Policy Builder Configuration

- Plug-in Configuration, on page 1
- Diameter Application, on page 26
- Routing AVP Definition, on page 33
- Custom Reference Data Tables, on page 37

Plug-in Configuration

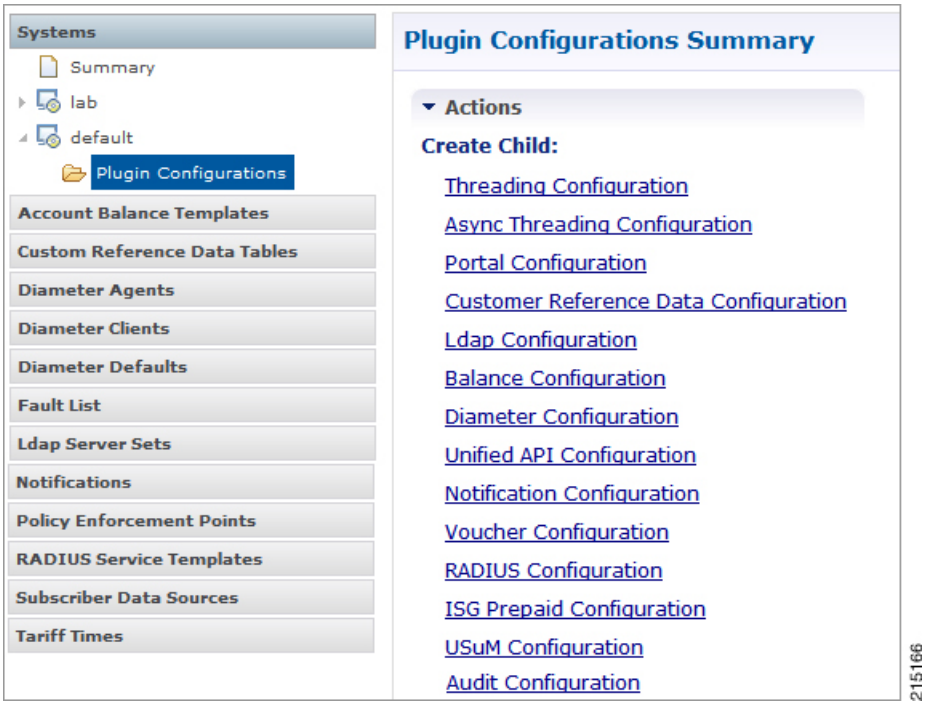
Cisco Policy Builder provides core plug-ins for customizing and optimizing your installation.

- Configurations set at the system level are system-wide except as noted in the bullet items below.
- Configurations set at the cluster level apply to that cluster and the instances in it. A value set here overrides the same value set at the system level.
- Configurations set at the instance level apply to the instance only and override the same value set at the cluster or system level.

Select the **Create Child** action in a **Plug-in Configuration** node in the **Systems** tree to define them. You can change any of the variables from the default, or choose not to use a plug-in, as necessary.

When you create a system from the example, the following configuration stubs appear at the cluster and instance level:

Figure 1: Create Child Action



Threading Configuration

A threading configuration utility is provided for advanced users.

Click **Threading Configuration** in the right pane to add the threading configuration to the system. If you are planning to run the system with higher TPS, then you need to configure Threading Configuration. For further information, contact your Cisco Technical Representative.

The Threading Plug-in controls the total number of threads in CPS vDRA that are executing at any given time.

The following parameters can be configured under Threading Configuration:

Table 1: Threading Configuration Parameters

Parameter	Description
Thread Pool Name	Name of the thread pool. Following names can be configured in CPS vDRA: • broadhop-bindings • broadhop-slf • broadhop-receivers • broadhop-qprocessor
Threads	Number of threads to set in the thread pool.

Parameter	Description
Queue Size	Size of the queue before they are rejected.
Scale By Cpu Core	Select this check box to scale the maximum number of threads by the processor cores.

Async Threading Configuration

Click **Async Threading Configuration** in the right pane to add the configuration in the system.

Use the default values for the Async Threading Plug-in. The Async configuration controls the number of asynchronous threads.

Note Currently, CPS vDRA does not have any asynchronous threads. However, you must add "Async Threading Configuration" and keep this table empty.

The following parameters can be configured under Async Threading Configuration.

Table 2: Async Threading Configuration

Parameter	Description
Default Processing Threads	The number of threads that are allocated to process actions based on priority.
Default Action Priority	The priority assigned to an action if it is not specified in the Action Configurations table.
Default Action Threads	The number of threads assigned to process the action if it is not specified in the Action Configurations table.
Default Action Queue Size	The number of actions that can be queued up for an action if it is not specified in the Action Configurations table.
Default Action Drop Oldest When Full	When checked, the oldest queued action is dropped from the queue when a new action is added to a full queue. Otherwise, the new action to add is ignored. This check box applies to all the threads specified. To drop a specific thread, leave this unchecked and use the Action Configurations table.
Action Configurations Table	
Action Name	The name of the action. This must match the implementation class name.
Action Priority	The priority of the action. Used by the default processing threads to determine which action to execute first.
Action Threads	The number of threads dedicated to processing this specific action.
Action Queue Size	The number of actions that can be queued up.

Parameter	Description
Action Drop Oldest When Full	For the specified action only: When checked, the oldest queued action is dropped from the queue when a new action is added to a full queue. Otherwise, the new action to add is ignored.

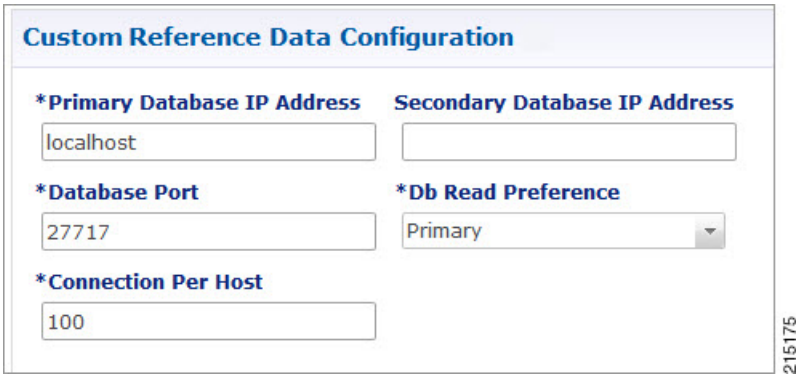
Custom Reference Data Configuration

Before you can create a custom reference data table, configure your system to use the Custom Reference Data Table plug-in configuration.

You only have to do this one time for each system, cluster, or instance. Then you can create as many tables as needed.

Click **Custom Reference Data Configuration** from right pane to add the configuration in the system.

Figure 2: Custom Reference Data Configuration



Here is an example for HA and AIO setups:

- HA example:
  - Primary Database Host/IP Address: sessionmgr01
  - Secondary Database Host/IP Address: sessionmgr02
  - Database Port: 27717
- AIO example:
  - Primary Database Host/IP Address: localhost or 127.0.0.1
  - Secondary Database Host/IP Address: NA (leave blank)
  - Database Port: 27017

The following parameters can be configured under Custom Reference Data Configuration.

Table 3: Custom Reference Data Configuration

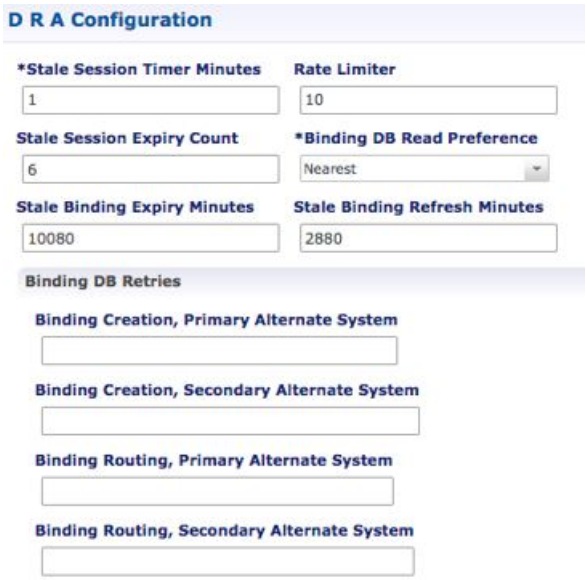
Parameter	Description
Primary Database IP Address	IP address of the primary sessionmgr database.
Secondary Database IP Address	Optional, this field is the IP address of a secondary, backup, or failover sessionmgr database.
Database Port	Port number of the sessionmgr. It should be the same for both the primary and secondary databases.
Db Read Preference	Read preference describes how sessionmgr clients route read operations to members of a replica set. You can select from the following drop-down list: • Primary: Default mode. All operations read from the current replica set primary. • PrimaryPreferred: In most situations, operations read from the primary but if it is unavailable, operations read from secondary members. • Secondary: All operations read from the secondary members of the replica set. • SecondaryPreferred: In most situations, operations read from secondary members but if no secondary members are available, operations read from the primary For more information, refer to http://docs.mongodb.org/manual/core/read-preference/.
Connection Per Host	Number of connections that are allowed per DB Host. Default value is 100.

For more information on Custom Reference Data configuration, refer to the *CPS Operations Guide* for this release.

DRA Configuration

Click **DRA Configuration** from the right pane in Policy Builder to add the configuration in the system.

Figure 3: DRA Configuration



The following parameters can be configured under DRA Configuration:

Table 4: DRA Configuration Parameters

Parameter	Description
Stale Session Timer Minutes	Indicates the time after which the audit RAR should be generated (in the subsequent audit RAR process cycle that runs every minute in CPS vDRA) for sessions that are stale. Default: 180 minutes (recommended value) Minimum: 10 minutes Maximum: 10080 minutes
Rate Limiter	Indicates the number of audit RARs per second that should be sent out by CPS vDRA. For example, if there are 100 stale sessions found in the audit RAR process, but the Rate Limiter is configured as 10, then audit RARs are generated at 10 RAR/sec for the next 10 seconds. Default: 10 (recommended value) Minimum: 1 Maximum: 1000 (maximum number of RAR messages per second from vDRA to PCEF)

Parameter	Description
Stale Session Expiry Count	Specifies the number of retries vDRA should do for a stale session if there is no response of audit RAR or if there is Result-Code in RAA (for audit RAR) other than 5002 or 2001. Default: 6 (recommended value) Minimum: 0 (Session deleted without sending RAR) Maximum: 10
Binding DB Read Preference	Used to select the mode when reading from Binding DB. Use "nearest" mode for better performance of traffic that needs only read operation on Binding DB. Default: Nearest (recommended setting)
Stale Binding Expiry Minutes	Duration after which the binding database records expire. The timer is initialized when the session is created. The records are deleted when the time since the last refresh exceeds Stale Binding Refresh Minutes. Default: 10080 minutes (168 hours or one week) (recommended value) Minimum: 10 minutes Maximum: 43200 minutes (28 days) For more information about binding DB audits and stale records, see Binding DB Audit, on page 11.
Stale Binding Refresh Minutes	Duration for which the expiry time of the binding database records is refreshed. Default: 2880 minutes (48 hours or 2 days - recommended value). Minimum: 10 minutes Maximum: 10080 minutes (one week) **Note** Stale Binding Expiry Minutes should be multiple of Stale Binging Refresh Minutes. Stale Binding Refresh Minutes should be greater than Stale Session Timer Minutes.

Parameter	Description
Binding Creation, Primary Alternative System	Name of vDRA system to retry Gx CCR-i When vDRA tries to route a Gx CCR-i request, but is unable to reach the database, the configured values of first the primary, then the secondary systems are used to route the Gx CCR-i to a different vDRA to try the database. The retry is stopped if that vDRA also cannot reach the database.
Binding Creation, Secondary Alternative System	Name of secondary vDRA system to retry Gx CCR-i
Binding Routing, Primary Alternative System	Name of vDRA system to retry Rx AAR When vDRA tries to route a Rx AAR request, but is unable to reach the database, the configured values of first the primary, then the secondary systems are used to route the Rx AAR to a different vDRA to try the database. The retry is stopped if that vDRA also cannot reach the database.
Binding Routing, Secondary Alternative System	Name of secondary vDRA system to retry Rx AAR
Settings	Refer to Settings.
Rate Limits	Refer to Rate Limits.
DRA Feature	Refer to DRA Feature.
DRA Inbound Endpoints	Refer to DRA Inbound Endpoints, on page 16.
DRA Outbound Endpoints	Refer to DRA Outbound Endpoints, on page 18.
Relay Endpoints	Refer to Relay Endpoints, on page 20.

Settings

Click **Settings** check box to open the configuration pane. An example configuration is shown below:

Figure 4: DRA Configuration - Settings

The following parameters can be configured under **Settings**:

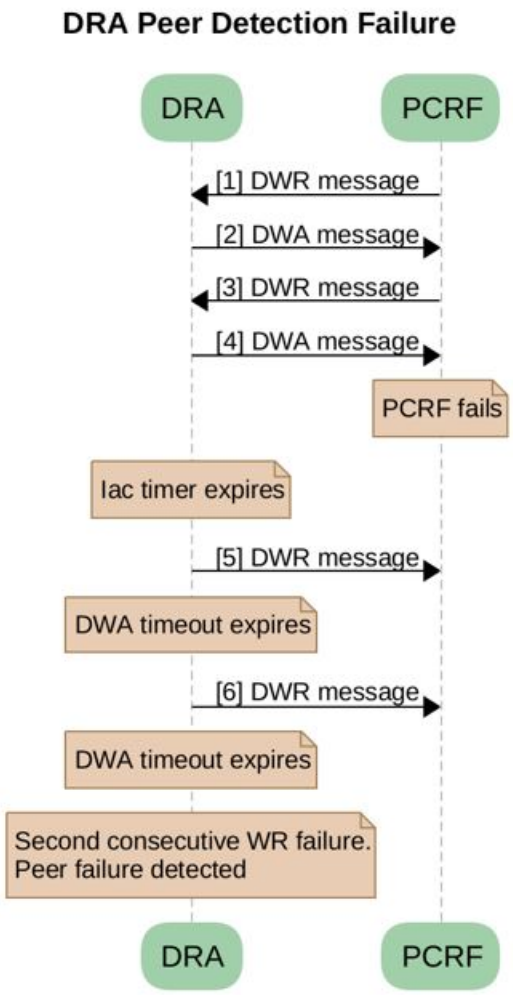
Table 5: DRA Configuration - Settings Parameters

Parameter	Description
Stop Timeout Ms	Determines how long the stack waits for all resources to stop. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)
Cea Timeout Ms	Determines how long it takes for CER/CEA exchanges to timeout if there is no response. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)

Parameter	Description
Iac Timeout Ms	Determines how long the stack waits before initiating a DWR message exchange on a peer connection from which no Diameter messages have been received. The timeout value is in milliseconds. Default: 5000 ms (recommended value) Minimum: 1000 ms Maximum: 30000 ms (30 seconds)
Dwa Timeout Ms	Determines how long the stack waits for a DWA message in response to a DWR message. If no Diameter message (DWA or other message) is received on the peer connection during the first timeout period, the stack counts a failure, sends another DWR message, and restarts the Dwa timer. If no Diameter messages are received during the second timeout period, the stack counts a second failure. After two consecutive failures, the stack considers the peer connection as failed, and closes the connection. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)
Dpa Timeout Ms	Determines how long it takes for a DPR/DPA exchange to timeout if there is no response. The delay is in milliseconds. Default: 5000 ms (recommended value) Minimum: 1000 ms Maximum: 30000 ms (30 seconds)
Rec Timeout Ms	Determines how long it takes for the reconnection procedure to timeout. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)

The following figure illustrates the timers in peer detection:

Figure 5: vDRA Peer Detection Failure



Binding DB Audit

The Binding DB Audit automatically deletes stale records from the binding DBs. When a Gx session record is created, binding records for the session binding keys are also created. When each binding record is created, the binding record expiry time is initialized to the sum of the session creation time and the Stale Binding Expiry Minutes (that you can configure in Policy Builder). A binding record is considered stale if it cannot be deleted when its associated session record is deleted (this occurs typically due to communication failures). The binding records are audited via a binding audit background process. If the audit process finds a binding record that is past the expiry time, the binding record is considered stale and deleted from the database. Note that the binding audit process does not perform a session DB lookup nor does it perform any Diameter signaling with the GW before deletion.

