Oracle® Communications Service Broker

Policy Controller Implementation Guide

Release 6.0

E23528-02

March 2012

ORACLE®

Oracle Communications Service Broker Policy Controller Implementation Guide, Release 6.0

E23528-02

Contents

3 Specifying Service and Charging Information with Policy Profiles

4 Creating Rules and Rulesets

5 Policy Controller Protocol Reference

Preface

This document describes how to install, configure, and use the Oracle Communications 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 the Policy Controller.

This manual assumes that you are already familiar with policy control strategies and tools used in the telecom industry, including:

- PCRF (Policy and Charging Rule Function).
- PCEF (Policy and Charging Enforcement Function).
- BBERF (Bearer Binding and Event Reporting Function).
- Diameter Gx and Rx protocols.
- AFs (Application Functions).

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 6.0 documentation set:

- *Oracle Communications Service Broker Installation Guide*
- *Oracle Communications Service Broker Concepts Guide*
- *Oracle Communications Service Broker Signaling Domain Configuration Guide*

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 Policy Controller and explains its capabilities and components.

Understanding What Policy Controller Does

Policy Controller is a sub-product of Oracle Communications Service Broker that you use to manage Quality of Service (QoS), optimize high bandwidth traffic, and enforce usage quotes levels for subscribers. Policy Controller is a Policy and Charging Rule Function (PCRF) product that makes business policy decisions as defined by the 3GPP TS 23.203 v9.9.0 (2011-06) specification.

Policy Controller offers real-time control over the way you allocate and charge for network capacity and services. You can easily expand and change these allocations and charging schemes as your product line of IP-based products changes.

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

- Specify the charging and policy details for your product offerings. These products define how broadband resources are allocated among your subscribers, and what you will charge for them.
- Create rules that Policy Controller uses to select broadband products for your subscribers. These rules define how broadband resources are allocated among your subscribers, and under what conditions.

Policy Controller then interprets these rules and applies them to the charging and policy details and passes these decisions on to your Policy Control Enforcement Function (PCEF) for enforcement. Policy Controller can update these decisions during data flow as the data flow itself causes conditions change and new rules and QoS levels to apply.

You can traffic highly-customized pricing plans for your services based on guaranteed maximum and minimum bandwidth levels. Your customers then choose the plan that best fits their needs and budgets. You can change these offerings quickly, and use Policy Controller to offer targeted promotions customized to individual subscribers (such as a deal on their birthday).

Policy Controller is based on the 3GPP standard and is Release 9 compliant. For details on the exact specifications supported, see "Policy Controller Specification Compliance".

Policy Controller Hardware and Software Requirements

This section lists the additional components that your Policy Controller implementation requires:

- There are no special hardware requirements; above and beyond those required by Service Broker. For details see the discussion on hardware requirements in *Oracle Communications Service Broker Installation Guide*.
- Obtain a PCEF to enforce Policy Controller policy decisions. For details, see your PCEF product documentation.
- Obtain a BBERF to connect to your IP-CAN, and handle policy signal flow.
- Obtain a Diameter charging server to charge your subscribers for services (if not using Oracle Communications BRM).

Understanding Policy Controller Terms

You need to understand the following term when using Policy Controller:

Application Function (AF)
AFs are the 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.

Charging Profiles
Charging profiles specify charging information for a Policy Profile. They reference the Rating Groups and Service IDs that you have set up in your charging engine.

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

Policy Control and Charging (PCC) Rules
See Policy 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.

Policy Profile
Policy 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 the Quality of Service and Charging Profiles that determine the level of service and charging details to use. Policy Profiles also include a Charging-Rule-Name AVP that identifies the Policy Profile for activation and deactivation.

Predefined PCC Rules
Predefined PCC rules are rules that are stored in the PCEF. Policy Controller refers to these rules as *static rules*.

Quality of Service (QoS) Profiles

QoS profiles specify maximum or minimum service bandwidth limits to use for a Policy profile. QoS profiles determine how much bandwidth a subscriber is entitled to.

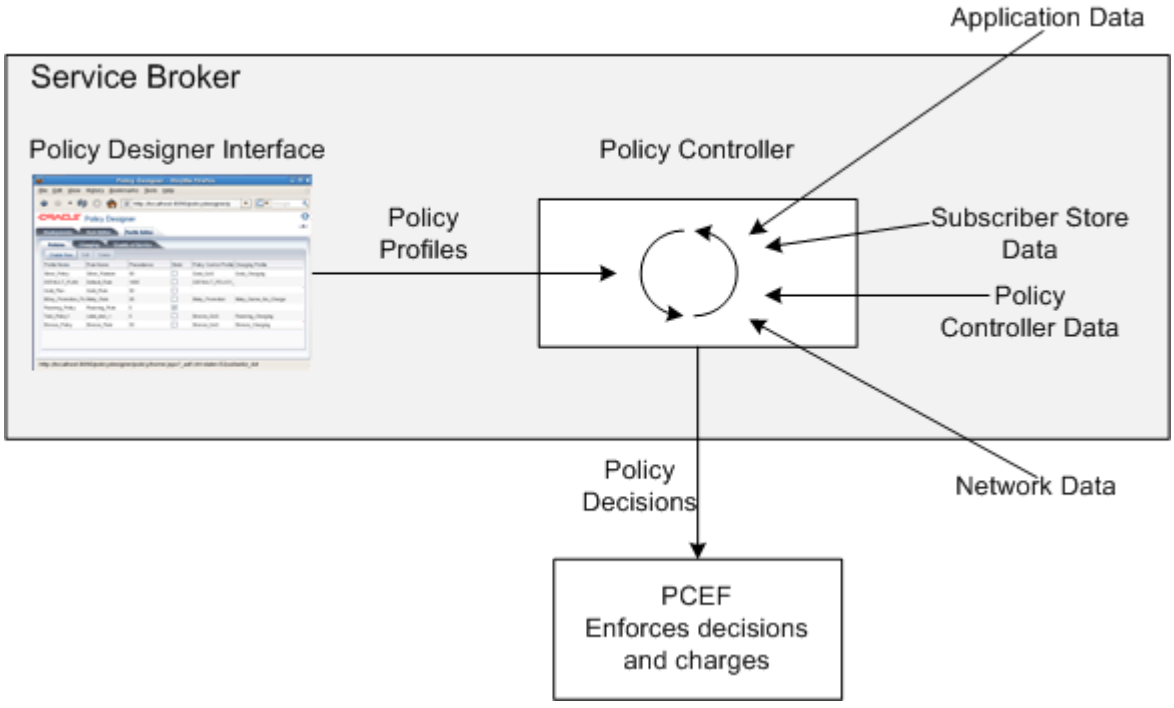
Note: Rule nomenclature for PCRFs can be confusing. This document refers to three types of rules that Oracle's Policy Controller uses:

- You use the Policy Designer **Rule Editor** to create rules that set conditions for service access. Policy Controller uses these rules in your policy decisions to select Policy Profiles. This document refers to these rules as simply "rules."
- *PCC Rules* defined in the 3GPP TS 23.203 v9.9.0 specification are implemented as *Policy Profiles* in Policy Controller.
- Policy Profiles themselves require an entry in a **Rule** field. This rule is either the name of a static rule or base rule, or the value of the Charging-Rule-Name Diameter Gx AVP (Code 1005) from the 3GPP TS 29.212 specification, depending on the type of Policy Profile.

Understanding How Policy Controller Works

Policy Controller is a decision engine. It communicates with your other policy control entities, such as PCEFs, application functions, and Operational Support Systems/Business Support Systems which enforce the bandwidth restrictions and perform charging. You can also decide to let your PCEF to store the charging and policy information using predefined PCC rules. Figure 1–1 shows the various entities that offer input to the Policy Controller decision making process.

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



You can create highly-customized pricing plans for your services based on guaranteed maximum and minimum bandwidth levels. Your customers then choose the plan that fits their needs and budget. You can change these offerings quickly, and also use Policy Controller to offer "on the fly" promotions, or promotions customized to individual subscribers.

Policy Controller is service- and subscriber profile-aware, so you can offer highly customized services based on service or profile parameters. For example you could offer special deals only good on each subscriber's birthday.

Once Policy Controller makes its decisions, it passes them on to your PCEF to implement. Your PCEF is the entity that directs the AF to changes the level of service and charging engine to charge for those services.

Figure 1–2 shows the software components of a Policy Controller implementation.

