



Services

- [Overview, on page 1](#)
- [Service Example, on page 5](#)
- [Service Screens, on page 6](#)
- [Custom Reference Data Tables, on page 24](#)

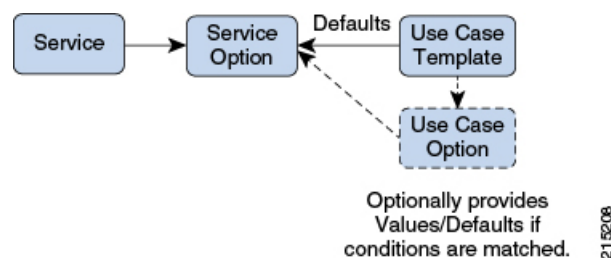
Overview

In CPS, a 'Service' is what is assigned to a subscriber (in USuM) to define how that subscriber is treated. Some basic examples of services would be a 'GOLD' user might get a high upload/download speed whereas a 'BRONZE' user would get a low one. Other examples would include having one type of user be redirected to a portal when their Quota is exhausted whereas another type would only have their speed downgraded.

As the Service maps as closely as possible to how a Service Provider wants to classify their customers, the Service in CPS is flexibly defined to allow configuration at different levels.

Below is an overview of the different objects referenced in the Services tab in PB. The detailed description of each object is provided in below sections.

Figure 1: Services



Service

- A service is effectively just a 'code' to label the service and a collection of Service Options which contain the definition of what a service 'is'.
- What a Customer Service Representative assigns to a subscriber to describe the user's plan.
- Multiple services can be assigned to a single subscriber

- If multiple services are assigned to a subscriber, the service options are combined between all assigned services.

Therefore, there is no logical difference between a subscriber with:

- A single service with 10 service options
- 10 services with 1 option each

Service Option

- Provides the concrete values which can be re-used for multiple services.

For example, one subscriber might have one service option which describes the values for 10 MB Upload/Download speed and another subscriber which describes 1 MB Upload/Download speed. Continuing the example from above, 10 MB could be assigned to a GOLD service and 1 MB could be assigned to BRONZE.

- What values are configurable in a Service Option are setup by the Use Case Template object. The Use Case Template can provide defaults to the Service Option or hide values in Service Configuration objects not necessary for certain use cases.
- If a Service Configuration's value is not defined in a Service Option, the value from the Use Case Template is used.

Service Configuration

- The low-level configuration objects used by the CPS code to drive functionality. These objects are used to drive functionality in the system. The whole point of the Service > Service Option > Use Case Template chain of functionality is to flexibly configure these Service Configuration objects which the code uses to drive system logic.
- These objects are defined by the CPS code.

Types of service configurations:

- **PriorityConfiguration:** Only one allowed to be active at a time. If multiples priority configurations are added, highest priority is used.

These are used in cases where only a single value makes sense. For example, when sending an 'Accept' message, we can only have one template and multiples do not make sense.

Objects of this type always have a priority field. If multiple priority configurations are added, the highest priority object is used.

Example: AccessAcceptConfiguration, RegisterMacAddress

- **GroupConfiguration (most common):** Only 1 per 'Group Name' are allowed to be active. If multiple configurations are added highest priority per 'Group Name' is used.

These are used in cases where a configuration only makes sense for a single 'group' (key). For example, if it makes sense to control the upload/download speed based on the network type (cell, Wi-Fi, and so on) a service configuration to control network speed with a group set for cell/Wi-Fi would allow multiple service configurations to be added.



Note RADIUS-based policy control is no longer supported in CPS 14.0.0 and later releases as 3GPP Gx Diameter interface has become the industry-standard policy control interface.

These objects always have a group field as well as a priority field. For each unique group value, the highest priority is used.

Example: IsgServiceConfiguration, All Diameter Configurations, OneTimeUsageCharge

- ServiceConfiguration: Multiples allowed. If multiple configurations are added, all are used. 'Modify' functionality in PB for Use Case Options/Service Options can override values conditionally.

Example: AutoChargeUpAccounts, AutoProvisionQuota, BalanceRateConfiguration

Use Case Templates

Use case templates are the building blocks of the Cisco Policy Builder Service Model architecture.

- Defines the Service Configuration objects to be set by a Service Option.
- Provide default values and/or hide values which do not need to be set by a use case
- Optionally, contains 'Initiators' (Conditions) which define when the template is active.
- Created by an advanced user (usually Engineering/AS).
- Makes Service Option and Service creation easier.

For example, a Use Case Template set up to create different upload or download speeds includes a 'DefaultBearer' QoS Service Configuration object. The user creating a Use Case Template can default and/or hide the values for 'ARP' and other values not directly related to upload or download speed. This allows the creation of the Service Option to be much simpler.

A copy of the Use Case Options is created while copying a Use Case Template.

Use Case Template tab Order

A new parameter `-DshowUseCaseInitiatorTabFirst` can be added in `pb.conf (/etc/broadhop/pb/)` file on perfcient01/02 to re-order the **Use Case Template** and **Use Case Option** tabs. This parameter also renames **Use Case Template** and **Use Case Option** tabs to **Actions** tab.

By default, `-DshowUseCaseInitiatorTabFirst` is set to `true` (does not required to be added in `pb.conf` file by default).

- If set to `true`, the tabs will be displayed in the following order:

Figure 2: Use Case Initiators > Actions > Documentation

Figure 3: Use Case Initiators > Actions > Documentation

- For backward compatibility, the configuration parameter `-DshowUseCaseInitiatorTabFirst` in `pb.conf (/etc/broadhop/pb/)` file on `pcrfclient01/02` can be set to `false`.

This parameter also renames **Actions** tab back to **Use Case Template** and **Use Case Option** tabs.

Use Case Template and **Use Case Option** tabs will be displayed in the following order:

Figure 4: Use Case Template > Use Case Initiators > Documentation

Figure 5: Use Case Option > Use Case Initiators > Documentation



Use Case Option

- A child of Use Case Template used to add or modify Service Configurations objects when certain conditions occur.
- Provides a way to separate Service Configurations within a use case based on conditions.
- Contains the same functionality of a Use Case Template.
- Adds or modifies new service options from parent Use Case Template.

While copying a Use Case Option, all the corresponding children Use Case Options get copied as well.

For example, if a user's upload or download speed should be decreased when they are out of quota, a **Use Case Option** is added with a condition indicating the user is out of quota. The service configurations in the use case options can have a higher priority than those in the use case template to override the normal values. The service option then allows setting both the normal upload or download speed and the upload or download speed when the user is out of quota.

Service Example

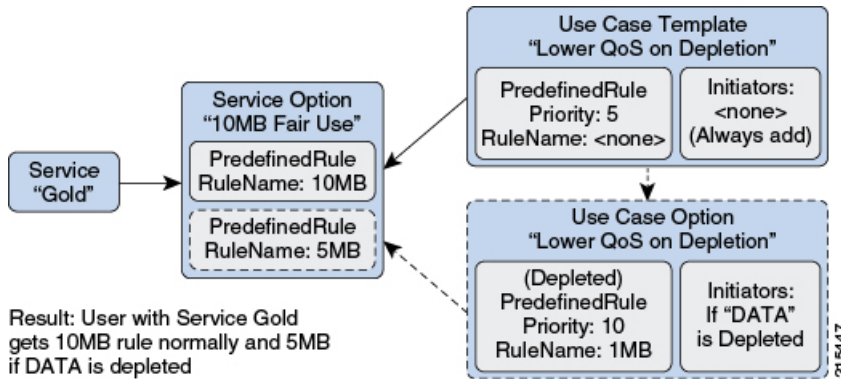
The following diagram illustrates how a 'Service Configuration' object called PredefinedRule can be flexibly configured as part of a service. For those unfamiliar, a predefined rule is just an identifier sent to PCEF which controls a users upload or download speed (QoS) among other things.

In this case, the service gold is assigned a Service Option called 10MB Fair Use. This service option results in a rule being passed to PCEF called 10MB by default and switching the rule to 5 MB when the users quota is depleted.

Notice that it would be easy to add another service option for 20MB Fair use, and so on.

The Use case template defines the low-level information that a 'PredefinedRule' be created of priority five. This rule is default and always present. Additionally, a Use Case Option defaults that another PredefinedRule of priority ten be added. The higher priority results in the new rule name being switched when quota is depleted.