To prevent a binding record from becoming stale, the session audit process periodically updates the expiry time for bindings associated with sessions in the session DB. The session maintains a stale binding refresh timer that is initialized to the sum of the session creation time and Stale Binding Refresh Minutes. When the session audit process finds a session with a refresh time that has passed, it updates a new expiry time (calculated from current time plus the Stale Binding Expiry Minutes) to its associated bindings. The write is conditional

on the session-id matching the Gx session-id in the binding record. This refresh action prevents the binding audit process from incorrectly deleting active bindings from its binding database. The following figures illustrate the working of binding DB audit and refresh:

Figure 6: Binding DB Audit

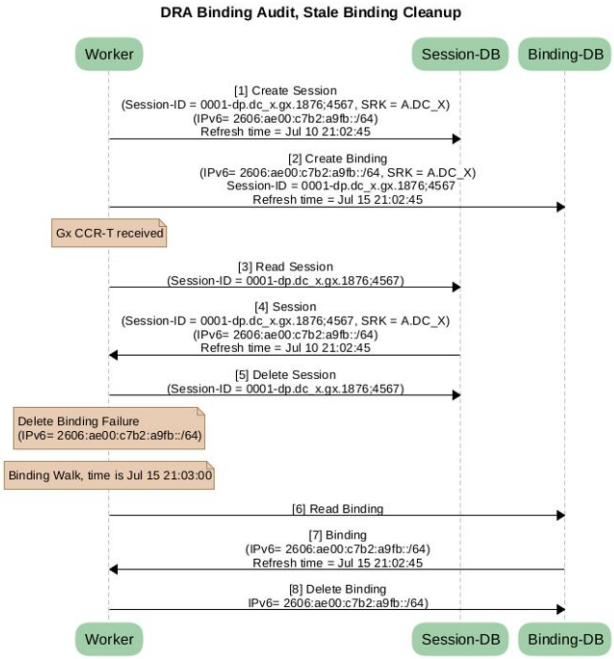
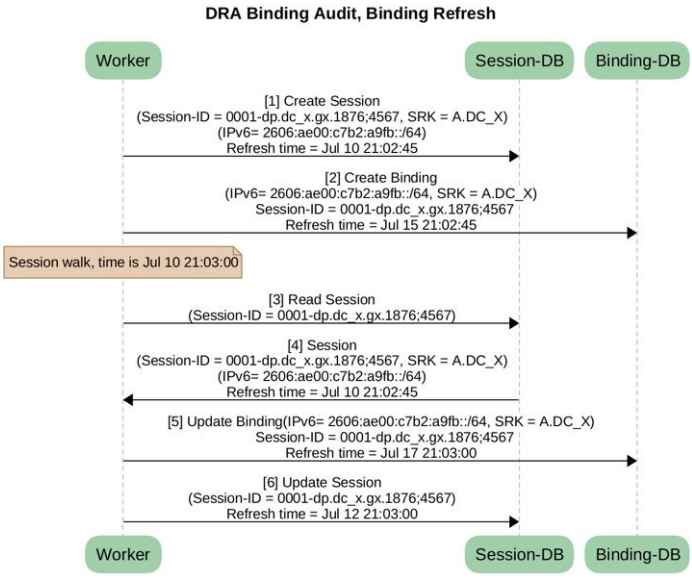


Figure 7: Binding Refresh



Rate Limits

Rate limit per process instance on Policy Director (lb) VM can be managed using this configuration.

Default is unchecked, that is, no rate limits for Diameter traffic (recommended setting).

If enabled, the following parameters can be configured under **Rate Limits**:

Table 6: DRA Configuration - Rate Limits

Parameter	Description
Rate Limit per Instance on Policy Director	Allowable TPS on a single instance of policy server (QNS) process running on the Policy Director. Minimum: 1 Maximum: 5000 **Note** Contact your Cisco representative for usecase-specific recommended values.
Result-Code in Response	Indicates the error code that must be used while rejecting requests, due to rate limits being reached. Default: 3004
Error Message in Response	Select the check box to drop the rate-limited messages without sending error response. If the check box is not selected, then the rate limited message are dropped with error response as configured.
Drop Requests Without Error Response	Select the check box to drop rate limited messages without sending error response. If the check box is unchecked, then the rate limited messages are dropped with error response as configured. To accommodate configuration to either drop the request or send an error response, a column *Discard Behavior* can be added under Peer Rate Limit Profile. The column may have one of the two possible values: • Send Error Response • Drop Message Default: Unchecked (recommended setting) For more information, refer to Peer Rate Limit. **Important** If both Rate Limit Error Code and Rate Limit Error String are provided along with Rate Limit Action as "Drop Message", the Rate Limit Action will take precedence and the other two fields will be ignored.

Here is the list of the available combinations for rate limiting:

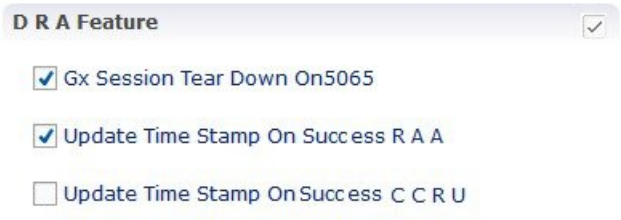
Table 7: Rate Limiting Combinations

Rate Limiting Type	With Error Code	With Error Code and Error Message	Without Error Code (Drop)
Instance Level	Yes	Yes	Yes
Peer Level Egress	Yes	Yes	Yes
Peer Level Egress with Message Level	Yes	Yes	Yes
Egress Message Level (No Peer Level RL)	Yes	Yes	Yes
Peer Level Ingress	Yes	Yes	Yes
Peer Level Ingress with Message Level	Yes	Yes	Yes
Ingress Message Level (No Peer Level RL)	Yes	Yes	Yes

DRA Feature

Click **DRA Feature** check box to open the configuration pane.

Figure 8: DRA Configuration - DRA Feature



The following parameters can be configured under **DRA Feature**:

Table 8: DRA Features

Parameter	Description
Gx Session Tear Down On5065	By default, **Gx Session Tear Down On5065** flag is enabled (recommended setting). When the PCRF responds with a Experimental Result Code of 5065 in AAAnswer on Rx Interface, DRA deletes its internal binding and session created for the transaction.A RAR with appropriate Session-Release-Cause AVP will also be sent to the PCEF. **Important** When using this flag, there will always be a database query to fetch Gx session id. So this means that the database transactions will linearly increase with AAR traffic on Rx Interface.
Update Time Stamp On Success RAA	When this check box is selected, session timestamp will be updated on receipt of success RAA (Result-Code: 2001) from PCEF. [1] Default is checked (recommended setting) **Important** When using this flag, there will always be a database query to fetch Gx session id. So this means that the database transactions will linearly increase with AAR traffic on Rx Interface.
Update Time Stamp On Success CCRU	When this check box is selected, session timestamp will be updated on receipt of success CCR-U (Result-Code: 2001) from PCEF. [2] Default is unchecked (recommended setting) **Important** When using this flag, there will always be a database query to fetch Gx session id. So this means that the database transactions will linearly increase with AAR traffic on Rx Interface.
Enable Proxy Bit Validation	Enables P bit validation. vDRA validates the P bit in the Diameter request and, if set, the message maybe proxied, relayed, or redirected. If this option is disabled, the P bit in the request is not checked and the request is not considered proxiable. Default: Enabled.

Parameter	Description
Enable Mediation	Enable advanced mediation capabilities in both egress and ingress direction. This feature allows you to configure vDRA to change the value of the Result-Code in Diameter Answer, use mediation to hide topology, prepend label to Destination Host AVP, etc.
Enable Doic	Enable or disable abatement action for Diameter requests towards PCRF, HSS, AAA, and OCS servers based on reporting of overloaded conditions using the architecture described in RFC 7683 Diameter Overload Indication Conveyance (DOIC). DOIC can be enabled/disabled at peer group level in Peer Group SRK Mapping table. If the destination peer is congested or overloaded, you can choose to either forward, divert, or drop messages.
Slf Max Bulk Provisioning TPS	Rate at which subscribers are provisioned in the SLF database. SLF bulk provisioning generates high number of database write operations in a short duration of time. To spread out the operations over a period of time and mitigate the performance issue, configure the TPS. The rate limit adds delay between transactions and thereby limits the number of transactions executed per second. For more information about SLF bulk provisioning, see the *CPS vDRA Operations Guide*.

[1] The time stamp will be updated on generation of Stale RAR. Also, if a success RAR/RAA(2001) comes after generation of Stale RAR, then the Stale RAR counter will be reset.

[2] The time stamp will be updated on generation of Stale RAR. Also, if a success CCR(U)/CAA(2001) comes after generation of Stale RAR, then the Stale RAR counter will be reset.

DRA Inbound Endpoints

The following parameters can be configured under **DRA Inbound Endpoints**:

Table 9: DRA Configuration - DRA Inbound Endpoints Parameters

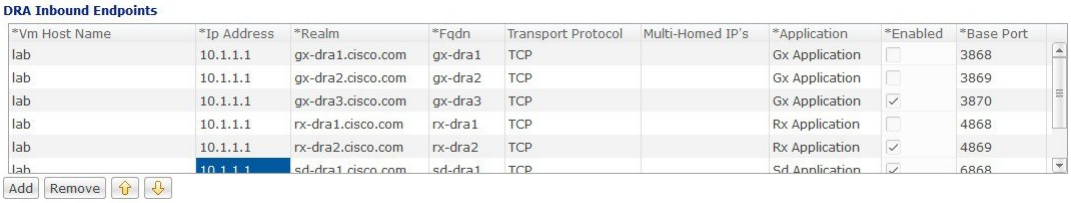
Parameter	Description
Vm Host Name	Host Name of the VM that hosts this CPS vDRA endpoint.
Ip Address	Address on which this CPS vDRA endpoint should bind to.
Realm	Realm of the CPS vDRA endpoint.

Parameter	Description
Fqdn	Fully Qualified Domain Name of the CPS vDRA end point.
Transport Protocol	Allows you to select either 'TCP' or 'SCTP' for the selected DRA endpoint. Default value is TCP. If the DRA/relay endpoint is to be configured for SCTP, the Transport Protocol should be selected as SCTP for those endpoints.
Multi-Homed IPs	This is a comma separated list of IP addresses that CPS vDRA will use to start the diameter stack with multi-homing enabled for SCTP transport. Diameter stack with TCP transport will still use the existing 'Local Bind Ip' field to specify any specific IP address for TCP stack. CPS vDRA will use the 'Local Bind Ip' to bring up SCTP stack and use it along with the 'Multi Homing Hosts' to start the SCTP transport with multi-homing support. **Note** While using SCTP multi-homing functionality review the Linux network and gateway configurations for supporting multiple networks on different subnets. CPS supports Centos 6 release and reverse path filtering kernel parameter (rp_filter) values can be set for allowing packets from different subnets on Policy Director VMs. The default behavior in Centos 6 is to discard the packets in such scenarios. **Note** Both IPv4 and IPv6 are supported in vDRA endpoint configuration. For IPv6, you can enter either short or long format. The configuration for multi-homing is validated by netstat command on lb01: `netstat -apn | grep 3898`
Application	Refers to 3GPP Application ID of the interface. You can select multiple applications on a peer connection. For example, S6a and SLg on a single IPv4/SCTP Multi-homed peer connection.
Enabled	Check to enable the endpoint.

Parameter	Description
Base Port	Refers to the port on which the CPS vDRA listens for incoming connections.

An example configuration is shown below:

Figure 9: DRA Inbound Endpoints - Example Configuration



DRA Inbound Endpoints

*Vm Host Name	*Ip Address	*Realm	*Fqdn	Transport Protocol	Multi-Homed IP's	*Application	*Enabled	*Base Port
lab	10.1.1.1	gx-dra1.cisco.com	gx-dra1	TCP		Gx Application		3868
lab	10.1.1.1	gx-dra2.cisco.com	gx-dra2	TCP		Gx Application		3869
lab	10.1.1.1	gx-dra3.cisco.com	gx-dra3	TCP		Gx Application	✓	3870
lab	10.1.1.1	rx-dra1.cisco.com	rx-dra1	TCP		Rx Application		4868
lab	10.1.1.1	rx-dra2.cisco.com	rx-dra2	TCP		Rx Application	✓	4869
lab	10.1.1.1	sd-dra1.cisco.com	sd-dra1	TCP		Sd Application	✓	6868

Add Remove

DRA Outbound Endpoints

The following parameters can be configured under **DRA Outbound Endpoints**:

Table 10: DRA Configuration - DRA Outbound Endpoints Parameters

Parameter	Description
Vm Host Name	Host Name of the VM that hosts this CPS vDRA endpoint.
Ip Address	Address on which this CPS vDRA endpoint should bind to.
Realm	Realm of the CPS vDRA endpoint.
Fqdn	Fully Qualified Domain Name of the CPS vDRA end point.
Transport Protocol	Allows you to select either 'TCP' or 'SCTP' for the selected CPS vDRA endpoint. Default value is TCP. If the DRA/relay endpoint is to be configured for SCTP, the Transport Protocol should be selected as SCTP for those endpoints.

Parameter	Description
Multi-Homed IPs	This is a comma separated list of IP addresses that CPS vDRA will use to start the diameter stack with multi-homing enabled for SCTP transport. Diameter stack with TCP transport will still use the existing 'Local Bind Ip' field to specify any specific IP address for TCP stack. CPS vDRA will use the 'Local Bind Ip' to bring up SCTP stack and use it along with the 'Multi Homing Hosts' to start the SCTP transport with multi-homing support. **Note** While using SCTP multi-homing functionality review the Linux network and gateway configurations for supporting multiple networks on different subnets. CPS supports Centos 6 release and reverse path filtering kernel parameter (rp_filter) values can be set for allowing packets from different subnets on Policy Director VMs. The default behavior in Centos 6 is to discard the packets in such scenarios. **Note** Both IPv4 and IPv6 are supported in vDRA endpoint configuration. For IPv6, you can enter either short or long format. The configuration for multi-homing is validated by netstat command on lb01: `netstat -apn | grep 3898`
Application	Refers to 3GPP Application ID of the interface.
Enabled	Check to enable the endpoint.
Peer Realm	Diameter server realm.
Peer Host	Diameter server host. By default, the connection is initiated on the standard diameter port (3868). If a different port needs to be used than the peer name must be defined using the host:port format.

An example configuration is shown below:

Figure 10: DRA Outbound Endpoints - Example Configuration

Relay Endpoints

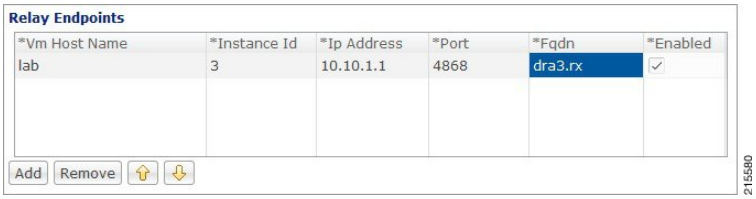
The following parameters can be configured under **Relay Endpoints**:

Table 11: DRA Configuration - Relay Endpoints Parameters

Parameter	Description
Vm Host Name	Host Name of the VM that hosts this Relay endpoint.
Instance Id	Instance Identifier is the ID of the current Instance.
Ip Address	Address on which this DRA endpoint should bind to. **Note** The relay endpoints must be configured on physical IPs and not on virtual IPs.
Port	Port is the listening port for this instance.
Fqdn	Fully Qualified Domain Name of the DRA end point.
Enabled	Check to enable endpoint.

An example configuration is shown below:

Figure 11: Relay Endpoints - Example Configuration



Policy Routing for Real IPs with Relay Endpoints

vDRA relay links consist of a control plane and a data plane.

The control plane uses virtual IPs and the data plane uses real IPs.

