

Oracle® Communications Diameter Signaling Router Mediation User Guide



Release 9.3.0.0.0
G53133-01
April 2026

ORACLE®

Copyright © 2011, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
1.1	Overview	1
1.2	My Oracle Support	1
2	Diameter Mediation	
2.1	Mediation overview	1
2.2	Rule Templates	3
2.2.1	Impact of Adding Rule Templates	7
2.2.1.1	Assessing a Rule Template for Safe Use	8
2.2.1.2	Raising the Limit of Active Rule Templates	10
2.2.2	Rule Template elements	10
2.2.3	Adding a Rule Template	36
2.2.4	Adding online help to a Rule Template	38
2.2.4.1	Rule Templates Help elements	39
2.2.5	Copying a Rule Template	39
2.2.6	Changing a Rule Template	39
2.2.7	Importing a Rule Template	40
2.2.8	Exporting a Rule Template	40
2.2.9	Deleting a Rule Template	41
2.2.10	Exporting all Rule Templates	42
2.3	Formatting Value Wizard	42
2.3.1	Formatting String Wizard elements	43
2.4	Enumerations	59
2.4.1	Mediation Enumerations elements	59
2.4.2	Adding an Enumeration	60
2.4.3	Editing an Enumeration	60
2.4.4	Deleting an Enumeration	61
2.5	Triggers	61
2.5.1	Mediation Triggers elements	64
2.5.2	Associating a Rule Set with a Trigger	65
2.5.3	Removing the Association of a Rule Set with a Trigger	66
2.6	State and Properties	66
2.6.1	Mediation State and Properties elements	67

2.6.2	Importing a Rule Template	68
2.6.3	Editing State and Properties	68
2.6.4	Deleting a Rule Template	69
2.7	Internal Variables	69
2.7.1	Mediation Internal Variables elements	70
2.7.2	Adding an Internal Variable	71
2.7.3	Editing an Internal Variable	71
2.7.4	Deleting an Internal Variable	71
2.8	Measurements	72
2.8.1	Mediation Measurements elements	72
2.8.2	Adding a Measurement	72
2.8.3	Editing a Measurement	73
2.8.4	Deleting a Measurement	73
2.9	Rule Sets	73
2.9.1	Rule Sets elements - View page	75
2.9.2	Rule Sets elements - Insert and Edit Pages	76
2.9.3	Adding a Rule to a Rule Set	78
2.9.4	Deleting All Rules from a Rule Set	79
2.9.5	Changing a Rule in a Rule Set	79
2.9.6	Deleting One Rule from a Rule Set	79
2.9.7	Export/Import Rules	79
2.9.7.1	Exporting Rules	81
2.9.7.2	Importing Rules	81
2.9.7.3	Exporting all Rules	82

Preface

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

Documentation Accessibility

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

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**.
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Whats New in This Guide

This section introduces the documentation updates for DSR 9.3.0.0.0 release.

Release 9.3.0.0.0 - G53133-01, April 2026

- Added the following sections in the Diameter Mediation section:
 - [Exporting all Rules](#)
 - [Exporting all Rule Templates](#)

1

Introduction

This chapter contains an overview of *Diameter Mediation*. The contents include sections on the scope, audience, and organization of the documentation, and how to contact Oracle for assistance.

1.1 Overview

The *Diameter Mediation* manual provides information about how to use the Mediation GUI.

The manual provides the following types of information:

- Creation and modification of Rule Templates
- Provision rules and data in Rule Sets

1.2 My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request
2. Select **3** for Hardware, Networking and Solaris Operating System Support
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

2

Diameter Mediation

The **Diameter**, and then **Mediation** pages on the **Overseer** allow you to manage the **Mediation** feature.

Diameter Mediation helps to solve interoperability issues. Mediation uses rules to manipulate header parts and Attribute-Value Pairs (**AVPs**) in an incoming routable message. When data in the message matches some specified conditions at a specified point of message processing, then actions are applied to modify the message routing decisions.

Note

When activated, the Mediation folder and a Rules Template sub-folder appear under Diameter in the left-hand GUI menu. For more information on how to activate the feature, see the *Mediation Feature Activation Procedure*.

2.1 Mediation overview

Diameter Mediation helps to solve interoperability issues. Mediation software uses rules to manipulate the header and Attribute-Value Pairs (**AVP**) in an incoming routable message. When data in the message matches specified condition(s) at a specified point in the message header, then rules are enacted to reroute the message..

Note

When activated, the Mediation folder and a Rules Template sub-folder appear under Diameter in the left-hand GUI menu. For more information on how to activate the feature, see the *Mediation Feature Activation Procedure*.

A rule template defines the conditions that must match in a message and the actions applied to reroute the message.

- A condition defines the part of the message used in the comparison, an operator for the type of comparison, and a type that must match in the message part. Up to five (5) conditions in the same rule template are collectively referred to as a **Condition Set**. The Condition Set can use AND, OR, or it can apply user specified Complex Expression for comparison.
- An action can add, alter, or delete AVPs; and modify the message header Version, Command-Code, or Application-ID Diameter components. Two or more Actions in a Rule Template are collectively referred to as an Action Set.

A Rule Set is generated by moving the rule template to a test or active status (conditions and the actions can be provisioned in one or more rules in the rule set as described in [Rule Sets](#)).

Rule sets can be associated with pre-defined Request or Answer **Trigger Points** in the message processing logic. When a message reaches a Trigger point and the Condition Set in an associated Rule Set is met, the Action set for that Rule Set is applied to the message. The

changes to the message content can result in modifying the message behavior and routing decision at that Trigger point. The available Diameter Mediation Triggers Points are described in [Triggers](#).

A designed Meta-Administrator can use the Rules Template and other GUI pages to perform the following tasks:

- Create, modify, delete, copy, import, and export Rule Sets as described in [Rules Template](#).
- Add help text to a Rule Template; the help text is available for the Rule Set generated from the Rule Template as described in [Rules Template](#).
- Change the state of a Rule Template as described in [State and Properties](#):
- Set the Action Error Handling property of a Rule Set as described in [State and Properties](#).
- Enable the Status of Rule Counters to display the Rule Counters as described in [State and Properties](#).
- Import previously exported Rule Templates as described in [State and Properties](#).
- Add, edit, and delete a rule to a Rule Set and provision the actual data used by the rule in the message matching process as described in [Rule Sets](#).
- Import/Export Rules provisioned in the rule templates in the Test or Active State as described in [Rule Sets](#).
- Associate Rule Sets with Triggers and remove Rule Set associations with Triggers as described in [Triggers](#).
- View, create, edit, and delete the **Internal Variables** used in the rules as described in [Internal Variables](#).
- View, create, edit, and delete the Enumeration types used in the rules as described in [Enumerations](#).
- View, create, edit, and delete the Vendors-specific AVPs used in Rule Templates as described in the *Diameter User's Guide*.
- Test a Rule Set.

The designed Rule Set Administrator can perform the following tasks:

- Add a rule to a Rule Set, and provision the actual data that used by the rule in the message matching process as described in [Rule Sets](#).
- Edit and delete rules in Rule Sets as described in [Rule Sets](#).
- Delete Rule Sets as described in [Rule Sets](#).
- Change the state of a Rule Template as described in [Rules Template](#).
- Set the Action Error Handling property of a Rule Set as described in [Rule Sets](#).
- Test a Rule Set.

A Diagnostics Tool is available at the **SOAM** that allows one to test Mediation rules before exposing the Mediation rules to live traffic. . The Diagnostics Tool logs the rules applied, Actions taken, and other diagnostics information when a test message is processed into the system. The tool generates traffic and sends Diameter messages on a test connection. As a test message traverses the system, the application logic generates diagnostics messages at Trigger points. View the diagnostics log reports in the **Diameter**, and then **Reports**, and then **Diagnostics Tool** page as described in the *Diameter User's Guide*.

- Associate Rule Sets with Triggers, and remove Rule Set associations with Triggers, as described in [Triggers](#).
- Import previously exported Rule Templates as described in [State and Properties](#).

- View the Internal Variables used in the rules as described in [Internal Variables](#).
- View the Enumeration types in the rules as described in [Enumerations](#).
- View the Vendors used in Rule Templates as described in [Rules Template](#).

2.2 Rule Templates

All Rule Templates are listed on the **Diameter**, and then **Mediation**, and then **Rule Templates** page with check boxes by each row. To automatically check all the rule templates, check the box by the rule template Name.

Rule templates are created by:

- Formulating the Conditions against which to match incoming requests or responses.
- Defining the Actions that are applied to the message when the conditions match.

A Rule template is created by configuring Settings, Conditions, and Actions sections.

Guidelines To Follow While Creating a Rule Template:

1. Do not maintain duplicate rule templates:

Figure 2-1 Rule template

17 Gx IMS_APN_DR_PCRF	Called-Station-Id[=]CC-Request-Type[=]Subscription-Id[=]	@msg.avp["Subscription-Id"]avp["Subscription-Id-Data"]
18 Gx IMS_APN_DR_PCRF-Subst	Called-Station-Id[=]CC-Request-Type[=]Subscription-Id[=]	@msg.avp["Subscription-Id"]["1"]avp["Subscription-Id-Data"]["1"]data

2. Try selecting the left value from the **wizard** and choose proper data types while configuring the conditions and actions in the rule templates based on RFC or 3Gpp specifications such as:
 - SubscriptionId-Data Type: UTF8String
 - OriginHost: DiameterIdentity
 - OriginRealm: DiameterIdentity
3. When creating the rule template, the possible preference order for selecting the operator for the condition:
 - Exact match
 - Regex match
 - Begins with (longest match): prefer for the longest match instead of combination of multiple strings like: "hss2, hss21, hss25" (This must be avoided, instead choose string having most common substring like hss2 or choose substrings individually like hss21 and hss25).

Scenario 1, Case 1

The following table shows the first approach of creating a rule:

Table 2-1 Scenario 1, Case 1

Application ID	Command Code	Origin host (Begins with (Longest match))	Egress Name	Action
16777251	316	hss25	mme01	add_avp/set_routelist
16777251	316	hss27	mme01	add_avp/set_routelist

Table 2-1 (Cont.) Scenario 1, Case 1

Application ID	Command Code	Origin host (Begins with (Longest match))	Egress Name	Action
16777251	316	hss29	mme01	add_avp/set_routelist

Scenario 1, Case 2

The following table shows the second approach of configuring the single rule which sets all the conditions such as hss25, hss27, hss29:

Table 2-2 Scenario 1, Case 2

Application ID	Command Code	Origin Host (Begins with (Longest match))	Egress name	Action
16777251	316	hss2	mme01	add_avp/set_routelist

Scenario 2, Case 1

The following table shows the first approach of creating a rule:

Table 2-3 Scenario 2, Case 1

Application Id	Command Code	Subscription Id1 (Begins with (longest match))	Action
16777251	316	28910	add_avp/set_routelist
16777251	316	28920	add_avp/set_routelist
16777251	316	28930	add_avp/set_routelist

Scenario 2, Case 2

The following table shows the second approach of configuring a single rule which sets all the conditions:

Table 2-4 Scenario 2, Case 2

Application Id	Command Code	Subscription Id1 (Begins with (longest match))	Action
16777251	316	289	add_avp/set_routelist

4. When configuring rules in the rule sets, do not add any trailing and leading spaces to the value.

Settings

- The Rule Template Name is placeholder for meaningful text to describe the purpose of the rule template and rule set.
- The Message Type Support is automatically determined depending upon the configured conditions and actions for the Rule Template.

Conditions

One or more (up to 5) matching conditions can be defined in the rule template. The expressions are combined into one logical expression using ANDed, ORed, or Complex Expression operators.

Each matching expression consists of a left-hand value or operand, an operator, and a right-hand value or operand.

- The Left-hand value or operand allows accessing any part of a message, any information stored by the previous rule template, and any information the application resolves runtime.
- The Operator allows comparison of the Left-hand and Right-hand values.
- The Right-hand value or operand allows performing the syntax check for the entered data on the generated Rule Sets page.
- The Condition Set is used to form one logical expression by combining all the Conditions of that Rule Template.

Conditions can be configured to cause Mediation to use fast database look-ups of the rule data. For information on fast search see [Fast Search versus slow search](#).

Actions

One or more (up to 5) actions indicate what to do when the conditions match (such as modify the part of a message, forward a message, send a reply, insert or remove AVP headers, or set attributes for further processing). Actions implement the mediation of a message.

When the message processing reaches a selected triggering point, the conditions of the rule template are examined for the message. If the conditions match, actions are applied to the message. The actions allow manipulation of some particular part of the message, adding or deleting information in the message, forwarding the message to a specific destination, or triggering of diameter message copy to send a copy of the message to a Diameter Application Server (**DAS**).

The rule template defines the actions to take when a Mediation operation is triggered and its condition set is matched that form an action set. A list of actions available are in the [Rule Template elements](#).

On the **Diameter**, and then **Mediation**, and then **Rule Templates** page, you can perform the following actions:

- Filter the list of rule template Names to display only the desired rule templates.
- Add a new rule template and its values on the **Diameter**, and then **Mediation**, and then **Rule Templates [Insert]** page. For information on adding a rule template see [Adding a Rule Template](#). If the maximum number of rule templates (100) already exists in the system, the Rule Templates [Insert] page does not open, and an error message displays.
- Import a rule template from a location outside the Diameter system, to which the rule template was previously exported from Mediation on the **Diameter**, and then **Mediation**, and then **Rule Templates [Import]** page. For information on importing a rule template see [Importing a Rule Template](#). If the maximum number of rule templates (100) already exists in the system, the Rule Templates [Import] page does not open, and an error message displays.
- Copy a rule template to create a new rule template on the **Diameter**, and then **Mediation**, and then **Rule Templates [Copy]** page. For information on copying a rule template see [Copying a Rule Template](#). If the maximum number of rule templates (100) already exists in

the system, the Rule Templates [Copy] page does not open, and an error message displays.

- Edit the selected rule template on the **Diameter**, and then **Mediation**, and then **Rule Templates [Edit]** page. For information on changing a rule template see [Changing a Rule Template](#).
- Delete a selected rule template. For information on deleting a rule template see [Deleting a Rule Template](#).
- Export one or more rule template(s) to a location outside of the Diameter system on the **Diameter**, and then **Mediation**, and then **Rule Templates [Export]** page. For information on exporting a rules template see [Exporting a Rule Template](#).
- Set Help creates online help for the selected rule template on the **Diameter**, and then **Mediation**, and then **Rule Templates [Set Help]** page. For information on adding online help to a rule template see [Adding online help to a Rule Template](#).

Fast Search versus slow search

The mediation templates can be fast search or slow search. The fast-search templates include only the conditions that can be looked up using the fast search algorithm. This algorithm is more efficient and allows having more templates in the system and more provisioned rules. The slow search template is a template that includes at least one slow search condition. It is allowed to have up to 2000 rules for the fast-search template and 250 only for the slow search template. Even if the template is a slow search template, there are still some recommendations to improve the performance:

- Optimize the template when it is possible. Try to make the template conditions to be fast search when possible.
- Move fast search conditions to the top when possible.
- Try to avoid usage of regular expressions when possible.

Fast search or slow search depends on the content of the condition, the condition evaluation order, and the condition grouping method in the following order:

1. Fast search dependency on the operator, right value type and the default value:
 - "yes" sign if one of the operators "==" , "=^", "=^", "is within", "exists", "does not exist", "is true", "is false" is selected and the right hand operand type is not "xl-value".
 - "yes sign" if the default value is fixed regardless of the selected operator and the right value type.
 - "no" sign for other cases
2. Fast search dependency on the condition evaluation order and condition grouping method:
 - Conditions that are ANDed:
 - "yes" sign if the condition is the first on the condition set layout or all the conditions above also have "yes" sign under the fast search label.
 - "no" sign for other case.
 - Conditions that are ORed:
 - "no" sign for all conditions
 - Conditions that form a complex expression that use a mix of AND and OR operations:
 - "yes" sign if the condition is the first condition evaluated in the expression or all the conditions before the expression also have the "yes" sign under the fast search label + the condition is simply ANDed to the rest of expression

- "no" sign if the condition is ORed
- "no" sign for other cases

For example assume, that A, B, C and D are fast-search based on 1.

- The specified complex expression is (C OR D) AND A AND B:
 - First (C OR D) is evaluated. Since C, D are ORed conditions, they are non fast-search. A, B conditions are evaluated after the non-fast search conditions; hence they are also non fast-search.
 - This expression can be optimized by changing the order to A AND B AND (C OR D). Where A, B conditions are evaluated first and they are fast search.

Another example assume that A,B,C are fast-search based on 1.

- The specified complex expression is (A AND B) OR (C AND A):
 - None of the conditions is simply ANDed to the rest of expression hence none of the conditions has fast-search.
 - This expression can be optimized by changing the order to A AND (B OR C). Where A is evaluated first and it is simply ANDed to the rest of expression hence it has the fast search.

Performance

[Table 2-5](#) describes the performance impact of the DSR in specific scenarios. The Simple, Medium, and Complex categories define the maximum percentage of degradation while meeting the existing message latency and reliability and currently measured CPU utilization.

Table 2-5 Mediation Task Classifications

Percentage of Degradation	Number of Conditions	Types of Operators used in Conditions	Number of Rules in the Rule Set
Simple <= 1%	1-2	exist, does not exist, equals ("=="), begins with ("=^")	1-50
Medium <=5%	3-4	Greater than (">"), Greater than or Equal to (">="), Less than ("<"), Less than or Equal to("<="), Hash Value hash(<string>,<range>)	51-150
Complex <=10%	5	Pattern matches	151-250

2.2.1 Impact of Adding Rule Templates

Using the mediation framework to manipulate messages can result in increased CPU usage. The CPU increase is dependent on the several factors including the type and number of searches as well as the number and types of changes being performed to the messages. For example, a simple numeric search or an action such as deleting an AVP may not result in increased CPU usage, but a search and replace based on complex pattern matching likely results in increased CPU usage.

The Mediation framework supports a large number of conditions, operators and actions and it is not practical to quantify the CPU increase for every possible combination of conditions and actions. For network planners and operations teams planning to use Mediation, this document provides guidance on the impact of given Mediation template and its associated rules.

While the system allows a maximum of 64 active templates, operators should be cautious about activating templates as it may impact performance. [Assessing a Rule Template for Safe Use](#) provides steps an operator can use to ensure that DSR can continue to support the capacity it has been deployed for, along with the planned mediation templates/rules.

Note

Operators on DSR releases 7.1.x to 7.x.x releases are required to run an additional MOP to unlock the number of templates that can be activated. [Raising the Limit of Active Rule Templates](#) provides steps to increase the number of templates.