Figure 6: Service Using Use Case Option

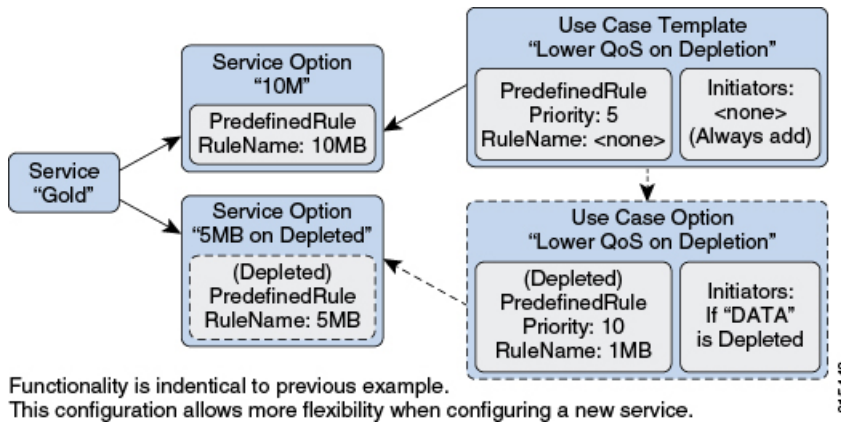


Instead of using a Use Case Option to define what happens when a user is depleted, they set up another Use Case Template.

This could have an advantage if a customer wanted services for a large combination of values (10 MB default with any combination of 1 MB - 9 MB depleted speed). Also, to support a use case where the end user could be independently assigned a default speed and a depleted or downgrade speed. A service for 10 MB default and 5 MB depleted would also be functionally equivalent.

This flexibility in the service model allows mapping CPS closely to the Service Providers concept of a service.

Figure 7: Service Using Use Case Template



Service Screens

The following screens set up the example seen in the previous section. Service 'Gold' with two separate service options. The 'Predefined Rule' that is part of the setup is just an example, so not all fields on the Predefined Rule are described.

For example, a 'Rule Name' is a label and can be assigned to a subscriber on a PCEF. PCEF is responsible for defining what the rule 'means'. In this example, we make the assumption that PCEF has rules set up for '10_MB' and '5_MB' which control the users QoS (Quality of Service - effectively upload or download speed) to 10 MB upload or download and 5 MB upload or download respectively.



Note The subtle difference between 10 MB and '10_MB' is to ensure that it is clear which name is used for internal reference and which is sent to PCEF.

Services

Figure 8: Service Page

The following parameters can be configured under Service:

Table 1: Service Parameters

Parameter	Description
Code	This is what is saved on the subscriber. It defaults to 'default'. This value is the link between the Services assigned a subscriber in Control Center and the Service in Policy Builder. This value is not updated if existing users have it assigned, as the 'link' is broken. Instead, the name is updated.
Name	This is the name displayed in Control Center.
Enabled	If this value is unchecked, the service is not evaluated by the Policy Engine and is not displayed in Control Center. Default value is checked (true).

Parameter	Description
Suppress in Portal	If this value is checked, this Service is not displayed in the Portal. This is specific for SP Wi-Fi call flows. Default value is unchecked (false). Note RADIUS-based policy control is no longer supported in CPS 14.0.0 and later releases as 3GPP Gx Diameter interface has become the industry-standard policy control interface.
Balance Service	If this value is checked, the Service runs through balance processing. This results in one database read or write against the balance database. Performance improves (due to fewer database read or writes). For the services which don't rely on Balance or Quota, this value is unchecked. Default value is checked (true).
Add to Sub Accounts	If this value is checked, this service is considered to be also assigned to any 'subaccounts' associated to the main subscriber. This is useful if you had a family plan where the QoS is controlled at the main subscriber level and 'trickles down' to the family sub accounts rather than being set at each individual level. Default value is unchecked (false).

Service Options Table

This table displays a read-only version of the service options associated to the Service. Service Options can be added or removed to the service.

Table 2: Service Options Table Parameters

Parameter	Description
Name	The name of the associated Service Option.
Use Case Template	The name of the associated use case template.
Add	Allows adding another Service Option to this service. The Add dialog shows the Use Case Template > Service Option hierarchy on the left and a preview of the parameters set by the Service Option or Use Case Template on the right.
Remove	Removes a Service Option from the Service.
Up or Down Arrow	Allows moving a Service Option up or down. This only affects the ordering of service options in the list and does not functionally affect the resolution of services.

View Service Option Parameters

This hyperlink shows a consolidated list of the Service Configuration Parameters from the Use Case Template and Service Options. This view is useful for simple service options or use case templates. However, since it

does not show the distinction between different Service Configuration objects. Hence, it can be difficult to read for more complicated configurations.

Service Options

The service option dialog allows setting the concrete values for service configuration parameters used in the Use Case Template. The groups under service options are Use Case Templates. Adding a new Use Case Template adds a new group under Service Options so you can provide concrete values for the Use Case Templates.

The following parameters can be configured under Service Option:

Table 3: Service Option Parameters

Parameter	Description
Name	This is the name of the service option which is referenced by the Service.
Use Case Template	This is a link that allows quickly seeing the associated Use Case Template and going straight to this Use Case Template. Note You can use the 'left' arrow in the Policy Builder toolbar to go 'back' to the service option after clicking into the Use Case Template.
Service Configurations	This is a list of the 'Service Configuration' objects that are to be set as part of the Service Option. This list can usually be used 'as is' upon creation. The Service Configuration objects from the Use case template is used as a default and any values set here 'overrides' the use case template. • Add: This allows adding a new Service Configuration that has been added to the Use Case Template. This is only really useful when a new Service Configuration object was added to a Use Case Template after this service option was created and you now what to override some values on this new object. • Remove: This is only really useful if you don't want to use your configured values at all and just use the values from the Use Case Template. In that case you can just remove the Service Configuration Objects and the values from the Use Case Template is used.
PreDefinedRule Parameters	These are the parameters being overridden from the use case template.
Add	This allows adding a parameter from the Use Case Template even if it is not marked as 'Allow Override'. It also allows customizing a parameter that didn't exist previously in the Use Case Template or was removed from the Service Option.
Remove	This allows removing a parameter from the Service Option. This means that the value specified for the Use Case Template's version of this parameter is used.

Parameter	Description
Display Name	This column shows the 'Display Name' of the parameter. This name can be updated by either the Service Option or the Use Case Template to make it clearer for users to understand what this value does. The 'true' name of the field can be seen by hovering over the cell.
Value	This is the value of the parameter which should be set. If there is a value set in the 'Pull Value From...' column, this value is used as the default and overridden with the 'Pull Value From...' information.
Pull Value From...	This allows setting this value dynamically through AVP's, Custom Reference Data or the 'Policy State'. For more information, refer to Table 4: Pull Value From..., on page 10 .

Table 4: Pull Value From...

Parameter	Description
Subscriber AVP Code	This allows pulling values from AVPs on the subscriber. This field now also supports AVP's on the subscriber's session and 'Policy Derived AVP's added in policies.
Custom Reference Data Column	This allows pulling the value from the Custom Reference Data table's column specified. Note Custom Reference Data takes care of which 'row' in the column is matched, so this ends up specifying a single value.
Bind to Session/Policy State	This allows pulling the value from the state of the system. This uses any of the preconfigured 'Policy State Data Retrievers' that are plug-in code that know how to get a certain value from the system.
Dynamic Reference Data Key	This allows pulling the value from other reference data configuration (Policy Builder or CRD, for example, Account Balance Templates) as value for the use case attribute. Currently, only Account Balance Template type attributes are supported. The intended Account Balance Template code can be configured in the text field. Both Policy Builder and CRD Balance templates can be pulled using this field. Policy Builder templates are checked first, if not found then CRD templates are searched.

Use Case Templates

Use case templates form the basis of a service and are in general more complicated than creating a service or service option. For more information on creating Use Case Templates, contact your Cisco Technical Representative.