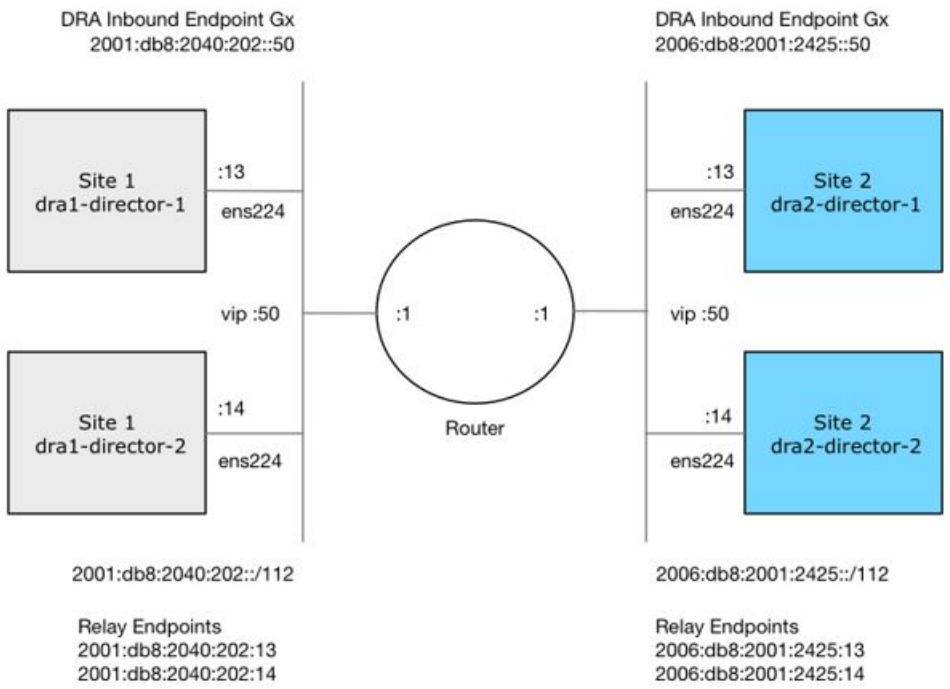
If the control and data plane use the same links, and those links are configured with VIPs, by default, the data plane uses the VIP as its source address for outgoing connections. The data plane uses the VIP as the source address only if the VIP is active on the data plane's outgoing interface.

To avoid this situation, policy routing is used to force the data plane to use the real IP address of the outgoing interface instead of the VIP.

Example of a vDRA Relay Endpoints

In the following example network, only the DRA director VMs and their relay links are displayed. In a real scenario, many more links may exist on the DRA director VMs.

Figure 12: Example of Relay Endpoints



Policy Routing

Linux policy routing includes rules and routing tables. The rules identify traffic and point to a user-defined routing table. The routing table contains customized routes.

To prevent the Relay Link's data plane from using the VIP as a source address, a rule is created to identify the real IP in the destination address and identify the desired routing table.

Configure Policy Routing

The following configuration procedure is performed on Site 1 dra1-director-1. Repeat the procedure for all other dra-directors and modify the IP addresses accordingly.

Perform the following steps on each dra-director VM to configure policy routing:

1. Create a custom routing table
2. Create an IP rule for each remote relay endpoint's real IP address
3. Add a route to the custom routing table that specifies the real IP source address

Set up Custom Routing Table

Set up the custom routing table as shown in the following example:

```
echo "200 dra.relay" | sudo tee --append /etc/iproute2/rt_tables
```

Define IP Rules

The following rules match the packets destined to the real IPs of interface ens224 on dra2-director1 and dra2-director2:

```
ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
```

Define the Route

The following example of the route uses the router's interface as the next hop and specifies ens224's real IP address as the source address for outgoing packets.

```
ip route add 2006:db8:2001:2425::/112 via
2001:db8:2040:202::1 src 2001:db8:2040:202::13 table dra.relay
```

Validate the Routing

Use the following example commands to validate the route selection for remote relay real IP and VIP addresses.

```
ip -6 route show table dra.relay
ip -6 route get 2006:db8:2001:2425::13
ip -6 route get 2006:db8:2001:2425::14
ip -6 route get 2006:db8:2001:2425::50
```

Persistent Configuration

In order for the Policy Routing configuration to survive a reboot, add the configuration commands to `/etc/network/interfaces` under interface ens224 as shown below:

```
auto ens224
iface ens224 inet static
address 192.169.22.13
netmask 255.255.255.0
iface ens224 inet6 static
address 2001:db8:2040:202::13
netmask 112
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
up ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
up ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src 2001:
db8:2040:202::13 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
down ip -6 rule del to 2006:db8:2001:2425::13 table dra.relay
down ip -6 rule del to 2006:db8:2001:2425::14 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay
```

Configure Policy Routing with Deployer/Installer

Configure the VM artifacts and the cloud config to set up policy routing using the deployer.

VM Artifacts

Add Policy Route configuration to the DRA director VM's interfaces.esxi file as shown in the following example:

```
cps@installer:/data/deployer/envs/dra-vnf/vms/dra-director
/dra-director-1$ cat interfaces.esxi
auto lo
iface lo inet loopback

auto ens160
```

```
iface ens160 inet static
address 10.81.70.191
netmask 255.255.255.0
gateway 10.81.70.1

auto ens192
iface ens192 inet static
address 192.169.21.13
netmask 255.255.255.0

auto ens224
iface ens224 inet static
address 192.169.22.13
netmask 255.255.255.0
iface ens224 inet6 static
address 2001:db8:2040:202::13
netmask 112
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
up ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
up ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
down ip -6 rule del to 2006:db8:2001:2425::13 table dra.relay
down ip -6 rule del to 2006:db8:2001:2425::14 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay

auto ens256
iface ens256 inet static
address 192.169.23.13
netmask 255.255.255.0
cps@installer:/data/deployer/envs/dra-vnf/vms/dra-director/dra-director-1$
```

Cloud Config

Create the dra.relay routing table on the dra-directors by adding the following bootcmd: to user_data.yml and storing the file at `/data/deployer/envs/dra-vnf/vms/dra-director/user_data.yml`. The sed command prevents adding a routing table every time the VM boots.

```
bootcmd:
 - "sed -i -e '/^200 *dra.relay/d' /etc/iproute2/rt_tables"
 - "sh -c \"echo '200     dra.relay' >> /etc/iproute2/rt_tables\""
```

Example of user_data.yml:

```
#cloud-config
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2gJmlMWcQ8wtejgAbi
XtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYAkZhZWX/zON9raavU8lgD9+YcRopWUtujIC71YjtoxIjWIBBbrtqt
PlUXMUXQsi91RQbUtslENP+tSatS3awoQupyBMMSutyBady/7Wq0UTwFsnYs5Jfs8jIQuMfVQ9uJ4mNn7wJ0
N+Iaf27rE0t3oiY5DRN6j07WhauM6lCnZ1JDlzqmTnTHQkgJ3uKmQa5x73tJ1OW89Whf+R+dfslVn/yUwK/
vf4extHTn32Dtsxkjz7kQeEDgCe/y7owimaEFcCIfEWEaj/50jegN cps@root-public-key

resize_rootfs: true

write_files:
```

```
  - path: /root/swarm.json
    content: |
     {
        "role": "{{ ROLE }}",
        "identifier": "{{ IDENTIFIER }}",
        "master": "{{ MASTER_IP }}",
        "network": "{{ INTERNAL_NETWORK }}",
        {% if WEAVE_PASSWORD is defined %}"weavePw": "{{ WEAVE_PASSWORD }}",
        {% endif %}
        "zing": "{{ RUN_ZING | default(1) }}",
        "cluster_id": "{{ CLUSTER_ID }}",
        "system_id": "{{ SYSTEM_ID }}"
     }
    owner: root:root
    permissions: '0644'
  - path: /home/cps/.bash_aliases
    encoding: text/plain
    content: |
      # A convenient shortcut to get to the Orchestrator CLI
      alias cli="ssh -p 2024 admin@localhost"
      alias pem="wget --quiet http://171.70.34.121/microservices/latest/cps.pem ;
      chmod 400
cps.pem ; echo 'Retrieved \"cps.pem\" key file'"
    owner: cps:cps
    permissions: '0644'
  - path: /etc/pam.d/common-password
    content: |
     #
     # /etc/pam.d/common-password - password-related modules common to all services
     #
     # This file is included from other service-specific PAM config files,
     # and should contain a list of modules that define the services to be
     # used to change user passwords.  The default is pam_unix.

     # Explanation of pam_unix options:
     #
     # The "sha512" option enables salted SHA512 passwords.  Without this option,
     # the default is Unix crypt.  Prior releases used the option "md5".
     #
     # The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
     # login.defs.
     #
     # See the pam_unix manpage for other options.

     # As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
     # To take advantage of this, it is recommended that you configure any
     # local modules either before or after the default block, and use
     # pam-auth-update to manage selection of other modules.  See
     # pam-auth-update(8) for details.

     # here are the per-package modules (the "Primary" block)
     password   requisite                      pam_pwquality.so retry=3 minlen=8
     minclass=2
     password   [success=2 default=ignore]      pam_unix.so obscure use_authtok
     try_first_pass sha512 remember=5
     password   sufficient                     pam_sss.so use_authtok
     # here's the fallback if no module succeeds
     password   requisite                      pam_deny.so
     # prime the stack with a positive return value if there isn't one already;
     # this avoids us returning an error just because nothing sets a success code
     # since the modules above will each just jump around
     password   required                       pam_permit.so
     # and here are more per-package modules (the "Additional" block)
     # end of pam-auth-update config
```

```
    owner: root:root
    permissions: '0644'
runcmd:
 - [vmware-toolbox-cmd, timesync, enable ]
bootcmd:
 - "sed -i -e '/^200 *dra.relay/d' /etc/iproute2/rt_tables"
 - "sh -c \"echo '200     dra.relay' >> /etc/iproute2/rt_tables\""
```

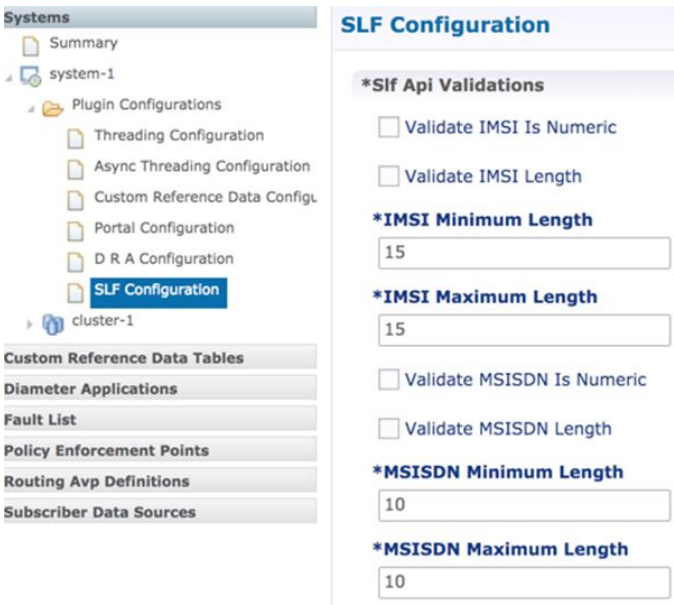
SLF Configuration

You can specify whether the IMSI and MSISDN values are validated in SLF API.

By default, SLF validation is disabled.

To set up SLF validation, create SLF Configuration from the Plugin Configuration in Policy Builder.

Figure 13: SLF Configuration



The following table describes the SLF API validations that you can configure:

Table 12: SLF Configuration

Field	Description
Validate IMSI is Numeric	If checked: IMSI received in the SLF API request must be numeric If unchecked: IMSI numeric validation is not performed on the IMSI received in the SLF API request

Field	Description
Validate IMSI Length	If checked: IMSI length is validated based on the specified IMSI Minimum Length (inclusive) and IMSI Maximum Length (inclusive) If unchecked: IMSI length validation is not performed on the IMSI received in the SLF API request
Validate MSISDN is Numeric	If checked: MSISDN received in the SLF API request must be numeric If unchecked: MSISDN numeric validation is not performed on the MSISDN received in the SLF API request
Validate MSISDN Length	If checked: MSISDN length is validated based on the specified MSISDN Minimum Length (inclusive) and MSISDN Maximum Length (inclusive) If unchecked: MSISDN length validation is not performed on the MSISDN received in the SLF API request

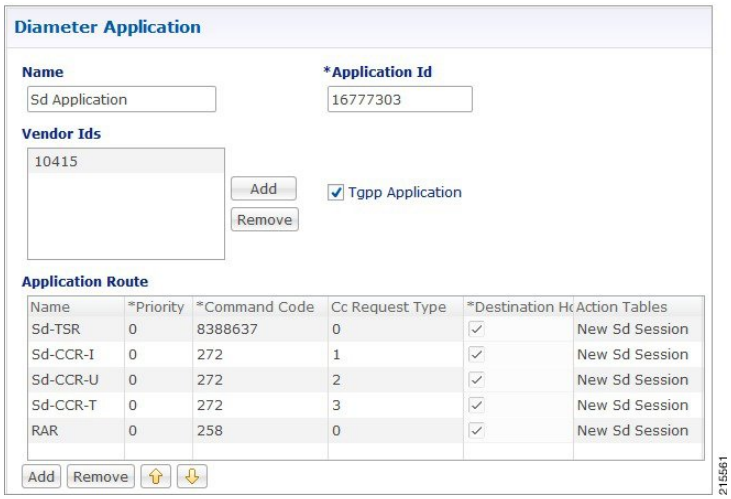
Diameter Application

Sd Application

For Sd, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table, typically, an Sd New Session table for routing Sd TSRs to a peer route. The Sd New Session CD table will choose a peer route based on the Destination-Realm. The peer route will then point to a Peer-Group which contains multiple peer connections to a TDF and the DRA will load balance among the TDF peer connections in the Peer Group.

An example configuration is shown below:

Figure 14: Diameter Application - Sd Application Example



The following parameters are configured under Sd Application:

Table 13: Sd Application Parameters

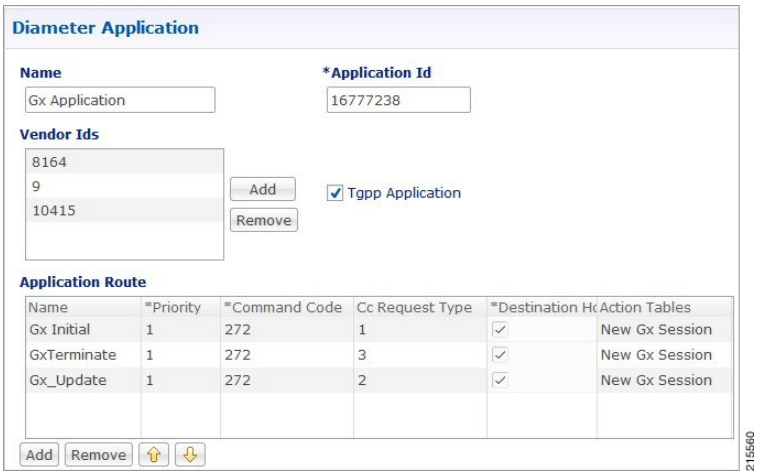
Parameter	Description
Name	Name of the Sd application.
Application Id	16777303, 3GPP specified Application Identifier for Sd interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Sd interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	Indicates if the Credit Control Request type is Initial(1)/Update(2) or Terminate(3).
Destination Host Null	If this check box is selected, indicates if Destination Host will be null in messages received for this application.
Action Tables	Identifies the request routing table for this interface and message.

Gx Application

For Gx, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table. When “Destination Host Null” is checked, it means Destination-Host AVP is null. It will then check for table driven routing.

An example configuration is shown below:

Figure 15: Diameter Application - Gx Application Example



C-DRA attempts to do Dest-Host routing before doing table driven routing. If the Dest-Host AVP is absent, empty, or equal to the CDRA FQDN, then we skip Dest-Host routing altogether and proceed to Table-Driven routing.

