the PCC profile to be active:
   - To make the new PCC profile effective immediately and indefinitely, leave the **Always Active** check box checked.
   - To limit the PCC profile to a specific time period deselect the **Always Active** check box and enter start and stop dates and times:
     - **Activation Time**: Select the **CEST** time zone icon and select a time zone from the **Select a Time Zone** list. Then select the calendar/cloak icon to display the **Select Date and Time** date editor. Select a date and time for the ADC profile to activate. Click **OK** to make your choice take effect.
     - **Deactivation Time**: The time zone you set for the **Activation Time** is automatically also applied to the **Deactivation Time**. However, you can set a different time zone for Deactivation Time if your implementation requires it.
8. Specify the **QoS** parameters:
   a. **QCI** - QoS Class Identifier type as defined in the *Policy Charging and Control Architecture 3GPP TS 23.203 v9.90 (2011-06)* specification.

      Can be one of the default (specified) QCI values listed in Table 8–1, or a new one that you create.

Table 8–1 Default QCI Values

QCI Value	Priority	Guaranteed Bit Rate?	Typical Service
1	2	Yes	Conversational voice.
2	4	Yes	Live video streaming.
3	3	Yes	Real time gaming.
4	5	Yes	Buffered video streaming.
5	1	No	IMS Signalling.
6	6	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).

Table 8–1 (Cont.) Default QCI Values

QCI Value	Priority	Guaranteed Bit Rate?	Typical Service
7	7	No	Voice, live video streaming, interactive gaming.
8	8	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).
9	9	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).

b. Specify a gate status from the **Gate Status** menu.

The gate status specifies the PCEF action (gate open or closed) when service data flow traffic arrives from Policy Controller. Can be one of the following:

- **Uplink Enabled** - Allows data uploading.
- **Downlink Enabled** - Allows data downloading.
- **Both Enabled** - Allows both data uploading and downloading.
- **Both Disabled** - Disallows both data uploading and downloading.
- **Removed** - This option is reserved for Oracle use.

c. Set the Maximum Bandwidth options:

- **Maximum Bandwidth - Upload** - Set a maximum upload data rate allowed by entering an integer in the dialog box and selecting a bit rate from the bit rate list.
- **Maximum Bandwidth - Download** - Set a maximum download data rate allowed by entering an integer in the dialog box and selecting a bit rate from the bit rate list.

d. Set the Guaranteed Bandwidth options:

- **Guaranteed Bandwidth - Upload** - Set a minimum download data rate to use by entering an integer in the dialog box and selecting a bit rate from the bit rate list.
- **Guaranteed Bandwidth- Download** - Set a minimum download data rate to use by entering an integer in the dialog box and selecting a bit rate from the bit rate list.

9. Specify **Charging** information:

a. Select a **Rating Group ID** from the list.

See "Creating Rating Group IDs to Use in PCC Profiles" for instructions on how to add rating group IDs if necessary.

b. Select a **Service ID** from the list.

See "Creating Service IDs to Use in PCC Profiles" for instructions on how to add service IDs if necessary.

c. Select a **Metering Method** - Can be one of the following

- **Duration** - Charges based on the amount of connect time (session-based).
- **Volume** - Charges based on the amount of data transferred.

- **Duration and Volume** - Charges for both the connect time and amount of data transferred.

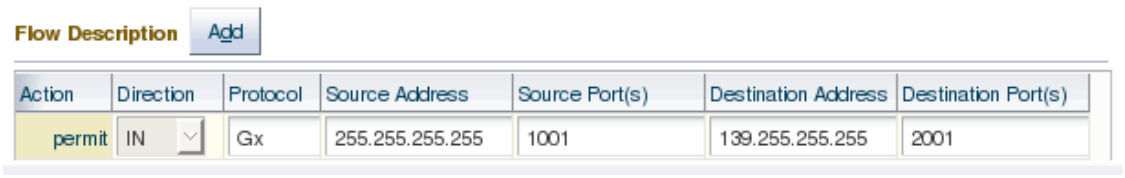
d. Select an **Online Charging** option - **Yes** specifies service data flow using on-line charging; **No** ignores service data flow using online charging.

e. Select an **Offline Charging** option - **Yes** specifies service data flow using offline charging; **No** ignores service data flow using offline charging.

10. Specify a Flow Description:

a. Click the Flow Description **Add** button to display the **Flow Description** dialog box shown in Figure 8–3:

Figure 8–3 Flow Description Dialog Box



b. **Direction** - IN or OUT. Applies the profile to service data flow coming in to, or being sent out from Policy Controller.

c. **Protocol** - (Mandatory) Filters the service data flow by protocol. Applies the profile to service data flow matching the protocols you specify.

d. **Source Port(s)** - A comma-separated list of integers that specify ports on the Source Address. Applies the profile to service data flow from the ports specified from the Source address. Dash-separated ranges are also allowed. For example:

```
2000,2002,4010-4020
```

e. **Destination Address** - (Mandatory) IP/mask number. Applies the profile to service data flow destined for entities specified by an IP address/subnet mask.

f. **Destination Port(s)** - A comma-separated list of integers specify ports on the Accentuation Address. Applies the profile to service data flow destined for the ports you list at the Destination Address. Dash-separated ranges are also allowed.

11. Click **OK**.

The new PCC profile appears in the **Policy Charging and Control Profiles** tab.

Creating a Predefined PCC Profile

This section explains how to create predefined PCC profiles that reference PCC rules that are already defined and stored in your PCEF.

This section assumes that you have obtained the names and Charging-Rule-Name AVP name(s) of any predefined PCC rules that you are using.

Follow these steps to create a predefined PCC Profile:

1. Start Policy Designer.
2. Navigate to **Policy Charging and Control Profiles**.
3. Click **Create**.

The PCC profile detail fields appear.

4. Enter the name of the PCC rule stored in your PCEF in the **PCC Profile Name** box.
5. Select the **Predefined Profile** setting in the dialog box.
6. Select either **Rule Name** or **Rule Group Name** from the dialog box depending on whether the PCC profile is a base rule or contains a group of PCC profiles.
7. Enter the name of the PCC profile in the dialog box.
8. Set a time limit for the PCC profile to be active:
   - To make the new PCC profile effective immediately and indefinitely, leave the **Always Active** check box checked.
   - To limit the PCC profile to a specific time period deselect the **Always Active** check box and enter start and stop dates and times:
     - **Activation Time**: Select the **CEST** time zone icon and select a time zone from the **Select a Time Zone** list. Then select the calendar/cloak icon to display the **Select Date and Time** dialog box. Select a date and time for the ADC profile to activate. Click **OK** to make your choice take effect.
     - **Deactivation Time**: Select the **CEST** time zone icon and select a time zone from the **Select a Time Zone** list. Select the calendar/clock icon to display the **Select Date and Time** dialog box. Select a date and time for the PCC profile to deactivate. Click **OK** to make your choice take effect.
9. Click **OK**.

   The new PCC profile appears in the **Policy Charging and Control Profiles** tab.

Changing the PCC Profile Table Display

You can change the list of PCC profiles columns or rearrange them using the **View** menu on the upper left of the **Policy and Charging Control Profiles** tab.

To change how PCC profiles are displayed:

1. Start the Policy Designer.
2. Create PCC Profiles or import a deployment that includes PCC profiles.
3. Select an item from the **View** list:
   - The **Columns** item displays a list of all columns. Select or deselect individual columns to display/hide them.
   - The **Reorder Columns** item displays the **Reorder Columns** box. Select columns and move them up or down using the arrows on the right side of the box. Then click **OK** to make your changes take effect.

Filtering PCC Profiles by Text, Bandwidth, or Date

You can limit the list of PCC profiles displayed in the **Policy Charging and Control Profiles** tab by entering parameters in the **Filter** field at the upper left side of the Policy and Charging Control tab.

At any time you can select the eraser icon to remove your filtering parameters and return to the default PCC profile display.

1. Start the Policy Designer.
2. Create PCC Profiles or import a deployment that includes PCC profiles.

3. Select **Text**, **Bandwidth**, or **Date** from the **Filter** list.
4. Enter parameters for the filter:
   - **Text** - As you type the list of PCC profiles is filtered by the text you enter.
   - **Bandwidth** - When you select **Bandwidth**, bandwidth dialog boxes appear. As you enter bandwidth limits, the list of PCC profiles is filtered by those parameters.
   - **Date** - When you select **Date**, the calendar/clock icons appear. As you select dates and times, the list of PCC profiles is filtered by the dates you choose.

Deleting PCC Profiles

You cannot delete a PCC profile until you have removed that profile from any rules that reference it. Ensure that the **Profile/Rule Relationships** column is empty for a PCC profile before you attempt to delete it Hover your cursor over that column to find the names of rules that reference the PCC profile.

To delete a PCC profile:

1. (If necessary) Start Policy Designer.
2. (If necessary) Import the deployment containing the PCC profile to delete.
3. Navigate to the **Policy Charging and Control Profiles** tab.
4. Ensure that the **Profile/Rule Relationships** column for the PCC profile is empty.

   Hover your cursor over a table cell in this column to find the names of the rules that reference the PCC profile. Remove any remaining rule references.
5. Select the PCC profile.
6. Click **Delete**.

9

Creating Application Detection and Control Profiles

This chapter explains how to create Oracle Communications Service Broker Policy Controller (Policy Controller) Application Detection and Control (ADC) profiles that specify Quality of Service (QoS) limits and application redirection information for services.

This chapter assumes that you know the name of any predefined (PCEF-based) ADC rules that you must reference, including the ADC-Rule-Name AVPs to use.

About ADC Profiles

ADC profiles specify QoS limits and any application redirection information for service data flow traffic that your rules select. Your Policy Controller rules actually specify the application traffic that ADC profiles apply to. The ADC profile then applies QoS limits, and if redirection is used, the Web location to where the subscriber is redirected.

None of these options are required for a ADC profile to function; you can select the set of options that best suit your implementation.

ADC profiles can be either be dynamic (created within Policy Controller) or predefined (created and stored in your PECF).

ADC profiles are included in deployments along with your rulesets and PCC profiles. You can deploy, export, and re-import them as **.rap** files.

You use the Policy Designer **Application Detection and Control Profile** tab to manage ADC profiles.

Each ADC profile contains:

- Activation/deactivation dates and times that specify how long the ADC profile is in force.
- QoS limits, including:
  - A QoS Class Identifier (QCI) that specifies the type of traffic. There are nine specified options, or you can create your own.
  - The actual upper and lower bandwidth limits in bps, kpbs, Mbps, or Gbps.
  - An uplink status that controls whether uplink or downlink or both are allowed.
- Redirect Information, including:
  - The type of Web location (IPV4, IPV6, URL or SIP-URL)

  - The address to where the server subscribers are redirected.
  - A switch that turns the redirection options on and off.

You may also use any ADC profiles stored in your PCEF with the Policy Designer (predefined PCC rules). To use a predefined PCC rule in Policy Designer, simply create a new ADC profile with just the predefined PCC rule's name as it appears in the PCEF and select **Predefined Profile** from Apply Predefined Profiles/Apply Dynamic Profiles menu box. Predefined ADC profiles are marked with a check mark in the **Predefined Profiles** column of the **Application Detection and Control Profiles** tab.

Planning Your ADC Profiles

The procedures in the following section assumes that you already know the details of the ADC profiles to create, such as:

- The QoS limits to set.
- Redirection details including:
  - The internet protocol used by the redirection target.
  - The server address of the redirection target. If no address is specified, the default redirection set in the PCEF (if any) is used.

