

**Oracle® Communications
Policy Management**

OSSI XML Interface Definitions Guide

Release 9.4

910-6368-001 Revision B

April 2014

Oracle® Communications OSSI XML Interface Definitions Guide, Release 9.4

Copyright © 2013, 2014, Oracle and/or its affiliates. All rights reserved.

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 or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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 on 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. 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.

Table of Contents

Chapter 1: About This Guide.....	8
Introduction.....	9
How This Guide is Organized.....	9
Scope and Audience.....	9
Documentation Admonishments.....	10
Customer Care Center.....	10
Emergency Response.....	13
Related Publications.....	13
Locate Product Documentation on the Customer Support Site.....	14
Chapter 2: Schema Definitions and Request Specifications.....	15
Schema Definitions.....	16
Request Specification.....	16
Chapter 3: Common Responses and Commands.....	17
Common Responses.....	18
Example XML Response for a Successful Operation.....	18
Example XML Response for an Operation that Fails at the System Level.....	18
Example XML Response for an Operation that Includes an Application-level Failure.....	18
Result Codes.....	19
Common Commands.....	19
Get Version.....	20
Distribute Updates.....	20
Serving Gateway/MCC-MNC Mapping.....	21
Chapter 4: Topology Interface.....	22
Topology Interface Requests.....	23
Network Elements.....	25
Add Network Element.....	25
Update Network Element.....	26
Add Network Element to Group.....	27

Remove Network Element from Group.....	27
Delete Network Element.....	28
Delete Network Element Group.....	29
Query Network Elements.....	29
Paths.....	31
Add Path.....	31
Update Path.....	32
Delete Path.....	33
Query Path.....	33
Applications.....	35
Add Application.....	35
Update Application.....	35
Delete Application.....	36
Query Application.....	37
Traffic Profiles.....	37
Add Traffic Profile.....	37
Update Traffic Profile.....	39
Delete Traffic Profile.....	40
Query Traffic Profile.....	40
Time Periods.....	41
Add Time Period.....	41
Update Time Period.....	42
Delete Time Period.....	43
Query Time Period.....	44
Charging Servers.....	45
Add Charging Server.....	45
Update Charging Server.....	46
Delete Charging Server.....	46
Query Charging Server.....	47
Match List.....	48
Add Match List.....	48
Update Match List.....	49
Delete Match List.....	50
Query Match List.....	50

Chapter 5: Subscriber Interface.....52

Accounts.....	53
Overview.....	53
Add Account.....	53
Update Account.....	54

Delete Account.....	55
Query Account.....	56
Tiers.....	57
Add Tier.....	57
Update Tier.....	58
Delete Tier.....	58
Query Tier.....	59

Chapter 6: Operational Measurements Interface Overview.....61

OM Statistics Scheduled Task.....	62
OM Statistics Requests.....	62
Attributes and Child Tags.....	62
Recorded Timestamp and Request Time Range.....	64
OM Statistics Response Format.....	64
Absolute versus Delta Values.....	64
Empty Data Set.....	65
Timezones and Start/End Times.....	65
Counter Reset and Failover.....	66
Comparisons Between the CMP GUI and OM Statistics.....	67

Chapter 7: Operational Measurement Requests.....68

Operational Measurements Requests Overview.....	69
Topology Update Statistics.....	70
Subscriber Update Statistics.....	71
Policy Server Statistics.....	72
MPE Database Activity Statistics.....	73
Message Processing Statistics.....	74
Network Element Statistics.....	75
Reserve Commit Statistics.....	79
Gate Statistics.....	80
Latency Statistics.....	81
RADIUS-S Statistics.....	90
RADIUS-S Network Element Statistics.....	93
Connected Network Elements Statistics.....	96
Mgpi Statistics.....	97
Diameter Operational Measurements Requests.....	98
Diameter Application Function Statistics.....	98
Diameter Application Function Peer Statistics.....	99
Diameter Charging Function Statistics.....	101
Diameter Charging Function Peer Statistics.....	102

Diameter Policy Charging Enforcement Function Statistics.....	103
Diameter Policy Charging Enforcement Function Peer Statistics.....	105
Diameter Adaptor Statistics.....	106
Diameter Adaptor Peer Statistics.....	108
KPI Statistics.....	110
KPI Interval Statistics.....	112
PCMM CMTS Statistics.....	113
PCMM Network Element Statistics.....	114
DQos CMTS Statistics.....	115
DQos Network Element Statistics.....	116
Connection Error Statistics.....	118
Protocol Error Statistics.....	119

List of Tables

Table 1: Admonishments.....	10
Table 2: OSSI Result Codes.....	19

Chapter 1

About This Guide

Topics:

- *Introduction.....9*
- *How This Guide is Organized.....9*
- *Scope and Audience.....9*
- *Documentation Admonishments.....10*
- *Customer Care Center.....10*
- *Emergency Response.....13*
- *Related Publications.....13*
- *Locate Product Documentation on the Customer Support Site.....14*

This guide describes the Configuration Management Platform (CMP) OSSI XML interface. This interface allows an operator or third party system to programmatically push configuration information to and retrieve operational statistics from the policy server deployment.

Introduction

This guide describes the Configuration Management Platform (CMP) OSSI XML interface. This interface allows an operator or third-party system to programmatically push configuration information to and retrieve operational statistics from the policy server deployment.

Conventions

The following conventions are used throughout this guide:

- **Bold text** in procedures indicates icons, buttons, links, or menu items that you click on.
- *Italic text* indicates variables.
- `Monospace text` indicates text displayed on screen.
- **Monospace bold text** indicates text that you enter exactly as shown.

How This Guide is Organized

The information in this guide is presented in the following order:

- [About This Guide](#) contains general information about this guide, the organization of this guide, and how to get technical assistance.
- [Schema Definitions and Request Specifications](#) describes the Schema Definitions and Request Specifications used within the OSSI XML interface.
- [Common Responses and Commands](#) describes the generic response format for many of the commands used and the commands that are common to the various OSSI interfaces.
- [Topology Interface](#) describes the interface that allows users to manage and query network elements, paths, and other topology-related objects within the system.
- [Subscriber Interface](#) describes the interface that allows users to manage and query subscriber elements within their system.
- [Operational Measurements Interface Overview](#) describes the interface used to retrieve operational counters from the system.
- [Operational Measurement Requests](#) describes the various Operational Measurements (OM) groups and the individual OM statistics that comprise them.

Scope and Audience

This guide is intended for operators or third party systems who are responsible for pushing configuration information to and retrieving operational statistics from a policy server deployment.

Documentation Admonishments

Admonishments are icons and text throughout this manual that alert the reader to assure personal safety, to minimize possible service interruptions, and to warn of the potential for equipment damage.

Table 1: Admonishments

 DANGER	Danger: (This icon and text indicate the possibility of <i>personal injury</i> .)
 WARNING	Warning: (This icon and text indicate the possibility of <i>equipment damage</i> .)
 CAUTION	Caution: (This icon and text indicate the possibility of <i>service interruption</i> .)
 TOPPLE	Topple: (This icon and text indicate the possibility of <i>personal injury</i> .)

Customer Care Center

The Tekelec Customer Care Center is your initial point of contact for all product support needs. A representative takes your call or email, creates a Customer Service Request (CSR) and directs your requests to the Tekelec Technical Assistance Center (TAC). Each CSR includes an individual tracking number. Together with TAC Engineers, the representative will help you resolve your request.

The Customer Care Center is available 24 hours a day, 7 days a week, 365 days a year, and is linked to TAC Engineers around the globe.

Tekelec TAC Engineers are available to provide solutions to your technical questions and issues 7 days a week, 24 hours a day. After a CSR is issued, the TAC Engineer determines the classification of the trouble. If a critical problem exists, emergency procedures are initiated. If the problem is not critical, normal support procedures apply. A primary Technical Engineer is assigned to work on the CSR and provide a solution to the problem. The CSR is closed when the problem is resolved.

Tekelec Technical Assistance Centers are located around the globe in the following locations:

Tekelec - Global

Email (All Regions): support@tekelec.com

- **USA and Canada**

Phone:

1-888-FOR-TKLC or 1-888-367-8552 (toll-free, within continental USA and Canada)

1-919-460-2150 (outside continental USA and Canada)

TAC Regional Support Office Hours:

8:00 a.m. through 5:00 p.m. (GMT minus 5 hours), Monday through Friday, excluding holidays

- **Caribbean and Latin America (CALA)**

Phone:

+1-919-460-2150

TAC Regional Support Office Hours (except Brazil):

10:00 a.m. through 7:00 p.m. (GMT minus 6 hours), Monday through Friday, excluding holidays

- **Argentina**

Phone:

0-800-555-5246 (toll-free)

- **Brazil**

Phone:

0-800-891-4341 (toll-free)

TAC Regional Support Office Hours:

8:00 a.m. through 5:48 p.m. (GMT minus 3 hours), Monday through Friday, excluding holidays

- **Chile**

Phone:

1230-020-555-5468

- **Colombia**

Phone:

01-800-912-0537

- **Dominican Republic**

Phone:

1-888-367-8552

- **Mexico**

Phone:

001-888-367-8552

- **Peru**

Phone:

0800-53-087

- **Puerto Rico**

Phone:

1-888-367-8552 (1-888-FOR-TKLC)

- **Venezuela**

Phone:

0800-176-6497

- **Europe, Middle East, and Africa**

Regional Office Hours:

8:30 a.m. through 5:00 p.m. (GMT), Monday through Friday, excluding holidays

- **Signaling**

Phone:

+44 1784 467 804 (within UK)

- **Software Solutions**

Phone:

+33 3 89 33 54 00

- **Asia**

- **India**

Phone:

+91-124-465-5098 or +1-919-460-2150

TAC Regional Support Office Hours:

10:00 a.m. through 7:00 p.m. (GMT plus 5 1/2 hours), Monday through Saturday, excluding holidays

- **Singapore**

Phone:

+65 6796 2288

TAC Regional Support Office Hours:

9:00 a.m. through 6:00 p.m. (GMT plus 8 hours), Monday through Friday, excluding holidays

Emergency Response

In the event of a critical service situation, emergency response is offered by the Tekelec Customer Care Center 24 hours a day, 7 days a week. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with the Tekelec Customer Care Center.

Related Publications

The Policy Management product set includes the following publications, which provide information for the configuration and use of Policy Management products in the following environments:

Cable

- *Feature Notice*
- *Cable Release Notice*
- *Roadmap to Hardware Documentation*
- *CMP Cable User Guide*
- *Troubleshooting Reference Guide*
- *SNMP User Guide*
- *OSSI XML Interface Definitions Reference Guide*
- *Platform Configuration User Guide*
- *Bandwidth on Demand Application Manager User Guide*
- *PCMM specification PKT-SP-MM-I06* (third-party document, used as reference material for PCMM)

Wireless

- *Feature Notice*
- *Wireless Release Notice*
- *Roadmap to Hardware Documentation*
- *CMP Wireless User Guide*
- *Multi-Protocol Routing Agent User Guide*
- *Troubleshooting Reference Guide*

- *SNMP User Guide*
- *OSSI XML Interface Definitions Reference Guide*
- *Analytics Data Stream Reference*
- *Platform Configuration User Guide*

Wireline

- *Feature Notice*
- *Wireline Release Notice*
- *Roadmap to Hardware Documentation*
- *CMP Wireline User Guide*
- *Troubleshooting Reference Guide*
- *SNMP User Guide*
- *OSSI XML Interface Definitions Reference Guide*
- *Platform Configuration User Guide*

Locate Product Documentation on the Customer Support Site

Access to Tekelec's Customer Support site is restricted to current Tekelec customers only. This section describes how to log into the Tekelec Customer Support site and locate a document. Viewing the document requires Adobe Acrobat Reader, which can be downloaded at www.adobe.com.

1. Log into the [Tekelec Customer Support](#) site.

Note: If you have not registered for this new site, click the **Register Here** link. Have your customer number available. The response time for registration requests is 24 to 48 hours.

2. Click the **Product Support** tab.
3. Use the Search field to locate a document by its part number, release number, document name, or document type. The Search field accepts both full and partial entries.
4. Click a subject folder to browse through a list of related files.
5. To download a file to your location, right-click the file name and select **Save Target As**.

Schema Definitions and Request Specifications

Topics:

- [Schema Definitions.....16](#)
- [Request Specification.....16](#)

To simplify integration, the OSSI XML interface uses an XML over HTTP(s) mechanism for requests and responses. Using XML as a transport allows the data to be represented in a portable, vendor-neutral and readable format.

The interface itself is divided into the following areas:

- Common Interface — defines generalized operations performed on the system.
- Topology Interface — allows users to manage and query network elements and paths.
- Subscriber Interface — allows users to manage and query subscriber elements, such as accounts and tiers.
- Operational Measurements (OM) Interface — allows users to retrieve statistical data from the system.

Each of these interfaces is described in detail in later chapters of this document.

This chapter describes the Schema Definitions and Request Specifications used within the OSSI XML interface.

Schema Definitions

The OSSI XML interface defines specific request and response messages for each of the areas outlined previously. To enable message validation and to accurately specify the syntax of each of the messages, XML schema definitions are provided as follows:

- `OssiXmlRequestResponse.xsd` — defines the schema for request and response formats.
- `OssiXmlCommon.xsd` — defines the schema for global types and requests used in all interfaces.
- `OssiXmlTopology.xsd` — defines the schema for the topology interface.
- `OssiXmlSubscriber.xsd` — defines the schema for the subscriber interface.
- `OssiXmlOm.xsd` — defines the schema for the OM interface.

Request Specification

Each of the requests and responses defined within the previous schema definitions are sent to and received from the Configuration Management Platform (CMP) using HTTP(s) messages. Specifically, an HTTP POST message is sent containing the specific request message. The HTTP response contains a response message indicating status and returning any data as required.

For an application desiring to use the OSSI XML interface, the only requirement is the ability to send an HTTP POST and to process any response. This guide contains examples that use the command line utility `wget` to send an HTTP POST request that contains data specified in an XML file as input and returns an output XML file. The example that follows demonstrates the basic `wget` options; additional options are available but not described here. Please note that the request URL is case sensitive and must be entered as seen here.

```
> wget --post-file=input.xml --output-document=output.xml
"http://1.2.3.4/mi/xmlInterfaceRequest.do?user=test&pwd=test"
```

Where the following describes each parameter:

- `--post-file=input.xml` (Required) — This parameter indicates the request input XML file.
- `--output-document=output.xml` (Optional) — This parameter is used to name the output file. If unspecified, the default filename is the URL string indicated in the `wget` request.
- `http://1.2.3.4/mi/xmlInterfaceRequest.do?user=test&pwd=test` (Required) — The HTTP request URL, including the authentication credentials.
- `--timeout=0` (Optional) — This parameter sets the network timeout to seconds. The default for this value is 900 (15min). A value of 0 disables timeout checking completely.
- `--progress=dot` (Optional) — This parameter is used to display the progress bar on the request.

Chapter 3

Common Responses and Commands

Topics:

- *Common Responses.....18*
- *Result Codes.....19*
- *Common Commands.....19*

This chapter describes the generic response format for many of the commands used and the commands that are common to the various OSSI interfaces.

Common Responses

This section describes the generic response format to many of the commands. The response format follows the generic response tag defined in the XSDs. The following are examples of successful and failed operations.

Example XML Response for a Successful Operation

The following is an example of an XML response for a successful operation:

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Successfully imported 2 network elements.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Example XML Response for an Operation that Fails at the System Level

The following is an example of an XML response for an operation that fails at the system level (for example, a malformed request):

```
<?xml version='1.0' ?>
<Response>
  <Result>103</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Incorrectly formatted XML. The element type
      "NetworkElementStats" must be terminated by the matching end-tag
      "</NetworkElementStats>".</Failure>
  </Command>
</Response>
```

Example XML Response for an Operation that Includes an Application-level Failure

The following is an example of an XML response for an operation that includes an application-level failure:

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 network elements.
      Network Element includes an invalid NetworkElementType.
      Network Element: RouterA</Failure>
  </Command>
</Response>
```

```
</Command>
</Response>
```

Result Codes

Many of the OSSI commands return a numeric result code in their response messages. This code represents the status of the operation at the system level, and shows whether the command itself could be executed or not. The following table describes the possible result codes.

Table 2: OSSI Result Codes

Result Code	Description
0	The command was valid at a system level
100	There was a general failure due to an internal server error
101	The command failed authentication; the account or password does not match between the OSSI and the URL
102	An input stream error occurred (for example, the requested data exceeds maximum size). The maximum size for a single request is 20,000,000 bytes. An example of the error message is: <code>errors.importExport.variableMessageMaxFileSize=Import file exceeds max size of 20M</code>
103	The request is malformed; for example, the XML tags may be invalid. Refer to the XSD of the OSSI query and ensure that no characters appear before the XML header (<code><?xml version="1.0" encoding="UTF-8"?></code>).

Common Commands

This section describes commands that are common to the various OSSI interfaces. These include:

- **Get Version** — allows users to request the OSSI API version number, as a string (for example, “2.0.3”). This enables the user to verify that the OSSI interface being used is the version expected, ensuring that the commands will operate as specified for that version.
- **Distribute Updates** — allows users to trigger an immediate distribution of any pending data changes to the policy servers. This enables the user to queue up many changes in the Configuration Management Platform (CMP) before sending them all at once to the policy servers in the deployment. Alternatively, changes can be distributed one-by-one on a per-command basis. The following types of data are pushed down to the policy servers as part of this request:
 - Topology data, which includes Network Elements and Paths.
 - Tiers
 - Subscriber accounts

- Serving Gateway/MCC-MNC Mapping — allows users to verify input xml files.
- Query OM Stats Setting — allows users to verify the persistent-interval settings for Reset Configuration and Collection Interval.
- Query Absolute Values for Statistics — allows users to retrieve absolute values when **Stats.Interval.Enabled.** is set to false.

Get Version

The following examples show the requests and responses that are defined in the XSDs for the GetVersion tag.

The following is an example of the request that follows the GetVersion tag.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <GetVersion></GetVersion>
</XmlInterfaceRequest>
```

The following example response to the previous request follows the generic Response tag.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command>
    <Command type="XmlInterfaceResponse">
      <Success count="1">3.1.0</Success>
      <Failure count="0"></Failure>
    </Command>
  </Response>
```

Distribute Updates

The following examples show the requests and responses that are defined in the XSDs for the DistributeUpdates tag.

The following is an example of the request that follows the DistributeUpdates tag.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DistributeUpdates></DistributeUpdates>
</XmlInterfaceRequest>
```

The optional ForceSynchronization attribute can be used to force a complete resynchronization of the CMP and Multimedia Policy Engine (MPE) databases. The default value for this attribute is false.

Typically, using this attribute should not be necessary as the system should always be kept in-sync automatically; it is merely provided as a fail-safe measure for unforeseen circumstances. Also note

that there is a performance impact from using this attribute, so it should not be used unless deemed necessary by Tekelec support. The following is an example of this attribute:

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DistributeUpdates ForceSynchronization="true"></DistributeUpdates>
</XmlInterfaceRequest>
```

The following example response to the previous request follows the generic Response tag.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="3">Topology updates successful. Tier updates successful.
    Account updates successful.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Serving Gateway/MCC-MNC Mapping

The following examples show the requests and responses that are defined in the XSDs for the SgwMapping tag.

The following is an example of the request that follows the SgwMapping tag.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<XmlInterfaceRequest>
  <AddSgwMapping>
    <SgwMapping>
      <Name>test</Name>
      <Description>description for the mcc-mnc
      mapping</Description>
      <MccMnc>111111</MccMnc>
      <SgwIpAddress>
        <IpAddress>10.0.0.129</IpAddress>
      </SgwIpAddress>
    </SgwMapping>
  </AddSgwMapping>
</XmlInterfaceRequest>
```

The following example response to the previous request follows the generic Response tag.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully imported 1
    Serving Gateway/MCC-MNC Mappings.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Chapter 4

Topology Interface

Topics:

- *Topology Interface Requests.....23*
- *Network Elements.....25*
- *Paths.....31*
- *Applications.....35*
- *Traffic Profiles.....37*
- *Time Periods.....41*
- *Charging Servers.....45*
- *Match List.....48*

The Topology Interface allows users to manage and query network elements, paths, and other topology-related objects within the system.

Topology Interface Requests

The Topology Interface consists of the following requests:

- Network Elements — A network element represents a node in the network, such as a router, CMTS, PDSN, B-RAS, and so on. The network elements can be organized into logical groups that subdivide the network space.
 - Add Network Element — Allows users to add new network elements and network element groups. If a network element already exists with the given identifier (Name or NeId), then this is considered an update and the new network element replaces the existing one. All relationships between that network element and other objects in the system will be maintained.
 - Update Network Element — Allows users to update existing network elements and network element groups. If no object exists with the given identifier (Name or NeId), then this is a failure condition and the object is not saved.
 - Add Network Element To Group — Allows users to associate a network element with a group.
 - Remove Network Element From Group — Allows users to remove a network element from within a group.
 - Delete Network Element — Allows users to delete a network element.
 - Delete Network Element Group — Allows users to delete a network element group.
 - Query Network Element — Allows users to query all network elements or a specific network element, returning their configured attributes.
- Paths — A path represents a route through the network, and is identified by source and destination network elements.
 - Add Path — Allows users to add new paths to the system. If a path already exists with the given identifier (Name), then this is considered an update and the new path replacing the existing one. All relationships between that path and other objects in the system are maintained.
 - Update Path — Allows users to update paths within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - Delete Path — Removes a path from the system.
 - Query Path — Allows users to query a specific path or list of paths, returning their configured attributes.
- Applications — An application is a service in the network (for example, VoIP) for which quality of service is managed.
 - AddApplication — Allows users to add new applications to the system. If an application already exists with the given identifier (Name), then this is considered an update and the new application replaces the existing one. All relationships between that application and other objects in the system are maintained.
 - UpdateApplication — Allows users to update application within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - DeleteApplication — Removes an application from the system.
 - QueryApplication — Allows users to query a specific application or list of traffic profiles, returning their configured attributes.
- Traffic Profiles — A traffic profile is a set of parameters and their values, which are used in protocol messages to define quality of service for sessions.