The following parameters are configured under Gx Application:

Table 14: Gx Application Parameters

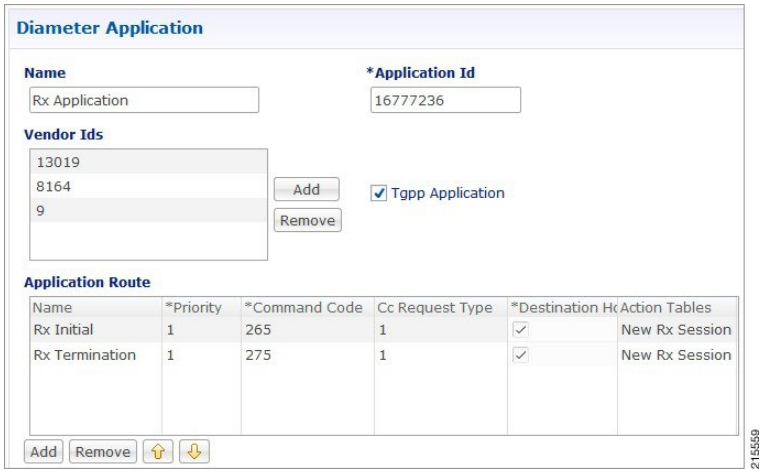
Parameter	Description
Name	Name of the Gx application.
Application Id	16777238, 3GPP specified Application Identifier for Gx interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Gx interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.

Parameter	Description
Cc Request Type	Indicates if the Credit Control Request type is Initial(1)/Update(2) or Terminate(3).
Destination Host Null	If this check box is selected, indicates the message will contain a Destination-Host.
Action Tables	Identifies the request routing table for this interface and message.

Rx Application

Identifies the request routing table for this interface and message.

Figure 16: Diameter Application - Rx Application Example



The following parameters are configured under Rx Application:

Table 15: Rx Application Parameters

Parameter	Description
Name	Name of the Rx application.
Application Id	16777236, 3GPP specified Application Identifier for Rx interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Rx interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.

Parameter	Description
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	Not supported for Rx interface.
Destination Host Null	If this check box is selected, indicates if Destination Host will be null in messages received for this application.
Action Tables	Identifies the request routing table for this interface and message.

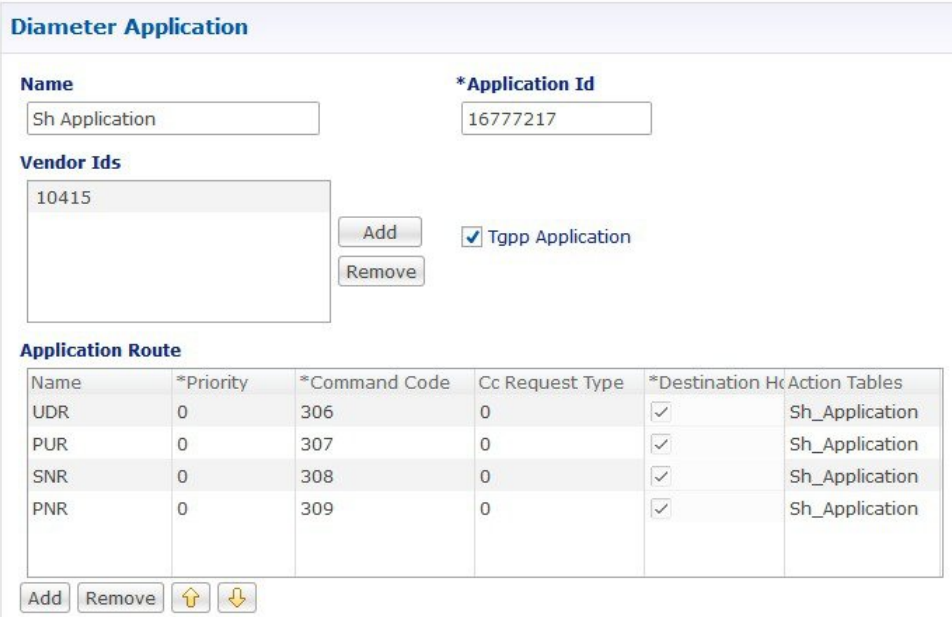
Sh Application

Sh interface is used for communication between AS and HSS for Call data query/Push subscriber profile and subscriber notification procedures.

Note In certain scenarios, the customer might use the Sh interface between PCRF and HSS also.

An example configuration is shown below:

Figure 17: Diameter Application - Sh Application Example



The following parameters are configured under Sh Application:

Table 16: Sh Application Parameters

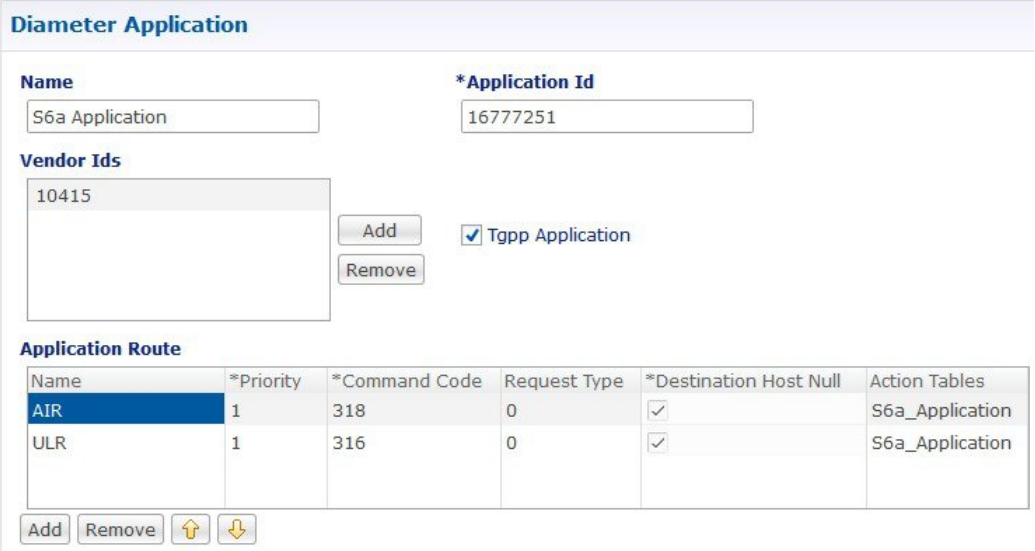
Parameter	Description
Name	Name of the Sh application.
Application Id	16777217, 3GPP specified Application Identifier for Sh interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Sh interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	CC-Request-Type is not applicable for Sh interface.
Destination Host Null	If this check box is selected, indicates the message will contain a Destination-Host.
Action Tables	Identifies the request routing table for this interface and message.

S6a Application

DRA supports S6a interface with the implementation of Subscriber Location Function(SLF) feature. S6a is an interface which supports the mobility management and subscriber data management procedures between MME and HSS in an LTE EPC network.

An example configuration is shown below:

Figure 18: Diameter Application - S6a Application Example



The following parameters are configured under S6a Application:

Table 17: S6a Application Parameters

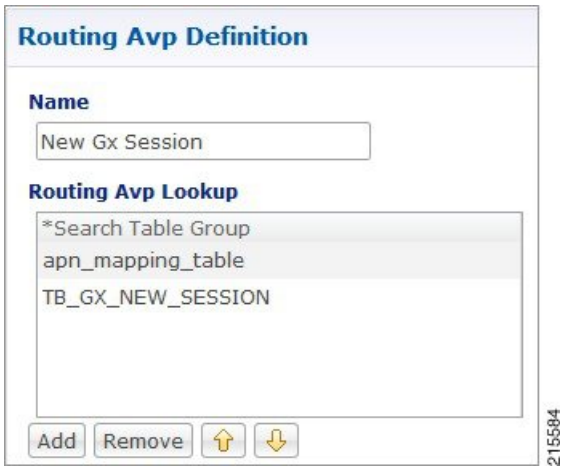
Parameter	Description
Name	Name of the S6a application.
Application Id	16777251, 3GPP specified Application Identifier for S6a interface.
Vendor Ids	Vendor Identifiers that are required to be supported on S6a interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	CC-Request-Type is not applicable for S6a interface.
Destination Host Null	If this check box is selected, indicates the message will contain a Destination-Host.
Action Tables	Identifies the request routing table for this interface and message.

Routing AVP Definition

Gx Session

An example configuration is shown below:

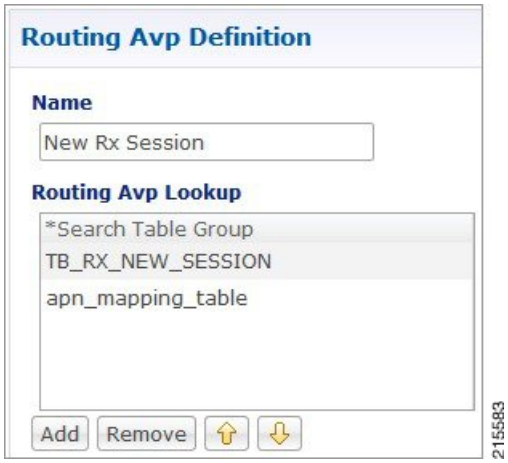
Figure 19: Routing AVP Definition - Gx Session



Rx Session

An example configuration is shown below:

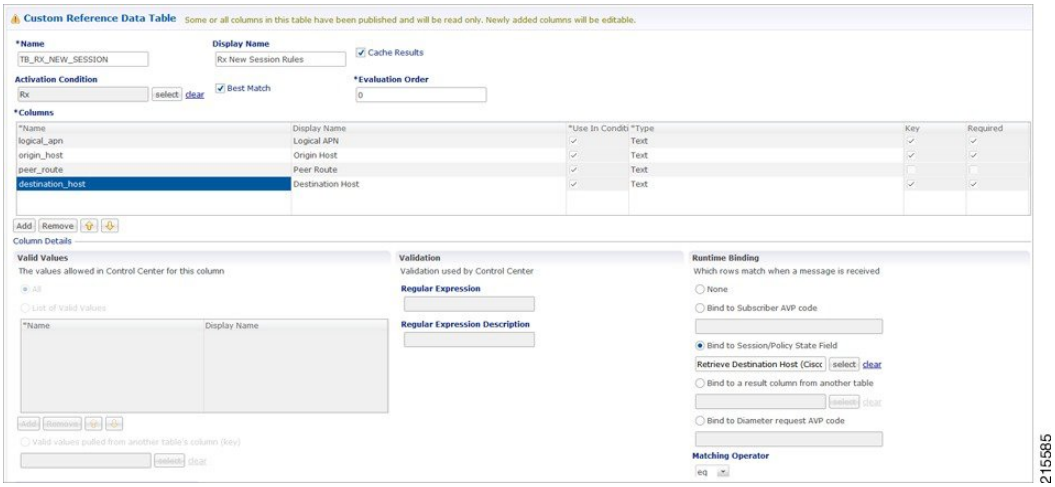
Figure 20: Routing AVP Definition - Rx Session



Rx New Session Rules - CRD Table

An example configuration is shown below:

Figure 21: Rx New Session Rules - CRD Table

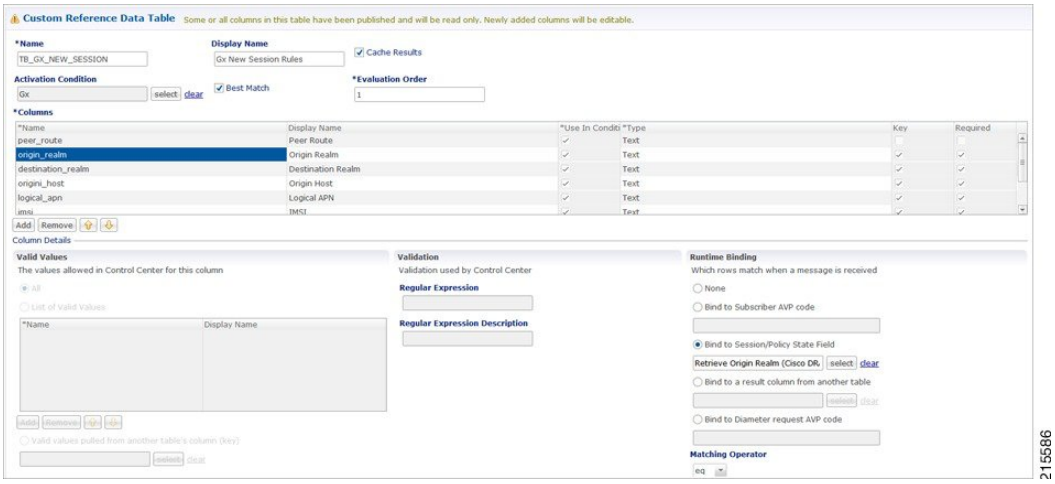


Gx New Session Rules - CRD Table

For Gx, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table, typically, for routing Gx CCR-Is. The Gx CCR-I should be routed based on a logical APN and the Origin-Host attribute. Regular expression matching of logical APNs and Origin-Hosts can also be configured. The implementation should be flexible to allow CRDs to be configured for routing of other attributes such as Destination-Realm and Origin-Realm.

An example configuration is shown below:

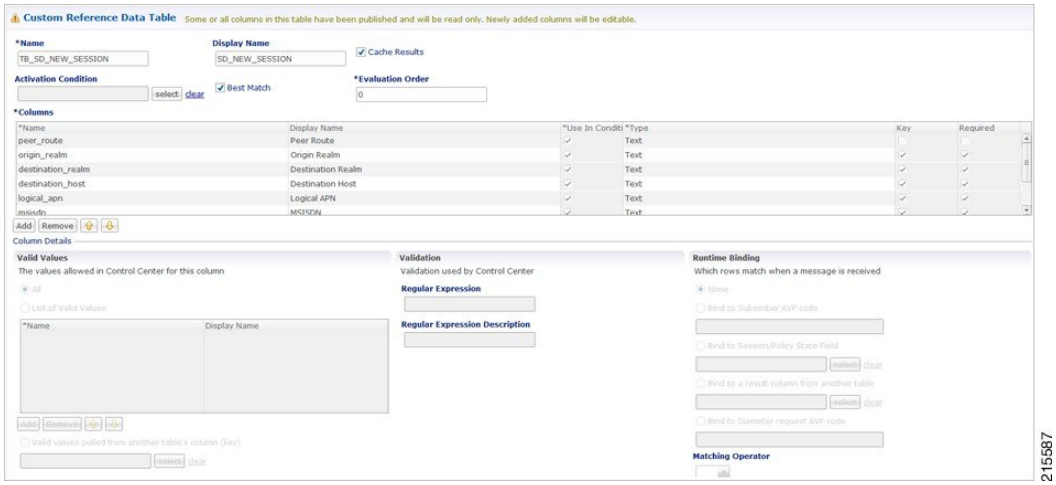
Figure 22: Gx New Session Rules - CRD Table



Sd New Session Rules - CRD Table

An example configuration is shown below:

Figure 23: Sd New Session Rules - CRD Table

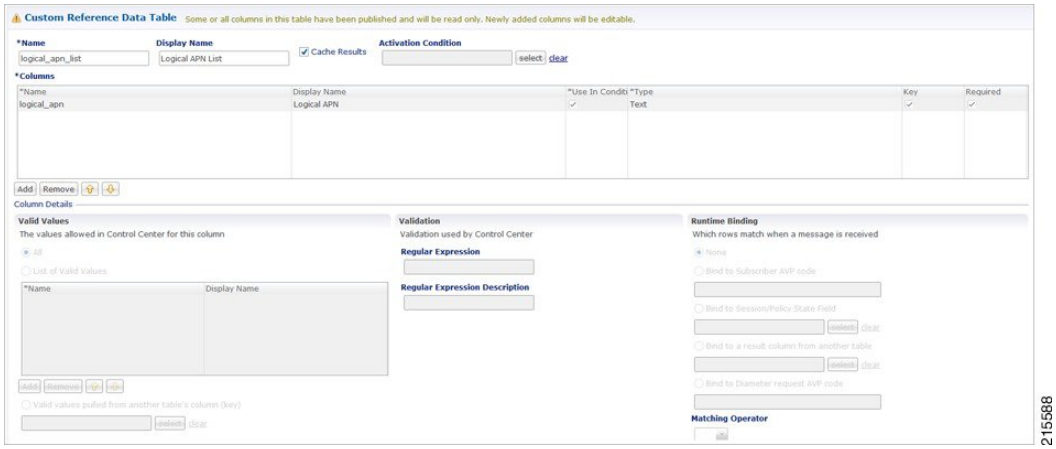


215587

Logical APN List - CRD Table

An example configuration is shown below:

Figure 24: Logical APN List - CRD Table



215588

Dynamic AVP Retriever for Routing

DRA supports routing messages based on the following AVPs from request message:

- Destination-Host
- Destination-Realm
- Origin-Host
- Origin-Realm
- APN (from Called-Station-ID)

- IMSI (from Subscription-ID)
- MSISDN (from Subscription-ID)

Regular-expression matching and combinations of AVPs is supported. This requirement is not applicable across all messages on different interfaces. The following table shows applicability of the AVP's at a message and interface level.