Figure 1–2 Policy Controller Architecture Overview

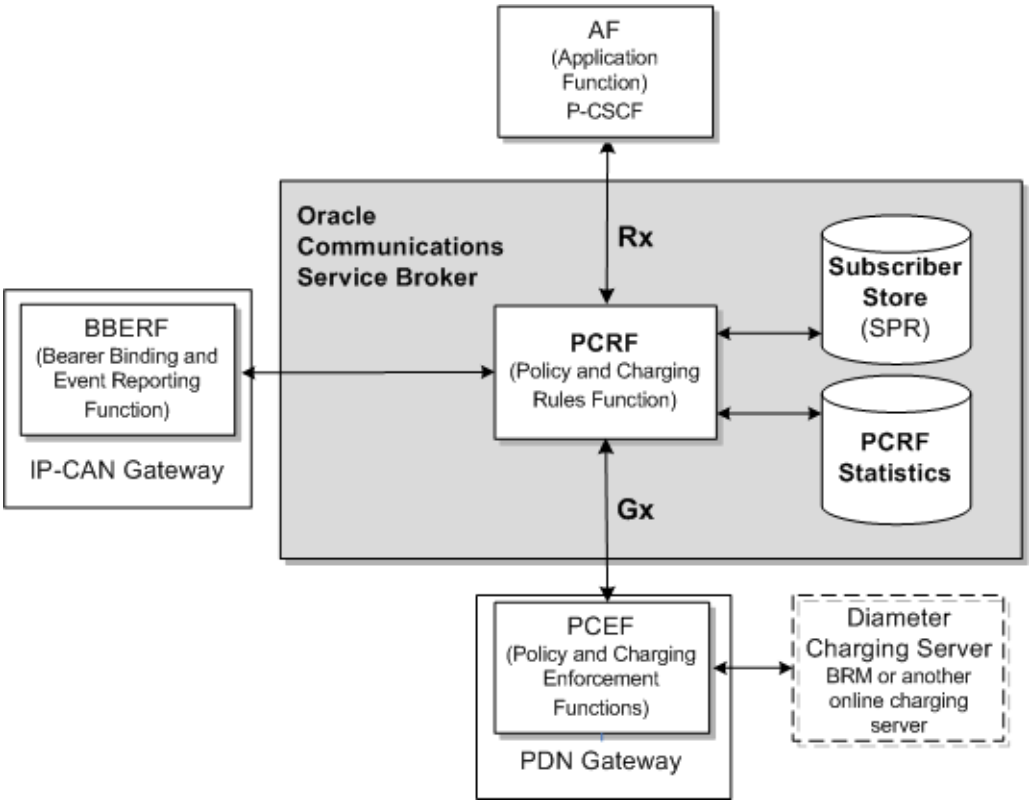
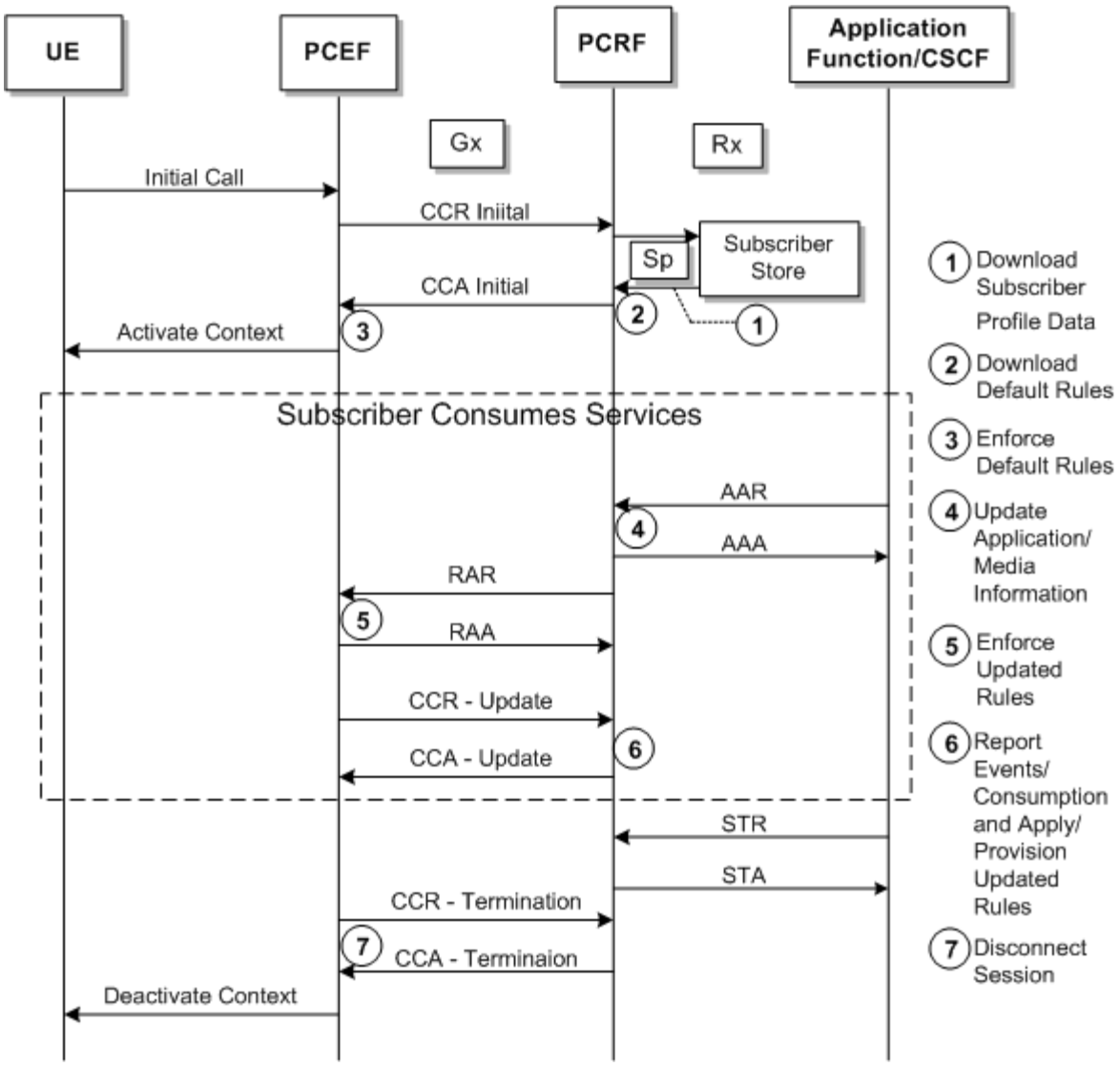


Figure 1–3 shows the call flow of a typical Policy Controller session. The call flow helps you understand the Policy Controller components and its features.

Figure 1–3 Typical Policy Controller Call Flow



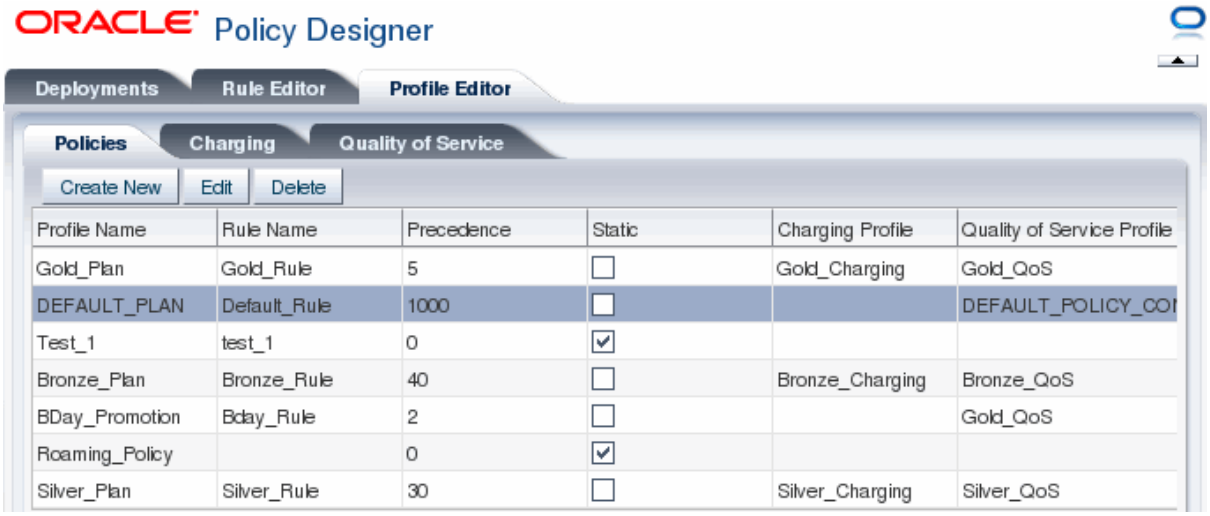
Understanding Policy Designer

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

- Create Policy Profiles that your PCEF implements as PCC rules.
- Create Charging Profiles that contain charging information for the services in a Policy Profile.
- Create Quality of Service Profiles that specify the bandwidth levels that a subscriber is entitled to for a Policy Profile.
- Associate Policy Profiles with Charging Profiles and Quality of Service Profiles.
- Create rules that decide which Policy Profiles to apply to which subscribers.

Figure 1–4 shows the **Profile Editor** tab of the Policy Designer that you can use to create Policy Profiles and associate them with Charging and QoS Profiles. The Policy Designer shows three tabs: **Deployments**, **Rule Editor** and **Profile Editor**. The **Policies** subtab is shown displayed with a variety of Policy Profiles, one per row.

Figure 1–4 The Policy Designer User Interface



The web-based Policy Designer interface is closely related to its Oracle Business Rules (OBR) predecessor. Your personnel can change business rules from any web browser without stopping business processes.

Note: The Policy Designer interface is incompatible with the stand-alone Administration Console. You can not run them both on the same Service Broker implementation.

Using Rules to Select Policy Profiles

The heart of the Policy Controller feature is the Policy Designer **Rule Editor** tab that you can use to create rules that select (or reselect) Policy Profiles. These rules are tests to decide whether a subscriber is entitled to the bandwidth capabilities and limits in a specific Policy Profile. Each rule takes the form of an if-then statement (or collection of if-then statements).

Rules can be session- or event-based. Policy Controller is based on deep packet inspection (DPI) so it also allows service data flow traffic gating; you can allow or disallow certain kinds of service data flow based on its origin.

The **Rule Editor** tab contains an extensive set of data and functions to use in your if-then statements. Typically these rules are reinterpreted occasionally as a session progresses and subscribers consume services.

If you need to create multiple rules that use the same if statement, it is considerably more efficient to create a fact out of the if statement and then reference the fact in those rules.

Using Policy Profiles to Set Bandwidth and Charging Levels

Using the Policy Designer, you create Policy and Charging Profiles that set bandwidth limits and charging instructions. Once you have created profiles that correspond to your products, you create rules to decide which subscribers are entitled to those policies (products). Rules can be simple or complex; they act on:

- Application and media information from the AF.

- Internal rule configurations (other Policy Profiles).
- Subscriber information from the SPR.
- Information such as event triggers from the PCEF.

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

- Pull (solicited provisioning). Sending Policy Profiles in response to a request from the PCEF. The request is answered in a CCA message.
- Push (unsolicited provisioning). Sending Policy Profiles to the PCEF based on new data. To do this the Policy Controller includes the profiles in an RA-request message. No CCR/CCA messages are triggered by this request.

Event triggers from the PCEF are often used as catalysts for push provisioning. For example, you can create rules that change behavior based on a subscriber's current location, credit status, Connectivity Access Network changes, and so on.

Policy Profiles contain information about a specific service and may also contain:

- A Quality of Service Profile containing bandwidth levels and limits for the subscriber/service.
- A Charging Profile with charging and rating information for the subscriber/service.

Policy Profiles generally contain information about a specific service, but they are flexible and you could create one that applies to all services. Whether you choose to include a Quality of Service Profile, a Charging Profile, or both depends on your implementation's requirements. If the Policy Profile does not contain a Quality of Service Profile or Charging Profile then your PCEF must provide that information.

The 3GPP TS 23.203 specification makes a distinction between static and dynamic PCC rules. You can create either static or dynamic Policy Profiles by using the Policy Designer. You specify the static/dynamic status when you create the profile, and you can reverse that decision later if you need to. You reference a static rule by adding the rule name and checking the **Static** box when you create a Policy Profile.

About Limiting Subscriber Bandwidth

You can use Policy Controller to limit the level of service that a subscriber may access. At its simplest, you can limit upload or download bandwidth to maximum or minimum bits per second rates. More complex scenarios include granting a subscriber access to multiple services, and establishing service-based consequences if they use more than their monthly allotment of bandwidth.

You can limit Subscriber bandwidth using combinations of the following metrics:

- Subscriber ID.
- Service.
- Content - For example, type of data such as video download or voice conversation.
- Time-of-day.
- Timezone.
- Location - For example, local or roaming.
- UE (User Equipment) - For example, some devices are allowed and others not.
- Access network type such as WiMax or GPRS.

- Life cycle state - For example you might degrade service for subscribers with suspended accounts.