Note

There is no performance impact to the DSR for having templates in the Development or Test state or for having a template in the Active state but not assigned to a trigger point.

The Diameter Routing Layer (DRL) request task (DRLRequestTask) and the DRL answer task (DRLAnswerTask) are processes that run on the DSR and are the work horses in the routing layer. These tasks are also responsible for all mediation related activities. Specifically, the DRLRequestTask, among other things, is responsible for any mediation performed on Request messages and therefore the performance impact of using a template at one of the Request Trigger Points is reflected in the %CPU usage of the DrlRequestTask processes.

Similarly, the DRLAnswerTask is responsible for any mediation performed on the Answer messages and the corresponding performance impact is reflected in %CPU usage of the DrlAnswerTask processes. It should be noted that on any given system, there are multiple instances of the DRLRequestTask and DRLAnswerTask processes.

[Assessing a Rule Template for Safe Use](#) the steps to monitor these processes and the permissible limits under which they should remain to ensure that the DSR can support the traffic it was engineered for.

2.2.1.1 Assessing a Rule Template for Safe Use

Follow this procedure to assess if it is safe to use a mediation template. Perform this procedure first during low traffic periods, and then repeat it during both moderate and peak load periods. Running this procedure does not impact the performance of the system.

1. Use **SSH** to access the DA-MP where the rule template is activated.
2. Activate the template and assign rules to the template.

Note

Do not assign the template to a trigger point at this time. Assigning the template to a trigger point impacts the performance of the system.

3. Run the top command to show the **CPU** usage of the DrlRequestTask DrlAnswerTask processes:

```
top -H -p `pidof dsr`
```

Figure 2-2 Top Command Example Output

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
31841	root	10	-10	6521m	3.2g	279m	S	13.7	13.7	89:36.28
DrlRequestTask										
31837	root	10	-10	6521m	3.2g	279m	S	13.0	13.7	89:40.01
DrlRequestTask										
31839	root	10	-10	6521m	3.2g	279m	S	13.0	13.7	89:35.54
DrlRequestTask										
31835	root	10	-10	6521m	3.2g	279m	R	12.7	13.7	89:46.33
DrlRequestTask										
31847	root	10	-10	6521m	3.2g	279m	S	6.3	13.7	40:55.98
DrlAnswerTask										
31845	root	10	-10	6521m	3.2g	279m	R	6.0	13.7	41:00.14
DrlAnswerTask										
31849	root	10	-10	6521m	3.2g	279m	S	6.0	13.7	40:57.99
DrlAnswerTask										
31803	root	10	-10	6521m	3.2g	279m	S	3.0	13.7	22:49.83
CsLRx										

Note

The top command continues to run showing real time updates to CPU usage until it is explicitly terminated

4. Monitor the "%CPU" usage of the respective threads.
 - a. If the template is assigned to a Request Trigger point, monitor the "%CPU" value associated with the DrlRequestTask process.
 - b. If the template is assigned to a Answer Trigger point, monitor the "%CPU" value associated with the DrlAnswerTask process.
5. Assign the template to the appropriate trigger point
6. Continue to monitor the "%CPU" usage of the respective threads as performed in [4](#)

In a geo-redundant setup (where traffic from the mate site can double in case of the mate site failure), operators can add templates or rules as long as the "%CPU" value on any of the DrlRequestTask or DrlAnswerTask processes does not exceed 40%.

When deployed in a non-geo-redundant setup, operators can add templates or rules as long as the "%CPU" value on any of the DrlRequestTask or DrlAnswerTask processes does not exceed 80%.

2.2.1.2 Raising the Limit of Active Rule Templates

Use this procedure to raise the maximum number of active rule templates to 64. Repeat these steps on every **SOAM**.

1. Use **SSH** to access the active **SOAM**.
2. Verify the current limit value using these commands:

```
iqt -L CapmOptions where "varId='CAPM_MaxActiveTasks'"
iqt -L CapmOptions where "varId='CAPM_MaxAssignedActiveTasks'"
```

3. Execute these commands to set new limits, replacing <NUMBER> with the desired value:

```
iset -fvalue=<NUMBER> CapmOptions where "varId='CAPM_MaxActiveTasks'"
iset -fvalue=<NUMBER> CapmOptions where
"varId='CAPM_MaxAssignedActiveTasks'"
```

4. Verify the limits have been increased using the same commands from step 2:

```
iqt -L CapmOptions where "varId='CAPM_MaxActiveTasks'"
iqt -L CapmOptions where "varId='CAPM_MaxAssignedActiveTasks'"
```

2.2.2 Rule Template elements

A Rule Template is created by configuring [Table 2-6](#), [Table 2-7](#), [Table 2-8](#), and [Table 2-9](#).

Settings

The Settings are the main Rule Template properties.

Table 2-6 Settings

Element	Description	Data Input Notes
Rule Template Name	Name used to label this Rule Template in this application. This field is required.	Format: a-z, A-Z, 0-9, -, ., @, and _ (Unset cannot be used as a Rule Template Name.) Range: 1-255 characters
Message type support	Indicates the type of message processing that is supported by the Rule Template (Request, Answer, or both). The Message Support Type depends on the selected conditions and actions.	Format: Check marks Range: Request, Answer, or both are checked. Default: Both are checked This field cannot be edited.

Conditions

The Conditions define a set of one to five matching expressions.

All conditions are supported by both requests and replies. Each condition is marked with a letter in the alphabetical order (A, B, C, D, E, and so on). Use the up and down arrows to move

the conditions within the Condition Set; however, the letters that label the conditions stay in alphabetical order.

Table 2-7 Conditions

Element	Description	Data Input Notes
Fast Search	If check marked, fast database lookup is used. Otherwise, the values of the specified field are checked one-by-one until the first match is found. See Fast Search. All Conditions with the Fast Search option checked must precede the others to maintain the Fast Search. When the Default value is Fixed, Fast Search is enabled regardless of the selected Operator and right value type.	Format: check mark or not sign; not editable Range: checked or unchecked Default: checked
* Name	The name for the left value to display for a Condition on the Rule Set page. This field is required.	Format: field Range: 1 to 64 characters
Description	The description that appears for a Condition on the Rule Sets page. If possible, provide information such as the format to be used (such as text string or telephone number format) and the range of values (such as 1 to 255 characters).	Format: field Range: 1 to 255 characters string
* Left value	The left value in a Condition. The left value typically refers to a regular or grouped AVP component (AVP header parts or value) or a Diameter Header component. The grouped AVP can be up to 8 levels deep. Grouped AVPs that have a depth of one are supported (one or more AVPs at the same level within an AVP). This field is required. The value can be defined using the Formatting String Wizard Specifiers.	Format: field Range: See Formatting String Wizard Specifiers
Operator	Operator is used to compare left value and right value in a Condition. Exist and not exist operators are used to check the presence of the specified left value. Is true and is not true operators are used to verify whether the specified left value is not 0 or equals 0 (is empty in the case of a string type).	Format: list Range: See Table 2-13 Default: equals (==)

Table 2-7 (Cont.) Conditions

Element	Description	Data Input Notes
Case Sensitive	The checkbox is enabled for the UTF8String right value. Case-sensitive search is possible only together with Fast Search. Without Fast Search, the lookup is always case-insensitive.	Format: checkbox Range: checked or not checked Default: not checked (not case-sensitive)
Right Value	The type of data that is compared to the field in the message (specified by the left value) in a Condition to determine if there is a match and Mediation should be performed. The right value can be: - Empty - the optional checkbox is checked (it can be left empty in the rule provisioning in a Rule Set), or the right value is not used by the selected Operator (such as exists). - One of the right value types shown in the Range list. Actual data of the specified type is entered in a rule in the Rule Set that is generated from the Rule Template, to use in the comparison. - An actual data value of the selected right value type, provisioned in the Default value field of the Condition in the Rule Template.	Format: list Range: Right value types are: - Integer32 - Integer64 - Unsigned32 - Unsigned64 - Float32 - Float64 - Address (IPv4 or IPv6 IP address) - Time (number of seconds since 0h on 1 January 1900) - UTF8string - DiameterIdentity (FQDN or Realm) - DiameterURI - IP/Netmask (IPv4 or IPv6 Netmask) - Enumerated (available Enum values; prefaced by enum:) - OctetString - xl-value (references to AVPs, LAVPs, or parts of the Diameter message) - Regular expression (Perl 5 regular expression) - Unsigned64Range enter two numbers (the lower and upper limit of the range inclusive) that fit into Unsigned64, separated by a dash (-). - Connection provisioned connection - Peer provisioned peer nodes Default: Integer32 All previously provisioned Enumerated Types are listed prefixed with enum:. For example: enum: xyz.
Default value	An actual data value to display for the right value of a Condition on the Rule Set page. When the Default value is Fixed, Fast Search is enabled regardless of the selected Operator and right value type.	Format: Text Range: Data value that is valid for selected right value type. When OctetString or UTF8String is selected, any human-readable character is valid. When the xl-value type is selected, all Default value entries must be xl-values.

Table 2-7 (Cont.) Conditions

Element	Description	Data Input Notes
Optional	The Optional checkbox can be checked so that the right value data could be deleted or left empty in the Rule Set rule, or unchecked indicating that the right value data must be entered and can be changed in the Rule Set rule.	Format: checkbox Range: Check mark or no check mark Default: Checked
Fixed	Indicates that the right value data that is entered in the Default value in the Rule Template Condition is actual data, and cannot be changed in the Rule Set rule.	Format: checkbox Range: Check mark or no check mark Default: Not checked

Condition Set

The Condition Set specifies whether the conditions are logically ANDed, ORed or they form a Complex Expression.

Table 2-8 Condition Set

Element	Description	Data Input Notes
Condition Set	The defined matching expressions (Conditions) are combined in one logical expression (Condition Set) and can be AND, OR or represent Complex Expression operators, so the set matches on the message if all the expressions are true. If no matching expression is defined, the message unconditionally matches.	Format: Options Range: ANDed, ORed or Complex Expression.

Actions

The Actions specify the possible settings for each action to be taken for this Rule Template.

Use the up or down arrows to move the order of the Actions.

Table 2-9 Actions

Action	Description	Data Input Notes
New Action	Add a new Action to the list that is applied when the conditions of the Rule Template match on the message.	Format: list Range: Actions listed in this section of this table.

The [Table 2-10](#) allows the Diameter Header and Set Command Flags to be modified.

Table 2-10 Actions Performed on the Diameter Header

Action	Description	Data Input Notes
Modify Diameter Header Parts	Allows modifying or overwriting of the Version, Command Code, and Application ID components of the Diameter Header. Note: Modifying values in the Diameter Header can result in incompatibility with the standard defined in IETF RFC3588bis (draft-ietf_dime_rfc3588bis-26.txt) <i>Diameter Base Protocol</i> .	Header Part - the component to modify Format: list Range: Version, Command Code, Application ID Default: Version Overwrite to - the new value of the component Format: Integer Range: New value; 8-bit, 24-bit, or 32-bit unsigned integer Optional Format: Checkbox Range: Check mark or no check mark
Set Command Flags	Allows modifying of one or more Command Flags in the processed message, including the reserved flags: - Set Command Flag - Clear Command Flag - Keep Original value Flags R, P, E, and T are supported; r4, r5, r6, and r7 are reserved for future use: - R - Request; shows whether the message is a Request or a Response. - P - Proxiable; shows if the message can be proxied, relayed, or redirected, or it must be locally processed. - E - Error; shows if the message contains protocol or semantic errors. - T - Shows that a message can potentially be a retransmitted message after a link fail-over, or is used to aid removal of duplicate messages.	Set Command Flag: Clear Command Flag: Keep original: Default: Keep original Format: Options Range: R, P, E, T, r4, r5, r6, r7 Optional is not available

The Actions performed on the Diameter Payload (AVPs) can be applied to a regular AVP, to a Grouped AVP, or to an AVP within the Grouped AVP.

To perform the action on a regular or Grouped AVP, the supported AVP definition from the dictionary and the instance number or value must be specified. The value is valid only for some of the actions.

For actions that are performed on an AVP within a Grouped AVP, the parent AVP and its instance number must be specified.

If an AVP is not present in the dictionary, it is unknown by the Mediation feature and must be defined in the dictionary before the specified action can be performed.

Many of the actions allow xl-values, which can be defined using the [Formatting String Wizard Specifiers](#).

[Table 2-11](#) describes available Diameter Payload actions.

Table 2-11 Actions Performed on the Diameter Payload (AVPs)

Action	Description	Data Input Notes
Add AVP	This action can support up to 8 levels deep AVPs. Note: The action fails, if the specified instance of the parent AVP is also added but the previous instances of this AVP do not exist in the message. The Flags and the Value must be set for the new AVP. For Grouped AVPs, - If the AVP is added within a Grouped AVP, the Parent AVP and its Instance must be specified. - A Parent AVP can be added if it not present in the message; Flags for the added Parent AVP must be set. - If the Parent AVP is not found in the message and it is not added to the message, the action fails. Flags V, M and P are supported; r3, r4, r5, r6 and r7 are reserved for future use. - V - Vendor-Specific; indicates whether the optional Vendor-ID field is present in the AVP header. When set, the AVP Code belongs to the specific vendor code address space. - M - Mandatory; indicates whether support of the AVP is required. If an AVP with the M bit set is received by a Diameter client, server, proxy, or translation agent and either the AVP or its value is unrecognized, the message MUST be rejected. Diameter Relay and Redirect Agents MUST NOT reject messages with unrecognized AVPs. AVPs with the M bit cleared are informational only. A receiver of a message with an AVP that is not supported, or whose value is not supported, can simply ignore the AVP. - P - Indicates the need for encryption for end-to-end security. Diameter base protocol specifies which AVPs must be protected by end-to-end security measures	Add parent AVP if it is not present: Format: Checkbox Range: Check mark or no check mark Default: No check mark AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of Integer32, Integer64, Unsigned32, Unsigned64 type. Set Flags: Format: checkbox for each flag Range: V, M, P, r3, r4, r5, r6, r7 Set Value: Format: Specify the AVP value in Formatting String Wizard Specifiers. Optional Format: Checkbox Range: Check mark or no check mark

Table 2-11 (Cont.) Actions Performed on the Diameter Payload (AVPs)

Action	Description	Data Input Notes
Set/Add AVP	(encryption) if the message is to pass through a Diameter agent. If a message includes any of those AVPs, the message must not be sent unless there is end-to-end security between the originator and the recipient of the message. The action allows changing the value of an AVP if it is found in the message or adding an AVP if it is not found in the message. This action can support up to 8 levels deep AVPs. It searches for the specific instance of an AVP in the message, overwriting its value when this AVP is found or appending this AVP (adding it as a last instance) when it's not present. The AVP can be looked up in the message either by the instance number or by the value. If the AVP is not found in the message it can be appended, which requires setting the Flags and specifying the Value of the AVP that is to be added. If a non-existing AVP that was looked up in the message by the specific instance number is about to be added to the message with the different instance number, the action fails.	AVP: Format: list Range: Available AVPs Instance Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of Integer32, Integer64, Unsigned32, Unsigned64 type. Default: First Set Flags Flag definitions in Add AVP: • If the flag must be set, the flag is checked and disabled. • If the flag must not be set, the flag is unchecked and disabled. • If the flag can be set, the checkbox is available to be changed. Format: checkboxes for the flags Range: V, M, P, r3, r4, r5, r6, r7 If the AVP is found, set its value (all specified flags are ignored) Format: Options Range: Search by the instance or Search by the value Else append the AVP to the message (all specified flags are considered). Add parent AVP if it is not present Format: checkbox Range: Checked or unchecked Set/Add Value Format: Value entered Range: Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-11 (Cont.) Actions Performed on the Diameter Payload (AVPs)

Action	Description	Data Input Notes
Delete AVP	Delete a specified AVP in the message. This action can support up to 8 levels deep AVPs. If the Instance of the specified AVP is All, the action is applied to all instances of the AVP or Grouped AVP in the message. If the specified AVP is within a Grouped AVP, the Parent AVP and its Instance must be specified. If the specified AVP is the last AVP within the Grouped AVP, the action can be defined to delete the Parent AVP. If the specified AVP is a Grouped AVP, the Grouped AVP and all of the AVPs within the group are deleted. If the deleted AVP has been the last AVP within the Grouped AVP, then Delete parent AVP if it is empty can be checked to delete the Parent AVP as well. If the specified AVP is not found in the message, the Delete AVP action is considered successful.	Delete parent AVP if it is empty: Format: Checkbox Range: Check mark or no check mark Default: Checked AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of Integer32, Integer64, Unsigned32, Unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Optional Format: Checkbox Range: Check mark or no check mark
Save AVP	A saved AVP is stored in the buffer as long as the transaction exists. Saved AVPs can be accessed through the Formatting String Wizard Specifiers as corresponding Linking-AVPs with the same AVP and instance number. If the Instance of the specified AVP is All, the action saves all instances of the AVP in the message. Note: A grouped AVP can be saved and restored, but sub-AVPs within the stored or restored grouped AVP cannot be retrieved (such as with @msg.avp["name"][index].avp["name"][index]), modified, or removed. If the same AVP is saved multiple times (the action is applied multiple times), the saved value is overwritten each time the AVP is saved. If the specified AVP is not found in the message, the Save AVP action is considered to have failed.	Save AVP: Format: list Range: Available AVPs Instance: First, Second, Third, Fourth, Fifth, All, all internal variables of integer32, integer64, unsigned32, unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: checkbox Optional Format: checkbox Range: Check mark or no check mark

Table 2-11 (Cont.) Actions Performed on the Diameter Payload (AVPs)

Action	Description	Data Input Notes
Restore AVP	AVPs can be restored in the message by either appending each AVP to the message or by replacing all of the same existing AVPs. The instance number of the saved AVP must be specified, to find the appropriate Linking-AVP (LAVP) that was stored. Note: A Grouped AVP can be saved and restored, but sub-AVPs within the stored or restored Grouped AVP cannot be retrieved (such as with @msg.avp["name"][index].avp["name"][index]), modified, or removed.	Delete before restore: Format: checkbox Range: Checked, unchecked; default is unchecked. Restore AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, All, all internal variables of integer32, integer64, unsigned32, unsigned64 type. Optional Format: Checkbox Range: Check mark or no check mark

Table 2-11 (Cont.) Actions Performed on the Diameter Payload (AVPs)

Action	Description	Data Input Notes
Set LAVP	Allows constructing a top-level non-Grouped AVP by setting the Flags and specifying the value, and placing it into the buffer associated with the Diameter transaction. The AVP can be accessed as a Linking-AVP through the Formatting String Wizard Specifiers. The value is stored in the buffer as long as the transaction exists. The LAVP can be used for the Restore AVP action. - Instance - Specify the instance number of the AVP within the buffer of the transaction. The new AVP specification always overwrites the existing if there is already one with the same instance number. - Set Value - The Input field is available when the selected LAVP has a data format other than Enumerated. - The list is available when the selected LAVP has the data format Enumerated. - An error message appears if the entered value of the Input field is not an x1-value and does not correspond to the data format required by the selected AVP. - Delete: Click to delete an existing Linking-AVP.	Set LAVP Format: list Range: All non-Grouped AVPs from the dictionary Default: First non-Grouped AVP definition from the dictionary Instance Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. Default: First Set Flags Flag definitions in Add AVP: - If the flag must be set, the flag is checked and disabled. - If the flag must not be set, the flag is unchecked and disabled. - If the flag can be set, the checkbox is available to be changed. Format: checkboxes for the flags Range: V, M, P, r3, r4, r5, r6, r7 Default: From the dictionary Set Value: Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Actions that modify an AVP

If the specified AVP is not found in the message, the action is considered to have failed.