The following parameters can be configured under **Actions** (Use Case Template) tab:

Table 5: Actions (Use Case Template) Parameters

Parameter	Description
Name	This is the name used to identify the Use Case Template. It can be changed.
Service Configurations	This is where you add the Service Configuration objects needed to configure your use case. • Add: The Add button brings up the full list of Service Configuration objects. Which to choose should be based on the use case documentation you are implementing. The right column of the screen displays the parameters that are in a specific service configuration. If you 'uncheck' a parameter, the default as seen on this screen (from the code) will be utilized. Unchecking is functionally equivalent to 'removing' a field from the table after adding it. • Add New: Implies that you are adding a new Service Configuration to be configured. This is almost always what you want. • Modify: Implies that you are modifying an existing Service Configuration. This is used in a 'Use Case Option' and in general not recommended as it's preferred to use the 'priority' fields to update the service configuration objects.
Create Child: Use Case Option	This allows creating a child use case option of this use case template.
PreDefinedRule Parameters	
Add	This allows adding a parameter to the Use Case Template even if it wasn't added initially (unchecked or a code update added a new parameter).
Remove	This allows removing a parameter from the Use Case Template. This means that the value specified in the code (as seen on the Add Service Configuration screen) will be utilized.
Display Name	This column shows the 'Display Name' of the parameter. This name can be updated by either in the Use Case Template to make it clearer for users to understand what this value does. The 'true' name of the field can be seen by hovering over the cell.
Value	This is the value of the parameter which should be set. If there is also a value set in the 'Pull Value From...' column, this value will be used as the default and overridden with the 'Pull Value From...' information.
Pull Value From...	This allows setting this value dynamically through AVP's, Custom Reference Data or the 'Policy State'.
Allow Override	This value indicates whether this option will show up for configuration in the Service Option by default. In general, it is best to only check this box for values you expect to be configured in the Service Option. This will make it easier to configure and maintain the service options.

Use Case Initiators

Use Case Initiators are groups of conditions that indicate whether or not the Service Configuration objects within this use case template are used. If no use case initiators are specified, the Service Configuration objects will always be added.

The following parameters can be configured under the **Use Case Initiators** tab.

Table 6: Use Case Initiators Parameters

Parameter	Description
Service Initiators (OR Together)	Service Initiators are a grouping of conditions. If ANY of the service initiators on a Use Case Template are true then that Use Case template is active and the Service Configurations are used. When you add multiple Service Initiators, the Use Case Template is activated and Service Configurations are used when ANY one of these initiators are true, as indicated by the caption “(OR Together)”. • Plus/X: Lets the user add or remove a service initiator. • Up/Down arrow: The user can move the initiators up and down. This affects the order in which the service initiators are evaluated.
Initiator Name	This is a logical name for this grouping of conditions.

Parameter	Description
Conditions (AND Together)	These are the conditions associated with the service initiator. Conditions usually pertain to this messages session, subscriber information, balance information, or the message itself. The current values of the condition are set up by the code/messaging. When you add multiple conditions, ALL of the listed conditions must be true, as indicated by the caption “(AND Together)”. • Add/Remove: Lets the user add or remove a condition. • Up/Down arrow: Lets the user move a condition up or down in the list. This is usually only needed when referencing a previous condition in this condition. For example, the user can check a 'session exists' in the first condition and then use the 'session user name' in the second condition. The user can only reference previous conditions. • Input Variables (AND Together): These input variables are specific to each condition. If the current value of the condition matches the input variable, this condition is true. When you define multiple input variables for a condition, ALL of the variables are evaluated and must be true for the condition to be true, as indicated by the caption "(AND Together)". • Input Variable: This is the name of the variable. • Type: This is either 'Literal' matching where we match against a hard-coded 'value' or 'Output' where the value is pulled from a previous (or current) condition. If output is chosen, a popup will appear where the values from the current or previous conditions can be selected and the 'value' box is replaced with the selection. • Operator: This allows editing how 'true' is determined given the current value of the condition and the value specified. The default and by far most common option is '=' equals.

Parameter	Description
Operator	= (Equals): The condition is true if the values are equal. This uses the Java equals() function. Multiple value with comma separated is not allowed with = operator for string datatype.
	<> (Not Equals): The condition is true if the values are not equal.
	isNull: The condition is true if the value is null (empty).
	!isNull: The condition is true if the value is not null (empty).
	Matches: The condition is true and the value is a regex that successfully matches the current value. Java regex is utilized. Example: .* matches anything.
	[0-9]+: Matches 1 or more numbers.
	^[0-9]*\$: Matches a string which is only numbers or empty.
	!matches: The logical opposite of matches (if matches would be true, !matches is false).
	In: This is true if the current value is equal to one of the values listed (comma separated).
	!In: This is the logical opposite of In (if In would be true; !In is false).
	Contains (Only for lists): True if the list contains the value or values.
	!Contains (Only for list): The logical opposite of contains.
	Value: This is the hard-coded value which should be compared against the current value in the system.
	Remove (link): This allows removing this input variable.

The following sections describe the input variables that can be configured for the following conditions:

A Customer Reference Data AVP Exists

This condition indicates that a custom reference data table exists in the Policy Builder configuration with one or more Column Names mapped to a unique Attribute Value pair. Any column marked as Use in conditions can be used.

In order to protect a customer reference column's integrity, once it has been created and published, it cannot be changed.



Note Under Use Case initiator tab for **A customer reference data AVP exists** condition, if you configure a regex expression that is a combination of multiple conditions, then specify each condition as a separate entry with AND together.

For example, instead of using the complex regex like “(?=MCC.*)(?!MCC1904)”, use two “A customer reference data AVP exists” condition (AND together). Configure one reference data with MCC.* and other reference data should not be equal to (<> as operator) MCC1904.

The following condition input variables can be exposed in the Policy Builder GUI:

Table 7: Available Input Variables used for A Customer Reference Data AVP Exists

Condition Input Variable	AVP Used/Description
tableName	The custom Reference Table Name.
code	The custom AVP column Name.
value	The custom AVP column's value.

A Customer Reference Data AVP Does Not Exist

This condition indicates that a custom reference data table exists in the Policy Builder configuration with one or more Column Names not mapped to a unique Attribute Value pair. Any column, which is not marked as Use in conditions cannot be used.

In order to protect a customer reference column's integrity, once it has been created and published, it cannot be changed.

The input variables that can be exposed for this condition in the Policy Builder GUI are the same as described in [A Customer Reference Data AVP Exists, on page 14](#).

A Diameter Gx TGPP Session Exists

This condition indicates that a valid Diameter Gx 3GPP session exists in the Policy Builder configuration for a given subscriber. The following condition input variables can be exposed in the Policy Builder GUI:

Table 8: Available Input Variables for Diameter Gx TGPP Session Exists

Condition Input Variable	AVP Used/Description
msisdn	The subscriber identity in the Subscription-Id AVP being msisdn.
imsi	The subscriber identity in the Subscription-Id AVP being imsi.
framedIpv6	The Ipv6 prefix allocated for the user.
framedIp	The Ipv4 address allocated for the user.
sessionId	The unique Id of the session generated by PCEF
commandCode	The unique Command Code to identify the Gx Request/Response Type – CCR/CCA or RAR/RAA.

Condition Input Variable	AVP Used/Description
requestType	The type of the request (initial, update, termination).
destRealm	The destination realm derived from the Origin-Realm of CCR-I and used in PCRF RAR request.
destHost	The destination host derived from the Origin-Host of CCR-I and used in PCRF RAR request.
ratType	Identifies the radio access technology that is serving the UE. PCRF supports devices using narrow band Internet of Things (NB-IoT) RAT that is a 3GPP radio interface to support IoT devices. PCRF can create a session with UE having RAT-type as NB-IOT and provides all functionalities (such as policy control and charging rule functionality) to an NB-IOT devices.
appId	The Unique Application Identifier to identify the Gx Session.
mccmnc	Concatenated value of Mobile Country Code (mcc) and Mobile Network Code.
calledStationId	The address the user is connected to. For GPRS the APN.
lac	Extracted from either the 3GPP-User-Location-Info AVP, which provides details of where the UE is currently located, or the RAI AVP, which contains the Routing Area Identity of the SGSN where the UE is registered.
rac	Extracted from either the 3GPP-User-Location-Info AVP, which provides details of where the UE is currently located, or the RAI AVP, which contains the Routing Area Identity of the SGSN where the UE is registered.
sac	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
ci	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
tgppRatType	3GPP-RAT-Type AVP. Indicates the Radio Access Technology that is currently serving the UE.
eventTriggers	NA
outofCredit	An OUT_OF_CREDIT trigger reported by Event-Trigger AVP.
cgi	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
Ecgi	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
Tai	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.