Table 18: Regular-expression Matching and Combinations of AVPs

Interface	Message	Origin Host	Origin Realm	Destination Host	Destination Realm	APN (Called-Station-ID)	IMSI	MSISDN
Gx	CCR-I	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	CCR-U	No	No	No	No	No	No	No
	RAR	No	No	Yes	No	No	No	No
Sd	TSR	Yes	Yes	Yes	Yes	No	No	No
	CCR-I	Yes	Yes	Yes	Yes	No	No	No
	CCR-U/T	No	No	Yes	No	No	No	No
	RAR	No	No	Yes	No	No	No	No
Rx	RAR	No	No	Yes	No	No	No	No

Dynamic AVP Retrievers are used mostly used in Custom Reference Data where data has to be fetched from messages at runtime.

Configure Dynamic AVP Retriever

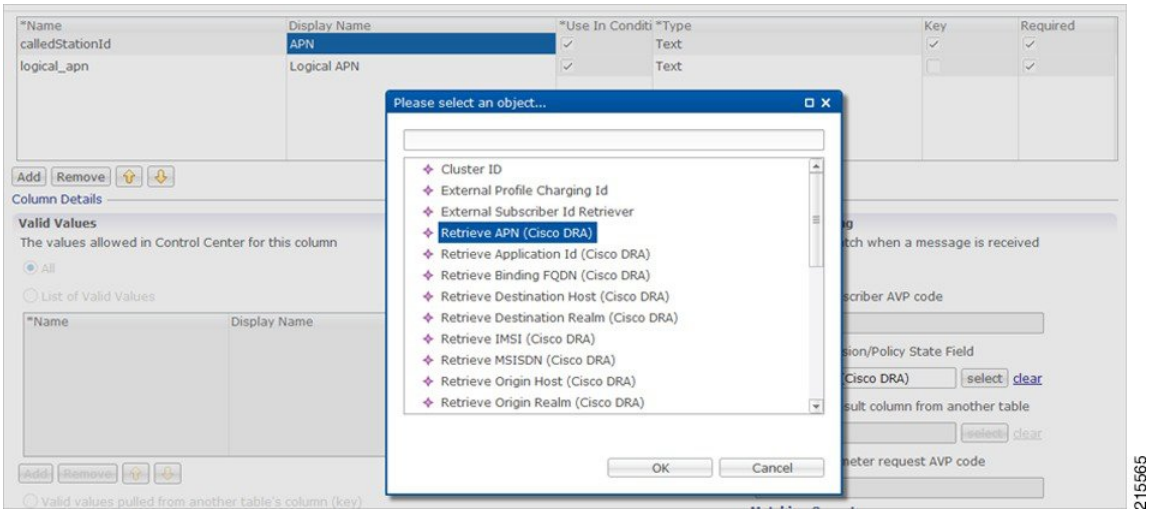
The following sample configuration shows how to retrieve the AVP and bind it to a Key Column in the CRD.

Step 1 Select the column name from the **Columns** table and click **select** near **Bind to Session/Policy State Field** to open the **Please select an object...** dialog box.

Note You can use **Bind to Session/Policy State Field** only for those columns in the **Columns** table where **Key** column has been selected.

Step 2 Select the required object from the dialog box and click **OK**.

Figure 25: Adding AVPs



Step 3 Repeat these steps to add additional AVPs.

Custom Reference Data Tables

Search Table Groups

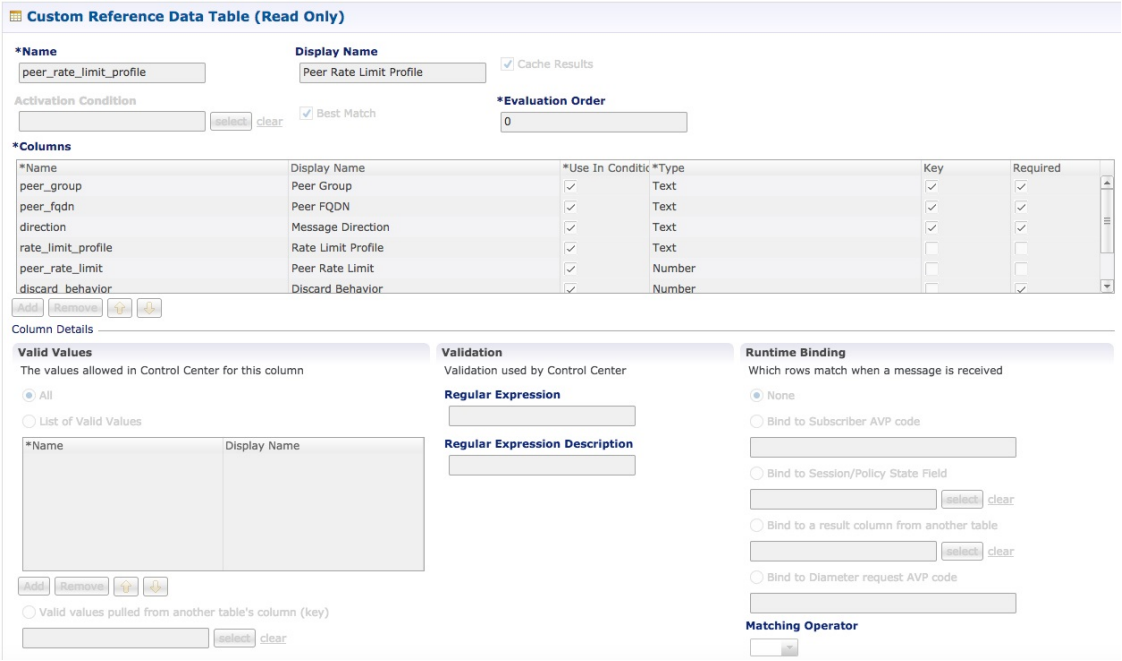
Peer Rate Limit Profile

This is a Search Table Group whose key columns are Peer Group, Peer FQDN or Origin Host in the message and Message Direction.

Using this search table group, the user can configure a maximum rate for each of the configured and defined diameter peers. It also allows the user to configure a maximum rate for each server process.

The peer rate limit is shown below:

Figure 26: Peer Rate Limit - STG



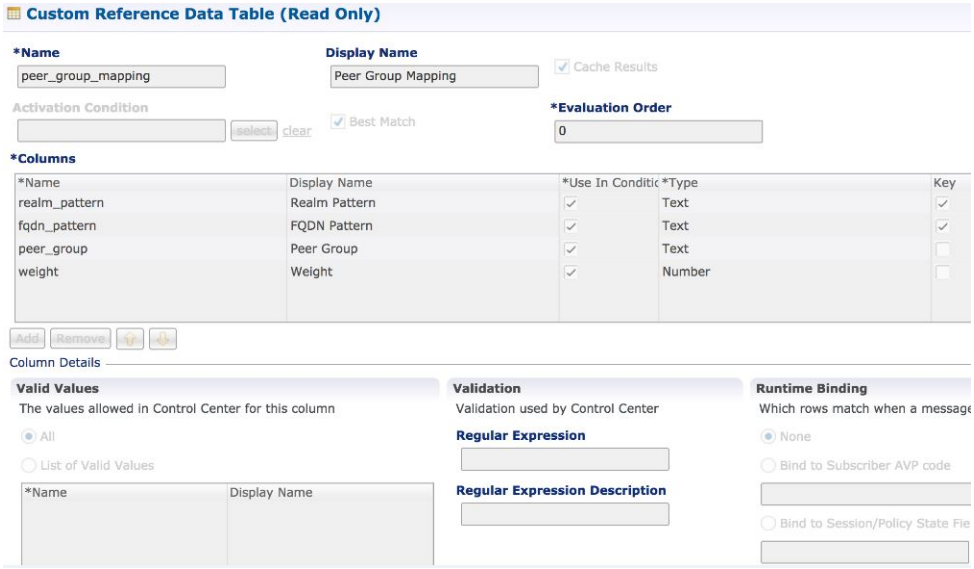
- Peer Group: This is the group of peers classified together using Peer Group and Peer Group Peer values initiating the message.
- Peer FQDN: The origin host of the peer. A specific diameter peer with its Fully Qualified Domain Name can be specified in this field or use wildcards specified by * in this field for any peer or matching peers like hss*.
- Direction: Message direction (Ingress and Egress).
  - Ingress: Any diameter messages received by CPS vDRA from diameter peer. The routing decision by CPS vDRA will be taken after the ingress side rate limiting has been applied.
  - Egress: Any diameter messages forwarded/routed by CPS vDRA to diameter peer. The egress side rate limiting will be applied after the routing decision has been taken by CPS vDRA.
- Peer Rate Limit: This field is to specify the threshold in TPS above which the diameter messages are discarded. This can be left empty if none of the messages are to be dropped or only message level rate limit is to be applied.
- Rate Limit Profile: Profile Name applicable for this Peer Group and Peer, if specified. This profile maps to Rate Limiting at message level. This field enables the rate limit at per message/command code level. See Message Rate Limit Profile for more details.
- Rate Limit Result Code: The result code sent by CPS vDRA for response message towards diameter peer when Discard Behavior is configured as Send Error Answer. In case Discard Behavior is configured as Drop Message, this field is ignored.
- Error String: The string specified in this field is populated by CPS vDRA in AVP Error Message for response message towards diameter peer when Discard Behavior is configured as Send Error Answer. In case Discard Behavior is configured as Drop Message, this field is ignored. This is an optional field when Discard Behavior is configured as Send Error Answer.

Note If both Rate Limit Error Code and Rate Limit Error String are provided along with Rate Limit Action as "Drop Message", the Rate Limit Action takes precedence and the other two fields will be ignored.

For more information, see Peer Rate Limit Profile.

Peer Group Mapping

Figure 27: Peer Group Mapping - STG

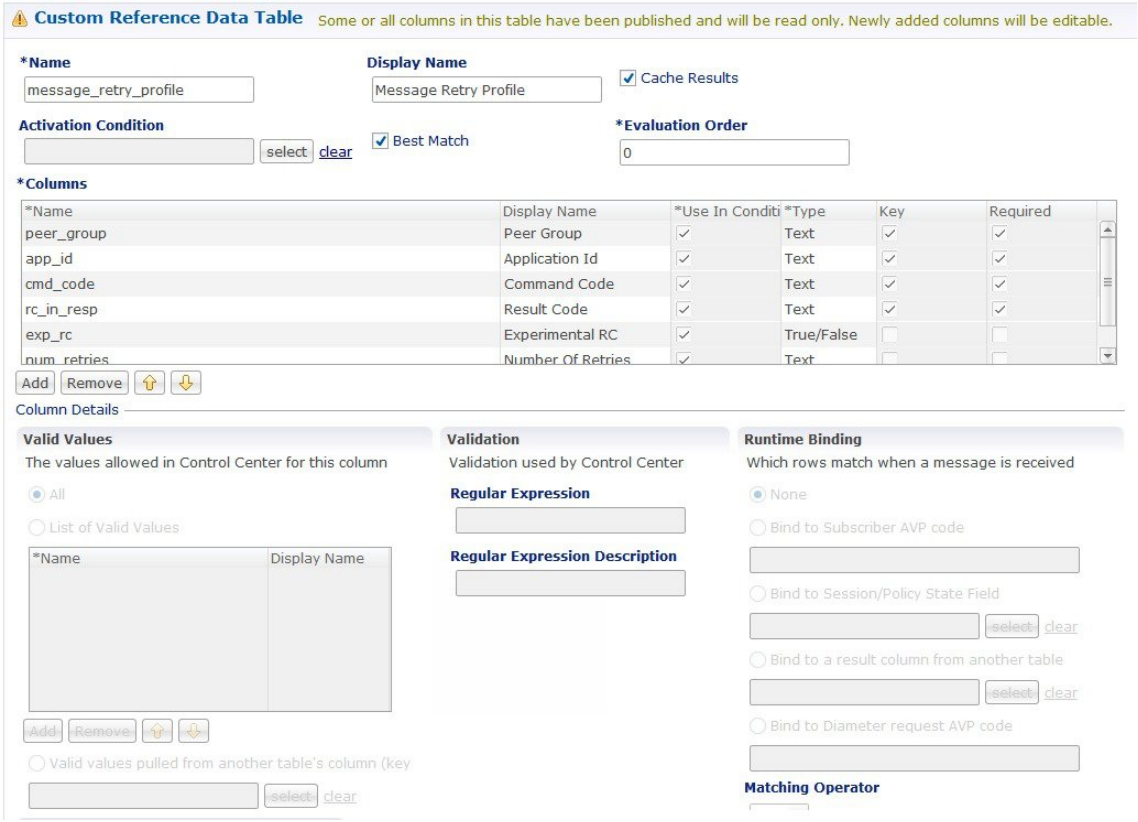


For more information, see Peer Group Mapping.

Message Retry Profile

Message retry profile has been added.

Figure 28: Message Retry Profile - STG



- Peer Group: Peer group for which the retry has to be happen.
- Application Id: Application Id of the diameter applications.
- Command Code: Command Code of the message.
- Result Code: Result code received from PCRF for timeout. The value is 7000.
- Experimental RC: Indicates whether result code is experimental or not. This is for future purpose and value in this has no effect on the message retry functionality.
- Number of Retries: Number of retries for the message.

For more information, see Message Retry Profile.

Message Mediation Profile

The message mediation profile is used to provide support for mediation of AVPs in Diameter request and answer.

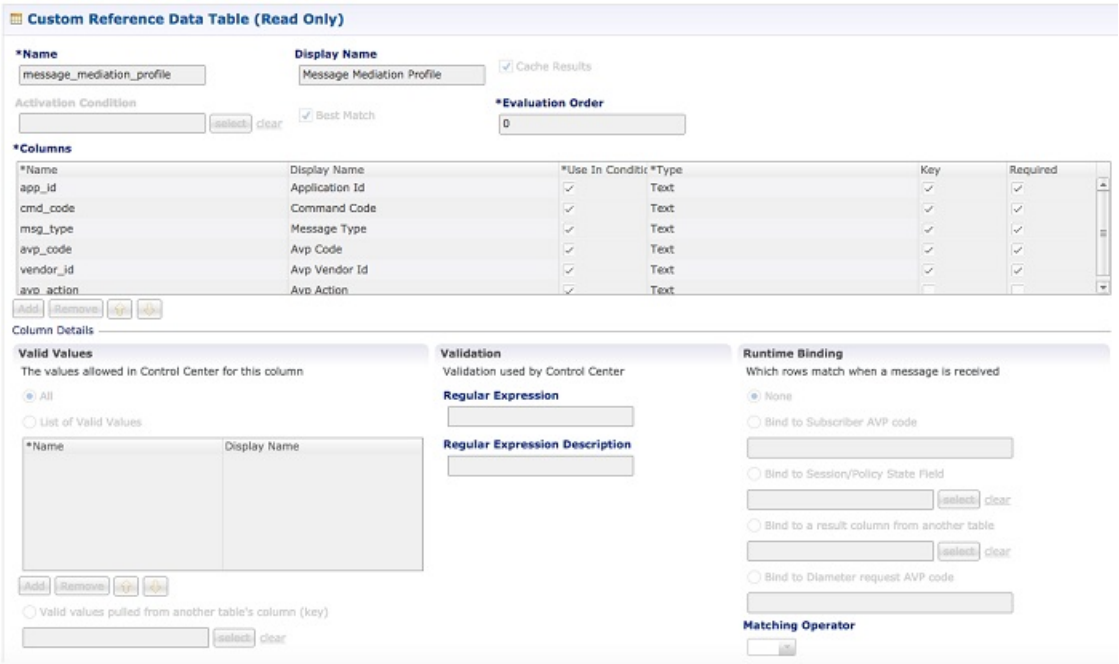
- For Diameter requests, only remove is supported.
- For Diameter answers, the following actions are supported:
  - "remove" meaning remove all matching AVPs in the request.
  - "copy" meaning copy from the request if no AVPs are present in the answer.

- If the AVP is present in answer, no action is performed.

- "overwrite" meaning first remove and then copy from the request.
  - Check if the AVP is present in answer, if so remove and add from request.
  - If AVP is not present in answer, copy from request.