Creating an ADC Profile

To create an ADC Profile:

1. Start Policy Designer.
2. Navigate to **Application Detection and Control Profiles**.
3. Click **Create**.

   The ADC profile detail fields appear.
4. Enter a unique name for the ADC profile.
5. Select either **Dynamic Profile** or **Predefined Profile** from the dialog box.
6. Set a time limit for the ADC profile to be active:
   - To make the new ADC profile effective immediately and indefinitely, leave the **Always Active** check box checked.
   - To limit the ADC profile to a specific time period deselect the **Always Active** check box and enter start and stop dates and times:
     - **Activation Time**: Select the **CEST** time zone icon and select a time zone from the **Select a Time Zone** list. Then select the calendar/cloak icon to display the **Select Date and Time** date editor. Select a date and time for the ADC profile to activate. Click **OK** to make your choice take effect.
     - **Deactivation Time**: The time zone you set for the **Activation Time** is automatically also applied to the **Deactivation Time**. However, you can set a different time zone for Deactivation Time if your implementation requires it.

     Your activation and deactivation times can be set in different time zones
7. Specify the **QoS** parameters:
   a. **QCI** - QoS Class Identifier type as defined in the 3GPP TS 23.203 v9.90 (2011-06) standard. See the standard for details:

http://www.3gpp.org/ftp/Specs/html-info/23203.htm

Can be one of the default QCI values listed in Table 9–1, or any new ones that you create. Multiple QCI values can be used for the same services.

Table 9–1 Default QCI Values

QCI Value	Priority	Guaranteed Bit Rate?	Typical Service
1	2	Yes	Conversational voice.
2	4	Yes	Live video streaming.
3	3	Yes	Real time gaming.
4	5	Yes	Buffered video streaming.
5	1	No	IMS Signalling.
6	6	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).
7	7	No	Voice, live video streaming, interactive gaming.
8	8	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).
9	9	No	Buffered video streaming, TCP-based services (for example, email, chat, FTP, p2p file sharing, progressive video).

b. **Gate Status** - Specifies the PCEF action (gate open or closed) when service data flow traffic arrives from Policy Controller. Can be one of the following:

- **Uplink Enabled** - Allows data uploading.
- **Downlink Enabled** - Allows data downloading.
- **Both Enabled** - Allows both data uploading and downloading.
- **Both Disabled** - Disallows both data uploading and downloading.
- **Removed** - This option is reserved for Oracle use.

c. Set the Maximum Bandwidth options:

- **Maximum Bandwidth - Upload** - Set a maximum upload data rate allowed by entering an integer in the dialog box and selecting a bit rate from the bit rate list.
- **Maximum Bandwidth - Download** - Enter maximum download data rate allowed by entering an integer in the dialog box and selecting a bit rate from the bit rate list.

d. Set the Guaranteed Bandwidth options:

- **Guaranteed Bandwidth - Upload** - Set a minimum download data rate to use by entering an integer in the dialog box and selecting a bit rate from the bit rate list.
- **Guaranteed Bandwidth- Download** - Enter a minimum download data rate to use by entering an integer in the dialog box and selecting a bit rate from the bit rate list.

8. Set the **Redirection Information** parameters:

a. Select an internet protocol for the redirection target from the **Type** list.

b. Enter the server IP address of the redirection target in the **Server Address** dialog box. If you do not specify an address, your PCEF uses it's own default redirection address.

c. Select a setting from the **Support** list to enable/disable the redirection. The default blank setting does not send the redirection AVP (**Redirect-Information**); **Redirection Enabled** sends the redirection AVP, and **Redirection Disabled** sends the redirection AVP but instructs the PCEF to not redirect traffic for the profile.

d. Click **OK**.

Changing the ADC Profile Table Display

You can change the list of ADC profiles columns or rearrange those columns using the **View** menu on the upper left of the **Application Detection and Control Profiles** tab.

To change the ADC profile display:

1. Start the Policy Designer.
2. Create ADC Profiles or import a deployment that includes ADC profiles.
3. Select an item from the **View** list:
   - The **Columns** item displays a list of all columns. Select or deselect individual columns to display/hide them.
   - The **Reorder Columns** item displays the **Reorder Columns** box. Select columns and move them up or down using the arrows on the right side of the box. Then click **OK** to make your changes take effect.

Filtering ADC Profiles by Text, Bandwidth, or Date

You can limit the list of ADC profiles displayed in the **Application Detection and Control Profiles** tab by entering parameters in the **Filter** field at the upper left side of the Policy and Charging Control tab.

At any time you can select the eraser icon to remove your filtering parameters and return to the default ADC profile display.

1. Start the Policy Designer.
2. Create ADC Profiles or import a deployment that includes ADC profiles.
3. Select **Text**, **Bandwidth**, or **Date** from the **Filter** list.
4. Enter parameters for the filter:
   - **Text** - As you type the list of ADC profiles is filtered by the text you enter.
   - **Bandwidth** - When you select **Bandwidth**, bandwidth dialog boxes appear. As you enter bandwidth limits, the list of ADC profiles is filtered by those parameters.
   - **Date** - When you select **Date**, the calendar/clock icons appear. As you select dates and times, the list of ADC profiles is filtered by the dates you choose.

Deleting ADC Profiles

You cannot delete an ADC profile until you have removed that profile from any rules that reference it. Ensure that the **Profile/Rule Relationships** column is empty for a ADC profile before you attempt to delete it. Hover your cursor over that column to find the names of rules that reference the ADC profile.

To delete an ADC profile:

1. Start Policy Designer.
2. (If necessary) Import the deployment containing the ADC profile to delete.
3. Navigate to the **Application Detection and Control Profiles** tab.
4. Ensure that the **Profile/Rule Relationships** column for the ADC profile is empty.

   You can hover your cursor over this table cell to find the names of the rules that reference the ADC profile. Remove any remaining rule references.
5. Select the ADC profile.
6. Click Delete.

10

Strategies for Creating Rules

This chapter provides information that is useful for planning and creating Oracle Communications Service Broker Policy Controller (Policy Controller) Policy Designer rules. These rules are the if-then statements that you use to determine which Policy Plans (bandwidth limits or requirements) individual subscribers are entitled to.

See "About Rules" for an overview of rules, and "Creating and Deleting Rules" for instructions on how to create rules.

About Creating Policy Controller Rules

You use Policy Controller rules to select which PCC and ADC profiles to apply to a subscriber in a specific context. This chapter contains conceptual information on, and examples of how to create rules that take advantage of the flexibility that Policy Controller offers. For example, this chapter explains how Policy Controller implicitly removes profiles if their condition tests no longer apply.

See "Creating Rules and Rulesets" for the step-by-step tasks that you perform to create rules, lists of values, and rulesets.

Understanding How Policy Controller Makes Policies Take Effect

Within a session, Policy Controller makes both PCC and ADC profiles take effect when the conditions that activate them are true, and removes them implicitly when the condition no longer exists. For example, within a session assume you have a rule that states: if condition A exists, then make profile 1 take effect. In this case profile 1 remains in effect if condition A exits within that session. If during the next event processing condition A no longer exists, Policy Controller automatically stops profile 1 from taking effect. There is no need to explicitly remove profiles if the conditions that make them take effect are removed.

Note: Rules created in Policy Controller release 6.0 are incompatible with this release.

You can however, create rules that (like the release 6.0 rules) do not use the default implicit profile removal strategy. To create and use rules that *explicitly* remove profiles, you must first use the instructions in "(Optional) Supporting Explicit Rule Removal" to configure Policy Controller to do so. Once configured, you use the **Remove_PCCProfile** action in rules to explicitly remove PCC or ADC profiles.

Selecting Among Subscriber Sessions

By default, Policy Controller applies any rule actions to the current active Gx session. This section explains how to use a different criteria to select a target session for the action.

Typically subscribers only have a single session running at a time, which makes specifying an output fact (action) for that session straightforward. In cases where a subscriber has multiple simultaneous sessions running, and you want to perform comparisons and actions across sessions, you can select one as the target for the Policy Controller action using the **Properties** window.

The **Properties** window is available once you select the **Show Gx/Rx Session Assignment** from the **Advanced** menu. Each Properties window is automatically populated with choices based on the individual rule's actions and output facts.

Figure 10–1 shows an example Properties window. This figure shows that this rule uses the **name**, **Gx**, and **Rx** properties. The only criteria entered is the **"Gold Profile"** value in the **name** property. So if the subscriber has multiple sessions open, the rule action is applied to the one associated with the "Gold Profile" PCC rule.

Figure 10–1 Properties Window

To select among multiple subscriber sessions:

1. Start the Policy Designer.
2. Select the **Rules** tab
3. Select a rule.
4. Select **Show Gx/Rx Session Assignment** from the **Advanced** menu on the upper left of the Rules tab.
5. Select the **xyz** properties icon in the rule.

   The **Properties** menu appears, populated with the properties options available based on the rule attributes.
6. For each row, select values to filter the session selection.

   Use the magnifying glass icons to display the available values for each property.

   Use the **Constant** check boxes to limit the display to constant values only.
7. Click **OK** to confirm your choices and return to the **Rule** tab.

Creating Rules Using OCS Subscriber Thresholds

Rules that use a subscriber's quota or usage-based counters are among the most flexible and popular types that you can create with the Policy Designer. These rules are

generally designed to cause Policy Controller to switch a subscriber to a different PCC profile when that subscriber reaches a specific numeric threshold. For example, when they exhaust a data quota, or reach a credit limit.

One important thing to remember when using these metrics is that Policy Controller will probably not be alerted the instant a subscriber counter reaches a certain threshold. Your OCS must update Policy Controller before it can reinterpret the rules and decide whether to apply a different profile. Consequently, you should design rules that assess the threshold levels in question as their first action, and at each significant action thereafter. For example first check whether a subscriber has an **active** status, or sufficient account balance before allowing them access to a service.

You generally use the Sy reference point facts available in the Policy Designer **Rules** tab to create these threshold rules. These AVPs from the Sy specification are available to you as rule facts:

- **Policy-Counter-Identifier**
- **Policy-Counter-Status**

Policy controller also offers you these facts that are not in the Sy specification:

- **status**
  - **remainingQuota**
  - **usageCounter**
- **total**
  - **remainingQuota**
  - **usageCounter**

Here is a subscriber threshold example:

Assume that a subscriber is allowed 6 Gb of data transfer per month. You probably want to write rules that perform some action when the subscriber reaches that 6 Gb limit. Typically you would charge the subscriber an additional fee for the over-limit data. However, to avoid bill shock let's assume that you also want to warn the subscriber when they have used 80% of their quota and that they will be charged more when they reach 101%. You can implements this example using this rule framework:

```
IF Sy.policyCounterStatus == "WITHIN_QUOTA"
THEN install a PCC profile that allows data tranfer

IF Sy.policyCounterStatus == "80_PERCENT_QUOTA"
THEN send the subscriber an SMS warning them of the impending fee increase

IF Sy.policyCounterStatus == "OVER_QUOTA"
THEN install a PCC profile that charges more for OVER_QUOTA data transfer than
WITHIN_QUOTA data transfer.
```

This example assumes that you:

- Configured your OCS to update your SPR with quota usage data.
- Set up your OCS to use WITHIN_QUOTA, 80_PERCENT_QUOTA, and OVER_QUOTA, and map them to the SPR statuses.
- Directed your SPR to update Policy Controller when the data usage reached 80% and 101% of quota.