About Guaranteeing Minimum Subscriber Bandwidth

Quality of Service (QoS) level guarantees that a subscriber's bandwidth never goes below a threshold that you specify. You can use the following parameters to set a QoS level:

- Duration of session or event.
- Type of Service.
- Time-of-day.
- Type of UE.
- Location.

Using Multi-service Products

Policy Controller can create policies that are valid for all services, or per-service. Managing bandwidth per service allows 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 policies that are valid for all services.

These profiles are optional. You can specify this information using rules and rulesets that you create using the 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.

Using Subscriber Profile Information to Change Service Offerings

Policy Controller uses Service Broker Subscriber Store to retrieve the subscriber profile information listed in "About Limiting Subscriber Bandwidth". You can use any of these subscriber profile fields in your rules to create custom service offerings for each individual subscriber. For example you could offer a special service valid only on each subscriber's birthday. 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*.

Putting it all Together

To use Policy Controller, first follow the instructions in "Configuring Service Broker for Policy Controller" to install and set up your policy implementation. Then you are ready to start the Policy Designer user interface. Once it's started, follow the instructions in "Specifying Service and Charging Information with Policy Profiles" and "Creating Rules and Rulesets" to create:

- Policy Profiles that define the bandwidth capabilities and limitations. Including:
  - PCC Policy Profiles that specify the limits and capabilities.
  - Charging Profiles that specify how much to charge for the specified bandwidth.
- Rules that select the Policy Profiles for a subscriber to use.

Policy Controller Specification Compliance

Policy Controller adheres to the specifications listed in Table 1–1.

Table 1–1 Policy Controller Diameter-based Protocol Interfaces

Standard	Title
3GPP TS 23.203 v9.9.0 (2011-06)	3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 9).
3GPP 29.214 v9.8.0 (2011-09)	3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Policy and Charging Control over Rx reference point (Release 9).
3GPP TS 29.212 v9.7.0 (2011-06)	3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Policy and Charging Control over Gx reference point (Release 9).

Requests for rules have mandatory and optional Attribute Value Pairs (AVPs). See "Policy Controller Protocol Reference" for information on the supported and mandatory Diameter Rx and Gx AVPs.

2

Configuring Service Broker for Policy Controller

This chapter explains how to install and configure a Policy Controller implementation.

Installing and Starting Policy Controller

Policy Controller is a part of Oracle Communications Service Broker 6.0, so you must install that release. For details on installation see *Oracle Communications Service Broker Installation Guide*. Be sure to select the **Policy Controller 6.0.0.0** option during the installation.

About Creating a Domain and Managed Server for Policy Controller

You must create a Service Broker managed server and domain for each Policy Controller implementation. See the discussions on creating the domain and adding a managed server to the domain in *Oracle Communication Service Broker Installation Guide* for details. That manual also has instructions for starting the domain and managed server.

About Configuring Policy Controller

This chapter explains how to configure a Policy Controller implementation. This section assumed that you have installed Policy Controller as part of Oracle Communications Service Broker 6.0 and created a domain for it as described in "About Creating a Domain and Managed Server for Policy Controller".

Configuring a Policy Controller implementation involves these tasks:

- Create, configure, and provision a Service Broker Subscriber Profile Store to store subscriber data for Policy Controller to use.
- Configure the Service Broker SSU Diameter to connect the Policy Controller PCRF to AFs (Application Functions) and PCEFs (Policy and Charging Enforcement Functions).
- Configure Policy Controller Global Parameters.
- Configure Policy Controller System Parameters.
- Configure Policy Controller Data Store (BDB or JDBC).
- Configure Your PCEF Server to work with Policy Controller.
- Configure your AF server to work with a Policy Controller.
- Configure and start the Policy Designer.

Create, Provision, and Connect a Subscriber Profile Repository

Policy Controller obtains individual subscriber information from a Subscriber Profile Repository (SPR) set up for that purpose. Policy Controller uses SPR data to interpret policy rules, and decide which services and bandwidth levels a subscriber is entitled to. The Service Broker Subscriber Store feature is an SPR designed for this purpose. See *Oracle Communications Service Broker Subscriber Store User's Guide* for instructions on how to set up Subscriber Store, provision it with data for the Policy Controller to use, and configure an Sp reference point to communicate with.

If you use another SPR for this purpose, see that product documentation for instructions on how to configure and provision that SPR, and communicate with its Sp reference point.

Table 2–1 lists the default Subscriber Store parameters.

Table 2–1 Policy Controller Subscriber Profile Parameters

Parameter	Multiciplicity	Level	Data Type	Description
Subscriber-Id	1..*	N/A	String	The subscriber ID value.
Subscriber-Id-Type	1..*	N/A	Integer	The subscriber ID type.
accountState	1	Global	String	The status of the account, such as active or inactive.
AccountType	1	Global	String	Can be **Prepaid**, **Postpaid**, **Hybrid**.
SubscriberActivationDate	1	Global	Calendar	A calendar date in the form of *YYYYMMDD*.
DateOfBirth	1	Global	Calendar	A calendar date in the form of *YYYYMMDD*.
SubscriberCategory	1	PCRF	String	These are the plan categories that you create to sell to subscriber. For example **premium**, **average**, and **economy**.
HomeZone	*	PCRF	HomeZone	The subscriber's home zone, indicating the subscriber's location and location type.

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 *Oracle Communications Service Broker Subscriber Store User's Guide*.

Configure the SSU Diameter to Mediate Policy Controller/AF Service Data Flow

Set your PCEF servers and AFs (application functions) as a Diameter peers, and define Diameter routing rules that directs all Gx service data flow from your PCEF to Policy Controller, and Rx service data flow from Policy Controller to the AF with the following steps. This section assumes that you know the host name and IP address of each of your PCEF and AF servers.

For details on the supported Gx and Rx messages see "Gx Command Codes Supported by Policy Controller" and "Diameter Rx Command Codes Supported by Policy Controller".

1. Open the Service Broker Administration Console.
2. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **SSU Diameter, DIAMETER**, then **Diameter Configuration**.
3. Click **Edit**, then the plus sign button to create a node (if not using the default).
4. Click the **Peers** subtab.
5. Click the plus sign button to create a new peer.
6. In the New Data popup screen, enter your PCEF identifying data:
   - **Address** - The URL of your PCEF server.
   - **Host** - The host name of your PCEF server.
   - **Port** - A server port number to use on your PCEF.
   - **Protocol** - A service data flow protocol to use.
   - **Watchdog** - Enables/disables the watchdog timer.
7. Click **OK**.
8. Repeat these steps to specify each of your PCEF servers and AF servers as a peer.

Configure the Diameter SSU 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. Open the Service Broker Administration Console.
2. Navigate to **Platform**, **OCSB**, **Signaling Tier**, **SSU Diameter, SSU Diameter, Routing**, then **Incoming Routing Rules**.
3. Click Edit, then **New** to create a new Incoming Routing Rule for Gx traffic.
   - **Name:** Enter an informal name for the new rule.
   - **Priority:** Leave the **0** default priority.
   - **Module Instance:** Enter **ssu:ocsb/pcrf**.
4. Click OK.
5. Click the **Incoming Routing Criteria** subtab to define the Gx traffic routing criteria.
6. Select your new incoming routing rule from the **Parent** dropdown list.
7. Click **New** and enter the following 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.
8. Click the **Incoming Routing Rules** subtab.
9. Click **New** to create a new Incoming Routing Rule for Rx traffic.

- **Name:** Enter an informal name for the new rule.
- **Priority:** Leave the **0** default priority.
- **Module Instance:** Enter **ssu:ocsb/pcrf**.

10. Click the **Incoming Routing Criteria** subtab to define the Rx traffic routing criteria.
11. Select your new incoming Rx routing rule from the **Parent** dropdown list.
12. Click New and enter the following routing rule criteria:
    - **Name:** Enter an informal name for the criteria.
    - **Attribute:** Select **APPLICATION_ID** from the dropdown list.
    - **Value:** Enter **16777236** to specify Rx service data flow.
13. Click Commit to save your new rules.

Configure Policy Controller Global 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 know the values you use to indicate rating groups and PCC service identifiers.

Configure Policy Controller Administration Console Global Parameters

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

By default Policy Controller comes with the global parameters listed in Table 2–2.

Table 2–2 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.
events-subscribed	15,16,17,22	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. The default trigger events are: ■ **OUT_OF_CREDIT** (15) ■ **REALLOCATION_OF_CREDIT** (16) ■ **REVALIDATION_TIMEOUT** (17) ■ **SUCCESSFUL_RESOURCE_ALLOCATION** (22).
primary-event-charging-fn	N/A	URL	The primary online charging server address.
secondary-event-charging-fn	N/A	URL	The secondary online charging server address.
primary-charging-collection-fn	N/A	URI	The primary online collection server address.

Table 2–2 (Cont.) Default Policy Controller Global Parameter Settings

Global Parameter	Default Value	Data Type	Description
secondary-charging-collection-fm	N/A	URL	The secondary online collection server address.
online	N/A	Boolean	Specifies whether online charging is allowed.
offline	N/A	Boolean	Specifies whether offline charging is allowed.
access-nw-charging-address	N/A	URL	A credit card charging IP address to use.
install-default-plan	true	Boolean	Whether to automatically install the default Policy Profile.

Set Policy Controller system parameters in the **PCRF** tab of the Service Broker Administration Console with the following steps:

1. Bring up the Service Broker Administration Console.
2. Navigate to **PCRF**, **OCSG**, **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.
6. Click **Apply**.
7. Click **Confirm**.

About Execution Blocks

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

Configure the System Parameters

By default, Policy Controller uses the timers and timer values listed in Table 2–3 to control and protect Rx and Gx sessions. These settings work for a test and evaluation system and may also work for a production implementation. Configure them to fit your implementation's requirements.

Table 2–3 RX and Gx Timer Names, Default Values, and Descriptions.

Timer	Timer Name (alternate name)	Default Value (ms)	Description
Gx Session Duration timer	TGXSESSION (Tcc)	36000000	Determines how long to keep sessions with pending Gx traffic open. Prevents stale sessions and releases session-related information. This timer is started on each Gx CCA-Initial or CCR-Update message, and canceled on any subsequent Gx CCR received from the network.

Table 2–3 (Cont.) RX and Gx Timer Names, Default Values, and Descriptions.

Timer	Timer Name (alternate name)	Default Value (ms)	Description
Gx Session Guard Timer	TGXGUARD (Tg)	1000	This timer is started each time a Gx request arrives at the PCRF, and is canceled when the response is sent. If the timer expires the response is sent with an result code of **3002**.
Rx Session Duration Timer	TRXSESSION (Tcc)	36000000	Determines how long to keep sessions with pending Rx traffic open. Prevents stale sessions and releases session-related information. This timer is started by the Rx AAR-Initial message, and canceled by the STR message.
Rx Session Guard Timer	TRXGUARD (Tg)	1000	Determines how long to keep sessions with pending Rx traffic open. Prevents stale sessions and releases session-related information. This timer is started on each Rx CCA-Initial or CCR-Update message, and canceled on any subsequent Gx CCR received from the network.