Condition Input Variable	AVP Used/Description
Sai	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
Tac	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
Eci	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
Imeisv	International Mobile Equipment Identifier along with Software Version. Extracted from User-Equipment-Info-Value AVP.
networkRequestSupport	Network-Request-Support AVP. Indicates whether or not the UE and access network supports the network requested bearer control mode.
Nai	The Network Access Identifier.
bearerControlMode	Bearer-Control-Mode AVP. Indicates the PCRF selected bearer control mode.
sgsnIpAddress	3GPP-SGSN-Address AVP. Indicates the Ipv4 address of the SGSN.
ipcanType	IP-CAN-Type AVP. Indicates the type of Connectivity Access Network in which the user is connected.
Mnc	Extracted from either the 3GPP-User-Location-Info AVP, which provides details of where the UE is currently located or the RAI AVP, which Contains the Routing Area Identity of the SGSN where the UE is registered.
Mcc	Extracted from either the 3GPP-User-Location-Info AVP, which provides details of where the UE is currently located, or the RAI AVP, which contains the Routing Area Identity of the SGSN where the UE is registered.
Rai	Extracted from either the 3GPP-User-Location-Info AVP, which provides details of where the UE is currently located, or the RAI AVP, which contains the Routing Area Identity of the SGSN where the UE is registered.
qosUpgrade	Qos-Upgrade AVP. Indicates whether the SGSN Supports the QOS Upgrade or not.
sgsnIpv6Address	3GPP-SGSN-IPv6-Address AVP. Indicates the Ipv6 address of the SGSN.
targetApn	Called-Station-Id AVP. The same as called Station Id condition.
globalCiCode	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
globalAreaCode	Extracted from 3GPP-User-Location-Info AVP. Provides details of where the UE is currently located.
imeiTac	Same as Imeisv extracted from User-Equipment-Info-Value AVP.
Mac	Extracted from User-Equipment-Info-Value AVP. The identification and capabilities of the terminal.

Condition Input Variable	AVP Used/Description
Eui64	Extracted from User-Equipment-Info-Value AVP. The identification and capabilities of the terminal.
modifiedEui64	Extracted from User-Equipment-Info-Value AVP. The identification and capabilities of the terminal.
qosNegotiation	Qos-Negotiation AVP. Indicates if the PCRF is allowed to negotiate the QoS.
Offline	Offline AVP. Defines whether the offline-charging interface from the TDF shall be enabled.
Online	Online AVP. Defines whether the offline charging interface from the TDF shall be enabled.
Ipv4AnGWAddress	IPv4 Gateway Address.
Ipv6AnGWAddress	IPv6 Gateway Address.
mpsEpsPriority	Value of the mspEpsPriority set.
tgppMsTimeZone	3GPP-MS-TimeZone AVP. Indicates the offset between universal time and local time in steps of 15 minutes of where the MS currently resides.
accessNetworkChargingAddress	Access-Network-Charging-Address AVP. Indicates the IP Address of the network entity within the access networks performing charging (e.g. the GGSN IP address).
usageReportET	NA
Meid	Extracted from User-Equipment-Info-Value AVP. The identification and capabilities of the terminal.
Bsid	Base Station Identifier.
chargingCharacteristics	3GPP-Charging-Characteristics AVP. The Charging Characteristics applied to the IP-CAN session.
chargingCharacteristicsBits	Extracted from 3GPP-Charging-Characteristics AVP. The Charging Characteristics applied to the IP-CAN session
appName	Application Name.

A Diameter Gy v8 TGPP Session Exists

This condition indicates that a valid Diameter Gy v8 3GPP session exists in the Policy Builder configuration for a given subscriber. The following condition input variables can be exposed in the Policy Builder GUI:

Table 9: Available Input Variables for Diameter Gy v8 TGPP Session Exists

Condition Input Variables	AVP Used/Description
Msisdn	Subscription-Id-Data AVP. The identification of the subscription (IMSI, MSISDN, etc.).

Condition Input Variables	AVP Used/Description
sessionId	The unique Id of the session generated by PCEF.
requestType	CC-Request-Type AVP. The type of the request (initial, update, termination).
requestNumber	CC-Request-Number AVP. The number of the request for mapping requests and answers.
destRealm	Destination-Realm AVP. The destination realm derived from the Origin-Realm of CCR-I and used in PCRF RAR request.
destHost	Destination-Host AVP. The destination host derived from the Origin-Host of CCR-I and used in PCRF RAR request.
ratType	3GPP-RAT-Type AVP. Indicates the Radio Access Technology is currently serving the UE.
appId	Auth-Application-Id AVP. Used to advertise support of the Authentication and Authorization portion of an application.
ggsnIpAddress	GGSN-Address AVP. PGW Address Used.
Apn	Called-Station-Id AVP. Access Point Name Network Identifier.
sgsnMccmnc	3GPP-SGSN-MCC-MNC AVP. Serving node PLMN Identifier.
userLocationInfo	3GPP-User-Location-Info AVP. User Location Information.
sgsnIpAddress	SGSN-Address AVP. S-GW Address used.
sharedBucketReservationStatus	A boolean that indicate whether or not the sharedBucketReservation Status is set.
selectionMode	3GPP-Selection-Mode AVP. APN Selection Mode.
chargingCharacteristics	3GPP-Charging-Characteristics AVP. The Charging Characteristics applied to the IP-CAN session.
chargingRuleBaseName	Charging-Rule-Base-Name AVP. Charging Rule Base Name.
eventTimeStamp	Event-Timestamp AVP. Corresponds to the exact time the accounting is requested.
chargingCharecteristicsBits	Extracted from 3GPP-Charging-Characteristics AVP. The Charging Characteristics applied to the IP-CAN session.
appName	Application Name.
chargingID	3GPP-Charging-Id AVP. Charging ID.
userName	User-Name AVP. Contains the identification of the user.

A Diameter Sh v11 Session exists

The following condition input variables can be exposed in the Policy Builder GUI:

Table 10: Available Input Variables for Diameter Sh v11 Session Exists

Condition Input Variable	AVP Used/Description
Connected	A Boolean, which indicates whether a successful Sh Connection between PCRF and HSS is established, and a success Response (Diameter Result Code 20001) was received for the Sh requests (UDR/SNR).

A Diameter Sy v11 Session exists

This condition indicates that a valid Diameter Sy v11 session exists in the Policy Builder configuration for a given subscriber.

The following condition input variables can be exposed in the Policy Builder GUI:

Table 11: Available Input Variables for Diameter Sy v11 Session Exists

Condition Input Variable	AVP Used/Description
destRealm	Destination realm for Sy.
destHost	Destination host for Sy.
failureReason	The reason for failure in case Sy session is not established due to error (Last error code only).
destQueue	Destination queue (Internal field to know which policy director instance pick the request for processing).
retryTime	Retry timer in case connection fails.
lastSLReqType	Last spending limit request type.
lastSyResultCode	Last Sy result code.
syCountersIdentifierAndStatus	Sy counter identifier and status.
subscriberAccState	Subscriber account status.
slaReceived	A Boolean, set to true when a successful SLA-Initial is received.
sessionId	The unique id of the session.
connected	A Boolean, which indicates whether a successful Sy peer connection is established or not.

A Policy Derived AVP Exists

This condition indicates that a custom policy AVP that has been derived out of PolicyState exists in the system. The following condition input variables can be exposed in the Policy Builder GUI:

Table 12: Available Input Variables used for A Policy Derived AVP Exists

Condition Input Variable	AVP Used/Description
code	A derived AVP Name from the PolicyState.
value	A derived AVP value from the PolicyState.

The following AVPs are treated as Policy-Derived AVPs:

- Gx

3GPP-MS-TimeZone and SN-Transparent-Data

The following policy-derived AVP codes from the above AVPs can be used in the Use case Initiators:

- UE HOUR OFFSET
- UE MINUTE OFFSET
- UE DST
- UE DST
- BganMonthlyFapVolume

- Rx

Dynamic-PCC-Parameter AVP

- Dynamic-PCC-APN-Aggregate-Max-Bitrate-DL
- Dynamic-PCC-APN-Aggregate-Max-Bitrate-UL
- DynamicPCC-Congestion-Level

- Profile Related Mapping

The following profile related derived AVP codes can be used in the Use Case Initiators:

- Sh Profile
- LDAP Profile
- SPR Profile
- Additional ones
- Charging – congestionNextHourLevel

A Policy Derived AVP Does Not Exist

This condition indicates that a custom policy AVP that has been derived out of PolicyState does not exist in the system.

The input variables that can be exposed for this condition in the Policy Builder GUI are the same as those described in [A Policy Derived AVP Exists, on page 20](#).

An APN Bearer Details exists

This condition indicates APN bearer details as received in Active Traffic Management attribute exists in PCRF. The following condition input variables can be exposed in the Policy Builder GUI:

Table 13: Available Input Variables for Diameter Gx TGPP Session Exists

Condition Input Variable	AVP Used/Description
stage	Stage string as received in Active Traffic Management Attribute notification from UDC FE. Valid value is string.
ackMode	Acknowledge mode QCI as received in Active Traffic Management Attribute notification from UDC FE. Valid value is an Integer.
unackMode	Unacknowledge mode QCI as received in Active Traffic Management Attribute notification from UDC FE. Valid value is an Integer.
logicalAPN	Logical APN string as received in Active Traffic Management Attribute notification from UDC FE. Valid value is string.
frontEndId	Front End ID string as received in Active Traffic Management Attribute notification from UDC FE. Valid value is string.
priority	Priority value as received in Active Traffic Management Attribute notification from UDC FE. Valid value is an Integer.