[Table 2-12](#) describes Modify AVP actions.

Table 2-12 Actions that Modify an AVP

Action	Description	Data Input Notes
Change AVP Code	Replace an AVP definition with a new one, keeping the original AVP value and flag that are not strictly defined in the dictionary (that can be set). This action can support up to 8 levels deep AVPs. Allows changing the Code of the specified AVP and modifying its Flags.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. AVP: Format: list Range: Available AVPs Instance: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox New AVP Format: list Range: Available AVPs Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Change AVP Flags	Allows setting, clearing, and keeping the original value of AVP flags. This action can support up to 8 levels deep AVPs. Flags V, M and P are supported; r3, r4, r5, r6 and r7 are reserved for future use. • V - Vendor-Specific; indicates whether the optional Vendor-ID field is present in the AVP header. When set, the AVP Code belongs to the specific vendor code address space. • M - Mandatory; indicates whether support of the AVP is required. If an AVP with the M bit set is received by a Diameter client, server, proxy, or translation agent and either the ABP or its value is unrecognized, the message MUST be rejected. Diameter Relay and Redirect Agents MUST NOT reject messages with unrecognized AVPx. AVPs with the M bit cleared are informational only. A receiver of a message with an AVP that is not supported, or whose value is not supported, can simply ignore the AVP. • P - Indicates the need for encryption for end-to-end security. Diameter base protocol specifies which AVPs must be protected by end-to-end security measures (encryption) if the message is to pass through a Diameter agent. If a message includes any of those AVPs, the message must not be sent unless there is end-to-end security between the originator and the recipient of the message.	Parent AVP Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. AVP: Format: list. Range: Available AVPs With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Set Command Flag: Clear Command Flag: Keep original: Default: Keep original Format: Options Range: R, P, E, T, r4, r5, r6, r7 Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Set AVP Value	Allows overwriting of the value of an AVP. This action can support up to 8 levels deep AVPs.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. AVP: Format: list. Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Set Value: Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Strip from AVP Value	Strips the defined number of characters from either the beginning or the ending of the AVP value. This action can be used in combination with the Prefix/Suffix to AVP Value action. This action can support up to 8 levels deep AVPs.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type AVP: Format: list Range: Available AVPs Instance: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Strip from: Format: Options Range: Beginning of the value or End of the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Prefix/Suffix to AVP Value	Add the defined data as a prefix or suffix to the AVP value. This action can be used in combination with the Strip for AVP Value action. This action can support up to 8 levels deep AVPs.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type AVP: Format: list Range: Available AVPs Instance: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Options: Prefix to the value, Suffix to the value Prefix to the value or Suffix to the value: Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Substitute in AVP Value	Use a defined pattern to locate a field in the AVP value, and replace the data in the field with the specified new data. This action can support up to 8 levels deep AVPs.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type AVP: Instance: First, Second, Third, Fourth, Fifth, all internal variables of integer32, integer64, unsigned32, unsigned64 type. With the value: Specify the AVP value in Formatting String Wizard Specifiers Format: Checkbox Pattern: Format: Text Range: Pattern to locate the field Replacement: Format: Text Range: Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Bit Set/Reset	Allows setting/resetting the specified bit of the AVP. This action can support up to 8 levels deep AVPs.	Parent AVP: Format: list Range: Available AVPs Instance: Format: list Range: First, Second, Third Fourth, Fifth and all internal variables of integer32, integer64, unsigned32, unsigned64 type. Default: First non-Grouped AVP definition from the dictionary Bit Position: Note: Disabled unless the AVP is selected. Format: list Range: Bit indexes between 0-31 or 0-63 depending on the format of an AVP. Default: First non-Grouped AVP definition from the dictionary Range: First, Second, Third Fourth, Fifth and all internal variables of integer32, integer64, unsigned32, unsigned64 type. Set/Reset Value: Note: Disabled unless the AVP is selected. Format: list Range: 1 or 0 - Default a 1 Note: If the name of the selected AVP is too long and does not fit on the page, the hint message is available; it displays the entire name. Optional Format: Checkbox Range: Check mark or no check mark
Set internal variable	Allows setting the value for an internal variable that is valid for the entire duration of a transaction.	Internal Variable Format: list Range: Available configured Internal Variables. Set Value Specify the AVP value in Formatting String Wizard Specifiers Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Message Copy	Trigger Diameter Message Copy for the message, based on the values in the Message Copy Configuration Set that is specified for the Action. See Configuration in the Diameter User's Guide.	Select Message Copy Configuration Set Format: list Range: Default or Message Copy Default: Blank Optional Format: Checkbox Range: Check mark or no check mark
Set Request Priority	Allows setting the Priority of a Request to Max Normal Request Priority, which can be used by congestion controls. The action is not applicable for Answer messages.	Request Priority Format: list Range: 0 - Max Normal Request Priority as configured in the system options. Note: Set Request Priority is mutually exclusive with this action. Optional Format: Checkbox Range: Check mark or no check mark
Set Route List	Allows setting the Route List to bypass the PRT. This is used when the Request has to be routed based on AVPs that are beyond the ones accessible via the Peer Route Tables. The action can also enforce skipping the Application Route Tables matching. The action is applicable only for the following options: • Diameter request message received from connection • Diameter request message ready to be forwarded to application • Diameter request message received from application trigger points It is ignored at all other trigger points. Note: If the ART is not skipped, then it is possible that an application can specify a PRT, which can result in the overwriting of the Route List specified by mediation.	Set Route List Format: list Range: Pre-defined Route List. Skip Application Route Tables Format: Checkbox Range: Check mark or no check mark Note: Abandon Message and Send Answer are mutually exclusive with this action. Optional Format: Checkbox Range: Check mark or no check mark
Peg Counter	Allows the user to peg the pre-defined measurements from the list in Diameter , and then Mediation , and then Measurements . A report is generated from Measurements , and then Report .	Measurement Format: list Range: Pre-defined measurements based on the list in Diameter , and then Mediation , and then Measurements . Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Send Answer	Allows answering the request with a configurable Result-Code. The action enables specifying the Result-Code or Experimental-Result-Code and Vendor-ID. The action is supported by Requests only	Answer Result-Code Value Format: Options Vendor-ID Format: Text Range: 1 - 4294967295 Answer Error Message Format: Text Range: N/A or specify the message placed in the Error-Message AVP in Answer message. Note: Abandon Message and Send Answer are mutually exclusive with this action. Exit from Execution Trigger Bypasses any subsequent Rule Template in it. Optional is not available for this Action.
Abandon Message	Allows to silently drop the request. The action is supported by Requests and Answers at the DCL trigger points. The DRL answer triggers (ATP1, ATP4, ATP6 and ATP10) ignore this action.	Format: Blank Range: Blank Note: Send Answer and Set Route List are mutually exclusive with this action. Exit from Execution Trigger Bypasses any subsequent Rule Template in it. Optional is not available for this Action.
Set Destination-Realm	This action populates Destination-Realm AVP based on the IMSI value present in the Request.	Format: Blank Range: Blank Note: Process Decorated NAI is mutually exclusive with this action. Exit from Execution Trigger Bypasses any subsequent Rule Template in it. Optional is not available for this Action.

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Process Decorated NAI	Performs the following actions: • Inspects the User - Name AVP for the presence of Decorated NAI, if not present, then the action stops. The request continues processing. • If present, extracts the realm value from the Decorated NAI and populates the Destination-Realm AVP. For more information see RFC 5729 and RFC 4282. • The action also modifies the User-Name AVP value. For more information, see RFC 4282. If Destination-Realm is deleted/absent during request processing, the action adds it back.	Format: Blank Range: Blank Note: Set Destination-Realm is mutually exclusive with this action. Optional is not available for this Action.

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Redirect Request-Host	This action generates an Answer message with a Result-Code value of 3006 and the (E(rror)) E bit set. The user can specify one of the following options for the Redirect-Host AVP: - Do not include the Redirect-Host AVP - The Redirect-Host AVP is prefixed with an aaa:// and set to the value of Destination-Host AVP from the request or populated manually. If the Destination-Host AVP is not present in the request, the Redirect-Host AVP is not included in the 3006 response - Include the Redirect-Host AVP with a user provisioned value (String Wizard available) Two additional redirect types can be added to the generated Answer message by checking Add: - Redirect-Host-Usage - if the user selects any non-zero value for this AVP the redirection should be cached (zero = DONT_CACHE) - Redirect-Max-Cache-Time - number of seconds to cache the redirection (if Redirect-Host-Usage equals zero, then it recommended that this AVP be zero). Note: If either of the Additional AVPs exist, then both AVPs must exist. Note: If the Redirect-Host is not included in the response, then the Redirect-Host-Usage and Redirect-Max-Cache-Time is not included Note: If Do not include Redirect-Host is selected, then the Add checkbox is disabled.	Do not include Redirect-Host AVP Format: Option Range: No Range available Include Redirect-Host AVP with the value of Destination-Host from the request with the aaa:// prefix Format: Option Range: No Range available Include Redirect-Host AVP with the value: Format: Option Range: Specify the AVP value in Formatting String Wizard Specifiers. Add Specify if the extra AVPs have to be added to 3006 response. Format: Checkbox Range: Check to disable or no check mark Default: Unchecked Redirect-Host-Usage AVP Format: list Range: DONT_CACHE, ALL_SESSION, ALL_REALM, REALM AND APPLICATION, ALL_APPLICATION, ALL_HOST, AND ALL_USER Default: DONT_CACHE Redirect-Max-Cache-Time AVP Format: Text Range: Specify the AVP value in Formatting String Wizard Specifiers Default: 0 Exit from Execution Trigger Bypasses any subsequent Rule Template in it. Optional is not available for this Action.

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Redirect Request-Realm	This action generates an Answer message with a Result-Code value of 3011 and the (E(rro)r) E bit set. The user can specify one of the following options for the Redirect-Realm AVP: - Do not include the "Redirect-Realm" AVP - Include the "Redirect-Realm" AVP with the value of "Destination-Realm" AVP from the Request or populated manually. If the "Destination-Realm" AVP is not present in the request, the "Redirect-Realm" AVP is not included in the 3011 response. - Include the Redirect-Realm AVP with a user provisioned value (String Wizard available) Two additional redirect avps can be added to the generated Answer message by checking the Add box: - Redirect-Realm-Usage - if the user selects any non-zero value for this AVP the redirection should be cached (zero = DON'T_CACHE) - Redirect-Max-Cache-Time - number of seconds to cache the redirection (if Redirect-Realm-Usage equals zero, then it recommended that this AVP be zero). Note: If either of the Additional AVPs exist, then both AVPs must exist. Note: If the Redirect-Realm is not included in the response, then the Redirect-Realm-Usage and Redirect-Max-Cache-Time is not included Note: If Do not include Redirect-Realm is selected, then the Add checkbox is disabled.	Do not include "Redirect-Realm" AVP Format: Option Range: No Range available Include "Redirect-Realm" AVP with the value of "Destination-Realm" from the request Format: Option Range: No Range available Include "Redirect-Realm" AVP with the value: Format: Option Range: Specify the AVP value in Formatting String Wizard Specifiers Add Specify if the extra AVPs have to be added to 3011 response. Format: checkbox Range: Check to disable or no check mark Default: Unchecked "Redirect-Realm-Usage" AVP Format: list Range: DONT_CACHE, ALL_SESSION, ALL_REALM, REALM AND APPLICATION, ALL_APPLICATION, ALL_HOST, AND ALL_USER Default: REALM AND APPLICATION "Redirect-Max-Cache-Time" AVP Format: checkbox Range: Specify the AVP value in Formatting String Wizard Specifiers Default: 3600 Exit from Execution Trigger Bypasses any subsequent Rule Template init. Optional Format: Checkbox Range: Check mark or no check mark

Table 2-12 (Cont.) Actions that Modify an AVP

Action	Description	Data Input Notes
Assert Alarm/Event	Define the severity of the alarm at template action. Allows the user to cause an Alarm or Event based on the Condition. The alarm instance value is automatically set to the Template-ID: Rule-ID. The alarms are throttled and auto cleared after 300 seconds.	Alarm/Event Format: list Range: Mediation Generic Event, Mediation Generic Alarm - Minor, Mediation Generic Alarm - Major, Mediation Generic Alarm - Critical Additional info - Write a short explanation to distinguish between different problems. Type in the text box or use the Wizard to write xl-expressions, which are resolved at the run time. Optional Format: Checkbox Range: Check mark or no check mark
Execute Rule template	The value needs to be set at the time the new Rule Template is defined. Only Rule Templates in Test or Active state are listed. This field is displayed on the Diameter Mediation Rule Template Insert and Edit pages, but not on the View page.	Format: list Range: Available Rule Templates in Test and Active states. Optional is not available for this Action.
Exit from Execution Trigger	Exits from the Execution Trigger, bypassing any subsequent Rule Set associated with it.	Optional is not available for this Action.
Execute Perl subroutine	The Execute perl subroutine can query the Diameter message headers and AVPs and change the Diameter message content, for example add, delete, change AVPs and get and set Internal Variables or Linking AVPs.	Script Format: Text Range: Name of the perl script Subroutine Format: Text Range: Specify the subroutine to execute. Optional is not available for this Action.

Rule Template Condition Operators

The [Rule Template Condition Operators](#) describes the operator that allows a comparison of the left-hand operand and the right-hand operand in a condition. The value can be an AVP, another part of a Diameter message, a constant, or an internal variable.

Table 2-13 Rule Template Condition Operators

Operator	Operator Type	Returns true when...
		Example of use
equals (==)	Generic	Value exists AND equals... @msg.command.code==316

Table 2-13 (Cont.) Rule Template Condition Operators

Operator	Operator Type	Returns true when...
		Example of use
does not equal (!=)	Generic	Value does not exist OR does not equal... @msg.command.code!=316
begins with (longest match) (=^^)	String	Value exists AND begins with (longest match)... @msg.avp["Destination-Realm"]=^^test
begins with (=^)	String	Value exists AND begins with... @msg.avp["Destination-Realm"]=^testlb
does not begin with (!=^)	String	Value does not exist OR does not begin with... @msg.avp["Destination-Realm"]!=^testlb
ends with (longest match) (=\$\$)	String	Value exists AND ends with (longest match) @msg.avp["Origin-Host"][1]=\$\$entity.com
ends with (=\$)	String	Value exists AND ends with... @msg.avp["Origin-Host"][1]=\$entity.com
does not end with (!=\$)	String	Value does not exist OR does not end with... @msg.avp["Origin-Host"][1]!=\$entity.com
regular expression match (=~)	String	Value exists AND matches the regular expression... @msg.avp[Session-Id]! =~.*\example\..*
regular expression does not match (!=~)	String	Value does not exist OR does not match the regular expression... @msg.avp["Session-Id"]! =~.*\example\..*
less than (<)	Numeric	Value exists AND is less than... @msg.avp["Validity-Time"]<30
greater than (>)	Numeric	Value exists AND is greater than... @msg.avp["Validity-Time"]>30
less than or equal to (<=)	Numeric	Value exists AND is less than or equal to... @msg.avp["Validity-Time"]<=30
greater than or equal to (>=)	Numeric	Value exists AND is greater than or equal to... @msg.avp["Validity-Time"]>=30
is within	Subnet	Value exists AND is within... @msg.avp["Served-Party-IP-Address"] is within 192.168.0.0/24

Table 2-13 (Cont.) Rule Template Condition Operators

Operator	Operator Type	Returns true when...
		Example of use
is not within	Subnet	Value does not exist OR is not within...
exists		@msg.avp["Served-Party-IP-Address"] is not within 192.168.0.0/24 AVP specified as Left value exists...
does not exist		@msg.avp["Vendor-Specific-Application"] exists AVP specified as Left value does not exist...
is true		@msg.avp["Vendor-Specific-Application"] does not exist AVP specified as Left value exists AND it is not empty/non-zero...
is false		@msg.avp["Disconnect-Cause"] is true AVP specified as Left value does not exist OR it is empty/0... @msg.avp["Disconnect-Cause"] is false

The is true and is false work only on numbers (Integer32, Integer32, Unsigned32, Unsigned64, Float32, Float64, Enumerated, Time) and strings (OctetString, UTF8String, DiameterIdentity, DiameterURI).

For an IP Address, is true always succeeds; the address can be converted to a string that is never empty.

If the condition cannot be evaluated (for example, the AVP does not exist or the xl-value is incompatible), then is true fails and is false succeeds.

Rule Template Condition Conversion Rules

Based on the type of operator selected, the left and right values are converted according to the rules in [Table 2-14](#).

Table 2-14 Rule Template Condition Conversion Rules

Left value Type	Operator Type	Right Value Type	Conversion
-	String	-	Convert left value and right value to strings.
-	Numeric	-	Convert left value and right value to numbers.
-	Subnet	-	Convert left value to an IP address. Convert right value to a subnet

Table 2-14 (Cont.) Rule Template Condition Conversion Rules

Left value Type	Operator Type	Right Value Type	Conversion
String	Generic	String	No conversion is needed.
Numeric	Generic	Numeric	No conversion is needed.
IP address	Generic	IP address	No conversion is needed.
String	Generic	Numeric	Convert left value to a number.
Numeric	Generic	String	Convert right value to a number.
IP address	Generic	String	Convert right value to an IP address.
String	Generic	IP address	Convert left value to an IP address.
None of these cases			Conversion cannot be done.
Operators by Type (see also Table 2-13)			
String	=~, !=~, =^, ^=^, !=^, =\$\$, =\$, !=\$		
Numeric	<, >, <=, >=		
Subset	is within, is not within		
Generic	==, !=		

The conversion fails if the input value is reasonably not convertible to the new format (such as the numeric input cannot be converted to an IP Address).