Change each timer settings with the following steps:

1. Bring up the Service Broker Administration Console.
2. Navigate to **PCRF**, **OCSG**, **Execution Blocks**, then **System Parameters**.
3. Click the **PCRF System Parameters** tab, then the **Timers** subtab.
4. Click **Edit**.
5. Select a timer to change.
6. Click **Update**.

   An **Update** window appears.
7. Enter new values in the timer-value fields.
8. Click **OK** to make your changes take effect.

Configure Data Storage for Policy Controller Data

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*.

Configure Your PCEF Server

This section explains the steps necessary to make your PCEF server work with Policy Controller. For details on the tasks to perform see your PCEF product documentation.

See your PCEF product documentation for instructions on how to set up and configure your PCEF server. Specifically:

- Configure your PCEF server to use the Diameter Gx messages listed in "Gx Command Codes Supported by Policy Controller".

Configure Your AF Server

This section explains the steps necessary to make your AF Server work with the Service Broker Policy Controller. For details on the tasks to perform see your AF product documentation.

- Configure your PCEF server to use the Diameter Gx messages listed in "Diameter Rx Command Codes Supported by Policy Controller".

Start and Configure Policy Designer

The following sections explain how to start and configure Policy Designer.

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 web access Administration Console with these commands (this also starts the Policy Controller JETTY server):

   ```
   cd Oracle_home/ocsg60/admin_console
   ./web.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.basic.auth=true** in *Oracle_home*/**ocsg60/admin_console/properties/web.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 stand-alone version of the Administration Console.

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

   ```
   cd Oracle_home/ocsg60/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 these default port numbers by using the **oracle.ocsb.app.rcc.pcrf.gui.port** (SSL off) or **pcrf.gui.http.port.secure** (SSL on) system property.

   If you enabled basic authorization (**axia.basic.auth=true** in *Oracle_home*/**ocsg60/admin_console/properties/web.properties**) you are prompted for the username and password that you entered when you created the Policy Controller domain.

   This example starts the 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. You can change this by following these steps:

1. Open a command-line shell.
2. Add this entry to the *Oracle_Home*/**admin_console/common.properties** file:
   - If you set SSL to false:

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

   - If you set SSL to true:

     ```
     pcrf.gui.port.secure=Port_Number
     ```

   Where *Port_Number* is the new port number to use.
3. Restart the Administration Console. For details, see the discussion on starting the Web Administration Console Server in *Oracle Communications Service Broker System Administrator's Guide*.

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_console/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 the discussion on starting the Web Administration Console Server in *Oracle Communications Service Broker System Administrator's Guide*.

3

Specifying Service and Charging Information with Policy Profiles

This chapter explains how to create Policy Profiles that specify Quality of Service (QoS) limits for services and associate those limits with charging information from your charging engine.

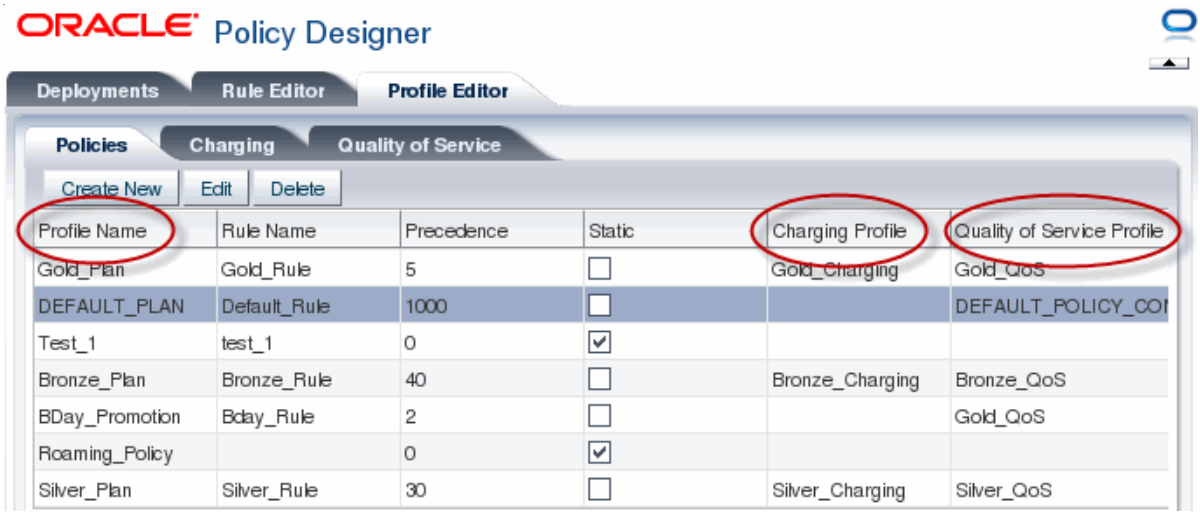
This chapter assumes that you know the details for any predefined PCC rules that you will reference, including the Charging-Rule-Name AVPs to use.

About Policy Profiles

Policy Profiles are the Policy Controller implementations of PCC rules as defined in the 3GPP TS 23.203 v9.9.0 specification. You can use Policy Profiles to specify QoS and charging details to apply to subscribers. Each Policy Profile contains a *Quality of Service Profile* that specifies the actual bandwidth limits, QoS Class Identifier (QCI), and gate status, and a *Charging Profile* that associates those limits with charging information in your charging engine. Neither QoS Profiles nor Charging Profiles are required in Policy Profiles. If you are going to use them however, you must create Charging Profiles and/or QoS Profiles first. You create these profiles using the Policy Designer **Profile Editor** tab and its subtabs.

You can use the Policy Designer interface's **Policies** subtab to create and manage Policy Profiles. Figure 3–1 shows the Policy Designer with the **Profile Editor** tab and **Policies** subtab selected. The **Profile Editor** tab shows the Policy Profiles listed by name in the column on the left, and columns on the right list the Charging Profiles and Quality of Service Profiles that each Policy Profile is associated with. The **Profile Name**, **Charging Profile** and **Quality of Service Profile** column headings are circled in red.

Figure 3–1 Policy Designer Profiles Tab



Policy Designer requires that you always have a default profile policy called **DEFAULT_PLAN** to charge subscribers if no other profiles apply. You need to define Charging Profiles and Quality of Service Profiles for it to actually charge subscribers. This policy profile has a precedence level of 1000 to ensure that it always has the lowest priority of all policies. You cannot delete this default profile.

Remember that you may also use pre-defined PCC rules from your PCEF with the Policy Designer. To use a predefined PCC Rule in Policy Designer, simply create a new Policy Profile using predefined PCC rule's name as it appears in the PCEF and check the **Static** box. A **Static** check box marks Policy Profiles as predefined PCC rules on the main Policies pane.

Figure 3–2 shows the Profile Editor tab and **Quality of Service Profile** subtab with the Quality of Service Profiles listed one per row. Quality of Service Profiles specify details for the actual bandwidth limits for subscriber accounts, and the settings that determine which services have priority when competing for available resources.

Figure 3–2 Quality of Service Profiles Subtab

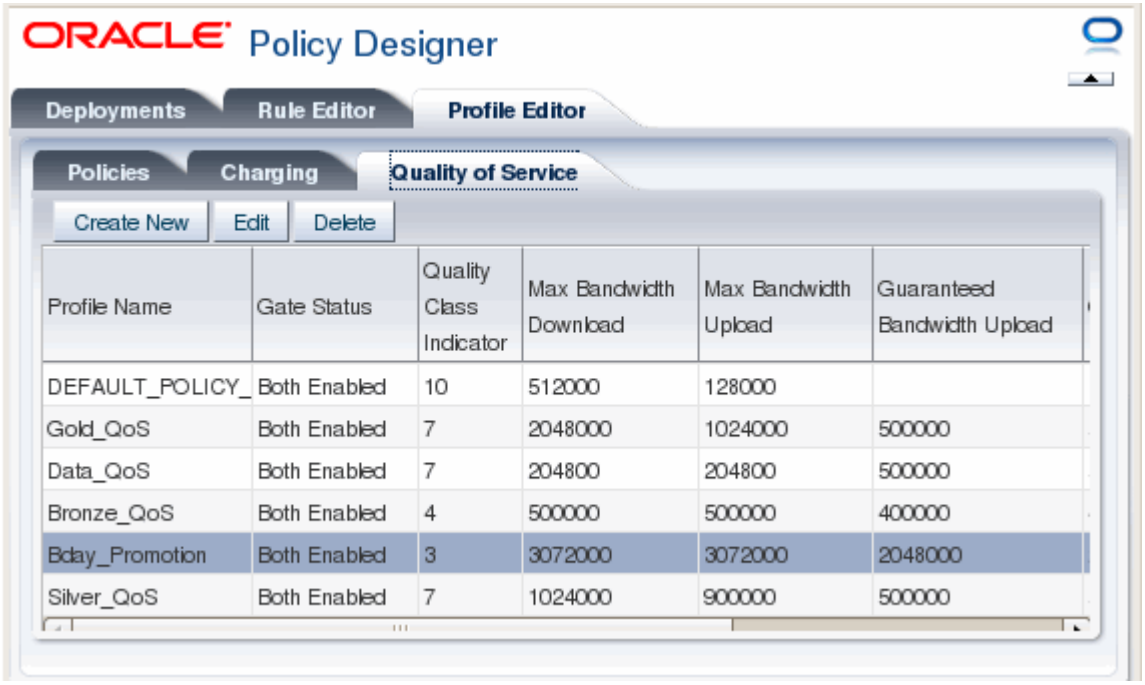
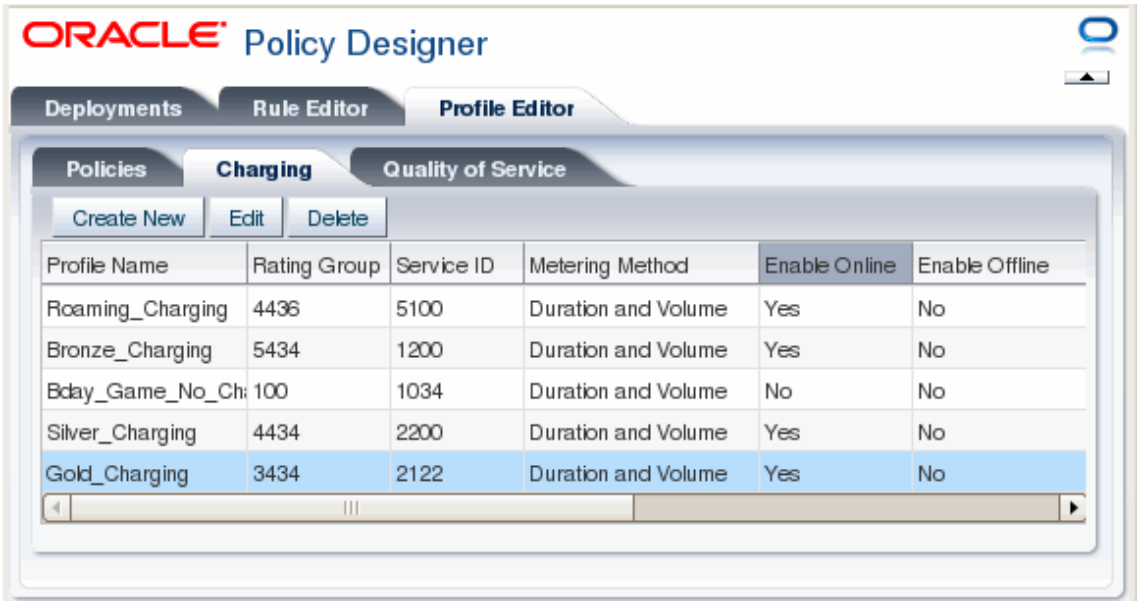


Figure 3–3 shows the **Profile Editor** tab and the **Charging Profiles** subtab. Charging Profiles do not contain specific currency amounts. Instead they reference billing and charging details specified in your charging engine and your other policy implementation entities.

