



## Custom Reference Data Configuration

---

- [Logical APN List, on page 2](#)
- [Avp Condition Profile, on page 2](#)
- [Avp Action Profile, on page 3](#)
- [APN Mapping Table, on page 4](#)
- [DOIC Profile, on page 5](#)
- [Diameter Avp Dictionary, on page 6](#)
- [Peer Access Control List, on page 7](#)
- [Peer Routes, on page 8](#)
- [Peer Group Mapping, on page 8](#)
- [Peer Group SRK Mapping, on page 9](#)
- [Peer Routing, on page 9](#)
- [IPv6 Ranges System ID Mapping, on page 10](#)
- [Binding Key Profile, on page 11](#)
- [AppId Key Profile Mapping, on page 11](#)
- [Message Class Profile, on page 12](#)
- [Message Retry Profile, on page 12](#)
- [Message Mediation Profile, on page 13](#)
- [Peer Group Answer Timeout, on page 14](#)
- [Message Rate Limit Profile, on page 15](#)
- [Error Result Code Profile, on page 16](#)
- [Gx New Session Rules, on page 16](#)
- [SLF Trigger Profile, on page 17](#)
- [SLF Routing, on page 18](#)
- [S6/Sh Table Driven Rules, on page 18](#)
- [Range Based Routing , on page 19](#)
- [IMSI Range, on page 20](#)
- [MSISDN Range, on page 20](#)
- [Binding Key Profile Creation Map, on page 21](#)
- [Binding Key Profile Read Map, on page 22](#)

## Logical APN List

The logical APN feature allows multiple users to access different physical target networks through a shared APN access point. The logical APN feature reduces the amount of APN provisioning required by consolidating access all real APNs through a single virtual APN. Therefore, only the virtual APN needs to be provisioned at Control Centre, instead of each of the real APNs to be reached.

For details on System ID, refer to [Peer Routing, on page 9](#).

For details on Peer Group, refer to Peer Group Mapping and Peer Group SRK Mapping.

An example configuration is shown below:

**Figure 1: Logical APN List**

Logical APN * (key)	Actions
INTERNET	<a href="#">Edit</a> <a href="#">Delete</a>
IMS-1	<a href="#">Edit</a> <a href="#">Delete</a>
ims4.com	<a href="#">Edit</a> <a href="#">Delete</a>

Filter CRD Tables

[+ Add Row](#)

Show 10 rows | 1 out of 1

## Avp Condition Profile

The Avp Condition profile is used to specify the value and condition to apply to AVP.

The following table describes the fields of Avp Condition Profile:

**Table 1: Avp Condition Profile**

Field	Description
Profile Name	Profile name of the condition. Each row in the table is a condition. You can define multiple conditions for one Profile Name.
Avp	Avp that condition is applied to.
Avp Value	Value of the condition. If there is no AVP, configure the value as AVP_IS_MISSING

Figure 2: Avp Condition Profile

Profile Name * (key)	Avp * (key)	Avp Value *
ResultCodels3xxx	ResultCode	match=3.*

+ Add Row

## Avp Action Profile

You can use the Avp Action Profile to perform mediation at different directions (ingress, egress). You can modify both request and response messages. You can replace, append, prepend value AVP. The value may be static or dynamically retrieved from another AVP or can be extracted as substring from another AVP.

The following table describes the AVP actions that you can perform in Avp Action Profile:

Table 2: AVP Actions

Avp Action	Description
Remove Avp	Removes the AVP from the message.
Add with value	If the AVP is not present in the message, add the AVP with the value defined in CRD.
Add with Avp from Request	If the AVP is not present in the message, the AVP received from the request message is added to it.
Overwrite with value	Overwrite the AVP with the value defined in CRD.
Overwrite with Avp from Request	Overwrite the AVP with the AVP received from the request message
Add Prefix	Add prefix to the value of AVP.
Add Postfix	Add postfix to the value of AVP.
Remove Prefix	Remove prefix from the value of AVP.
Remove Postfix	Remove postfix from the value of AVP.