A new **Message Mediation Profile** STG has been added:

Figure 29: Message Mediation Profile - STG



- Application Id: Application ID of the Diameter applications.
- Command Code: Command code of the message.
- Message Type : Request/Answer for which the rule has to be applied.
- Avp Code : AVP code of the Diameter message.
- Vendor Id : AVP vendor ID.
- Avp Action : Provides options for copy/remove/overwrite.

Note Application ID, Command Code, AVP Code and Vendor Id are used as key, so no duplicate rows could be defined for this combination and the same AVP action. For example, you cannot define both "remove" and "Copy from request" for the same set of Application ID, Command Code, AVP Code and Vendor Id.

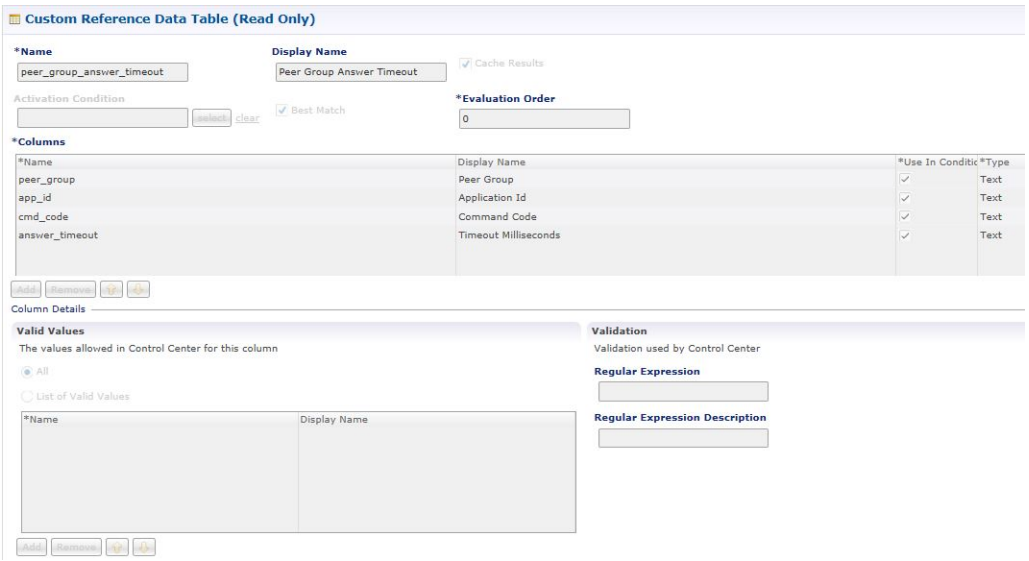
Best Match check box needs to be checked if you want to use the wildcard feature.

For more information, see Message Mediation Profile in Custom Reference Data Tables chapter.

Peer Group Answer Timeout

New search table Peer Group Answer Timeout has been added.

Figure 30: Peer Group Answer Timeout - STG



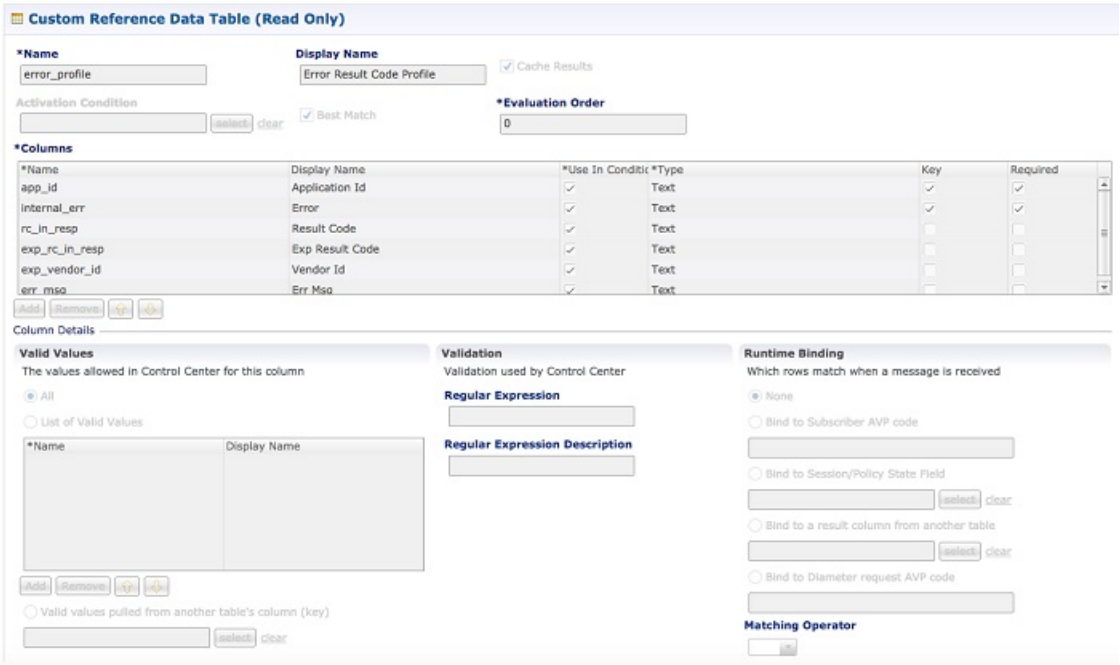
- Application Id: Application Id of the diameter applications.
- Peer Group: Peer group for which the timeout is applied.
- Command code (to enable different timeouts for different Diameter commands)
- Timeout: Timeout in milliseconds.

For more information, see Peer Group Answer Timeout.

Error Result Code Profile

Error result code profile can be used to map errors to Result-Code value and an error message string for the Error-Message AVP. It also provides support for configurable error result codes.

Figure 31: Error Result Code Profile - STG



Valid values is the place where all the valid error values can be configured in STG so that they are visible in CRD drop-down.

- ApplicationId: Application ID for which the mapping of Result-Code has to be done.
- Error: Internal error list.
- ResultCode: Result Code to be sent in answer.
- ExpResultCode: Experimental result code to be sent in answer. Vendor-Id will be sent in Answer only for Experimental result-Code.
- ErrMsg: Error message AVP sent in answer.

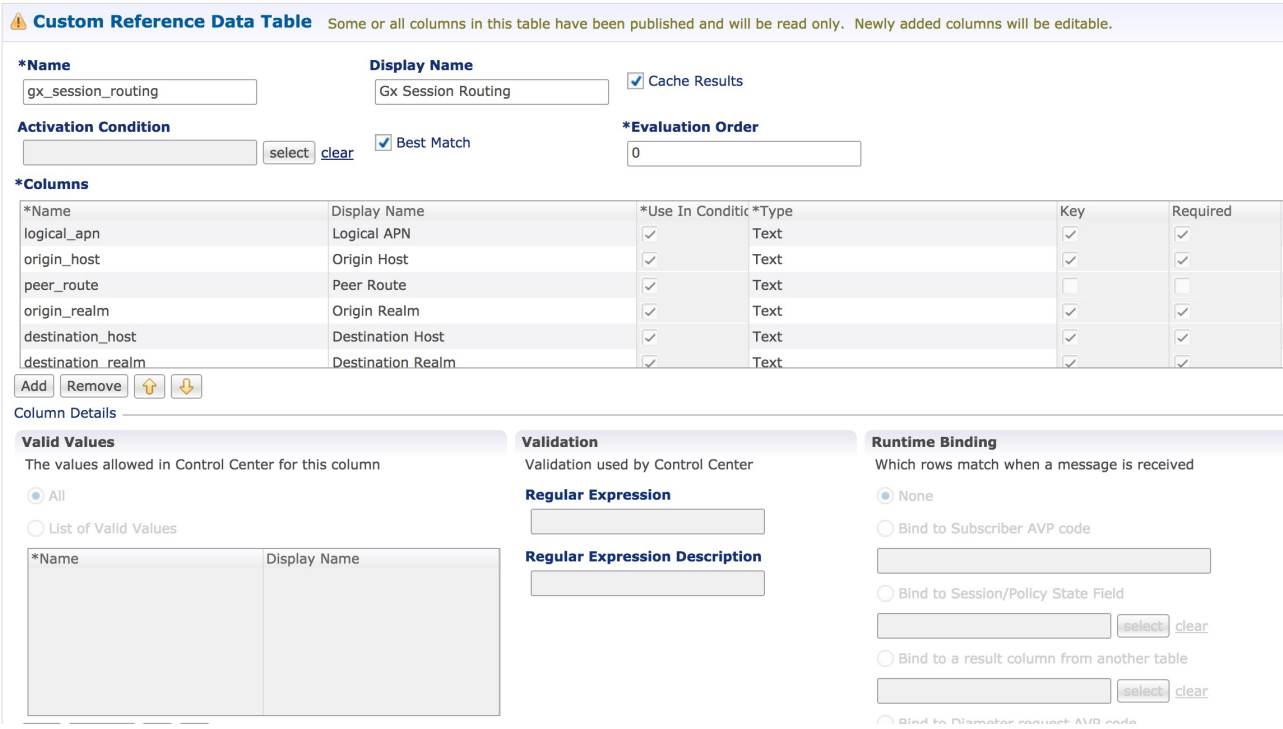
Note Experiment result code will be sent when Result-Code is not configured. If both Result-Code and experimental Result-Code are present, Result-Code would take precedence.

For more information, see Error Result Code Profile.

Gx Session Routing

Gx Session Routing table is required for "table driven routing". Here an example for Gx New Session Rules is provided. If table driven routing is required for Rx or Sd, user needs to create similar tables for Sd and Rx as well.

Figure 32: Gx Session Routing



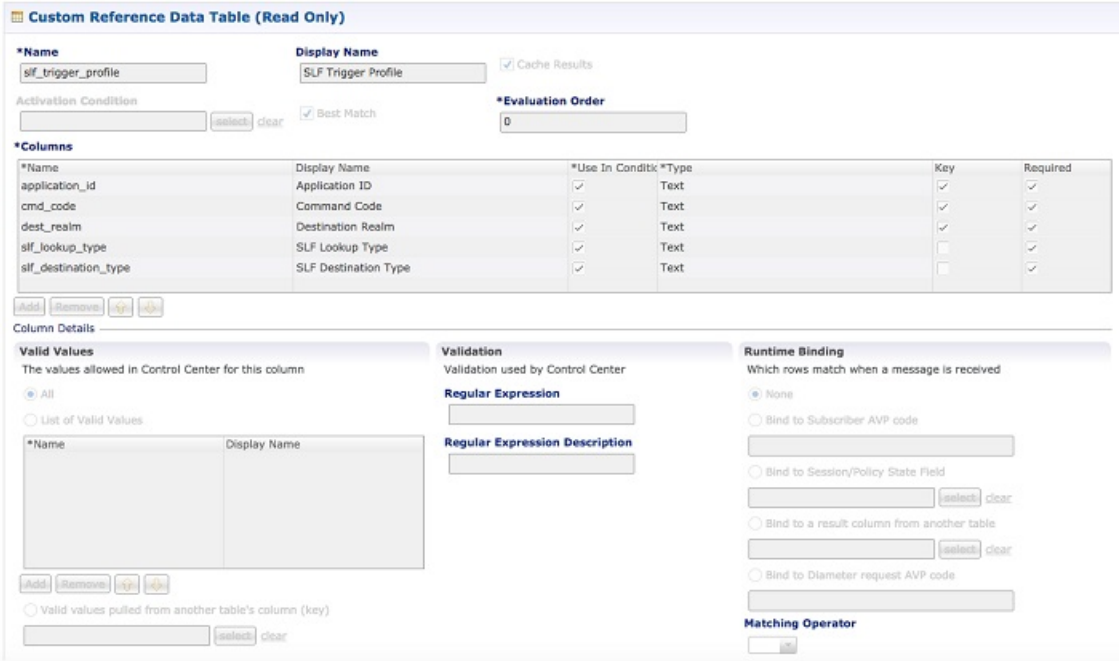
For more information, see Gx New Session Rules.

SLF Trigger Profile

This table is used to derive SLF destination type and SLF lookup type. Keys used for this table are: Application Id, cmd_code, and dest_realm. Output of this table are slf_lookup_type and slf_destination_type.

An example configuration is given.

Figure 33: SLF Trigger Profile - STG

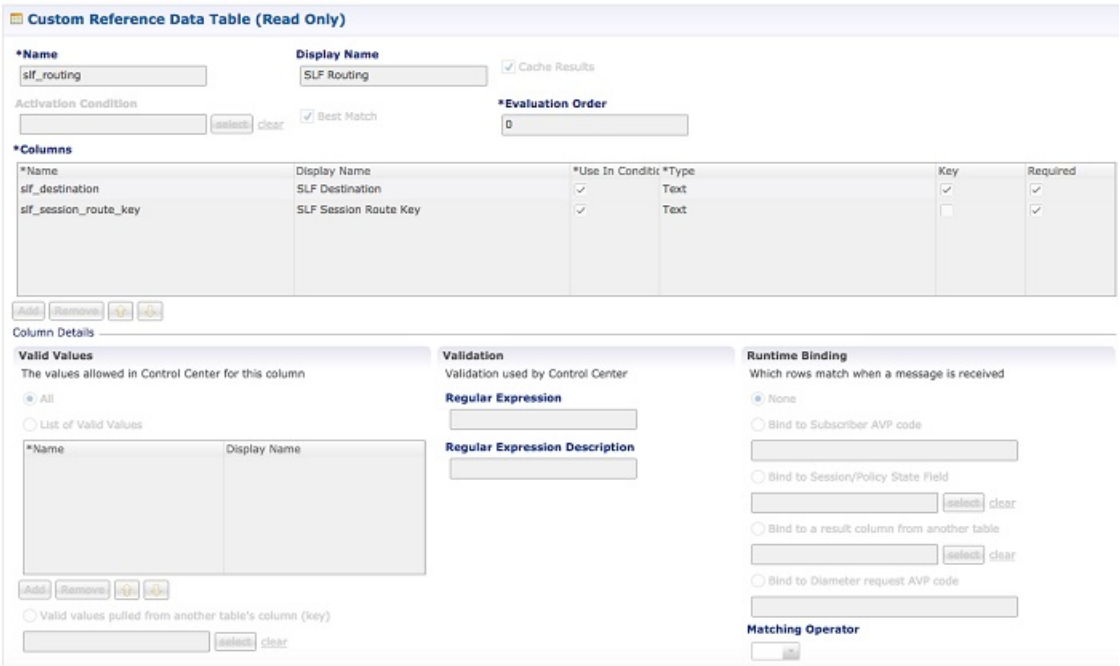


For more information, see SLF Trigger Profile.

SLF Routing

This table is used to derive SLF session route key from SLF Destination. An example configuration is given.

Figure 34: SLF Routing - STG



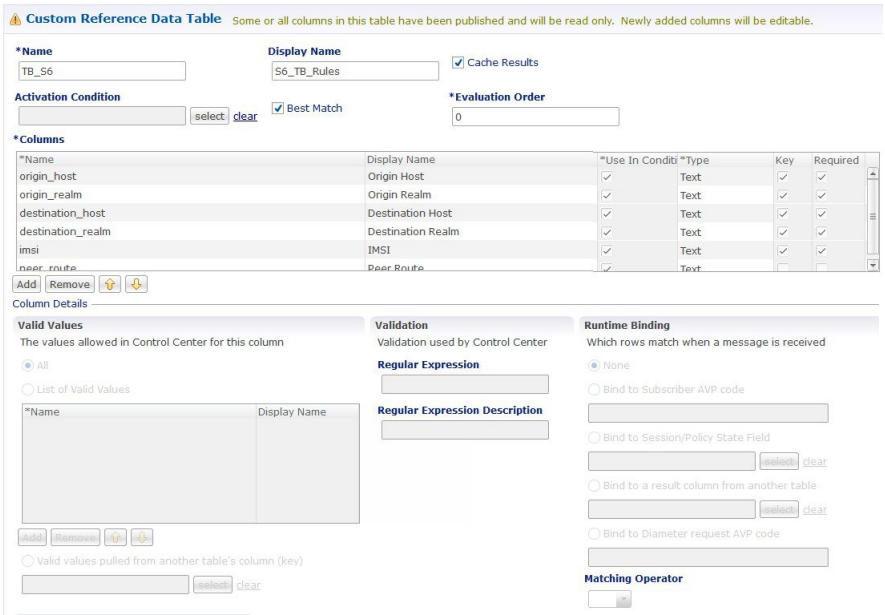
For more information, see SLF Routing.