- **AddTrafficProfile** — Allows users to add new traffic profiles to the system. If a traffic profile already exists with the given identifier (Name), then this is considered an update and the new traffic profile replaces the existing one. All relationships between that traffic profile and other objects in the system are maintained.
- **UpdateTrafficProfile** — Allows users to update traffic profiles within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
- **DeleteTrafficProfile** — Removes a traffic profile from the system.
- **QueryTrafficProfile** — Allows users to query a specific traffic profile or list of traffic profiles, returning their configured attributes.
- **Time Periods** — A time period defines a set of related time slots, which receives the same quality of service (for example, “the peak time of network usage is Monday through Friday from 9:00 am to 5:00 pm”).
 - **AddTimePeriod** — Allows users to add new time periods to the system. If a time period already exists with the given identifier (Name), then this is considered an update and the new time period replaces the existing one. All relationships between that time period and other objects in the system are maintained.
 - **UpdateTimePeriod** — Allows users to update time periods within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - **DeleteTimePeriod** — Removes a time period from the system.
 - **QueryTimePeriod** — Allows users to query a specific time period or list of time periods, returning their configured attributes.
- **Charging Servers** — A charging server is a network element that processes accounting information for billing purposes.
 - **AddChargingServer** — Allows users to add new charging servers to the system. If a charging server already exists with the given identifier (Name), then this is considered an update and the new charging server replaces the existing one. All relationships between that charging server and other objects in the system are maintained.
 - **UpdateChargingServer** — Allows users to update charging servers within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - **DeleteChargingServer** — Removes a charging server from the system.
 - **QueryChargingServer** — Allows users to query a specific charging server or list of charging servers, returning their configured attributes.
- **Match List** — A match list is a set of values that can be used within one or more policy definitions.
 - **AddMatchList** — Allows users to add new match list to the system. If a match list already exists with the given identifier (Name), then this is considered an update and the new match list replaces the existing one. All relationships between that match list and other objects in the system are maintained.
 - **UpdateMatchList** — Allows users to update match lists within the system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - **DeleteMatchList** — Removes a match list from the system.
 - **QueryMatchList** — Allows users to query a specific match list or list of match lists, returning their configured attributes.

Network Elements

A network element represents a node in the network, such as a router, CMTS, PDSN, B-RAS, and so on. The network elements can be organized into logical groups that subdivide the network space. The following sections describe the available network tags.

Add Network Element

The following examples show the request and response that are defined in the XSDs for the AddNetworkElement tag.

Request

This request follows the AddNetworkElement tag defined in the XSDs.

The following example creates a new Network Element to represent a router in the network. It also creates a group, and puts the router in that group.

The router contains the following attributes:

- Name (unique identifier): Router 23
- Description: Core router for the north east
- HostName (IP address or DNS hostname): 12.1.1.1
- NE-ID (another unique identifier): 112222
- Capacity (aggregate capacity in bps): 456000

The group contains the following attributes:

- Name: NE Group 1
- Description: All core routers

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <AddNetworkElement>
    <NetworkElement>
      <Name>Router 23</Name>
      <Description>Core router for the north east</Description>
      <HostName>12.1.1.1</HostName>
      <NeId>112222</NeId>

      <NetworkElementType>Router</NetworkElementType>
      <NeSubType></NeSubType>
      <Capacity>456000</Capacity>
    </NetworkElement>
    <NetworkElementGroup>
      <Name>NE Group 1</Name>
      <Description>All core routers</Description>
      <RootGroup>true</RootGroup>
      <ElementRef>
        <Name>router 23</Name>
        <SubGroup>>false</SubGroup>
      </ElementRef>
    </NetworkElementGroup>
  </AddNetworkElement>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

The example XML response that follows represents a successful operation that creates a new network element and/or groups:

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Successfully imported 1 network elements.
    Successfully imported 1 group.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

The example XML response that follows represents an operation that includes both a success and a failure (for example, an attempt to update a group's description field (success), and add an unknown network element "router 24" to the group "NE Group 1" (failed)).

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 groups.</Success>
    <Failure count="1">Failed to update 1 groups. Network Element Group
    update returned an error attempting to add the following: router 24
    Group NE Group 1</Failure>
  </Command>
</Response>
```

Update Network Element

The following examples show the request and response that are defined in the XSDs for the UpdateNetworkElement tag.

Request

This request follows the UpdateNetworkElement tag defined in the XSDs.

The following example changes the capacity attribute of the network element named "Router 23" to the new value: 4567000.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <UpdateNetworkElement>
    <NetworkElement>
      <Name>Router 23</Name>
      <Capacity>4567000</Capacity>
    </NetworkElement>
  </UpdateNetworkElement>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 network elements.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Add Network Element to Group

The following examples show the request and response that are defined in the XSDs for the AddNetworkElementToGroup tag.

Request

This request follows the AddNetworkElementToGroup tag defined in the XSDs.

The following example adds three network elements (Router 2, Router 34, and Router 131) to a group (Group 1).

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <AddNetworkElementToGroup>
    <GroupName>Group 1</GroupName>
    <Name>Router 2</Name>
    <Name>Router 34</Name>
    <Name>Router 131</Name>
  </AddNetworkElementToGroup>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="3">Added 3 elements to a group.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Remove Network Element from Group

The following examples show the request and response that are defined in the XSDs for the RemoveNetworkElementFromGroup tag.

Request

This request follows the RemoveNetworkElementFromGroup tag defined in the XSDs.

The following example removes two network elements (Router 34 and Router 131) from a group (Group 1).

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <RemoveNetworkElementFromGroup>
    <GroupName>Group 1</GroupName>
    <Name>Router 34</Name>
    <Name>Router 131</Name>
  </RemoveNetworkElementFromGroup>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Removed 2 elements from group.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Network Element

The following examples show the request and response that are defined in the XSDs for the DeleteNetworkElement tag.

Request

This request follows the DeleteNetworkElement tag defined in the XSDs.

The following example deletes two network elements (Node 1 and Node 2).

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeleteNetworkElement>
    <Name>Node1</Name>
    <Name>Node2</Name>
  </DeleteNetworkElement>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Deleted 2 elements.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

```

    </Command>
</Response>

```

Delete Network Element Group

The following examples show the request and response that are defined in the XSDs for the DeleteNetworkElementGroup tag.

Request

This request follows the DeleteNetworkElementGroup tag defined in the XSDs.

The following example deletes two network element groups (Group 1 and Group 2).

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeleteNetworkElementGroup>
    <Name>Group 1</Name>
    <Name>Group 2</Name>
  </DeleteNetworkElementGroup>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Deleted 2 network element groups.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Query Network Elements

The following examples show the request and response that are defined in the XSDs for the QueryNetworkElement tag.

Request

This request follows the QueryNetworkElement tag defined in the XSDs.

Example 1 — XML for querying multiple network elements by name.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryNetworkElement>
    <Name>CMTS A</Name>
    <Name>CMTS B</Name>
  </QueryNetworkElement>
</XmlInterfaceRequest>

```

Example 2 — XML for querying all network elements and groups.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryNetworkElement/>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> NetworkElement tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <NetworkElement>
    <Name>CMTS A</Name>
    <Description></Description>
    <HostName>10.0.2.6</HostName>
    <BackupHostName></BackupHostName>
    <NeId></NeId>

  <NetworkElementType>Router</NetworkElementType>
    <NeSubType></NeSubType>
    <Managed>true</Managed>
    <PcmmPort>3918</PcmmPort>
    <DqosPort>2126</DqosPort>
    <ReadCommunity>public</ReadCommunity>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
    <AdminState>0</AdminState>
    <LearnedSubnet>
      <IpAddress>10.128.3.0</IpAddress>
      <Mask>24</Mask>
    </LearnedSubnet>
    <LearnedSubnet>
      <IpAddress>10.254.3.0</IpAddress>
      <Mask>24</Mask>
    </LearnedSubnet>
  </NetworkElement>
  <NetworkElement>
    <Name>CMTS B</Name>
    <Description></Description>
    <HostName>10.0.3.2</HostName>
    <BackupHostName></BackupHostName>
    <NeId></NeId>

  <NetworkElementType>Router</NetworkElementType>
    <NeSubType></NeSubType>
    <Managed>true</Managed>
    <PcmmPort>3918</PcmmPort>
    <DqosPort>2126</DqosPort>
    <ReadCommunity>public</ReadCommunity>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
    <AdminState>0</AdminState>
    <LearnedSubnet>
      <IpAddress>10.128.1.0</IpAddress>
      <Mask>24</Mask>
    </LearnedSubnet>
```

```

    <LearnedSubnet>
      <IpAddress>10.254.1.0</IpAddress>
      <Mask>24</Mask>
    </LearnedSubnet>
    <LearnedSubnet>
      <IpAddress>11.129.0.0</IpAddress>
      <Mask>16</Mask>
    </LearnedSubnet>
  </NetworkElement>
</ConfigurationData>

```

Paths

A path represents a route through the network, and is identified by source and destination network elements.

Add Path

The following examples show the request and response that are defined in the XSDs for the AddPath tag.

Request

This request follows the AddPath tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <AddPath>
    <Path>
      <Name>Path1</Name>
      <Description/>
      <Hops>
        <Hop>
          <NeName>arris1</NeName>
        </Hop>
        <Hop>
          <NeName>cisco1</NeName>
        </Hop>
        <Hop>
          <NeName>moto1</NeName>
        </Hop>
      </Hops>
    </Path>
    <Path>
      <Name>Path2</Name>
      <Description/>
      <Hops>
        <Hop>
          <NeName>moto2</NeName>
        </Hop>
        <Hop>
          <NeName>cisco2</NeName>
        </Hop>
        <Hop>
          <NeName>arris2</NeName>
        </Hop>
      </Hops>
    </Path>
  </AddPath>
</XmlInterfaceRequest>

```

```

        </Hop>
      </Hops>
    </Path>
  <Path>
    <Name>Path3</Name>
    <Description/>
    <Hops>
      <Hop>
        <NeName>arris3</NeName>
      </Hop>
      <Hop>
        <NeName>moto3</NeName>
      </Hop>
    </Hops>
  </Path>
</AddPath>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="3">Successfully imported 3 paths.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Update Path

The following examples show the request and response that are defined in the XSDs for the UpdatePath tag.

Request

This request follows the UpdatePath tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <UpdatePath>
    <Path>
      <Name>Path1</Name>
      <Hops>
        <Hop>
          <NeName>motol</NeName>
        </Hop>
        <Hop>
          <NeName>cisco1</NeName>
        </Hop>
        <Hop>
          <NeName>arris2</NeName>
        </Hop>
      </Hops>
    </Path>
  </UpdatePath>
</XmlInterfaceRequest>

```

```
</UpdatePath>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 paths.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Path

The following examples show the request and response that are defined in the XSDs for the DeletePath tag.

Request

This request follows the DeletePath tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeletePath>
    <Name>Path1</Name>
    <Name>Path2</Name>
  </DeletePath>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Deleted 2 paths.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Query Path

The following examples show the request and response that are defined in the XSDs for the QueryPath tag.

Request

This request follows the QueryPath tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryPath>
    <Name>Path1</Name>
    <Name>Path2</Name>
  </QueryPath>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> Path tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="3.0.0">
  <Path>
    <Name>Path1</Name>
    <Description/>
    <Source>
      <Address>1.1.1.2</Address>
      <Mask>24</Mask>
    </Source>
    <Destination>
      <Address>3.3.3.2</Address>
      <Mask>24</Mask>
    </Destination>
    <Hops>
      <Hop>
        <NeName>arris</NeName>
      </Hop>
      <Hop>
        <NeName>cisco</NeName>
      </Hop>
      <Hop>
        <NeName>moto</NeName>
      </Hop>
    </Hops>
  </Path>
  <Path>
    <Name>Path2</Name>
    <Description/>
    <Source>
      <Address>3.3.3.2</Address>
      <Mask>24</Mask>
    </Source>
    <Destination>
      <Address>1.1.1.2</Address>
      <Mask>24</Mask>
    </Destination>
    <Hops>
      <Hop>
        <NeName>moto</NeName>
      </Hop>
      <Hop>
        <NeName>cisco</NeName>
      </Hop>
      <Hop>
        <NeName>arris</NeName>
      </Hop>
    </Hops>
```

```

    </Path>
</ConfigurationData>

```

Applications

An application is a service in the network (for example, VoIP) for which Quality of Service (QoS) is managed.

Add Application

The following examples show the request and response that are defined in the XSDs for the AddApplication tag.

Request

This request follows the AddApplication tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddApplication>
    <Application>
      <Name>VoD</Name>
      <Description>Video on Demand</Description>
      <LatencySensitive>true</LatencySensitive>
      <Tracked>true</Tracked>
      <UpGatesPerSession>1</UpGatesPerSession>
      <DownGatesPerSession>1</DownGatesPerSession>
      <TrackingTimeoutTime>240</TrackingTimeoutTime>
      <TrackingTimeoutUnit>1</TrackingTimeoutUnit>
    </Application>
  </AddApplication>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully imported 1 applications.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Update Application

The following examples show the request and response that are defined in the XSDs for the UpdateApplication tag.

Request

This request follows the UpdateApplication tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateApplication>
    <Application>
      <Name>VoD</Name>
      <Description>Video on Demand</Description>
      <LatencySensitive>>false</LatencySensitive>
      <UpGatesPerSession>3</UpGatesPerSession>
    </Application>
  </UpdateApplication>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 applications.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Application

The following examples show the request and response that are defined in the XSDs for the DeleteApplication tag.

Request

This request follows the DeleteApplication tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteApplication>
    <Name>VoD</Name>
  </DeleteApplication>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Deleted 1 applications.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

```

    </Command>
</Response>

```

Query Application

The following examples show the request and response that are defined in the XSDs for the QueryApplication tag.

Request

This request follows the QueryApplication tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <QueryApplication/>
</XmlInterfaceRequest>

```

Response

The response to this request follows the ConfigurationData -> Application tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version=" " >
  <Application>
    <Name>VoD</Name>
    <Description>Video on Demand</Description>
    <LatencySensitive>true</LatencySensitive>
    <Tracked>true</Tracked>
    <UpGatesPerSession>1</UpGatesPerSession>
    <DownGatesPerSession>1</DownGatesPerSession>
    <TrackingTimeoutTime>240</TrackingTimeoutTime>
    <TrackingTimeoutUnit>1</TrackingTimeoutUnit>
  </Application>
</ConfigurationData>

```

Traffic Profiles

A traffic profile is a set of parameters and their values, which are used in protocol messages to define QoS for sessions.

Add Traffic Profile

The following examples show the request and response that are defined in the XSDs for the AddTrafficProfile tag.

Note: The Sd protocol supports only App Detection Control (ADC) profiles.

Request

This request follows the AddTrafficProfile tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
XmlInterfaceRequest
  <AddTrafficProfile>
    <QosProfile>
      <Name>default-rule-traffic-profile</Name>
      <ProfileType>0</ProfileType>
      <QosProfileType>Diameter QoS</QosProfileType>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.profileTypeNumber
          </ProfileParamUid>
        <ParamValue>10</ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.qci
          </ProfileParamUid>
        <ParamValue>3</ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.maxAuthUL
          </ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.maxAuthDL
          </ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.minRateUL
          </ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.minRateDL
          </ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.ARP.PriorityLevel
          </ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.ARP
          .PreemptionCapability</ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile.ARP
          .PreemptionVulnerability</ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
      <QosProfileProfileParam>
        <ProfileParamUid>param.diameter.qosProfile
          .ResourceAllocationNotification</ProfileParamUid>
        <ParamValue></ParamValue>
      </QosProfileProfileParam>
    </QosProfile>
    <QosProfileGroup>
      <Name>group1</Name>
      <Description></Description>
    </QosProfileGroup>
  </AddTrafficProfile>
</XmlInterfaceRequest>
```

```

    <RootGroup>true</RootGroup>
    <ElementRef>
      <Name>default-rule-traffic-profile</Name>
      <SubGroup>>false</SubGroup>
    </ElementRef>
  </QosProfileGroup>
</AddTrafficProfile>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully imported 1 Traffic Profile(s).</Success>

    <Failure count="0"></Failure>
  </Command>
</Response>

```

Update Traffic Profile

The following examples show the request and response that are defined in the XSDs for the UpdateTrafficProfile tag.

Request

This request follows the UpdateTrafficProfile tag defined in the XSDs.

```

<<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateTrafficProfile>
    <QosProfile>
      <Name>default-rule-traffic-profile</Name>
      <ProfileType>0</ProfileType>
      <QosProfileType>Second Predefined PCC Rule</QosProfileType>
    </QosProfile>
  </UpdateTrafficProfile>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 Traffic Profile(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Delete Traffic Profile

The following examples show the request and response that are defined in the XSDs for the DeleteTrafficProfile tag.

Request

This request follows the DeleteTrafficProfile tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteTrafficProfile>
    <Name>default-rule-traffic-profile</Name>
  </DeleteTrafficProfile>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Deleted 1 Traffic Profile(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Query Traffic Profile

The following examples show the request and response that are defined in the XSDs for the QueryTrafficProfile tag that queries all traffic profiles:

Request

This request follows the QueryTrafficProfile tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <QueryTrafficProfile/>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> QosProfile tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <QosProfile>
    <Name>default-rule-traffic-profile</Name>
    <ProfileType>0</ProfileType>
    <QosProfileType>Predefined PCC Rule</QosProfileType>
```

```

    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.profileTypeNumber
    </ProfileParamUid>
    <ParamValue>11</ParamValue>
    </QosProfileProfileParam>
    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.name
    </ProfileParamUid>
    <ParamValue>PCRF-DEFAULT</ParamValue>
    </QosProfileProfileParam>
    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.description
    </ProfileParamUid>
    <ParamValue>This is the static rule that gets applied by default
    </ParamValue>
    </QosProfileProfileParam>
  </QosProfile>
  <QosProfile>
    <Name>p2p-rule-traffic-profile</Name>
    <ProfileType>0</ProfileType>
    <QosProfileType>Predefined PCC Rule</QosProfileType>
    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.profileTypeNumber
    </ProfileParamUid>
    <ParamValue>11</ParamValue>
    </QosProfileProfileParam>
    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.name </ProfileParamUid>
    <ParamValue>P2P-RATE-LIMIT-64K</ParamValue>
    </QosProfileProfileParam>
    <QosProfileProfileParam>
    <ProfileParamUid>param.diameter.pccPredefRule.description
    </ProfileParamUid>
    <ParamValue>Rule limits bit-torrent and other peer to peer applications
    to 64Kbps downstream</ParamValue>
    </QosProfileProfileParam>
  </QosProfile>
</ConfigurationData>

```

Time Periods

A time period defines a set of related time slots, which receives the same QoS (for example, “the peak time of network usage is Monday through Friday from 9:00 am to 5:00 pm”).

Add Time Period

The following examples show the request and response that are defined in the XSDs for the AddTimePeriod tag.

Request

This request follows the AddTimePeriod tag defined in the XSDs.

The StartTime and EndTime tags represent a time-of-day value expressed in milliseconds. For example, “01:30” time-of-day corresponds to 5,400,000 ((60+30)*60*1000) when expressed in milliseconds.

The Mask tag represents the day-of-week expressed in the form of a bitmask. The value for each day-of-week corresponds to a power of two: Sunday is 1, Monday is 2, Tuesday is 4, Wednesday is 8, Thursday is 16, Friday is 32, and Saturday is 64. To specify a set of days, add together the values associated with the respective days. For example, "Monday and Wednesday" corresponds to a value of 10 (i.e. 2 + 8).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddTimePeriod>
    <TimePeriod>
      <Name>Weekday Peak</Name>
      <Description></Description>
      <Precedence>1</Precedence>
      <TimeSlot>
        <Mask>32</Mask>
        <StartTime>24300000</StartTime>
        <EndTime>39600000</EndTime>
        <StartDate></StartDate>
        <EndDate></EndDate>
      </TimeSlot>
    </TimePeriod>
  </AddTimePeriod>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully imported 1 Time Period(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Update Time Period

The following examples show the request and response that are defined in the XSDs for the UpdateTimePeriod tag.

Request

This request follows the UpdateTimePeriod tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateTimePeriod>
    <TimePeriod>
      <Name>Weekday Peak</Name>
      <Description></Description>
      <Precedence>44</Precedence>
      <TimeSlot>
        <Mask>32</Mask>
        <StartTime>24300000</StartTime>
```

```

        <EndTime>39600000</EndTime>
        <StartDate></StartDate>
        <EndDate></EndDate>
    </TimeSlot>
</TimePeriod>
</UpdateTimePeriod>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 Time Period(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Delete Time Period

The following examples show the request and response that are defined in the XSDs for the DeleteTimePeriod tag.

Request

This request follows the DeleteTimePeriod tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteTimePeriod>
    <Name>Weekday Peak</Name>
  </DeleteTimePeriod>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Deleted 1 Time Period(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Query Time Period

The following examples show the request and response that are defined in the XSDs for the QueryTimePeriod tag.

Request

This request follows the QueryTimePeriod tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <QueryTimePeriod/>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> TimePeriod tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <TimePeriod>
    <Name>Weekday Peak</Name>
    <Description></Description>
    <Precedence>1</Precedence>
    <TimeSlot>
      <Mask>32</Mask>
      <StartTime>24300000</StartTime>
      <EndTime>39600000</EndTime>
      <StartDate></StartDate>
      <EndDate></EndDate>
    </TimeSlot>
  </TimePeriod>
  <TimePeriod>
    <Name>Weekend Peak</Name>
    <Description></Description>
    <Precedence>7</Precedence>
    <TimeSlot>
      <Mask>32</Mask>
      <StartTime>25200000</StartTime>
      <EndTime>37800000</EndTime>
      <StartDate></StartDate>
      <EndDate></EndDate>
    </TimeSlot>
  </TimePeriod>
  <TimePeriod>
    <Name>Weekend Offpeak</Name>
    <Description></Description>
    <Precedence>0</Precedence>
    <TimeSlot>
      <Mask>32</Mask>
      <StartTime>23400000</StartTime>
      <EndTime>39600000</EndTime>
      <StartDate></StartDate>
      <EndDate></EndDate>
    </TimeSlot>
  </TimePeriod>
</ConfigurationData>
```

Charging Servers

A charging server is a network element that processes accounting information for billing purposes.