It necessarily follows then that in order create rules like the ones in this example, you must know what the OCS data model is so you can map them to these Sy facts.

The chapter on using the PCP profile provider in *Service Broker Subscriber Store User's Guide* describes the Service Broker SPR data model and the mapping between the SPR and Oracle Communications Billing and Revenue Management. The information in that chapter helps you configure the Service Broker SPR, and the Oracle Communications Billing and Revenue Management if you are using it for an SPR. If you use a 3rd-party SPR as explained in "Configuring Subscriber Profile and Charging Information", see the third-party documentation for the data map of that SPR.

Redirecting Users to a URL

You use the **Redirect** condition from the Condition Browser in the THEN portion of a rule to redirect subscribers to a URL of any kind. Click Edit Properties to start a **Properties** window. Add the URL to redirect subscribers to in the **Value** field of the **Properties** window.

Using Vendor-specific and Default Gx Event Triggers to Reinterpret Rules

Policy Controller supports the Gx **Event-Trigger** AVP that you use to specify IP-CAN session modifications or other specific events that cause Policy Controller to reinterpret rules. Policy Controller supports all the events specified in the 3GPP 29.212 Gx Specification. You may either use the default specified values for the **Event-Trigger** AVP, or you may add custom values to this list as required by your implementation. You must first populate the Policy Controller with a complete list of events to use, or your own custom events to use.

You specify event triggers in rules by selecting **Gx.Event-Trigger** left value of an IF statement, using the **containsEventTrigger** condition, and then selecting an event trigger from list in the right value of the rule. See "Using an Event Trigger to Change a PCC Profile" for an example.

See "Supported Gx Event-Trigger Event Values" for a list of the default Gx event triggers supported by Policy Controller.

See "Add any Vendor-specific Values for Event-Trigger to Use as Event Triggers" for details on how to add your own event triggers to the default list, and see *Oracle Communications Policy Controller PICS* for a list of the supported Gx **Event-Trigger** AVP values.

Redirecting Service Data Flow for a Session or Individual Service

Session redirection enables you to redirect Gx service data flow to a URL or other Web address. Redirection is implemented using the Gx CCR and RAR messages. The redirected session is not terminated, so the subscriber can return to using services after visiting the address. The address can be any Web address, such as a simple informational note, an offer for a different service such as topping-up accounts or authorizing a faster bandwidth, or anything else your implementation requires.

You have these options for redirecting sessions:

- *Global redirection* that you use to redirect all traffic for a session to a single Web address using the **Redirect** rule action.
- *Service level redirection* that you use to redirect individual services within a session to a Web address to which a PCC profile applies, using the **RedirectPCCProfile** rule action.

Note: To redirect all traffic for a specific *application*, use an ADC profile. See "Creating Application Detection and Control Profiles" for details on creating ADC profiles.

The redirected sessions are protected by the default revalidation timers. See "Configuring the Policy Controller Session Guard Timers" for details on setting the revalidation timers.

Both types of session redirection support these addresses types:

- URL
- IPv4
- IPv6
- SIP URI

You can enter individual redirection addresses in the rules, or create aliases for the addresses you use frequently. See "Creating Aliases for Redirection Target Addresses" for details on setting up address aliases.

The following sections explain how to set up and use session redirection.

Globally Redirecting All Services in a Session

Global redirection redirects all service data flows for a session to an address that you specify.

To globally redirect a session, specify a Gx session type in the conditions section of a rule, and then assert a **Redirect** action and select values for these parameters in the **Properties** window that appears:

- address (string) - A Web address to redirect the session to. You can enter an address or an alias from a list of values that you set up beforehand. See "Creating Aliases for Redirection Target Addresses" details on setting up URL aliases.
- addressType (Enum) - The Web address type. One of **URL**, **SIP_URI**, **IPv4**, or **IPv6**.

Global redirection example:

```
IF Gx.IPcantype == IPCanType._3GPP_GPRS
THEN
assert Redirect (address:www.yourco.com/topup1,
addressType:RedirectAddressType.URL)
```

Redirecting Individual Services Inside a Session

Service-based redirection redirects a single service data flow within a Gx session to a Web address and applies a new PCC profile. This redirection requires that you have a PCC profile to apply to the service, so create it before you create the redirection rule.

To redirect a single service data flow, specify a Gx session type in the conditions part of the rule, then assert a **RedirectProfile** action, and enter values for these parameters in the **Properties** window that appears:

- policyProfileName (String) - A PCC profile to apply to the service.

- address (string) - A URL to redirect the session to. You can enter an address or an alias from a list of values that you set up beforehand. See "Creating Aliases for Redirection Target Addresses" details on setting up URL aliases.
- addressType (Enum) - The URL address type. One of **URL**, **SIP_URI**, **IPv4**, or **IPv6**.

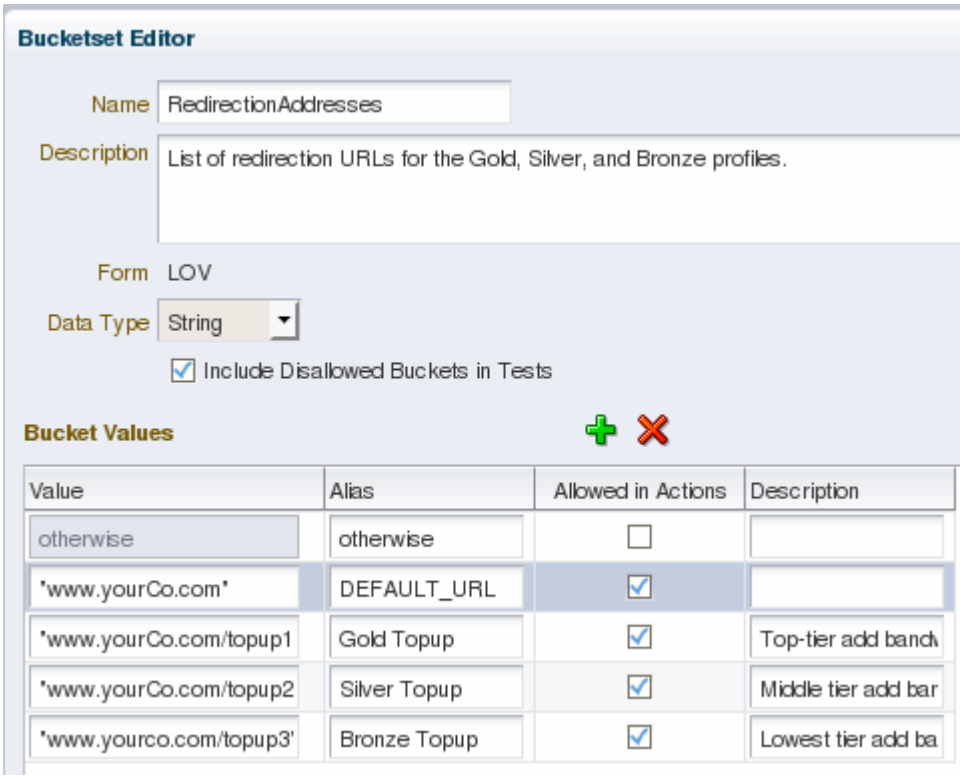
Service-based redirection example:

```
IF Gx.IPcantype == IPCanType.WIMAX
THEN
assert Redirect (profileName:"Silver Profile", address:www.myco.com,
addressType:RedirectAddressType.URL, Gx:Gx)
```

Creating Aliases for Redirection Target Addresses

If your list of redirection addresses is relatively static, use the Policy Designer **List of Values** feature to create a list of address aliases to use as redirection targets in rules. Figure 10–2 shows the **List of Values** window that you use to create a list of addresses. In this example the **www.yourCo.com** website has the **topup1**, **topup2**, and **topup3** pages that are available to redirect subscribers to. The idea is to use different topup pages depending on which tiered service the subscriber is using.

Figure 10–2 Example List of Topup URLs



To create a specific list of URLs for redirection:

1. Start Policy Designer interface
2. Navigate to **Rules** tab
3. Click **List of Values**.

The **List of Values** screen appears.

4. Click **RedirectionAddresses** from the list and then the Edit List of Values (pencil) icon.

   The List of Values Editor window appears.

5. Click the Add Values (green cross) icon.

   A row for the new value appears in the table.

6. Fill in the new row:

   - **Value** - enter the new URL.
   - **Alias** - Enter a string to use in your rules. The name should be short but meaningful to those creating the rules.
   - **Allowed in Actions** - Checking this box allows the value to be used in the actions (THEN) portions of a rule. If unchecked, your values are only allowed in the conditions (IF) section of a rule.
   - **Description** - Enter an informal description of the new URL.

7. Repeat step 6 for each address that you want to use in your rules.

8. Click OK.

Using Rules to Send SMS Messages

Policy Designer enables you to send an SMS message as part of the action of a rule. You do this by first setting up a test for any fact available in the IF portion of the rule, and then using the **SendSMS** action in the THEN portion of the rule. The test is performed each time a subscriber creates a session and thereafter any time your rules are reinterpreted.

Policy Controller adheres to the Interface protocols for the connection of Short Message Service Centres (SMSCs) to Short Message Entities (SMEs) (Release 5) 3GPP TR 23.039 V5.0.0 (2002-06) specification for SMS messages.

You can create rules that send SMS messages based on any changed fact that causes rules to be reinterpreted, such as:

- Simply wishing the subscriber a happy birthday.
- Alerting a subscriber that they have changed location and are about to incur roaming charges.
- Warning a subscriber that they have reached a monetary threshold and offering to top-up their account.
- Alerting a subscriber that they have reached their monthly bandwidth usage limit and are being charged at a higher rate for the remainder of the period.
- Alerting a subscriber that they have exceeded their download usage threshold and their service is being throttled back.
- Alerting your security personnel of suspicious activity on behalf of a subscriber.

Note: The rule test for sending an SMS (or any other action) is performed each time a subscriber starts a session and perhaps more often. So for example a subscriber would receive a birthday greeting every time they log in on their birthday.

The test is performed each time the subscriber logs in. So each time they log in on their birthday they receive the birthday greeting SMS.

The SMS can be a simple text message or it can be a part of a rule that automatically changes a subscriber's PCC profile. The SMS can contain a URL that directs the subscriber to a website for more information, or to a Web portal they use to change their account details. You can also probe for additional facts and send different SMS messages within a single session.

You can send the SMS to one, or any number of MSIDN numbers. The SMS can alert an entire group that its bandwidth is used up.

The SMS messages are sent in "fire and forget" mode. That is, Policy Controller does not wait for or record any response from the MSISDN receiving the SMS.

See "(Optional) Configuring Policy Controller to Send SMS Messages" for details on configuring this feature.

Configuring Policy Controller to Send SMS Messages

Before you can add SMS messages to rules you must first configure Policy Controller as an External Short Messaging Entity (ESME). You do this by configuring a Short Message Service Center (SMSC) in the SSU SMPP. See the discussion on configuring an SMPP signaling server unit in *Service Broker Signaling Server Units Configuration Guide* for details.