An APN Bearer Details does not exists

This condition indicates no APN bearer details exist in Active Traffic Management attribute in PCRF. The input variables that can be exposed for this condition in the Policy Builder GUI are the same as described in [An APN Bearer Details exists](#) , on page 22.

There Exists a Network Session

This condition indicates that a valid network session exists in the Policy Builder configuration for a given subscriber. The network session exists irrespective of the interface type – Gx/Gy/Sh/Sy.

The following condition input variables can be exposed in the Policy Builder GUI:

Table 14: Available Input Variables for There Exists a Network Session

Condition Input Variable	AVP Used/Description
Mac Address	The mac address present in the session.

Condition Input Variable	AVP Used/Description
User Id	User Identifier – Can be msisdn or extracted from Network Access Identifier.
Framed IP	Framed-IP-Address AVP. The Ipv4 address allocated for the user.
Start Time	When the session starts.
Access Type	How the subscriber is accessing network.
Imsi	The subscriber identity in the Subscription-Id AVP being imsi.
Msisdn	The subscriber identity in the Subscription-Id AVP being msisdn.
Cell Site Id	The ID of the cell tower.
End Time	If the session is set to end, this gets set.
Framed IPV6 Prefix	Framed-IPv6-Prefix AVP. The Ipv6 prefix allocated for the user.
Msbm Initialized	NA
Credential Id	The Subscriber Profile Repository credential.
Spr Version	Version loaded from SPR in case an update is needed.
Current Session Duration	How long the session has been going.
Next Evaluation Date	The time when CPS will wake itself up and re-evaluate the session if another message is not received.
Expiration Date	Used on soft delete to note when the session should be cleaned up

There does not exist a network session

This condition indicates that a valid network session does not exist in the Policy Builder configuration for a given subscriber. The network session does not exist irrespective of the interface type – Gx/Gy/Sh/Sy.

The input variables that can be exposed for this condition in the Policy Builder GUI are the same as described in [There Exists a Network Session](#), on page 22.

ADTM Attribute Bearer Details Exists

This condition provides option to check for total of ACK mode, UNACK mode and combined for the subscriber across all the existing APNs.

The following condition input variables can be exposed in the Policy Builder GUI:

Table 15: Available Input Variables used for ADTM Attribute Bearer Details Exists

Condition Input Variable	AVP Used/Description
totalAckMode	Total ACK mode bearers (count) across all existing APNs for a subscriber. Valid value is an Integer.

Condition Input Variable	AVP Used/Description
totalUnAckMode	Total UNACK mode bearers (count) across all existing APNs for a subscriber. Valid value is an Integer.
totalCount	Total number of bearers (count) across all existing APNs for a subscriber. Valid value is an Integer.

A MSBMRolloverQuota exists

This condition is used to handle rollover use cases.



Note When rollover occurs for different rollover quotas (for same subscriber) at the same time, CPS sends a single notification with *Quota ID* as comma separated and consolidated roll-over amount. In such scenarios, it's highly recommended to configure rollover condition only with *rollOverOccur*.

The following condition input variables can be exposed in the Policy Builder GUI:

Table 16: Available Input Variables for MSBMRolloverQuota exists

Condition Input Variables	AVP Used/Description
rollOverOccur	Identifies rollover occurs or not. Value is true when rollover occurs.
accountBalance Code	Value will be account balance codes of rollover quota to which rollover occurs. If more than one rollover quota gets created at the same time, then value is a comma separated account balance codes.
quotaCode	Value will be quota codes of rollover quota to which rollover occurs. If more than one rollover quota gets created at the same time, then value is a comma separated quota codes.

Documentation

The documentation tab allows writing notes about the implementation which can be referred to later.

Custom Reference Data Tables

Overview

Custom Reference Data tables allow defining custom derived data for your installation and making decision based upon that data.

For example, a customer may not want to directly assign each subscriber a service so they get the appropriate QoS, instead, they want to derive the QoS based on data we get from the subscribers session. Custom Reference Data tables are the way to do that. Then (as per the previous section) we can wire the derived data directly into the service with the “Bind Field...” or “Pull Value From...” options.

Example

The following screens have a simple example setup. The example is that we want to derive the PCEF Rule Name based on the users APN (Access Point Name) and RAT (Radio Access Technology) name.

The logic is as follows:

Table 17: Example Table

APN	RAT	Rule Name	Rule Name Depleted
Cisco	3G	10_MB	5_MB
Cisco	Wi-Fi	20_MB	5_MB
Cisco	*	5_MB	1_MB
CiscoTest	3G	1_MB	5_KB
CiscoTest	Wi-Fi	2_MB	50_KB
*	*	1_MB	500_KB

The * (asterisk) can be read as a wildcard, but the table should match the 'most specific' entry. So, if my APN and RAT are Cisco and 3G, match the first row, not the 3rd row (Cisco, *) or the last row (*, *).



Note

These values are for example purposes only. Things like the RAT type value have been simplified to make the example easier to understand.



Note

RADIUS-based policy control is no longer supported in CPS 14.0.0 and later releases as 3GPP Gx Diameter interface has become the industry-standard policy control interface.

Screens

Search Table Groups

Search table groups allow grouping multiple Custom Reference Data Tables together logically, only executing the searching when needed. It also allows ordering the execution of tables so that tables can reference output of one table group in another to provide consistent and expected results. Additionally, Search Table Groups allows multiple different tables to populate the same AVPs in different ways.



Note

Search table groups and their respective CRD tables are listed based on the evaluation order value. If the evaluation order value is same for two or more tables then they are listed alphabetically.

The following parameters are configured under Search Table Group:

Table 18: Search Table Group Parameters

Parameter	Description
Name	The name of the Search Table Group.
Evaluation Order	The order in which groups get evaluated, starting with 0 and going higher.
Results Columns	These are the AVPs that will be added into processing. These need to be mapped to be the same as values from underlying tables. This allows populating the same AVPs from different tables. • Name: The name of the AVP. It should start with alphanumeric characters, should be lowercase, and should not start with numbers, no special characters are allowed, use “_” to separate words. For example, logical_apn = GOOD, logicalAPN = BAD, no_spaces • Display Name: The more human readable name of the AVP. • Use In Condition: This represents whether this row will be available for conditions in Policies or Use Case Templates. There is a performance cost to having these checked, so we recommend not checking them unless they are required. Default value is checked (true). • Default Value: The default value if no results are found from a Custom Reference Data Table. • Add/Remove: Adds or removes a row from the table. • Up/Down: Allows ordering the list, but has no impact on processing.
Table Search Initiators (OR Together)	This section controls whether or not the Search Table Group and all tables below will be executed. When you add multiple Initiators to the Search Table Group, the search is activated when ANY one of these initiators are true, as indicated by the caption “(OR Together)”. For more information, refer to table in section Use Case Initiators, on page 12.

Custom Reference Data Table

Policy Builder can be thought of as defining the 'schema' for the Custom Reference Data and the Control Center is responsible for filling out the schema.



Note Cisco recommends creating Custom Reference Data Tables under Search Table Groups. This allows you to specify the order in which the tables are searched and provide the default values if nothing is found. It is possible to re-parent a Custom Reference Data Table to another Search Table Group.

The following parameters can be configured under Custom Reference Data Table:

Table 19: Custom Reference Data Table Parameters

Parameter	Description
Name	This is the name of the table that is stored in the database. It must: • Start with alphanumeric characters • Have lowercase OR uppercase but no mixed case • Not start with numbers • Not have special characters • Use “_” to separate words For example, logical_apn = GOOD, logicalAPN = BAD, no_spaces.
Display Name	This is the name of the table that is displayed in Control Center.
Cache Results	This indicates whether the tables are cached in memory. This must be checked for production.
Activation Condition	This is the Custom Reference Data Trigger, which must be true before evaluating this table. This can be used to have multiple tables create the same data depending on conditions or to improve performance if tables are not required to be evaluated based on an initial condition.
Svn Crd Data	When enabled, indicates that the CRD table is an SVN CRD table and CRD data for the table is fetched from CRD CSV file present in SVN data source. When disabled, indicates that the CRD table data needs to be fetched from Mongo database.
Best Match	If checked, look-ups occur within a CRD table in the following order: • Exact string match • Higher priority regex match (if multiple regex patterns match) • Regular expression match (default behavior) • Wildcard character (*) For more information, refer to Best Match, on page 29.
Evaluation Order	This indicates the order the tables within the search table group should be evaluated. Starting with 0 and increasing.