Figure 3: Avp Action Profile

Avp Action Profile				
Profile Name * (key)	Avp * (key)	Avp Action	Avp Value	Actions
ChangeResultCode	ResultCode	Overwrite with value	3002	<a href="#">✎</a> <a href="#">🗑</a>
PrependLabel	DestinationHost	Add Prefix	core-	<a href="#">✎</a> <a href="#">🗑</a>
Hiding	OriginHost	Overwrite with value	dra	<a href="#">✎</a> <a href="#">🗑</a>
Hiding	OriginRealm	Overwrite with value	dra.cisco.com	<a href="#">✎</a> <a href="#">🗑</a>
ExtractHostFromSessionId	DestinationHost	Overwrite with value	SubString("\${SessionId}", "[^;]+", 1)	<a href="#">✎</a> <a href="#">🗑</a>
AppendLabel	DestinationHost	Add Postfix	-suffix	<a href="#">✎</a> <a href="#">🗑</a>
StripPrependLabel	OriginHost	Remove Prefix	core-	<a href="#">✎</a> <a href="#">🗑</a>
StripAppendLabel	OriginHost	Remove Postfix	-suffix	<a href="#">✎</a> <a href="#">🗑</a>
AddAvp	User-Name	Overwrite with Avp from Request		<a href="#">✎</a> <a href="#">🗑</a>
AddQos	QoS-Information	Add with Avp from Request		<a href="#">✎</a> <a href="#">🗑</a>
AddSubscriptionId	Subscription-Id	Add with Avp from Request		<a href="#">✎</a> <a href="#">🗑</a>
DeleteAvp	User-Name	Remove Avp		<a href="#">✎</a> <a href="#">🗑</a>
AddAvpValue	User-Name	Add with value	0011223344	<a href="#">✎</a> <a href="#">🗑</a>
AddRatType	RAT-Type	Overwrite with value	1004	<a href="#">✎</a> <a href="#">🗑</a>

## APN Mapping Table

The APN is composed of two parts which are as follows:

- The APN Network Identifier. This part of the APN is mandatory.
- The APN Operator Identifier. This part of the APN is optional.

The actual APN of any interface is filled-in with Called-Station-Id AVP. This table keeps a mapping of actual APNs and logical APNs configured in logical APN list.

An example configuration is shown below:

Figure 4: APN Mapping Table

Apn Mapping Table		
CalledStationId * (key)	Logical APN	Actions
internet.com	INTERNET	<a href="#">✎</a> <a href="#">🗑</a>
ims-1.com	IMS-1	<a href="#">✎</a> <a href="#">🗑</a>
ims4.com	ims4.com	<a href="#">✎</a> <a href="#">🗑</a>

[+ Add Row](#)

Show 10 rows | < 1 out of 1 >

## Peer Rate Limit Profile

CPS vDRA can rate limit traffic coming from and going towards a particular peer. This can work for both Ingress and Egress traffic. User needs to define the peer group, FQDN, traffic direction and the CPS vDRA behavior, whether to silently drop or send error message. User can also define the error code and the error message when error responses need to be sent back.

**Figure 5: Peer Rate Limit Profile**

Peer Group * (key)	Peer FQDN * (key)	Message Direction * (key)	Rate Limit Profile	Peer Rate Limit	Discard Behavior *	Result Code	Error String	Actions
match=GX_DC.*	*	Ingress	GX-CCR-I	3	Send Error Answer	3002	OVERLOAD_GX	

+ Add Row

Show 10 rows | 1 out of 1

## DOIC Profile

Use the DOIC Profile table to define the abatement action for Diameter messages in case of Diameter peer overload or congestion.

For more information about DOIC, see [Configure Throttling of Diameter Messages Using DOIC](#).

The following table describes the DOIC Profile table parameters:

**Table 3: DOIC Profile**

Fields	Description	Value
Egress Peer Group	Name of egress peer group.	Referenced from the Peer Group name in Peer Group SRK Mapping.
Message Class	Message classification.  Priority P0 is considered for emergency message class. Hence, it cannot be configured in DOIC for throttling.  In case abatement treatment is applicable for P0 message as per Loss Algorithm (with random number), the message is forwarded.	P1, P2, P3, P4
Abatement Action (output)	The abatement action to be taken by vDRA.	Divert, Forward, Drop

The following abatement actions are supported:

- Forward: Forward allows the message to be sent to the destination peer, even though the peer is overloaded.

- **Divert:** Messages are diverted to the non-congested secondary peer as found using Peer Group SRK Mapping. If the secondary peer is congested, the next non-congested peer is used. If all peers are congested, the messages are dropped.
- **Drop:** Message is throttled with error response 3002

When a message is throttled, a default result-code of 3002 and default message "Throttled due to DOIC congestion" is sent.

The error message can be configured in Error Result Code Profile table with the Error key as Doic Throttled/Dropped.

Default Error Message is "3002: 012 - Throttled due to DOIC congestion"

**Figure 6: DOIC Profile**

Egress Peer Group * (key)	Message Class * (key)	Abatement Action	Actions
*	P1	Drop	

## Diameter Avp Dictionary

Use the Diameter Avp Dictionary CRD table to define the AVP name, AVP code, vendor ID, and the type.

The following table describes the parameters of the Diameter Avp Dictionary:

**Table 4: Diameter Avp Dictionary**

Field	Description
Name	Name of the AVP.
Avp Code	Code of the AVP.
Vendor Id	Vendor ID of the AVP.
Avp Type	Type of AVP (Integer32, Unsigned32, Integer64, OctetString, UTF8String, Grouped)

Figure 7: Diameter Avp Dictionary

Diameter Avp Dictionary

Name * (key)	Avp Code	Vendor Id	Avp Type
ResultCode	268	10415	Unsigned32

+ Add Row

Show 10 rows 1 out of 1

## Peer Access Control List

You can use the Peer Access Control List to specify the list of peers (by realm, FQDN, and application ID) that can establish peer connections to vDRA.