Adding SMS details to a Rule

You use the **SendSMS** function in the **THEN** portion of your rule to create an SMS message. Select **Assert New** then **SendSMS,** and then click the Properties icon. The **Properties** screen appears with fields for:

- **Source Number** - The number of the SSU SMPP SMSC you have configured to send SMS messages. The number sending the SMS, for example `"11111"`
- **Destination Number** - A comma-separated list of one or more MSISDN numbers that specify the SMS recipients. For example `"11111,22222,44444"` sends this message to the numbers `11111`, `22222`, and `44444`.
- **Message** - The text of the message. The SMS supports only Latin-based characters. Enter text, or select the search icon to include Sp and Sy facts, and use **Redirect.Address** to include a URL. For example `"Your license has expired. Please add resources to your account."`

You can fill in these fields with text and click the **Constant** check box, or select the search icon to use dynamic data (rule facts) in the SMS.

After filling in these text fields, click **OK** to save your changes. The text of the SMS is displayed in the Rule Editor window next to the **Edit Properties** icon.

Using Custom Diameter AVPs in Your Rules

The Service Broker Diameter stack supports the Diameter AVPs listed in the *Oracle Communications Service Broker Policy Controller Protocol Implementation Conformance Statement*. However, many Policy Controller nodes such as Policy Controller Enforcement Functions (PCEFs), Application Functions (AFs), and so on, use AVPs from protocols that Policy Controller does not support. Service Broker includes the ability to add these custom AVPs to its Diameter stack. This allows Policy Controller to

accept these AVPs in messages, but it does not do anything with their contents by default.

After adding custom AVPs to Service Broker, you must create rules that use their content. Before adding a custom AVP, you first need to plan the kinds of IF-THEN rule tests required for using the new AVP.

See "Adding Custom Diameter AVPs" for details on adding custom AVPs to Service Broker for Policy Controller to use, and also the data types available and the supported functions for the new AVPs.

Example Rules

Because of the flexibility and extensibility of the Policy Controller rules engine, you can create any number of rules and rule strategies. The following examples show a few of the most common scenarios.

Using Subscriber Data to Change a PCC Profile

A common rule task is to install a PCC profile based on a change in subscriber data.

For example, the Policy Controller can regularly compare the current date to the month and date of the subscriber's birthday and install a special PCC profile on the subscriber's birthday:

```
IF
Sp.dateOfBirth.month is SystemVariables.currentTime.month
AND
Sp.dateOfBirth.dayOfMonth is SystemVariables.currentTime.dayOfMonth
THEN
InstallPCCProfile (name: "BirthDay")
```

Applying a New Service to an Existing Service

This rule specifies that subscribers using **Prepaid** PCC profile services from a **69.63.189.*** IP address also receive the **Prepaid_SocialVoice_Plan** PCC profile:

```
IF
Sp.accountType is "PREPAID
AND
Gx.TFT-Packet-Filter-Information matchToAddress "69.63.189.*")
THEN
InstallPCCProfile (name: "Prepaid_SocialVoice_Plan")
```

Applying a Profile for Part of a Day

This example creates a "happy hour" rule that applies a PCC profile when the current time is between 18:00 and 20:00 each day:

```
IF
Gx.MS-TimeZone_3GPP.hourOfDay is less than 18
THEN
setAbsoluteRevalidatonTime.date to (date:getCalendarAtHour(18))

IF
Gx.MS-TimeZone_3GPP.hourOfDay is same or more than 18
AND
Gx.MS-TimeZone_3GPP.hourOfDay is less than 20
THEN
InstallPCCProfile HAPPYHOUR
```

```
AND
setAbsoluteRevalidationTime.date to (date:GetCalendarATHour(20))

If Gx.MS-TimeZone_3GPP.hourOfDay is greater than or equal to 20
THEN
setAbsoluteRevalidationTime.date to (date:GetCalendarAtHour(18))
```

This ruleset first checks whether the current time is earlier than 18:00, and if so it directs Policy Controller to revalidate its rules at 18:00.

If the ruleset finds that the current time is between 18:00 and 20:00, it applies a PCC profile called **HAPPYHOUR**, and directs Policy Controller to reapply its rules when the current time reaches 20:00. Once the rules are reapplied and the time is not between 18:00 and 20:00, the **HAPPYHOUR** profile is no longer applied.

If the current time is later than 20:00, it directs Policy Controller to revalidate its rules when the current time reaches 18:00 the next day.

This ruleset requires the use of the **GetCalendarAtHour** function. See "Using the Expression Builder" for details on using the Expression Builder to select a function.

Using an Event Trigger to Change a PCC Profile

Receipt of a trigger from the PCEF can initiate a change in a PCC profile.

For example, upon receiving the **OUT_OF_CREDIT** trigger, Policy Controller can remove a subscriber's installed PCC profile and install another one:

```
IF
GxEventTrigger containsEventTrigger EventTrigger.OUT_OF_CREDIT)
AND
Sp.subscriberCategory in "GOLD" , "BRONCE" , "SILVER"
THEN
InstallPCCProfile (name: "LEAD")
```

Throttling Back QoS When Credit Expires

This example shows two rules that would work to throttle back service for a subscriber that has run out of credit. These rules specify that when the Policy Controller receives a Gx-based **OUT_OF_CREDIT** event, a **NoCredit_Plan** PCC profile is applied which contains throttled back service.

This rule applies a **NoCredit_Plan** PCC profile when an **OUT_OF_CREDIT** event is received:

```
IF
Gx.Event-Trigger containsEventTrigger EventTrigger.OUT_OF_CREDIT
AND
Gx.RAT-Type is RATType.HSPA_EVOLUTION
THEN
InstallPCCProfile (name: "NoCredit_Plan")
AddEventTriger(eventTrigger:EventTrigger.REALLOCATION_OF_CREDIT )
```

This rule removes the **NoCredit_Plan** PCC profile when a **REALLOCATION_OF_CREDIT** event is received indicating that the subscriber has.

```
Gx.Eevent-Trigger containsEventTrigger EventTrigger.REALLOCATION_OF_CREDIT
AND
Gx.RaTType is RATType.HSPA_EVOLUTION
THEN
RemovePCCProfile (name:"NoCreditPlan")
```

```
AddEventTrigger(eventTrigger:EventTrigger.OUT_OF_CREDIT)
```

Using a Local Fact to Apply a PCC Profile

You can create a local fact for special custom values.

Local facts are especially useful for complex scenarios involving multiple rules. For example, suppose there are three different rules:

1. RULE_1 applies to subscribers over age 25.
2. RULE_2 applies to members of the **SILVER** subscriber category.
3. RULE_3 applies to subscribers using WIFI.

You want to install a particular profile named **SPECIAL_RULE** if any two of these three rules apply.

You could create a local fact named **numberOfRule**s with an integer value initialized to 0 and increment that value every time a rule is added. When the local fact's value reaches 2, the **SPECIAL_RULE** PCC Profile is applied. The following rules accommodate this scenario:

```
IF Gx isn't null
THEN
LocalFact (integerValue:0; stringValue:"numberOfRules")
IF
Sp.dateOfBirth.year more than System Variables.currentTime.year - 25,
AND
LocalFact.name is "numberOfRules"
THEN
InstallPCCProfile (name:"RULE_1)
Modify LocalFact (integerValue:localFact.integerValue +1)
IF
SP.subscriberCategory is "SILVER" and
LocalFact.name is "numberOfRules"
THEN
InstallPCCProfile (name:"Reule_2")
Modify LocalFact (IntegerValue:LocalFact.integerValue+1)
IF
Gx.IP-CAN-Type is IPCanType.WIMAX and
LocalFact.name is 'numberOfRules'
THEN
InstallPCCProfile (name:"Rule_3"
Modify LocalFact (integerValue:LocalFact.integerValue+1)
IF
LocalFact.integerValue is 2
THEN
InstallPCCProfile (name:"SPECIAL_RULE")
```

11

Creating Rules and Rulesets

This chapter describes how to create rules and rulesets using the **Rules** tab of the Policy Designer interface to Oracle Communications Service Broker Policy Controller (Policy Controller).

The rules engine used by Policy Controller is built on the rules engine for Oracle Business Rules, an Oracle Fusion Middleware product. For general information about Oracle Business Rules and details on the operands and facts you use to build them, see *Oracle Fusion Middleware User's Guide for Oracle Business Rules 11g Release 1 (11.1.1.5.0 Feature Pack)* at:

http://www.oracle.com/technetwork/middleware/soasuite/documentation/11gr1ps4featurepackdoc-462677.html

There is one significant difference between Oracle Business Rules and Policy Designer. Policy Designer uses the term *Lists of Values* instead of the term "Buckesets" used by Oracle Business Rules.

About Rules and Rulesets

Policy Controller applies PCC profiles to subscribers that specify service flow limits. The PCC profiles themselves are selected by the rules that you create and deploy in dictionaries. You can create collections of rules to work with one another in a single *deployment*. This chapter explains how to create these rules and deploy them in rulesets.

You create rules by using the **Rules** tab of the Policy Controller Policy Designer interface. A knowledge of programming with a third-generation programming language is very helpful for understanding the **Rules** tools and creating rules.

Each rule has two parts: an IF section and a THEN section. The IF section contains a condition test based on evaluation of data from various sources. The THEN section contains actions that can be performed if the test passes. If the condition in the IF section evaluates to true, the rule performs the actions in the THEN section. The actions usually involve dynamically applying a PCC profile to a subscriber or removing a PCC profile from a subscriber, but they may involve other actions as well.

Your rules and rulesets can be as simple or as complex as your deployment requires. You can set up the rules to select a single PCC profile for the entire service flow, or implement any number of PCC profiles. Policy Controller also enables you to reinterpret your rules dynamically as the service flow continues and the input data, such as the approach of a resource limit, changes. You can specify rules that change the PCC profile, for example, if a subscriber reaches a service limit or credit limit.

Rules can be added, modified, and deleted whenever required and can also be activated and deactivated individually.

You manage the following components, which are related to rules, using the **Rules** tab:

Rulesets
A ruleset provides a unit of execution for a collection of rules. You can prioritize the rules within a ruleset.

Rule
Rules are individual test for conditions that determine which profiles your subscribers are entitled to.

Lists of Values
A list of values defines a set of values for a particular fact or a property of a fact.

Complex Lists
See "Using Grouped AVPs in the Rules" for details.

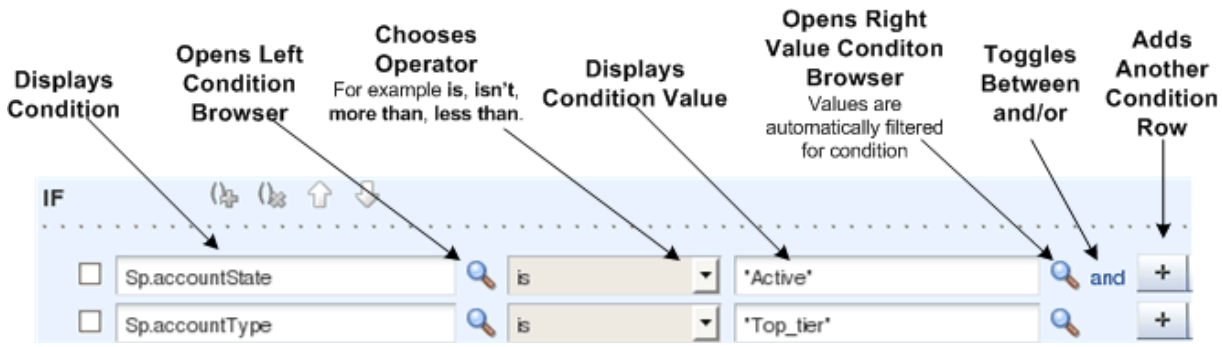
Working with Rules

Each rule takes the form of an IF-THEN statement or collection of IF-THEN statements. The IF section of the rule defines a boolean condition, which may be composed of multiple conditions called tests, associated with each other by logical operators. The THEN section describes one or more actions. If the IF condition evaluates to true, the actions in the THEN section of the rule are performed.