Parameter	Description
Columns	Columns correspond to the schema for each column you are creating for this custom reference data table. • Name: The name of the column in the database. • Display Name: A more readable display name. • Use In Conditions: This represents whether this row is available for conditions in Policies or Use Case Templates. There is a performance cost to having these checked, so we recommend leaving them unchecked unless they are required. Default value is checked (true). • Type: the type determines what values are allowed when creating them in Control Center. • Text: The value is allowed to be any characters. For example, example123!. • Number: The value is allowed to be any whole number. For example, 1234. • Decimal: The value is allowed to be any number (including decimals). For example, 1.234. • True or False: The value must be true or false. For example, true. • Date: The value is a date without a time component (May 17th, 2020). • DateTime: The value is a date + time (May 17th, 2020 5:00pm). • Key: This indicates that this column is all or part of the key for the table that makes this row unique. By default, a key is required. Keys are also allowed to set the Runtime Binding fields to populate the data from the current message and session. Typically, keys are bound to data from the current session (APN, RAT Type) and other values are derived from them. Keys can also be set to a value derived from another custom reference data table. • Required: This indicates whether this field is marked required in Control Center. A key is always required.
Valid Values	These are the valid values which are allowed in Control Center (creates a list box). • List of Valid Values: A list of name and display name pairs which are used to create the list. Valid values can also contain a name which is the actual value of the column and a display value which allows Control Center to display an easier to use name. • Valid Values pulled from another Table: This allows initializing the list based on another custom reference data table. The name value is pulled from another table. There is no way to customize a display name in this manner.

Parameter	Description
Validation	The validation set here is checked by the Control Center before allowing a row to be added. • Regular Expression: This is the Java regular expression that is run on the proposed new cell value to validate it as described here. • Regular Expression Description: This is a message to the user indicating what the regular expression is trying to check.
Runtime Binding	Runtime binding is how key column data gets filled out (bound) from data in the current session. There are multiple ways to bind this data and it is also possible to set an operator to define what must match (equals, less than, and so on). • Bind to Subscriber AVP Code: This pulls the value from an AVP on the subscriber. It also pulls values from a session AVP or a Policy Derived AVP. • Bind to Session and Policy State Field: This pulls the value from a Policy State Data Retriever which knows how to retrieve a single value for a session The following Policy State Data Retrievers cannot be used for binding in CRD Tables: • Gx Predefined Charging Rules Retriever • Gx Charging Rulebase Names Retriever • Sd TDF Application Id List Retriever • Bind to a Result Column from another Table: This allows the key to be filled out from a columns value from another table. This allows normalizing the table structure and not having a large table with many duplicated values. • Bind to Diameter Request AVP code: This allows the key be filled out from an AVP on the Diameter request. • Matching Operator: This allows the row to be matched in other ways than having the value be equals. Default value is equals. • eq: Equal • ne: Not Equal • gt: Greater than • gte: Greater than or equal • lt: Less than • lte: Less than or equal

Best Match

CRD best match uses a hierarchical structure to perform match. The hierarchical structure is based on the input columns that is, key columns. The order of the input column is important in this hierarchy.

The data in the CRD table is cached in memory according to the hierarchy described in the following section:

Values in the first column are the root of the hierarchy. Number of unique values in the first column creates those many hierarchies. Then under the root, the values of the second key column similarly create those many children, and so on, and so forth, for other columns in the order.

While performing the best match, the value for the first column is used to select the root of the hierarchy. Once the root is selected, then the next match is performed only under the selected hierarchy so on, and so forth, for the children. At any level, first exact match is performed, if that fails then the pattern match is performed. Once the root or the child is selected, next search for the remaining column is restricted only to that hierarchy.

Example 1

If there are four columns in a best match table that is, {A,B,C,D} out of which key columns are {A,B,C} and output column is D and the table has the following values:

A	B	C	D
1	2	3	4
1	*	3	5
1	*	*	6
*	2	3	7

This results in the hierarchy as follows:

```
{ 1=
  { 2=
    { 3=4 }
    { *=
      { 3=5 }
      { *=6 }
    }
  }
{ *=
  { 2=
    { 3=7 }
  }
}
```

In the hierarchy, following input {4,6,3} results in no output. 4 matches "*" hierarchy thereby restricting the next input key to have only value 2 and the other key to have only value 3.

Regex Example:

Regex can be assigned priority by using numbers such as, "match1=" instead of "match=". The higher the number, higher is the priority.

Table 20: First Column is the Output Column

Test1	"a"	"b"	"*"
Test2	"a"	"b"	"c"
Test3	"*"	"b"	"d"
Test4	"*"	"*"	"*"
Test5	"a"	"c"	"*"

Test6	"*"	"*"	"7"
Test7	"a"	"match=[a-zA-Z]"	"*"
Test8	"a"	"7"	"match=[a-zA-Z]"
Test9	"a"	"match=[a-zA-Z]"	"1"
Test10	"a"	"match=[a-zA-Z]"	"2"
Test11	"Internet"	" match=P03.*"	"2001"
Test12	"Internet"	"match3=P03-Sy.*"	"2001"
Test13	"Internet"	"match2=P03-S.*"	"2001"
Test14	"Internet"	"match2=P03-[abc].*"	"2001"
Test15	"Internet"	"match1=P03-.*"	"2001"
Test16	"Internet"	"match3=P03-Gy.*"	"2001"

Table 21: Output with following Input

("a", "b", "4")	("Test1")
("a", "b", "c")	("Test2")
("7", "b", "d")	("Test3")
("7", "b", "e")	("Test4")
("a", "c", "4")	("Test5")
("a", "c", "7")	("Test5")
("a", "3", "7")	("Test6")
("a", "dd", "7")	("Test6")
("a", "7", "7")	("Test6")
("a", "W", "79")	("Test7")
("a", "7", "a")	("Test8")
("a", "W", "1")	("Test9")
("a", "W", "2")	("Test10")
("Internet", "P03", "2001")	("Test11")
("Internet", "P03-Sy", "2001")	("Test12")
("Internet", "P03-S", "2001")	("Test13")

("Internet", "P03-a", "2001")	("Test14")
("Internet", "P03-", "2001")	("Test15")
("Internet", "P03-Gy", "2001")	("Test16")

Example 2

Based on first key column (if all first key columns are same, then based on second key column and so on):

1. CPS selects root node of the search and gives preference to that row to continue the search in that row.
2. While traversing from top to bottom, this root node selection happens by giving preference to exact match.
3. If there is a priority configured, CPS gives preference to that row irrespective of the position of the row (except exact match scenario).

In case column values are same and one row has priority and other does not, then CPS ignores the row with the priority.

If there are four columns in a best match table that is, {A,B,C,D} out of which input columns are {A,B,C} and output column is D:

For example, if M, A_B, 12.2 are the first three inputs in the following rows, other are outputs:

- **Case 1:**

A	B	C	D
match=R P M	match=^(A?_)?B\$ ^(A?_)?B_C.	*	Test1
match=M S	match=^A_B.*	12.2	Test2

In this case, while traversing from top to bottom, CPS took the Row1 and continued the search and was able to satisfy the input.

- **Case 2:** Here Row2, first column has exact match for input, so preference is given to Row2.

A	B	C	D
match=R P M	match=^(A?_)?B\$ ^(A?_)?B_C	*	Test1
MYPHONE	match=^A_B.*	12.2	Test2

In this case, based on first key, column Row2 is chosen (as it is exact match for the input) and search is continued in Row2. If Row2 search fails, then preference is given to Row1.

- **Case 3:** In case of priority configuration, root node is selected based on the priority (if exact match is present, then CPS goes for exact match row).

A	B	C	D
match=M	match=^A_B.*	12.2	Test2

A	B	C	D
match1=R P M	match=^(A?_)?B\$ ^(A?_)?B_ C.*	^B_C.*	Test1

In this case Row2 has the priority. CPS selects Row2 as the root node and takes Row2 as output, or else it goes for Row1.

- **Case 3a:** When first column has priority but has same values, then priority is ignored.

A	B	C	D
match=R P M	match=^A_B.*	12.2	Test2
match1=R P M	match=^(A?_)?B\$ ^(A?_)?B_ C.*	^B_C.*	Test1

In this case, Row1 is the output.

Custom Reference Data Trigger

A custom reference Data Trigger is group of conditions that can be used to decide whether to evaluate a table or not. This can be used to derive the same data in different ways depending on conditions.