If the conversion is impossible or fails, the condition is evaluated to false unless the operator is negated (begins with !, or is not within).

For float to string conversion, the double argument is rounded and converted to decimal notation in the style [-]ddd.dddddd, with 6 characters of precision. If the conversion does not fit into 21 characters, then it fails.

For IPv6 to string conversion, the following rules apply:

- Leading zeros are ignored (01->1)
- Lowercase to uppercase (ffff->FFFF)
- 1:0:0:0:0:0:0->1:0:0:0:0:0:0
- 1::2->1:0:0:0:0:0:2
- ::ffff->0:0:0:0:0:0:0:FFFF
- ffff::->FFFF:0:0:0:0:0:0:
- ::->0:0:0:0:0:0:0:0

2.2.3 Adding a Rule Template

Meta-Administrator and Rule Set Administrator use this procedure to define a new Rule Template.

There are three sections of the **Diameter**, and then **Mediation**, and then **Rule Templates** [Insert] page: **Settings**, **Conditions**, and **Actions**. For a detailed description of each section,

see [Rule Templates](#). For a list of Rule Template elements and their definitions, see [Rule Template elements](#).

1. If Mediation - Triggered Message Copy is used (the Message Copy Action is selected in one or more Rule Templates), then the following **Diameter**, and then **Configuration** is required before configuring the actions:
 - Configure one or more **Route Groups** that used for one or more **Route Lists** for the Message Copy **DAS**.
 - Configure one or more **Route Lists** for the **DAS**.
 - Configure one or more **Configuration Sets**, and then **Message Copy** that can be assigned to one or more Message Copy Actions.

2. Verify the required **Enumeration** types, **Internal Variables**, **Measurements**, **AVP Dictionary entries**, and **Vendors** have been defined in the system.

Use the following GUI pages to view the entries, and to access the GUI pages to enter, change, or delete entries as needed:

- [Internal Variables](#)
- [Enumerations](#)
- [Measurements](#)
- All-AVP Dictionary in the *Diameter User's Guide*
- Vendors in the *Diameter User's Guide*

3. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.

4. Click **Insert**.

If the maximum number of Rule Templates (100) already exists in the system, the Rule Templates [Insert] page does not open, and an error message displays.

5. Enter the **Rule Template Name** in the Settings section. **Settings** values for the Rule Template. For a description of the setting elements, see [Table 2-6](#).

The **Message type support** depends on the selected conditions and actions.

6. Enter the **Conditions** values.

You can define up to five Conditions in the Rule Template. The grouped AVPs can be up to 8 levels deep. The rule template with no conditions is unconditionally matched for all processed messages.

Note

Enable the fast search option for as many conditions as possible as described in [Fast Search](#). For information on how fast search impacts the performance of DSR, see [Performance](#).

The order the conditions are in alphabetical order. To change the order use up and down arrows.

A check mark in **Optional** indicates a matching expression is optional. This means you can leave this condition's **Value** field blank on the **Diameter**, and then **Mediation**, and then **Rule Sets** page, and the condition is not used during message processing. For a description of the conditions elements, see [Table 2-7](#).

7. Select the **Condition Set** options as defined in [Table 2-8](#).

Note

The maximum number of provisioned rules per template/rule set depends on the template type. For slow-search templates, which have at least one slow-search condition, the maximum number of provisioned rules is 250. For fast-search templates, the maximum number of provisioned rules is 2000. To determine if the template is a fast-search reference [Fast Search versus slow search](#).

8. Select the **Actions** value as defined in [Table 2-9](#).

When any defined Conditions are met, the Actions specified in this section of the page are taken. At least one Action must be specified for a Rule Template.

9. When the Rule Template definition is complete, click:
 - **OK** and change the state of the rule template in **Diameter**, and then **Mediation**, and then **State & Properties**
 - **Apply**
 - **Cancel**
10. If you want to add online help to describe this Rule Template in its generated Rule Set, perform the [Adding online help to a Rule Template](#) procedure.
11. If one or more Rule Template Actions for Message Copy have been configured, then go to the **Diameter**, and then **Configuration**, and then **System Options** page, select the **Message Copy Options** tab, and select the **Enabled** option for the **Message Copy Feature** element.

2.2.4 Adding online help to a Rule Template

When a **Rule Template** is created, online help can be added to the Rule Template to describe it in its generated **Rule Set**.

After online help is added, when the user clicks the context-sensitive help icon in the upper right corner of the associated **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page, this online help explains how to use the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page to configure the Rule Set.

This help is standalone, and is not part of the online help provided by Customer Care Center as part of the Mediation feature.

Use this procedure to add context-sensitive online help to an existing Rule Template:

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Select the **Rule Template Name** to which online help will be added.
3. Click **Set Help**.
4. Change the contents of the **Title** box to an appropriate title for this help.

The Rule Template Name appears in the **Title** box as the default.

5. In the **Text** field, add specific details on how to configure a rule using this Rule Set, such as a procedure and result. You can also add detail on how the various fields interrelate, and provide any cautions to prevent loss of data.
6. To see how the help text you have entered looks from the Rule Set page, click **Preview**.

A separate window opens and displays the help text. Close the preview window when you are finished previewing the help text.

7. Click **OK**, **Apply** or **Cancel**.

The new help text is now available from the help icon on the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page for this Rule Template.

2.2.4.1 Rule Templates Help elements

When **Set Help** is clicked for an existing Rule Template on the Diameter Mediation Rule Templates page, the information in [Table 2-15](#) appears.

Table 2-15 Rule Templates Help Elements

Element	Description	Data Input Notes
Title	Title to appear at the top of the Help page. This field is required when providing Help.	Format: field Range: 1-64 characters
Text	Detailed explanation of this Rule Set: how to use it and description of any interrelated features.	Format: field (HTML tags allowed) Range: 1 - 1500 characters
Path	Generated and used by software	

2.2.5 Copying a Rule Template

Use this procedure to copy an existing Rule Template and save it as a new Rule Template. For a list of Rule Template elements and their definitions, see [Rule Template elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Select the **Rule Template** you want to copy.
3. Click **Copy**.

If the maximum number of Rule Templates (100) already exists in the system, the Rule Templates [Copy] page does not open, and an error message displays.

4. Enter a different **Rule Template Name** for the new Rule Template.
5. Make any changes as needed.
6. Click **OK**, **Apply** or **Cancel**.

2.2.6 Changing a Rule Template

Use this procedure to change values for an existing **Rule Template** (for a list of Rule Template elements and their definitions, see [Rule Template elements](#)).

When a Rule Template is in the development state, all elements can be changed.

After the Rule Template state has been changed to test or active, only the following elements can be changed:

- Rule Template Name
- Name of the Conditions
- Default value of the Conditions (except when the Fixed box has been checked)
- Description of the Condition
- Default value of the Actions (except for the value of the Execute Rule Template)

The Rule Template state must be set back to development to change any other elements (all provisioning of rules for the Rule Template are lost when the state is changed back to development), as defined in [State and Properties](#).

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Click **Edit** on the Rule Template row to be changed.
3. Change **Conditions** and **Actions** as needed.
4. Click **OK**, **Apply** or **Cancel**.

2.2.7 Importing a Rule Template

A Rule Template can be imported into the system using the **Import** function on the **Diameter**, and then **Mediation**, and then **Rule Templates** page.

Existing Rule Templates can be imported. Existing Rule Templates are previously defined Rule Templates that have been exported from Diameter Mediation using the **Export** function.

The Mediation version in the file selected for importing must be compatible with the system release into which the file is imported.

A successfully imported Rule Template file appears in the list on the **Diameter**, and then **Mediation**, and then **Rule Templates** page, the **Diameter**, and then **Mediation**, and then **State & Properties** page, and as a Rule Set in the **Diameter**, and then **Mediation**, and then **Rule Sets** menu folder (no Rule Set is generated if the only Action is Execute Rule Template).

The imported Rule Template is automatically set to the Test state.

The Enumeration types used in the Rule Template are imported, if they do not already exist in the system.

If the selected Rule Template references another Rule Template (as an Execute Rule Template action) that is not already present in the system, the referenced Rule Template is also imported, unless there is already a Rule Template with the same Name but a different definition.

Use the following procedure to import an existing Rule Template located outside of the file system:

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Click **Import**.
If the maximum number of Rule Templates (100) already exists in the system, the **Diameter**, and then **Mediation**, and then **Rule Templates [Import]** page does not open, and an error message displays.
3. Click **Browse** to open the Choose File popup window.
4. Locate of the Rule Template file you want to import, and select the file.
5. With the Rule Template filename displayed in the **File name** field, click **Open**.
6. Click **Import File**.

2.2.8 Exporting a Rule Template

Use this procedure to export a Rule Template from within the system to an external location.

The saved .xml file contains the following information:

- The **Rule Template** without any provisioned data

- All of the **Enumeration** type definitions with the possible values to which the Rule Template refers
- Mediation version number
- Help pages related to the Rule Template

Note

Export is not available (grayed out) for the Rule Set in the Development state (see [State and Properties](#)).

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Select the **Rule Template Name** row(s) for the Rule Template to be exported by checking the check boxes beside each Rule Template Name.
3. Click **Export**.
A File Download popup window appears.
4. Click **Browse** to open the Choose File window.
5. Locate where you want to export the Rule Template.
6. Click **Export File**.

2.2.9 Deleting a Rule Template

When a Rule Template is deleted, it is removed from the entire system, including the **Diameter**, and then **Mediation**, and then **State and Properties** page, the **Diameter**, and then **Mediation**, and then **Triggers** page, and the **Diameter**, and then **Mediation**, and then **Rule Sets** page.

Any **Rule Sets** generated from this **Rule Template** are also deleted automatically.

If a Rule Set belonging to the selected Rule Template is enabled for live traffic (Active state on the State and Properties page), an error message appears indicating the Rule Template cannot be deleted as long as it is being used by the system for live traffic.

Use this procedure to delete an existing Rule Template.

1. Click **Diameter**, and then **Mediation**, and then **Rule Templates**.
2. Select the **Rule Template Name** of the Rule Template to be removed.
3. Click **Delete**.

Note

A popup window appears to confirm the delete when the selected Rule Template is in the development state or the test state (see the State and Properties page).

4. Click **OK** or **Cancel**

Note

If the selected Rule Template has any data provisioned, another confirmation popup window appears indicating all of the provisioned data that belongs to any Rule Set generated from the Rule Template will be deleted.

5. Click **OK** or **Cancel**

2.2.10 Exporting all Rule Templates

Perform the following procedure to export all Rule Templates from within the system to the File Management Directory.

The saved .xml file contains the following information for each Rule Template:

- The Rule Template without any provisioned data.
- The Rule Template includes references to all enumeration type definitions and their possible values.
- Mediation version number.
- Help pages related to the Rule Template.

Note

Export all exports every Rule Template, including templates in the development state. For more information, see [State and Properties](#).

1. From the main menu, click **Diameter**, then **Mediation**, and then **Rule Templates**.
2. Click **Export All**. A popup window appears to confirm, click **OK**.
3. A task will be created to process the **Export All** in the background. The progress of the task may be monitored on the **Status & Manage**, and **Tasks**, and then **Active Tasks** page.

2.3 Formatting Value Wizard

The Formatting Value **Wizard** is a popup window available from both the **Diameter**, and then **Mediation**, and then **Rule Templates**, and then **[Insert]**, **[Edit]**, and **Copy** pages and the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **[Insert]** and **[Edit]** pages. The wizard simplifies entry of xl-formatted strings, which require specific syntax coding.

An xl-formatted string can contain references to the state of the server, or to the message being processed. For example, `%@ruri.user` refers to the user part of the Request URI within an xl-formatted string. The references are replaced with their actual values during a condition validation or an action execution.

Accessing request content during an Answer processing feature provides a read only access to all AVPs in the request message while processing an answer message to all answer trigger points.

2.3.1 Formatting String Wizard elements

[Table 2-16](#) describe the elements shown when **[wizard]** is clicked, from the **Diameter**, and then **Mediation**, and then **Rule Templates [Insert]** page.

Table 2-16 Formatting String Wizard Elements

Element	Description
Value	The value of the variable in xl-format. The components of this value can be entered manually by clicking on one or more specifiers, or both.
Specifiers	List of elements that can be part of an xl-formatted string. A specifier is either a single item or a group of items forming a sub-list. Every specifier selected is put into the Value field where the cursor is currently located. The specifiers are described in Formatting String Wizard Specifiers .
Preview	The readable description of the xl-formatted string in the Value field.

Formatting String Wizard Specifiers

[Table 2-17](#) describe the variables in the **Value** field in the Formatting String Wizard page.

Note

[Index] that is either a [<number>] or [any] can be excluded from all of the expressions that refer to the first instance of the **AVP**.
The instance number any can be present in the Left value of the Condition only once.

The instance number any can be present in the Right value of the Condition only once.

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description		
New Line	Sub-Items	xl-formatted Value	Preview Value
		\r\n	This causes a line break on the GUI page.
String Constant	Type the string constant	<i>"string constant"</i>	{"string constant"}
Diameter Header	Sub-Items	xl-formatted Value	Preview Value
	From the current messages being processed		
	Version	@msg.version	{Msg Version}
	Message Length	@msg.length	{Message Length}
	Command Flags: R	@msg.command.flags.R	{Msg R Command Flag}
	Command Flags: P	@msg.command.flags.P	{Msg P Command Flag}
	Command Flags: E	@msg.command.flags.E	{Msg E Command Flag}

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description		
	Command Flags: T	@msg.command.fl ags.T	{Msg T Command Flag}
	Command Flags: r4	@msg.command.fl ags.r4	{Msg r4 Command Flag}
	Command Flags: r5	@msg.command.fl ags.r5	{Msg r5 Command Flag}
	Command Flags: r6	@msg.command.fl ags.r6	{Msg r6 Command Flag}
	Command Flags: r7	@msg.command.fl ags.r7	{Msg r7 Command Flag}
	Command Code	@msg.command.co de	{Msg Command Code}
	Application ID	@msg.applicatio n_id	{Msg Application ID}
	Hop-by-Hop Identifier	@msg.hbh_id	{Msg Hop-to-Hop Identifier}
	End-to-End Identifier	@msg.e2e_id	{Msg End-to-End Identifier}
	From the request during the answer processing		
	Version	@req.version	{Req Version}
	Message Length	@req.length	{Request Length}
	Command Flags: R	@req.command.fl ags.R	{Req R Command Flag}
	Command Flags: P	@req.command.fl ags.P	{Req P Command Flag}
	Command Flags: E	@req.command.fl ags.E	{Req E Command Flag}
	Command Flags: T	@req.command.fl ags.T	{Req T Command Flag}
	Command Flags: r4	@req.command.fl ags.r4	{Req r4 Command Flag}
	Command Flags: r5	@req.command.fl ags.r5	{Req r5 Command Flag}
	Command Flags: r6	@req.command.fl ags.r6	{Req r6 Command Flag}
	Command Flags: r7	@req.command.fl ags.r7	{Req r7 Command Flag}
	Command Code	@req.command.co de	{Req Command Code}
	Application ID	@req.applicatio n_id	{Req Application ID}
	Hop-by-Hop Identifier	@req.hbh_id	{Req Hop-to-Hop Identifier}
	End-to-End Identifier	@req.e2e_id	{Req End-to-End Identifier}
AVP	Sub-Items		
	AVP List containing all AVP definitions from the base and custom dictionaries. If the AVP selected is of Type Group, then the list contains only those AVPs belonging to the ParentAVP.		

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description
	AVP Component List containing the following components: • <i>Data</i> • <i>Data_Length</i> • <i>AVP Code</i> • <i>Flag V</i> • <i>Flag M</i> • <i>Flag P</i> • <i>Flag r3</i> • <i>Flag r4</i> • <i>Flag r5</i> • <i>Flag r6</i> • <i>Flag r7</i> • <i>Vendor-ID</i> Flags V, M, and P are supported; flags r3, r4, r5, r6, and r7 are reserved for future use.
	AVP Instance number List the indexes of AVP (<i>First, Second, Third, Fourth, Fifth, Any</i>). If <i>Any</i> is selected in the AVP instance number list, then the Store matched instance into list appears. It contains all provisioned internal variables that are of <i>integer32</i>, <i>integer64</i>, <i>unsigned32</i>, and <i>unsigned64</i> type.
	Sub-type Specifically used with AVP Visited-PLMN-Id List containing types: • <i>mccmnc</i> • <i>mcc</i> • <i>mnc</i>
	Sub-type Specifically used with AVP User - Name List containing types: • <i>realm</i> • <i>user</i> • <i>realm_decoration</i> • <i>user_decoration</i> • <i>stripped_decoration</i> • <i>proper_user</i> • <i>imsi</i> • <i>imsi.mccmnc</i> • <i>imsi.mcc</i> • <i>imsi.mnc</i>
	Store matched instance into The matching instance value can be saved into any if the <i>integer types</i>, <i>signed/unsigned 32/64 bit</i>.

