

**Oracle® Communications
Policy Management**

OSSI XML Interface Definitions Reference Guide

Release 10.4

910-6897-001 Revision B

March 2014

Oracle® Communications OSSI XML Interface Definitions Reference Guide, Release 10.4

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.....	7
Introduction.....	8
How This Guide is Organized.....	8
Scope and Audience.....	8
Related Publications.....	8
Chapter 2: Schema Definitions and Request Specifications.....	10
Schema Definitions.....	11
Request Specification.....	11
Chapter 3: Common Responses and Commands.....	12
Common Responses.....	13
Example XML Response for a Successful Operation.....	13
Example XML Response for an Operation that Fails at the System Level.....	13
Example XML Response for an Operation that Includes an Application-level Failure.....	13
Result Codes.....	14
Common Commands.....	14
Get Version.....	15
Distribute Updates.....	23
OSSI Response Failure Reason Code.....	24
Chapter 4: Topology Interface.....	26
Topology Interface Requests.....	27
Network Elements.....	28
Add Network Element.....	28
Update Network Element.....	29
Add Network Element to Group.....	30
Remove Network Element from Group.....	31
Delete Network Element.....	31
Query Network Elements.....	32
Paths.....	34

Add Path.....	34
Update Path.....	35
Delete Path.....	36
Query Path.....	37
Applications.....	38
Add Application.....	38
Update Application.....	39
Delete Application.....	39
Query Application.....	40
Chapter 5: Subscriber Interface.....	41
Accounts.....	42
Overview.....	42
Add Account.....	42
Update Account.....	43
Delete Account.....	43
Query Account.....	44
Tiers.....	45
Add Tier.....	45
Update Tier.....	46
Delete Tier.....	46
Query Tier.....	47
Chapter 6: Operational Measurements Interface Overview.....	48
OM Statistics Scheduled Task.....	49
OM Statistics Requests.....	49
Attributes and Child Tags.....	49
Recorded Timestamp and Request Time Range.....	51
OM Statistics Response Format.....	51
Interval Statistics.....	51
Absolute versus Delta Values.....	52
Empty Data Set.....	52
Timezones and Start/End Times.....	52
Counter Reset and Failover.....	53
Comparisons Between the CMP GUI and OM Statistics.....	54
Chapter 7: Operational Measurement Requests.....	56
Operational Measurements Requests Overview.....	57
Topology Update Statistics.....	57

Subscriber Update Statistics.....	58
Policy Server Statistics.....	59
Message Processing Statistics.....	60
Network Element Statistics.....	61
VoD Server SD/HD Session Statistics.....	64
Connected Network Elements Statistics.....	67
Diameter Synchronized Statistics.....	67
Diameter Policy Charging Enforcement Function Statistics.....	68
Diameter Policy Charging Enforcement Function Peer Statistics.....	70

Chapter 8: Identity Management (IDM).....72

Identity Management Overview.....	73
Add a User.....	73
Update a User.....	79
Query a User.....	84
Delete a User.....	86
Query User Role.....	87
Query User Scope.....	93

List of Tables

Table 1: OSSI Result Codes.....14

Table 2: Possible Version Error Codes.....15

Table 3: AddSysAdminUser Operation Parameters.....74

Table 4: AddSysAdminUser Operation Errors.....74

Table 5: UpdateSysAdminUser Operation Parameters.....80

Table 6: UpdateSysAdminUser Operation Errors.....80

Table 7: QuerySysAdminUser Operation Parameters.....85

Table 8: Output Fields for QuerySysAdminUser Operation85

Table 9: DeleteSysAdminUser Operation Parameters.....86

Table 10: DeleteSysAdminUser Operation Errors.....86

Table 11: QueryRole Operation Parameters.....88

Table 12: Output Fields for QueryRole Operation88

Table 13: QueryScope Operation Parameters.....93

Table 14: Output Fields for QueryScope Operation93

Chapter 1

About This Guide

Topics:

- [Introduction.....8](#)
- [How This Guide is Organized.....8](#)
- [Scope and Audience.....8](#)
- [Related Publications.....8](#)

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.

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.
- *Identity Management (IDM)* describes the interface that allows the CMP to configure user names, passwords and roles using the OSSI interface.

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.

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*

Chapter 2

Schema Definitions and Request Specifications

Topics:

- [Schema Definitions.....11](#)
- [Request Specification.....11](#)

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 `XmlInterfaceRequest` tag and the associated sub-element type definition. This `xsd` is used to validate add/update requests.
- `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 such as `AddAccount` and `UpdateAccount`.
- `OssiXmlOm.xsd` — defines the schema for the OM interface and the OSSI OM stats query and response type definition.
- `OssiXmlRuntimeResponse.xsd` - defines the runtime response type such as `SubscriberLog`, `SubscriberRealtimeStats` and `ResponseType`.
- `OssiXmlUserMgr.xsd` - defines the User, Role-related type (such as: `AddRole` and `AddSysAdminUser`).

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.....13*
- *Result Codes.....14*
- *Common Commands.....14*
- *OSSI Response Failure Reason Code.....24*

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 1: 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

- Query OM Stats Setting — allows users to verify the persistent-interval settings for Reset Configuration and Collection Interval.