Peers that are not listed with realm or host in the CRD are allowed to establish peer connections by default.

Specify the following parameters:

- Origin Host - Diameter identity or FQDN(host) of the client either in full or as a regular expression
- Origin Realm - Diameter Identity or realm of the client either in full or as a regular expression
- Authorization Action: Specifies whether the incoming client connection is allowed or denied.
- Authorization Deny - Result Code: Configurable result code. If not configured, the default value of 3010 (Unknown Application) or 3007 (Unsupported Application) is sent. Applicable only when the Authorization action is set to “Deny”
- Authorization Deny - Error Message: Configurable Message. If not configured default values are Unknown Peer or Unsupported Application.  
Applicable only when the Authorization action is set to “Deny”
- Application ID: single, comma-separated, or regular expression.

The key fields are Origin Host and Origin realm, hence it is possible to have only one row for each unique pair.

If the peer connection is rejected due to mismatch of Applications, customized result-code / error messages are not applicable in this case.

**Figure 8: Peer Access Control List**

Peer Access Control List						
Origin Host *	Origin Realm *	Authorization Action *	Authorization Deny - Result Code	Authorization Deny - Error Message	Application Id *	Actions
gx-pcef10	match=gx-pcef1.*	Permit			16777238	<a href="#">✎</a> <a href="#">🗑</a>
match=gx-pcef1.*	gx-pcef11.cisco.com	Deny			16777238	<a href="#">✎</a> <a href="#">🗑</a>
gx-pcef14	gx-pcef14.cisco.com	Deny	3008	Peer is Blacklisted	16777238, 16777236	<a href="#">✎</a> <a href="#">🗑</a>
match=gx-pcef.*	gx-pcef12.cisco.com	Permit			16777236	<a href="#">✎</a> <a href="#">🗑</a>

## Peer Routes

Request forwarding is done using Peer Routes to discover peers. These routes are different for different interfaces. There can be multiple peer routes for a particular interface.

**Figure 9: Peer Routes**

Peer Routes	
Peer Route * (key)	Actions
GX_CONSUMER	<a href="#">✎</a> <a href="#">🗑</a>
GX_ENTERPRISE	<a href="#">✎</a> <a href="#">🗑</a>
RX_CONSUMER	<a href="#">✎</a> <a href="#">🗑</a>
RX_CONSUMER_SITE	<a href="#">✎</a> <a href="#">🗑</a>
GX_CONSUMER_C	<a href="#">✎</a> <a href="#">🗑</a>
SD_CONSUMER	<a href="#">✎</a> <a href="#">🗑</a>
SD_CONSUMER_1	<a href="#">✎</a> <a href="#">🗑</a>
RX_ENTERPRISE	<a href="#">✎</a> <a href="#">🗑</a>
SD_CONSUMER_2	<a href="#">✎</a> <a href="#">🗑</a>

[+ Add Row](#)

Show 10 rows | 1 out of 1

## Peer Group Mapping

One or more peers are combined into single peer group based on their realms patterns and FQDN patterns. Peer groups have respective peer routes.



Figure 10: Peer Group Mapping

Peer Group Mapping

Realm Pattern *	FQDN Pattern *	Peer Group	Weight	Actions
pcrf-rx-dra2.seagull.com	pcrf-rx-dra2-seagull	RX_PG	100	<a href="#">✎</a> <a href="#">🗑</a>
pcrf-rx3.seagull.com	pcrf-rx3-seagull	RX_PG	200	<a href="#">✎</a> <a href="#">🗑</a>
match=pcrf-gx.*	match=pcrf-gx.*	GX_PG	300	<a href="#">✎</a> <a href="#">🗑</a>
pcef-gx-dra2.seagull.com	pcef-gx-dra2.seagull.com	PCEF_GX	200	<a href="#">✎</a> <a href="#">🗑</a>
match=pcef-gx.*	match=pcef-gx.*	PCEF_GX	500	<a href="#">✎</a> <a href="#">🗑</a>
rx-af-dra2.seagull.com	rx-af-dra2.seagull	PCEF_RX	100	<a href="#">✎</a> <a href="#">🗑</a>
match=rx-af-dra2.*	match=rx-af-dra2.*	PCEF_RX	300	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows out of 1

## Peer Group SRK Mapping

All the peer groups consisting of one or more peers are listed in this table. Also various features like Session Key Routing or Destination Host Routing can be configured as Only, Never, Preferred depending upon the need. Use the DOIC Enabled column (YES/NO) to enable or disable Diameter Overload Indication Conveyance (DOIC). This option is used to throttle or divert Diameter requests towards PCRF, HSS, AAA, and OCS servers based on reporting of overloaded conditions.