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description
Table 2-18 From the Current Messages Being Processed	
xl-Formatted Value	Preview Value
@msg.avp["name"][index].data	{Msg AVP:avp["Name"] [Index].Data}
@msg.avp["name"] [index].data_length	{Msg AVP:avp["Name"] [Index].Data_Length}
@msg.avp["name"][index].code	{Msg AVP:avp["Name"] [Index].Code}
@msg.avp["name"][index].flags.V	{Msg AVP:avp["Name"] [Index].Flag V}
@msg.avp["name"][index].flags.M	{Msg AVP:avp["Name"] [Index].Flag M}
@msg.avp ["name"] [index].flags.P	{Msg AVP:avp["Name"] [Index].Flag P}
@msg.avp ["name"] [index].flags.r3	{Msg AVP:avp["Name"] [Index].Flag r3}
@msg.avp ["name"] [index].flags.r4	{Msg AVP:avp["Name"] [Index].Flag r4}
@msg.avp ["name"] [index].flags.r5	{Msg AVP:avp["Name"] [Index].Flag r5}
@msg.avp ["name"] [index].flags.r6	{Msg AVP:avp["Name"] [Index].Flag r6}
@msg.avp ["name"] [index].flags.r7	{Msg AVP:avp["Name"] [Index].Flag r7}
@msg.avp ["name"] [index].vendor_id	{Msg AVP:avp["Name"] [Index].Vendor-ID}
@msg.avp["name"][index].avp ["name"][index].data	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Data}
@msg.avp["name"][index].avp ["name"][index].data_length	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Data_Length}
@msg.avp["name"][index].avp ["name"][index].code	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Code}
@msg.avp["name"][index].avp ["name"][index].flags.V	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag V}
@msg.avp["name"][index].avp ["name"][index].flags.M	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag M}
@msg.avp["name"][index].avp ["name"][index].flags.P	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag P}

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description														
	Table 2-18 From the Current Messages Being Processed <table> <tr> <th>xl-Formatted Value</th><th>Preview Value</th></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].flags.r3</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r3}</td></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].flags.r4</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r4}</td></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].flags.r5</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r5}</td></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].flags.r6</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r6}</td></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].flags.r7</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r7}</td></tr> <tr> <td>@msg.avp["name"][index].avp ["name"][index].vendor_id</td><td>{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Vendor-ID}</td></tr> </table>	xl-Formatted Value	Preview Value	@msg.avp["name"][index].avp ["name"][index].flags.r3	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r3}	@msg.avp["name"][index].avp ["name"][index].flags.r4	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r4}	@msg.avp["name"][index].avp ["name"][index].flags.r5	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r5}	@msg.avp["name"][index].avp ["name"][index].flags.r6	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r6}	@msg.avp["name"][index].avp ["name"][index].flags.r7	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r7}	@msg.avp["name"][index].avp ["name"][index].vendor_id	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Vendor-ID}
xl-Formatted Value	Preview Value														
@msg.avp["name"][index].avp ["name"][index].flags.r3	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r3}														
@msg.avp["name"][index].avp ["name"][index].flags.r4	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r4}														
@msg.avp["name"][index].avp ["name"][index].flags.r5	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r5}														
@msg.avp["name"][index].avp ["name"][index].flags.r6	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r6}														
@msg.avp["name"][index].avp ["name"][index].flags.r7	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Flag r7}														
@msg.avp["name"][index].avp ["name"][index].vendor_id	{Msg AVP:avp["ParentAVP Name"] [Index].avp["AVP Name"] [Index].Vendor-ID}														

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description																						
	Depending on the trigger point, when the processed message being retrieved by the @msg xl-formatted values, the following message data is accessible:																						
	Table 2-19 Trigger Point for Message Data																						
	<table> <tr> <th>Trigger Points</th><th>What is Returned by the Select @msg</th></tr> <tr> <td>RTP1</td><td>Header/AVP components from the previously modified (at RTP1) incoming request.</td></tr> <tr> <td>RTP4</td><td>Header/AVP components from the previously modified request before the application invocation.</td></tr> <tr> <td>RTP6</td><td>Header/AVP components from the previously modified request after the application invocation. (If an application, for ex. MAP-Diameter IWF generates request) Header/AVP component from the request generated by the application.</td></tr> <tr> <td>RTP10</td><td>Header/AVP components from the previously modified request that is ready to be forwarded by the system.</td></tr> <tr> <td>RTP11</td><td>Header/AVP components from the previously modified re-routed request that is ready to be forwarded by the system.</td></tr> <tr> <td>ATP1</td><td>Header/AVP components from the previously modified(at ATP1) incoming answer.</td></tr> <tr> <td>ATP4</td><td>Header/AVP components from the previously modified answer before the application invocation.</td></tr> <tr> <td>ATP6</td><td>Header/AVP components from the previously modified answer after the application invocation. (If the application, for ex. Map-Diameter IWF generates answer)Header/AVP component from the answer generated by the application.</td></tr> <tr> <td>ATP10</td><td>Header/AVP components from the modified answer ready to be forwarded by the system.</td></tr> <tr> <td>DCL triggers</td><td>Header/AVP components from the system request/answer being processed.</td></tr> </table>	Trigger Points	What is Returned by the Select @msg	RTP1	Header/AVP components from the previously modified (at RTP1) incoming request.	RTP4	Header/AVP components from the previously modified request before the application invocation.	RTP6	Header/AVP components from the previously modified request after the application invocation. (If an application, for ex. MAP-Diameter IWF generates request) Header/AVP component from the request generated by the application.	RTP10	Header/AVP components from the previously modified request that is ready to be forwarded by the system.	RTP11	Header/AVP components from the previously modified re-routed request that is ready to be forwarded by the system.	ATP1	Header/AVP components from the previously modified(at ATP1) incoming answer.	ATP4	Header/AVP components from the previously modified answer before the application invocation.	ATP6	Header/AVP components from the previously modified answer after the application invocation. (If the application, for ex. Map-Diameter IWF generates answer)Header/AVP component from the answer generated by the application.	ATP10	Header/AVP components from the modified answer ready to be forwarded by the system.	DCL triggers	Header/AVP components from the system request/answer being processed.
Trigger Points	What is Returned by the Select @msg																						
RTP1	Header/AVP components from the previously modified (at RTP1) incoming request.																						
RTP4	Header/AVP components from the previously modified request before the application invocation.																						
RTP6	Header/AVP components from the previously modified request after the application invocation. (If an application, for ex. MAP-Diameter IWF generates request) Header/AVP component from the request generated by the application.																						
RTP10	Header/AVP components from the previously modified request that is ready to be forwarded by the system.																						
RTP11	Header/AVP components from the previously modified re-routed request that is ready to be forwarded by the system.																						
ATP1	Header/AVP components from the previously modified(at ATP1) incoming answer.																						
ATP4	Header/AVP components from the previously modified answer before the application invocation.																						
ATP6	Header/AVP components from the previously modified answer after the application invocation. (If the application, for ex. Map-Diameter IWF generates answer)Header/AVP component from the answer generated by the application.																						
ATP10	Header/AVP components from the modified answer ready to be forwarded by the system.																						
DCL triggers	Header/AVP components from the system request/answer being processed.																						

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description
Table 2-20 From the Request During the Answer Processing	
xl-formatted Value	Preview Value
@req.avp["name"][index].data	{Req AVP:avp["Name"] [Index].Data}
@req.avp["name"] [index].data_length	{Req AVP:avp["Name"] [Index].Data_Length}
@req.avp["name"][index].code	{Req AVP:avp["Name"] [Index].Code}
@req.avp["name"][index].flags.V	{Req AVP:avp["Name"] [Index].Flag V}
@req.avp["name"][index].flags.M	{Req AVP:avp["Name"] [Index].Flag M}
@req.avp["name"][index].flags.P	{Req AVP:avp["Name"] [Index].Flag P}
@req.avp["name"] [index].flags.r3	{Req AVP:avp["Name"] [Index].Flag r3}
@req.avp["name"] [index].flags.r4	{Req AVP:avp["Name"] [Index].Flag r4}
@req.avp["name"] [index].flags.r5	{Req AVP:avp["Name"] [Index].Flag r5}
@req.avp["name"] [index].flags.r6	{Req AVP:avp["Name"] [Index].Flag r6}
@req.avp["name"] [index].flags.r7	{Req AVP:avp["Name"] [Index].Flag r7}
@req.avp["name"] [index].vendor_id	{Req AVP:avp["Name"] [Index].Vendor-ID}
@req.avp["name"][index].avp ["name"][index].data	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Data}
@req.avp["name"][index].avp ["name"][index].data_length	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Data_Length}
@req.avp["name"][index].avp ["name"][index].code	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Code}
@req.avp["name"][index].avp ["name"][index].flags.V	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag V}
@req.avp["name"][index].avp ["name"][index].flags.M	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag M}
@req.avp["name"][index].avp ["name"][index].flags.P	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag P}

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description																																															
	Table 2-20 From the Request During the Answer Processing <table><thead><tr><th>xl-formatted Value</th><th>Preview Value</th></tr></thead><tbody><tr><td>@req.avp["name"][index].avp ["name"][index].flags.r3</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r3}</td></tr><tr><td>@req.avp["name"][index].avp ["name"][index].flags.r4</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r4}</td></tr><tr><td>@req.avp["name"][index].avp ["name"][index].flags.r5</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r5}</td></tr><tr><td>@req.avp["name"][index].avp ["name"][index].flags.r6</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r6}</td></tr><tr><td>@req.avp["name"][index].avp ["name"][index].flags.r7</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r7}</td></tr><tr><td>@req.avp["name"][index].avp ["name"][index].vendor_id</td><td>{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Vendor-ID}</td></tr></tbody></table> Depending on the trigger point, when the request is retrieved by using the "@req" xl-formatted values and based on the origin of the answer message, the following request data is accessible: Table 2-21 Trigger Point for Request Data <table><thead><tr><th>Trigger Points</th><th>Answer is Received</th><th>Answer is Generated</th></tr></thead><tbody><tr><td>RTP1</td><td></td><td>N/A</td></tr><tr><td>RTP4</td><td></td><td>N/A</td></tr><tr><td>RTP6</td><td></td><td>N/A</td></tr><tr><td>RTP10</td><td></td><td>N/A</td></tr><tr><td>RTP11</td><td></td><td>N/A</td></tr><tr><td>ATP1</td><td>Header/AVP components from the modified request forwarded by the system</td><td>N/A</td></tr><tr><td>ATP4</td><td>Header/AVP components from the modified request forwarded by the system</td><td>N/A or Header/AVP components from the modified request.</td></tr><tr><td>ATP6</td><td>Header/AVP components from the modified request forwarded by the system</td><td>N/A or Header/AVP components from the modified request.</td></tr><tr><td>ATP10</td><td>Header/AVP components from the modified request forwarded by the system</td><td>Header/AVP components from the modified request.</td></tr><tr><td>DCL triggers</td><td></td><td>N/A</td></tr></tbody></table> Note: A condition/action fails at the DCL triggers when using @req xl-formatted values in the conditions and actions.	xl-formatted Value	Preview Value	@req.avp["name"][index].avp ["name"][index].flags.r3	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r3}	@req.avp["name"][index].avp ["name"][index].flags.r4	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r4}	@req.avp["name"][index].avp ["name"][index].flags.r5	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r5}	@req.avp["name"][index].avp ["name"][index].flags.r6	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r6}	@req.avp["name"][index].avp ["name"][index].flags.r7	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r7}	@req.avp["name"][index].avp ["name"][index].vendor_id	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Vendor-ID}	Trigger Points	Answer is Received	Answer is Generated	RTP1		N/A	RTP4		N/A	RTP6		N/A	RTP10		N/A	RTP11		N/A	ATP1	Header/AVP components from the modified request forwarded by the system	N/A	ATP4	Header/AVP components from the modified request forwarded by the system	N/A or Header/AVP components from the modified request.	ATP6	Header/AVP components from the modified request forwarded by the system	N/A or Header/AVP components from the modified request.	ATP10	Header/AVP components from the modified request forwarded by the system	Header/AVP components from the modified request.	DCL triggers		N/A
xl-formatted Value	Preview Value																																															
@req.avp["name"][index].avp ["name"][index].flags.r3	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r3}																																															
@req.avp["name"][index].avp ["name"][index].flags.r4	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r4}																																															
@req.avp["name"][index].avp ["name"][index].flags.r5	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r5}																																															
@req.avp["name"][index].avp ["name"][index].flags.r6	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r6}																																															
@req.avp["name"][index].avp ["name"][index].flags.r7	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Flag r7}																																															
@req.avp["name"][index].avp ["name"][index].vendor_id	{Req AVP:avp["ParentAVP Name"] [Index].avp[AVP Name] [Index].Vendor-ID}																																															
Trigger Points	Answer is Received	Answer is Generated																																														
RTP1		N/A																																														
RTP4		N/A																																														
RTP6		N/A																																														
RTP10		N/A																																														
RTP11		N/A																																														
ATP1	Header/AVP components from the modified request forwarded by the system	N/A																																														
ATP4	Header/AVP components from the modified request forwarded by the system	N/A or Header/AVP components from the modified request.																																														
ATP6	Header/AVP components from the modified request forwarded by the system	N/A or Header/AVP components from the modified request.																																														
ATP10	Header/AVP components from the modified request forwarded by the system	Header/AVP components from the modified request.																																														
DCL triggers		N/A																																														
Linking AVP	Sub-Items																																															

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description
	Linking-AVP List containing all AVP definitions from the dictionary (except for the case where the selected Parent AVP is grouped, then only those AVPs that belong to the group are available). Note: Sub-LAVPs within a grouped LAVP cannot be retrieved (such as with <code>@store.avp["name"][index].avp["name"][index]</code>), modified, or removed.
	Linking-AVP Instance number List containing the indexes of the ParentAVP (<i>First, Second, Third, Fourth, Fifth, Any</i>).
	Linking-AVP Component List containing the following components: • <i>Data</i> • <i>Data Length</i> • <i>AVP Code</i> • <i>Flag V</i> • <i>Flag M</i> • <i>Flag P</i> • <i>Flag r3</i> • <i>Flag r4</i> • <i>Flag r5</i> • <i>Flag r6</i> • <i>Flag r7</i> • <i>Vendor ID</i> Flags V, M, and P are supported; flags r3, r4, r5, r6, and r7 are reserved flags.

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description																																								
	Table 2-22 Linking-AVP xl-Formatted Value <table> <tr> <th>xl-Formatted Value</th><th>Preview Value</th></tr> <tr> <td>@store.avp["name"][index].data</td><td>{LAVP:avp["Name"][Index].Data}</td></tr> <tr> <td>@store.avp["name"] [index].data_length</td><td>{LAVP:avp["Name"] [Index].Data_Length}</td></tr> <tr> <td>@store.avp["name"][index].code</td><td>{LAVP:avp["Name"][Index].code}</td></tr> <tr> <td>@store.avp["name"] [index].flags.V</td><td>{LAVP:avp["Name"][Index].Flag V}</td></tr> <tr> <td>@store.avp["name"] [index].flags.M</td><td>{LAVP:avp["Name"][Index].Flag M}</td></tr> <tr> <td>@store.avp["name"] [index].flags.P</td><td>{LAVP:avp["Name"][Index].Flag P}</td></tr> <tr> <td>@store.avp["name"] [index].flags.r3</td><td>{LAVP:avp["Name"][Index].Flag r3}</td></tr> <tr> <td>@store.avp["name"] [index].flags.r4</td><td>{LAVP:avp["Name"][Index].Flag r4}</td></tr> <tr> <td>@store.avp["name"] [index].flags.r5</td><td>{LAVP:avp["Name"][Index].Flag r5}</td></tr> <tr> <td>@store.avp["name"] [index].flags.r6</td><td>{LAVP:avp["Name"][Index].Flag r6}</td></tr> <tr> <td>@store.avp["name"] [index].flags.r7</td><td>{LAVP:avp["Name"][Index].Flag r7}</td></tr> <tr> <td>@store.avp["name"] [index].vendor_id</td><td>{LAVP:avp["Name"] [Index].Vendor-ID}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].data</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].data_length</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data_Length}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].code</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Code}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].flags.V</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.V}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].flags.M</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.M}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].flags.P</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.P}</td></tr> <tr> <td>@store.avp["name"][index].avp ["name"][index].flags.r3</td><td>{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r3}</td></tr> </table>	xl-Formatted Value	Preview Value	@store.avp["name"][index].data	{LAVP:avp["Name"][Index].Data}	@store.avp["name"] [index].data_length	{LAVP:avp["Name"] [Index].Data_Length}	@store.avp["name"][index].code	{LAVP:avp["Name"][Index].code}	@store.avp["name"] [index].flags.V	{LAVP:avp["Name"][Index].Flag V}	@store.avp["name"] [index].flags.M	{LAVP:avp["Name"][Index].Flag M}	@store.avp["name"] [index].flags.P	{LAVP:avp["Name"][Index].Flag P}	@store.avp["name"] [index].flags.r3	{LAVP:avp["Name"][Index].Flag r3}	@store.avp["name"] [index].flags.r4	{LAVP:avp["Name"][Index].Flag r4}	@store.avp["name"] [index].flags.r5	{LAVP:avp["Name"][Index].Flag r5}	@store.avp["name"] [index].flags.r6	{LAVP:avp["Name"][Index].Flag r6}	@store.avp["name"] [index].flags.r7	{LAVP:avp["Name"][Index].Flag r7}	@store.avp["name"] [index].vendor_id	{LAVP:avp["Name"] [Index].Vendor-ID}	@store.avp["name"][index].avp ["name"][index].data	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data}	@store.avp["name"][index].avp ["name"][index].data_length	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data_Length}	@store.avp["name"][index].avp ["name"][index].code	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Code}	@store.avp["name"][index].avp ["name"][index].flags.V	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.V}	@store.avp["name"][index].avp ["name"][index].flags.M	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.M}	@store.avp["name"][index].avp ["name"][index].flags.P	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.P}	@store.avp["name"][index].avp ["name"][index].flags.r3	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r3}
xl-Formatted Value	Preview Value																																								
@store.avp["name"][index].data	{LAVP:avp["Name"][Index].Data}																																								
@store.avp["name"] [index].data_length	{LAVP:avp["Name"] [Index].Data_Length}																																								
@store.avp["name"][index].code	{LAVP:avp["Name"][Index].code}																																								
@store.avp["name"] [index].flags.V	{LAVP:avp["Name"][Index].Flag V}																																								
@store.avp["name"] [index].flags.M	{LAVP:avp["Name"][Index].Flag M}																																								
@store.avp["name"] [index].flags.P	{LAVP:avp["Name"][Index].Flag P}																																								
@store.avp["name"] [index].flags.r3	{LAVP:avp["Name"][Index].Flag r3}																																								
@store.avp["name"] [index].flags.r4	{LAVP:avp["Name"][Index].Flag r4}																																								
@store.avp["name"] [index].flags.r5	{LAVP:avp["Name"][Index].Flag r5}																																								
@store.avp["name"] [index].flags.r6	{LAVP:avp["Name"][Index].Flag r6}																																								
@store.avp["name"] [index].flags.r7	{LAVP:avp["Name"][Index].Flag r7}																																								
@store.avp["name"] [index].vendor_id	{LAVP:avp["Name"] [Index].Vendor-ID}																																								
@store.avp["name"][index].avp ["name"][index].data	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data}																																								
@store.avp["name"][index].avp ["name"][index].data_length	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Data_Length}																																								
@store.avp["name"][index].avp ["name"][index].code	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Code}																																								
@store.avp["name"][index].avp ["name"][index].flags.V	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.V}																																								
@store.avp["name"][index].avp ["name"][index].flags.M	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.M}																																								
@store.avp["name"][index].avp ["name"][index].flags.P	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.P}																																								
@store.avp["name"][index].avp ["name"][index].flags.r3	{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r3}																																								