▶ Deleted Serving Gateway/MCC-MNC from list ◀

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 type="XmlInterfaceResponse">
    <Success count="1">2.2.14</Success>
    <Failure count="0"></Failure>
  </Command>
</Response>
```

Version Errors

This table lists possible errors if a version does not exist in the system.

Table 2: Possible Version Error Codes

Error Code	Error Type	Error Message
10100	ERROR_INVALID_JAXB_TYPE	\nImported object invalid
10103	ERROR_FILESIZE_TOO_LARGE	Import file exceeds max size of 20M.
10110	ERROR_PARSING_XML	Error parsing import file: {0}
10111	ERROR_UNKNOWN_EXCEPTION	Exception: {0}
10200	ERROR_DUPLICATE_RESOURCELIMIT	\nResource Limit value {0} contains duplicate type and direction for {1}
10201	ERROR_RESOURCE_DIRECTIONLIMIT	\nResource direction value {0} is required in [0,1] for {1}
10202	ERROR_RESOURCECETYPELIMIT	\nResource type value {0} is required in [0,1] for {1}

Common Responses and Commands

10207	ERROR_APPLYTO_RC_HOSTNAME	Unable to apply updates to Policy Server(s): {0}, hostname: {1}.
10208	ERROR_SUBSCRIBER_TIMESTAMP_MISMATCH	\nWARNING: Subscriber Account mismatch was detected between Manager and Policy Server: {0}. Subscriber accounts may need to be reconciled.
10211	ERROR_CREATE_ERROR	\n{0} creation/update returned an error. {0}: {1}
10212	ERROR_CREATE_ERROR_DETAIL	\n{0} creation encountered an exception: {1}
10213	ERROR_INVALID_RESOURCELIMIT	\nResource Limit value '{0}' for '{1}' , is invalid.
10214	ERROR_RESOURCELIMIT_TOOLARGE	\nResource Limit value '{0}' for '{1}' , exceeds max of {2}.
11100	ERROR_NE_NAME_NOT_EXIST	\nSpecified Network Element does not exist: '{0}'
11101	ERROR_INVALID_NE_NAME	\nNetwork Element includes an invalid Name: '{0}'
11102	ERROR_NE_NAME_TOO_LONG	\nNetwork Element:{0} exceeds max length of 250 characters, actually length is {1}
11103	ERROR_DUPLICATE_NEID	\nNetwork Element contains duplicate ID: '{0}'
11104	ERROR_DUPLICATE_NENAME	\nNetwork Element contains duplicate Name: '{0}'
11105	ERROR_INVALID_NE_TYPE	\nNetwork Element includes an invalid NetworkElement Type and/or SubType. Network Element: {0}
11106	ERROR_DESC_TOO_LONG	\nNetwork Element Description exceeds max length of 250 characters: '{0}'
11108	ERROR_NE_TYPE_NOT_ALLOWED_TO_UPDATE	\nNetwork Element update of NetworkElement Type and/or SubType is not allowed. Network Element: {0}
11109	ERROR_NE_REQUIRE_ID	\nNetwork Element must include an ID: {0}
11110	ERROR_INVALID_NEID	\nNetwork Element contains invalid ID: '{0}'

Common Responses and Commands

11111	ERROR_NEID_TOO_LONG	\nNeId exceeds max length of 250 characters. Network Element: '{0}'
11113	ERROR_MAX_NE_CAPACITY	\nNetwork Element Capacity exceeds maximum allowed value of {0}. Network Element: {1}
11114	ERROR_FAIL_ADD_TO_NEGROUP	Not used in Wireline mode
11115	ERROR_FAIL_CREATE_NE	\nNetwork Element creation(update) returned an error. Network Element: {0}
11116	ERROR_FETCH_PARENT_NE	\nNetwork Element returned an error retrieving Parent Network Elements. {0}
11118	ERROR_NE_LINK_END1	\nNetwork Element Link returned an error retrieving Network Element NE1. Link: {0}
11119	ERROR_DIAMETER_ID_NOT_UNIQUE	\nThe following Diameter Identities are already taken: {0}. You must enter unique Diameter Identities.
11120	ERROR_NE_LINK_END2	\nNetwork Element Link returned an error retrieving Network Element NE2. Link: {0}
11121	ERROR_NE_LINK_INTERFACE_END1	\nNetwork Element Link returned an error retrieving Network Element Interface EndPoint1. Link: {0}
11122	ERROR_LINK_INTERFACE_END2	\nNetwork Element Link returned an error retrieving Network Element Interface EndPoint2. Link: {0}
11123	ERROR_NE_LINK_CREATION	\nNetwork Element Link creation returned error. Link: {0}
11124	ERROR_NE_LINK_UPDATE	\nNetwork Element Link update returned error. Link: {0}
11125	ERROR_NE_LINK_DUPLICATE	\nDuplicate found. Network Element Link already exists. Link: {0}
11126	ERROR_SERVICE_CLASS	Not used in Wireline Mode
11129	ERROR_INVALID_LEARNED_SUBNET_IP	\nNetwork Element Learned Subnet includes an invalid IP. Network Element: {0}
11130	ERROR_INVALID_LEARNED_SUBNET_MASK	\nNetwork Element Learned Subnet includes an invalid Mask. Network Element: {0}

Common Responses and Commands

11131	ERROR_INVALID_OSSI_SUBNET_IP	\nNetwork Element Ossi Subnet includes an invalid IP. Network Element: {0}
11132	ERROR_INVALID_OSSI_SUBNET_MASK	\nNetwork Element Ossi Subnet includes an invalid Mask. Network Element: {0}
11133	ERROR_INVALID_NE_INTERFACE_NAME	\nNetwork Element Interface includes an invalid Name. Network Element: {0}
11134	ERROR_INVALID_NE_INTERFACE_CAPACITY	\nNetwork Element Interface Capacity exceeds maximum allowed value of {0}. Network Element: {1}
11135	ERROR_DUPLICATE_NE_INTERFACE_NAME	\nNetwork Element Interface Name already exists: '{0}'
11136	ERROR_NE_INVALID_HOSTNAME	\nNetwork Element includes invalid Hostname: '{0}'