Figure 11: Peer Group SRK Mapping

Peer Group SRK Mapping

Peer Group * (key)	Session Routing Key	Destination Host Routing Rule *	Destination Host Replace	Destination Realm Replace	Doic Enabled	Actions
pcrf-g	pcrf-cluster.pcrf1	Preferred	YES	YES	NO	<a href="#">✎</a> <a href="#">🗑</a>
pcef-g	pcef-cluster.pcef1	Preferred	YES	YES	NO	<a href="#">✎</a> <a href="#">🗑</a>
mme-g	mme-cluster.mme1	Preferred	YES	YES	NO	<a href="#">✎</a> <a href="#">🗑</a>
hss-g	hss-cluster.hss1	Preferred	YES	YES	YES	<a href="#">✎</a> <a href="#">🗑</a>
hss2-g	hss-clusterb.hss1	Preferred			YES	<a href="#">✎</a> <a href="#">🗑</a>

## Peer Routing

This table consists of a mapping of Peer Groups to Peer Routes on a particular CPS vDRA. It also has precedence and weight columns which play a vital role in load balancing behavior of CPS vDRA.

Figure 12: Peer Routing

Peer Route * (key)	System Id * (key)	Peer Group * (key)	Precedence *	Weight *	Actions
SD_CONSUMER_2	system-1	SD_DC_1	1	1	<a href="#">edit</a> <a href="#">delete</a>
GX_CONSUMER	system-1	GX_DC_1	1	1	<a href="#">edit</a> <a href="#">delete</a>
RX_CONSUMER	system-1	GX_DC_1	1	1	<a href="#">edit</a> <a href="#">delete</a>
RX_CONSUMER	system-1	GX_DC_2	1	1	<a href="#">edit</a> <a href="#">delete</a>
SD_CONSUMER	system-1	SD_DC_2	1	1	<a href="#">edit</a> <a href="#">delete</a>
GX_CONSUMER_C	system-1	GX_DC_3	1	1	<a href="#">edit</a> <a href="#">delete</a>
SD_CONSUMER_1	system-1	SD_DC_2	1	1	<a href="#">edit</a> <a href="#">delete</a>

+ Add Row

Show 10 rows | < 1 out of 1 >

## IPv6 Ranges System ID Mapping

Use this table to specify a range of IPv6 addresses and the primary and secondary vDRA system ID.

Figure 13: IPv6 Ranges System ID Mapping

IPv6 Start Range	IPv6 End Range	Primary System Id	Secondary System
No rows available on the server.			

Table 5: IPv6 Ranges System ID Mapping Fields

Fields	Description
IPv6 Start Range	Indicates the start of range in ASCII.
IPv6 End Range	Indicates the end of range in ASCII.
Primary System ID	Mandatory field. Indicates the System ID of vDRA in a vDRA cluster to which the request can be relayed.
Secondary System ID	Secondary vDRA System ID where lookup can happen.



### Note

The ranges are expected to be mutually exclusive and unique. Verify the values when provisioning the same.

# Binding Key Profile



**Important** For routing to work in DRA, user must configure **AppId Key Profile Mapping** and **Binding Key Profile** tables.

The available fields are Boolean fields and can be edited by selecting the check boxes.



**Note** It is expected a minimum of one row to be configured with the value “DefaultProfile”. This will be used in case there is nothing configured for an application Id. For this “DefaultProfile”, “imsiAPN” and “FramedIPv6Prefix” should be enabled.



**Note** The field **MSISDN APN Key Enabled** is a place holder only. Modifying this field will not have an effect on the application behavior.

**Figure 14: Binding Key Profile**

Profile Name * (key)	IMSI APN Key Enabled	MSISDN APN Key Enabled	Framed IPv6 Enabled	Framed IPv4 Enabled	Actions
DefaultProfile	true	false	false	false	
Rx_Profile	false	false	true	false	

+ Add Row

Show 10 rows | < 1 out of 1 >

# AppId Key Profile Mapping



**Important** For routing to work in CPS vDRA, you must configure **AppId Key Profile Mapping** and **Binding Key Profile** tables.

Figure 15: AppId Key Profile Mapping

Application Id * (key)	Profile Name	Actions
Gx	DefaultProfile	<a href="#">✎</a> <a href="#">✖</a>
Rx	Rx_Profile	<a href="#">✎</a> <a href="#">✖</a>

+ Add Row

Show 10 rows | 1 out of 1

The Binding Key Profile column is tied to the Profile Name column from the previous CRD and takes the available Profile Name in the system.

There are two application Identifiers that have been provisioned in the system which are Gx and Rx and can be tied to the same or different Bind Key Profile as the case may be.

## Message Class Profile

To determine the abatement action from the DOIC Profile table (for throttling or diverting Diameter requests), you require a Message class. You can query the Message class from the Message Class Profile table.