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description		
	Table 2-22 Linking-AVP xl-Formatted Value		
	xl-Formatted Value		Preview Value
	@store.avp["name"][index].avp["name"][index].flags.r4		{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r4}
	@store.avp["name"][index].avp["name"][index].flags.r5		{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r5}
	@store.avp["name"][index].avp["name"][index].flags.r6		{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r6}
	@store.avp["name"][index].avp["name"][index].flags.r7		{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Flags.r7}
	@store.avp["name"][index].avp["name"][index].vendor-id		{LAVP:avp["ParentLAVP Name"] [Index].avp["LAVP Name"] [Index].Vendor-ID}
Functions	Sub-Items	xl-Formatted Value	Preview Value
	Length of	strlen(<i>STRING</i>)	{Length of (<i>STRING</i>)}
	Used to determine the length of a number and if additional digits should be added or removed. For example, if a 7-digit number is received, then a default area code might have to be added to the number. Length of always works on string types. If the parameter happens to be a number, then it is automatically treated as a string by these functions. Hence, strlen(123) works the same as strlen(123), and returns 3. The input of the function string might include other xl-values such as constants, Diameter Header parts, AVP, LAVP parts, or other functions.		
	Hash	hash(<i>STRING</i> , <i>RANGE</i>)	{Hash (<i>STRING</i> , <i>RANGE</i>)}
	Used for making a routing decision based on the hash generated on the session-id AVP. This AVP is present in charging messages such as ACR and CCR. For example, if session-id hashes to 1, then set dest-host to host1; if it hashes to 2, then set dest-host to host2. Because all messages in a session need to go to the same host and they all have the same session-id, the mechanism can be used to send them to the same host without maintaining state. The input of the function string might include other xl-values such as contents, Diameter Header parts, AVP or LAVP parts, or other functions.		
	Substring	substr(<i>STRING</i> , <i>P</i> <i>OSITION</i> , <i>LENGTH</i>) Position can be negative, (counted from the end).	Substring (<i>STRING</i> , <i>POSITION</i> , <i>LENGTH</i>)}

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description	
	Used to inspect a part of a string or number and make changes if needed. For example, if the first 4 characters match +011, then delete the characters. Substring works always on string types. The input of the function position specifies the position(character) at which the counting of the substring begins. Position 0 indicates the first character of the string. -1 indicates the last character of the string. The input of the function length specifies the number of characters to include in the substring. The specified substring is extracted. For example: substr(@msg.avp[APN-OI-Replacement][1],0,5)	
X hours	hour2sec(<i>HOURS</i>)	{HOURS hours}
Y minutes	min2sec(<i>MINUTES</i>)	{MINUTES minutes}
GMT	time()	{GMT time}
	Can be used to perform time of day routing. Certain AVPs carry time, which can be compared against a specified hour and minute to perform time of day routing. The inputs hours or minutes might include other xl-values.	
TBCD_decode	TBCD_decode(<i>OCTETSTRING</i>)	{TBCD_decode (<i>OCTETSTRING</i>)}
TBCD_encode	TBCD_encode(<i>STRING</i>)	TBCD_encode (<i>STRING</i>)}
	TBCD (Telephony Binary Decimal String) is used to represent the digits from 0 to 9, *,#, a,b,c. TBCD is of an octet string type, includes two digits per octet, each digit encoded between 0000 to 1001(0-9), 1010(*), 1011(#), 1100(a), 1101(b) or 1110 (c); 1111 is used as a filler in case of an odd number of digits. The bits 8765 of octet n are encoding digits 2n The bits 4321 of octet n are encoding digits 2(n-1) +1 TBCD_decode function(TBCD_decode(octetstring)): Decode digits(0-9) and specific telephony characters(*, #, a,b,c) from a TBCD-string and return an UTF8String. The input of the function octetstring includes AVP or LAVP xl-values. TBCD_encode function(TBCD_encode(string)): Encode digits(0-9) and specific telephony characters(*, #,a,b,c) to a TBCD-string and return an octet string. The input of the function string includes other xl-values (constants, Diameter header parts, AVP or LAVP parts, other functions, and so on).	
Position	Position Integer32(<i>value</i>)	{Position (<i>SUBSTRING, STRING</i>)}
	Position function(position(substring,string)) is used to search for a substring in a string and return the position of the first occurrence of a substring. Position works always on string and octet string types. The input of the function substring and string includes other xl-values (constants, Diameter header parts, AVP or LAVP parts, other functions, and so on). The corresponding position is returned. Position 0 indicates the first character of the string. Example: position(om,Thomas) returns 2.	
Get Bit	getBit(<i>AVP, position</i>)	{Get Bit (<i>AVP, Position</i>)}

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description		
	Get Bit function (getBit (AVP, Position)): The function allows retrieving a particular bit from the AVP value in the rule condition part. The input of the function includes an AVP value and the position index of the bit. The function works only with the AVPs of Unsigned32 ,Unsigned64, Integer32, Integer64 data types.		
	Convert to Integer32	Integer32 (<i><value></i>)	{Convert to Integer32 (<i><value></i>)}
	Convert to Integer64	Integer64 (<i><value></i>)	{Convert to Integer64 (<i><value></i>)}
	Convert to Unsigned32	Unsigned32 (<i><value></i>)	{Convert to Unsigned32 (<i><value></i>)}
	Convert to Unsigned64	Unsigned64 (<i><value></i>)	{Convert to Unsigned64 (<i><value></i>)}
	Convert to Float32	Float32 (<i><value></i>)	{Convert to Float32 (<i><value></i>)}
	Convert to Float64	Float64 (<i><value></i>)	{Convert to Float64 (<i><value></i>)}
	Convert to String	String (<i><value></i>)	{Convert to String (<i><value></i>)}
	Convert to OctetString	OctetString (<i><value></i>)	{Convert to OctetString (<i><value></i>)}
	Convert to Address	Address (<i><value></i>)	{Convert to Address (<i><value></i>)}
	Usually the entered xl -formatted values are cast automatically. In some cases, the Administrator might need to enforce the certain interpretation for the entered xl-data. The purpose for Explicit casting is to enable the conversion of an xl-value to the certain DIAMETER protocol format (type), the Administrator enters Type, where Type is one of the DIAMETER data types in front of the entered data. Example: String (@msg.avp["Value-Digits"][1].data) The conversion is not possible when the value is reasonably non convertible to some specific format, for example, an integer cannot be converted to an IP address. See no scenarios in Explicit Casting. The conversion might be possible if the certain conditions are met. For example, an OctetString can be converted to any other format in case, the encoded data fits into the specified enforced format. See Explicit Casting. While converting intereger64 to integer32, (the same for unsigned integer 32/64, float32/64) the check is performed to verify if the data loss occurs. In such case the conversion fails.		
	Call Perl Subroutine	perlsub (<i>script</i> , <i>subroutine</i>)	{Call Perl Subroutine(<i>script</i> , <i>subroutine</i> , <i>\$</i>)}
	The script is used in the left or right side of the conditions or in the actions and executes an existing perl subroutine in a Condition (See Table 2-7) or Action (See Execute Perl Subroutine in the Actions that modify an AVP.)		
Operators	Provide the ability to perform mathematical operations on the AVP. - Plus - Minus	+ -	+ - The connection is on which the request was received.

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description		
Connections	DSR Ingress Connection is the connection id on which the request was received.	@dsr.ingress.connection	{DSR Ingress Connection}
	DSR Egress Connection is the connection ID from which the request is sent.	@dsr.egress.connection	{DSR Egress Connection}
	DSR Ingress Connection-Name is the connection name (UTF8String) on which the request was received.	@dsr.ingress.connection.name	{DSR Ingress Connection-Name}
	DSR Egress Connection-Name is the connection name (UTF8String) from which the request is sent.	@dsr.egress.connection.name	{DSR Egress Connection-Name}
Peers	DSR Ingress Peer is the peer node id from which the request was received.	@dsr.ingress.peer	{DSR Ingress Peer}
	DSR Egress Peer is the peer node ID to which the request is sent.	@dsr.egress.peer	{DSR Egress Peer}
	DSR Ingress Peer-Name is the peer node name (UTF8String) from which the request was received.	@dsr.ingress.peer.name	{DSR Ingress Peer-Name}
	DSR Egress Peer-Name is the peer node name (UTF8String) to which the request is sent.	@dsr.egress.peer.name	{DSR Egress Peer-Name}
Internal Variables	List containing all internal variables specified in the system on the page	<i>\$name</i>	{Internal Variable: <i>name</i> }
Message	Message Priority is the corresponding priority is returned, if the specified xl-value contains the function	@msg.priority	{Message Priority}
Back Reference	Number of occurrence of the back reference: input field for one digit; default is 0.	\<number>	\<number>

Table 2-17 Formatting Value Wizard Specifiers

Specifier	Description
	Because Back Reference can be part of only a replacement string, this specifier is presented only for the Substitute in AVP Value Action.

[Table 2-23](#) contains conversion values for explicit casting.

Table 2-23 Explicit Casting

Convert to	Yes	Maybe	No
Integer32 to Integer32			x
Integer32 to Unsigned32	x		
Integer32 to Integer64	x		
Integer32 to Unsigned64	x		
Integer32 to Float32	x		
Integer32 to Float64	x		
Integer32 to String	x		
Integer32 to OctetString	x		
Integer32 to Address			x
Unsigned32 to Integer32	x		
Unsigned32 to Unsigned32			x
Unsigned32 to Integer64	x		
Unsigned32 to Unsigned64	x		
Unsigned32 to Float32	x		
Unsigned32 to Float64	x		
Unsigned32 to String	x		
Unsigned32 to OctetString	x		
Unsigned32 to Address			x
Integer64 to Integer32		x	
Integer64 to Unsigned32		x	
Integer64 to Integer64			x
Integer64 to Unsigned64	x		
Integer64 to Float32	x		
Integer64 to Float64	x		
Integer64 to String	x		
Integer64 to OctetString	x		
Integer64 to Address			x
Unsigned64 to Integer32		x	
Unsigned64 to Unsigned32		x	
Unsigned64 to Integer64	x		
Unsigned64 to Unsigned64			x
Unsigned64 to Float32	x		
Unsigned64 to Float64	x		
Unsigned64 to String	x		
Unsigned64 to OctetString	x		

Table 2-23 (Cont.) Explicit Casting

Convert to	Yes	Maybe	No
Unsigned64 to Address			x
Float32 to Integer32		x	
Float32 to Unsigned32		x	
Float32 to Integer64		x	
Float32 to Unsigned64		x	
Float32 to Float32			x
Float32 to Float64	x		
Float32 to String		x	
Float32 to OctetString	x		
Float32 to Address			x
Float64 to Integer32		x	
Float64 to Unsigned32		x	
Float64 to Integer64		x	
Float64 to Unsigned64		x	
Float64 to Float32		x	
Float64 to Float64			x
Float64 to String		x	
Float64 to OctetString	x		
Float64 to Address			x
String to Integer32		x	
String to Unsigned32		x	
String to Integer64		x	
String to Unsigned64		x	
String to Float32		x	
String to Float64		x	
String to String			x
String to OctetString	x		
String to Address		x	
OctetString to Integer32		x	
OctetString to Unsigned32		x	
OctetString to Integer64		x	
OctetString to Unsigned64		x	
OctetString to Float32		x	
OctetString to Float64		x	
OctetString to String		x	
OctetString to OctetString			x
OctetString to Address		x	
Address to Integer32			x
Address to Unsigned32			x
Address to Unsigned64			x
Address to Integer64			x
Address to Float32			x
Address to Float64			x

Table 2-23 (Cont.) Explicit Casting

Convert to	Yes	Maybe	No
Address to String	x		
Address to OctetString	x		
Address to Address			x

2.4 Enumerations

An Enumeration Type (Enum Type) consists of a name and a set of values. The purpose of the **Enum** Type is to define the possible values of a data input field.

The allowed values are either comma separated integers³² or in the form of <label>:<value>, where value is an integer³². If an item does not contain a colon, then the value and label are the same.

Pre-defined Enum Types are provided with the Diameter Mediation feature. New Enum Types can be defined with their possible values. When a new Enum Type is created, it automatically appears in the **Conditions** section of the **Diameter**, and then **Mediation**, and then **Rule Templates**, and then **[Insert] and [Edit] Insert, Copy, and Edit** pages, within the list of Right value types. The Enum Type must be created before a Rule template Condition or Action can use it. The values of the Enum Type (Integer32 type) used by the Mediation Rule Set can be modified after the Rule Template has been created.

When a Right value of a Rule Template Condition is set to an Enum Type, the actual value can be set in a rule only to one of the valid values of the specified Enum Type. This is enforced by presenting a list instead of an input field on the **Diameter**, and then **Mediation**, and then **Rule Sets [Insert] and [Edit]** pages.

On the **Diameter**, and then **Mediation**, and then **Enumerations** page, you can perform the following actions:

- **Insert** new Enumeration Type and its values, as defined in [Adding an Enumeration](#)
If the maximum number of Enumeration Types (64) already exist in the system, the **Diameter**, and then **Mediation**, and then **Enumerations [Insert]** page does not open and an error message displays.
- **Edit** the name of an Enumeration, as defined in [Editing an Enumeration](#).
- **Delete** an Enumeration Type, as defined in [Deleting an Enumeration](#).

2.4.1 Mediation Enumerations elements

[Mediation Enumeration elements](#) describe the fields on the insert, edit, and view pages. Data input notes apply only to the insert and edit pages; the view pages.

Table 2-24 Mediation Enumeration Elements

Element	Description	Data Input Notes
Name	Name used to label this Enumeration Type in the system. A unique value is required in this field.	Format: field Range: 1-64 characters

Table 2-24 (Cont.) Mediation Enumeration Elements

Element	Description	Data Input Notes
Values	Comma separated list of possible values as an integer32 (1, 2, 3, and so on) or in the form of <label>:<value> (a:1, b:2,c:3, and so on.) Note: Either in form of <value>, <value>, <value>, and so on or in the form of <label>:<value>, <label>:<value>, and so on; where value is Integer32.	Format: field Range: up to 2048 characters string Default: none

2.4.2 Adding an Enumeration

The following procedure can be used to configure a new **Enumeration** type.

A new Enumeration type can be used when defining **Rule Template Conditions** and **Linking-AVPs**.

The fields are described in [Mediation Enumerations elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Enumerations**.
2. Click **Insert**.
3. Enter **Name**.
4. Enter one or more **Values** using a comma to separate multiple values.

Note

Either in the form of <value>, <value>, <value>, and so on or <label>:<value>, <label>:<value>, and so on, where value is Integer32.

5. Click **OK**, **Apply**, or **Cancel**

2.4.3 Editing an Enumeration

Use this procedure to change the Enumeration.

An item cannot be removed from the comma-separated list of values that is already used by the configured data of a Rule Template or by the Rule Template.

The fields are described in [Mediation Enumerations elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Enumerations**.
2. Select the row containing the Enumeration to be changed.
3. Click **Edit**.
4. Change the **Name** or **Values**, or both, associated with the selected Enumeration.

Note

Either in the form of <value>, <value>, <value>, and so on or <label> :<value>, <label> :<value>, and so on; where value is Integer32.

5. Click **OK**, **Apply**, or **Cancel**

2.4.4 Deleting an Enumeration

Use the following procedure to delete an Enumeration.

An Enumeration type cannot be deleted if any Rule Templates refer to the **Enumeration** type.

1. Click **Diameter**, and then **Mediation**, and then **Enumerations**.
2. Select the **Name** of the Enumeration type to be deleted.
3. Click **Delete**.
A popup window appears to confirm the delete.
4. Click **OK** or **Cancel**.

2.5 Triggers

An execution trigger defines a Triggering Point within the message processing logic. When the triggering point is reached, the mediation operations (Rule Sets) associated with that triggering point are executed. The type of the Trigger defines whether the triggering point is part of the request or the answer processing. The Rule Set can be defined to be executed as a part of the Actions of another Rule Set, or it can be triggered at some specific point of the message processing. A Trigger is executed during an ingress and egress message.

Associations of a Trigger with new Rule Sets can be added, existing associations can be removed, and the sequence of the Rule Set Name list can be changed to modify the dra-Worker behavior based on the Rule Set execution. The behavior of an dra-Worker is the same with and without a Trigger if no Rule Set is associated with the Trigger.

Rule Sets associated with a Trigger are executed in the sequence in which they are listed under the Trigger name on the **Diameter**, and then **Mediation**, and then **Triggers** page.

A Trigger is created by setting a specific point at the Diameter Routing Layer (DRL), Diameter Connection Layer (DCL) or Diameter Application Layer (DAL).

Diameter Routing Layer (DRL) Triggers

Diameter Routing Layer (DRL) Triggers, which enable request and answer messaging using RTP1, RTP10, ATP1, ATP10 and RTP11 trigger points. The Triggers described in [Diameter Routing Layer \(DRL\) Triggers](#) are available for Diameter Mediation. [Table 2-25](#) lists the routing layer execution trigger names and related information.

Table 2-25 Diameter Mediation Routing Layer Triggers

Execution Trigger Name	Message Type	Triggering Point
Diameter request message received from connection	Request	Request Trigger Point 1; occurs upon receipt of a request (RTP1)

Table 2-25 (Cont.) Diameter Mediation Routing Layer Triggers

Execution Trigger Name	Message Type	Triggering Point
Diameter request message ready to be forwarded to connection	Request	Request Trigger Point 10; occurs just before forwarding the request upstream (RTP10)
Diameter answer message received from connection	Response	Answer Trigger Point 1; occurs upon receipt of an answer (ATP1)
Diameter answer message ready to be forwarded to connection	Response	Answer Trigger Point 10; occurs just before forwarding the answer downstream (ATP10)
Diameter request message attempted for re-route	Request	Request Trigger Point 11 (RTP11); occurs just prior to re-routing the request upstream.

Diameter Connection Layer (DCL)

Diameter Connection Layer (DCL) Triggers enable peer-to-peer messaging using CEx, DWx, DPx trigger points.

The data stored as an Internal Variable or a linking AVP at one of the DCL triggers is valid only inside of the current trigger and can not be accessed by other triggers. The Triggers described in [Diameter Mediation Connection Triggers](#) are available for Diameter Mediation.

[Table 2-26](#) lists the connection layer execution trigger names and related information.