11137	ERROR_NE_DUPLICATE_HOSTNAME	\nNetwork Element includes duplicate Hostname: '{0}'
11138	ERROR_NE_INVALID_BACKUP_NAME	\nNetwork Element includes invalid backup Hostname: '{0}'
11139	ERROR_NE_DUPLICATE_BACKUP_HOSTNAME	\nNetwork Element includes duplicate backup Hostname: '{0}'
11140	ERROR_NE_BACKUP_HOSTNAME_NOT_SAME	\nThe backup host name / IP address cannot be the same as the primary one.
11141	ERROR_NE_CAPACITY_INVALID	\nCapacity value {0} is invalid. NetworkElement: {1}
11142	ERROR_NE_DELETE_NAME	Error deleting element Name: {0}
11143	ERROR_NE_DELETE_NAME_INVALID	Error deleting element, Name is not found in database: {0}
11144	ERROR_NE_DELETE_NEID	Error deleting element ID: {0}
11145	ERROR_NE_DELETE_NEID_INVALID	Error deleting element, NeId is not found in database: {0}
11200	ERROR_INVALID_NEGROUP_NAME	\nNetwork Element Group includes an invalid Name: '{0}'
11201	ERROR_MAX_NEGROUP_NAME	\nNetwork Element Group Name exceeds max length, group name: '{0}'
11202	ERROR_UPDATE_NEGROUP	\nNetwork Element Group update returned an error attempting to add the following: {0}

Common Responses and Commands

11203	ERROR_UPDATE_NEGROUP_INFO	\nError during update of Network Element Group {0}: '{1}'
11204	ERROR_INVALID_NENAME_ADD_TO_GROUP	\nNetwork Element Group update returned an error attempting to add an element without a Name. {0}
11205	ERROR_NEGROUP_REF_ITSELF	\nNetwork Element Group update returned an error attempting to add reference to itself. {0}
11206	ERROR_SUBGROUP_WITHOUT_PARENT	\nNetwork Element Group returned error attempting to create Subgroup without a parent: {0}
11207	ERROR_UPDATE_NEGROUP_RETRIEVE	\nNetwork Element Group update returned an error. Failure to retrieve the group: '{0}'
11208	ERROR_UPDATE_NEGROUP_ADDROOTGROUP	\nNetwork Element Group update returned an error attempting to add RootGroup {0}
11209	ERROR_UPDATE_NEGROUP_EXISTED	\nNetwork Element Group update returned an error attempting to add existing Group '{0}' in multiple places.
11210	ERROR_ADD_NE_TO_NEGROUP	\nNetwork Element: '{0}' could not be added to given NeGroupName: '{1}'.
11211	ERROR_NEGROUP_DESC_TOO_LONG	\nNetwork Element Group Description exceeds max length, group name: '{0}'
11212	ERROR_ADD_NEGROUP_NAME_ALL_NOTPERMIT	Addition of 'ALL' Group as a subgroup is not permitted.
11213	ERROR_ADD_NE_TO_NEGROUP_CREATEGROUP	Create NeGroup "{0}" returned an error.
11214	ERROR_NE_TO_NEGROUP_NAME_INVALID	Name is invalid, not found in database : {0}
11216	ERROR_NE_TO_NEGROUP_NEID_INVALID	NeId is invalid, not found in database : {0}
11217	ERROR_ADD_NE_TO_NEGROUP_ADDING_NEID	Error adding element ID: {0}
11219	ERROR_REMOVE_NE_FROM_GROUP_GROUPNAME_INVALID	Group Name is invalid, not found in database: {0}
11221	ERROR_REMOVE_NE_FROM_GROUP_REMOVING_NENAME	Error removing element Name: {0}
11223	ERROR_REMOVE_NE_FROM_GROUP_REMOVING_NEID	Error removing element ID: {0}

Common Responses and Commands

11224	ERROR_DELETE_NEGROUP_NAME_ALL_NOTPERMITTED	Delete of 'ALL' Group is not permitted.
11225	ERROR_DELETE_NEGROUP_NAME_INVALID	Group Name is invalid, not found in database: {0}
11226	ERROR_DELETE_NEGROUP_DELETING_ERROR	Error deleting group: {0}
11300	ERROR_MAX_PATH_NAME	\nPath Name exceeds max length. Path: '{0}'
11301	ERROR_MAX_PATH_DESCRIPTION	\nPath Description exceeds max length of 250 characters. Path: '{0}'
11302	ERROR_PATH_NOTEXISTED	\nSpecified Path does not exist: '{0}'
11303	ERROR_PATH_DELETE	Error deleting path Name: {0}
11304	ERROR_PATH_IPMASK_INVALID	Path source/destination includes an invalid IP/Mask. Path: '{0}'
11400	ERROR_APP_NAME_INVALID	\nInvalid Application Name for: '{0}'
11401	ERROR_APP_NAME_TOO_LONG	\nApplication Name exceeds max length for: '{0}'
11402	ERROR_APP_DESCRIPTION_TOO_LONG	\nApplication Description exceeds max length. Application: '{0}'
11403	ERROR_APP_INVALID_AMIDNUM	\nAmid includes an invalid AmidNum: '{0}' Application: '{1}'
11404	ERROR_APP_DUPLICATE_AMIDNUM	\nAmid includes a duplicate AmidNum: '{0}' Application: '{1}'
11405	ERROR_APP_INVALID_AFAPPIDSTR	\nAfappid includes an invalid AfappidStr: '{0}' Application: '{1}'
11406	ERROR_APP_INVALID_AMDIAMETERID	\nAmdiameterid includes an invalid AmdiameteridStr: '{0}' Application: '{1}'
11407	ERROR_APP_INVALID_AMCONNECTION	\nAmconnectionip includes an invalid AmconnectionipStr: '{0}' Application: '{1}'
11408	ERROR_APP_DUPLICATE_AMCONNECTIONIPSTR	\nConnection IP includes a duplicate value: '{0}' Application: '{1}'
11409	ERROR_APP_INVALID_SESSIONCLASSIDNUM	\nSessionClassId includes an invalid SessionClassIdNum: '{0}' Application: '{1}'
11410	ERROR_APP_DELETE	Error deleting application Name: {0}
11500	ERROR_ACCOUNT_NAME_EMPTY	\nAccount ID cannot be empty.

Common Responses and Commands

11501	ERROR_ACCOUNT_NAME_TOO_LONG	\nAccount ID exceeds max length. Account: '{0}'