The Message Class Profile table takes inputs such as Ingress Peer Group, Application Id, Command Code, Message/Request Type and provides the Condition Profile and Message Class. Message Class can be one of P0, P1, P2, P3, P4.

Figure 16: Message Class Profile

Ingress Peer Group * (key)	Application Id * (key)	Command Code * (key)	Message/Request Type * (key)	Condition Profile * (key)	Message Class	Actions
pcef-g	16777238	272	None	IsEmergency	P0	<a href="#">✎</a> <a href="#">✖</a>
*	16777251	318	None	*	P1	<a href="#">✎</a> <a href="#">✖</a>

## Message Retry Profile

CPS vDRA supports configurable retries, so that the specific behavior of CPS vDRA in congestion scenarios can be configured.

Configurable retry mechanism (i.e., number of retries) per:

- Application ID
- Peer Group
- Answer Timeout error occurred
- Error Result Code of Response

This should be in the form of a CRD and applied to a peer group. The user can use the SRK peers to select an alternate peer.

If all SRK peers fail, the user should use one alternate CPS vDRA if it connects to the SRK. If the SRK matches exactly, CPS vDRA would look for the second label match of SRK like clusterb.dc1 and clusterc.dc1 and retry the message to other peer group.

**Figure 17: Message Retry Profile - Control Center**

Peer Group * (key)	Application Id * (key)	Command Code * (key)	Result Code * (key)	Experimental RC	Number Of Retries	Actions
GX_DC_1	16777238	272	3003	false	6	
GX_DC_1	16777238	272	3004	false	6	
GX_DC_2	16777238	272	3003	false	6	
GX_DC_2	16777238	272	3004	false	6	
match=SD_DC.*	16777303	8388637	3004	false	6	
RX_DC_1	16777236	258	7000	false	6	
GX_DC_2	16777238	272	7000	false	6	

+ Add Row

Show 10 rows | 1 out of 1

Wild card match is supported for Peer Group, Application Id, Command Code, Result Code columns. For example, 300.\* supports all RC starting with 300.

- \* is supported to allow all RC.
- \* is supported for all peer groups.
- Match = GX\_DC\_.\* is supported for groups starting with GX\_DC

RC = 7000 is interpreted as retry for timeout.

Experimental result code is for future purposes and value in that column has no effect on retry processing.



**Note** Best Match check box needs to be checked in Policy Builder if you want to use the wildcard feature.

Refer to [Message Retry Profile](#) for configuration in **Search Table Group**.

## Message Mediation Profile

CPS vDRA supports message mediation profile for following use cases:

- Store an AVP from the request and insert it into the answer. The answer is forwarded from an endpoint or an error generated by CPS vDRA. The known use case for this is storing the MSISDN from a request from the OCS (Sy SNR, Gy RAR) and inserting it in the answer to the OCS. The endpoint cannot handle all cases since the DRA can generate the error response in case of request timeout or inability to route the request to a peer. The MSISDN is in the Subscription-ID AVP.
- Remove an AVP from a request or answer.

Figure 18: Message Mediation Profile - Control Center

Message Mediation Profile

Filter CRD Tables

Application Id * (key)	Command Code * (key)	Message Type * (key)	Peer Group * (key)	Message Direction * (key)	Condition Profile	Action Profile * (key)	Actions
16777302	8388635	Request	*	Ingress		RemovePolicyCounterStatusReport	<a href="#">✎</a> <a href="#">🗑</a>
16777302	8388635	Answer	*	Egress		OverwriteSubscriptionId	<a href="#">✎</a> <a href="#">🗑</a>
16777302	8388635	Request	*	Ingress		RemoveSubscriptionId	<a href="#">✎</a> <a href="#">🗑</a>
4	258	Request	*	Ingress		RemoveSubscriptionId	<a href="#">✎</a> <a href="#">🗑</a>
4	258	Answer	*	Egress		OverwriteSubscriptionId	<a href="#">✎</a> <a href="#">🗑</a>
16777302	8388635	Answer	*	Egress		RemovePolicyCounterStatusReport	<a href="#">✎</a> <a href="#">🗑</a>
4	258	Answer	*	Egress		RemoveRatingGroup	<a href="#">✎</a> <a href="#">🗑</a>

[+ Add Row](#)

Show 10 rows | 1 out of 1

## Peer Group Answer Timeout

CPS vDRA support for the following use cases:

- Configurable answer timeout for initial try and subsequent retries for the following parameters:
  - Application ID
  - Peer Group (to which request is sent)
  - Command code (to enable different timeouts for different Diameter commands)
  - Timeout value (in milliseconds)
- Default value if unspecified is 1700 milliseconds.