Figure 3–3 Charging Profiles Subtab



Planning Your Profiles

The procedures below assume that you already know the details of the policies you will create. To create policies you should already know details about your policy implementation such as:

- The bandwidth limits to set.
- How your policies relate to each other.

  Precedence level determines the order in which Policy Profiles are evaluated. These values are positive integers with lower integers having higher priority. The default profile has a precedence level of 1000. It should always be evaluated with the lowest priority (have the highest precedence level), so be sure to use lower integers for your other Profiles or if needed, raise the default profile priority.

  Profiles with the same priority are evaluated in a random order so be sure profiles with the same priority have non-overlapping criteria.
- Charging details such as:
  - The rating groups and service IDs of the services you will charge for.
  - The IP addresses and ports of the entities you will accept billing service data flow from.
  - The configuration details of the charging engine that you use.

Creating a Quality of Service Profile

Follow these steps to create a new Quality of Service Profile:

1. Start the Policy Designer.
2. Navigate to **Profile**, then **Quality of Service**.
3. Click **Create New**.

   The **Quality of Service** screen appears.
4. Specify parameters for the new Charging Control Profile:
   - **Profile Name** - A descriptive name for the profile.
   - **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** - Reserved for Oracle use.
   - **Quality Class Indicator** - 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 QCI values listed in Table 3–1. Note that multiple QCI values can be useful for the same services.

Table 3–1 Supported 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).

- **Max Download Bandwidth (Bit/s)** - The maximum allowed bandwidth rate.
- **Max Upload Bandwidth (Bit/s)** - The maximum allowed data upload rate.
- **Guaranteed Download Bandwidth (bit/s)** - The minimum data download rate to use.
- **Guaranteed Upload Bandwidth (Bit/s)** - The minimum data upload rate to use.

5. Click **Save**.

   The new Quality of Service profile appears in the **Quality of Service** subtab.

Creating a Charging Profile

Follow these steps to create a new Charging Profile:

1. Start the Policy Designer.
2. Navigate to **Profile** Editor, then **Charging**.
3. Click **Create New**.

   The **Charging** screen appears
4. Fill in the parameters for a new charging profile:
   - **Profile Name** - A descriptive name for the new Charging Profile.
   - **Rating Group** - The name of the rating group to use. See your charging engine documentation for details.
   - **Service ID** - The Service ID to use. See your charging engine documentation for details.
   - **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.
- **Enable Online** - **Yes** specifies service data flow using on-line charging; **No** ignores service data flow using online charging.
- **Enable Offline** - **Yes** specifies service data flow using offline charging; **No** ignores service data flow using offline charging.

5. Select **Save**.

   The new Charging Profile appears in the **Charging** subtab.

Creating Policy Profiles

This section explains how to create the Policy Profiles that are called by your rules to specify quality of service levels and charging information. Policy Profiles typically include a Quality of Service Profile and a Charging Profile, but they are not required.

There are two types of Policy Profiles that you can create as explained in the sections that follow:

- Policy Profiles that use QoS Profiles and Charging Profiles to set bandwidth levels and limits, and charging information.
- Static Policy Profiles that obtain bandwidth levels and limits and charging information from either individual or base level predefined PCC rules.

Creating a Policy Profile Using QoS and Charging Profiles

This section explains how to create a Policy Profile using the bandwidth settings and charging information contained in QoS and Charging Profiles. This section assumes that you have:

- Created the Charging Profiles and Quality of Service Profiles to use in each Policy Profile. For more information, see "Creating a Charging Profile" and "Creating a Quality of Service Profile".
- Obtained the Charging-Rule-Name AVP name(s) to use with the new Policy Profile.

To create a Policy Profile to be stored in Policy Controller:

1. Start the Policy Designer.
2. Navigate to **Profile Editor**, then **Policies**.
3. Click **Create New**.

   The **Policies** screen appears.
4. Fill in a **Profile Name** (mandatory). This is an informal unique name for the new profile.
5. Fill in a **Rule Name** (mandatory). This is the Diameter Gx Charging-Rule-Name AVP as defined in the 3GPP TS 29.212 v9.7.0 (2011-06) specification.
6. Click **Edit Flow Description** to bring up the Flow Description Editor shown in Figure 3–5. These fields filter service data flow by using the Rx **Flow-Description** AVP parameters listed below. For details, see the 3GPP TX 29.214 v9.x specification.

Fill in any parameters required to filter service data flow for the profile to affect:

- **Direction** - IN or OUT. Applies the profile to service data flow coming in to, or being sent out from Policy Controller.
- **Protocol** - (Required) Filters the service data flow by protocol. Applies the profile to service data flow matching the protocols you specify.
- **Source Address** - (Required) IP/mask number. Applies the profile to service data flow matching entities specified by a IP address/subnet mask.
- **Source Port(s)** - A comma-separated list 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:

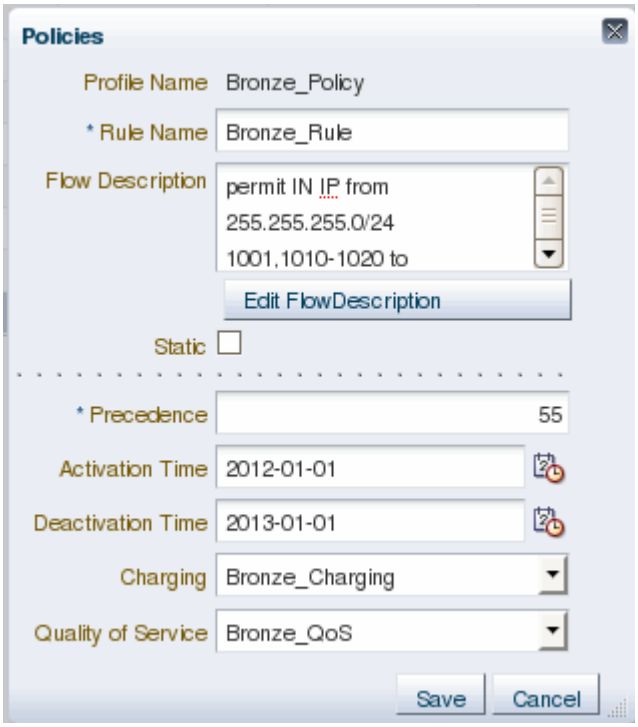
  ```
  2000,2002,4010-4020
  ```

- **Destination Address** - (Required) IP/mask number. Applies the profile to service data flow destined for entities specified by an IP address/ubnet mask.
- **Destination Port(s)** - A comma-separated list 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.

7. Click **Save**.

   The **Policies** screen reappears with the flow information added. Figure 3–4 shows an example.

Figure 3–4 Updated Policies Screen



8. Fill in the remaining Policy Profile fields:

- **Precedence** - (Mandatory) Specifies the Policy Profile evaluation order. See "Planning Your Profiles" for details on the precedence levels.
- **Activation Time** - Sets the date and time that the new Policy Profile becomes active. Select the calendar icon and enter a month, year, day, and time for the policy to become active. The default values make the Policy Profile active from the current date and time. Select OK to confirm your choices.
- **Deactivation Time** - Sets the date and time that the new Policy Profile expires. Select the calendar icon and enter a month, year, day, and time for the policy to stop being active. When you bring up the calendar pane, it applies the current date and time as the default values for deactivating the Policy Profile, so be sure to change them. Select OK to confirm your choices.
- **Charging** - Select a Charging Profile from the drop-down list.
- **Quality of Service** - Select a Quality of Service Profile from the drop-down list.

9. Click **Save**.
10. The new Policy Profile appears in the **Profile Name** column.

Creating a Policy Profile Using Predefined PCC Rules

This section explains how to create a Policy Profile that uses bandwidth metrics and charging information from predefined PCC rules already stored on your PCEF.

This section assumes that you have:

- Created the Charging Profiles and Quality of Service Profiles to use in each Policy Profile. For more information, see "Creating a Charging Profile" and "Creating a Quality of Service Profile".
- Obtained the names and Charging-Rule-Name AVP name(s) of any predefined PCC rules that you are using.

To create a Policy Profile using Policy Controller:

1. Start the Policy Designer.
2. Navigate to **Profile Editor**, then **Policies**.
3. Click **Create New**.

   The **Policies** screen appears.
4. Check the Static box.

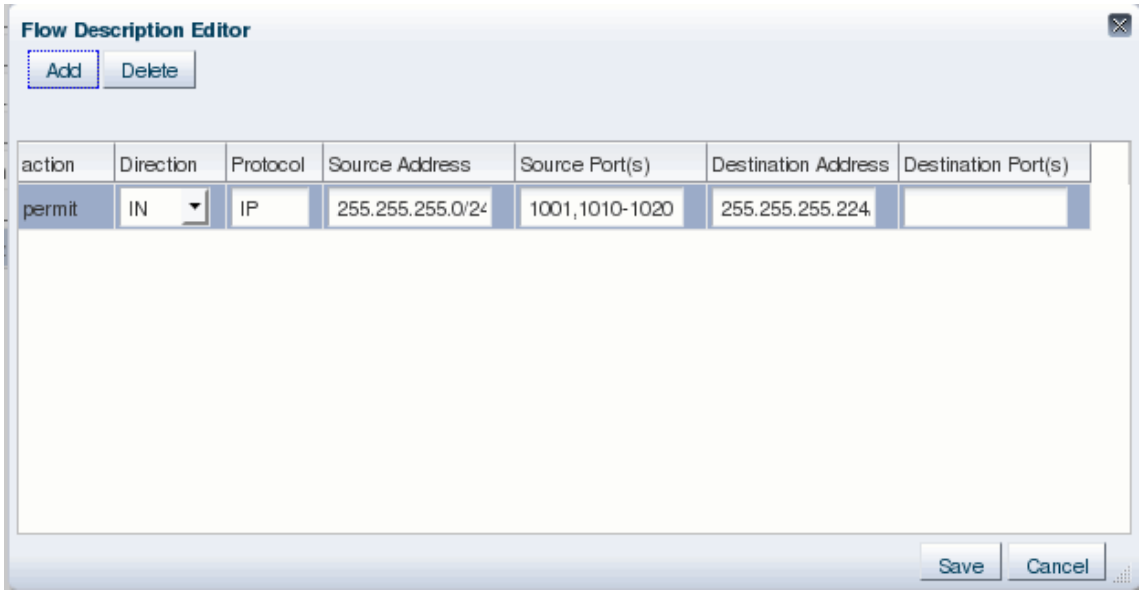
   The unneeded fields disappear and the **Rule Name is Rule Base Name** check box appears.
5. If the PCC rule is a base rule, check the **Rule Name is Rule Base Name** check box. This installs all of the PCC rules in the base rule.