11502	ERROR_ACCOUNT_SUBSCRIBERDATA_TOO_LONG	\nSubscriber Data exceeds max length for account: '{0}'
11503	ERROR_ACCOUNT_NENAME_TOO_LONG	\nAccount NetworkElementName exceeds max length for account: '{0}'
11504	ERROR_ACCOUNT_STATICIP_DECLARE_TOO_LONG	\nExceeded maximum (1) Static IP declarations allowed. Account: {0}
11505	ERROR_ACCOUNT_STATICIP_TOO_LONG	\nExceeded maximum (125) Static IP Addresses allowed. Account: {0}
11506	ERROR_ACCOUNT_STATICIP_MIN	\nStatic IP Count must be declared as an integer between 1-125. Account: {0}
11507	ERROR_ACCOUNT_INVALID_STATICIP	\nInvalid Static IP Address for Account: {0}
11508	ERROR_ACCOUNT_ADD_TIER_WITH_NO_NAME	\nAccount update returned an error attempting to add a Tier without a Name. {0}
11509	ERROR_ACCOUNT_NOT_EXIST	\nSpecified Account does not exist: '{0}'
11510	ERROR_ACCOUNT_STATICIP_REQUIRED	\nStatic IP Count field required. Account: {0}
11511	ERROR_ACCOUNT_ENTITLEMENT_NAME_NOT_EXIST	\nUpdate returned an error attempting to add an entitlement without a Name. {0}
11512	ERROR_ACCOUNT_DELETE	Error deleting account ID: {0}
11513	ERROR_ACCOUNT_MAX_SUBSCRIBERGROUPNAME	\nSubscriber Group for Account exceeds max length of 250 characters. Account: '{0}'
11514	ERROR_ACCOUNT_DUPLICATE_STATICIP	\nError on Account '{0}' attempting to import a duplicate Static Ip Address = '{1}'
11515	ERROR_ACCOUNT_NAME_INVALID	Account ID cannot contain wildcard characters ? or *
11600	ERROR_TIER_NAME_INVALID	\nInvalid Tier Name for: '{0}'
11601	ERROR_TIER_NAME_TOO_LONG	\nTier Name exceeds max length. Tier: '{0}'
11602	ERROR_TIER_DESCRIPTION_TOO_LONG	\nTier Description exceeds max length. Tier: '{0}'

Common Responses and Commands

11603	ERROR_TIER_WITHOUT_ENTITLEMENT	\nUpdate returned an error attempting to add an entitlement without a Name. {0}
11604	ERROR_TIER_DELETE	Error deleting tier Name: {0}
11605	ERROR_TIER_NAME_NOT_EXIST	\nSpecified Tier does not exist: '{0}'
11800	ERROR_UNKNOWN_VERSION	Unknown Version
11900	ERROR_SCOPE_NAME_INVALIDE	\nInvalid Scope Name for: '{0}'
11901	ERROR_SCOPE_NAME_TOO_LONG	\nScope Name exceeds max length for: '{0}'
11902	ERROR_SCOPE_DESCRIPTION_TOO_LONG	\nScope Description exceeds max length. Scope: '{0}'
11903	ERROR_SCOPE_CDNAMGROUP_LINK	\nScope and CdnAmGroup Link creation error. Scope Name: {0}
11904	ERROR_SCOPE_NEGROUP_LINK	\nScope and NEGroup Link creation error. Scope Name: {0}
11905	ERROR_SCOPE_PATHGROUP_LINK	\nScope and PathGroup Link creation error. Scope Name: {0}
11906	ERROR_SCOPE_RCGROUP_LINK	\nScope and RCGroup Link creation error. Scope Name: {0}
11907	ERROR_SCOPE_SIPAMGROUP_LINK	\nScope and SipAmGroup Link creation error. Scope Name: {0}
11908	ERROR_SCOPE_DELETE	Error deleting scope Name: {0}
11909	ERROR_SCOPE_NAME_NOT_EXIST	\nSpecified Scope does not exist: '{0}'
12000	ERROR_ROLE_DESCRIPTION_TOO_LONG	\nRole Description exceeds max length. Role: '{0}'
12001	ERROR_ROLE_PRIVILEGE_LINK	\nRole and privilege Link creation error. Privilege ID: {0}
12002	ERROR_ROLE_NAME_INVALID	\nInvalid Role Name for: '{0}'
12003	ERROR_ROLE_NAME_TOO_LONG	\nRole Name exceeds max length for: '{0}'
12004	ERROR_ROLE_DELETE	Error deleting role Name: {0}
12005	ERROR_ROLE_NAME_NOT_EXIST	\nSpecified Role does not exist: '{0}'
12100	ERROR_USER_DESCRIPTION_TOO_LONG	\nUser Description exceeds max length. User: '{0}'
12101	ERROR_USER_NOROLE_OR_NOSCOPE	\nUser: '{0}' must have an associated Role and Scope.
12102	ERROR_USER_INVALID_ROLE_NAME	\nInvalid Role Name for: '{0}'

12103	ERROR_USER_INVALID_SCOPE_NAME	\nInvalid Scope Name for: '{0}'
12104	ERROR_USER_ROLE_LINK	\nThe user: {0}'s associated role does not exist. Please create it first.
12105	ERROR_USER_SCOPE_LINK	\nThe user: {0}'s associated scope does not exist. Please create it first.
12106	ERROR_USER_INVALID_NAME	\nInvalid User Name for: '{0}'
12107	ERROR_USER_NAME_TOO_LONG	\nUser Name exceeds max length for: '{0}'
12108	ERROR_USER_DELETE	Error deleting User Name: {0}
12109	ERROR_USER_PWD_EMPTY	\nUser's Password should not be empty. User: '{0}'
12110	ERROR_USER_NAME_NOT_EXIST	\nSpecified User does not exist: '{0}'
12111	ERROR_USER_INVALID_LOCKEDSTATUS	\nThe value for LockedStatus is invalid. LockedStatus: '{0}'
12112	ERROR_USER_NAME_ILLEGAL_CHARACTERS	\nThe value for Name contains illegal characters: '{0}'
12114	ERROR_USER_INVALID_EXPIRATION_PERIOD	\nThe Expiration Period is invalid. Range: 7 - 365. (0 = never expires). User: '{0}'
12115	ERROR_USER_UPDATE_EXPIRATION_PERIOD_DISABLED	\nNot able to update Expiration Period when it's disabled. Change it to be Enabled in System Settings firstly. User: '{0}'
12113	ERROR_MATCHLIST_VALUE_TOO_LONG	\nThe value for Matchlist '{0}' should be not more than {1}.

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 Policy support personnel. 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>
```