Add Charging Server

The following examples show the request and response that are defined in the XSDs for the AddChargingServer tag.

Request

This request follows the AddChargingServer tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddChargingServer>
    <ChargingServer>
      <Name>chargingServer_import1</Name>
      <Description></Description>
      <Security>true</Security>
      <HostName>80.20.20.101</HostName>
      <Port>4040</Port>
      <Transport>udp</Transport>
      <Protocol>radius</Protocol>
    </ChargingServer>
    <ChargingServer>
      <Name>chargingServer_import2</Name>
      <Description></Description>
      <Security>>false</Security>
      <HostName>12.6.43.201</HostName>
      <Port>7089</Port>
      <Transport>tcp</Transport>
      <Protocol>tacacs+</Protocol>
    </ChargingServer>
  </AddChargingServer>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Successfully imported 2 Charging Server(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Update Charging Server

The following examples show the request and response that are defined in the XSDs for the UpdateChargingServer tag.

Request

This request follows the UpdateChargingServer tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateChargingServer>
    <ChargingServer>
      <Name>chargingServer_3</Name>
      <Description></Description>
      <Security>true</Security>
      <HostName>80.20.20.100</HostName>
      <Port>6040</Port>
      <Transport>tcp</Transport>
      <Protocol>radius</Protocol>
    </ChargingServer>
  </UpdateChargingServer>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 Charging Server(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Charging Server

The following examples show the request and response that are defined in the XSDs for the DeleteChargingServer tag.

Request

This request follows the DeleteChargingServer tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteChargingServer>
    <Name>chs1</Name>
  </DeleteChargingServer>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Deleted 1 Charging Server(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Query Charging Server

The following examples show the request and response that are defined in the XSDs for the QueryChargingServer tag.

Request

This request follows the QueryChargingServer tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryChargingServer>
    <Name>chargingServer_3</Name>
    <Name>chargingServer_4</Name>
    <Name>chargingServer_import1</Name>
    <Name>chargingServer_import2</Name>
  </QueryChargingServer>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> ChargingServer tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <ChargingServer>
    <Name>chargingServer_3</Name>
    <Description></Description>
    <Security>true</Security>
    <HostName>80.20.20.100</HostName>
    <Port>4040</Port>
    <Transport>udp</Transport>
    <Protocol>radius</Protocol>
  </ChargingServer>
  <ChargingServer>
    <Name>chargingServer_4</Name>
    <Description></Description>
    <Security>>false</Security>
    <HostName>12.6.43.200</HostName>
    <Port>7089</Port>
    <Transport>tcp</Transport>
    <Protocol>tacacs+</Protocol>
  </ChargingServer>
  <ChargingServer>
    <Name>chargingServer_import1</Name>
    <Description></Description>
```

```

    <Security>true</Security>
    <HostName>80.20.20.101</HostName>
    <Port>4040</Port>
    <Transport>udp</Transport>
    <Protocol>radius</Protocol>
  </ChargingServer>
  <ChargingServer>
    <Name>chargingServer_import2</Name>
    <Description></Description>
    <Security>>false</Security>
    <HostName>12.6.43.201</HostName>
    <Port>7089</Port>
    <Transport>tcp</Transport>
    <Protocol>tacacs+</Protocol>
  </ChargingServer>
</ConfigurationData>

```

Match List

The match list is a set of values that can be used within one or more policy definitions. When the user writes a policy rule to compare against a value in a list, the list or set of lists to be used for comparison is explicitly specified. For example:

“Where the SGSNIPAddress is contained in MaltaSgsnList, GreekSgsnList”

The Match List Type determines what kind of values can be specified in that list and how those values are matched within the policy rule. Types can be “IPv4 Address” (allows address and subnet), “IPv6 Address”(allows address and prefix), “String”, and “Wildcard”.

Add Match List

The following examples show the request and response that are defined in the XSDs for the AddMatchList tag.

Request

This request follows the AddMatchList tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddMatchList>
    <TokenList>
      <Name>match_list_01a_sgsn</Name>
      <Description>This is a match list that will match the given
        SGSN</Description>
      <ItemType>3</ItemType>
      <TokenListItem>
        <Value>44.44.44.44</Value>
      </TokenListItem>
      <TokenListItem>
        <Value>55.55.55.55</Value>
      </TokenListItem>
      <TokenListItem>
        <Value>66.66.66.66</Value>
      </TokenListItem>
    </TokenList>
  </AddMatchList>
</XmlInterfaceRequest>

```

```

    </TokenListItem>
  </TokenList>
<TokenList>
  <Name>match_list_01b_sgsn_black</Name>
  <Description />
  <ItemType>1</ItemType>
  <TokenListItem>
    <Value>44.44.44.44</Value>
  </TokenListItem>
  <TokenListItem>
    <Value>55.55.55.55</Value>
  </TokenListItem>
  <TokenListItem>
    <Value>66.66.66.66</Value>
  </TokenListItem>
</TokenList>
</AddMatchList>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Successfully imported 2 Match List(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Update Match List

The following examples show the request and response that are defined in the XSDs for the UpdateMatchList tag.

Request

This request follows the UpdateMatchList tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
<UpdateMatchList>
  <TokenList>
    <Name>match_list_01a_sgsn</Name>
    <Description>This is an updated match list that will match the
      given SGSN</Description>
    <ItemType>3</ItemType>
    <TokenListItem>
      <Value>44.44.44.77</Value>
    </TokenListItem>
    <TokenListItem>
      <Value>66.66.66.77</Value>
    </TokenListItem>
  </TokenList>
</UpdateMatchList>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 Match List(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Match List

The following examples show the request and response that are defined in the XSDs for the DeleteMatchList tag.

Request

This request follows the DeleteMatchList tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteMatchList>
    <Name>match_list_01a_sgsn</Name>
  </DeleteMatchList>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Deleted 1 Match List(s).</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Query Match List

The following examples show the request and response that are defined in the XSDs for the QueryMatchList tag.

Request

This request follows the QueryMatchList tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
```

```

    <QueryMatchList>
      <Name>match_list_01a_sgsn</Name>
      <Name>match_list_01b_sgsn_black</Name>
    </QueryMatchList>
  </XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <TokenList>
    <Name>match_list_01a_sgsn</Name>
    <Description>This is a match list that will match the given
      SGSN</Description>
    <ItemType>3</ItemType>
    <TokenListItem>
      <Value>44.44.44.77</Value>
    </TokenListItem>
    <TokenListItem>
      <Value>66.66.66.77</Value>
    </TokenListItem>
  </TokenList>
  <TokenList>
    <Name>match_list_01b_sgsn_black</Name>
    <Description />
    <ItemType>1</ItemType>
    <TokenListItem>
      <Value>44.44.44.44</Value>
    </TokenListItem>
    <TokenListItem>
      <Value>55.55.55.55</Value>
    </TokenListItem>
    <TokenListItem>
      <Value>66.66.66.66</Value>
    </TokenListItem>
  </TokenList>
</ConfigurationData>

```

Chapter 5

Subscriber Interface

Topics:

- [Accounts.....53](#)
- [Tiers.....57](#)

The Subscriber Interface allows users to manage and query subscriber elements within their system. Subscriber data includes accounts and tiers.

Accounts

An account represents a billable entity that contains the individual users.

Overview

The Subscriber Interface consists of the following requests:

- Accounts — An account represents a billable entity that contains the individual users.
 - Add Account — Allows users to add and update subscriber accounts within their system. If an account already exists with the given identifier (Account ID), then this is considered an update and the new account replaces the existing one. All relationships between that account and other objects in the system (for example, tiers) are maintained.
 - Update Account — Allows users to update accounts in their system. If no object exists with the given identifier (Account ID), then this is a failure condition and the object is not saved.
 - Delete Account — Allows users to delete an account.
 - Query Account — Allows users to query a specific account or list of accounts, returning their configured attributes.
- Tiers — A tier represents a class of service or a logical grouping of subscriber accounts.
 - Add Tier — Allows users to add new tiers to their system. If a tier already exists with the given identifier (Name), then this is considered an update and the new tier replaces the existing one. All relationships between that tier and other objects in the system (for example, accounts) are maintained.
 - Update Tier — Allows users to update tiers within their system. If no object exists with the given identifier (Name), then this is a failure condition and the object is not saved.
 - Delete Tier — Allows users to delete a tier.
 - Query Tier — Allows users to query a specific tier or list of tiers, returning their configured attributes.

Add Account

The following examples show the request and response that are defined in the XSDs for the AddAccount tag.

Request

This request follows the AddAccount tag defined in the XSDs.

The following example creates an account with the following attributes:

- Account ID: 000123
- Association to a network element named: Node1
- One user in this account, with the following attributes:
 - User ID: bill_smith@company.com
- One cable modem in this account, with the following attributes:

- MAC address: 10:10:10:10:10:11
 - IP address: 30.0.0.3
 - Two CPEs behind this cable modem: 10.0.0.3, 20.0.0.3
- Association to a tier named: Premium

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
<AddAccount>
  <Account>
    <AccountId>000123</AccountId>
    <NetworkElementName>Node1</NetworkElementName>
    <EndUser>
      <EndUserId>john_smith@company.com</EndUserId>
    </EndUser>
    <CableModem>
      <CmMacAddr>10:10:10:10:10:11</CmMacAddr>
      <CmIpAddr>30.0.0.3</CmIpAddr>
      <Cpe>
        <CpeIpAddr>10.0.0.3</CpeIpAddr>
      </Cpe>
      <Cpe>
        <CpeIpAddr>20.0.0.3</CpeIpAddr>
      </Cpe>
    </CableModem>
    <TierRef>
      <Name>Premium</Name>
    </TierRef>
  </Account>
</AddAccount>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse" >
    <Success count="1">Successfully imported 1 accounts.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Update Account

The following examples show the request and response that are defined in the XSDs for the UpdateAccount tag.

Request

This request follows the UpdateAccount tag defined in the XSDs.

The following example updates the account identified by ID 000123 to add a new user with ID jane_doe@company.com.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <UpdateAccount>
    <Account>
      <AccountId>000123</AccountId>
      <EndUser operator='add'>
        <EndUserId>jane_doe@company.com</EndUserId>
      </EndUser>
    </Account>
  </UpdateAccount>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 accounts.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Delete Account

The following examples show the request and response that are defined in the XSDs for the DeleteAccount tag.

Request

This request follows the DeleteAccount tag defined in the XSDs.

The following example deletes two accounts: Account1 and Account2.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeleteAccount>
    <AccountId>Account1</AccountId>
    <AccountId>Account2</AccountId>
  </DeleteAccount>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
```

```

    <Success count="2">Deleted 2 accounts.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Query Account

The following examples show the request and response that are defined in the XSDs for the QueryAccount tag.

Request

This request follows the QueryAccount tag defined in the XSDs.

The following example queries multiple accounts by account ID.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryAccount>
    <AccountId>000123</AccountId>
  </QueryAccount>
</XmlInterfaceRequest>

```

Response

The response to this request follows the ConfigurationData -> Account tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="">
  <Account>
    <AccountId>000123</AccountId>
    <NetworkElementName>Node1</NetworkElementName>
    <EndUser>
      <EndUserId>john_smith@company.com</EndUserId>
    </EndUser>
    <CableModem>
      <CmMacAddr>10:10:10:10:10:11</CmMacAddr>
      <CmIpAddr>30.0.0.3</CmIpAddr>
      <UpServerIndex>0</UpServerIndex>
      <UpChannelIndex>0</UpChannelIndex>
      <DownServerIndex>0</DownServerIndex>
      <DownChannelIndex>0</DownChannelIndex>
      <DocsisVersion>0</DocsisVersion>
      <MaxFlows>0</MaxFlows>
      <MaxClassifiers>0</MaxClassifiers>
      <Cpe>
        <CpeMacAddr>00:00:00:00:00:00</CpeMacAddr>
        <CpeIpAddr>10.0.0.3</CpeIpAddr>
      </Cpe>
      <Cpe>
        <CpeMacAddr>00:00:00:00:00:00</CpeMacAddr>
        <CpeIpAddr>20.0.0.3</CpeIpAddr>
      </Cpe>
    </CableModem>
    <TierRef>
      <Name>Premium</Name>
    </TierRef>
  </Account>
</ConfigurationData>

```

```

    </Account>
</ConfigurationData>

```

Tiers

A tier represents a class of service or a logical grouping of subscriber accounts.

Add Tier

The following examples show the request and response that are defined in the XSDs for the AddTier tag.

Request

This request follows the AddTier tag defined in the XSDs.

The following example creates two tiers:

- Tier1: Upstream bandwidth limit: 1 Mbps
- Tier2: Upstream bandwidth limit: 2 Mbps

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
<AddTier>
  <Tier>
    <Name>Tier1</Name>
    <Description/>
    <ResourceLimit>
      <ResourceType>0</ResourceType>
      <Direction>1</Direction>
      <LimitValue>1000000</LimitValue>
    </ResourceLimit>
  </Tier>
  <Tier>
    <Name>Tier2</Name>
    <Description/>
    <ResourceLimit>
      <ResourceType>0</ResourceType>
      <Direction>1</Direction>
      <LimitValue>2000000</LimitValue>
    </ResourceLimit>
  </Tier>
</AddTier>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">

```

```

    <Success count="2">Successfully imported 2 tiers.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Update Tier

The following examples show the request and response that are defined in the XSDs for the UpdateTier tag.

Request

This request follows the UpdateTier tag defined in the XSDs.

The following example updates the tier named Tier1 to change the upstream bandwidth limit to 555000 bps.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <UpdateTier>
    <Tier>
      <Name>Tier1</Name>
      <ResourceLimit>
        <ResourceType>0</ResourceType>
        <Direction>1</Direction>
        <LimitValue>555000</LimitValue>
      </ResourceLimit>
    </Tier>
  </UpdateTier>
</XmlInterfaceRequest>

```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="1">Successfully updated 1 tier.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>

```

Delete Tier

The following examples show the request and response that are defined in the XSDs for the DeleteTier tag.

Request

This request follows the DeleteTier tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeleteTier>
    <Name>Tier1</Name>
    <Name>Tier2</Name>
  </DeleteTier>
</XmlInterfaceRequest>
```

Response

The response to this request follows the generic Response tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="2">Deleted 2 Tiers.</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Query Tier

The following examples show the request and response that are defined in the XSDs for the QueryTier tag.

Request

This request follows the QueryTier tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryTier>
    <Name>Tier Gold</Name>
    <Name>Tier Bronze</Name>
  </QueryTier>
</XmlInterfaceRequest>
```

Response

The response to this request follows the ConfigurationData -> Tier tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="3.0.0">
  <Tier>
    <Name>Tier Gold</Name>
    <Description/>
    <ResourceLimit>
      <ResourceType>0</ResourceType>
      <Direction>1</Direction>
      <LimitValue>9000000</LimitValue>
    </ResourceLimit>
  </Tier>
```

```
<Tier>
  <Name>Tier Bronze</Name>
  <Description/>
  <ResourceLimit>
    <ResourceType>0</ResourceType>
    <Direction>1</Direction>
    <LimitValue>1000000</LimitValue>
  </ResourceLimit>
</Tier>
</ConfigurationData>
```

Chapter 6

Operational Measurements Interface Overview

Topics:

- *OM Statistics Scheduled Task.....62*
- *OM Statistics Requests.....62*
- *OM Statistics Response Format.....64*
- *Timezones and Start/End Times.....65*
- *Counter Reset and Failover.....66*
- *Comparisons Between the CMP GUI and OM Statistics.....67*

The Operational Measurements (OM) XML interface is used to retrieve operational counters from the system. This chapter describes the interface and how it should be used.

OM Statistics Scheduled Task

The OM interface requires that the “OM Statistics” scheduled task be running on the Configuration Management Platform (CMP). This task performs the function of collecting the operational counters from the MPEs in the network and recording them in the CMP database; the data is then available to be queried via the OM XML interface. This task can be configured by the user to poll at intervals between 5 minutes and 24 hours, with a default value of 15 minutes, and keeps the data available for query for 1 to 30 days, with the default value being 7 days. The recommended settings for this task varies with each deployment, dependant on the volume of data being collected.

When OM requests are made, the data for the response is taken from the information that has been collected by this task. Without the data that is collected through this scheduled task, there would be no data available for OM queries.

Most values returned as part of the response are presented (by default) as deltas, representing the positive change between the start time and end time. In order to calculate a delta there must be a minimum of two recorded values available, so the OM Statistics task must have been run at least twice in order to provide any data through the OM XML interface.

OM Statistics Requests

OM statistic requests consist of the following:

- Attributes and Child Tag
- Recorded Timestamp and Request Time Range

Attributes and Child Tags

The following attributes can be used with any of the OM Statistics requests as part of the “QueryOMStats” tag:

- DeltaCount — The values for statistics which are cumulative in nature (for example, monotonically increasing counters) are returned by default as a delta value. For these cumulative statistics, the delta value returned represents the positive change in that value since the last recorded time period. By setting the DeltaCount attribute to “false” the user can request that all statistics be returned as absolute values instead, that is the total values since the beginning of time.

Some statistics are always returned as absolute values. These non-cumulative statistics have values that can increase or decrease over time, for example Upstream and Downstream Bandwidth or active Session Counts. These statistics are explicitly called out in this document as always returning “absolute” values.

The following is an example of the DeltaCount attribute:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats DeltaCount="false">
  <StartTime>2006-10-12T11:15:00Z</StartTime>
```

```
<EndTime>2006-10-12T11:25:00Z</EndTime>
<NetworkElementStats></NetworkElementStats>
</QueryOmStats>
```

- **AggregateTimeSamples** — The `AggregateTimeSamples` attribute allows the user to request that all sample buckets in the response be aggregated into a single sample bucket. This allows users to request data for a time period and see a summary of all recorded data for that specific time period, rather than all the individual samples that make up the requested time period.

The following is an example of the `AggregateTimeSamples` attribute:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats AggregateTimeSamples="true">
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <NetworkElementStats/>
</QueryOmStats>
```

The following child tags are available for OM XML Commands:

- **StartTime/EndTime** — These tags define the time range over which you want statistical data returned. All statistic data recorded by the OM Statistics task over that requested time range is returned. Also, the `EndTime` tag is an optional tag. If no `EndTime` tag is specified, then the time range extends from the `StartTime` and continues until the current time. This provides a means to capture the most up-to-date data.

The following is an example of the `StartTime/EndTime` child tags:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <PolicyServerStats/>
  <IsComplete>true</IsComplete>
</QueryOmStats>
```

- **PolicyServer** — This optional tag, specified as a child tag of some individual statistic groups, allows the user to request statistics specific to an individual or a set of MPEs. The default behavior displays statistics for all MPEs.

The following is an example of the `PolicyServer` child tag:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <PolicyServerStats>
    <PolicyServer>Atlanta105</PolicyServer>
    <IsComplete>true</IsComplete>
  </PolicyServerStats>
</QueryOmStats>
```

Recorded Timestamp and Request Time Range

The OM Statistics task runs on a scheduled interval. The statistics, however, are time stamped as they are received from the MPEs to increase accuracy. For example, while the OM Statistics task may start at 12:00:00, the statistics recorded for MPE1 may not be time stamped until 12:00:01, and MPE2 at 12:00:02, and so on.

This detail impacts the way the captured data should be requested through the XML interface so as to ensure all the desired data is returned as part of the response. To ensure correct results, if the intent is to retrieve data for a full 24 hour period, then the start and end times should extend beyond the intended 24 hour period to ensure that any delayed results returned for an MPE are included in the response.

For example, to retrieve statistics recorded for the entire day of March 10, the following request time range is recommended (note that the EndTime is 15 minutes past midnight):

```
<StartTime>2008-03-10T00:00:00Z</StartTime>
<EndTime>2008-03-11T00:15:00Z</EndTime>
```

OM Statistics Response Format

The XSD defines the schema for responses to OM XML requests. Results are returned in “Sample” buckets representing a range of time. If Persistent Interval Statistics are enabled (Stats.Interval.Enabled), this range is determined by the **Stats Collection Period**, found under **Global Configuration Settings > Stats Settings** on the CMP, but also includes any “Manual” runs of the task initiated by the user.

If Stats.Interval.Enabled is set to false (meaning that Persistent Interval Statistics are not enabled), then the CMP collects absolute counter values only; this range is determined by the interval settings of the OM Statistics scheduled task and includes any “Manual” runs of the task initiated by the user.

Responses do not return data sets that exceed a maximum size. In this case, a user is directed to reduce the scope of their query to accommodate for the size limitation of the XML response.

Absolute versus Delta Values

For all cumulative statistics (for example, monotonically increasing counters), the returned values are presented as deltas, by default. These deltas represent the positive change in value between the start and end times. This behavior can optionally be changed to return absolute values for all statistics (see [OM Statistics Requests](#)).

Non-cumulative statistics, those statistics whose value can either increase or decrease between intervals, are returned as absolute values. This prevents negative value responses as a result of a delta calculation. For example, in “NetworkElementStats” the Upstream Bandwidth, Downstream Bandwidth, and “active” Session Counts are returned as absolute values. These absolute statistics are explicitly called out in this document as always returning “absolute” values. All absolute data points are taken as the value at the End Time of the reported Sample.

Empty Data Set

Requests that do not result in any statistics contain the following message:

Not enough statistical data available to fulfill request.

If you receive this message, check the request parameters or select a different time range, as the message indicates that for the parameters submitted as part of the request there is not enough recorded data to respond.

This response can occur for a number of reasons. If you see the following in the response:

```
<IsComplete>false</IsComplete>
```

then the MPE or MRA has detected errors when collecting the stats, usually because a complete interval was not available.

The OM Statistics Task must have been run at least two times within the specified time range (StartTime and EndTime). The task itself runs on a schedule but the data is recorded dynamically as it is received from the MPEs so this should be taken into account when requesting a particular range of time. Tekelec recommends that users start with a larger time window for testing and narrow that down, adjusting the scheduled task as needed, until the desired data is returned. This message could also occur if recorded data is unavailable for a specific parameter, such as a certain NetworkElement or MPE name.

Timezones and Start/End Times

OM responses include "Sample" buckets, each with a specified start and end time. These values are returned in UTC (Coordinated Universal Time) format, which is the international time standard. The CMP is unaware of the timezone of the originating request and is therefore unable to return a "localized" timezone to the user.

The XSDs define request/responses with the following:

```
<!-- Time Range -->
<xsd:element name="StartTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
<xsd:element name="EndTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
```

The type is defined as "**xsd:dateTime**". This is a UTC format. This data type describes instances identified by the combination of a date and a time.

Its lexical space is the extended format:

```
[ - ]CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]
```

The time zone may be specified as Z (UTC) or (+|-)hh:mm. Time zones that are not specified are considered localized to the MPE Manager.

The following are examples of valid values for xsd:dateTime:

Example 1:

2007-03-26T21:32:52Z - UTC format. This is the default response format for all timestamps. It is also the recommend format for user requests.

Example 2:

2007-03-26T21:32:52+02:00 - Localized time with 2 hour offset to UTC time.

Example 3:

2007-03-26T19:32:52 - Localized time. This will be interpreted as localized to the MPE Manager.

Example 4:

2007-03-26T19:32:52+00:00 - Localized time with no offset.

Note: All date and time numbers are two digits. For example, 3:00am, April 5th is written as "2008-04-05T03:00:00" and not "2008-4-5T3:00:00".

A request to retrieve statistics can be made either in UTC or as a time that is localized to the CMP. Tekelec recommends that users make their requests in UTC format to remain consistent with the UTC output. UTC is also recommended, as end users may not be aware they are making requests from a different timezone than the CMP and therefore may not be accounting for that difference when analyzing results. If the MPEs, CMP, and end user are all in different timezones then the user would be attempting to correlate values and draw conclusions from the reported statistics across all of those timezones.

Counter Reset and Failover

MPEs and MRAs write historical data to a distributed comcol database and save it for 24 hours.

Note: CMP stores cluster-level interval stats to its own database. Data is available for external OSS system to retrieve after the end of the next interval period.

There are two cases where statistics can be reset:

- The user can click the "Reset All Counters" button on the Policy Server "Reports" page of the CMP Graphical User Interface (GUI). This button is typically used to reset counters as a baseline from which to run tests (refer to *CMP User Guide* for detailed description of this page.).
- The user can enable the Interval mode; when configured, numeric values are reset at regular intervals controlled by the Stats Collection Period. When in Interval mode, a reset will occur on the hour and then every 5, 10, 15, 20, 30 or 60 minutes afterwards depending on the value of the Stats Collection Period. (When interval periods are changed, the database is purged.) See the *CMP User Guide* for details.

In either case, all memory counters are reset to 0. This means that the delta values calculated for the time period during which the reset occurred are not accurate.

For example, if at the StartTime the SuccessSessionCount value was 100, and a reset occurred resetting the count to 0, then the next reported delta would be -100. If the SuccessSessionCount started at 100, and during the sample time period, five successful sessions completed, then a reset occurred after which 10 more sessions completed successfully, then the resulting delta for that time period would be -90. This example shows that some information is lost; the response will indicate this with the `IsComplete` flag set to `false`. In either case, the value could not be seen as an accurate representation of activity for that time and should be ignored. For this reason, negative delta values are always returned as "0".

The CMP will get incomplete interval stats from MPE/MRA in the following conditions:

- The active MPE/MRA blade switches over one or more times within an interval period.
- MPE/MRA has initially started up.
- Under high-load conditions, the MPE/MRA cannot store data to the database at the end of an interval cycle (within a margin of a few seconds).

Comparisons Between the CMP GUI and OM Statistics

Relationships between the OM XML responses and the CMP Reports GUI can be drawn and used for comparison.

The OM XML interface persists statistic values over time. This tool is intended for historical analysis of statistics and can be used to track usage. The interface allows users to request data over a user-defined time range and returns data for that period. The default behavior for this interface is to return data as delta values. The delta is calculated as the positive change in value between the start and end times returned in "Sample" buckets. Certain non-cumulative statistics are always reported as absolutes and those individual statistics are explicitly documented. This behavior can also be changed to return absolute values for all statistics; even those which are cumulative in nature (see [OM Statistics Requests](#)).

The "Reports" page in the Policy Server section of the CMP GUI displays statistics in real time (Refer to the *CMP User's Guide* for details). The intent of this page is for monitoring current statistics. The page dynamically updates the displayed statistics every 10 seconds. These statistics are categorized by protocol. The default behavior of this page displays statistics as absolute values. There is a "Show Deltas" button at the top of the page; this button can be used to switch the display to calculated delta values. The delta value shown is the difference between the current value and the last refresh of the page (approximately 10 seconds).

The top section on any statistics page, contained within the CMP GUI, usually correlates with the "Message Processing Stats" of the OM XML interface. The individual statistics for each Network Element can be found in the section at the bottom of the reporting page. These statistics correlate with the "Network Element Stats" of the OM XML interface.

Labels for specific statistics may differ between the CMP GUI and the OM XML interface responses. The specific types of statistics displayed may also differ by protocol. While the CMP GUI can use labels and statistics specific to a certain protocol, the OM XML interface requests must use generic language across protocols in order to remain consistent with published XSD definitions.

For deployments that collect statistics across multiple protocols, for example PCMM and DQoS, the CMP GUI displays statistics for each protocol individually. However, in the XML Interface, the statistics for available protocols are summarized in the response. For example, session count statistics for PCMM and DQoS would be added together to display with each network element. Statistics for individual protocols are not be displayed separately.

Additional details for the GUI's Reports/Statistics section of the CMP GUI can be found in the *CMP User's Guide*.

Operational Measurement Requests

Topics:

- *Operational Measurements Requests Overview.....69*
- *Topology Update Statistics.....70*
- *Subscriber Update Statistics.....71*
- *Policy Server Statistics.....72*
- *MPE Database Activity Statistics.....73*
- *Message Processing Statistics.....74*
- *Network Element Statistics.....75*
- *Reserve Commit Statistics.....79*
- *Gate Statistics.....80*
- *Latency Statistics.....81*
- *RADIUS-S Statistics.....90*
- *RADIUS-S Network Element Statistics.....93*
- *Connected Network Elements Statistics.....96*
- *Mgpi Statistics.....97*
- *Diameter Operational Measurements Requests.....98*
- *KPI Statistics.....110*
- *KPI Interval Statistics.....112*
- *PCMM CMTS Statistics.....113*
- *PCMM Network Element Statistics.....114*
- *DQos CMTS Statistics.....115*
- *DQos Network Element Statistics.....116*
- *Connection Error Statistics.....118*
- *Protocol Error Statistics.....119*

This chapter defines the various Operational Measurements (OM) groups and the individual OM statistics that comprise them. For several of the more generic statistics, the instrumentation on the Multimedia Policy Engine (MPE) and the Multi-Protocol Routing Agent (MRA) may differ by protocol, and therefore specific statistics may increment differently across those protocols.

Operational Measurements Requests Overview

The OM interface consists of the following requests, grouped by category:

- OSSI XML interface statistics
 - Topology Update Statistics — retrieves statistics on all updates made through the Topology Interface.
 - Subscriber Update Statistics — retrieves statistics on all updates made through the Subscriber Interface.
 - Policy Server Statistics — retrieves statistics on each policy server in the system, including the associated subscribers and network elements.
- Protocol-specific statistics
 - Message Processing Statistics — retrieves statistics on session events.
 - Network Element Statistics — retrieves statistics on session events for specific network elements and interfaces.
 - Reserve Commit Statistics — retrieves statistics on reserve and commit requests specific to the SPC DQoS protocol.
 - Gate Statistics — retrieves statistics on gate set messages, gate status messages, and gate delete messages specific to PCMM, DQoS and SPC DQoS protocols.
 - RADIUS-S Statistics — retrieves statistics on accounting and change-of-authorization messages specific to the RADIUS-S protocol.
 - Connected Network Element Statistics — retrieves statistics, reporting the number of connected network elements on a per-MPE basis. Connected network elements are those that have a connection (for example, COPS) established to the MPE. This statistic is specific to wireline customers.
 - Diameter Statistics — retrieves statistics on Diameter Application Functions, Charging Functions, and Policy Charging Enforcement Functions.
 - Timeout Statistics — retrieves timeout statistics not received in a predefined amount of time, that are tracked per network element, MPE, and MRA.
 - Latency Statistics — retrieves latency statistics information for incoming and outgoing messages tracked per network element, MPE, and MRA.
 - Event Trigger Statistics — retrieves related event trigger statistics used for tracking activity.
 - Session Cleanup Statistics — retrieves the number of session cleanup statistics where sessions are aged-out due to inactivity. The reauthorization is submitted and if it is unresponsive or an error occurs indicating that the session is inactive, then it is removed from the system.
 - Traffic Profile Statistics — retrieves traffic profile statistics which counters the number of times a PCRF attempts to install, remove, or fails.
 - Quota Profile Statistics — retrieves quota profile statistics to counter the number of times the configured threshold reaches a maxed out limit.

Topology Update Statistics

The following examples show the request and response that are defined in the XSDs for the TopologyUpdateStats tag.

Request

This request follows the **QueryOmStats > TopologyUpdateStats** tag defined in the XSDs.

The following is an example of a request for a topology update:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <TopologyUpdateStats></TopologyUpdateStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > TopologyUpdateStats** tag defined in the XSDs.

Individual statistics are defined as follows:

- **TopologyUpdateCount:** The number of topology changes made in the Configuration Management Platform (CMP) that resulted in changes to an MPE. Topology data includes Network Elements, Paths, Interfaces, and Links. Changes to this data can occur through either the CMP Graphical User Interface (GUI) or the XML interface. The counter increments for each MPE that updates as a result of the change. For example, when a user updates a Network Element, the topology update counter increments for each MPE that the Network Element is associated with.

Batch changes are treated as a single update to an MPE. An OSSI XML interface update may be made across multiple elements but all of those changes are pushed at one time to the MPE. This is treated as a single topology change with regard to the counter.

- **TopologyUpdateFailCount:** The number of topology changes made in the CMP that fail to update an associated MPE. For example, if an MPE is offline while a Network Element is changed, the fail counter will increment once for that failed MPE update.

The following is an example of the TopologyUpdateStats tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <TopologyUpdateStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <TopologyUpdateCount>942</TopologyUpdateCount>
      <TopologyUpdateFailCount>6</TopologyUpdateFailCount>
    </Sample>
  </TopologyUpdateStats>
</Statistics>
```

Subscriber Update Statistics

The following examples show the request and response that are defined in the XSDs for the `SubscriberUpdateStats` tag.

Request

This request follows the `QueryOmStats > SubscriberUpdateStats` tag defined in the XSDs.

For example:

The following is an example of a request for subscriber update statistics:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <SubscriberUpdateStats></SubscriberUpdateStats>
</QueryOmStats>
```

Response

The response to this request follows the `Statistics > SubscriberUpdateStats` tag defined in the XSDs.

Individual statistics are defined as follows:

- **SubscriberUpdateCount:** The number of subscriber changes made in the CMP that resulted in changes to an MPE. Subscriber data includes accounts and tiers. Changes to this data can occur through either the CMP GUI or the XML Interface. The counter is incremented for each MPE that updates as a result of the change. For example, when a user updates an account, the subscriber update counter increments for each MPE that the account is associated with.

Batch changes are treated as a single update to an MPE. An OSSI XML interface update may be made across multiple accounts but all of those changes are pushed at one time to the MPE. This is treated as a single subscriber change with regard to the counter.

- **SubscriberUpdateFailCount:** The number of subscriber changes made in the CMP that fail to update an associated MPE. For example, if an MPE is offline while an account is changed, the fail counter will increment once for that failed MPE update.

The following is an example of the `SubscriberUpdateStats` tag:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Statistics>
  <SubscriberUpdateStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <SubscriberUpdateCount>942</SubscriberUpdateCount>
      <SubscriberUpdateFailCount>6</SubscriberUpdateFailCount>
    </Sample>
  </SubscriberUpdateStats>
</Statistics>
```

Policy Server Statistics

The following examples show the request and response that are defined in the XSDs for the PolicyServerStats tag.

Request

This request follows the **QueryOmStats** > **PolicyServerStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <PolicyServerStats>
    <PolicyServer>Atlanta105</PolicyServer>
  </PolicyServerStats>
</QueryOmStats>
```

The following is an example of a request for all policy servers in the system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <PolicyServerStats></PolicyServerStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics** > **PolicyServerStats** tag defined in the XSDs.

Individual statistics are defined as follows:

- **TotalNetworkElementCount:** The total (absolute) number of network elements associated with that MPE. The absolute value is the value taken at the EndTime of the reported sample.
- **TotalSubscriberCount:** The total (absolute) number of accounts associated with that MPE.

The following is an example of the PolicyServerStats tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PolicyServerStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>true</IsComplete>
      <TotalNetworkElementCount>52</TotalNetworkElementCount>
      <TotalSubscriberCount>1000000</TotalSubscriberCount>
    </Sample>
  </PolicyServerStats>
</Statistics>
```

MPE Database Activity Statistics

The following examples show the request and response that are defined in the XSDs for the **QueryOmStats** tag.

Request

This request follows the **QueryOmStats** tag defined in the XSDs.

This is an example of XML for querying multiple network elements by name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryOmStats AggregateTimeSamples="false" DeltaCount="false">
    <StartTime>2010-06-15T00:01:00</StartTime>
    <DbStats/>
  </QueryOmStats>
</XmlInterfaceRequest>
```

Response

The response to this request follows the **QueryOmStats** tag defined in the XSDs.

```
<?xml version='1.0' ?>
<Statistics>
  <DbStats>
    <Sample>
      <StartTime>2011-08-25T07:45:00Z</StartTime>
      <EndTime>2011-08-25T08:00:00Z</EndTime>
      <PolicyServer>10.60.24.47</PolicyServer>
      <DatabaseName>DbStatus</DatabaseName>
      <SessionReadCount>0</SessionReadCount>
      <SessionWriteCount>0</SessionWriteCount>
      <ResetTime>N/A</ResetTime>
      <CollectTime>Thu Aug 25 16:00:00 CST 2011</CollectTime>
    </Sample>
    <Sample>
      <StartTime>2011-08-25T08:00:00Z</StartTime>
      <EndTime>2011-08-25T08:15:00Z</EndTime>
      <PolicyServer>10.60.24.47</PolicyServer>
      <DatabaseName>DbStatus</DatabaseName>
      <SessionReadCount>0</SessionReadCount>
      <SessionWriteCount>0</SessionWriteCount>
      <ResetTime>N/A</ResetTime>
      <CollectTime>Thu Aug 25 16:15:00 CST 2011</CollectTime>
    </Sample>
    <Sample>
      <StartTime>2011-08-25T08:15:00Z</StartTime>
      <EndTime>2011-08-25T08:30:00Z</EndTime>
      <PolicyServer>10.60.24.47</PolicyServer>
      <DatabaseName>DbStatus</DatabaseName>
      <SessionReadCount>1</SessionReadCount>
      <SessionWriteCount>3</SessionWriteCount>
      <ResetTime>N/A</ResetTime>
      <CollectTime>Thu Aug 25 16:30:00 CST 2011</CollectTime>
    </Sample>
  </DbStats>
</Statistics>
```

```

        <StartTime>2011-08-25T08:30:00Z</StartTime>
        <EndTime>2011-08-25T08:45:00Z</EndTime>
        <PolicyServer>10.60.24.47</PolicyServer>
        <DatabaseName>DbStatus</DatabaseName>
        <SessionReadCount>7</SessionReadCount>
        <SessionWriteCount>8</SessionWriteCount>
        <ResetTime>N/A</ResetTime>
        <CollectTime>Thu Aug 25 16:45:00 CST 2011</CollectTime>
    </Sample>
</DbStats>
</Statistics>

```

Message Processing Statistics

The following examples show the request and response that are defined in the XSDs for the MessageProcessingStats tag.

Request

This request follows the **QueryOmStats** > **MessageProcessingStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <MessageProcessingStats/>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics** > **MessageProcessingStats** tag defined in the XSDs.

Individual statistics are defined as follows:

- **SessionCount:** Session requests received.
- **SessionSuccessCount:** Session requests successfully created.
- **SessionFailCount:** Session request failures. For example, this could be due to a policy denying a request.
- **SessionProtocolFailCount:** Number of session failures due to an invalid message or parameter. This count is incremented whenever the MPE determines that an incoming message from the AM has an invalid message and has to be dropped by the MPE.
- **SessionPolicyFailCount:** Number of session requests that trigger a policy. This count is maintained in the MPE, one per policy. When the condition of a policy triggers, the count for that policy is incremented. The value displayed is a total trigger count. That is, the sum of this value for all the policies. For example, if the definition of a policy is defined as, "when the device usage is greater than 80% of capacity, reject message". When the MPE executes this policy in response to a request, if the policy triggers (i.e. if the MPE determines that the device usage is > 80%), the trigger count for that policy is incremented.

The following is an example of the MessageProcessingStats tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <MessageProcessingStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>>true</IsComplete>
      <SessionCount>100</SessionCount>
      <SessionSuccessCount>97</SessionSuccessCount>
      <SessionFailCount>3</SessionFailCount>
      <SessionProtocolFailCount>1</SessionProtocolFailCount>
      <SessionPolicyFailCount>2</SessionPolicyFailCount>
    </Sample>
  </MessageProcessingStats>
</Statistics>
```

Network Element Statistics

The following examples show the request and response that are defined in the XSDs for the NetworkElementStats tag.

Request

This request follows the **QueryOmStats** > **NetworkElementStats** tag defined in the XSDs.

The following is an example of a request for a single network element using the Name parameter:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <NetworkElementStats>
    <Name>Router1</Name>
  </NetworkElementStats>
</QueryOmStats>
```

The following is an example of a request for multiple network elements using the Name(s) and NeId(s) parameters. This example returns statistics for three different network elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <NetworkElementStats>
    <Name>Router1</Name>
    <Name>Router2</Name>
    <Name>Router3</Name>
  </NetworkElementStats>
</QueryOmStats>
```

The following is an example of a request for all network elements in the system. This example returns statistics for each network element:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <NetworkElementStats></NetworkElementStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > NetworkElementStats** tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Sample groups are ordered by policy server, network element, available interfaces, and then by time.

This OM Group, by default, contains a combination of delta and absolute values. Individual statistics returned as absolutes are described as follows:

- **Name:** Unique name identifying the Network Element for the following statistics.
 - **NeId:** Optional identifier field for the Network Element.
 - **InterfaceName:** Identifying field for the Network Element's Interface.
 - **SessionCount:** Current active sessions for that Network Element or Interface. This is a non-cumulative value and is displayed as an absolute.
 - **SessionSuccessCount:** Successful sessions.
 - **SessionFailCount:** Session failures.
 - **AbnormalDisconnectCount:** Number of network elements that have disconnected from the MPE abnormally. For example, due to a break in the network.
 - **Capacity:** The currently defined maximum capacity for this Network Element or Interface. This is a static absolute value defined in the CMP for that object.
 - **BandwidthUpstream:** This is the current reserved upstream bandwidth allocated for this Network Element or Interface. This is a non-cumulative value displayed as an absolute.
 - **BandwidthDownstream:** This is the current reserved downstream bandwidth allocated for this Network Element or Interface. This is a non-cumulative value displayed as an absolute.