Peer group answer timeout is applicable for every message routed using:

- Destination host routing
- SRK routing
- Table driven routing

Sample peer group answer timeout is shown below:

Figure 19: Peer Group Answer Timeout

Peer Group Answer Timeout

Peer Group *	Application Id *	Command Code *	Timeout Milliseconds	Actions
SD_DC_2	16777303	*	15000	
RX_DC_2	16777236	*	22000	
GX_DC_3	16777238	*	20000	
SD_DC_2	16777303	8388637	25000	
SD_DC_1	16777303	272	25000	
RX_DC_1	16777236	*	20000	
GX_DC_2	16777238	272	22000	
GX_DC_1	16777238	match=2.*	25000	

+ Add Row

Show 10 rows out of 1

Wild card match is supported for application\_id, peer\_group, command code. \* indicates all application\_id, peer\_group.

The following rules have been applied for answer timeout:

- Default timeout for any message routed from CPS vDRA is 1700 ms.
- In case of retry, if an alternate group is chosen for routing, corresponding timeout for the peer group is applied.

For Policy Builder related configuration, refer to [Peer Group Answer Timeout](#).

## Message Rate Limit Profile

Further to peer level rate limit, CPS vDRA provides the granularity of limiting diameter traffic at message level for each peer. Message level rate limit always works in conjunction with peer level rate limit and is an additional control in peer level rate limit configuration. Since message level rate limit works in conjunction with peer level rate limit, all the fields specified for peer level rate limit are applicable to message level rate limit.

Message Rate Limit Profile table is used to get the condition for such rate limiting. User can define the type of message, command code and the application for which the limiting has to be implemented.

Figure 20: Message Rate Limit Profile

Message Rate Limit Profile

Rate Limit Profile Name * (key)	Application Identifier * (key)	Command Code * (key)	Message/Request Type * (key)	Message Rate Limit *	Actions
GX-CCR-I	Gx	272	1	7	

+ Add Row

Show 10 rows out of 1

## Error Result Code Profile

Sample CRD data looks like this:

**Figure 21: Error Result Code Profile**

Error Result Code Profile						
Application Id * (key)	Error * (key)	Result Code	Exp Result Code	Vendor Id	Err Msg	Actions
*	No Available Peer	4004	5004	10415	Peer not available	<a href="#">✎</a> <a href="#">🗑</a>
*	No Peer Group	4005		10415	No Peer Group Available	<a href="#">✎</a> <a href="#">🗑</a>
*	No Binding Found	3024		10415	No Binding found for request	<a href="#">✎</a> <a href="#">🗑</a>
*	Message Loop Detected	4007		10415	Loop Detected in Message	<a href="#">✎</a> <a href="#">🗑</a>
*	No Binding Key For Lookup		4008	10415	No Binding Key for Lookup	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows | 1 out of 1

- For any CPS vDRA error or message timeout, CPS vDRA has the ability to map the error to a Result-Code value and an error message string for the Error-Message AVP.
- Errors include things like "binding not found", "message timeout", "no peer connections".
- The Result Code value is sent in the Result-Code AVP in the response.
- The error message string is sent in the Error-Message AVP in the response.
- When both Result Code and Exp Result Code are configured in this table, Result Code will take precedence. In case Result Code is not configured in this table, Exp Result Code will be sent with Vendor-ID.

## Gx New Session Rules

Gx New Session Rules table is used by CPS vDRA when performing Table Driven routing. CPS vDRA could derive the "Peer Route" from this table, when the incoming message has no destination host to be routed to. From peer route, CPS vDRA derives further route where the request could be sent. This table supports both wildcard and exact match for the various parameters. The "Peer Route" used in this table should be defined in "Peer Routes" table. Here an example for Gx New Session Rules is provided. Similar tables can be created for Rx or Sd.



Figure 22: Gx New Session Rules

Gx New Session Rules

Filter CRD Tables

Logical APN * (key)	Origin Host * (key)	Peer Route	Origin Realm * (key)	Destination Host * (key)	Destination Realm * (key)	MSISDN * (key)	IMSI * (key)	Actions
ims4.com	gx-pcef	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims3.com	*	GX_CONSUMER	gx-pcef.cisco.com	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	*	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims2.com	*	GX_CONSUMER	*	*	*	*	45005978851107	<a href="#">✎</a> <a href="#">🗑</a>
ims1.com	*	GX_CONSUMER	*	*	*	match=*2829	*	<a href="#">✎</a> <a href="#">🗑</a>
ims5.com	*	GX_CONSUMER	*	*	gx-dra1.cisco.com	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims6.com	*	GX_CONSUMER	*	gx-dra1	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef1	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef2	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef3	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows | 1 out of 3

# SLF Trigger Profile

In this table, there are three input keys: Application Id, Command Code and Destination Realm. If all these input keys are matched from the Diameter incoming requests and trigger condition for the SLF trigger table is matched, then CPS vDRA derives the Primary Lookup Type (IMSI/MSISDN) and SLF Destination Type as output of SLF trigger table. Then a query is made in the SLF Database using the Primary Lookup Type (IMSI/MSISDN) and SLF-Destination-Type.