OSSI Response Failure Reason Code

This section describes commands that are common to the add, update, and delete OSSI responses that are affected by the failure reason code.

Failure Reason Code in a Response attribute,

If, and only if, the request has the Version="2.2.14" the response will include the reason code. If Version equals any value less, or Version does not exist, the responses are the same as they were.

This is a request and response for the following example:

DeleteNetworkElement for No existing Network Element (NE) with Version number greater than or equal to version 2.2.14.

Request

This is a request using the Version attribute next to XmlInterfaceRequest. Placing the Version attribute in this position enables the user to view details on version error codes when the version number is greater than or equal to version 2.2.14.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest Version="2.2.14" >
  <DeleteNetworkElement>
    <Name>No-existing NE</Name>
  </DeleteNetworkElement>
</XmlInterfaceRequest>
```

Response

This is the response for DeleteNetworkElement:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response>
  <Result>0</Result>
  <Command>
    <Success count="0"/>
    <Failure count="1">
      <Operation code="6113" count="1">
        <ErrObject key="Name" value="No-existing NE">
          <Error code="11143">
            <ErrCode key="Name" value="No-existing NE"/>
          </Error>
        </ErrObject>
      </Operation>
    </Failure>
  </Command>
</Response>
```

This is a request and response for the following example:

DeleteNetworkElement for No existing Network Element (NE) with Version number less than 2.2.14 or if the version does not exist.