S6/Sh Table Driven Rules

This table is used for the table driven routing of S6/Sh messages. Fields origin_host, origin_realm, dest_realm, dest_host, msisdn, imsi are used as keys to derive the peer_route.

An example configuration is given.

Figure 35: S6 Table Driven Rules - STG



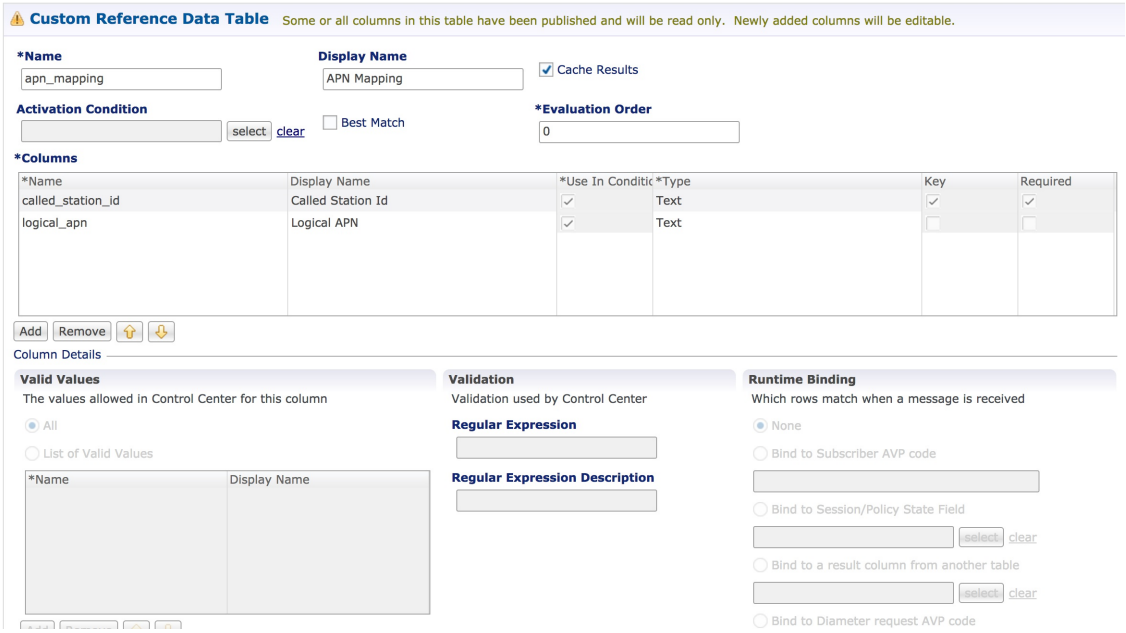
For more information, see S6/Sh Table Driven Rules.

Custom Reference Data Tables

APN Mapping

This table provides information related to APN Mapping. The read-only APN Mapping are shown below:

Figure 36: APN Mapping - CRD Table



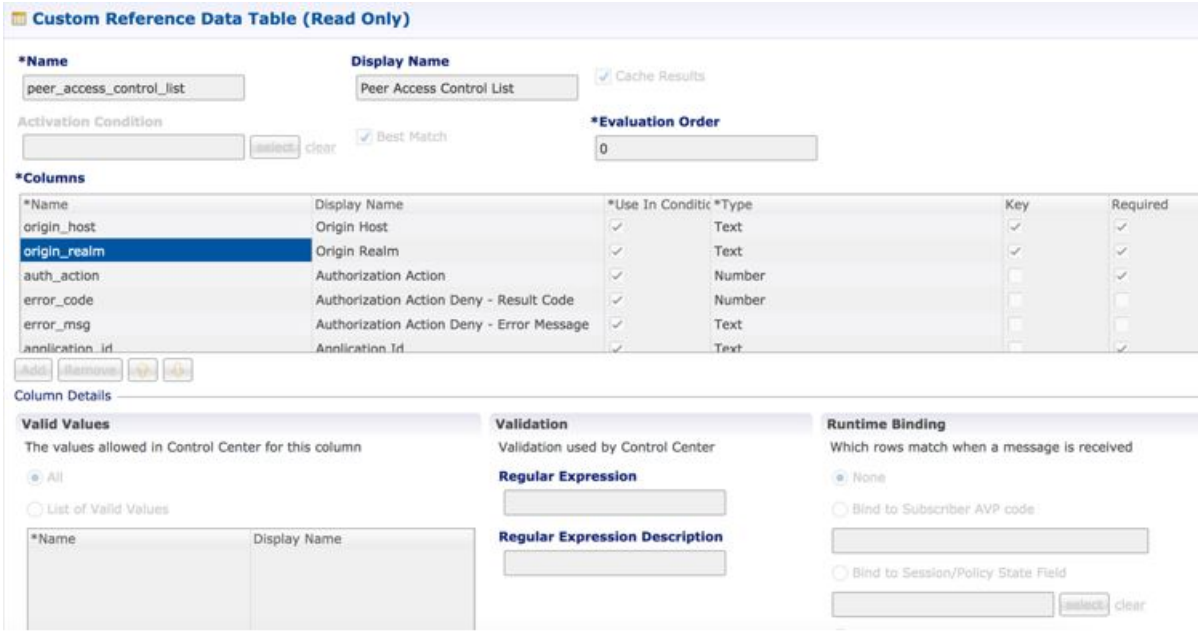
- Called-Station-Id: This is the AVP from which APN is derived. This also is the key column for this table. It is bound to the session or Policy State field as shown in the snapshot.
- Logical_APN: This is the mapped logical name that is used for referencing and processing the message within the system.

Note For sample data configuration, refer the *CPS Control Center Interface Guide for Full Privilege Administrators* for this release.

Peer Access Control List

You can use the Peer Access Control List to specify the list of peers (by realm, FQDN, and applications) that can establish peer connections to vDRA so that unknown peers are not permitted to create Diameter peer connections.

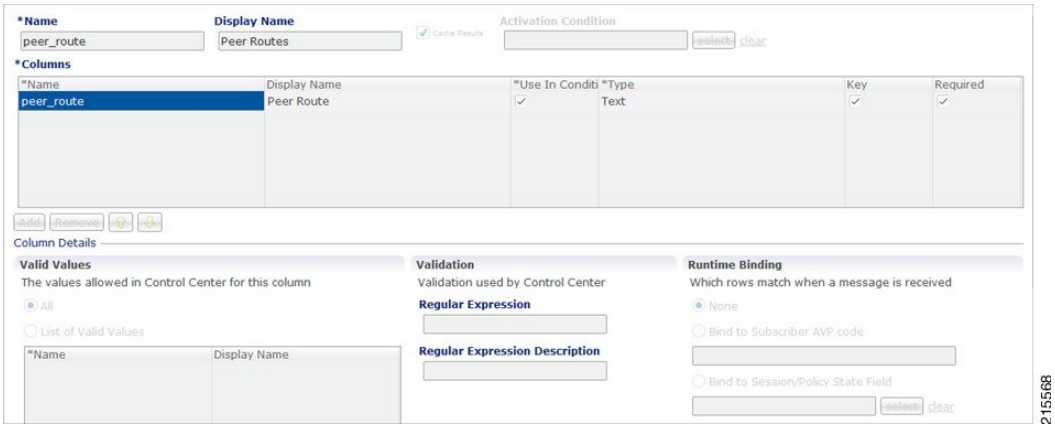
Figure 37: Peer Access Control List



Peer Routes

This tables provides the information related to Peer Routes available in the system. The read-only peer routes are shown below:

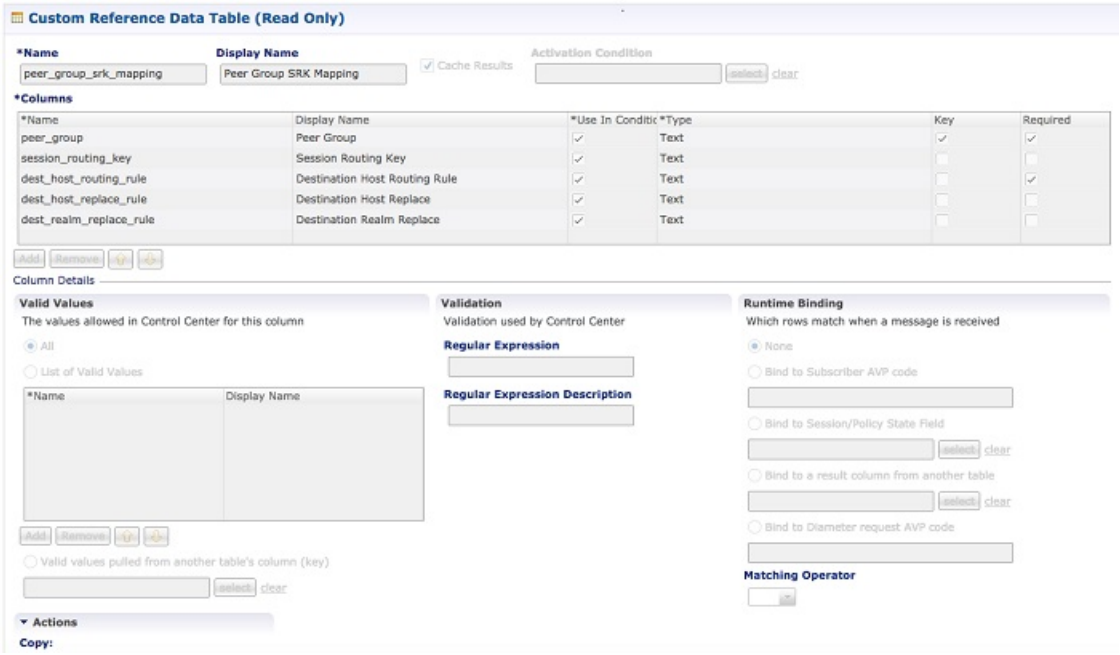
Figure 38: Peer Routes - CRD Table



Peer Group SRK Mapping

This table provides the information related to Peer Groups in the system. The read-only peer groups are shown below:

Figure 39: Peer Group - CRD Table

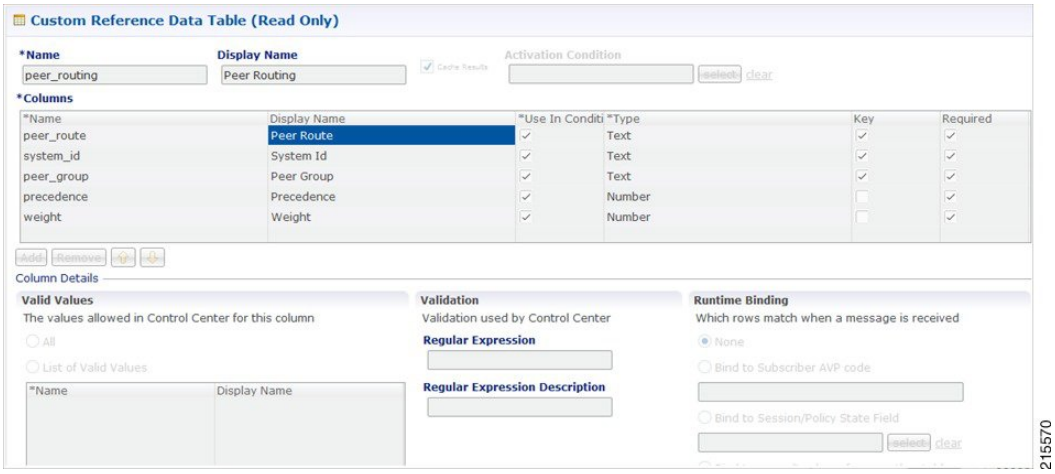


- Peer Group: Name of the peer group.
- Session Routing Key: Routing token for this Peer Group.
- Destination Host Routing Rule: Defines Routing behavior of this group.

Peer Routing

This table provides the information related to peer routing in the system. The read-only peer routings are shown below:

Figure 40: Peer Routing - CRD Table



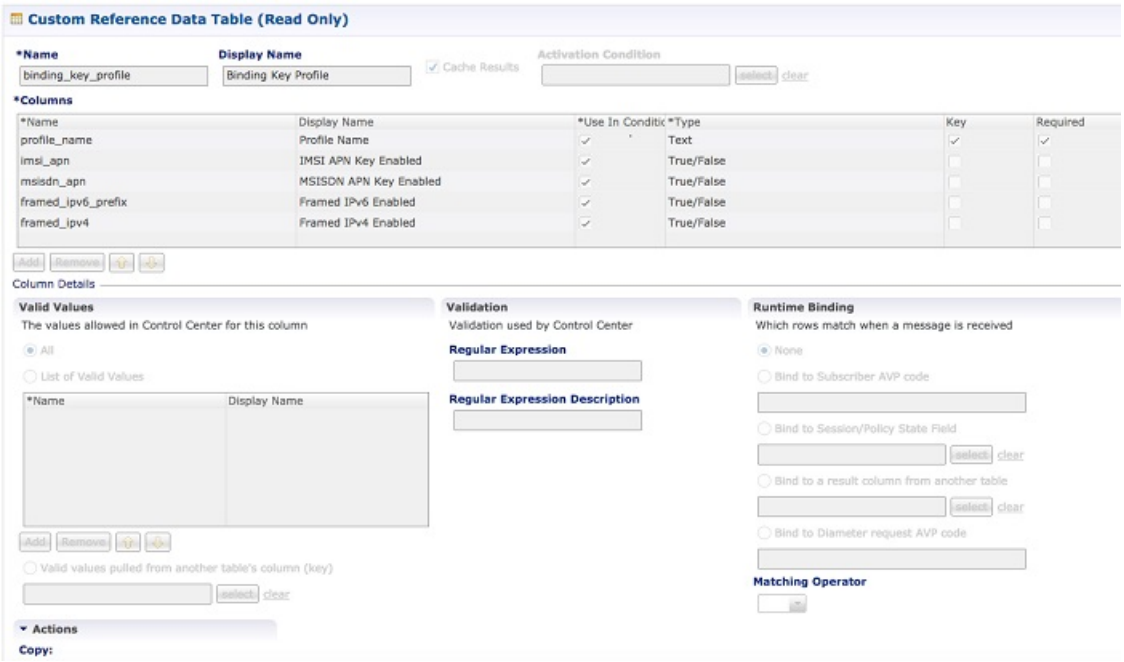
- Peer Route: Identifier of this Peer Route.

- System ID: System Identifier for this VM.
- Peer Group: Identifier of the Peer group on this peer Route.
- Precedence: of the peer group on this Peer Route.
- Weight: Weight of the peer group on this Peer Route.

Binding Key Profile

This table provides the information related to binding key profile in the system. The read-only keys are shown below:

Figure 41: Binding Key Profile - CRD Table



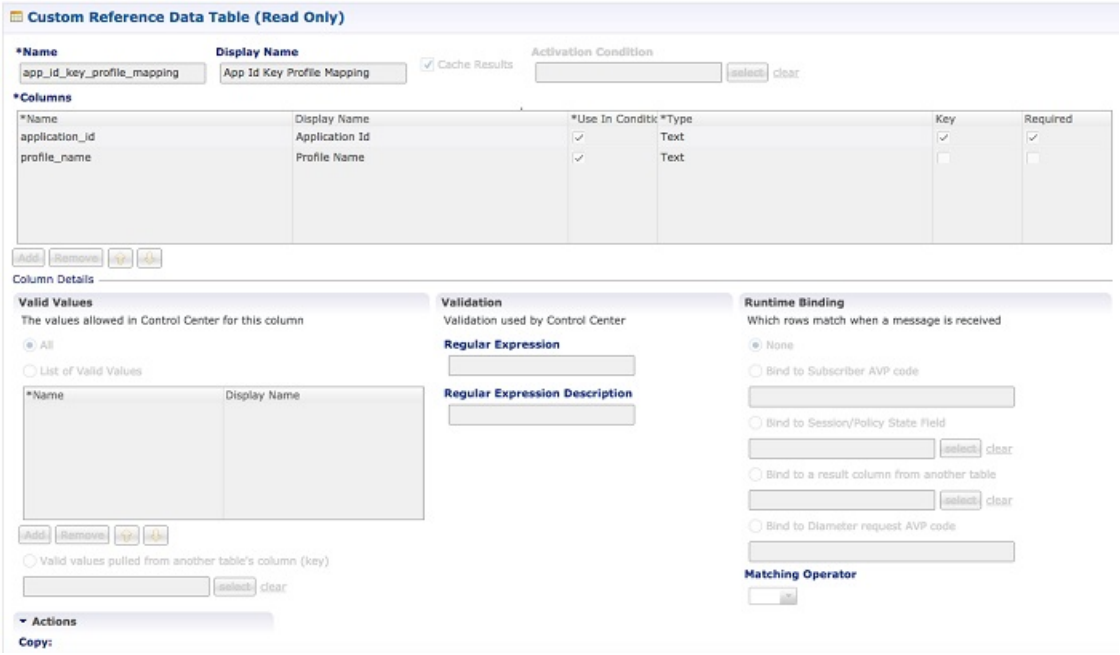
- Profile Name: This is the name given to the Bind profile that is associated with keys that are either enabled and/or disabled.
- MSI APN Key Enabled: Enabling this field would mean that bindings will be stored in IMSI APN collections in bindings database.
- MSISDN APN Key Enabled: Enabling this field would mean that bindings will be stored in MSISDN APN collections in bindings database.
- Framed IPv6 Enabled: Enabling this would mean binding data would be stored in “ipv6bindings” collection.
- Framed IPv4 Enabled: Enabling this would mean binding data getting stored in “ipv4bindings” collection.