   Note: Because these are “grouped” rules there is no single rule name to display in the **Rule Name** column of the **Policies** tab. This field is left blank.

6. Fill in a **Profile Name** (mandatory). This is an informal name for the Policy Profile that is displayed in the **Profiles** tab.

7. Fill in a **Rule Name** (mandatory). This is the name of a predefined PCC rule or base rule.
8. Click **Edit Flow Description** to bring up the Flow Description Editor shown in Figure 3–5. These fields filter service data flow by using the Rx **Flow-Description** AVP parameters listed below. For details, see the 3GPP TX 29.214 v9.x specification.

Figure 3–5 Example Flow Description Editor Screen



Fill in any parameters required to filter service data flow for the profile to affect:

- **Direction** - IN or OUT. Applies the profile to service data flow coming in to, or being sent out from Policy Controller.
- **Protocol** - (Required) Filters the service data flow by protocol. Applies the profile to service data flow matching the protocols you specify.
- **Source Address** - (Required) IP/mask number. Applies the profile to service data flow matching entities specified by a IP address/subnet mask.
- **Source Port(s)** - A comma-separated list 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
  ```

- **Destination Address** - (Required) IP/mask number. Applies the profile to service data flow destined for entities specified by an IP address/ubnet mask.
- **Destination Port(s)** - A comma-separated list 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.

9. Click **Save**.

   The **Policies** screen reappears with the flow information added.
10. Click **Save**.

11. The new Policy Profile appears in the **Profile Name** column.

4

Creating Rules and Rulesets

This chapter describes how to create rules and rulesets using the **Rule Editor** tab of the Policy Designer.

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, see *Oracle Fusion Middleware User's Guide for Oracle Business Rules* at:

http://docs.oracle.com/cd/E17904_01/integration.1111/e10228/toc.htm

About Rules, Rulesets, and Dictionaries

Policy Controller uses rules to select policy profiles to apply to subscribers.

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

A rule has two parts: an IF section and a THEN section. The IF section contains a condition based on evaluation of data from various sources. The THEN section contains actions that can be performed. 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 Policy Profile to a subscriber or removing a Policy Profile from a subscriber, but they may involve other actions as well.

You can create collections of rules to work together with one another.

You can set up the rules to select a single Policy Profile for the entire service flow. Policy Controller also allows you to reinterpret the rules as the service flow continues and the input data, such as the approach of a resource limit, changes.

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

The following components, which are related to rules, appear in the rule editor:

Facts

Rules are based on facts. A fact represents a piece of data. In the context of the rules engine, a fact is an asserted instance of a class. It has a type and a value.

Rulesets

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

Bucketsets

A bucketset defines a set of values for a particular fact or a property of a fact.

Decision Functions

A decision function provides a contract for invoking rules. Policy Controller uses a single predefined decision function: **PCRF_DF**.

Dictionaries

A dictionary is a container for any number of rulesets, and their supporting bucketset and set of decision functions. They can be saved as is xml files with a **.rules** extension. You can import a dictionary into the rule editor to modify it, and then export it from the rule editor to a file.

The general workflow for implementing rules is:

1. Create a ruleset.
2. Create the rules in the ruleset.
3. Create the bucketset and decision functions as needed to support the rules.
4. Validate the ruleset.
5. Deploy the ruleset and supporting bucketset and functions as a dictionary. Once deployed, the rules are applied to your Policy Controller subscribers.

You perform these procedures in the **Rule Editor** tab of the Policy Designer. The **Deployments** tab displays the rulesets for individual dictionaries, but does not show details for the bucketset and functions.

When you click the **Rule Editor** tab, a default dictionary of currently deployed rules is loaded into the rule editor. This dictionary contains facts on which you can base your rules. You can also create a custom fact for use in your local rules.

To save an incomplete set of rulesets, with their bucketset, and decision functions to work on later, export the dictionary that contain them to an external **.rules** file. When you are ready to resume work, import the **.rules** file containing the dictionary back into the rule editor. When you are ready to resume work, import the **.rules** file containing the dictionary back into the rule editor.

A rule is dynamically associated with a policy when an **Assert New** action asserts the **Out_InstallPolicy** fact. A rule is dynamically disassociated from a policy when an **Assert New** action asserts the **Out_RemovePolicy** fact. See "Defining an Assert New Action" for more information.

You can create a policy profile from the **Policy Profiles** menu of the rule editor, as well as from the profile editor. See "Creating Policy Profiles" for information on how to do this.

About Advanced Settings

In the rule editor, you have the option of showing or hiding advanced settings for rules and rulesets by toggling these advanced settings icons:

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

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 dictionary 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 dictionary name.

Other names and aliases, used for rules, facts, bucketsets and so on, must begin with a letter and contain only letters, numbers, ".", "-", "_","", ":", "/", and single spaces.

Note that 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 Dictionaries

Dictionaries are created when you deploy a ruleset. The ruleset is saved along with its supporting bucketset and decision functions to a dictionary, and displayed in the **Deployments** tab of the Policy Designer interface.

Viewing Dictionaries

You view deployed dictionaries from the Policy Designer **Deployments** tab. The rulesets are listed in the **Version History** column on the left. Select one and it is displayed in the **Decision Rulesets** tab, along with it's Start Date, End Date, a checkbox indicating whether it is active, and the number of rules it contains. The bucketset and functions details are not displayed here; select the **Rules Editor** tab to view or edit them.

Modifying a Dictionary

To make changes to a dictionary, select it and Click the **Load into Rule Editor** button and it appears in the Rule Editor tab. See "Creating and Deleting Rules" and "Creating, Editing, and Deleting Bucketsets" for details on changing the details of a dictionary.

Exporting a Dictionary

To export a dictionary to a file:

1. In the **Rulesets** list, select the ruleset you want to export.
2. Click **Export**.

   A dialog appears in which you can specify the location to save the dictionary file.
3. Choose **Save File**.

To restore the dictionary, import it back into the editor.

Importing a Rule Dictionary

To import a dictionary:

1. Click the **Import** button.

2. Using the Import a Rule Dictionary File browser, select the **.rules** dictionary file to import.
3. Click **Open**.
4. In the Import a Rule Dictionary File dialog box, click **OK**.

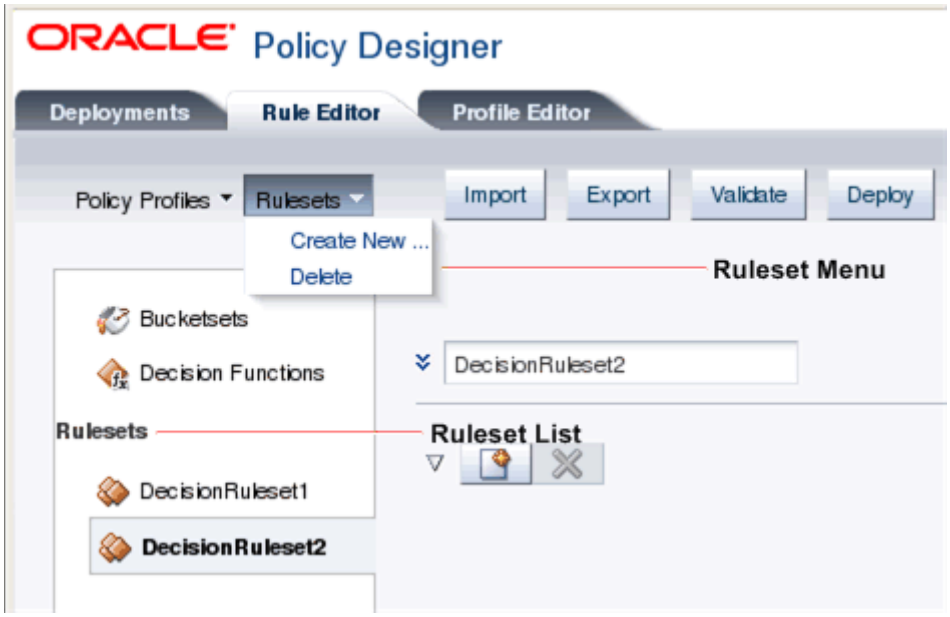
   The dictionary is imported into the editor.

Creating and Deleting Rulesets

A ruleset is a unit of execution for rules, which are fired 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. Figure 4–1 shows the **Rule Editor** subtab with the Ruleset dropdown menu displayed.

Figure 4–1 Rule Editor with DecisonRuleset2 Selected



Creating a Ruleset

To create a new ruleset:

1. Select **Create New** from the Rulesets menu.

   The new ruleset appears in the list of rulesets.
2. Select the ruleset in the Rulesets list if it is not already selected.
3. Assign a name to the ruleset by overwriting the default name in the editor.
4. If you want to configure advanced settings, toggle the advanced settings icon to the left of the ruleset name field.

   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. Check the **Active** checkbox to activate the ruleset or clear the checkbox to deactivate it. The default is Active. [Advanced Setting]

5. To save your work you have these options:

- Deploy the ruleset to a dictionary. See "Deploying Rulesets to a Dictionary" for details.
- Save the ruleset to a file. See "Exporting a Dictionary" for details.

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

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 dictionary that has already been exported or deployed. To make the deletion permanent, you must export or re-deploy the dictionary.

Setting the Effective Date for a Rule or Ruleset

The Effective Date menu is displayed only when advanced settings are shown.

To set the effective date of a rule or ruleset:

1. From the Effective Date menu, select one of the options described in Table 4–1:

Table 4–1 Effective Date Settings

Value	Description
Always	The rule or ruleset is always in effect.
Range	The rule or ruleset if in effect from the specified start date to the specified end date.
From	The rule or ruleset if in effect from the specified start date with no end date.
To	The rule or ruleset if 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.

Changing the Order of Rulesets

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

You can change the order in which the rulesets are applied by editing the **PCRF_DF** decision function.

To change the order of execution for rulesets:

1. On the left side of the rule editor, click **Decision Functions**.

   The Decision Functions list, containing a single entry for the **PCRF_DF** decision function, appears.
2. Select the row for the **PCRF_DF** decision function.
3. Click the pencil icon above the list.

   The Decision Function Editor appears.