Request

This is a request without the Version attribute or Version number next to the XmlInterfaceRequest where the Version number is less than 2.2.14.

```
<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <DeleteNetworkElement>
    <Name>No-existing NE</Name>
  </DeleteNetworkElement>
</XmlInterfaceRequest>
```

Response

This is the response for DeleteNetworkElement:

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to delete 1 elements.
      Error deleting element, Name is not found in database: No-existing
      NE</Failure>
  </Command>
</Response>
```

Note: See [OSSl Response Failure Reason Code](#) for descriptions of possible errors.

Chapter 4

Topology Interface

Topics:

- *Topology Interface Requests.....27*
- *Network Elements.....28*
- *Paths.....34*
- *Applications.....38*

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 or B-RAS. The network elements can be organized into logical groups (NetworkGroup) that subdivide the network space. The following are operations that can be requested on Network Elements:
 - 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. The following are operations that can be requested on Paths:
 - 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. The following are operations that can be requested on Applications:
 - 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.

Network Elements

A network element represents a node in the network, such as a router or B-RAS. 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.

Note: See [Network Elements](#) for a complete description of Network Elements and Network Element Groups

Note: For a list of all the attributes that can be associated with a Network Element see: [Topology Interface](#)

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
-  (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 group:

```

<?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.

Note: See [Network Elements](#) for a complete description of Network Elements and Network Element Groups

Note: For a list of all the attributes that can be associated with a Network Element see: [Topology Interface](#)

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>

```

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>NE A</Name>
    <Name>NE B</Name>
  </QueryNetworkElement>
</XmlInterfaceRequest>

```

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



```

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

```



Response

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



```

<ConfigurationData version="2.1.6">
  <NetworkElement>
    <Name>vod-1</Name>
    <Description/>
    <HostName/>
    <BackupHostName/>
    <NeId>vod-1</NeId>
    <NetworkElementType>Server</NetworkElementType>
    <NeSubType>None</NeSubType>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
    <ManualSubnet>
      <IpAddress>88.0.0.0</IpAddress>
      <Mask>8</Mask>
    </ManualSubnet>
  </NetworkElement>
  <NetworkElement>
    <Name>erx-1</Name>
    <Description/>
    <HostName/>
    <BackupHostName/>
    <NeId>erx-1</NeId>
    <NetworkElementType>B-RAS</NetworkElementType>
    <NeSubType>ERX</NeSubType>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
    <ManualSubnet>
      <IpAddress>1.0.0.0</IpAddress>
      <Mask>8</Mask>
    </ManualSubnet>
    <NEInterface>
      <Name>8/0</Name>
      <Description/>
      <Capacity>0</Capacity>
    </NEInterface>
  </NetworkElement>
  <NetworkElement>
    <Name>router-1</Name>
    <Description/>
    <HostName/>
    <BackupHostName/>
    <NeId>router-1</NeId>
    <NetworkElementType>Router</NetworkElementType>
    <NeSubType>None</NeSubType>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
  </NetworkElement>
  <NetworkElement>
    <Name>router-2</Name>
    <Description/>
    <HostName/>
    <BackupHostName/>
    <NeId>router-2</NeId>
    <NetworkElementType>Router</NetworkElementType>
    <NeSubType>None</NeSubType>
    <Capacity>0</Capacity>
    <X>0</X>
  </NetworkElement>

```

```

    <Y>0</Y>
  </NetworkElement>
  <NetworkElement>
    <Name>mx-1</Name>
    <Description/>
    <HostName>10.60.100.101</HostName>
    <BackupHostName/>
    <NeId/>
    <NetworkElementType>Wireline Gateway</NetworkElementType>
    <NeSubType>MX Series</NeSubType>
    <Capacity>0</Capacity>
    <X>0</X>
    <Y>0</Y>
    <DiameterRealm>juniper.com</DiameterRealm>
    <ManualSubnet>
      <IpAddress>11.0.0.0</IpAddress>
      <Mask>8</Mask>
    </ManualSubnet>
    <Nediameterid>
    <NediameteridStr>mx.juniper.com</NediameteridStr>
    </Nediameterid>
  </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>
  <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>

```

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="2.1.6">
  <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>
  </Path>
</ConfigurationData>

```

```

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

```



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>OpenStream.VIDEO</Name>
        <Description></Description>
        <HDThreshold>2000000</HDThreshold>
      </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>
```

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="2.1.6">
  <Application>
    <Name>OpenStream.VIDEO</Name>
    <Description/>
    <HDThreshold>-1</HDThreshold>
  </Application>
</ConfigurationData>
```



Chapter 5

Subscriber Interface

Topics:

- [Accounts.....42](#)
- [Tiers.....45](#)

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.

```
<?xml version="1.0" encoding="UTF-8" ?> <XmlInterfaceRequest> <AddAccount>
  <Account>
    <AccountId>Account_BRAS_1</AccountId>
    <SubscriberData>8/0.2</SubscriberData>
    <NetworkElementName>erx-1</NetworkElementName>      <SubscriberGroupName/>
    <StaticIp>
      <IpAddress>72.68.130.71</IpAddress>      <IpCount>2</IpCount>
    </StaticIp>      <TierRef>      <Name>tier-1</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>Account_BRAS_1</AccountId>
  <SubscriberData>8/0.2</SubscriberData>
  <NetworkElementName>erx-1</NetworkElementName>
  <SubscriberGroupName/>
  <StaticIp>
    <IpAddress>72.68.130.71</IpAddress>
  </StaticIp>
  <IpCount>2</IpCount>
</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="2.1.6">
  <Account>
    <AccountId>000123</AccountId>
    <SubscriberData>8/0.2</SubscriberData>
    <NetworkElementName>erx-1</NetworkElementName>
    <StaticIp>
      <IpAddress>72.68.130.71</IpAddress>
      <IpCount>2</IpCount>
    </StaticIp>
    <TierRef>
      <Name>tier-1</Name>
    </TierRef>
  </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.....49*
- *OM Statistics Requests.....49*
- *OM Statistics Response Format.....51*
- *Timezones and Start/End Times.....52*
- *Counter Reset and Failover.....53*
- *Comparisons Between the CMP GUI and OM Statistics.....54*

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/> </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>
  </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.