The following is an example XML response to a multiple network element request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <NetworkElementStats>
    <Sample>
      <StartTime>2006-10-12T11:18:30Z</StartTime>
      <EndTime>2006-10-12T11:19:20Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>Router1</Name>
      <NeId>12345</NeId>
      <SessionCount>3</SessionCount>
      <SessionSuccessCount>3</SessionSuccessCount>
      <SessionFailCount>0</SessionFailCount>
      <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
      <Capacity>50000000</Capacity>
      <BandwidthUpstream>3000000</BandwidthUpstream>
```

```

    <BandwidthDownstream>100000</BandwidthDownstream>
  </Interface>
  <InterfaceName>If1</InterfaceName>
  <SessionCount>3</SessionCount>
  <SessionSuccessCount>3</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
</Interface>
</Sample>
<Sample>
  <StartTime>2006-10-12T11:19:20Z</StartTime>
  <EndTime>2006-10-12T11:20:10Z</EndTime>
  <PolicyServer>Atlanta105</PolicyServer>
  <IsComplete>true</IsComplete>
  <Name>Router1</Name>
  <NeId>12345</NeId>
  <SessionCount>0</SessionCount>
  <SessionSuccessCount>0</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
  <Interface>
    <InterfaceName>If1</InterfaceName>
    <SessionCount>0</SessionCount>
    <SessionSuccessCount>0</SessionSuccessCount>
    <SessionFailCount>0</SessionFailCount>
    <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    <Capacity>50000000</Capacity>
    <BandwidthUpstream>3000000</BandwidthUpstream>
    <BandwidthDownstream>100000</BandwidthDownstream>
  </Interface>
</Sample>
<Sample>
  <StartTime>2006-10-12T11:20:10Z</StartTime>
  <EndTime>2006-10-12T11:21:00Z</EndTime>
  <PolicyServer>Atlanta105</PolicyServer>
  <IsComplete>true</IsComplete>
  <Name>Router1</Name>
  <NeId>12345</NeId>
  <SessionCount>9</SessionCount>
  <SessionSuccessCount>9</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
  <Interface>
    <InterfaceName>If1</InterfaceName>
    <SessionCount>9</SessionCount>
    <SessionSuccessCount>9</SessionSuccessCount>
    <SessionFailCount>0</SessionFailCount>
    <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    <Capacity>50000000</Capacity>
    <BandwidthUpstream>3000000</BandwidthUpstream>
    <BandwidthDownstream>100000</BandwidthDownstream>
  </Interface>
</Sample>
<Sample>
  <StartTime>2006-10-12T11:18:30Z</StartTime>

```

```

<EndTime>2006-10-12T11:19:20Z</EndTime>
<PolicyServer>Atlanta105</PolicyServer>
<IsComplete>>true</IsComplete>
<Name>Router 2</Name>
<NeId>12341</NeId>
<SessionCount>11</SessionCount>
<SessionSuccessCount>11</SessionSuccessCount>
<SessionFailCount>0</SessionFailCount>
<AbnormalDisconnectCount>0</AbnormalDisconnectCount>
<Capacity>50000000</Capacity>
<BandwidthUpstream>3000000</BandwidthUpstream>
<BandwidthDownstream>100000</BandwidthDownstream>
<Interface>
  <InterfaceName>If2</InterfaceName>
  <SessionCount>11</SessionCount>
  <SessionSuccessCount>11</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
</Interface>
</Sample>
<Sample>
  <StartTime>2006-10-12T11:19:20Z</StartTime>
  <EndTime>2006-10-12T11:20:10Z</EndTime>
  <PolicyServer>Atlanta105</PolicyServer>
  <IsComplete>>true</IsComplete>
  <Name>Router 2</Name>
  <NeId>12341</NeId>
  <SessionCount>2</SessionCount>
  <SessionSuccessCount>2</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
  <Interface>
    <InterfaceName>If2</InterfaceName>
    <SessionCount>2</SessionCount>
    <SessionSuccessCount>2</SessionSuccessCount>
    <SessionFailCount>0</SessionFailCount>
    <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    <Capacity>50000000</Capacity>
    <BandwidthUpstream>3000000</BandwidthUpstream>
    <BandwidthDownstream>100000</BandwidthDownstream>
  </Interface>
</Sample>
<Sample>
  <StartTime>2006-10-12T11:20:10Z</StartTime>
  <EndTime>2006-10-12T11:21:00Z</EndTime>
  <PolicyServer>Atlanta105</PolicyServer>
  <IsComplete>>true</IsComplete>
  <Name>Router 2</Name>
  <NeId>12341</NeId>
  <SessionCount>9</SessionCount>
  <SessionSuccessCount>9</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>50000000</Capacity>
  <BandwidthUpstream>3000000</BandwidthUpstream>
  <BandwidthDownstream>100000</BandwidthDownstream>
  <Interface>
    <InterfaceName>If2</InterfaceName>

```

```

        <SessionCount>9</SessionCount>
        <SessionSuccessCount>9</SessionSuccessCount>
        <SessionFailCount>0</SessionFailCount>
        <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
        <Capacity>50000000</Capacity>
        <BandwidthUpstream>3000000</BandwidthUpstream>
        <BandwidthDownstream>100000</BandwidthDownstream>
    </Interface>
</Sample>
</NetworkElementStats>
</Statistics>

```

Reserve Commit Statistics

The following examples show the request and response that are defined in the XSDs for the **ReserveCommitStats** tag.

Request

This request follows the **QueryOmStats > ReserveCommitStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <ReserveCommitStats></ReserveCommitStats>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics > ReserveCommitStats** tag defined in the XSDs.

Individual statistics are defined as follows:

- **ReserveCount:** Number of GateSet Reserve messages received.
- **ReserveSuccessCount:** Number of GateSet Reserve messages acknowledged.
- **ReserveFailCount:** Number of GateSet Reserve messages failed.
- **ReserveErrorCount:** Number of GateSet Reserve messages that were errors.
- **CommitCount:** Number of GateSet Commit messages received.
- **CommitSuccessCount:** Number of GateSet Commit messages acknowledged.
- **CommitFailCount:** Number of GateSet Commit messages failed.
- **CommitErrorCount:** Number of GateSet Commit messages that were errors.
- **ReserveCommitCount:** Number of GateSet Reserve+Commit messages received.
- **ReserveCommitSuccessCount:** Number of GateSet Reserve+Commit messages acknowledged.
- **ReserveCommitFailCount:** Number of GateSet Reserve+Commit messages failed.
- **ReserveCommitErrorCount:** Number of GateSet Reserve+Commit messages that were errors.

The following is an example XML response to a **ReserveCommitStats** tag request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <ReserveCommitStats>
    <Sample>
      <StartTime>2007-10-26T14:30:22Z</StartTime>
      <EndTime>2007-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>true</IsComplete>
      <ReserveCount>100</ReserveCount>
      <ReserveSuccessCount>99</ReserveSuccessCount>
      <ReserveFailCount>1</ReserveFailCount>
      <ReserveErrorCount>0</ReserveErrorCount>
      <CommitCount>99</CommitCount>
      <CommitSuccessCount>99</CommitSuccessCount>
      <CommitFailCount>0</CommitFailCount>
      <CommitErrorCount>0</CommitErrorCount>
      <ReserveCommitCount>99</ReserveCommitCount>
      <ReserveCommitSuccessCount>99</ReserveCommitSuccessCount>
      <ReserveCommitFailCount>0</ReserveCommitFailCount>
      <ReserveCommitErrorCount>0</ReserveCommitErrorCount>
    </Sample>
  </ReserveCommitStats>
</Statistics>
```

Gate Statistics

The following examples show the request and response that are defined in the XSDs for the **GateStats** tag.

Request

This request follows the **QueryOmStats** > **GateStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <GateStats></GateStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics** > **GateStats** tag defined in the XSDs.

- **GateSetCount** — Number of Gate Set messages processed.
- **GateSetSuccessCount** — Number of Gate Set Success messages processed.
- **GateSetErrorCount** — Number of Gate Set Error messages processed.
- **GateStatusCount** — Number of Gate Status messages processed.
- **GateStatusSuccessCount** — Number of Gate Status Success messages processed.
- **GateStatusErrorCount** — Number of Gate Status Error messages processed.

- GateDeleteCount — Number of Gate Delete messages processed.
- GateDeleteSuccessCount — Number of Gate Delete Success messages processed.
- GateDeleteErrorCount — Number of Gate Delete Error messages processed.

The following is an example XML response to a GateStats tag request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <GateStats>
    <Sample>
      <StartTime>2007-10-26T14:30:22Z</StartTime>
      <EndTime>2007-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>true</IsComplete>
      <GateSetCount>100</GateSetCount>
      <GateSetSuccessCount>100</GateSetSuccessCount>
      <GateSetErrorCount>0</GateSetErrorCount>
      <GateStatusCount>100</GateStatusCount>
      <GateStatusSuccessCount>100</GateStatusSuccessCount>
      <GateStatusErrorCount>0</GateStatusErrorCount>
      <GateDeleteCount>100</GateDeleteCount>
      <GateDeleteSuccessAckCount>100</GateDeleteSuccessCount>
      <GateDeleteErrorCount>0</GateDeleteErrorCount>
    </Sample>
  </GateStats>
</Statistics>
```

Latency Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterAfLatencyStats tag.

Request

This request follows the **QueryOmStats > DiameterAfLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <DiameterAfLatencyStats>
      <PolicyServer>PolicyServer</PolicyServer>
    </DiameterAfLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>
```

Response

The response to this request follows the **Statistics > DiameterAfLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <DiameterAfLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <CurrentConnectionsCount>0</CurrentConnectionsCount>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
      <TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
      <TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
      <TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
      <TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
      <TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
      <TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
      <TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
      <TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
      <TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
      <TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
      <TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
      <TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
    </Sample>
  </DiameterAfLatencyStats>
</Statistics>
```

Request

This request follows the **QueryOmStats > DiameterAfPeerLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <DiameterAfPeerLatencyStats>
      <Name>ApplicationName</Name>
    </DiameterAfPeerLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>
```

Response

The response to this request follows the **Statistics > DiameterAfPeerfLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <DiameterAfPeerfLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>ApplicationName</Name>
      <NeId>NeId</NeId>
      <NetworkElementType>NetworkElementType</NetworkElementType>
      <NetworkElementSubType>NetworkElementSubType</NetworkElementSubType>
      <ConnectAddress>10.60.4.56</ConnectAddress>
      <ConnectPort>3868</ConnectPort>
      <ConnectType>TCP</ConnectType>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
      <TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
      <TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
      <TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
      <TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
      <TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
      <TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
      <TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
      <TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
      <TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
      <TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
      <TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
      <TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
      <TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
    </Sample>
  </DiameterAfPeerfLatencyStats>
</Statistics>
```

Request

This request follows the **QueryOmStats > PcmCmtsLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
```

```

    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmCmtsLatencyStats>
      <PolicyServer>PolicyServer</PolicyServer>
    </PcmmCmtsLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

The response to this request follows the **Statistics > PcmmCmtsLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PcmmCmtsLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <CurrentConnectionsCount>0</CurrentConnectionsCount>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
      <TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
      <TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
      <TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
      <TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
      <TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
      <TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
      <TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
      <TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
      <TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
      <TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
      <TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
      <TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
      <TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
    </Sample>
  </PcmmCmtsLatencyStats>
</Statistics>

```

Request

This request follows the **QueryOmStats > PcmmCmtsPeerLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmCmtsPeerLatencyStats>
      <Name>NetworkElementName</Name>
    </PcmmCmtsPeerLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>
```

Response

The response to this request follows the **Statistics > PcmmCmtsPeerLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PcmmCmtsPeerLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>NetworkElementName</Name>
      <NeId>NeId</NeId>
      <NetworkElementType>NetworkElementType</NetworkElementType>
      <NetworkElementSubType>NetworkElementSubType</NetworkElementSubType>
      <ConnectAddress>10.60.4.56</ConnectAddress>
      <ConnectPort>3868</ConnectPort>
      <ConnectType>TCP</ConnectType>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
      <TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
      <TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
      <TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
      <TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
      <TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
      <TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
      <TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
      <TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
      <TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
      <TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
      <TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
```

```

    <TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
    <TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
  </Sample>
</PcmmCmtsPeerLatencyStats>
</Statistics>

```

Request

This request follows the **QueryOmStats** > **PcmmAmLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmAmLatencyStats>
      <PolicyServer>PolicyServer</PolicyServer>
    </PcmmAmLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

The response to this request follows the **Statistics** > **PcmmAmLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PcmmAmLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <CurrentConnectionsCount>0</CurrentConnectionsCount>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
      <TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
      <TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
      <TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
      <TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
      <TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
      <TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
      <TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
    </Sample>
  </PcmmAmLatencyStats>
</Statistics>

```

```

    <TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
    <TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
    <TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
    <TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
    <TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
    <TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
  </Sample>
</PcmmAmLatencyStats>
</Statistics>

```

Request

This request follows the **QueryOmStats > PcmmAmPeerLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmAmPeerLatencyStats>
      <Name>ApplicationName</Name>
    </PcmmAmPeerLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

The response to this request follows the **Statistics > PcmmAmPeerLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```

<Statistics>
  <PcmmAmPeerLatencyStats>
    <Sample>
      <StartTime>2013-04-12T01:45:00Z</StartTime>
      <EndTime>2013-04-12T01:00:00Z</EndTime>
      <PolicyServer>policyserver</PolicyServer>
      <IsComplete>>true</IsComplete>
      <Name>ApplicationName</Name>
      <AmId>AmId</AmId>
      <ConnectAddress>10.60.56.60</ConnectAddress>
      <MaxTransactionInProcessingTime>863</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>178</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
      <TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
      <TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
      <TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
    </Sample>
  </PcmmAmPeerLatencyStats>
</Statistics>

```

```

<TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
<TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
<TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
<TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
<TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
<TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
<TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
<TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
<TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
<TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
<TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
<TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
<TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
</Sample>
</PcmmAmPeerLatencyStats>
</Statistics>

```

Request

This request follows the **QueryOmStats** > **PcmmDpsLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmDpsLatencyStats>
      <PolicyServer>PolicyServer</PolicyServer>
    </PcmmDpsLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

The response to this request follows the **Statistics** > **PcmmDpsLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PcmmDpsLatencyStats>
    <Sample>
      <StartTime>2001-12-31T12:00:00</StartTime>
      <EndTime>2001-12-31T12:15:00</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <CurrentConnectionsCount>0</CurrentConnectionsCount>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>

      <TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
      <TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
      <TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
      <TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
      <TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
      <TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
    </Sample>
  </PcmmDpsLatencyStats>
</Statistics>

```

```

<TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
<TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
<TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
<TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
<TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
<TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
<TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
<TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
<TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
<TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
<TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
<TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
<TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
<TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
<TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
<TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
  </Sample>
</PcmmDpsPeerLatencyStats>
</Statistics>

```

Request

This request follows the **QueryOmStats > PcmmDpsPeerLatencyStats** tag defined in the XSDs.

The following is an example of a request for a single policy server:

```

<?xml version="1.0" encoding="UTF-8"?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2001-12-31T12:00:00</StartTime>
    <EndTime>2001-12-31T12:15:00</EndTime>
    <PcmmDpsPeerLatencyStats>
      <Name>DpsName</Name>
    </PcmmDpsPeerLatencyStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

The response to this request follows the **Statistics > PcmmDpsPeerLatencyStats** tag defined in the XSDs.

The following is an example of a response for a single policy server:

```

<Statistics>
  <PcmmDpsPeerLatencyStats>
    <Sample>
      <StartTime>2012-06-07T01:00:00Z</StartTime>
      <EndTime>2012-06-07T01:15:00Z</EndTime>
      <PolicyServer>PolicyServer</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>DpsName</Name>
      <DpsId>NeId</DpsId>
      <ConnectAddress>10.60.4.56</ConnectAddress>
      <MaxTransactionInProcessingTime>0</MaxTransactionInProcessingTime>
      <AverageTransactionInProcessingTime>0</AverageTransactionInProcessingTime>

      <MaxTransactionOutProcessingTime>0</MaxTransactionOutProcessingTime>
      <AverageTransactionOutProcessingTime>0</AverageTransactionOutProcessingTime>
    </Sample>
  </PcmmDpsPeerLatencyStats>
</Statistics>

```

```

<TransactionTime_In_0_20_Count>0</TransactionTime_In_0_20_Count>
<TransactionTime_In_20_40_Count>0</TransactionTime_In_20_40_Count>
<TransactionTime_In_40_60_Count>0</TransactionTime_In_40_60_Count>
<TransactionTime_In_60_80_Count>0</TransactionTime_In_60_80_Count>
<TransactionTime_In_80_100_Count>0</TransactionTime_In_80_100_Count>
<TransactionTime_In_100_120_Count>0</TransactionTime_In_100_120_Count>
<TransactionTime_In_120_140_Count>0</TransactionTime_In_120_140_Count>
<TransactionTime_In_140_160_Count>0</TransactionTime_In_140_160_Count>
<TransactionTime_In_160_180_Count>0</TransactionTime_In_160_180_Count>
<TransactionTime_In_180_200_Count>0</TransactionTime_In_180_200_Count>
<TransactionTime_In_gt_200_Count>0</TransactionTime_In_gt_200_Count>
<TransactionTime_Out_0_20_Count>0</TransactionTime_Out_0_20_Count>
<TransactionTime_Out_20_40_Count>0</TransactionTime_Out_20_40_Count>
<TransactionTime_Out_40_60_Count>0</TransactionTime_Out_40_60_Count>
<TransactionTime_Out_60_80_Count>0</TransactionTime_Out_60_80_Count>
<TransactionTime_Out_80_100_Count>0</TransactionTime_Out_80_100_Count>
<TransactionTime_Out_100_120_Count>0</TransactionTime_Out_100_120_Count>
<TransactionTime_Out_120_140_Count>0</TransactionTime_Out_120_140_Count>
<TransactionTime_Out_140_160_Count>0</TransactionTime_Out_140_160_Count>
<TransactionTime_Out_160_180_Count>0</TransactionTime_Out_160_180_Count>
<TransactionTime_Out_180_200_Count>0</TransactionTime_Out_180_200_Count>
<TransactionTime_Out_gt_200_Count>0</TransactionTime_Out_gt_200_Count>
</Sample>
</PcmmDpsPeerLatencyStats>
</Statistics>

```

RADIUS-S Statistics

The following examples show the request and response that are defined in the XSDs for the RadiusStats tag.

Request

This request follows the **QueryOmStats > RadiusStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <RadiusStats></RadiusStats>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics > RadiusStats** tag defined in the XSDs.

- MessagesInCount — Number of messages in.
- MessagesOutCount — Number of messages out.
- AccountingStartCount — Number of Accounting Start messages received.
- AccountingStopCount — Number of Accounting Stop messages received.
- AccountingUpdateCount — Number of Accounting Interim Update messages received.
- AccountingResponseCount — Number of Accounting Responses sent.
- AccessRequestCount — Number of Access-Request messages received.

- AccessAcceptCount — Number of Access-Accept messages sent.
- AccessRejectCount — Number of Access-Reject messages sent.
- ReceivedMd5ErrorCount — Number of received messages dropped because of MD5 errors.
- SendMd5ErrorCount — Number of CoA requests that failed because of MD5 errors.
- ReceivedErrorCount — Number of received messages dropped because of errors.
- CoACount — Number of CoA messages sent.
- CoASuccessCount — Number of CoA-ACK messages received.
- CoAProvisionCount — Number of CoA messages to provision a default QOS profile.
- CoaApplicationCount — Number of CoA messages to adjust QOS for an application request.
- CoANckCount — Number of CoA-NAK messages received.
- NoResponseCount — Number of requests that received no response.
- UnknownGatewayRequestCount — Number of received messages dropped from unknown gateways.
- ResendCount — Number of CoA requests that were retransmitted.
- SendErrorCount — Number of CoA requests that were not sent because of errors.

The following is an example XML response to a RadiusStats tag request:

```
<?xml version="1.0" ?>
<Statistics>
  <RadiusStats>
    <Sample>
      <StartTime>2007-01-29T17:32:59Z</StartTime>
      <EndTime>2007-01-29T17:35:37Z</EndTime>
      <PolicyServer>localhost</PolicyServer>
      <IsComplete>>true</IsComplete>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <AccountingStartCount>0</AccountingStartCount>
      <AccountingStopCount>0</AccountingStopCount>
      <AccountingUpdateCount>0</AccountingUpdateCount>
      <AccountingResponseCount>0</AccountingResponseCount>
      <AccessRequestCount>0</AccessRequestCount>
      <AccessAcceptCount>0</AccessAcceptCount>
      <AccessRejectCount>0</AccessRejectCount>
      <ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
      <SendMd5ErrorCount>0</SendMd5ErrorCount>
      <ReceivedErrorCount>0</ReceivedErrorCount>
      <CoACount>0</CoACount>
      <CoASuccessCount>0</CoASuccessCount>
      <CoAProvisionCount>0</CoAProvisionCount>
      <CoaApplicationCount>0</CoaApplicationCount>
      <CoANckCount>0</CoANckCount>
      <NoResponseCount>0</NoResponseCount>
      <UnknownGatewayRequestCount>0</UnknownGatewayRequestCount>
      <ResendCount>0</ResendCount>
      <SendErrorCount>0</SendErrorCount>
    </Sample>
    <Sample>
      <StartTime>2007-01-29T17:35:37Z</StartTime>
      <EndTime>2007-01-29T17:37:13Z</EndTime>
      <PolicyServer>localhost</PolicyServer>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <AccountingStartCount>0</AccountingStartCount>
      <AccountingStopCount>0</AccountingStopCount>
      <AccountingUpdateCount>0</AccountingUpdateCount>
```

```

<AccountingResponseCount>0</AccountingResponseCount>
<AccessRequestCount>0</AccessRequestCount>
<AccessAcceptCount>0</AccessAcceptCount>
<AccessRejectCount>0</AccessRejectCount>
<ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
<SendMd5ErrorCount>0</SendMd5ErrorCount>
<ReceivedErrorCount>0</ReceivedErrorCount>
<CoACount>0</CoACount>
<CoASuccessCount>0</CoASuccessCount>
<CoAProvisionCount>0</CoAProvisionCount>
<CoaApplicationCount>0</CoaApplicationCount>
<CoANckCount>0</CoANckCount>
<NoResponseCount>0</NoResponseCount>
<UnknownGatewayRequestCount>0</UnknownGatewayRequestCount>
<ResendCount>0</ResendCount>
<SendErrorCount>0</SendErrorCount>
</Sample>
<Sample>
<StartTime>2007-01-29T17:37:13Z</StartTime>
<EndTime>2007-01-29T17:45:00Z</EndTime>
<PolicyServer>localhost</PolicyServer>
<IsComplete>>true</IsComplete>
<MessagesInCount>0</MessagesInCount>
<MessagesOutCount>0</MessagesOutCount>
<AccountingStartCount>0</AccountingStartCount>
<AccountingStopCount>0</AccountingStopCount>
<AccountingUpdateCount>0</AccountingUpdateCount>
<AccountingResponseCount>0</AccountingResponseCount>
<AccessRequestCount>0</AccessRequestCount>
<AccessAcceptCount>0</AccessAcceptCount>
<AccessRejectCount>0</AccessRejectCount>
<ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
<SendMd5ErrorCount>0</SendMd5ErrorCount>
<ReceivedErrorCount>0</ReceivedErrorCount>
<CoACount>0</CoACount>
<CoASuccessCount>0</CoASuccessCount>
<CoAProvisionCount>0</CoAProvisionCount>
<CoaApplicationCount>0</CoaApplicationCount>
<CoANckCount>0</CoANckCount>
<NoResponseCount>0</NoResponseCount>
<UnknownGatewayRequestCount>0</UnknownGatewayRequestCount>
<ResendCount>0</ResendCount>
<SendErrorCount>0</SendErrorCount>
</Sample>
<Sample>
<StartTime>2007-01-29T18:30:00Z</StartTime>
<EndTime>2007-01-29T18:45:00Z</EndTime>
<PolicyServer>localhost</PolicyServer>
<IsComplete>>true</IsComplete>
<MessagesInCount>0</MessagesInCount>
<MessagesOutCount>0</MessagesOutCount>
<AccountingStartCount>0</AccountingStartCount>
<AccountingStopCount>0</AccountingStopCount>
<AccountingUpdateCount>0</AccountingUpdateCount>
<AccountingResponseCount>0</AccountingResponseCount>
<AccessRequestCount>0</AccessRequestCount>
<AccessAcceptCount>0</AccessAcceptCount>
<AccessRejectCount>0</AccessRejectCount>
<ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
<SendMd5ErrorCount>0</SendMd5ErrorCount>
<ReceivedErrorCount>0</ReceivedErrorCount>
<CoACount>0</CoACount>
<CoASuccessCount>0</CoASuccessCount>
<CoAProvisionCount>0</CoAProvisionCount>

```

```

    <CoaApplicationCount>0</CoaApplicationCount>
    <CoANckCount>0</CoANckCount>
    <NoResponseCount>0</NoResponseCount>
    <UnknownGatewayRequestCount>0</UnknownGatewayRequestCount>
    <ResendCount>0</ResendCount>
    <SendErrorCount>0</SendErrorCount>
  </Sample>
</RadiusStats>
</Statistics>

```

RADIUS-S Network Element Statistics

The following examples show the request and response that are defined in the XSDs for the RadiusNetworkElementStats tag.