This table is used in the case when the Diameter Request does not contain any “Destination-Host” AVP or, in case the “Destination-Host” AVP comes with the Diameter Host Name of CPS vDRA.

Figure 23: SLF Trigger Profile

SLF Trigger Profile

Filter CRD Tables

Application ID * (key)	Command Code * (key)	Destination Realm * (key)	Primary Lookup Type *	Secondary Lookup Type	SLF Destination Type *	Actions
16777251	*	s6-hss.com	IMSI		LTE-HSS	<a href="#">✎</a> <a href="#">🗑</a>
16777291	*	slh-hss.com	IMSI		LTE-HSS	<a href="#">✎</a> <a href="#">🗑</a>
16777217	*	sh-hss.com	MSISDN		LTE-HSS	<a href="#">✎</a> <a href="#">🗑</a>
16777251	316	mnc286.mcc311.3gppnetwork.org	IMSI		S6a-HSS	<a href="#">✎</a> <a href="#">🗑</a>
16777251	*	mnc286.mcc311.dranetwork.org	IMSI		S6a-HSS	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows | 1 out of 1

Based on Application ID 16777251, Command Code 316 and Destination Realm of `ims.mnc286.mcc311.3gppnetwork.org`, Primary Lookup Type selected is IMSI and SLF Destination Type is selected as S6a-HSS. This Primary Lookup Type and SLF Destination Type is used to query SLF database for the configured lookup type.

# SLF Routing

This table contains the mapping of SLF destination and the SRK of peer groups where the message could be routed. The SLF destination is derived from SLF subscriber database.

**Figure 24: SLF Routing**

SLF Destination * (key)	SLF Session Route Key *	Actions
lte_hss_a	hss_cluster_dc1	<a href="#">✎</a> <a href="#">🗑</a>
S6a-HSS1	LTE.S6a.HSS	<a href="#">✎</a> <a href="#">🗑</a>
SLF-S6a-HSS	LTE.S6a.HSS	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows out of 1

# S6/Sh Table Driven Rules

This table is used for table driven routing of S6/Sh messages when the destination host is not available in the incoming request and there is no match SRK found in SLF Trigger table/SLF Mapping table. Keys used for deriving the peer route are Origin Host, Origin Realm, Destination Host, Destination Realm, MSISDN, IMSI and the output is Peer Route.

An S6 Table Driven Rules example configuration is given.

**Figure 25: S6/Sh Table Driven Rules**

Logical APN * (key)	Origin Host * (key)	Peer Route	Origin Realm * (key)	Destination Host * (key)	Destination Realm * (key)	MSISDN * (key)	IMSI * (key)	Actions
ims4.com	gx-pcef	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims3.com	*	GX_CONSUMER	gx-pcef.cisco.com	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	*	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims2.com	*	GX_CONSUMER	*	*	*	*	45005978851107	<a href="#">✎</a> <a href="#">🗑</a>
ims1.com	*	GX_CONSUMER	*	*	*	match=*2829	*	<a href="#">✎</a> <a href="#">🗑</a>
ims5.com	*	GX_CONSUMER	*	*	gx-dra1.cisco.com	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims6.com	*	GX_CONSUMER	*	gx-dra1	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef1	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef2	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>
ims.com	gx-pcef3	GX_CONSUMER	*	*	*	*	*	<a href="#">✎</a> <a href="#">🗑</a>

+ Add Row

Show 10 rows out of 3

# Range Based Routing

CPS vDRA provides range-based routing based on MSISDN and IMSI values so that Diameter requests are routed to the correct HSS or AAA server. Range-based routing occurs if the destination-host routing, binding-based routing and SLF-based routing fails.

- vDRA checks whether the primary lookup type is IMSI or MSISDN and also checks whether the IMSI/MSISDN value present in the request matches against the range configured in CRD.
- The primary lookup type is evaluated first and if it fails, the secondary lookup type is evaluated.
- If primary lookup type evaluation fails and if the secondary lookup type is not configured, the request is routed with table-driven routing (if configured).
- If both the primary lookup type credential and the secondary lookup type evaluation fail, the request is rejected or routed with table driven routing (if configured).

vDRA matches the request against the Range Based Routing table and based on the result of the credential match, SRK routing is initiated.

**Table 6: Range Based Routing**

Field	Description	Value
Application Id (input)	The diameter application of the message received	Integer value of the application id
Command Code (input)	The message command code	Integer value of the command code
Destination Realm (input)	The destination realm in the message	String value of destination realm
Primary Lookup Type (input)	Primary lookup type for range based routing	IMSI or MSISDN
Secondary Lookup Type (input)	Secondary lookup type for range based routing	NONE or IMSI or MSISDN
Routing Profile (output)	Routing profile	Any string value. (Should match the routing profile in either or both the IMSI and MSISDN range CRD for a successful match).