See "Using the Advanced Menu Features for All Rulesets" for details on using the advanced feature of the **Rules** tab.

Figure 11–1 shows the IF statement components. In this example, the IF statement is testing whether the subscriber's account state is **Active** and whether the account type is **Top_tier**. If both statements are true then the THEN statement action (shown in Figure 11–3) is performed.

Figure 11–1 Policy Designer Rule IF Statement Components



Tip: If you must create multiple rules that use the same IF statement, it is considerably more efficient to create a local fact out of the IF statement using a **LocalFact** operator and then reference the local fact in your rules.

Figure 11–2 shows the left condition browser with **Sy** and **System Variables** expanded. The left condition browser displays all the conditions available for you to create rules. Once you pick a condition in the left condition browser, the conditions displayed in the right condition browser are automatically filtered, and only offer the appropriate available conditions.

Figure 11–2 Policy Designer Rule Condition Browser

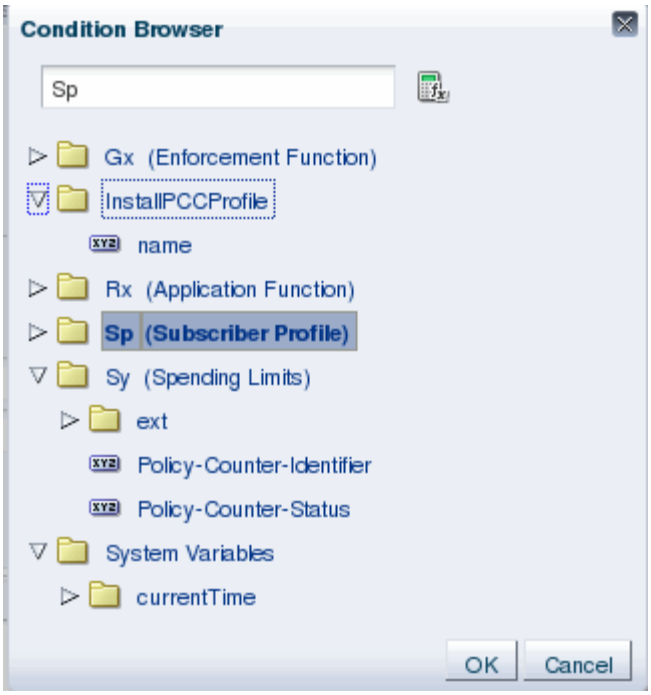


Figure 11–3 shows the THEN statement components. THEN statements are the objects of IF statements. They serve as the rule action.

Figure 11–3 Policy Designer THEN Statement Components

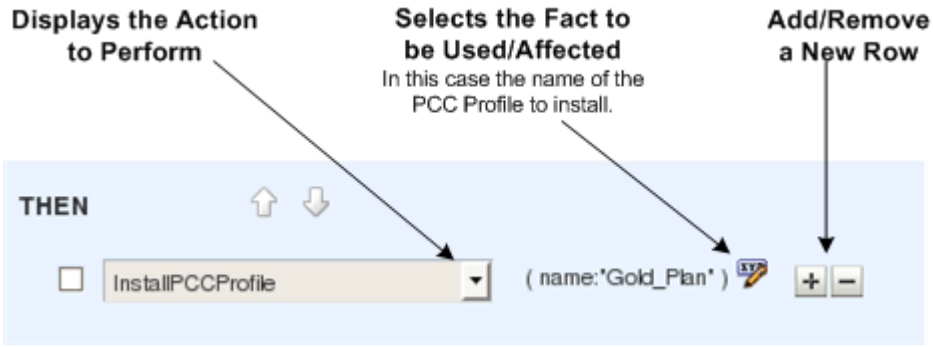
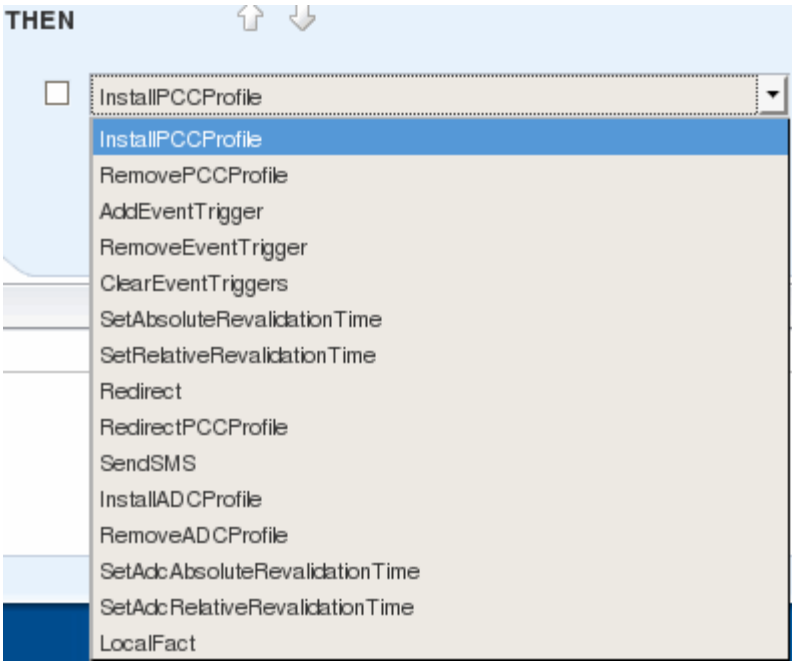


Figure 11–4 shows the THEN statement actions menu displaying the actions available to use in rules. After selecting action, you then select or enter a fact for the rule. The example in Figure 11–3 uses the "install PCC profile" action, that will install the **Gold_Plan** PCC profile created for the example. Because the PCC profiles had already been created, the fact menu (not shown) was automatically populated with the profiles to select from.

Figure 11–4 Policy Designer Rule THEN Actions



Implementing Rules

The general workflow for implementing rules is:

1. Create a ruleset. See "Creating a Ruleset" for details.
2. Create the rules in the ruleset. See "Creating a Rule" for details.
3. Create the list of values and decision functions as needed to support the rules. See "Managing Lists of Values" for details.
4. Validate the ruleset.
5. Deploy the ruleset and supporting list of values and functions as a deployment. Once deployed, the rules are applied to your Policy Controller subscribers. See "Deploying Rulesets to a Deployment" for details.

You create rules inside a ruleset using the **Rules** tab of the Policy Designer. You can change the order in which the rules are displayed in the ruleset; see "Changing the Display Order of Rules in a Ruleset". You can page through the rules in a ruleset using the arrows to the right of the ruleset name. The **Deployments** tab displays individual deployments and shows some of their details.

You can dynamically associate a rule with a profile when an **Assert New** action asserts the **InstallPolicy** fact. A rule is dynamically disassociated from a profile when an **Assert New** action asserts the **RemovePolicy** fact. See "Asserting a New Action" for more information.

You can also create rules by importing a previously created deployment. See "Working with Deployments" and "Managing Deployments" for information about importing a deployment.

You can create a PCC profile from the **PCC Profiles** menu of the rule editor, and also from the profile editor. See "Creating a Dynamic PCC Profile" for information on how to do this.

Rule Editor Naming Conventions

A ruleset name must start with a letter and can contain only the letters (a to z and A to Z), numbers (0 to 9), the following characters: “.”, “-”, “_”,”_”,””, “:”, “/”, and single spaces.

A deployment name can contain only letters (a to z and A to Z), numbers (0 to 9), and the underscore (_) character. Special characters are not permitted in a deployment name.

Other names and aliases, used for rules, facts, list of values and so on, must begin with a letter and contain only letters, numbers, “.”, “-”, “_”,”_”,””, “:”, “/”, and single spaces.

A type name defined by the specification as containing a hyphen (-) may need to have its hyphen changed to an underscore (_) to accommodate the Java-based rules engine. For example, the specification defines the **IP-CAN_CHANGE** event trigger AVP, but this value does not validate and must be changed to **IP_CAN_CHANGE** when used in a rule. The Expression Builder automatically enters this value as **IP_CAN_CHANGE**. See "Using the Expression Builder" for information about the Expression Builder.

Viewing and Modifying Deployments

You manage deployments from the Deployments tab. See "Working with Deployments" for details.

Creating and Deleting Rulesets

This section explains the tasks required to create or detect rulesets.

Creating a Ruleset

To create a ruleset:

1. Start the Policy Designer
2. Click the **Rules** tab.
3. Select **Create** from the Rulesets menu.

   The new ruleset appears in the list of rulesets.
4. Click the ruleset if it is not already selected
5. Assign a name to the ruleset by overwriting the default name in the text field.
6. Select the New Rule (page and plus sign) icon.

   The IF-THEN fields appear.
7. Rename the rule by overwriting the default name in the text field.
8. If you want to configure advanced settings, toggle the advanced settings icon to the left of the Rule name field. See "About the Advanced Settings for Rulesets, Rules, and Rule Actions" for more information on the advanced settings.

   The advanced settings fields appear. Configuration of these fields is optional.

   a. In the **Description** field, you can enter an option textual description of the ruleset. [Advanced Setting]

b. From the **Effective Date** menu, select a configuration for the effective date of the ruleset. See "Setting the Effective Date for a Rule or Ruleset" for details. The default is **Always**. [Advanced Setting]

c. Confirm that the **Active** check box is checked. To deactivate the ruleset, check this box. [Advanced Setting]

9. Add rules to your ruleset. See "Creating and Deleting Rules" for details.

10. Click **Validate** to confirm that your new rule has a valid syntax.

    You can export an invalid ruleset to a file to work on later, but it must validate successfully before you can deploy it.

11. To save your work you have these options:

    - Deploy the ruleset to a deployment. See "Deploying Rulesets to a Deployment" for details. All rules must validate successfully before you can deploy them.
    - Save the ruleset to a file. See "Working with Deployments" for details. Use this option to save incomplete rules and rulesets to work on later. They do not need to pass validation to be saved to a file.

    WARNING: If you do not export or deploy the ruleset, your work is not saved.

12. Your ruleset now appears in the **Rulesets** section on the left side of the **Rules** tab

Deleting a Ruleset

To delete a ruleset:

1. From the Rulesets list, select the ruleset that you want to delete.
2. Select **Delete** from the Rulesets menu.

   The selected ruleset is deleted.

Deleting a ruleset does not affect a deployment that has already been exported or deployed. To make the deletion permanent, you must export or re-deploy the deployment.

Setting the Effective Date for a Rule or Ruleset

The Effective Date menu is displayed only when advanced settings are shown. See "About the Advanced Settings for Rulesets, Rules, and Rule Actions" for details on the advanced settings.

To set the effective date of a rule or ruleset:

1. From the Effective Date menu, select one of the options described in Table 11–1:

Table 11–1 Effective Date Settings

Value	Description
Always	The rule or ruleset is always in effect.
Range	The rule or ruleset is in effect from the specified start date to the specified end date.