Interval Statistics

In the interval-statistics mode, all numeric data generated by MPEs are reset at regular intervals controlled by the Stats Collection Period. When in Interval mode, a reset occurs on the hour and then every 5, 10, 15, 20, 30 or 60 minutes afterwards depending on the value of the Stats Collection Period. See the *CMP Wireless User Guide* for details on setting interval statistics.

When in Interval Mode, the XML request and response will look similar to the example below.

Request:

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

Response:

```
<OmStatsSetting>
  <ResetConfiguration>Interval</ResetConfiguration>
  <CollectionInterval>15</CollectionInterval>
</OmStatsSetting>
```

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.

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. It is recommended 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 then 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 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. 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 the MPE in the following conditions:

- The active MPE blade switches over one or more times within an interval period.
- The MPE has initially started up.
- Under high-load conditions, the MPE 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

Operational Measurements Interface Overview

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.....57*
- *Topology Update Statistics.....57*
- *Subscriber Update Statistics.....58*
- *Policy Server Statistics.....59*
- *Message Processing Statistics.....60*
- *Network Element Statistics.....61*
- *VoD Server SD/HD Session Statistics.....64*
- *Connected Network Elements Statistics.....67*
- *Diameter Synchronized Statistics.....67*
- *Diameter Policy Charging Enforcement Function Statistics.....68*
- *Diameter Policy Charging Enforcement Function Peer Statistics.....70*

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) 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.
 - 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.
 - Diameter Statistics — retrieves statistics on Policy Charging Enforcement Functions.
 -  Diameter Synch State Statistics - retrieves statistics on synchronized message between MX gateway router and MPE server using Gx-plus protocol. 

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>
      <TotalNetworkElementCount>52</TotalNetworkElementCount>
      <TotalSubscriberCount>1000000</TotalSubscriberCount>
    </Sample>
  </PolicyServerStats>
</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.
- **PeakTransactionRate:** An absolute value representing the peak number of transactions occurring within a given sample period.
- **SessionUnknownPathFailCount:** The number of session failures resulting from an unknown or nonexistent path.
- **SessionUnknownSubscriberCount:** The number of session failures resulting from an unknown subscriber ID.

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>
      <SessionCount>100</SessionCount>
      <SessionSuccessCount>97</SessionSuccessCount>
      <SessionFailCount>3</SessionFailCount>
      <SessionUnknownPathFailCount>0</SessionUnknownPathFailCount>
      <SessionUnknownSubscriberFailCount>0</SessionUnknownSubscriberFailCount>
      <SessionProtocolFailCount>1</SessionProtocolFailCount>
      <SessionPolicyFailCount>2</SessionPolicyFailCount>
      <PeakTransactionRate>0</PeakTransactionRate>
    </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.
 - **▶ ActiveSessionCount:** Current active sessions for that Network Element or Interface. This is a cumulative value and is displayed as an absolute. ◀
 - **▶ Session Count:** Session requests received. ◀
 - **SessionSuccessCount:** Successful sessions.
 - **SessionFailCount:** Session failures.

Note: Session failures can occur for a variety of reasons, for example, when a gate is created that is greater than the gate limit, the MPE responds with an error code. The default in this instance is "0" which means "no limit".

- **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.
- **MaxBandwidthDownstream:** This statistic represents the maximum committed bandwidth flows allocated for this Network Element or Interface. This value is also an absolute value that indicates the peak bandwidth usage in the history of a Network Element.

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



```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <NetworkElementStats>
    <Sample>
      <StartTime>2014-02-27T08:45:12Z</StartTime>
      <EndTime>2014-02-27T08:50:10Z</EndTime>
      <PolicyServer>MPE3-138</PolicyServer>
      <Name>MX-0250</Name>
      <Description></Description>
      <NeId>MX-0250</NeId>
    </Sample>
    <NetworkElementType>Wireline Gateway</NetworkElementType>
    <NeSubType>MX Series</NeSubType>
    <NeGroupName>NE-Group-02</NeGroupName>
    <ActiveSessionCount>167</ActiveSessionCount>
    <SessionCount>7</SessionCount>
    <SessionSuccessCount>7</SessionSuccessCount>
    <SessionFailCount>0</SessionFailCount>
    <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    <Capacity>0</Capacity>
    <BandwidthUpstream>0</BandwidthUpstream>
    <BandwidthDownstream>8550400</BandwidthDownstream>
    <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
    <MaxBandwidthDownstream>8652800</MaxBandwidthDownstream>
  </NetworkElementStats>
  <Interface>
    <InterfaceName>MARLBOROUGH:NANJING-MX-0250-22/0</InterfaceName>
    <Description></Description>
    <ActiveSessionCount>0</ActiveSessionCount>
    <SessionCount>0</SessionCount>
    <SessionSuccessCount>0</SessionSuccessCount>
    <SessionFailCount>0</SessionFailCount>
    <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    <Capacity>100000000</Capacity>
    <BandwidthUpstream>0</BandwidthUpstream>
    <BandwidthDownstream>0</BandwidthDownstream>
    <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
    <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
  </Interface>
</Sample>
```

```

    </NetworkElementStats>
  </Statistics>