Request

This request follows the **QueryOmStats > RadiusNetworkElementStats** tag defined in the XSDs.

The following is an example of a request for a single network element using the Name parameter:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <RadiusNetworkElementStats>
    <Name>Router1</Name>
  </RadiusNetworkElementStats>
</QueryOmStats>

```

The following is an example of a request for multiple network elements using the Name(s) and NeId(s) parameters. This example returns statistics for three different network elements:

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <RadiusNetworkElementStats>
    <Name>Router1</Name>
    <Name>Router2</Name>
    <Name>Router3</Name>
  </RadiusNetworkElementStats>
</QueryOmStats>

```

The following is an example of a request for all network elements in the system. This example returns statistics for each network element:

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <RadiusNetworkElementStats></RadiusNetworkElementStats>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics > RadiusNetworkElementStats** tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Sample groups are ordered by policy server, network element, and then by time.

- Name — Unique name identifying the Network Element for the following statistics.
 - NeId — Optional identifier field for the Network Element.
 - NetworkElementType — Type of the Network Element.
 - NetworkElementSubType — Sub Type of the Network Element.
 - AccountingStartCount — Number of Accounting Start messages received.
 - AccountingStopCount — Number of Accounting Stop messages received.
 - AccountingUpdateCount — Number of Accounting Interim Update messages received.
 - AccountingResponseCount — Number of Accounting responses sent.
 - AccessRequestCount — Number of Access-Request messages received.
 - AccessAcceptCount — Number of Access-Accept messages received.
 - AccessRejectCount — Number of Access-Reject messages received.
 - ReceivedMd5ErrorCount — Number of received messages dropped because of MD5 errors.
 - SendMd5ErrorCount — Number of CoA requests that failed because of MD5 errors.
 - ReceivedErrorCount — Number of messages dropped because of errors.
 - CoACount — Number of CoA messages sent.
 - CoASuccessCount — Number of CoA-ACK messages received.
 - CoAProvisionCount — Number of CoA messages to provision a default QoS profile.
 - CoaApplicationCount — Number of CoA messages to adjust QoS for an application request.
 - CoANckCount — Number of CoA-NAK messages received.
 - NoResponseCount — Number of CoA requests that received no response.
 - SendErrorCount — Number of CoA requests that were not sent because of errors.
 - ResendCount — Number of CoA requests that were retransmitted.

The following is an example XML response to a multiple network element request:

```
<?xml version="1.0" ?>
<Statistics>
  <RadiusNetworkElementStats>
    <Sample>
      <StartTime>2007-01-29T17:32:59Z</StartTime>
      <EndTime>2007-01-29T17:35:37Z</EndTime>
      <PolicyServer>localhost</PolicyServer>
      <Name>simPDSN</Name>
      <NeId/>
      <NetworkElementType>PDSN</NetworkElementType>
      <NetworkElementSubType>Starent</NetworkElementSubType>
      <AccountingStartCount>0</AccountingStartCount>
      <AccountingStopCount>0</AccountingStopCount>
      <AccountingUpdateCount>0</AccountingUpdateCount>
      <AccountingResponseCount>0</AccountingResponseCount>
      <AccessRequestCount>0</AccessRequestCount>
      <AccessAcceptCount>0</AccessAcceptCount>
      <AccessRejectCount>0</AccessRejectCount>
      <ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
      <SendMd5ErrorCount>0</SendMd5ErrorCount>
```

```

    <ReceivedErrorCount>0</ReceivedErrorCount>
    <CoACount>0</CoACount>
    <CoASuccessCount>0</CoASuccessCount>
    <CoAProvisionCount>0</CoAProvisionCount>
    <CoaApplicationCount>0</CoaApplicationCount>
    <CoANckCount>0</CoANckCount>
    <NoResponseCount>0</NoResponseCount>
    <SendErrorCount>0</SendErrorCount>
    <ResendCount>0</ResendCount>
  </Sample>
  <Sample>
    <StartTime>2007-01-29T17:35:37Z</StartTime>
    <EndTime>2007-01-29T17:37:13Z</EndTime>
    <PolicyServer>localhost</PolicyServer>
    <Name>simPDSN</Name>
    <NeId/>
    <NetworkElementType>PDSN</NetworkElementType>
    <NetworkElementSubType>Starent</NetworkElementSubType>
    <AccountingStartCount>0</AccountingStartCount>
    <AccountingStopCount>0</AccountingStopCount>
    <AccountingUpdateCount>0</AccountingUpdateCount>
    <AccountingResponseCount>0</AccountingResponseCount>
    <AccessRequestCount>0</AccessRequestCount>
    <AccessAcceptCount>0</AccessAcceptCount>
    <AccessRejectCount>0</AccessRejectCount>
    <ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
    <SendMd5ErrorCount>0</SendMd5ErrorCount>
    <ReceivedErrorCount>0</ReceivedErrorCount>
    <CoACount>0</CoACount>
    <CoASuccessCount>0</CoASuccessCount>
    <CoAProvisionCount>0</CoAProvisionCount>
    <CoaApplicationCount>0</CoaApplicationCount>
    <CoANckCount>0</CoANckCount>
    <NoResponseCount>0</NoResponseCount>
    <SendErrorCount>0</SendErrorCount>
    <ResendCount>0</ResendCount>
  </Sample>
  <Sample>
    <StartTime>2007-01-29T17:37:13Z</StartTime>
    <EndTime>2007-01-29T17:45:00Z</EndTime>
    <PolicyServer>localhost</PolicyServer>
    <Name>simPDSN</Name>
    <NeId/>
    <NetworkElementType>PDSN</NetworkElementType>
    <NetworkElementSubType>Starent</NetworkElementSubType>
    <AccountingStartCount>0</AccountingStartCount>
    <AccountingStopCount>0</AccountingStopCount>
    <AccountingUpdateCount>0</AccountingUpdateCount>
    <AccountingResponseCount>0</AccountingResponseCount>
    <AccessRequestCount>0</AccessRequestCount>
    <AccessAcceptCount>0</AccessAcceptCount>
    <AccessRejectCount>0</AccessRejectCount>
    <ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
    <SendMd5ErrorCount>0</SendMd5ErrorCount>
    <ReceivedErrorCount>0</ReceivedErrorCount>
    <CoACount>0</CoACount>
    <CoASuccessCount>0</CoASuccessCount>
    <CoAProvisionCount>0</CoAProvisionCount>
    <CoaApplicationCount>0</CoaApplicationCount>
    <CoANckCount>0</CoANckCount>
    <NoResponseCount>0</NoResponseCount>
    <SendErrorCount>0</SendErrorCount>
    <ResendCount>0</ResendCount>
  </Sample>

```

```

<Sample>
  <StartTime>2007-01-29T17:45:00Z</StartTime>
  <EndTime>2007-01-29T18:00:01Z</EndTime>
  <PolicyServer>localhost</PolicyServer>
  <Name>simPDSN</Name>
  <NeId/>
  <NetworkElementType>PDSN</NetworkElementType>
  <NetworkElementSubType>Starent</NetworkElementSubType>
  <AccountingStartCount>0</AccountingStartCount>
  <AccountingStopCount>0</AccountingStopCount>
  <AccountingUpdateCount>0</AccountingUpdateCount>
  <AccountingResponseCount>0</AccountingResponseCount>
  <AccessRequestCount>0</AccessRequestCount>
  <AccessAcceptCount>0</AccessAcceptCount>
  <AccessRejectCount>0</AccessRejectCount>
  <ReceivedMd5ErrorCount>0</ReceivedMd5ErrorCount>
  <SendMd5ErrorCount>0</SendMd5ErrorCount>
  <ReceivedErrorCount>0</ReceivedErrorCount>
  <CoACount>0</CoACount>
  <CoASuccessCount>0</CoASuccessCount>
  <CoAProvisionCount>0</CoAProvisionCount>
  <CoaApplicationCount>0</CoaApplicationCount>
  <CoANckCount>0</CoANckCount>
  <NoResponseCount>0</NoResponseCount>
  <SendErrorCount>0</SendErrorCount>
  <ResendCount>0</ResendCount>
</Sample>
</RadiusNetworkElementStats>
</Statistics>

```

Connected Network Elements Statistics

The following examples show the request and response that are defined in the XSDs for the `ConnectedNetworkElementStats` tag.

Request

This request follows the `QueryOmStats > ConnectedNetworkElementStats` tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <ConnectedNetworkElementStats></ConnectedNetworkElementStats>
</QueryOmStats>

```

Response

The response to this request follows the `Statistics > ConnectedNetworkElementStats` tag defined in the XSDs.

`ConnectedNetworkElementCount` — The absolute number of network elements maintaining a consistent connection to each MPE. For example, B-RAS elements.

The following is an example XML response to a ConnectedNetworkElementStats tag request:

```
<xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <ConnectedNetworkElementStats>
    <Sample>
      <StartTime>2007-10-26T14:30:22Z</StartTime>
      <EndTime>2007-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <ConnectedNetworkElementCount>52</ConnectedNetworkElementCount>
    </Sample>
  </ConnectedNetworkElementStats>
</Statistics>
```

Mgpi Statistics

The following examples show the request and response that are defined in the XSDs for the MgpiStats tag.

Request

This request follows the **QueryOmStats** > **MgpiStats** tag defined in the XSDs.

The following is an example of a request for all policy servers within the system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <MgpiStats></MgpiStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics** > **MgpiStats** tag defined in the XSDs.

- **TotalFlowsCount** — The number of active upstream diameter session flows in memory.
- **ActualGatesCount** — The number of actual upstream gates in memory.
- **MultiFlowGatesCount** — The number of gates in memory that have been aggregated, (for example, possessing multiple diameter session flows on this gate).
- **EffectiveGatesCount** — A calculation using the formula $\text{Effective Gates} = (\text{Active Gate Count} - \text{Actual Gate Count}) + \text{Total Flow Count}$.

Note: Active gate count is the number of active gates, including the upstream gates and downstream gates.

Example:

An MPE Upstream Service Flow Limit for Triggering MGPI = 2, Maximum Number of Grants per Interval = 2 and MGPI = yes.

Send 3 same AAR requests to MPE, Active Gate Count = 5, Actual Gate Count = 2, Total Flow Count = 3.

The Effective Gate Count = $(5-2)+3 = 6$

The following is an example XML response to a MgpStats tag request:

```
<xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <MgpStats>
    <Sample>
      <StartTime>2007-10-26T14:30:22Z</StartTime>
      <EndTime>2007-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <IsComplete>true</IsComplete>
      <TotalFlowsCount>10</TotalFlowsCount>
      <ActualGatesCount>52</ActualGatesCount>
      <MultiFlowGatesCount>52</MultiFlowGatesCount>
      <EffectiveGatesCount>15</EffectiveGatesCount>
    </Sample>
  </MgpStats>
</Statistics>
```

Diameter Operational Measurements Requests

This section shows the requests and responses for:

- [Diameter Application Function Statistics](#)
- [Diameter Application Function Peer Statistics](#)
- [Diameter Charging Function Statistics](#)
- [Diameter Charging Function Peer Statistics](#)
- [Diameter Policy Charging Enforcement Function Statistics](#)
- [Diameter Policy Charging Enforcement Function Peer Statistics](#)

Diameter Application Function Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterAfStats tag.

Request

This request follows the **QueryOmStats** > **DiameterAfStats** tag defined in the XSDs.

The following is an example of a request for all policy servers within the system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-10-26T14:30:00Z</StartTime>
  <EndTime>2007-10-26T14:40:00Z</EndTime>
  <DiameterAfStats/>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics** > **DiameterAfStats** tag defined in the XSDs.

The following is an example XML response to a DiameterAfStats tag request:

```

<Statistics>
  <DiameterAfStats>
    <Sample>
      <StartTime>2011-09-01T00:00:02Z</StartTime>
      <EndTime>2011-09-01T00:15:02Z</EndTime>
      <PolicyServer>10.15.24.106</PolicyServer>
      <IsComplete>>true</IsComplete>
      <PendingConnectionsCount>0</PendingConnectionsCount>
      <CurrentConnectionsCount>1</CurrentConnectionsCount>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <ASRMessagesReceivedCount>0</ASRMessagesReceivedCount>
      <ASRMessagesSentCount>0</ASRMessagesSentCount>
      <ASASuccessMessagesReceivedCount>0</ASASuccessMessagesReceivedCount>
      <ASASuccessMessagesSentCount>0</ASASuccessMessagesSentCount>
      <ASAFailureMessagesReceivedCount>0</ASAFailureMessagesReceivedCount>
      <ASAFailureMessagesSentCount>0</ASAFailureMessagesSentCount>
      <RARMessagesReceivedCount>0</RARMessagesReceivedCount>
      <RARMessagesSentCount>0</RARMessagesSentCount>
      <RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
      <RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
      <RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
      <RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
      <STRMessagesReceivedCount>0</STRMessagesReceivedCount>
      <STRMessagesSentCount>0</STRMessagesSentCount>
      <STASuccessMessagesReceivedCount>0</STASuccessMessagesReceivedCount>
      <STASuccessMessagesSentCount>0</STASuccessMessagesSentCount>
      <STAFailureMessagesReceivedCount>0</STAFailureMessagesReceivedCount>
      <STAFailureMessagesSentCount>0</STAFailureMessagesSentCount>
      <AARMessagesReceivedCount>0</AARMessagesReceivedCount>
      <AARMessagesSentCount>0</AARMessagesSentCount>
      <AARInitialMessagesReceivedCount>0</AARInitialMessagesReceivedCount>
      <AARInitialMessagesSentCount>0</AARInitialMessagesSentCount>
      <AARModificationMessagesReceivedCount>0</AARModificationMessagesReceivedCount>
      <AARModificationMessagesSentCount>0</AARModificationMessagesSentCount>
      <AAASuccessMessagesReceivedCount>0</AAASuccessMessagesReceivedCount>
      <AAASuccessMessagesSentCount>0</AAASuccessMessagesSentCount>
      <AAAFailureMessagesReceivedCount>0</AAAFailureMessagesReceivedCount>
      <AAAFailureMessagesSentCount>0</AAAFailureMessagesSentCount>
      <RxPcmmMessagesTimeoutCount>0</RxPcmmMessagesTimeoutCount>
      <ActiveSessionsCount>0</ActiveSessionsCount>
      <MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
      <PeerOkayCount>1</PeerOkayCount>
      <PeerDownCount>0</PeerDownCount>
      <PeerSuspectCount>0</PeerSuspectCount>
      <PeerReopenCount>0</PeerReopenCount>
    </Sample>
  </DiameterAfStats>
</Statistics>

```

Diameter Application Function Peer Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterAfPeerStats tag.

Request

This request follows the QueryOmStats ->DiameterAfPeerStats tag defined in the XSDs.

The following is an example of a request for a single peer using the Name parameter:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <DiameterAfPeerStats>
    <Name>Server1</Name>
  </DiameterAfPeerStats>
</QueryOmStats>
```

The following is an example of a request for all peer elements in the system. This request returns stats for each peer element:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2007-01-29T17:30:00Z</StartTime>
  <EndTime>2006-01-29T18:10:00Z</EndTime>
  <DiameterAfPeerStats></DiameterAfPeerStats>
</QueryOmStats>
```

Response

The response to this request follows the Statistics -> DiameterAfPeertStats tag defined in the XSDs.

Both the Name and NeId are returned, along with the statistics and actual recorded times for those statistics. Samples are ordered by policy server, network element, and then by time.

The following is an example XML response to a multiple network element tag request:

```
<Statistics>
  <DiameterAfPeerStats>
    <Sample>
      <StartTime>2011-08-19T11:15:02Z</StartTime>
      <EndTime>2011-08-19T11:30:00Z</EndTime>
      <PolicyServer>10.60.24.47</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>identity2</Name>
      <NeId></NeId>
      <NetworkElementType></NetworkElementType>
      <NetworkElementSubType></NetworkElementSubType>
      <ConnectTime></ConnectTime>
      <DisconnectTime></DisconnectTime>
      <ConnectAddress></ConnectAddress>
      <ConnectPort></ConnectPort>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <ASRMessagesReceivedCount>0</ASRMessagesReceivedCount>
      <ASRMessagesSentCount>0</ASRMessagesSentCount>
      <ASASuccessMessagesReceivedCount>0</ASASuccessMessagesReceivedCount>
      <ASASuccessMessagesSentCount>0</ASASuccessMessagesSentCount>
      <ASAFailureMessagesReceivedCount>0</ASAFailureMessagesReceivedCount>
      <ASAFailureMessagesSentCount>0</ASAFailureMessagesSentCount>
      <RARMessagesReceivedCount>0</RARMessagesReceivedCount>
      <RARMessagesSentCount>0</RARMessagesSentCount>
      <RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
      <RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
```

```

<RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
<RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
<STRMessagesReceivedCount>0</STRMessagesReceivedCount>
<STRMessagesSentCount>0</STRMessagesSentCount>
<STASuccessMessagesReceivedCount>0</STASuccessMessagesReceivedCount>
<STASuccessMessagesSentCount>0</STASuccessMessagesSentCount>
<STAFailureMessagesReceivedCount>0</STAFailureMessagesReceivedCount>
<STAFailureMessagesSentCount>0</STAFailureMessagesSentCount>
<AARMessagesReceivedCount>0</AARMessagesReceivedCount>
<AARMessagesSentCount>0</AARMessagesSentCount>
<AARInitialMessagesReceivedCount>0</AARInitialMessagesReceivedCount>
<AARInitialMessagesSentCount>0</AARInitialMessagesSentCount>

<AARModificationMessagesReceivedCount>0</AARModificationMessagesReceivedCount>
<AARModificationMessagesSentCount>0</AARModificationMessagesSentCount>

<AAASuccessMessagesReceivedCount>0</AAASuccessMessagesReceivedCount>
<AAASuccessMessagesSentCount>0</AAASuccessMessagesSentCount>
<AAAFailureMessagesReceivedCount>0</AAAFailureMessagesReceivedCount>
<AAAFailureMessagesSentCount>0</AAAFailureMessagesSentCount>
<RxPcmmMessagesTimeoutCount>0</RxPcmmMessagesTimeoutCount>
  </Sample>
</DiameterAfPeerStats>
</Statistics>

```

Diameter Charging Function Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterCTFStats tag.

Request

This request follows the QueryOmStats ->DiameterCTFStats tag defined in the XSDs.

The following is an example of a request for all policy servers in the system:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-01-29T17:30:00Z</StartTime>
  <EndTime>2007-01-29T18:50:00Z</EndTime>
  <DiameterCTFStats></DiameterCTFStats>
</QueryOmStats>

```

Response

The response to this request follows the Statistics -> DiameterCTFStats tag defined in the XSDs.

The following is an example XML response to a multiple network element tag request:

```

<?xml version='1.0' ?>
<Statistics>
  <DiameterCTFStats>
    <Sample>
      <StartTime>2009-08-19T19:51:28Z</StartTime>
      <EndTime>2009-08-19T19:52:47Z</EndTime>
      <PolicyServer>localps</PolicyServer>
      <IsComplete>>true</IsComplete>
      <CurrentConnectionsCount>0</CurrentConnectionsCount>
    </Sample>
  </DiameterCTFStats>
</Statistics>

```

```

<PeerOkayCount>0</PeerOkayCount>
<PeerDownCount>0</PeerDownCount>
<PeerSuspectCount>0</PeerSuspectCount>
<PeerReopenCount>0</PeerReopenCount>
<MessagesInCount>0</MessagesInCount>
<MessagesOutCount>0</MessagesOutCount>
<CCRMessagesReceivedCount>0</CCRMessagesReceivedCount>
<CCRMessagesSentCount>0</CCRMessagesSentCount>
<CCASuccessMessagesReceivedCount>0</CCASuccessMessagesReceivedCount>
<CCASuccessMessagesSentCount>0</CCASuccessMessagesSentCount>
<CCAFailureMessagesReceivedCount>0</CCAFailureMessagesReceivedCount>
<CCAFailureMessagesSentCount>0</CCAFailureMessagesSentCount>
<RARMessagesReceivedCount>0</RARMessagesReceivedCount>
<RARMessagesSentCount>0</RARMessagesSentCount>
<RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
<RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
<RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
<RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
<ASRMessagesReceivedCount>0</ASRMessagesReceivedCount>
<ASRMessagesSentCount>0</ASRMessagesSentCount>
<ASASuccessMessagesReceivedCount>0</ASASuccessMessagesReceivedCount>
<ASASuccessMessagesSentCount>0</ASASuccessMessagesSentCount>
<ASAFailureMessagesReceivedCount>0</ASAFailureMessagesReceivedCount>
<ASAFailureMessagesSentCount>0</ASAFailureMessagesSentCount>
<ActiveSessionsCount>0</ActiveSessionsCount>
<MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
</Sample>
</DiameterCTFStats>
</Statistics>

```

Diameter Charging Function Peer Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterCTFPeerStats tag.

Request

This request follows the **QueryOmStats > DiameterCTFPeerStats** tag defined in the XSDs.