4. To change the order of a ruleset, use the arrow to move it from the Available to the Selected list if it is not there already. The double arrow moves all the rulesets.
5. In the Selected list, select the ruleset that you want to move.
6. 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:

   
7. When you have finished reordering the rulesets, click **OK**.

Creating and Deleting Rules

A rule expresses a PCC policy decision in an IF - THEN format. 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.

You can create rules using the fields in the rule editor or you can import a previously created dictionary. See "Importing a Rule Dictionary" for information about importing a dictionary.

In the rule editor, you create rules inside a ruleset. 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.

Creating a Rule

To create a rule:

1. With the containing ruleset selected, click the new rule icon.

The rule is created.

2. Overwrite the default rule name with the name of your choice in the rule name field.
3. 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.

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 Priority menu, select the priority of this rule relative to the priority of the other rules in the ruleset. [Advanced Setting]

      Higher priority rules are fired before lower priority rules. The default priority is **medium**.
   c. 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]
   d. Check the **Rule Active** checkbox to activate the rule or clear the checkbox 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 allows 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]

4. In the IF section of the rule editor, define the rule's condition. See "Defining the Condition of a Rule" for details.
5. In the THEN section of the rule editor, define the rule's actions. See "Defining the Actions of a Rule" for details.
6. To save your work you have these options:
   - Deploy the ruleset to a dictionary. See "Deploying Rulesets to a Dictionary" for details.
   - Save the ruleset to a file. See "Exporting a Dictionary" 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 allows 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 at the end of the test row to add another test.

3. Toggle the logical operator at the end 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 checkboxes 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 checkboxes 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 checkbox 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 at the end of an existing row to create a new action row.

A new action row appears.

2. From the left menu, select the action to perform from the options described in Table 4–2:

Table 4–2 Common Rule Actions

Action	Description
Assert New	Asserts a new fact. An assert action adds a fact instance to Policy Controller memory.
Modify	Modifies a data value associated with a matched fact.
Retract	Retracts a fact. A retract action removes a fact instance from Policy Controller memory.

Note: If you are working in advanced mode, a variety of additional actions are available to you. See the discussion of how to use advanced mode actions in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for information about advanced mode actions. Only the three basic actions are covered here.

3. Do one of the following:
   - If the action is **Assert new**, follow the instructions in the "Defining an Assert New Action" section.
   - If the action is **Modify**, follow the instructions in the "Defining a Modify Action"section.
   - If the action is **Retract**, follow the instructions in the "Defining a Retract Action" section.

Defining an Assert New Action

You can assert the following facts. These facts are all outputs of the rules engine.

If you select one of the **Out**_ 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 all possible values for the new action you chose.

Out_InstallPolicy

Associates the rule with the specified policy profile. The **value** property contains the policy profile name.

Out_RemovePolicy

Disassociates the rule from the specified policy profile. The **value** property contains the policy profile name to disassociate.

Out_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.

Out_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.

Out_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.

Out_SetAbsoluteRevalidationTime
Sets a deadline, before which the PCEF should re-request an update of the PCC rules. The timer is set to an absolute date/time; for example: 2011-12-30 14-55-30 PST.

Out_SetRelativeRevalidationTime
Sets a deadline, before which the PCEF should re-request an update of the PCC 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** checkbox.
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**.

Defining a Modify Action

To define a modify action:

1. In the THEN section of the rule editor, select **Modify** in the left menu.
2. Select the fact to modify from the right menu.
3. Click the pencil icon.

   A properties form appears in which you modify the value of the fact.
4. If the value is a constant, check the **Constant** checkbox.
5. To modify 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**.

Defining a Retract Action

To define a Retract action:

1. In the THEN section of the rule editor, select **Retract** in the left menu.
2. Select the fact to retract from the right menu.

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 checkbox 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

The 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.

In the Condition Browser under the In node there are entries for two types of Gx triggers:

- In.GxEventTriggerList

  These are incoming triggers from the PCEF. The PCEF sends a list of triggers that occur in the gateway. These triggers are made available to the rules engine in the **In.GxEventTriggerList**.

  In.GxEventTriggerList contains facts based on Event-Trigger attribute-value pairs (AVP) received in a credit control request (CCR) from the PCEF. The following functions operate on these triggers: **FindGxEventTrigger** and **ContainsGxEventTrigger**.

- In.GxInstalledEventTriggerSet

  These are session triggers.

The **In.GxInstalledEventTriggerSet** fact contains values based on event triggers installed by an **Assert New (Out_AddEventTrigger)** action or uninstalled by **Assert New (Out_RemoveEventTrigger)** action in a rule created with the rule editor. The following functions operate on these triggers: **FindGxInstalledEventTrigger** and **ContainsGxInstalledEventTrigger**.

In addition to the Gx session-level triggers, there are four system-level triggers that are installed by default:

- OUT_OF_CREDIT (15)
- REALLOCATION_OF_CREDIT (16)
- REVALIDATION_TIMEOUT (17)
- SUCCESSFUL_RESOURCE_ALLOCATION(22)

You can remove these system-level triggers with **Out_RemoveEventTrigger** and re-install them with **Out_AddEventTrigger** actions.

Using the Condition Browser

The Condition Browser contains a field, a hierarchical tree view of the rules metadata, and an embedded Expression Builder. 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.

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 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 value into the field in the browser 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 nodes displayed in the browser tree reference the data listed in Table 4–3.

Table 4–3 Condition Browser Node Data

Node	Description
CurrentDate	The CurrentDate node accesses the current date and time and includes functions for referencing the following components of the current date: ■ date ■ day ■ hours ■ minutes ■ month ■ seconds ■ timezone offset ■ year

Table 4–3 (Cont.) Condition Browser Node Data

Node	Description
In	The In node contains data facts that are generated outside the rules engine and used as input to the rules engine. The In node contains many leaves and some child nodes. Facts with names having **gx** prefixes are defined in the *Policy and charging control over Gx reference point (3GPP TS 29.212 Release 9*) specification. Facts with names having **rx** prefixes are defined in the *Policy and charging control over Rx reference point (3GPP TS 29.214 Release 9*) specification. Facts pertaining to applicationType refer to whether the application is **APP_GX** or **APP_RX** Facts in the subscriberProfile child node represent subscriber data defined in the Service Broker subscriber store. See the discussion of the subscriber store data model in *Oracle Communications Service Broker Subscriber Data and Lifecycle User's Guide* for information about these values. Facts in the CalendarMsTimeZone child node are used for date and time data. CalendarMsTimezone provides the same extraction functions as CurrentDate.

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")`

Out_*	Facts with names having **OUT_** prefixes are output of the rules engine that result from rule actions. As well as being the basis for actions, these OUT facts are sometimes used as input to the rules engine to change behavior:
	
	For example, you might want to test the **Out_InstallPolicy** fact to get a subscriber's current policy profile so that you can upgrade it to a a higher level of service on the subscriber's birthday:
	
	`IF`
	`month(In.subscriber profile.dateOfBirth.time.month is CurrentDate.date.time.month) and In.subscriberProfile.dateOfBirth.time.date is CurrentDate.date.time.date)`
	`and`
	`Out_InstallPolicy.name is "GOLD"`
	`THEN`
	`ASSERT NEW Out_InstallPolicy(name: "PLATINUM")`
	
	See "Defining an Assert New Action" for information about individual OUT facts.

Using the Expression Builder

Expression Builder is used to build expressions used in tests. You access the Expression Builder from the Condition Browser by clicking the Expression Builder icon.

You can directly type an expression in the Expression 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 field.

After you have created the expression and clicked **OK**, the expression appears in the field in the Condition Browser.

The **Functions** tab of Expression Builder is especially useful for providing functions that can be used in tests. For example, to retrieve the data instance of a SubscriptionID that has the type "END_USER_SIP_URI" and the data "sip:john.doe@oracle.com", you could use the following test, in which:

- **In** is the **In** parameter in the Condition Browser
- S**ubscriptionIDType.END_USER_SIP_URI** is the SubscriptionIDType of the entry
- **sip:john.doe@oracle.com** is the SubscriptionIDData of the entry