**Figure 26: Range Based Routing**

## Range Based Routing

Filter CRD Tables						
Application ID *	Command Code *	Destination Realm *	Primary Lookup Type *	Secondary Lookup Type *	Routing Profile *	Actions
16777238	272	pcrf-gx-dra2.seagull.com	MSISDN	IMSI	<a href="#">routingProfile</a>	 

## IMSI Range

The IMSI Range is used in range-based routing to configure the range of IMSI values.

**Table 7: IMSI Range**

Field	Description	Value
Routing Profile (input)	The routing profile name	Any string value
IMSI lower bound (input)	The lower bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, use the syntax: <code>match=&lt;regex&gt;</code>
IMSI upper bound (input)	The upper bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, leave it blank.
SRK (output)	The SRK key	Any string value

Examples:

- For configuring numeric range between 9840510345 to 984059999: Lower bound: 9840510345, Upper bound: 9840598823
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: `match=98405[0-9]*`, Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840501333 to 9840502999: Lower bound: `match=984050(1|2)[3-9]*` , Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: `match=98405(((2-7)[0-9]*)(8[0-8][0-4][0-5][0-6])(1[0-9][2-9][3-9][4-9]))`, Upper bound : <leave it empty>

**Figure 27: IMSI Range**

The screenshot shows a configuration table for IMSI Range. The table has a search bar at the top right labeled 'Filter CRD Tables'. Below the search bar, the table columns are: Routing Profile \*, IMSI lower bound \*, IMSI upper bound, SRK \*, and Actions. The first row contains the values: routingProfile, 333333333300000, 333333333344444, and srk.dc3. The Actions column contains edit and delete icons.

Routing Profile *	IMSI lower bound *	IMSI upper bound	SRK *	Actions
routingProfile	333333333300000	333333333344444	srk.dc3	

## MSISDN Range

The MSISDN Range is used in range-based routing to configure the range of MSISDN values.

Table 8: MSISDN Range

Field	Description	Value
Routing Profile (input)	The routing profile name	Any string value
IMSI lower bound (input)	The lower bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, use the syntax: match=<regex>
IMSI upper bound (input)	The upper bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, leave it blank.
SRK (output)	The SRK key	Any string value

Examples:

- For configuring numeric range between 9840510345 to 984059999: Lower bound: 9840510345, Upper bound: 9840598823
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405[0-9]\*, Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840501333 to 9840502999: Lower bound: match=98405(1|2)[3-9]\* , Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405(([2-7][0-9]\*)(8[0-8][0-4][0-5][0-6])(1[0-9][2-9][3-9][4-9])), Upper bound : <leave it empty>

Figure 28: MSISDN Range

Routing Profile *	MSISDN lower bound *	MSISDN upper bound	SRK *	Actions
routingProfile	match=99999[0-9]*		srk.dc3	

# Binding Key Profile Creation Map

The available fields are Boolean fields and can be edited by selecting the check boxes.

Figure 29: Binding Key Profile Creation Map

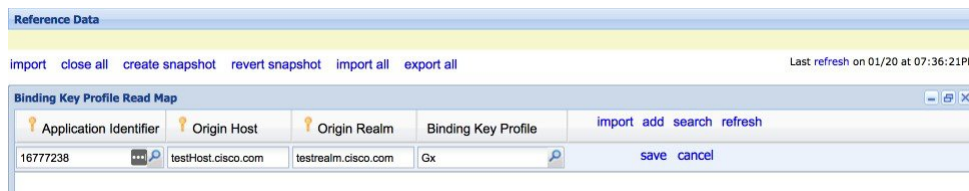
- APN field supports both wildcard "\*" and regex match like "match=ims.\*".

- For Binding Key profile Read Map, both Origin-Host and Origin-Realm support wildcard and regex match.
- Binding Profile and Binding Key Profile fields use values from Profile name field in Binding Key Profile table. Any profile first has to be defined in Binding Key Profile and then only it could be used in create or read tables.

## Binding Key Profile Read Map

The available fields are Boolean fields and can be edited by selecting the check boxes.

**Figure 30: Binding Key Profile Read Map**



The screenshot displays a web interface for managing Binding Key Profile Read Maps. At the top, there is a 'Reference Data' section with a yellow background. Below it, a toolbar contains buttons for 'import', 'close all', 'create snapshot', 'revert snapshot', 'import all', and 'export all'. A timestamp indicates 'Last refresh on 01/20 at 07:36:21PM'. The main area is titled 'Binding Key Profile Read Map' and features a table with the following columns: 'Application Identifier', 'Origin Host', 'Origin Realm', and 'Binding Key Profile'. The table contains one row with the values '16777238', 'testHost.cisco.com', 'testrealm.cisco.com', and 'Gx'. To the right of the table, there are buttons for 'import', 'add', 'search', and 'refresh'. Below the table, there are 'save' and 'cancel' buttons.

Application Identifier	Origin Host	Origin Realm	Binding Key Profile
16777238	testHost.cisco.com	testrealm.cisco.com	Gx