Table 11–1 (Cont.) Effective Date Settings

Value	Description
From	The rule or ruleset is in effect from the specified start date with no end date.
To	The rule or ruleset is in effect from the deployment of the ruleset to the specified end date.

2. If you specified a value other than **Always**, select from the date-time menu whether you want to specify a **date**, a **time** or **both** a date and a time for the start date and end date values. This setting applies to both the start and end dates. You cannot assign different configurations to the start and end dates.
3. If you are specifying a start date, click the date-time icon to the right of the Start Date field.

   An editable calendar appears.
4. In the calendar, set the date and / or the time and time zone, depending on whether you are setting a date, a time or both.
5. Click **OK**.
6. If you are specifying a range or an end date, click the date-time icon next to the End Date field and repeat steps 4 and 5.

Managing Rulesets

You can activate/deactivate rulesets or change the order in which they are applied by using the **Manage Rulesets** item from the **Advanced** menu. See "About the Advanced Settings for Rulesets, Rules, and Rule Actions" for more information on the advanced settings.

To activate/deactivate or change the execution order of rulesets:

1. Start the Policy Designer
2. Navigate to the **Rules** tab.
3. On the left side of the **Rule** tab select **Manage Rulesets** from the **Advanced Menu**.

   The Manage Rulesets screen appears, containing entries for all of your rulesets.
4. To deactivate a ruleset, use the single left arrow button to move it from the **Active** list to the **Inactive** list. The double arrow moves all the rulesets.

   Note: You can also activate a ruleset using the **Active** check box on the **Rules** tab. If you set a ruleset as inactive using one of these tools, it remains inactive, regardless of the other setting.
5. To activate a ruleset, use the single right arrow button to move it from the Inactive list to the Active list. The double arrow moves all the rulesets.
6. To change the order in which ruleset are applied, select the ruleset to reorder.
7. Do one of the following:

   Click this icon to move the ruleset one position up:

or

Click this icon to move the ruleset one position down:

or

Click this icon to move the ruleset to the top of the list:

or

Click this icon to move the ruleset to the bottom of the list:

8. Navigate away from the **Manage Rulesets** window once you have made your changes.

Creating and Deleting Rules

This section explains the procedures for creating and deleting rulesets.

Creating a Rule

To create a rule:

1. Start the Policy Designer interface.
2. Select the **Rules** tab.
3. Select a ruleset.
4. Click the new rule icon.

The rule is created.

5. Overwrite the default rule name with the name of your choice in the rule name field.
6. If you want to configure advanced settings, toggle the advanced settings icon to the left of the rule name field to display the advanced settings fields.

The advanced settings fields appear. See "About the Advanced Settings for Rulesets, Rules, and Rule Actions" for more information on the advanced settings.

Configuration of the advanced settings fields is optional.

a. In the Description field, you can enter an option textual description of the rule. [Advanced Setting]

b. From the **Effective Date** menu, select a configuration for the effective date of the rule. See "Setting the Effective Date for a Rule or Ruleset" for details. The default is **Always**. [Advanced Setting]

c. Select a priority level from the **Priority** menu relative to the priority of the other rules in the ruleset. The options are: **highest**, **higher**, **high**, **medium** (the default), **now**, **lower**, and **lowest**. [Advanced Setting] Higher priority rules are interpreted before lower priority rules. The default priority is **medium**.

d. Check the **Rule Active** check box to activate the rule or clear the check box to deactivate it. The default is Rule Active. [Advanced Setting]

e. To enable advanced mode, check the **Advanced Mode** check box. Advanced mode allows additional pattern-matching options for creating conditions and actions. Advanced mode also enables you to test the rule with specific data values. See the discussion of advanced mode rules in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for information about advanced mode. [Advanced Setting]

f. To enable tree mode, check the **Tree Mode** check box. Tree mode is used for master detail rule hierarchies. See the discussion of tree mode rules in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for information about tree mode. [Advanced Setting]

7. In the IF section of the rule editor, define the rule's condition. See "Defining the Condition of a Rule" for details.

> **Note:** By default the grouped AVPs are not displayed. See "Using Grouped AVPs in the Rules" for details on displaying grouped AVPs.

8. In the THEN section of the rule editor, define the rule's actions. See "Defining the Actions of a Rule" for details.

9. To save your work you have these options:

   - Deploy the ruleset to a deployment. See "Deploying Rulesets to a Deployment" for details.
   - Save the ruleset to a file. See "Working with Deployments" for details.

Deleting a Rule

To delete a rule:

Click the delete icon next to the rule that you want to delete.

Defining the Condition of a Rule

A condition is composed of one or more tests, connected by **and** or **or** logical operators. Each test evaluates to true or false. A single row of fields in the IF section of the rule editor represents a single test. If the entire condition defined in the IF section of the rule evaluates to true, the actions defined in the THEN section of the rule are

performed. If the condition does not evaluate to true, none of the actions are performed.

Defining the condition of a rule involves constructing one or more tests and combining them with the correct logical operators to create the condition.

You create a test by editing a row of fields in the editor in the IF section of the rule editor. A test consists of three components:

- Left operand
- Comparison operator
- Right operand

See the "Example Rules" section to examine sample rules that you can use as models for your own rules.

Note: For ease of use Policy Controller enables you to enter UTF-8 values for the AVPs that the 3GPP specifications specify in octet format, such as **GxUser-Equipment-Info-Value**.

Creating a Test

To create a test:

1. Do one of the following:
   - If a blank test row is displayed in the IF section of the rule, continue to step 2.

   or

   - To add a new test row, in the IF section of the rule click the insert test button to the right of the existing test row.

   You may have to scroll horizontally to see the end of the row.

   A new test row appears.
2. Define the left operand of the test by doing one of the following:
   a. Click the search icon to the right of the left operand field.

      The Condition Browser appears.

   b. In the Condition Browser, select the value to use for the left operand.

      See "Using the Condition Browser" for information about the Condition Browser.

   or

   - Enter a literal value by typing the value directly into the left operand field.
3. Repeat step 2 for the right operand, using the search icon to the right of the right operand field to display the Condition Browser.
4. From the menu of comparison functions between the two operand fields, select the function to use to compare left and right operands.

   The menu is context sensitive, so its items vary depending on the contents of the left and right operands.

Deleting a Test from a Rule

To delete a test from a rule:

Click the delete test button to the right of the test that you want to delete.

Creating a Condition with Multiple Tests

To create a condition that contains multiple tests:

1. Create a test as described in "Creating a Test".
2. Click the insert test button on the right side of the test row to add another test.

3. Toggle the logical operator on the right side of the first test to be **and** or **or** depending on the logic of the condition you are creating.
4. Create another test.
5. Continue to add tests and connect them with logical operators until you have constructed the entire condition.
6. If you want to enclose multiple tests in a parentheses to create nested tests, check the check boxes to the left of the rows that you want to enclose and click the add parentheses icon above the condition.

The tests being enclosed in a single set of parentheses must be contiguous in the rule editor. If necessary, change the order of the tests before applying the parentheses. See "Changing the Order of Tests" for information on how to do this.

To remove the parentheses:

1. Check the check boxes to the left of the tests around which you want to remove parentheses.
2. Click the remove parentheses icon above the condition.

Changing the Order of Tests

The tests are evaluated in the order in which they appear in the rule.

To change the order in which the tests are evaluated:

1. Check the check box to the left of the test for which you want to change the position.
2. Click the up or down arrow at the top of the IF section to change the position of the test in the rule. Every click moves the test up or down one row.

Defining the Actions of a Rule

In the THEN section of the rule, you define the actions to be performed if the condition evaluates to true. Each row of fields in the THEN section of a rule defines a single action.

To define an action:

1. Do one of the following:
   - If there are no action rows displayed, click **Insert Action**.

   or

   - If action rows are displayed, click the insert action button on the right side of an existing row to create an action row.

   A new action row appears.
2. The default rule action is **InstallPCCProfile**. If you are installing a PCC profile select the **Edit Properties** box, and from within it select from the PCC profiles available. is automatically populated with actions that you are allowed.

   If you are not installing a PCC profile, select another option from the rule action menu. See "Asserting a New Action" for details on the menu items.

Asserting a New Action

You can assert the following actions in the actions (THEN) section of you rules. These actions are all outputs of the rules engine.

If you select one of these actions facts, and then select the **Edit Properties** (pencilXYZ) icon, the **Edit Properties** form appears. You can type an entry in the Value field, or select the search icon to choose from a list of all acceptable values. The acceptable values list is automatically populated with just the permissible values for the action you chose.

InstallPCCProfile

Associates the rule with the specified PCC profile. The **value** property contains the PCC profile name.

RemovePCCProfile

Disassociates the rule from the specified PCC profile. The **value** property contains the PCC profile name to disassociate.

AddEventTrigger

Adds the specified event trigger to the Gx session. The **value** property contains the event trigger to add. You can specify any event trigger from the *Policy and charging control over Gx reference point (3GPP TS 29.212 Release 9*) specification. See "About Event Triggers" for more information.

RemoveEventTrigger

Removes the specified event trigger from the Gx session. The **value** property contains the event trigger to remove. You can specify any event trigger from the *Policy and*

charging control over Gx reference point (3GPP TS 29.212 Release 9) specification. See "About Event Triggers" for more information.

ClearEventTriggers
Removes all existing event triggers from the Gx session set, including all the system-level event triggers and adds the NO_EVENT_TRIGGERS (14) event trigger to the Gx session.

SetAbsoluteRevalidationTime
Sets a deadline, before which the PCEF should re-request an update of the rules. The timer is set to an absolute date/time; for example: 2011-12-30 14-55-30 PST.

SetRelativeRevalidationTime
Sets a deadline, before which the PCEF should re-request an update of the rules. The timer is set as the number of seconds from the time that the rule was invoked; for example: 3600.

Redirect
Redirects all service data flows to a Web address (global session redirection). See "Globally Redirecting All Services in a Session" for details.

RedirectPCCProfile
Redirects a single service inside a session to a Web address (service based redirection), and associates a new PCC profile to the rule. See "Redirecting Individual Services Inside a Session" for details.

SendSMS
Sends and SMS message as part of the rule action. See "Adding SMS details to a Rule" for details.

InstallADCProfile
Associates the rule with the specified ADC profile. The **value** property contains the PCC profile name.

RemoveADCProfile
Disassociates the rule from the specified ADC profile. The **value** property contains the ADC profile name to disassociate.

SetAdcAbsoluteRevalidtionTime
Sets a deadline, before which the PCEF should re-request an update of the rules. The timer is set to an absolute date/time; for example: 2011-12-30 14-55-30 PST.

SetAdcARelativeevalidtionTime
Sets a deadline, before which the PCEF should re-request an update of the rules. The timer is set as the number of seconds from the time that the rule was invoked; for example: 3600.

LocalFact
Creates a custom fact. See the discussion of **LocalFact** in "Using the Condition Browser" for more information.

To define an assert new action:

1. In the THEN section of the rule editor, select **Assert New** in the left menu.
2. Select the fact to assert from the right menu.
3. Click the pencilXYZ icon.

   A properties form appears in which you set the value of the fact.

4. If the value is a constant, check the **Constant** check box.
5. To set the value of the fact do one of the following:
   a. Click the search icon to the right of the Value field.

      The Condition Browser appears.

   b. In the Condition Browser, select the value.

      See "Using the Condition Browser" for information.

   or

   - Enter a literal value by typing the value directly into the Value field.
6. Click **OK**.