The following is an example of a request for a single network element using the Name parameter:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <DiameterCTFPeerStats>
    <Name>Server1</Name>
  </DiameterCTFPeerStats>
</QueryOmStats>

```

The following is an example of a request for all network elements in the system. This request returns statistics for each network element.

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2007-01-29T17:30:00Z</StartTime>
  <EndTime>2006-01-29T18:10:00Z</EndTime>

```

```
<DiameterCTFPeerStats/>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > DiameterCTFPeerStats** tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Samples are ordered by policy server, network element, and then by time.

The following is an example XML response to a multiple network element tag request:

```
<?xml version='1.0' ?>
<Statistics>
  <DiameterCTFPeerStats>
    <Sample>
      <StartTime>2009-08-19T20:08:53Z</StartTime>
      <EndTime>2009-08-19T20:15:00Z</EndTime>
      <PolicyServer>localps</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>ctf1</Name>
      <NeId></NeId>
      <NetworkElementType></NetworkElementType>
      <NetworkElementSubType></NetworkElementSubType>
      <ActiveConnectionCount>0</ActiveConnectionCount>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <MessagesErrorInCount>0</MessagesErrorInCount>
      <MessagesErrorOutCount>0</MessagesErrorOutCount>
      <CCRMessagesReceivedCount>0</CCRMessagesReceivedCount>
      <CCRMessagesSentCount>0</CCRMessagesSentCount>
      <CCASuccessMessagesReceivedCount>0</CCASuccessMessagesReceivedCount>
      <CCASuccessMessagesSentCount>0</CCASuccessMessagesSentCount>
      <CCAFailureMessagesReceivedCount>0</CCAFailureMessagesReceivedCount>
      <CCAFailureMessagesSentCount>0</CCAFailureMessagesSentCount>
      <RARMessagesReceivedCount>0</RARMessagesReceivedCount>
      <RARMessagesSentCount>0</RARMessagesSentCount>
      <RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
      <RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
      <RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
      <RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
      <ASRMessagesReceivedCount>0</ASRMessagesReceivedCount>
      <ASRMessagesSentCount>0</ASRMessagesSentCount>
      <ASASuccessMessagesReceivedCount>0</ASASuccessMessagesReceivedCount>
      <ASASuccessMessagesSentCount>0</ASASuccessMessagesSentCount>
      <ASAFailureMessagesReceivedCount>0</ASAFailureMessagesReceivedCount>
      <ASAFailureMessagesSentCount>0</ASAFailureMessagesSentCount>
      <ActiveSessionsCount>0</ActiveSessionsCount>
      <MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
    </Sample>
  </DiameterCTFPeerStats>
</Statistics>
```

Diameter Policy Charging Enforcement Function Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterPcefStats tag.

Request

This request follows the **QueryOmStats > DiameterPcefStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2013-05-17T16:30:00Z</StartTime>
  <EndTime>2013-05-17T17:10:00Z</EndTime>
  <DiameterPcefStats></DiameterPcefStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > DiameterPcefStats** tag defined in the XSDs.

The following is an example XML response to a DiameterPcefStats tag request:

```
<?xml version='1.0' ?>
<Statistics>
  <DiameterPcefStats>
    <Sample>
      <StartTime>2013-05-17T16:45:00Z</StartTime>
      <EndTime>2013-05-17T17:00:00Z</EndTime>
      <PolicyServer>MPE Site 3</PolicyServer>
      <IsComplete>>true</IsComplete>
      <CurrentConnectionsCount>2</CurrentConnectionsCount>
      <MessagesInCount>1013973</MessagesInCount>
      <MessagesOutCount>1013973</MessagesOutCount>
      <RARMessagesReceivedCount>0</RARMessagesReceivedCount>
      <RARMessagesSentCount>0</RARMessagesSentCount>
      <RARMessagesTimeoutCount>0</RARMessagesTimeoutCount>
      <RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
      <RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
      <RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
      <RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
      <CCRMessagesReceivedCount>1013973</CCRMessagesReceivedCount>
      <CCRMessagesSentCount>0</CCRMessagesSentCount>
      <CCRMessagesTimeoutCount>0</CCRMessagesTimeoutCount>
      <CCASuccessMessagesReceivedCount>0</CCASuccessMessagesReceivedCount>
      <CCASuccessMessagesSentCount>1013973</CCASuccessMessagesSentCount>
      <CCAFailureMessagesReceivedCount>0</CCAFailureMessagesReceivedCount>
      <CCAFailureMessagesSentCount>0</CCAFailureMessagesSentCount>
      <CCRIMessagesReceivedCount>634980</CCRIMessagesReceivedCount>
      <CCRIMessagesSentCount>0</CCRIMessagesSentCount>
      <CCRIMessagesTimeoutCount>0</CCRIMessagesTimeoutCount>
      <CCRUMessagesReceivedCount>0</CCRUMessagesReceivedCount>
      <CCRUMessagesSentCount>0</CCRUMessagesSentCount>
      <CCRUMessagesTimeoutCount>0</CCRUMessagesTimeoutCount>
      <CCRTMessagesReceivedCount>378993</CCRTMessagesReceivedCount>
      <CCRTMessagesSentCount>0</CCRTMessagesSentCount>
      <CCRTMessagesTimeoutCount>0</CCRTMessagesTimeoutCount>
      <CCAISuccessMessagesReceivedCount>0</CCAISuccessMessagesReceivedCount>
      <CCAISuccessMessagesSentCount>634980</CCAISuccessMessagesSentCount>
      <CCAIFailureMessagesReceivedCount>0</CCAIFailureMessagesReceivedCount>
      <CCAIFailureMessagesSentCount>0</CCAIFailureMessagesSentCount>
      <CCAUSuccessMessagesReceivedCount>0</CCAUSuccessMessagesReceivedCount>
      <CCAUSuccessMessagesSentCount>0</CCAUSuccessMessagesSentCount>
      <CCAUFailureMessagesReceivedCount>0</CCAUFailureMessagesReceivedCount>
      <CCAUFailureMessagesSentCount>0</CCAUFailureMessagesSentCount>
      <CCATSuccessMessagesReceivedCount>0</CCATSuccessMessagesReceivedCount>
      <CCATSuccessMessagesSentCount>378993</CCATSuccessMessagesSentCount>
```

```

<CCATFailureMessagesReceivedCount>0</CCATFailureMessagesReceivedCount>
<CCATFailureMessagesSentCount>0</CCATFailureMessagesSentCount>
<ActiveSessionsCount>255987</ActiveSessionsCount>
<MaximumActiveSessionsCount>257999</MaximumActiveSessionsCount>
<PeerOkayCount>2</PeerOkayCount>
<PeerDownCount>0</PeerDownCount>
<PeerSuspectCount>0</PeerSuspectCount>
<PeerReopenCount>0</PeerReopenCount>
</Sample>
</DiameterPcefStats>
</Statistics>

```

Diameter Policy Charging Enforcement Function Peer Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterPcefPeerStats tag.

Request

This request follows the **QueryOmStats** > **DiameterPcefPeerStats** tag defined in the XSDs.

The following is an example of a request for a single network element, using the Name parameter.

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2013-04-25T14:30:00Z</StartTime>
  <EndTime>2013-04-25T14:35:00Z</EndTime>
  <DiameterPcefPeerStats>
    <Name>Server1</Name>
  </DiameterPcefPeerStats>
</QueryOmStats>

```

The following is an example of a request for all network elements in the system. This example returns statistics for each network element.

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2013-04-25T17:30:00Z</StartTime>
  <EndTime>2013-04-25T18:10:00Z</EndTime>
  <DiameterPcefPeerStats></DiameterPcefPeerStats>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics** > **DiameterPcefPeerStats** tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Samples are ordered by policy server, network element, and then by time.

The following is an example XML response to a DiameterPcefPeerStats tag request:

```

<?xml version='1.0' ?>
<Statistics>
  <DiameterPcefPeerStats>
    <Sample>
      <StartTime>2013-04-25T16:45:00Z</StartTime>

```

```

<EndTime>2013-04-25T17:00:00Z</EndTime>
<PolicyServer>MPE</PolicyServer>
<IsComplete>>true</IsComplete>
<Name>mra.name.com</Name>
<NeId></NeId>
<NetworkElementType></NetworkElementType>
<NetworkElementSubType></NetworkElementSubType>
<ConnectTime>Fri Jan 25 22:58:27 EST 2013</ConnectTime>
<DisconnectTime>Thu Jan 24 09:30:21 EST 2013</DisconnectTime>
<ConnectAddress>10.15.23.193</ConnectAddress>
<ConnectPort>57669</ConnectPort>
<ConnectType>TCP</ConnectType>
<MessagesInCount>0</MessagesInCount>
<MessagesOutCount>0</MessagesOutCount>
<MessagesErrorInCount>0</MessagesErrorInCount>
<MessagesErrorOutCount>0</MessagesErrorOutCount>
<RARMessagesReceivedCount>0</RARMessagesReceivedCount>
<RARMessagesSentCount>0</RARMessagesSentCount>
<RARMessagesTimeoutCount>0</RARMessagesTimeoutCount>
<RAASuccessMessagesReceivedCount>0</RAASuccessMessagesReceivedCount>
<RAASuccessMessagesSentCount>0</RAASuccessMessagesSentCount>
<RAAFailureMessagesReceivedCount>0</RAAFailureMessagesReceivedCount>
<RAAFailureMessagesSentCount>0</RAAFailureMessagesSentCount>
<CCRMessagesReceivedCount>0</CCRMessagesReceivedCount>
<CCRMessagesSentCount>0</CCRMessagesSentCount>
<CCRMessagesTimeoutCount>0</CCRMessagesTimeoutCount>
<CCASuccessMessagesReceivedCount>0</CCASuccessMessagesReceivedCount>
<CCASuccessMessagesSentCount>0</CCASuccessMessagesSentCount>
<CCAFailureMessagesReceivedCount>0</CCAFailureMessagesReceivedCount>
<CCAFailureMessagesSentCount>0</CCAFailureMessagesSentCount>
<CCRIMessagesReceivedCount>0</CCRIMessagesReceivedCount>
<CCRIMessagesSentCount>0</CCRIMessagesSentCount>
<CCRIMessagesTimeoutCount>0</CCRIMessagesTimeoutCount>
<CCRUMessagesReceivedCount>0</CCRUMessagesReceivedCount>
<CCRUMessagesSentCount>0</CCRUMessagesSentCount>
<CCRUMessagesTimeoutCount>0</CCRUMessagesTimeoutCount>
<CCRTMessagesReceivedCount>0</CCRTMessagesReceivedCount>
<CCRTMessagesSentCount>0</CCRTMessagesSentCount>
<CCRTMessagesTimeoutCount>0</CCRTMessagesTimeoutCount>
<CCAI SuccessMessagesReceivedCount>0</CCAI SuccessMessagesReceivedCount>
<CCAI SuccessMessagesSentCount>0</CCAI SuccessMessagesSentCount>
<CCAI FailureMessagesReceivedCount>0</CCAI FailureMessagesReceivedCount>
<CCAI FailureMessagesSentCount>0</CCAI FailureMessagesSentCount>
<CCAUSuccessMessagesReceivedCount>0</CCAUSuccessMessagesReceivedCount>
<CCAUSuccessMessagesSentCount>0</CCAUSuccessMessagesSentCount>
<CCAUFailureMessagesReceivedCount>0</CCAUFailureMessagesReceivedCount>
<CCAUFailureMessagesSentCount>0</CCAUFailureMessagesSentCount>
<CCATSuccessMessagesReceivedCount>0</CCATSuccessMessagesReceivedCount>
<CCATSuccessMessagesSentCount>0</CCATSuccessMessagesSentCount>
<CCATFailureMessagesReceivedCount>0</CCATFailureMessagesReceivedCount>
<CCATFailureMessagesSentCount>0</CCATFailureMessagesSentCount>
<ActiveSessionsCount>0</ActiveSessionsCount>
<MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
</Sample>
</DiameterPcefPeerStats>
</Statistics>

```

Diameter Adaptor Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterShStats tag.

Request

This request follows the QueryOmStats ->DiameterShStats tag defined in the XSDs.

The following is an example of a request for a single network element, using the Name parameter.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2010-05-06T16:15:00-05:00</StartTime>
  <EndTime>2010-05-06T16:45:00-05:00</EndTime>
  <Name>hss.test.com</Name>
  <DiameterShStats/>
</QueryOmStats>
```

The following is an example of a request for all network elements in the system. This example returns statistics for each network element.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2010-01-29T17:30:00Z</StartTime>
  <EndTime>2010-01-29T18:10:00Z</EndTime>
  <DiameterShStats/>
</QueryOmStats>
```

Response

The response to this request follows the Statistics -> DiameterShStats tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Samples are ordered by policy server, network element, and then by time.

The following is an example XML response to a DiameterShStats tag request:

```
<?xml version="1.0" ?>
- <Statistics>
- <DiameterShStats>
- <Sample>
  <StartTime>2010-05-06T16:15:00Z</StartTime>
  <EndTime>2010-05-06T16:30:00Z</EndTime>
  <PolicyServer>mpe6-31</PolicyServer>
  <Name>hss.test.com</Name>
  <CurrentConnectionsCount>0</CurrentConnectionsCount>
  <MessagesInCount>0</MessagesInCount>
  <MessagesOutCount>0</MessagesOutCount>
  <UDRMessagesReceivedCount>0</UDRMessagesReceivedCount>
  <UDRMessagesSentCount>0</UDRMessagesSentCount>
  <UDASuccessMessagesReceivedCount>0</UDASuccessMessagesReceivedCount>
  <UDASuccessMessagesSentCount>0</UDASuccessMessagesSentCount>
  <UDAFailureMessagesReceivedCount>0</UDAFailureMessagesReceivedCount>
  <UDAFailureMessagesSentCount>0</UDAFailureMessagesSentCount>
  <PNRMessagesReceivedCount>0</PNRMessagesReceivedCount>
  <PNRMessagesSentCount>0</PNRMessagesSentCount>
  <PNASuccessMessagesReceivedCount>0</PNASuccessMessagesReceivedCount>
  <PNASuccessMessagesSentCount>0</PNASuccessMessagesSentCount>
  <PNAFailureMessagesReceivedCount>0</PNAFailureMessagesReceivedCount>
  <PNAFailureMessagesSentCount>0</PNAFailureMessagesSentCount>
  <PURMessagesReceivedCount>0</PURMessagesReceivedCount>
  <PURMessagesSentCount>0</PURMessagesSentCount>
  <PUASuccessMessagesReceivedCount>0</PUASuccessMessagesReceivedCount>
```

```

<PUASuccessMessagesSentCount>0</PUASuccessMessagesSentCount>
<PUAFailureMessagesReceivedCount>0</PUAFailureMessagesReceivedCount>
<PUAFailureMessagesSentCount>0</PUAFailureMessagesSentCount>
<SNRMessagesReceivedCount>0</SNRMessagesReceivedCount>
<SNRMessagesSentCount>0</SNRMessagesSentCount>
<SNASuccessMessagesReceivedCount>0</SNASuccessMessagesReceivedCount>
<SNASuccessMessagesSentCount>0</SNASuccessMessagesSentCount>
<SNAFailureMessagesReceivedCount>0</SNAFailureMessagesReceivedCount>
<SNAFailureMessagesSentCount>0</SNAFailureMessagesSentCount>
<ActiveSessionsCount>0</ActiveSessionsCount>
<MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
<PeerOkayCount>0</PeerOkayCount>
<PeerDownCount>0</PeerDownCount>
<PeerSuspectCount>0</PeerSuspectCount>
<PeerReopenCount>0</PeerReopenCount>
</Sample>
</DiameterShStats>
</Statistics>

```

Diameter Adaptor Peer Statistics

The following examples show the request and response that are defined in the XSDs for the DiameterShPeerStats tag.

Request

This request follows the QueryOmStats ->DiameterShPeerStats tag defined in the XSDs.

The following is an example of a request for a single network element, using the Name parameter.

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2010-05-06T16:15:00-05:00</StartTime>
  <EndTime>2010-05-06T16:45:00-05:00</EndTime>
  <Name>hss.test.com</Name>
  <DiameterShPeerStats/>
</QueryOmStats>

```

The following is an example of a request for all network elements in the system. This example returns statistics for each network element.

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2010-01-29T17:30:00Z</StartTime>
  <EndTime>2010-01-29T18:10:00Z</EndTime>
  <DiameterShPeerStats/>
</QueryOmStats>

```

Response

The response to this request follows the Statistics -> DiameterShPeerStats tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Samples are ordered by policy server, network element, and then by time.

The following is an example XML response to a DiameterShPeerStats tag request:

```
<?xml version="1.0" ?>
- <Statistics>
- <DiameterShPeerStats>
- <Sample>
  <StartTime>2010-05-10T20:45:00Z</StartTime>
  <EndTime>2010-05-10T21:00:00Z</EndTime>
  <PolicyServer>mpe6-31</PolicyServer>
  <Name>hss.test.com</Name>
  <NeId />
  <NetworkElementType />
  <NetworkElementSubType />
  <ConnectTime />
  <DisconnectTime />
  <ConnectAddress />
  <ConnectPort />
  <MessagesInCount>0</MessagesInCount>
  <MessagesOutCount>0</MessagesOutCount>
  <UDRMessagesReceivedCount>0</UDRMessagesReceivedCount>
  <UDRMessagesSentCount>0</UDRMessagesSentCount>
  <UDASuccessMessagesReceivedCount>0</UDASuccessMessagesReceivedCount>
  <UDASuccessMessagesSentCount>0</UDASuccessMessagesSentCount>
  <UDAFailureMessagesReceivedCount>0</UDAFailureMessagesReceivedCount>
  <UDAFailureMessagesSentCount>0</UDAFailureMessagesSentCount>
  <PNRMessagesReceivedCount>0</PNRMessagesReceivedCount>
  <PNRMessagesSentCount>0</PNRMessagesSentCount>
  <PNASuccessMessagesReceivedCount>0</PNASuccessMessagesReceivedCount>
  <PNASuccessMessagesSentCount>0</PNASuccessMessagesSentCount>
  <PNAFailureMessagesReceivedCount>0</PNAFailureMessagesReceivedCount>
  <PNAFailureMessagesSentCount>0</PNAFailureMessagesSentCount>
  <PURMessagesReceivedCount>0</PURMessagesReceivedCount>
  <PURMessagesSentCount>0</PURMessagesSentCount>
  <PUASuccessMessagesReceivedCount>0</PUASuccessMessagesReceivedCount>
  <PUASuccessMessagesSentCount>0</PUASuccessMessagesSentCount>
  <PUAFailureMessagesReceivedCount>0</PUAFailureMessagesReceivedCount>
  <PUAFailureMessagesSentCount>0</PUAFailureMessagesSentCount>
  <SNRMessagesReceivedCount>0</SNRMessagesReceivedCount>
  <SNRMessagesSentCount>0</SNRMessagesSentCount>
  <SNASuccessMessagesReceivedCount>0</SNASuccessMessagesReceivedCount>
  <SNASuccessMessagesSentCount>0</SNASuccessMessagesSentCount>
  <SNAFailureMessagesReceivedCount>0</SNAFailureMessagesReceivedCount>
  <SNAFailureMessagesSentCount>0</SNAFailureMessagesSentCount>
  <ActiveSessionsCount>0</ActiveSessionsCount>
  <MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
</Sample>
</DiameterShPeerStats>
</Statistics>
```

```
<?xml version="1.0" ?>
- <Statistics>
- <DiameterShPeerStats>
- <Sample>
  <StartTime>2010-05-10T21:00:00Z</StartTime>
  <EndTime>2010-05-10T21:15:00Z</EndTime>
  <PolicyServer>mpe6-31</PolicyServer>
  <Name>hss.test.com</Name>
  <NeId />
  <NetworkElementType />
  <NetworkElementSubType />
  <ConnectTime />
```

```

<DisconnectTime />
<ConnectAddress />
<ConnectPort />
<MessagesInCount>0</MessagesInCount>
<MessagesOutCount>0</MessagesOutCount>
<UDRMessagesReceivedCount>0</UDRMessagesReceivedCount>
<UDRMessagesSentCount>0</UDRMessagesSentCount>
<UDASuccessMessagesReceivedCount>0</UDASuccessMessagesReceivedCount>
<UDASuccessMessagesSentCount>0</UDASuccessMessagesSentCount>
<UDAFailureMessagesReceivedCount>0</UDAFailureMessagesReceivedCount>
<UDAFailureMessagesSentCount>0</UDAFailureMessagesSentCount>
<PNRMessagesReceivedCount>0</PNRMessagesReceivedCount>
<PNRMessagesSentCount>0</PNRMessagesSentCount>
<PNASuccessMessagesReceivedCount>0</PNASuccessMessagesReceivedCount>
<PNASuccessMessagesSentCount>0</PNASuccessMessagesSentCount>
<PNAFailureMessagesReceivedCount>0</PNAFailureMessagesReceivedCount>
<PNAFailureMessagesSentCount>0</PNAFailureMessagesSentCount>
<PURMessagesReceivedCount>0</PURMessagesReceivedCount>
<PURMessagesSentCount>0</PURMessagesSentCount>
<PUASuccessMessagesReceivedCount>0</PUASuccessMessagesReceivedCount>
<PUASuccessMessagesSentCount>0</PUASuccessMessagesSentCount>
<PUAFailureMessagesReceivedCount>0</PUAFailureMessagesReceivedCount>
<PUAFailureMessagesSentCount>0</PUAFailureMessagesSentCount>
<SNRMessagesReceivedCount>0</SNRMessagesReceivedCount>
<SNRMessagesSentCount>0</SNRMessagesSentCount>
<SNASuccessMessagesReceivedCount>0</SNASuccessMessagesReceivedCount>
<SNASuccessMessagesSentCount>0</SNASuccessMessagesSentCount>
<SNAFailureMessagesReceivedCount>0</SNAFailureMessagesReceivedCount>
<SNAFailureMessagesSentCount>0</SNAFailureMessagesSentCount>
<ActiveSessionsCount>0</ActiveSessionsCount>
<MaximumActiveSessionsCount>0</MaximumActiveSessionsCount>
</Sample>
</DiameterShPeerStats>
</Statistics>

```

KPI Statistics

The following examples show the request and response that are defined in the XSDs for the KpiStats tag.

Request

This request follows the **QueryOmStats > KpiStats** tag defined in the XSDs.

The following is an example of this request.