Refer to [Use Case Initiators, on page 12](#) as it is identical in function of setting up conditions and only different in what it is used for.

OOC and ROC Policy CRD Event Trigger Configuration

To enable a trigger for Out of Credit (OOC) and Retrieval of Credit (ROC) events, you must configure the Gx Out of Credit Retriever event in the session_action_mapping_gx CRD table. The configuration is used in conjunction with the other columns in the Session_Actions_Gx table in Control Center to derive the actions in the Session_Action output column.

Procedure

- Step 1** In Policy Builder, select the **Reference Data** tab.
- Step 2** In the left pane, select **Custom Reference Data Tables > Search Table Groups > session_action_mapping > session_action_mapping_gx**.
- Step 3** In the **Custom Reference Data Table** pane under **Columns**, select **ET_OOC_ROC** in the **Display Name** column.

Figure 9: ET OOC ROC Column Display Name

Custom Reference Data Table Some or all columns in this table have been

***Name** session_action_mapping_gx **Display Name** session_action_mapping_gx

Activation Condition ☒ Best

***Columns**

*Name	Display Name
LOGICAL_APN	LOGICAL APN
SH_PROFILE	SH PROFILE
ET_OOC_ROC	ET OOC ROC
RTRRULE3300_INSTALLED	RTRRULE3300 INSTALLED
RTRRULE3300_STATUS	RTRRULE3300 STATUS
RTR5300_INSTALLED	RTR5300 INSTALLED

Step 4 In the **Runtime** area, select the **Bind to Session/Policy State** option, and click **Select**.

Figure 10: Bind to Session/Policy State Option

Runtime

Which rows match when a message is received

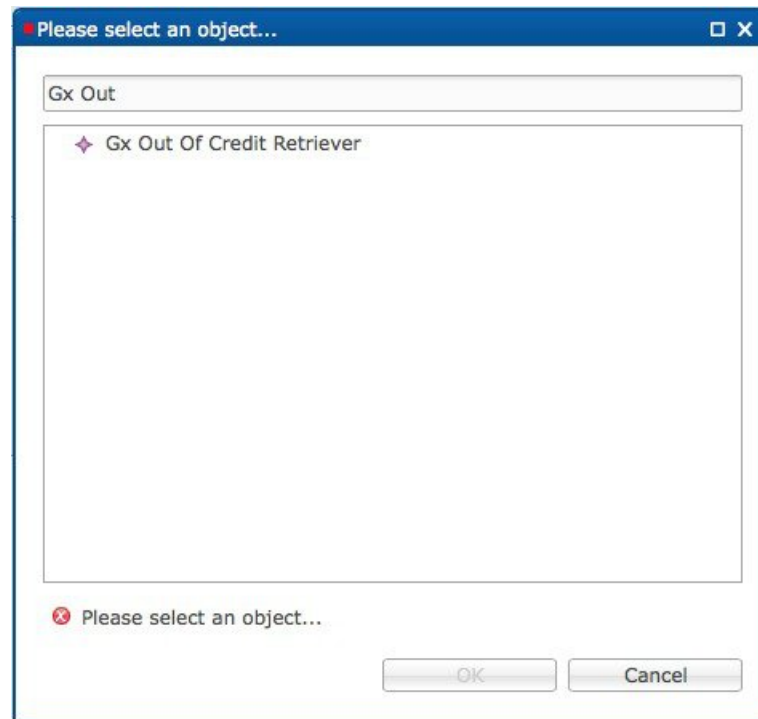
☐ None

☐ Bind to Subscriber AVP code

☒ Bind to Session/Policy State

Step 5 In the **Please select an object** dialog box, locate and select **Gx Out Of Credit Retriever**, and click **OK**.

Figure 11: Gx Out Of Credit Retriever Object



In the CCR-U message, the PCEF/PGW sends the following AVPs:

```
<messageid="CCR_U_01"appInterface="GX_TGPP">
<avp name="Charging-Rule-Report">
  <avp name="Charging-Rule-Name" value="SessionAction-Table_output-Column"></avp>
</avp>
```

The following AVP is sent for an Out of Credit event:

```
<avpname="Event-Trigger" value="15"></avp>
```

The following AVP is sent for a Reallocation of Credit event:

```
<avp name="Event-Trigger" value="16"></avp>
```

Expose Rules Installed to Policy CRD

Two parameters have been provided in the Gx PreConfiguredRule, TableDrivenChargingRule, and TableDrivenPredefinedChargingRule service configuration objects to expose installed PCC rules to the policy engine to be used for policy decisions in the CRD. These parameters are described below:

- Use In Rule Status Condition – Controls whether or not the PCC rule reported status AVPs are created. By default, this parameter is set to **true** for PreConfiguredRule and TableDrivenChargingRule, and to **false** for TableDrivenPredefinedChargingRule.
- Use in Rule Install Condition – Controls whether or not the PCC rule installed AVPs are created. By default, this parameter is set to false.



Note To expose installed PCC rules to the policy CRD, you must set this parameter to true.

These parameters are shown in the following two figure as they appear in the TableDrivenChargingRule and PreConfiguredRule service configuration objects.

Figure 12: TableDrivenChargingRule

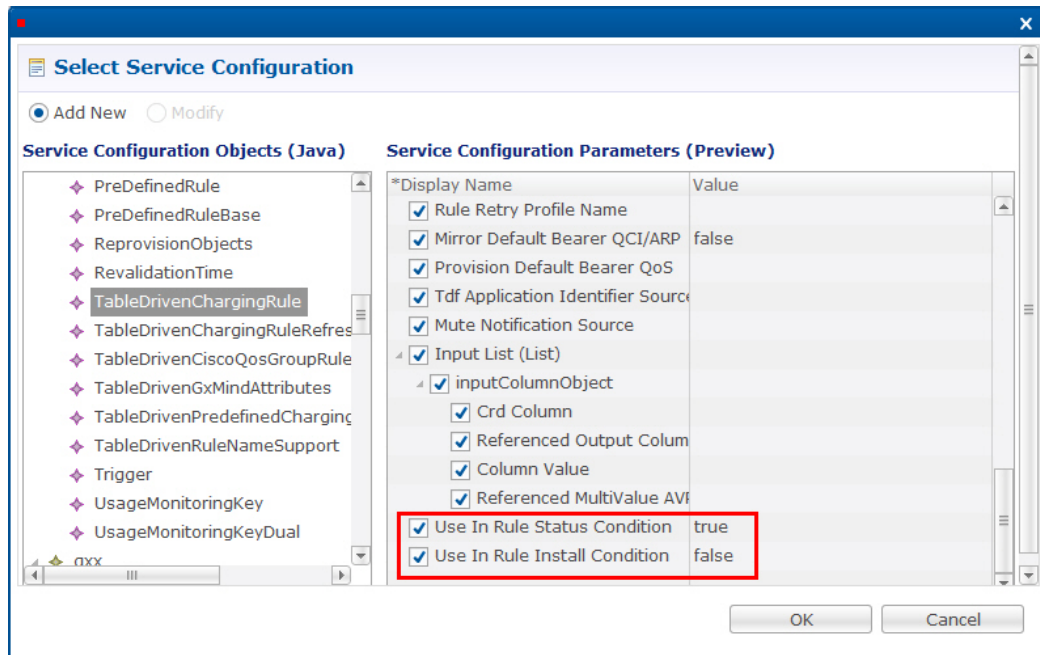
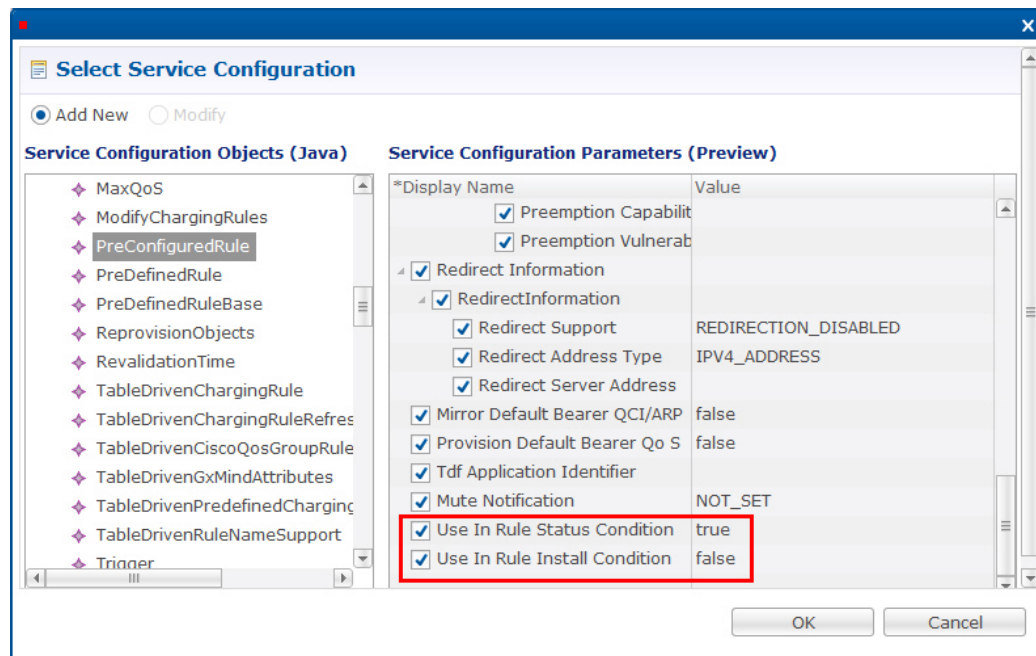
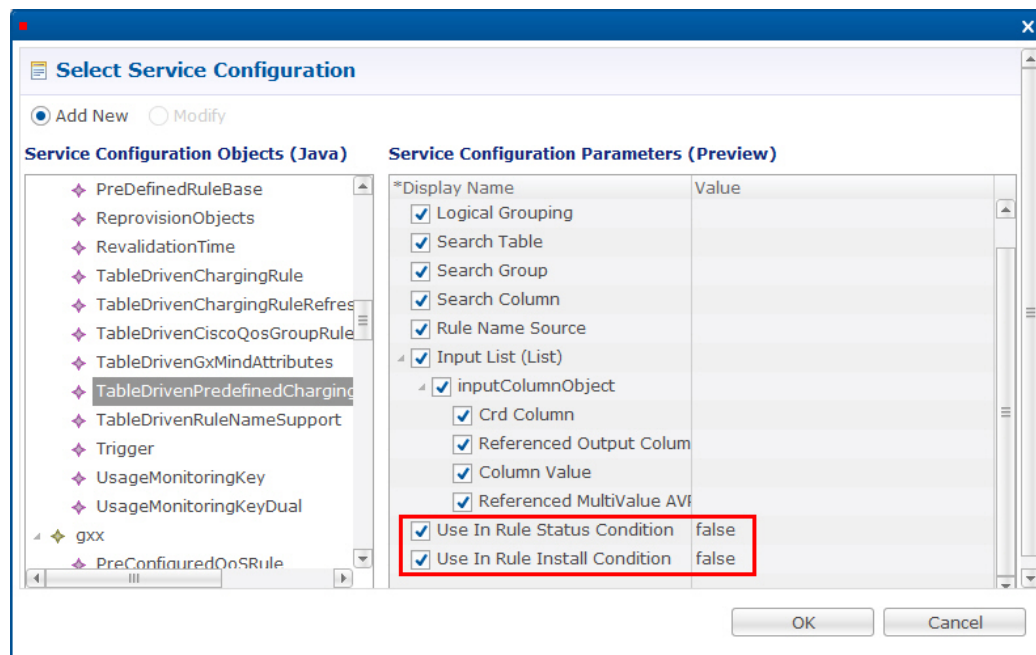