Table 2-26 Diameter Mediation Connection Layer Triggers

Execution Trigger Name	Message Type	Triggering Point
CER message received from connection	Request	Capabilities - Exchange-Request Trigger Point 1; occurs upon receipt of a CER message (CER1)
CER message ready to be sent	Request	Capabilities - Exchange-Request Trigger Point 10; occurs just prior to sending the CER message (CER10)
CEA message received from connection	Response	Capabilities - Exchange-Answer Trigger Point 1; occurs upon receipt of a CEA message (CEA1)
CEA message ready to be sent	Response	Capabilities - Exchange-Answer Trigger Point 10; occurs just prior to sending the CEA message(CEA10)
DWR message received from connection	Request	Device-Watchdog-Request Trigger Point 1; occurs upon receipt of a DWR message (DWR1)
DWR message ready to be sent	Request	Device-Watchdog-Request Trigger Point 10; occurs just prior to sending the DWR message (DWR10)

Table 2-26 (Cont.) Diameter Mediation Connection Layer Triggers

Execution Trigger Name	Message Type	Triggering Point
DWA message received from connection	Response	Device-Watchdog-Answer Trigger Point 1; occurs upon receipt of a DWA message (DWA1)
DWA message ready to be sent	Response	Device-Watchdog-Answer Trigger Point 10; occurs just prior to sending the DWA message (DWA10)
DPR message received from connection	Request	Disconnect-Peer-Request Trigger Point 1; occurs upon receipt of a DPR message (DPR1)
DPR message ready to be sent	Request	Disconnect-Peer-Request Trigger Point 10; occurs just prior to sending the DPR message (DPR10)
DPA message received from connection	Response	Disconnect-Peer-Answer Trigger Point 1; occurs upon receipt of a DPA message (DPA1)
DPA message ready to be sent	Response	Disconnect-Peer-Answer Trigger Point 10; occurs just prior to sending the DPA message (DPA10)

Diameter Application Layer (DAL)

Diameter Application Layer (DAL) Triggers can modify the requests and answers based on the condition of the message forwarded to an application or in case the message exits an application. The application can be verified before the template or rule set's execution when the trigger points are assigned. The **Diameter**, and then **Mediation**, and then **Triggers [Insert]** page provides an association between the template (rule set) and the application, with the options to:

- Execute the template (rule set), if the involved application is name of the application.
- Execute the template (rule set) unconditionally, in other words, for all involved applications.

The Triggers described in [Diameter Mediation Application Triggers](#) are available for Diameter Mediation. [Table 2-27](#) lists the application trigger names and related information.

Table 2-27 Diameter Mediation Application Triggers

Execution Trigger Name	Message Type	Triggering Point
Diameter request message ready to be forwarded to application	Request	Request Trigger Point 4; occurs just prior to the invocation of an application (RTP4)
Diameter request message received from application	Request	Request Trigger Point 6; occurs immediately after exiting an application (RTP6)
Diameter answer message ready to be forwarded to application	Response	Answer Trigger Point 4; occurs just prior to the invocation of an application (ATP4)
Diameter answer message received from application	Response	Answer Trigger Point 6; occurs immediately after exiting an application (ATP6)

Templates are not executed if:

- A template (rule set) is assigned to a specific application and the application is disabled, the associated templates (rule sets) are not executed.
- An application generates an Answer, templates at RTP6 and ATP4 are not executed.
- An application generates a Request, templates at RTP4 and ATP6 are not executed.
- An application does not stay in the loop for answer processing, templates at ATP4 or ATP6 are not executed.

Rule Sets associated with a Trigger are executed in the sequence in which they are listed under the Trigger name and you can perform the following actions:

- **Insert** a Trigger for the associated Rule Set, as defined in [Associating a Rule Set with a Trigger](#).
- **Remove** a selected Rule Set Name in the list under a Trigger name as defined in [Removing the Association of a Rule Set with a Trigger](#)
- Click **Up** and **Down** buttons to alter the sequence of execution of the Rule Sets associated with a Trigger.

2.5.1 Mediation Triggers elements

[Table 2-28](#) and [Table 2-29](#) describes the fields on the **Diameter**, and then **Mediation**, and then **Triggers** and **Diameter**, and then **Mediation**, and then **Triggers [Insert]** pages. The Triggers page is read-only.

Table 2-28 Mediation Triggers Elements

Element	Description
Rule Set Name	The name of each Rule Set that is associated with a Trigger and executed by the triggering point.
Live	A yes sign (check mark) indicates that the Rule Set has been set to the "Active" state (enabled for the live traffic).
DSR Application	The DSR application is present when an application Trigger is selected.
Request	The Request list is present when RTP1, RTP4, RTP6, RTP10 or RTP11 is selected on the Triggers Insert page. If a non-request trigger is selected, then the Request list is not present.
Scope	The Scope allows activating a template (Rule Set) on either all MPs or a specific MP(s) under the SO.

Insert Triggers elements

Table 2-29 Adding Triggers Elements

Element	Description	Data Input Notes
Rule Set Name	Name of the Rule Set that is to be executed by the triggering point.	Format: list Range: The Rule Sets (supported by the Trigger and in the Active or Test state) are listed in the Rule Set Name list. Default: First Rule Set that is supported by the Trigger and is in the Active or Test state.

Table 2-29 (Cont.) Adding Triggers Elements

Element	Description	Data Input Notes
Request	The Request list is present when RTP1, RTP10, RTP11, RTP4, or RTP6 is selected on the Triggers Insert page. If a non-request trigger is selected, then the Request list is not present. See Diameter Mediation Routing Layer Triggers and Diameter Mediation Application Triggers .	Format: list Range: Normal Request, Redirected Request, All Default: Normal Request
DSR Application	The DSR Application is present when an application Trigger is selected. See Diameter Mediation Application Triggers	Format: list Range: All enabled DSR Applications Default: All
Scope	The Scope allows activating a template (Rule Set) on either all MPs or a specific MP(s) under the SO.	Enable on all MPs Format: Check box Range: Checked or not checked Available MPs Format: Drag and drop with >> to move the available MPs to Enable Rule Set on MP(s) . Range: Available MPs Enable Rule Set on MP(s) Format: Drag and drop with << to move the available MPs to Available MPs . Range: Enable Rule Set on MP(s)

2.5.2 Associating a Rule Set with a Trigger

Use this procedure to associate a Rule Set with a Trigger.

Only Rule Sets with Rule Templates in Test or Active state can be associated with a Trigger.

The fields are described in [Mediation Triggers elements](#)

1. Click **Diameter**, and then **Mediation**, and then **Triggers**.
2. The newly assigned Rule Set appears at the bottom of the list of Rule Sets for the Trigger. If the Rule Set sequence needs to be changed, use the **Up** and **Down** buttons to move the Rule Sets to different positions in the list.

Click a Rule Set and the **Up** and **Down** buttons to move the selected Rule Set up or down one position on the list.

The **Live** column shows a check mark if the Rule Template for the newly associated Rule Set is in the Active state for use with live traffic (see [State and Properties](#)).

3. Click **Insert** under the Trigger with which the new Rule Set is to be associated.

The **Diameter**, and then **Mediation**, and then **Triggers [Insert]** page does not open and an error message appears if any of the following conditions exist:

- There are no Rule Sets that support the Trigger and that are in the Active or Test state
- Associating another Rule Set to the Trigger would cause the total allowed number of associated Test Rule Sets (10) or Active Rule Sets (64) to be exceeded

4. Select the **Rule Set Name** from the list.
The default is the first Rule Set in the list.
5. Select the **Request** type from the list.
6. Select the **Scope** of the Rule Set.
7. Click **OK** or **Apply**

2.5.3 Removing the Association of a Rule Set with a Trigger

Use the following procedure to remove the association of a **Rule Set** with a **Trigger** and delete the **Rule Set Name** from the list for the Trigger.

1. Click **Diameter**, and then **Mediation**, and then **Triggers**.
2. Select the **Rule Set Name** in the list under the Trigger name.
3. Click **Remove** below the **Rule Set Name** list for the Trigger.
A popup window appears to confirm the removal.
4. Click **OK** or **Cancel**

2.6 State and Properties

The **Diameter**, and then **Mediation**, and then **State & Properties** page lists all of the Rule Templates that are configured in the system, and shows the **State**, **Action Error Handling** and **Status of Rule Counters** settings for each Rule Template.

Each Rule Template is in one of the following **States** at any point of time:

- Development
- Test
- Active

The **Action Error Handling** defines the error handling strategy to be used if any Action in the Rule Template fails.

The **Status of Rule Counters**, when enabled, analyze the rules and track the number of times each rule successfully matches all the conditions in the template. The Rule Template must be in a Test or Active state.

Each Rule Template starts in the development state when it is being created. Rule Templates in the Development state cannot be assigned to Triggers, until it changes to a test state.

After all of the necessary **Conditions** and **Actions** are added, the Rule Template must be set to the test state, to indicate that the Rule Template is complete. A Rule Set entry is generated in the Rule Sets left-hand menu folder; the Rule Set can be provisioned with actual data in one or more rules, and can be associated with a Trigger. In the Test state, only limited changes can be made to the contents of the Rule Template, as define in [Rules Template](#).)

The Rule Template state can be set back to development only when the Administrator privileges are activated for the Diameter Mediation feature. All provisioned data for the Rule Template is lost if the state is set back to Development.

The Rule Template state can be set to test or the association between the Rule Set and a Trigger can be removed to disable the Rule Set for live traffic.

In the Test state a Mediation Rule Set does not affect the live traffic, but the operator can test the newly created, imported, or modified Rule Set that was generated from the Rule Template. The Diagnostics Tool can be used to exercise and test the Rule Templates in the Test state, along with Rule Templates in the Active state. See Maintenance in the Diameter User's Guide and Reports in the *Diameter User's Guide*.

When the state of a Rule Template is set to Active, the Rule Set associated with the Rule Template begins to participate in processing of real traffic messages.

The Import function from the **Diameter**, and then **Mediation**, and then **Rule Templates** page is duplicated on the **Diameter**, and then **Mediation**, and then **State & Properties** page for use when the Administrator privileges are not activated and the **Diameter**, and then **Mediation**, and then **Rule Templates** page cannot be accessed. An imported Rule Template is set to Test state.

On the **Diameter**, and then **Mediation**, and then **State & Properties** page, you can perform the following actions:

- Import Rule Template to import a previously exported Rule Template from a location outside of the system, as defined in [Importing a Rule Template](#).
If importing a Rule Template would cause the maximum number of Rule Templates (100) in the system to be exceeded, the Rule Template is not imported and an error message appears.
- **Edit** a Rule Template Name in the list, and click . You can change the State, Action Error Handling and enable or disable Status of Rule Counters for the selected Rule Template, as defined in [Editing State and Properties](#).
When the Administrator privileges are not activated for the Diameter Mediation feature, the state of a Rule Template cannot be changed back to Development.
- **Delete** a **Rule Template Name**, as defined in [Deleting a Rule Template](#).
When a Rule Template is deleted from the **Diameter**, and then **Mediation**, and then **State & Properties** page, it is deleted from all other pages at the same time.

2.6.1 Mediation State and Properties elements

[Table 2-30](#) describes the fields on the view and edit pages. Data Input Notes apply only to the edit page.

Table 2-30 Mediation State & Properties Elements

Element	Description	Data Input Notes
Rule Template Name	The name of a configured Rule Template.	Format: field
State	The state of the Rule Template. Development - the Rule Template is disabled for any live or test traffic; it is under development. Test - the Rule Sets entry is generated and the Rule Set is enabled only for the special test messages. Active - the Rule Template and Rule Set are enabled for any kind of traffic.	Format: list Range: Development (only for creating and modifying Rule Templates), Test, Active Default: Development

Table 2-30 (Cont.) Mediation State & Properties Elements

Element	Description	Data Input Notes
Action Error Handling	Specifies the type of error handling to be used if an Action in a Rule Template fails.	Format: list Range: Ignore the error, Immediately exit from the rule template, Immediately exit from the trigger point Default: Ignore the error
Status of Rule Counters	Enables or disables the rule counters for a specific rule template. In the view only screen, the status of rule counters column is set to active or stopped respectively to the checked or unchecked box. Active status means the Rule Set counter started counting the number of matches per rule, which allows you to view the rule counters from the enabled Show Counters button on the Rule Set page. Stopped status means the counter is disabled.	Format: checkbox Range: N/A

2.6.2 Importing a Rule Template

A Rule Template can be imported into the system using the **Import Rule Template** action on the **Diameter**, and then **Mediation**, and then **State & Properties** page.

Existing Rule Templates can be imported. Existing Rule Templates are previously-generated Rule Templates that have been exported from Diameter Mediation using the **Export** action on the **Diameter**, and then **Mediation**, and then **Rule Templates** page.

Use the following procedure to import a Rule Template located outside of the file system:

1. Click **Diameter**, and then **Mediation**, and then **State & Properties**.
2. Click **Import Rule Template**.
3. Click **Browse** to open the Choose File popup window.
4. Navigate to the location of the Rule Template file that you want to import.
5. With the Rule Template filename displayed in the **File name** field, click **Open**.

The filename appears in the **Choose a file to import** field.

6. Click **Import File**.

The selected Rule Template file is imported and appears in the **Rule Template Name** list on the **Diameter**, and then **Mediation**, and then **State & Properties** page.

2.6.3 Editing State and Properties

Meta-Administrator and Rule Set Administrator use this procedure to change the State and Properties associated with a Rule Template.

The state of a Rule Template can be changed to or from the development state only when an Administrator privileges are activated for the Diameter Mediation feature.

A Rule Template state cannot be changed from Test to Development for a Rule Template that is referenced by another instance such as another Rule Template or the Execution Trigger.

When a Rule Template state is changed back to development, any associated Rule Sets are deleted from the **Rule Sets** folder.

[Table 2-30](#) provides definitions of each field.

1. Click **Diameter**, and then **Mediation**, and then **State & Properties**.
2. Select the Rule Template to change.
3. Click **Edit**.
4. Change the **State** and **Action Error Handling** according to the [Table 2-30](#) guidelines.
5. Check or uncheck the **Status of Rule Counters**.

Checking this box, with the Rule Template Test or Active, sets the **Status of Rule Counters** column to **Active** on the **State & Properties** page.

6. Click **OK**, **Apply**, or **Cancel**

2.6.4 Deleting a Rule Template

Use the following procedure to delete a Rule Template from the **Diameter**, and then **Mediation**, and then **State & Properties** list.

When a Rule Template is deleted from the **Diameter**, and then **Mediation**, and then **State & Properties** page, it is deleted from all other pages at the same time.

1. Click **Diameter**, and then **Mediation**, and then **State & Properties**.
2. Select the **Rule Template Name** to be deleted.
3. Click **Delete**.

A popup window appears to confirm the delete.

4. Click **OK** or **Cancel**

2.7 Internal Variables

The **Diameter**, and then **Mediation**, and then **Internal Variables** page lists all of the Internal Variables that are configured in the system, and shows the Variable Name, Type, Default Value, and Description available for each Rule Template. Internal variables represent buffers distinguished by the names and can store data of the specified format. The data stored in the buffers can be used both in the conditions and actions (via the Formatting Value Wizard page) and it is accessible during the whole transaction.

Internal Variables are created by:

- Setting the unique name.
- Selecting the type for enabling syntax checks of the stored values.
- Setting default value (optional). The default value is used in case the user does not provision/assign a new value to the variable.

Note

It is also possible to leave the internal variable empty.

- Writing a short description (optional) to explain the purpose of the variable and how it is used in the templates/rules.

After the new variable is created it becomes accessible on the [Formatting Value Wizard](#). The user can provision/assign the value of the internal variable by means of newly introduced Set Internal Variable action. The internal variable is SET when:

- The default value is provisioned and not overwritten by the action Set Internal variable.
- The action Set Internal Variable assigns a value to the internal variable.
- Any of the AVPs has matched the criteria and the option Store matched instance into has been chosen. The index number of the matched AVP is stored under the specified internal variable.

The value of the variable can be a static value, the value coming from the message, some previously stored value, and so on. The next template can contain the condition which uses the previously provisioned internal variable. All newly created variables of integer types (both signed and unsigned) are accessible as AVP instance numbers on the Formatting Value Wizard page. If the specified xl-value contains an internal variable (\$name), the value of the variable set previously is used.

2.7.1 Mediation Internal Variables elements

[Mediation Internal Variables elements](#) describe the fields on the view, edit, and view pages. Data input notes apply only to the insert and edit pages.

Table 2-31 Mediation Internal Variables Elements

Element	Description	Data Input Notes
* Variable Name	Name used to label this Internal Variable in the system. A unique value is required in this field.	Format: field Range: 32 character string Default: none
* Type	The data format of the variable. A value is required in this field.	Format: list Range: • Integer32 • Integer64 • Unsigned32 • Unsigned64 • Float32 • Float64 • Address • Time • UTF8String • DiameterIdentity • DiameterURI • OctetString Default: Integer32
Default Value	An optional value to initialize the variable in this field.	Format: field Range: maximum of 255 character string Default: none

Table 2-31 (Cont.) Mediation Internal Variables Elements

Element	Description	Data Input Notes
Description	Short description of the variable. A value is not required in this field.	Range: maximum 255 character string. Default: none

2.7.2 Adding an Internal Variable

The following procedure can be used to configure a new **Internal Variables**.

A new **Internal Variables** can be used when defining **Rule Template Conditions**.

The fields are described in [Mediation Internal Variables elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Internal Variables**.
2. Click **Insert**.
3. Enter the **Variable Name**.
4. Enter a **Type**.
5. Enter a **Default Value**.

When the **Type** selected is an *OctetString* and the default value does not start with 0x, it is automatically added. When the **Type** is an *OctetString* the entered default value is automatically down-cased.

6. Enter a short **Description** associate with this **Variable Name**.
7. Click **OK**, **Apply**, or **Cancel**.

2.7.3 Editing an Internal Variable

Use this procedure to change the Internal Variable **Default Value** and **Description** associated with an **Internal Variables**.

The fields are described in [Mediation Internal Variables elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Internal Variables**.
2. Select the row containing the Internal Variable to be changed.
3. Click **Edit**.
4. Change the **Default Value** and/or **Description** associated with the selected Internal Variable.
5. Click **OK**, **Apply**, or **Cancel**.

2.7.4 Deleting an Internal Variable

Use the following procedure to delete an Internal Variable.

An Internal Variable Type cannot be deleted if any Rule Templates refer to the Internal Variable.

1. Click **Diameter**, and then **Mediation**, and then **Internal Variable**.

2. Select the **Name** of the Internal Variable to be deleted.
3. Click **Delete**.

A popup window appears to confirm the delete.

4. Click **OK** or **Cancel**

When **OK** is clicked and any configured Rule Templates refer to the Internal Variable that is being deleted, the Internal Variable is not deleted and an error message appears.

2.8 Measurements

The available Mediation Measurements are:

- Predefined Measurements (pegged automatically)
- Custom Measurements (that can be pegged by the special action)

The **Diameter**, and then **Mediation**, and then **Measurements** page lists up to 2000 custom measurements created by the user. Once you define the custom measurement, it can be used to set a peg counter action in the Rule Template.