Changing the Order of Actions

The actions are performed in the order in which they appear in the rule.

To change the order in which the actions are performed:

1. Check the check box to the left of the action for which you want to change the position.
2. Click the up or down arrow at the top of the THEN section to change the position of the action in the rule. Every click moves the action up or down one row.

Deleting an Action

To delete an action from a rule:

Click the delete action button to the right of the action that you want to delete.

About Event Triggers

Policy Controller uses event triggers to inform the PCEF that it should trigger a new request for rules when any of the subscribed events, such as an IP-CAN change, occurs at the gateway. You can use the default specified triggers or add your own. See "Using an Event Trigger to Change a PCC Profile" for details.

Using the Condition Browser

The Condition Browser is generally used for creating unusually complex expressions to use in rules. You use it to browse for values to use in the operands of a test in the IF section of a rule, and values used for the properties of facts in the THEN section of a rule. The Condition Browser contains a text field, a hierarchical tree view of the rules metadata, organized by protocol, and an embedded Expression Builder. The Condition Browser is also a mechanism for including any method parameters in rules.

There are three ways to enter a value this browser:

- You can select a leaf item from the tree.
- You can type values directly in the browser text field.
- You can use the Expression Builder embedded in the browser to create an expression. See "Using the Expression Builder" for more information.

After you enter a value into the in the browser's text field by one or a combination of these methods and click **OK**, the value appears in the operand field that you are editing in the rule editor.

The default nodes displayed in the browser tree reference the data listed in Table 11–2. The Condition Browser tree also includes any facts that you have already used in rules. In general the nodes contain data facts generated outside the rules engine and used as input for the rules engine. The individual nodes can contain child nodes.

Table 11–2 Condition Browser Node Data

Node	Description
Gx (Enforcement Function)	Attributes in the **gx** node specify data concerning traffic between your PCEF and Policy Controller. They are defined in the *Policy and charging control over Gx reference point 3GPP TS 29.212 (Release 9*) specification.

LocalFact	**LocalFact** is used for custom values that can be used in actions and tests. **LocalFact** supports integerValue, name, booleanValue, and doubleValue values.
	You can assert a **LocalFact**:
	`assert new LocalFact (integerValue:42)`
	then create a test based a **LocalFact** value:
	`IF`
	`LocalFact.integerValue isn't 42`
	and then modify **LocalFact** in another action:
	`modify LocalFact (name:"strange value")`

Rx (Application Function)	Attributes in the **Rx** node specify data from the traffic between Policy Controller and your Application Functions. They are defined in defined in the *Policy and Charging Control Over Rx reference point 3GPP TS 29.214 v9.8.0 (2011-09)(Release 9*) specification.
Sp (Subscriber Profile)	Attributes in the **Sp** node specify data from the traffic between Policy Controller and the SPR portion of the SPR/OCS that you are using. Policy Controller uses an Sh-like interface for Sp data. For details on Sh, see the *Sh Interface based on the Diameter Protocols (3GPP TS 29.329 V10.4 (2011-12)* and the discussion of the subscriber store data model in *Oracle Communications Service Broker Subscriber Data and Lifecycle User's Guide*.
Sy (Spending Limits)	Attributes in the **Sy** node specify data from the traffic between Policy Controller and the OCS portion of the SPR/OCS that you are using. They are defined in defined in the *Sy Interface based on the Diameter Protocol (3GPP TS 29.219 Release V11.2 (2012-09*) (Release 11) specification
System Variables	Attributes in this node currently include the **currentTime** variables, such as **amPm**, **dayOfMonth**, and so on.

Using the Expression Builder

You use the Expression Builder to build advanced expressions used in rule tests.

You access the Expression Builder from the **Rules** tab Condition Browser by clicking the Expression Builder icon:

You can directly type an expression in the Expression text field in the Expression Builder. You can also insert values from the rules metadata using the four tabs: **Variables**, **Functions**, **Operators**, and **Constants**. Each tab displays the rules metadata in a tree structure.

To build an expression, select an item in the tree and click the Insert Into Expression button to insert the selected item at the cursor position into the Expression text field. You can switch among the tabs for the different items needed to build the expression. You can also type directly in the text field.

After you have created the expression and clicked **OK**, the expression appears in the text field in the Condition Browser.

Expression Builder provides a variety of functions to simplify rule creation.

The **Functions** tab of Expression Builder is especially useful for providing functions that can be used in tests. The "Applying a Profile for Part of a Day" example uses the Gx **GetCalendarAtHour** function to This function requires you to use the Expression Builder **Functions** tab.

To select the **GetCalendarAtHour** function, click the search icon in the right side of the rule IF statement. In this example, the left side of the IF statement specifies the Gx **MS-TimeZone_3GPP.date** object, so the **Set Date and Time** window appears. Click the **Date Expression** option and then click the search icon from this window. The **Condition Browser** appears. Click the Expression Builder icon from the top left and the Expression Builder window appears. Select the Functions tab of the Expression Builder, and select the **GetCalendarAtHour** function from where it appears under **Gx**. The **GetCalendarAtHour** function appears in the right side of the IF statement.

Changing the Display Order of Rules in a Ruleset

To change the order in which the rules are displayed in the rule editor, use the up and down arrows to the right of the rule name field to move the rule. Each click moves the rule one position up or down in the ruleset.

The display order of rules in the rule editor has no effect on the order in which the rules are fired.

Deploying Rulesets to a Deployment

To deploy a rules deployment:

1. Click the **Deploy** button at the top of the rule editor.

   A dialog box appears containing a Note text field.

2. Optionally add a note about the deployed deployment in the Note text field.

   The text that you enter appears near the top of the screen in the **Deployments** tab of the Policy Designer when the deployment is loaded.

3. Click **Deploy** in the dialog box to deploy the deployment.

Future updates to the deployment do not affect a deployed deployment. To include updates in the release, you must re-deploy the deployment.

12

Working With Service Data Records

This chapter describes how to monitor Oracle Communications Service Broker Policy Controller (Policy Controller) using Service Data Records (SDRs).

About Service Data Records

Policy Controller SDRs are records of network and application events. The information they contain is recorded as relevant events occur, and the details are saved to SDR files.

SDRs enhance the debugging and tracing capability of Policy Controller. For example, SDRs provide you with the capability to trace any particular session of a subscriber. You can also use external products such as the Oracle Communications Data Model to analyze SDRs, transforming their data into actionable information.

See "Application SDRs" for more information.

About Service Data Record Types

Policy Controller supports two types of SDRs:

- Network SDRs
- Application SDRs

Network SDRs

Network SDRs are mainly used for messages exchanged with other entities in the network, for example: the Policy Enforcement Rules Function (PCEF), application function (AF), or an internal or external Subscriber Profile Repository (SPR).

Network SDRs support Diameter and SPR events. The SDR details are captured at Policy Controller entry and exit points. For example: Incoming Diameter Gx CCR message events are logged in a network SDR.

A Network SDR is written in two parts:

- Common Header:

  This part is always written for each incoming and outgoing network message. The common header fields are separated by a delimiter. See "Common Header Fields for Network SDRs" for more information.

- Variable fields:

These fields are written according to what is configured as part of the template parameters section. Policy Controller is shipped with built-in templates. You can use the Administration Console to view and customize the template fields.

Each built-in template is defined with certain field names. If a field is present in a message, it is assigned a value and recorded as part of the SDR.

If there is no template available for a message type, nothing gets written for the variable part.

A template for a message is defined using a combination of these three fields:

- Protocol
- Interface
- Opcode

Application SDRs

Application SDRs are used to record data specific to the Policy Controller application. Each SDR contains fields specific to Policy Controller.

An application SDR is written in two parts:

- Common Header part:

  This part is always written for each incoming and outgoing network message. The common header fields are separated by a delimiter. See "Common Header Fields for Application SDRs" for more information.

- Variable Fields part:

  There is no template for application SDRs because its fields are based on the specific application, Policy Controller, that generates it.

Application SDRs can support this type:

- Correlation Application SDR:

  This type of SDR associates the correlation key with the session key. No variable parameters are included. The Application Record Type is 2.

About Service Record Data Formats

This section describes the common header fields for network and application SDRs.

Common Header Fields for Network SDRs

Table 12–1 describes the common header fields for network SDRs.

Table 12–1 Common Header Fields For Network SDRs

Field	Description	Mandatory	Example
Timestamp	Time stamp in format yyyy-mm-dd hh:mi:ss,msec	Yes	N/A
Version	Version number for the SDR	Yes	100
SDR Type	SDR Type (Network/Application)	Yes	NW
ServerId	Unique Server ID where the SDR was generated	Yes	managed_1
Protocol	Protocol used	Yes	DM(Diameter)
Interface	Interface	Yes	GX

Table 12–1 (Cont.) Common Header Fields For Network SDRs

Field	Description	Mandatory	Example
Opcode	Operation code	Yes	CCR
Direction	Direction of message: IN/OUT Note: This is from the application perspective	Yes	IN
Origin	(Optional) Source entity sending this message	No	server1 pcef.com
Destination	(Optional) Destination for the message	No	server us.oracle.com
SessionKey	Unique ID for the session	Yes	session12345
CorrelationKey	Correlation ID for the session (secondary key)	No	123456789

How to enter values for the Origin and Destination fields:

- SPR messages: The Origin and Destination fields are empty.
- Diameter requests: Origin-Host or Origin-Realm values are used for the Origin field. If there is a destination value, Destination-Host or Destination-Realm values are used for the Destination field.

Common Header Fields for Application SDRs

Table 12–2 describes the common header fields for application SDRs.

Table 12–2 Common Header Fields For Application SDRs

Field	Description	Mandatory	Example
Timestamp	Time stamp in format yyyy-mm-dd hh:mi:ss,msec	Yes	N/A
Version	Version number for the SDR	Yes	100
SDR Type	SDR Type (Network/Application)	Yes	NW
ServerId	Unique Server ID where the SDR was generated	Yes	managed_2
App Record Type	Record type chosen by the application	Yes	105
SessionKey	Unique Id for the session	Yes	session12345
CorrelationKey	Correlation Id for the session(secondary key)	No	123456789

About Service Record Data Templates

Policy Controller provides built-in templates that provide values for fields in Diameter and SOAP SDRs.

Diameter Templates

Table 12–3 describes the templates that are available for Diameter network SDRs.

Table 12–3 Built-in Diameter SDR Templates

Interface	Opcode	Parameters
GX	CCR	Subscription-Id | IP-CAN-Type | RAT-Type | Supported-Features
GX	CCA	Result-Code | Charging-Rule-Install | Revalidation-Time | Event-Trigger
GX	RAR	Charging-Rule-Install | Revalidation-Time | Event-Trigger

Table 12–3 (Cont.) Built-in Diameter SDR Templates

Interface	Opcode	Parameters
GX	RAA	Result-Code
RX	AAR	Subscription-Id | AF-Application-Identifier | Specific-Action
RX	AAA	Result-Code

The templates define the variable part of the network SDRs for Diameter interfaces using attribute-value-pairs (AVP). The AVPs match those defined as part of the Diameter signaling server units (SSU).

The AVPs are scalar or grouped:

- Scalar AVP: The name and its value are written in the SDR and are specified from its top level.

  Example: "IP-CAN-Type":1

- Grouped AVP: The entire group AVP, with all of its child AVPs, are written in the SDR.