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2001-12-31T12:00:00</StartTime>
  <EndTime>2001-12-31T12:15:00</EndTime>
  <KpiStats>
  </KpiStats>
</QueryOmStats>

```

Response

The response to this request follows the **Statistics > KpiStats** tag defined in the XSDs. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Statistics>
  <KpiStats>
    <Sample>
      <StartTime>2013-06-27T07:45:00Z</StartTime>
      <EndTime>2013-06-27T07:55:01Z</EndTime>
      <PolicyServer>mpe57</PolicyServer>
      <IsComplete>>false</IsComplete>
      <TransactionStartCount>0</TransactionStartCount>
      <TransactionEndCount>0</TransactionEndCount>
      <MessagesInCount>0</MessagesInCount>
      <MessagesOutCount>0</MessagesOutCount>
      <CurrentTransactionsPerSecond>0</CurrentTransactionsPerSecond>
      <MaxTransactionsPerSecond>0</MaxTransactionsPerSecond>
      <CurrentTPSPercentageOfCapacity>0</CurrentTPSPercentageOfCapacity>
      <MaxTPSPercentageOfCapacity>0</MaxTPSPercentageOfCapacity>
      <CurrentRXTransactionsPerSecond>0</CurrentRXTransactionsPerSecond>
      <CurrentPCMMTransactionsPerSecond>0</CurrentPCMMTransactionsPerSecond>
      <MaxPCMMTransactionsPerSecond>0</MaxPCMMTransactionsPerSecond>
      <CurrentPCMMTpsPercentageOfCapacity>0</CurrentPCMMTpsPercentageOfCapacity>
      <MaxPCMMTpsPercentageOfCapacity>0</MaxPCMMTpsPercentageOfCapacity>
      <MaxRXTransactionsPerSecond>0</MaxRXTransactionsPerSecond>
      <CurrentRXTpsPercentageOfCapacity>0</CurrentRXTpsPercentageOfCapacity>
      <MaxRXTpsPercentageOfCapacity>0</MaxRXTpsPercentageOfCapacity>
      <LoadSheddingStatus>0</LoadSheddingStatus>
      <LoadSheddingEfficiency>0</LoadSheddingEfficiency>
      <LoadSheddingDistressCount>0</LoadSheddingDistressCount>
      <CurrentSessionCount>0</CurrentSessionCount>
      <MaxSessionCount>0</MaxSessionCount>
      <CurrentSessionPercentageOfCapacity>0</CurrentSessionPercentageOfCapacity>
      <MaxSessionPercentageOfCapacity>0</MaxSessionPercentageOfCapacity>
      <CurrentMPEConnectionCount>0</CurrentMPEConnectionCount>
      <ConfiguredMPEConnectionCount>0</ConfiguredMPEConnectionCount>
      <CurrentConnectedNECount>0</CurrentConnectedNECount>
      <ConfiguredNECount>1</ConfiguredNECount>
      <CurrentProtocolErrorSentCount>0</CurrentProtocolErrorSentCount>
      <CurrentProtocolErrorReceivedCount>0</CurrentProtocolErrorReceivedCount>
      <ServerACPUUtilizationPercentage>14</ServerACPUUtilizationPercentage>
      <ServerAMemoryUtilizationPercentage>78</ServerAMemoryUtilizationPercentage>
      <ServerADiskUtilizationPercentage>2</ServerADiskUtilizationPercentage>
      <ServerABladeFailureCount>6</ServerABladeFailureCount>
      <ServerAUpTimeMillis>16208000</ServerAUpTimeMillis>
      <ServerABladeStatus>12</ServerABladeStatus>
      <ServerBCPUUtilizationPercentage>0</ServerBCPUUtilizationPercentage>
      <ServerBMemoryUtilizationPercentage>0</ServerBMemoryUtilizationPercentage>
      <ServerBDiskUtilizationPercentage>0</ServerBDiskUtilizationPercentage>
      <ServerBBladeFailureCount>0</ServerBBladeFailureCount>
      <ServerBUpTimeMillis>0</ServerBUpTimeMillis>
      <ServerBBladeStatus>0</ServerBBladeStatus>
      <ServerCCPUUtilizationPercentage>0</ServerCCPUUtilizationPercentage>
      <ServerCMemoryUtilizationPercentage>0</ServerCMemoryUtilizationPercentage>
      <ServerCDiskUtilizationPercentage>0</ServerCDiskUtilizationPercentage>
      <ServerCBladeFailureCount>0</ServerCBladeFailureCount>
      <ServerCUpTimeMillis>0</ServerCUpTimeMillis>
      <ServerCBladeStatus>0</ServerCBladeStatus>
    </Sample>
  </KpiStats>
</Statistics>
```

```
</KpiStats>
</Statistics>
```

KPI Interval Statistics

The following examples show the request and response that are defined in the XSDs for the IntervalStats tag.

Request

This request follows the **QueryOmStats > IntervalStats** tag defined in the XSDs.

The following is an example of this request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2001-12-31T12:00:00</StartTime>
  <EndTime>2001-12-31T12:15:00</EndTime>
  <IntervalStats>
  </IntervalStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > IntervalStats** tag defined in the XSDs.

- Interval Max Active Session Count — Maximum session count for AF and PCMM.
- Interval Max PCMM TransactionsPerSecond — The peak of PCMM transaction rate.
- Interval Max RX TransactionsPerSecond — The peak of RX transaction rate.

The response to this request follows the **Statistics > IntervalStats** tag defined in the XSDs. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Statistics>
  <IntervalStats>
    <Sample>
      <StartTime>2013-06-27T07:55:01Z</StartTime>
      <EndTime>2013-06-27T08:00:00Z</EndTime>
      <PolicyServer>mpe57</PolicyServer>
      <IntervalStartTime>2013-06-27T07:55:01Z</IntervalStartTime>
      <IntervalEndTime>2013-06-27T08:00:00Z</IntervalEndTime>
      <ConfiguredLength>300</ConfiguredLength>
      <ActualLength>300</ActualLength>
      <IsComplete>>true</IsComplete>
      <IntervalMaxTransactionsPerSecond>0</IntervalMaxTransactionsPerSecond>
      <IntervalMaxSessionCount>0</IntervalMaxSessionCount>
      <IntervalMaxPCMMTransactionsPerSecond>0</IntervalMaxPCMMTransactionsPerSecond>
      <IntervalMaxRXTransactionsPerSecond>0</IntervalMaxRXTransactionsPerSecond>
    </Sample>
  </IntervalStats>
</Statistics>
```

PCMM CMTS Statistics

The following examples show the request and response that are defined in the XSDs for the PcommCmtsStats tag.

Request

This request follows the **QueryOmStats > PcommCmtsStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system.

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <PcommCmtsStats></PcommCmtsStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > PcommCmtsStats** tag defined in the XSDs.

SessionCount — The number of session requests received.

SessionSuccessCount — The number of session requests successfully created.

SessionFailCount — The number of session request failures. For example, this could be due to a policy denying a request.

SessionProtocolFailCount — The number of session failures due to an invalid message or parameter. This count is incremented whenever the MPE determines that an incoming message from the AM has an invalid message and as a result the message has to be dropped by the MPE.

SessionPolicyFailCount — The number of session requests that trigger a policy. This count is maintained in the MPE, one per policy. When the condition of a policy triggers, the count for that policy is incremented. The value displayed is a total trigger count, (that is, the sum of this value for all the policies). For example, if the definition of a policy is defined as, “when the device usage is greater than 80% of capacity, reject message”. When the MPE executes this policy in response to a request, if the policy triggers (that is, if the MPE determines that the device usage is > 80%), the trigger count for that policy is incremented.

The following is an example XML response to a PcommCmtsStats tag request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Statistics>
  <PcommCmtsStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <SessionCount>100</SessionCount>
      <SessionSuccessCount>97</SessionSuccessCount>
      <SessionFailCount>3</SessionFailCount>
      <SessionProtocolFailCount>1</SessionProtocolFailCount>
      <SessionPolicyFailCount>2</SessionPolicyFailCount>
    </Sample>
```

```

    </PcmmCmtsStats>
  </Statistics>

```

PCMM Network Element Statistics

The following examples show the request and response that are defined in the XSDs for the `PcmmNetworkElementStats` tag.

Request

This request follows the `QueryOmStats > PcmmNetworkElementStats` tag defined in the XSDs.

The following is an example of a request for a single PCMM network element using the `Name` parameter:

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <PcmmNetworkElementStats>
    <Name>Router1</Name>
  </PcmmNetworkElementStats>
</QueryOmStats>

```

The following is an example of a request for multiple PCMM network elements using the `Name(s)` and `NeId(s)` parameters. This example returns statistics for three different network elements:

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <PcmmNetworkElementStats>
    <Name>Router1</Name>
    <Name>Router2</Name>
    <Name>Router3</Name>
  </PcmmNetworkElementStats>
</QueryOmStats>

```

The following is an example of a request for all PCMM network elements in the system. This example returns statistics for each PCMM network element:

```

<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <PcmmNetworkElementStats></PcmmNetworkElementStats>
</QueryOmStats>

```

Response

The response to this request follows the `Statistics > PcmmNetworkElementStats` tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Sample groups are ordered by policy server, network element, available interface, and then by time.

- Name — Unique name identifying the network element for the following statistics.
 - NeId — Optional identifier field for the network element.
 - InterfaceName — Identifying field for the network element's Interface.
 - SessionCount — Current active sessions for that network element or Interface. This is a non-cumulative value and is displayed as an absolute.
 - SessionSuccessCount — Successful sessions.
 - SessionFailCount — Session failures.
 - Capacity — The currently defined, maximum capacity for this network element or Interface. This is a static absolute value defined in the CMP for that object.

The following is an example XML response to a multiple network element request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <PcmmNetworkElementStats>
    <Sample>
      <StartTime>2006-10-12T11:18:30Z</StartTime>
      <EndTime>2006-10-12T11:19:20Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <Name>Router1</Name>
      <NeId>12345</NeId>
      <SessionCount>3</SessionCount>
      <SessionSuccessCount>3</SessionSuccessCount>
      <SessionFailCount>0</SessionFailCount>
      <Capacity>50000000</Capacity>
    </Sample>
    <Sample>
      <StartTime>2006-10-12T11:19:20Z</StartTime>
      <EndTime>2006-10-12T11:20:10Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <Name>Router1</Name>
      <NeId>12345</NeId>
      <SessionCount>0</SessionCount>
      <SessionSuccessCount>0</SessionSuccessCount>
      <SessionFailCount>0</SessionFailCount>
      <Capacity>50000000</Capacity>
    </Sample>
  </PcmmNetworkElementStats>
</Statistics>
```

DQos CMTS Statistics

The following examples show the request and response that are defined in the XSDs for the DqosCmtsStats tag.

Request

This request follows the **QueryOmStats > DqosCmtsStats** tag defined in the XSDs.

The following is an example of a request for all policy servers in the system.

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:40:00Z</EndTime>
  <DqosCmtsStats/>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics > DqosCmtsStats** tag defined in the XSDs.

SessionCount — The number of session requests received.

SessionSuccessCount — The number of session requests successfully created.

SessionFailCount — The number of session request failures. For example, this could be due to a policy denying a request.

SessionProtocolFailCount — The number of session failures due to an invalid message or parameter. This count is incremented whenever the MPE determines that an incoming message from the AM has an invalid message and as a result, the message has to be dropped by the MPE.

SessionPolicyFailCount — The number of session requests that trigger a policy. This count is maintained in the MPE, one per policy. When the condition of a policy triggers, the count for that policy is incremented. The value displayed is a total trigger count, (that is, the sum of this value for all the policies). For example, if the definition of a policy is defined as, “when the device usage is greater than 80% of capacity, reject message”. When the MPE executes this policy in response to a request, if the policy triggers (that is, if the MPE determines that the device usage is > 80%), the trigger count for that policy is incremented.

The following is an example XML response to a DqosCmtsStats tag request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <DqosCmtsStats>
    <Sample>
      <StartTime>2006-10-26T14:30:22Z</StartTime>
      <EndTime>2006-10-26T14:35:11Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <SessionCount>100</SessionCount>
      <SessionSuccessCount>97</SessionSuccessCount>
      <SessionFailCount>3</SessionFailCount>
      <SessionProtocolFailCount>1</SessionProtocolFailCount>
      <SessionPolicyFailCount>2</SessionPolicyFailCount>
    </Sample>
  </DqosCmtsStats>
</Statistics>
```

DQos Network Element Statistics

The following examples show the request and response that are defined in the XSDs for the DqosNetworkElementStats tag.

Request

This request follows the **QueryOmStats** > **DqosNetworkElementStats** tag defined in the XSDs.

The following is an example of a request for a single DQoS network element using the Name parameter:

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryOmStats>
  <StartTime>2006-10-26T14:30:00Z</StartTime>
  <EndTime>2006-10-26T14:35:00Z</EndTime>
  <DqosNetworkElementStats>
    <Name>Router1</Name>
  </DqosNetworkElementStats>
</QueryOmStats>
```

The following is an example of a request for all DQoS network elements in the system. This example returns statistics for each DQoS network element:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryOmStats>
  <StartTime>2006-10-12T11:15:00Z</StartTime>
  <EndTime>2006-10-12T11:25:00Z</EndTime>
  <DqosNetworkElementStats></DqosNetworkElementStats>
</QueryOmStats>
```

Response

The response to this request follows the **Statistics** > **DqosNetworkElementStats** tag defined in the XSDs.

Both Name and NeId are returned along with the statistics and actual recorded times for those statistics. Sample groups are ordered by policy server, network element, available interface, and then by time.

- Name — Unique name identifying the network element for the following statistics.
 - NeId — Optional identifier field for the network element.
 - InterfaceName — Identifying field for the network element's Interface.
 - SessionCount — Current active sessions for that network element or Interface. This is a non-cumulative value and is displayed as an absolute.
 - SessionSuccessCount — Successful sessions.
 - SessionFailCount — Session failures.
 - Capacity — The currently defined, maximum capacity for this network element or Interface. This is a static absolute value defined in the CMP for that object.

The following is an example XML response to a multiple network element request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <DqosNetworkElementStats>
    <Sample>
      <StartTime>2006-10-12T11:18:30Z</StartTime>
      <EndTime>2006-10-12T11:19:20Z</EndTime>
      <PolicyServer>Atlanta105</PolicyServer>
      <Name>Router1</Name>
      <NeId>12345</NeId>
      <SessionCount>3</SessionCount>
```

```

        <SessionSuccessCount>3</SessionSuccessCount>
        <SessionFailCount>0</SessionFailCount>
        <Capacity>50000000</Capacity>
    </Sample>
    <Sample>
        <StartTime>2006-10-12T11:19:20Z</StartTime>
        <EndTime>2006-10-12T11:20:10Z</EndTime>
        <PolicyServer>Atlanta105</PolicyServer>
        <Name>Router1</Name>
        <NeId>12345</NeId>
        <SessionCount>0</SessionCount>
        <SessionSuccessCount>0</SessionSuccessCount>
        <SessionFailCount>0</SessionFailCount>
        <Capacity>50000000</Capacity>
    </Sample>
</DqosNetworkElementStats>
</Statistics>

```

Connection Error Statistics

The following examples show the request and response that are defined in the XSDs for the `ConnectionErrorStats` tag.

Request

This request follows the `QueryOmStats > ConnectionErrorStats` tag defined in the XSDs.

The following is an example of a request for connection error statistics from an NE or AM within the system.

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2011-12-21T00:40:00Z</StartTime>
    <EndTime>2012-12-29T01:00:00Z</EndTime>
    <ConnectionErrorStats>
      </ConnectionErrorStats>
    </QueryOmStats>
  </XmlInterfaceRequest>

```

Response

The response to this request follows the `Statistics > ConnectionErrorStats` tag defined in the XSDs.

The following is an example of the `ConnectionErrorStats` tag:

```

<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <ConnectionErrorStats>
    <Sample>
      <StartTime>2012-10-25T12:15:00Z</StartTime>
      <EndTime>2012-10-25T12:30:00Z</EndTime>
      <PolicyServer>mpell16</PolicyServer>
      <IsComplete>true</IsComplete>
      <Name>123456</Name>
    </Sample>
  </ConnectionErrorStats>
</Statistics>

```

```

    <NeId>ne71</NeId>
    <PcmmInvalidsubscriberReceived>0</PcmmInvalidsubscriberReceived>
    <PcmmInvalidsubscriberSent>1</PcmmInvalidsubscriberSent>
    <PcmmUnknowngateidReceived>0</PcmmUnknowngateidReceived>
    <PcmmUnknowngateidSent>3</PcmmUnknowngateidSent>
    <PcmmOtherReceived>0</PcmmOtherReceived>
    <PcmmOtherSent>10</PcmmOtherSent>
  </Sample>
</ConnectionErrorStats>
</Statistics>

```

Individual statistics defined for connection errors are:

- **PCMM_INSUFFICIENT_RESOURCE**: The server had no sufficient resource to handle it.
- **PCMM_UNKNOWN_GATEID**: The request contained an unknown Gate-Id.
- **PCMM_MISSING_OBJECT**: The request missed some objects.
- **PCMM_INVALID_OBJECT**: The request contained invalid parameters.
- **PCMM_VOLUME_EXCEEDED**: The request exceeded volume definition.
- **PCMM_TIME_EXCEEDED**: The request exceeded the time definition.
- **PCMM_CLASS_EXCEEDED**: The request exceeded the class definition.
- **PCMM_UNDEFINED_CLASS**: The request undefined class:
- **PCMM_INCOMPATIBLE_ENVELOPE**: The request was incompatible envelope.
- **PCMM_INVALID_SUBSCRIBER**: The request had invalid subscriber information.
- **PCMM_UNAUTHORIZED_AMID**: The request included a unauthorized AM-Id.
- **PCMM_NUMBLASS_NOT_SUPPORTED**: There are some settings which are not supported.
- **PCMM_POLICY_EXCEPTION**: It encountered the policy exception.
- **PCMM_INVALID_FIELD_VALUE**: The request included some invalid filed values.
- **PCMM_TRANSPORT_ERROR**: It encountered transport errors.
- **PCMM_UNKNOWN_GATECMD**: The request contained an unknown Gate-CMD.
- **PCMM_DOCSIS10CM**: The request contained a DOCSIS 1.0 CM error.
- **PCMM_NUMSIDS_EXCEEDED_IN_CM**: The request exceeded numbers of SIDs in CM.
- **PCMM_NUMSIDS_EXCEEDED_IN_CMTS**: The request exceeded the numbers of SIDs in CMTS.
- **PCMM_UNAUTHORIZED_PSID**: The request had an unauthorized PS-ID.
- **PCMM_NOSTATE_FOR_PDP**: The request did not have a state for PDP.
- **PCMM_UNSUPPORTED_SYNCH**: The request didn't support synchronization.
- **PCMM_DATA_INCOMPLETE**: The request encountered incomplete data.
- **PCMM_UPSTREAMDROP_UNSUPPORTED**: The request didn't support an upstream-drop.
- **PCMM_OTHER**: Some other errors except above definition, for example, network error, socket error or program error, etc.

Protocol Error Statistics

The following examples show the request and response that are defined in the XSDs for the ProtocolErrorStats tag.

Request

This request follows the **QueryOmStats > ProtocolErrorStats** tag defined in the XSDs.

The following is an example of a request for MPE protocol errors in the system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryOmStats>
    <StartTime>2011-12-21T00:40:00Z</StartTime>
    <EndTime>2012-12-29T01:00:00Z</EndTime>
    <ProtocolErrorStats>
    </ProtocolErrorStats>
  </QueryOmStats>
</XmlInterfaceRequest>
```

Response

The response to this request follows the **Statistics > ProtocolErrorStats** tag defined in the XSDs.

The following is an example of the ProtocolErrorStats tag:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Statistics>
  <ProtocolErrorStats>
    <Sample>
      <StartTime>2012-10-25T12:15:00Z</StartTime>
      <EndTime>2012-10-25T12:30:00Z</EndTime>
      <PolicyServer>mpell16</PolicyServer>
      <IsComplete>true</IsComplete>
      <PcmmInvalidsubscriberReceived>0</PcmmInvalidsubscriberReceived>
      <PcmmInvalidsubscriberSent>1</PcmmInvalidsubscriberSent>
      <PcmmUnknowngateidReceived>0</PcmmUnknowngateidReceived>
      <PcmmUnknowngateidSent>3</PcmmUnknowngateidSent>
      <PcmmOtherReceived>0</PcmmOtherReceived>
      <PcmmOtherSent>10</PcmmOtherSent>
    </Sample>
  </ProtocolErrorStats>
</Statistics>
```

Individual statistics are defined for protocol errors are:

- **PCMM_INSUFFICIENT_RESOURCE**: The server had no sufficient resource to handle it.
- **PCMM_UNKNOWN_GATEID**: The request contained an unknown Gate-Id.
- **PCMM_MISSING_OBJECT**: The request missed some objects.
- **PCMM_INVALID_OBJECT**: The request contained invalid parameters.
- **PCMM_VOLUME_EXCEEDED**: The request exceeded volume definition.
- **PCMM_TIME_EXCEEDED**: The request exceeded the time definition.
- **PCMM_CLASS_EXCEEDED**: The request exceeded the class definition.
- **PCMM_UNDEFINED_CLASS**: The request undefined class:
- **PCMM_INCOMPATIBLE_ENVELOPE**: The request was incompatible envelope.
- **PCMM_INVALID_SUBSCRIBER**: The request had invalid subscriber information.
- **PCMM_UNAUTHORIZED_AMID**: The request included a unauthorized AM-Id.
- **PCMM_NUMBLASS_NOT_SUPPORTED**: There are some settings which are not supported.
- **PCMM_POLICY_EXCEPTION**: It encountered the policy exception.
- **PCMM_INVALID_FIELD_VALUE**: The request included some invalid filed values.
- **PCMM_TRANSPORT_ERROR**: It encountered transport errors.
- **PCMM_UNKNOWN_GATECMD**: The request contained an unknown Gate-CMD.

Operational Measurement Requests

- **PCMM_DOCSIS10CM:** The request contained a DOCSIS 1.0 CM error.
- **PCMM_NUMSIDS_EXCEEDED_IN_CM:** The request exceeded numbers of SIDs in CM.
- **PCMM_NUMSIDS_EXCEEDED_IN_CMTS:** The request exceeded the numbers of SIDs in CMTS.
- **PCMM_UNAUTHORIZED_PSID:** The request had an unauthorized PS-ID.
- **PCMM_NOSTATE_FOR_PDP:** The request did not have a state for PDP.
- **PCMM_UNSUPPORTED_SYNCH:** The request didn't support synchronization.
- **PCMM_DATA_INCOMPLETE:** The request encountered incomplete data.
- **PCMM_UPSTREAMDROP_UNSUPPORTED:** The request didn't support an upstream-drop.
- **PCMM_OTHER:** Some other errors except above definition, for example, network error, socket error or program error, etc.