```
IF
In.gxSubscriptionIdList.size() more than 0
and
((SubscriptionID)In.gxSubscriptionIdList.get(0)).idType is
SubscriptionIDType.END_USER_SIP_URI
and
((SubscriptionID)In.gxSubscriptionIdList.get(0)).idData is
"sip:john.doe@oracle.com"
```

but it would be simpler to use the **FindGxSubscriptionId** function to construct an equivalent test as follows:

```
IF
FindGxSubscriptionId(In, SubscriptionIDType.END_USER_SIP_URI,
"sip:john.doe@oracle.com") isn't  null
```

Similarly, to check the existence of a specific SubscriptionID, you can use the following test, which checks for a SubscriptionID of type **END_USER_SIP_URI** and the data **sip:john.doe@oracle.com**:

```
IF
In.gxSubscriptionIdList.size() more than 0
and
((SubscriptionID)In.gxSubscriptionIdList.get(0)).idType is
SubscriptionIDType.END_USER_SIP_URI
and
((SubscriptionID)In.gxSubscriptionIdList.get(0)).idData is
"sip:john.doe@oracle.com"
```

or you could construct an equivalent test the **ContainsGxSubscriptionId** function:

```
IF
ContainsGxSubscriptionId(In, SubscriptionIDType.END_USER_SIP_URI,
"sip:john.doe@oracle.com") is true
```

Expression Builder provides several of these types of find and contains functions to simplify rule creation.

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.

Creating, Editing, and Deleting Bucketsets

A bucketset is a list of values or a list of value ranges used to define a group of constant values used in rules.

> **WARNING: The default dictionary includes bucketsets for PCCProfiles and for GeographicLocationTypes. Do not delete or modify these bucketsets directly as this could cause undefined behavior in the rules engine. Instead, allow these bucketsets to be populated automatically by the Policy Controller interface.**

For example, suppose you have a set of account types that have long names. You could create an accountValues bucketset with a String data type in which the real 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 bucketsets in *Oracle Fusion Middleware User's Guide for Oracle Business Rules* for more information.

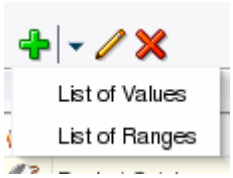
Creating a Bucketset

To create a bucketset:

1. In the left panel, click **Bucketsets**.

   The Bucketsets list appears.

2. Click the Add Bucketset menu and select whether the bucketset will define a range or a list of values (LOV).



   The new bucketset appears in the list with a default name.

3. In the bucketset list, select the bucketset that you just created.
4. Click the pencil icon.

   The Bucketsets Editor appears.

5. In the Name field, overwrite the default name with the name you want to assign to the bucketset. Do not use the same name for the bucketset 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 bucketset.
7. From the Data Type menu, select the data type of the values in the bucketset.

   All the values in a bucketset must be of the same type.

8. If you want to allow invalid values in tests, check the **Include Disallowed Buckets in Tests** checkbox. This allows you to test for invalid values.
9. Do one of the following:

   If the bucketset 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 actions, check the **Allowed in Actions** box. Otherwise clear it.

      For more information, see the discussion of the bucketsets 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 bucketset.
   g. Click **OK**.

   If the bucketset 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** checkbox 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 range is allowed in actions, check the **Allowed in Actions** box. Otherwise clear it.

   For more information, see the discussion of the bucketsets 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 bucketset.

10. Click **OK**.

Editing a Bucketset

To edit a bucketset:

1. In the left panel, click **Bucketset**s.

   The Bucketsets list appears.

2. Select the bucketset that you want to edit.

3. Click the pencil icon.

   The Bucketset Editor appears.

4. Make your changes in the Bucketset Editor following the guidance for creating a new bucketset.

5. Click **OK**.

Deleting an Item in a Bucketset

To delete an item in a bucketset:

1. In the Bucketsets Editor, select the item in the list that you want to delete.

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

Deleting a Bucketset

To delete a bucketset:

1. In the Bucketsets list, select the bucketset that you want to delete.

2. Click the delete icon above the Bucketsets list.

Deploying Rulesets to a Dictionary

Deploying a rules dictionary activates the rulesets (using its supporting bucketsets and decision functions) in your Policy Controller implementation. To check whether a rules dictionary has been be deployed, click the **Deployments** tab at the top of the Policy Designer interface to see whether it appears in the list of **Decision Rulesets** tab.

To deploy a rules dictionary:

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 dictionary 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 dictionary is loaded.

3. Click **Deploy** in the dialog box to deploy the dictionary.

Future updates to the dictionary do not affect a deployed dictionary. To include updates in the release, you must re-deploy the dictionary.

Example Rules

Because of the flexibility and extensibility of the PCRF rules engine, the possibilities for creating rules are infinite. Following are examples of a few of the most common scenarios.

Using Subscriber Data to Change a Policy Profile

A common rule task is to install a policy 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 policy profile on the subscriber's birthday:

```
IF
In.subscriberProfile.dateOfBirth.time.month is CurrentDate.date.time.month
AND
In.subscriberProfile.dateOfBirth.time.date is CurrentDate.date.time.date
THEN
ASSERT NEW Out_InstallPolicy(name: "BIRTHDAY")
```

Applying a New Service to an Existing Service

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

```
IF
In.subscriberProfile.accountType is AccountType.Prepaid
AND
MatchToAddress(In.gxTftPacketFilterInformationList, "69.63.189.*") is true
AND
In.applicationType is ApplicationType.APP_GX
THEN
Assert New Out_InstallPolicy(name: "Prepaid_SocialVoice_Plan")
```

Using PCEF Triggers to Change a Policy Profile

Receipt of a trigger from the PCEF can be used to initiate a change in a policy profile.

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

```
IF
FindGxEventTrigger(In, EventTrigger.OUT_OF_CREDIT) is true
AND
In.subscriberProfile.subscriberCategory is "GOLD"
OR
In.subscriberProfile.subscriberCategory is "SILVER"
OR
In.subscriberProfile.subscriberCategory is "BRONZE"
THEN
ASSERT NEW Out_InstallPolicy(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** Policy Profile is applied which contains throttled back service.

This rule applies a **NoCredit_Plan** Policy Profile when an **OUT_OF_CREDIT** event is received:

```
IF
In.gxEventTriggerList isn't null
AND
In.gxEeventTriggerList contains (EventTrigger.OUT_OF_CREDIT )
AND
In.getRATType() is RATType.EVOLUTION
THEN
Assert New Out_InstallPolicy(name: "NoCredit_Plan")
Assert New Out_AddEventTriger(EventTrigger.REALLOCATION_OF_CREDIT )
```

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

```
In.gxEeventTriggerList contains(In, EvetnTrigger.REALLOCATION_OF_CREDIT
AND
IN.gxRatType is RATType.EVOLUTION
THEN
Assert New Out_RemovePolicy(name:"NoCreditPlan")
Assert New Out_AddEventTrigger(EventTrigger:EventTrigger.OUT_OF_CREDIT)
```

Using a Local Fact to Apply a Policy 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 policy 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** Policy Profile is applied. The following rules accommodate this scenario:

```
IF In != null
THEN
assert new LocalFact(integerValue:0, name: "numberOfRules")

IF In.subscriberProfile.birthdate.get(Calendar.YEAR) >
(CurrentDate.get(Calendar.YEAR - 25)
   AND LocalFact.name == "numberOfRules"
assert new InstallPolicy(name: "RULE_1")
modify( LocalFact (integerValue: LocalFact.integerValue + 1)

IF In.subscriberProfile.subscriberCategory == "SILVER"
   AND LocalFact.name == "numberOfRules"
assert new InstallPolicy(name: "RULE_2")
modify( LocalFact (integerValue: LocalFact.integerValue + 1)

IF In.ipcantype == "WIFI"
   AND LocalFact.name == "numberOfRules"
assert new InstallPolicy(name: "RULE_3")
modify( LocalFact ( integerValue: LocalFact.integerValue + 1)

IF LocalFact.integerValue == 2
assert new InstallPolicy(name: "SPECIAL_RULE")
```

5

Policy Controller Protocol Reference

This chapter lists Policy Controller reference material.

Diameter Rx Command Codes Supported by Policy Controller

Table 5–1 lists the Diameter Rx messages that Policy Controller uses to communicate with the Application Functions.

Table 5–1 Diameter Rx Messages Supported

Message	Command Code	Application Function Initiates?	Policy Controller Initiates?
AA- Request (AAR)	265, 'R' bit set	Yes	No
AA-Answer (AAA)	265, 'R' bit cleared	No	Yes

Gx Command Codes Supported by Policy Controller

Table 5–2 lists the Attribute Value Pairs (AVPs) used by Policy Controller for PCEF-AF traffic.

Application-ID: 16777238

Policy Controller acts as a Diameter Server; PCEF is client which requests PCC rules.

Table 5–2 Supported GX Messages

Message	Command Code	PCEF Initiates?	Policy Controller Initiates?
Credit-Control-Request (CCR)	272	Yes	No
Credit-Control-Answer (CCA)	272	No	Yes
Re-Auth-Request (RAR)	258	No	Yes
Re-Auth-Answer (RAA)	258	Yes	No

Diameter Gx AVPs Supported by Policy Controller

Table 5–3 lists the Diameter Gx protocol AVP supported by Policy Controller.

Table 5–3 Supported Gx Protocol AVPs

Attribute Name	AVP Code	Data Type
Charging-Rule-Install	1001	Grouped
Charging-Rule-Remove	1002	Grouped
Charging-Rule-Definition	1003	Grouped
Charging-Rule-Base-Name	1004	UTF8String
Charging-Rule-Name	1005	OctetString
Charging-Rule-Report	1018	Grouped
Event-Trigger	1006	Enumerated
IP-CAN-Type	1027	Enumerated
Guaranteed-Bitrate-DL	1025	Unsigned32
Guaranteed-Bitrate-UL	1026	Unsigned32
Metering-Method	1007	Enumerated
Offline	1008	Enumerated
Online	1009	Enumerated
Packet-Filter-Content	1059	IPFilterRule
Packet-Filter-Information	1061	Grouped
Precedence	1010	Unsigned32‘
PCC-Rule-Status	1019	Enumerated
Session-Release-Cause	1045	Enumerated
QoS-Class-Identifier	1028	Enumerated
QoS-Information	1016	Grouped
Rule-Failure-code	1031	Enumerated
TFT-Filter	1012	IPFilterRule
TFT-Packet-Filter-Information	1013	Grouped
ToS-Traffic-Class	1014	OctetString
RAT-Type	1032	Enumerated
Revalidation-Time	1042	Time

Re-Used Diameter Gx AVPs Supported by Policy Controller

Policy Controller supports these re-used Diameter Gx protocol AVPs. See the 3GPP TS 29.212 specification for details on individual AVPs.

- 3GPP-RAT-Type
- 3GPP-SGSN-Address
- 3GPP-SGSN-MCC-MNC
- 3GPP-User-Location-Info
- Called-Station-ID
- CC-Request-Number
- CC-Request-Type

- Charging-Information
- Flow-Description
- Flows
- Flow-Status
- Framed-IP-Address
- Max-Requested-Bandwidth-UL
- Max-Requested-Bandwidth-DL
- Rating-Group
- Service-Identifier
- Subscription-Id
- User-Equipment-Info
- 3GPP-MS-TimeZone

Diameter Rx AVPs Supported by Policy Controller

Table 5–4 lists the Diameter Rx protocol AVPs supported by Policy Controller. Information in other AVPs is ignored.

Table 5–4 Supported Rx Protocol AVPs

Attribute Name	AVP Code	Data Type
AF-Application -Identifier	504	OctetString
Flow-Description	507	IPFilterRule
Flows	509	Unsighed32
Flow-Status	511	Enumerated
Flow-Usage	512	Enumerated
Media-Component-Description	517	Grouped
Media-Component-Number	518	Unsigned32
Media-Sub-ComponentAVP	519	Grouped
Media-Type	520	Enumerated
Experimental-Result-Code	298	Unsigned32

Subscriber Profile Data Available to Policy Controller

Table 5–5 lists the Subscriber Store data stored for Policy Controller and available to use in a PCC Rule. For details on Subscriber Store, see *Subscriber Store User's Guide*.

Table 5–5 Subscriber Profile Data for Policy Controller

Field	Multciplicity	Level	Descriptions/Format	Data Type
Subscriber-Id	1..*	N/A	Subscriber ID value.	String
Subscriber-Id-Type	1..*	N/A	The subscriber ID type.	Integer
AccountState	1	Global	The account state value.	String

Table 5–5 (Cont.) Subscriber Profile Data for Policy Controller

Field	Multciplicity	Level	Descriptions/Format	Data Type
AccountType	1	Global	Prepaid, postpaid, Hybrid	String
SubscriberActivationDate	1	Global	A calendar entry in the form: *YYYYMMDD*	Calendar
DateOfBirth	1	Global	A calendar entry in the form: *YYYYMMDD*	Calendar
SubscriberCategory	1	PCRF	A service level, for example: gold, silver, bronze.	String
HomeZone	*	PCRF	The HomeZone value	HomeZone