Figure 13: PreConfiguredRule



The following figure shows the two parameters as they appear in the TableDrivenPredefinedChargingRule service configuration object.

Figure 14: TableDrivenPredefinedChargingRule



In the CCR-U request, the PCEF/PGW sends the following AVPs:

```
<avp name="Charging-Rule-Report">
  <avp name="Charging-Rule-Name" value="DTL3300"> </avp>
```

```
<avp name="PCC-Rule-Status" value="0"> </avp>
</avp>
```

You must also configure your CRD table by binding the **<RuleName>_installed** column with RI-<RuleName> and by binding the **<RuleName>_Status** column with RS-<RuleName>.

Procedure

- Step 1** In Policy Builder, select the **Services** tab.
- Step 2** In the left pane, click **Services**.
- Step 3** Navigate to any service options that use the PreConfiguredRule, TableDrivenChargingRule, and TableDrivenPredefinedChargingRule service configuration objects, and change their **Use In Rule Install Condition** parameters to **true**.
- Step 4** In Policy Builder, select the **Reference Data** tab.
- Step 5** In the left pane, select **Custom Reference Data Tables > Search Table Groups**.
- Step 6** Go to your CRD table; for example, under **session_action_mapping > session_action_mapping_gx**.
- Step 7** In the **Custom Reference Data Table** pane under **Columns**, select **<RuleName>_Installed**.
- Step 8** In the **Runtime** area, select the **Bind to Subscriber AVP code** option, and type **RI-<RuleName>**.

An example is shown in the following figure.

Figure 15: Binding to RI-RTRULE3300 Subscriber AVP code

The screenshot shows the Cisco Policy Builder interface. The left pane displays a tree view of Custom Reference Data Tables, with 'session_action_mapping_gx' selected. The main pane shows the configuration for this table. The 'Columns' section lists several columns, including 'RTRRULE3300_INSTALLED' and 'RTRRULE3300_STATUS'. The 'Runtime' section shows the 'Bind to Subscriber AVP code' option selected, with the value 'RI-RTRRULE3300' entered in the text field.

*Name	Display Name	*Use In Conditic	*Type	Key	Required
LOGICAL_APN	LOGICAL APN	☒	Text	☒	☒
SH_PROFILE	SH PROFILE	☒	Text	☒	☒
ET_OOC_ROC	ET OOC ROC	☒	Text	☒	☒
RTRRULE3300_INSTALLED	RTRRULE3300 INSTALLED	☒	Text	☒	☒
RTRRULE3300_STATUS	RTRRULE3300 STATUS	☒	Text	☒	☒
RTR5300_INSTALLED	RTR5300 INSTALLED	☒	Text	☒	☒

Runtime
Which rows match when a message is received

☒ Bind to Subscriber AVP code
RI-RTRRULE3300

☐ Bind to Session/Policy State

☐ Bind to a result column from another table

- Step 9** Select **<RuleName>_STATUS** under **Columns**.
- Step 10** In the **Runtime** area, select the **Bind to Subscriber AVP code** option, and type **RS-<RuleName>**.

An example is shown in the following figure.

Figure 16: Binding to RS-RTRULE3300 Subscriber AVP code

CISCO POLICY BUILDER REFERENCE DATA SERVICES POLICIES

File Tools

Systems

- Account Balance Templates
- Andof Clients
- Custom Reference Data Tables
 - Summary
 - Search Table Groups
 - apn_domain_mapping (0)
 - roaming_mapping (1)
 - subscriber_profile_mapping (2)
 - session_action_mapping (3)
 - session_action_mapping_sv
 - session_action_mapping_gx
 - default_bearer_mapping (4)
 - anbr_mapping (5)
 - dedicated_bearer_mapping (6)
 - rs_mapping (7)
 - anbr_profile (9)
 - qos_profile (8)
 - default_bearer_rule_profile (10)
 - predefined_rule_profile (11)
 - override_control_profile (12)
 - event_trigger_profile (13)
 - trigger_type_profile (14)
 - cc_group_profile (15)
 - dedicated_bearer_rule_profile (16)
 - Custom Reference Data Triggers
 - Custom Reference Data Tables
- DM Configuration

Custom Reference Data Table Some or all columns in this table have been published and will be read only. Newly added columns will be editable.

***Name** session_action_mapping_gx **Display Name** session_action_mapping_gx ☒ Cache

Activation Condition select clear ☒ Best ***Evaluation Order** 1

***Columns**

*Name	Display Name	*Use In Condit	*Type	Key	Required
LOGICAL_APN	LOGICAL APN	☒	Text	☒	☒
SH_PROFILE	SH PROFILE	☒	Text	☒	☒
ET_OOC_ROC	ET OOC ROC	☒	Text	☒	☒
RTRRULE3300_INSTALLED	RTRRULE3300 INSTALLED	☒	Text	☒	☒
RTRRULE3300_STATUS	RTRRULE3300 STATUS	☒	Text	☒	☒
RTRS300_INSTALLED	RTRS300 INSTALLED	☒	Text	☒	☒

Add Remove

Column Details

Valid Values
The values allowed in Control Center for this column

☐ All
☐ List of Valid

Validation
Validation used by Control Center

Regular Expression

Regular Expression Description

Runtime
Which rows match when a message is received

☐ None
☒ Bind to Subscriber AVP code
RS-RTRRULE3300
☐ Bind to Session/Policy State
☐ Bind to a result column from another table