Measurements are set up by:

- Setting the unique name of the measurement.
- Writing a short description (optional) explaining the purpose of the measurement.

For information about how to generate a measurement report, see *Generating a measurements report* in the *Alarms, KPIs and Measurements Reference*. The custom measurements are listed as sub-measurements as CAPM_MediationCustomerMeasurements within the CAPM group in the reports.

2.8.1 Mediation Measurements elements

[Table 2-32](#) describe the fields on the insert, edit, and view pages. Data input notes apply only to the insert and edit pages.

Table 2-32 Mediation Measurements Elements

Element	Description	Data Input Notes
* Measurement Name	Name used to label this Measurement in the system.	Format: field
	A unique value is required in this field.	Range: 32 character string Default: none
Description	Short description of the measurement.	Format: field
	A value is not required in this field.	Range: maximum of 255 character string Default: none

2.8.2 Adding a Measurement

Use this procedure to configure a new **Measurement**.

The fields are described in [Mediation Measurements elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Measurements**.

Note

The view screen displays all Measurements provisioned in the system. The column entries shall be **Measurement Name** and **Description**.

The Measurement Name is in the list of Peg Counter action.

2. Click **Insert**.

Note

An error message appears if the maximum number of Mediation Measurements (2000) has already been configured in the system.

3. Enter a unique name for the **Measurement Name** being added.
4. Enter a short **Description** associated with this Measurement Name.
5. Click **OK**, **Apply**, or **Cancel**.

2.8.3 Editing a Measurement

Use this procedure to change the Measurement Description associated with a **Measurement**.

The fields are described in [Mediation Measurements elements](#).

1. Click **Diameter**, and then **Mediation**, and then **Measurements**.
2. Select the **Measurement Name**.
3. Click **Edit**.
4. Change the **Description** associated with the selected Measurement.
5. Click **OK**, **Apply**, or **Cancel**.

2.8.4 Deleting a Measurement

Use the following procedure to delete a **Measurement**.

1. Click **Diameter**, and then **Mediation**, and then **Measurements**.
2. Select the row containing the **Measurement Name** to be deleted.
3. Click **Delete**.

A popup window appears to confirm the delete.

4. Click **OK** or **Cancel**.

When **OK** is clicked and any configured Rule Templates or rules refer to the Measurement that is being deleted, the Measurement is not deleted and an error message appears.

2.9 Rule Sets

Rule Sets are rules corresponding to a Rule Template that is in a test or active status.

The **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page provides a means to provision rule(s) to the Rule Set ({name} is the name of the Rule Set).

Condition set expression and actions are displayed above the Rule Set in a form of an **IF** (Condition set expression) and **THEN** (Actions).

The **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page displays the following columns:

- **Rule Id** is a unique number from the database table that identifies each rule in the Rule Set.
 - **Move the rule** is a means to move the order of the rules by using the **Up** and **Down** buttons.
Up and **Down** buttons appear in the **Move the rule** columns for a rule or rule group when the order of the rules is allowed to be changed, with the following restrictions:
 - When the **Filter** function or clicking a Condition column heading is used to sort the columns, the **Move the rule** columns are not displayed. Click **Restore Order** to return the list to its original order.
 - If all of the conditions in the rule support **Fast Search**, then the **Move the rule** columns are not displayed. See [Fast Search](#).
 - If there is at least one condition that does not support **Fast Search**, then the **Up** and **Down** buttons are displayed according to the following rules:
 - * All of the rules that support **Fast Search** always appear in the list before any rules that do not support **Fast Search**.
 - * The rows that have exactly the same data in the conditions that support **Fast Search** form a group. Rows can be moved only within their group; the **Up** and **Down** buttons are enabled and disabled accordingly.
- [Table 2-33](#) shows an example of default ordering of rules.

Table 2-33 Example of Default Ordering of Rules in a Rule Set

Fast-Search Condition A	Fast-Search Condition B	Non-Fast-Search Condition C
abc	1	-
abc	12	-
abc	-	-
abcd	1	-
abcd	1	a1
abcd	-	b1
-	1	a1
-	1	b1
-	-	-

Each row across the columns is inserted (created) in the list when a rule is provisioned. The rules on a **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** are looked up in the database in the order in which they are listed on the page. By default, the rules are sorted in the list by condition as defined in [Fast Search](#).

When a Rule Set entry is selected in the **Rule Sets** folder, the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page opens for the selected Rule Set.

On each **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page, a user can perform the following actions:

- If the **Move a rule** columns are displayed and contain **Up** and **Down** buttons, move rules up and down in the list to change the order of execution of the rules in the Rule Set.

- **Insert** to add a new rule.

Note

The maximum number of provisioned rules per template/rule set depends on the template type. For slow-search templates, which have at least one slow-search condition, the maximum number of provisioned rules is 250. For fast-search templates, the maximum number of provisioned rules is 2000. To determine if the template is a fast-search reference [Fast Search versus slow search](#).

The maximum total number of provisioned rules in the system cannot exceed 148,000, provided all the templates are of fast search type ((10 test templates + 64 active templates * 2000)).

The rule templates with no conditions is unconditionally matched for all processed messages. The Rule Sets generated from these Rule Templates allow only one rule to be provisioned.

- **Import** to import rules from an either XML-file (Rule Template) that is in a Test or Active state.
- **Delete All Rules** to delete all of the rules that have been provisioned for this Rule Set.
- **Edit** a rule.
- **Delete** a rule from the Rule Set list.
- **Export** to export rules to create an XML-file containing the Mediation version number, Template Definition, the provisioned values of the conditions and actions for each rule.
- **Show Counters** to open the Rule Counters column of the Rule Set. A Total Pegs column displays the number of times the specific rule is matched.
If Show Counters is grayed out and the Rule Counters are not available. To enable the rules counter see [Editing State and Properties](#).

When the Rule Sets View page is expanded with the Rule Counters, the following buttons are available:

- **Hide Counters** to hide the Rule Counters column.
- **Expand Counters** expands more sub-columns under the Rule Counters for all MPs under the SO. Each sub-column is MP name and displays the counters for each.
- **Collapse Counters** to hide the MP counters.
- The **Pause Updates** box is unchecked by default. This means the counters refreshes every 30 second to provide the user a printout on the GUI page. Checking the **Pause Updates** box pauses the counters. To disable the counter, see [Editing State and Properties](#).

2.9.1 Rule Sets elements - View page

[Table 2-34](#) describes the fields on each view page of a rule set.

Table 2-34 Rule Sets Elements - View Page

Element	Description	Data Notes
IF	The order of conditions for the rule set separated by an AND, OR, or complex expression.	Auto generated
THEN	The actions of the rule set separated by commas	Auto generated
Rule ID	A unique number from the database table that identifies each rule in the Rule Template. Each rule set is a condition set expression and actions summarized as IF (condition set expression) and THEN (actions) above the rule set.	Auto generated
A, B, C, D, E, and so on Condition [Name]	Each condition has a separate column with the letter (A, B, C, D, E, and so on) in alphabetical order as defined by the condition set in the Rule Templates. The name of the condition is below the letter.	Format, Range, and Default Value configured in the Rule Template.
All Actions defined on the Rule Template page for this Rule Set	Each action has a separate column in the order as defined by the action on the rule template and one or more sub-columns that show the attributes defined for the Action and the current values of the attributes. If the Parent AVP or AVP is indexed, then the index is displayed in the square brackets after the AVP attribute name. If an AVP is looked up in the message by its value, AVP contains the value prefixed with = (if it is a constant) or an xl-value prefixed with = (if it is an xl-value). A value can be prefixed with an appropriate indicator of its type or function (such as =, beginning, end, prefix, or suffix).	Format, Range, and Default Value vary depending on the Rule Template that was configured for the Rule Set.
Each rule in the Rule Set is a row in the list. The Values assigned to the Conditions and the Values assigned to each attribute of the Actions for a rule are shown in the row for that rule.		

2.9.2 Rule Sets elements - Insert and Edit Pages

[Table 2-35](#) indicates the maximum number of rule sets and rules that are allowed.

Table 2-35 Maximum Allowed Rule Sets and Rules

Description	Value
Maximum number of provisioned rules in the system (provided all the templates are of fast search type)	148,000
Maximum number of provisioned rules per Rule Set (depended on slow- or fast-search template type)	250/2000

[Table 2-36](#) describe the fields on the insert, edit, and view pages. Data input notes apply only to the insert and edit pages.

Table 2-36 Rule Sets Elements - Insert and Edit Pages

Element	Description	Data Input Notes
Field: These elements are display at the top of the page: IF, condition and THEN. The condition section contains the name for the condition followed by the operator followed by Rule Template Condition Operators .		
IF	Alphabetical Condition names ANDed, ORed or both appears between each of the Conditions.	Auto generated
Condition	A letter that organizes the template layout for each condition	Format: field A, B, C, D, E, and so on. and the condition set of either ANDed, ORed or Complex Expression.
THEN	The name of each Action that was defined for the Rule Template.	Format: Name An Action cannot be deleted and a new Action cannot be defined for the rule.
Value: This element shows the fields for the condition and action data values that can be entered or changed. For the [Insert] page, the fields are either empty or show default values. For the [Edit] page, the fields show the currently defined or default values.		
Condition expression Right value	For each defined Condition in the rule, the data value for the Right value type in the Condition. If the Optional check box was checked in the Rule Template for this Rule Set, the Right value can be empty (not provisioned). A red asterisk appears after each data value that is required (not optional) in the rule. If the Fixed check box was checked in the Rule Template for this Rule Set, the Right value cannot be changed in the rules. If the selected Right value type was an Enumerated Type , then the Value column contains a list with the corresponding Enumerated Type values, unless the selected Operator was exists, does not exist, is true, or is false .	Format: text box Range: Varies depending on the Right value type Default: Varies depending on the Right value type See Rule Template elements . The Formatting Value Wizard is available to provision Condition Right values that are xl-formatted values; click the [wizard] that appears after the data value field.
Action fields	The fields to use to define the data values for Action attributes.	Format: Varies for each type of attribute Range: Varies for each type of attribute See Rule Template elements . The Formatting Value Wizard is available to provision Action attributes that are xl-formatted values; click the [wizard] that appears after the data value field.

Table 2-36 (Cont.) Rule Sets Elements - Insert and Edit Pages

Element	Description	Data Input Notes
Description	The description that was defined in the Rule Template for a Condition or an Action on the Rule Sets page. The description can provide information such as the format to be used (such as text string or telephone number format) and the range of values (such as 1 to 255 characters).	Format: descriptive text Range: 1 to 255 characters string

2.9.3 Adding a Rule to a Rule Set

Use this procedure to define a new rule in a Rule Set.

Note

The maximum number of provisioned rules per template/rule set depends on the template type. For slow-search templates, which have at least one slow-search condition, the maximum number of provisioned rules is 250. For fast-search templates, the maximum number of provisioned rules is 2000. To determine if the template is a fast-search reference [Fast Search versus slow search](#).

There are two sections of a **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name} [Insert]** page: **IF** (zero, one, or more Conditions and the condition set of the ANDed, ORed and/or complex expression) and **THEN** (an Action). For a list of the Rule Template elements that appear in a rule and their definitions, see [Rule Template elements](#).

When a Rule Template is in the Active or Test state, this Rule Template appears as a Rule Set in the **Diameter**, and then **Mediation**, and then **Rule Sets** menu folder. The order in which rules appear on the page determines the order in which the conditions are processed.

1. Click **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}**.
2. Click **Insert**.
3. Enter the value for each condition that appears under **IF** in the **Field** section for the new rule.
4. Enter the value for each attribute of the Action that appears under **THEN** in the **Field** section for the new rule.
5. When the rule definition is complete, click:
 - **OK** to save the new rule and return to the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page. The rule name appears in the list on the page.
 - **Apply** to save the new rule and remain on the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name} [Insert]** page for additional changes.
 - **Cancel** to return to the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page without saving the changes.
6. When the Rule Set definition and testing are complete, go to the **Diameter**, and then **Mediation**, and then **State & Properties** page.

- a. Change the Rule Template **State** from Test to Active.
- b. Set the **Action Error Handling** property, if needed.

The state can be changed to Active after the testing is successful, the Rule Set data is provisioned, the Rule Set is associated with a Trigger, and the Rule Set is ready to use in live traffic.

2.9.4 Deleting All Rules from a Rule Set

Use this procedure to delete all rules from a Rule Set.

1. Click **Diameter** , and then **Mediation**, and then **Rule Sets**, and then **{name}**.
2. Click **Delete All Rules**.
A popup window appears to confirm the delete.
3. Click **OK** or **Cancel**

2.9.5 Changing a Rule in a Rule Set

Use this procedure to change values for a rule in a Rule Set (for a list of Rule Sets elements and their definitions, see [Rule Sets elements - Insert and Edit Pages](#)):

1. Click **Diameter**, and then **Mediation**, and then **Rule Sets**.
2. Select the **Rule Set** containing the rule to be edited.
3. On the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** page, select the rule that needs to be changed.
4. Click **Edit**.
5. Change values for Conditions under **IF** and Actions under **THEN** as needed.
6. Click **OK**, **Apply** or **Cancel**.

2.9.6 Deleting One Rule from a Rule Set

Use this procedure to delete one rule from a Rule Set.

1. Click **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}**.
2. Select the row for the rule to be deleted.
3. Click **Delete**.
A popup window appears to confirm the delete.
4. Click **OK** or **Cancel**

2.9.7 Export/Import Rules

The Rule Template can be provisioned with up to 250 rules and data related to the template. To Export or Import rules from provisioned Rule Templates use the **Export** function on the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** **[Export]** or **Import** function on the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** **[Import]** page.

The **Export/Import** function is used to solve the following two scenarios:

1. The rules of template X are imported back to template X. Each template is uniquely identified by a UUID number. With the exported rules, the XML-file contains the UUID number as part of a template definition. If the UUID number is the same, the rules are imported back to the original template and no mapping is necessary for the conditions or actions. The rules are imported even if the user changes the name of the template, while in a **Test** or **Active** state. A new UUID number is assigned to the template when the user moves the original template to the development state and changes in conditions/actions.
2. The rules of template X are imported to template Y. In this case the UUID number of the original template does not match with the UUID number of the target template. The process is performed to import by mapping the conditions and actions as follows:
 - a. [Conditions correlated by the condition name](#) table means that if the condition name of the target template is the same as in the original template, the values from the xml-file are imported for this condition.

Table 2-37 Conditions Correlated by the Condition Name

Template X	Template Y
A: User-name	A: User-name
B: Host IP	B: Connection
C: Visited-PLMN-Id	C: Visited-PLMN-Id
D: Proxy-Host	D: Host IP

The rules mapping process for the conditions are exported from Template X are imported to Template Y as follows:

- A => A
- B => D
- C => C

The values of condition D of Template X are not used.

Condition B of template Y uses the default value defined in the template since no match was found for it. If no default value was provisioned for the mandatory condition in the template, the error message is expected.

- b. [Table 2-38](#) table means that if the values of the action from the xml-file are imported to the action of the same type and matched columns in the target template.

Table 2-38 Actions are Correlated by their Type and Order

Template X	Template Y
A: Add AVP	A: Add AVP
B: Change AVP code	B: Add AVP
C: Add AVP	C: Add AVP
D: Add AVP	D: Add AVP
E: Set Route List	E: Set AVP value
F: Set AVP value	F: Set AVP value
G: Set AVP value	G: Set Route List
H: Set AVP value	N/A

The actions are correlated as follows:

- A => A

- C => B (second Add AVP to second Add AVP)
- D => C (third Add AVP to third Add AVP)
- default values => D (there is not fourth Add AVP action in the XML file)
- F => E (first Set AVP value to first Set AVP value)
- G => F (second Set AVP value to second Set AVP value)
- E => G (first Set Route List to first Set Route List)

The failures are logged during the import process.

The Import of Rules is very similar to the Insert of Rules. The difference is the data source (xml file instead of values entered on the web page). During the Import, the same checks are performing as seen [Adding a Rule to a Rule Set](#).

2.9.7.1 Exporting Rules

Use this procedure to export Rules XML-file from within the system to an external location.

The saved .xml contains the following information:

- The Mediation version number used for compatibility check.
- Rule Template definition
- Provisioned right values of the conditions for each rule (importable back from the file)
- Provisioned values of the actions for each rule (importable back from the file)

Note

Export is not available (grayed out) for the Rule Set in the Development state (see [State and Properties](#)).

1. Click the **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** to export.
 2. Click **Export**.
- A File Download popup window appears.
3. Click **Browse** to pop up the Choose File window.
 4. Navigate to the location to which you want to export the Rule Template.
 5. Click **Export File**.

The selected file is saved to the specified location.

2.9.7.2 Importing Rules

Use the following procedure to import existing Rules that is located outside of the file system:

1. Click **Diameter**, and then **Mediation**, and then **Rule Sets**, and then **{name}** folder.
2. Select the **Rule Set** containing the rules to be imported.
3. Click **Import**.

At the top of the page, you find an input field with this text, **Choose a file to import**, a **Browse** button, text **(No file selected)** an **Abort on First Error**: checkbox, and an **Import File** button.

4. Check the **Abort on First Error** allows the user to specify if the selected import operation should abort on the first error (when checked-in) or continue.
5. Click **Browse** to open the Choose File popup window.
6. Navigate to the location of the Rule Template file you want to import, and select the file.
7. With the Rule Template filename displayed in the **File name** field, click **Open**.
The filename appears in the **Choose a file to import** field.
8. Click **Import File**.
The process is performed to import by mapping the conditions and actions.
 - a. **Conditions** are correlated by the condition name (the values for the conditions are imported in the conditions with the same name), as defined in [Conditions correlated by the condition name](#).
 - b. **Actions** are correlated by their type (values of the actions are imported into the same actions) and order, as defined in [Actions are correlated by their type and order](#).

2.9.7.3 Exporting all Rules

Perform the following procedure to export all Rules from within the system to the File Management Directory.

The saved .xml file consists one XML file per each Rule:

- The Mediation version number used for compatibility check.
- Rule Template definition.
- Provisioned right values of the conditions for each rule (importable back from the file).
- Provisioned values of the actions for each rule (importable back from the file).

Note

Export All is not available for the Rule Sets in the development state , see [State and Properties](#).

1. From the main menu, click **Diameter**, **Mediation**, then **Rule Sets** and then **Rule Sets Export**.
2. Click **Export All**. A popup window appears to confirm, click **OK**.
3. A task will be created to process the **Export All** in the background. The progress of the task may be monitored on the **Status & Manage**, and **Tasks**, and then **Active Tasks** page.