  Example:
  "Supported-Features":{"Vendor-Id":1,"Feature-List-ID":2255,"Feature-List":22,"Vendor-Id":3,"Feature-List-ID":4255,"Feature-List":44}

Note: If you omit defining the variable part, the mandatory parameter values will still be written to the SDRs. For example, Origin, Destination, and Session Key.

SOAP Templates

Table 12–4 describes the templates that are available for SOAP network SDRs.

Table 12–4 Built-in SOAP SDR Templates

Interface	Opcode	Parameters
SP	PREQ	Uid Note: Unique ID assigned by the subscriber store.
SP	PRES	Uid | Response-Status | Profile Note: Status for the profile response (OK, ERROR_INVALID_DATA, ERROR_SERVER_ERROR).
SP	PUPD	Uid | Profile Note: Profile for the subscriber including counters.

Customizing Templates

SDR templates can be modified if required.

To modify an SDR temple:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Select **PCRF** then **System Parameters**.
3. Open the **Service Data Record Templates** tab.
4. Click **Lock and Edit**.
5. Edit the template according to the information in Table 12–3 and Table 12–4. You can add AVPs.
6. When you are finished, select **Commit**.

Note: It is possible to define multiple SDR templates with the same key: Protocol+Interface+Opcode. If this occurs, the first of the duplicate keys is selected.

About SDR Output Files

This section describes the output files when SDRs are generated.

SDR Output Format

Both the fixed and variable fields written as part of the SDR are delimiter separated.

- Fixed header fields: These are written without any parameter name and are separated by a delimiter.
- Variable parameters: JSON is the default output format for variable parameters. Within JSON, all binary data is written in hexadecimal format.

Note: Time fields are in seconds.

Output Fields

The SDR headers for fixed fields and the actual SDR files are written into separate files:

- SDR Headers: The header for the fixed fields are written in the **ocpc_sdr_hdr.log** file. This log file provides information on the parameter names that are part of the fixed header for that version.

  Example: SDR Header (includes two lines)

  TIMESTAMP | VERSION | SDRTYPE | SERVERID | PROTOCOL | INTERFACE | OPC ODE | DIRECTION | ORIGIN | SESSIONKEY | CORRELATIONKEY |

  TIMESTAMP | VERSION | SDRTYPE | SERVERID | APPRECORDTYPE | SESSIONK EY | CORRELATIONKEY |

- SDR Files: The actual SDRs are written in the **ocpc_sdr.log** file.

SDR Examples

This section provides examples of SDR fields and examples of actual SDRs.

SDR Example Fields

Table 12–5 shows an example of SDR field values

Table 12–5 SDR Example Field Values

Field Name	Value
Timestamp	2012-07-05 16:31:00.689
Version	100
SDR Type	NW
ServerId	managed_1
Protocol	DM
Interface	GX
Opcode	CCR
Direction	IN
Origin	pcef1@client.com
Destination	pcrf@us.oracle.com
SessionKey	GxSession-tc1_basic
CorrelationKey	14128771501
Variable Parameters	{IP-CAN-Type":1,"RAT-Type":0,Supported-Features":{Vendor-Id":1,"Feature-List-ID":2255,"Feature-List":22,"vendor-Id":3"Feature-List-Id":4255,"Feature-List":44}}

SDR Record Examples

Table 12–5 shows an example of SDR records:

Example 12–1 SDR Example Records

```
2012-07-31 11:10:36.722|100|NW|managed_1|DM|GX|CCR|IN|pcef1
client.com|us.oracle.com|GxSession-tc09-2|||
{"Subscription-Id":{"Subscription-Id-Type":0,"Subscription-Id-Data":"14128771509"}
,"IP-CAN-Type":1,"RAT-Type":0}|
2012-07-31 11:10:36.882|100|NW|managed_1|DM|GX|CCA|OT|||GxSession-tc09-2|||
{"Result-Code":"2001"}|
2012-07-31 11:10:55.599|100|NW|managed_1|DM|GX|CCR|IN|pcef1
client.com|us.oracle.com|GxSession-tc11-2|||{"Subscription-Id":{"Subscription-Id-T
ype":0,"Subscription-Id-Data":"14128771511"},"IP-CAN-Type":1,"RAT-Type":0}|
2012-07-31 11:10:55.602|100|NW|managed_
1|SO|SP|PREQ|OT|||e003188b935c41079e2d75cc73a98d24|GxSession-tc11-2||
{"Uid":"e003188b935c41079e2d75cc73a98d24"}|
```

Enabling SDR Generation

By default, Policy Controller does not generate SDRs.

To enable Policy Controller SDRs:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.

2. Select **PCRF** and then select **System Parameters**.

Note: If you have deployed Policy Controller in a multi-domain topology, use the Administration Console for the processing domain to enable SDRs.

3. Select the **Parameters** tab.
4. Click **Lock and Edit**.
5. Locate **Enable SDR** and select **True**.
6. When you are finished, select **Commit**.

Customizing Templates

SDR templates can be modified if required.

To modify an SDR temple:

1. Start the Administration Console.

   For details see *Oracle Communications Service Broker Administrator's Guide*.
2. Select **PCRF** and then select **System Parameters**.
3. Open the **Service Data Record Templates** tab.
4. Click **Lock and Edit**.
5. Edit the template according to the information in Table 12–3 and Table 12–4. You can add AVPs.
6. When you are finished, select **Commit**.

Configuring SDR Logging

Depending on your system capacity, you may want to modify maximum SDR file size and the maximum allowed number of SDR files.

Log4j controls SDR logging. Policy Controller includes preconfigured Log4J parameters with key-value pairs that define SDR logging.

The Log4J configuration Mbeans reflect the structure of the Log4J XML configuration file. See Log4J documentation at:

`http://wiki.apache.org/logging-log4j/Log4jXmlFormat`

The MBean Object Name is:

oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig

The default **configuration** MBean has an object name with **name1** set to **configuration[0]**:

oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig,name1=configuration[0]]

Setting the Maximum File Size and Number of Files

Using the Scripting Engine or an MBean browser:

1. Locate the Log4J Configuration MBean.

2. Locate the **configuration** MBean with object name:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig,name1=configuration[0]]

3. Locate the **appender** MBean that has the attribute **name** set to **pcrf_sdr_file**.

   Example Object Name:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig,name1=configuration[0],name2=appender[3]

4. Locate the **param** MBean with the attribute **name** set to **MaxFileSize**.

   Example Object Name:

   oracle:type=oracle.axia.cm.ConfigurationMBean,[[BR]] name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig,name1=configuration[0],name2=appender[3],name3=param[1]

5. Set the attribute **value** for the **param** MBean to the maximum file size. Default value is 5MB.

6. Locate the **param** MBean with the attribute **name** set to **MaxBackupIndex**.

   Example Object Name:

   oracle:type=oracle.axia.cm.ConfigurationMBean,[[BR]] name=oracle.axia.logging.log4jconfig,version=1.0.0.0,name0=log4jConfig,name1=configuration[0],name2=appender[3],name3=param[2]]

7. Set the attribute **value** for the **param** MBean to the number of files. Default value is 100.

Administrative Issues

The following are several known issues:

- SDR file backup is not handled by Policy Controller.
- SDR file aggregation and analytics is not handled by Policy Controller.

A

Policy Controller Reference

This appendix lists Policy Controller reference material.

Policy Controller Specification Reference

Policy Controller adheres to the specifications listed in Table A–1.

See the *Oracle Communications Service Broker Policy Controller Protocol Conformance Guide* for lists of the supported protocol messages and AVPs.

Table A–1 Policy Controller Diameter-based Protocol Interfaces

Communication Category	Comments/Standard
Policy Controller	*Policy Charging and Control Architecture 3GPP TS 23.203 v9.9.0 (2011-06)*
AF (application function)	*Policy and Charging Control Over Rx reference point (3GPP 29.214 v9.8.0 (2011-09)*. See *Policy Controller Protocol Implementation Conformance Statement* for details on the supported messages and attributes.
OCS (Online Charging System)	Policy Controller communicated with the OCS portion of your SPR/OSC using this Sy interface standard: *Sy Interface based on the Diameter Protocol (3GPP TS 29.219 Release V11.2 (2012-09*). See *Policy Controller Protocol Implementation Conformance Statement* for details on the supported messages and attributes.
PCEF (Policy Controller Enforcement Function)	*Policy and charging control over Gx reference point 3GPP TS 29.212 v9.7.0 (2011-06)*. See *Policy Controller Protocol Implementation Conformance Statement* for details on the supported messages and attributes
SMSC so SME (Short Message Service Centers to Short Message Entities	*Service Centres (SMSCs) to Short Message Entities (SMEs) 3GPP TR 23.039 V5.0.0 (2002-06)*.
SPR (Subscriber Profile Repository)	Policy Controller communicates with the SPR portion of your SPR/OCS using an Sh-like interface as the Sp reference point. For details on the Sh reference point, see *Spending Limit Reporting over the Sh reference point 3GPP TS 29.329 v11.1.0 (2012-06)*.

Supported Gx Event-Trigger Event Values

This section lists the supported values that Policy Controller supports for Gx **Event-Trigger** AVP (Code 1006). You can add to this list by specifying additional values for **Event-Trigger** using the Service Broker Administrator Console. See "Add

any Vendor-specific Values for Event-Trigger to Use as Event Triggers" for details on adding custom values. Table A–2 lists the default Event-Trigger values.

Table A–2 Supported Default Gx Event-Trigger Values

Default supported Event-Trigger AVP	Event-Trigger Value
SGSN_CHANGE	0
QOS_CHANGE	1
RAT_CHANGE	2
TFT_CHANGE	3
PLMN_CHANGE	4
LOSS_OF_BEARER	5
RECOVERY_OF_BEARER	6
IP-CAN_CHANGE	7
QOS_CHANGE_EXCEEDING_ AUTHORIZATION	11
RAI_CHANGE	12
USER_LOCATION_CHANGE	13
NO_EVENT_TRIGGERS	14
OUT_OF_CREDIT	15
REALLOCATION_OF_CREDIT	16
REVALIDATION_TIMEOUT	17
UE_IP_ADDRESS_ALLOCATE	18
UE_IP_ADDRESS_RELEASE	19
DEFAULT_EPS_BEARER_QOS_CHANGE	20
AN_GW_CHANGE	21
SUCCESSFUL_RESOURCE_ALLOCATION	22
RESOURCE_MODIFICATION_REQUEST	23
PGW_TRACE_CONTROL	24
UE_TIME_ZONE_CHANGE	25
TAI_CHANGE	26
ECGI_CHANGE	27
CHARGING_CORRELATION_EXCHANGE	28
APN_AMBR_MODIFICATION_FAILURE	29
USER_CSG_INFORMATION_CHANGE	30
USAGE_REPORT	33
DEFAULT_EPS_BEARER_QOS_ MODIFICATION_FAILURE	34
USER_CSG_HYBRID_SUBSCRIBED_ INFORMATION_CHANGE	35
USER_CSG_HYBRID_UNSUBSCRIBED_ INFORMATION_CHANGE	36
ROUTING_RULE_CHANGE	37

Table A–2 (Cont.) Supported Default Gx Event-Trigger Values

Default supported Event-Trigger AVP	Event-Trigger Value
MAX_MBR_APN_AMBR_CHANGE	38
APPLICATION_START	39
APPLICATION_STOP	40
ADC_REVALIDATION_TIMEOUT	41