Refer to Binding Key Profile for configuration in Control Center.

AppId Key Profile Mapping

This table stores the mapping between Application Identifiers and Bind Key Profile Names. The Application Identifiers are pre-provisioned for two Application Identifiers as Gx and Rx. Similarly, the BindingKeyProfile is also tied to the Profile Name column of the “BindingKeyType_Profile” table:

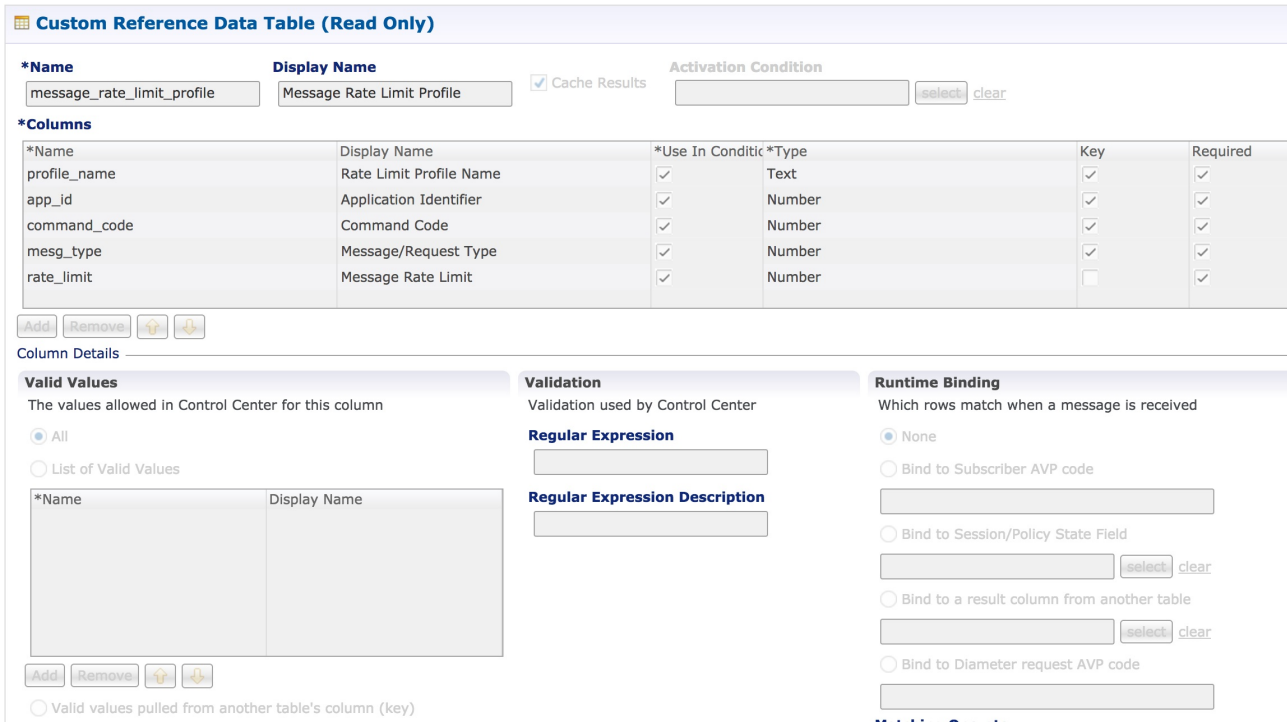
Figure 42: AppId Key Profile Mapping- CRD Table



Message Rate Limit Profile

This table gives a provision to configure Message Rate Limits at a profile level.

Figure 43: Message Rate Limit Profile - CRD Table



- Profile Name: Unique Identifier for a profile.
- Application ID: Application Identifier for this row. 3GPP App Ids only are allowed here.
- Command Code: Command Code of the message that is applicable on the said interface specified by Application Id above.
- Message Type: Initial/Update/Terminate or None for messages that do not have them. The message request type should be same as specified for the command code in Policy Builder under Diameter Application.
- Rate Limit: This field is to specify the threshold in TPS above which the diameter messages are discarded. This value should be more than the Peer Rate Limit in order for message level rate limit to be applied.
- Profile Name: Unique Identifier for a profile.

Refer to Message Rate Limit Profile for configuration in Control Center.

Reserved IMSI

You can configure the Reserved IMSI CRD table to validate a parsed IMSI for SLF routing against a configured list of reserved MCC ranges.

The CRD has two main columns : MCC Start range and MCC End Range. The MCC consists of the first three digits of an IMSI.

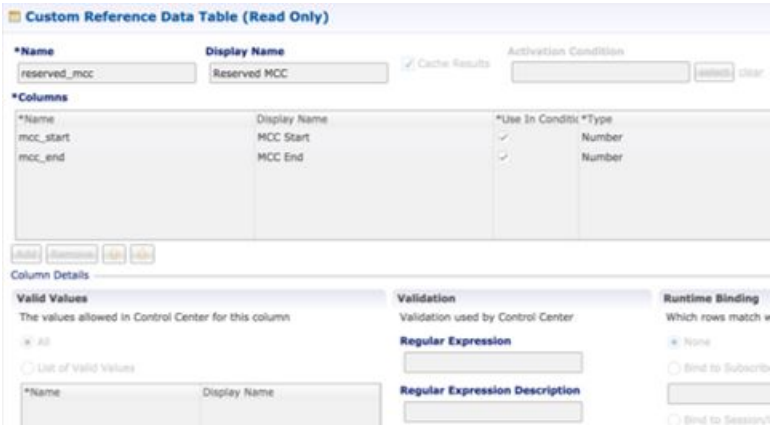
If the IMSI matches a reserved IMSI, the value is ignored for SLF routing.

You can provide support up to ten distinct (non-overlapping) MCC ranges as Reserved IMSIs.

The DRA/SLF ignores AVPs that contain such IMSIs, and continues searching other AVPs in the Diameter request, for a valid address to be used for address resolution.

The following image shows a sample Reserved IMSI configuration:

Figure 44: Reserved IMSI

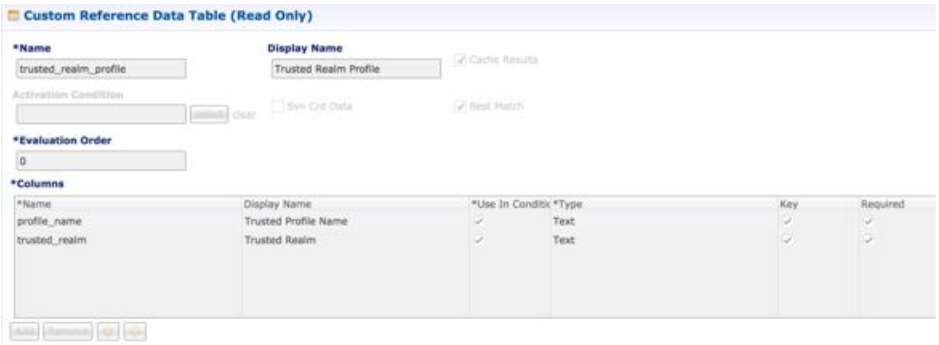


Trusted Realm Profile

Trusted Realm Profile is used for topology hiding. The CRD includes the following columns:

- Trusted Profile Name: Profile Name having a trusted realm mapped to it.
- Trusted Realm: Realm for which Topology Hiding is not required.

Figure 45: Trusted Realm Profile



Protected Realm Trusted Profile Mapping

Protected Realm Trusted Profile Mapping is used for topology hiding. The CRD includes the following columns:

- Protected Realm: Realm that is protected (topology hiding is required).
- Profile Name: Profile having realms that are trusted for this protected realm and that do not require topology hiding.

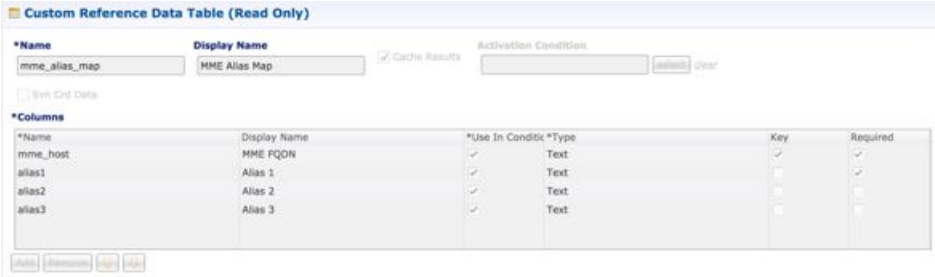
Figure 46: Protected Realm Trusted Profile Mapping

MME Alias Map

MME Alias Map is used for topology hiding. The CRD includes the following columns:

- MME FQDN: FQDN of MME that requires topology hiding.
- Alias1: Mandatory. An alias identity used for the protected host that belongs to an MME in the network.
- Alias 2: Optional. Alternate Alias that can be used for Topology Hiding for the given MME FQDN.
- Alias 3: Optional. Alternate Alias that can be used for Topology Hiding for the given MME FQDN.

Figure 47: MME Alias Map

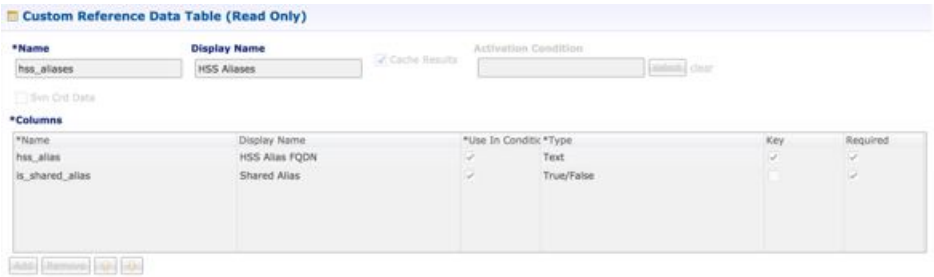


HSS Aliases

HSS Aliases is used for topology hiding. The CRD includes the following columns:

- HSS Alias FQDN: Alias FQDN used to replace a protected HSS FQDN.
- Shared Alias: Boolean variable used to indicate whether the Alias FQDN is shared across multiple HSS servers or not.

Figure 48: HSS Aliases

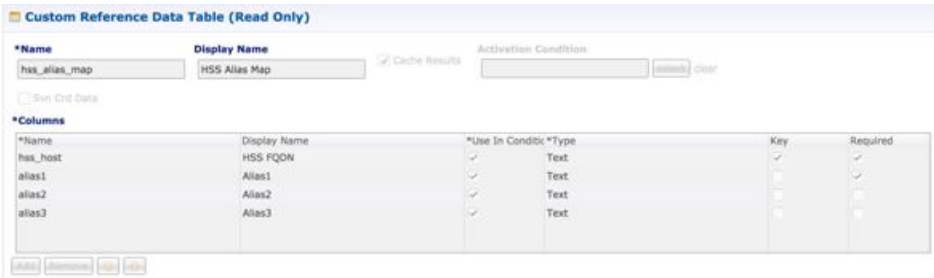


HSS Alias Map

HSS Alias Map is used for topology hiding. The CRD includes the following columns:

- HSS FQDN: FQDN of HSS peer.
- Alias1: Required field which is derived from HSS Alias CRD.
- Alias2: Optional. Alias for the HSS FQDN.
- Alias3: Optional. Alias for the HSS FQDN.

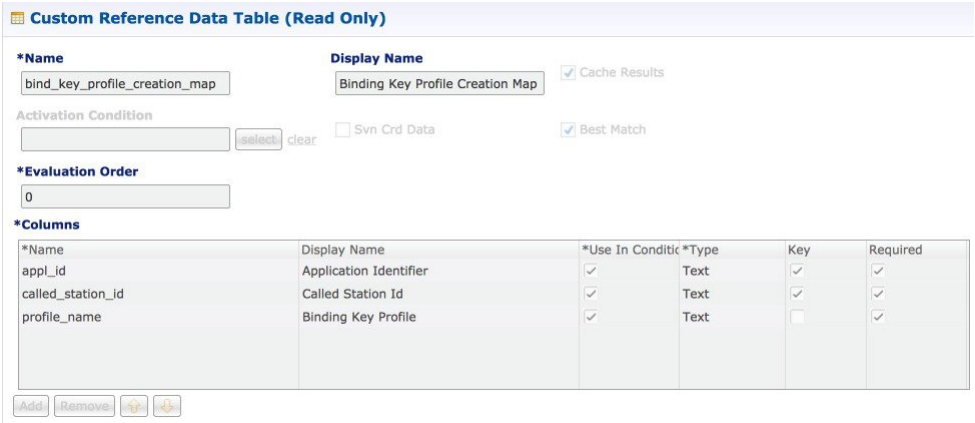
Figure 49: HSS Alias Map



Binding Key Profile Creation Map

This table provides the information related to binding key type profile creation map in the system. The read-only keys are shown below:

Figure 50: Binding Key Profile Creation Map - CRD Table



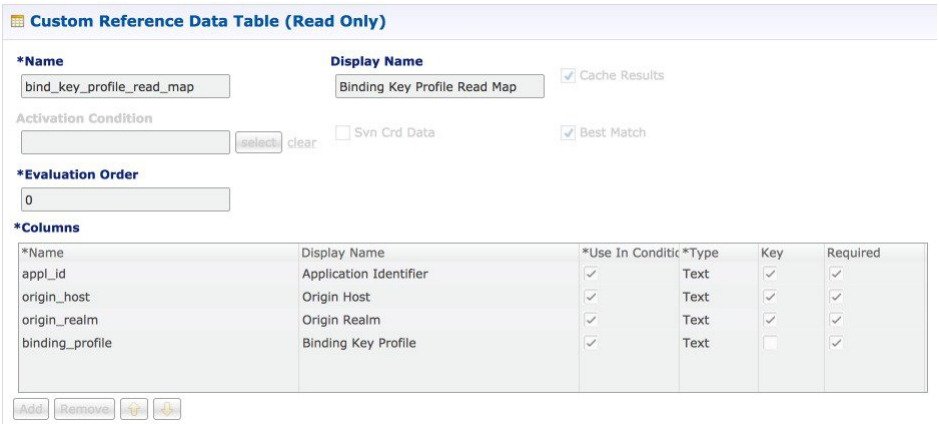
- Application Identifier: Application ID of the message.
- Called Station Id: Called-Station-Id AVP value from the Diameter message.
- Binding Key Profile: Profile name from binding key profile.

Refer to Binding Key Profile Creation Map for configuration in CPS Central.

Binding Key Profile Read Map

This table provides the information related to binding key type profile read map in the system. The read-only keys are shown below:

Figure 51: Binding Key Profile Read Map - CRD Table



- Application ID: Application ID from the message.
- Origin Host: Origin host from the message.
- Origin Realm: Origin realm from the message.
- Binding Key Profile: Profile name from binding key profile.

Refer to Binding Key Profile Read Map for configuration in CPS Central.