```



VoD Server SD/HD Session Statistics

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

Request

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

This is an example of a request for all policy servers in the system for a version that is equal or greater than 2.2.14:

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryOmStats Version="2.2.14">
    <StartTime>2013-09-24T17:30:00Z</StartTime>
    <EndTime>2013-09-24T18:10:00Z</EndTime>
    <NetworkElementStats>
      <Name></Name>
    </NetworkElementStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

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

The following is an example XML response to this specific, (version 2.2.14 or greater), tag request (with version 2.2.14 showing HD and SD session counts):

```

<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <NetworkElementStats>
    <Sample>
      <StartTime>2013-09-24T17:30:00Z</StartTime>
      <EndTime>2013-09-24T17:35:00Z</EndTime>
      <PolicyServer>MPE001_Blade</PolicyServer>
      <Name>Router-C-203</Name>
      <Description></Description>
      <NeId>Router-C-203</NeId>
      <NetworkElementType>Router</NetworkElementType>
      <NeSubType>None</NeSubType>
      <NeGroupName>NE-Group-01</NeGroupName>
      <ActiveSessionCount>811</ActiveSessionCount>
      <SessionCount>39</SessionCount>
      <SessionSuccessCount>39</SessionSuccessCount>
      <SessionFailCount>0</SessionFailCount>
      <HDSessionCount>0</HDSessionCount>
      <SDSessionCount>39</SDSessionCount>
      <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
    </Sample>
  </NetworkElementStats>
</Statistics>

```

```

<Capacity>0</Capacity>
<BandwidthUpstream>0</BandwidthUpstream>
<BandwidthDownstream>1474</BandwidthDownstream>
<MaxBandwidthUpstream>0</MaxBandwidthUpstream>
<MaxBandwidthDownstream>0</MaxBandwidthDownstream>
<Interface>
  <InterfaceName>MARLBORO:NANJIN-Router-C-203-01/0</InterfaceName>

  <Description></Description>
  <ActiveSessionCount>0</ActiveSessionCount>
  <SessionCount>0</SessionCount>
  <SessionSuccessCount>0</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <HDSessionCount>0</HDSessionCount>
  <SDSessionCount>0</SDSessionCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>100000000</Capacity>
  <BandwidthUpstream>0</BandwidthUpstream>
  <BandwidthDownstream>0</BandwidthDownstream>
  <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
  <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
</Interface>
<Interface>
  <InterfaceName>MARLBORO:NANJIN-Router-C-203-02/0</InterfaceName>

  <Description></Description>
  <ActiveSessionCount>811</ActiveSessionCount>
  <SessionCount>39</SessionCount>
  <SessionSuccessCount>39</SessionSuccessCount>
  <SessionFailCount>0</SessionFailCount>
  <HDSessionCount>0</HDSessionCount>
  <SDSessionCount>39</SDSessionCount>
  <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
  <Capacity>100000000</Capacity>
  <BandwidthUpstream>0</BandwidthUpstream>
  <BandwidthDownstream>1474</BandwidthDownstream>
  <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
  <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
</Interface>
</Sample>
</NetworkElementStats>
</Statistics>

```

Alternate Request

This is an example of a request for all policy servers in the system for a version that is less than 2.2.14 or if this version does not exist:

```

<?xml version="1.0" encoding="UTF-8" ?>
<XmlInterfaceRequest>
  <QueryOmStats >
    <StartTime>2013-09-24T17:30:00Z</StartTime>
    <EndTime>2013-09-24T18:10:00Z</EndTime>
    <NetworkElementStats>
      <Name></Name>
    </NetworkElementStats>
  </QueryOmStats>
</XmlInterfaceRequest>

```

Response

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

This is an example XML response to this specific, (if version is less than 2.2.14 or version 2.2.14 does not exist in the system):

```
<?xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <NetworkElementStats>
    <Sample>
      <StartTime>2013-09-24T17:30:00Z</StartTime>
      <EndTime>2013-09-24T17:35:00Z</EndTime>
      <PolicyServer>MPE001_Blade</PolicyServer>
      <Name>Router-C-203</Name>
      <Description></Description>
      <NeId>Router-C-203</NeId>
      <NetworkElementType>Router</NetworkElementType>
      <NeSubType>None</NeSubType>
      <NeGroupName>NE-Group-01</NeGroupName>
      <ActiveSessionCount>811</ActiveSessionCount>
      <SessionCount>39</SessionCount>
      <SessionSuccessCount>39</SessionSuccessCount>
      <SessionFailCount>0</SessionFailCount>
      <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
      <Capacity>0</Capacity>
      <BandwidthUpstream>0</BandwidthUpstream>
      <BandwidthDownstream>1474</BandwidthDownstream>
      <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
      <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
      <Interface>
        <InterfaceName>MARLBORO:NANJIN-Router-C-203-01/0</InterfaceName>

        <Description></Description>
        <ActiveSessionCount>0</ActiveSessionCount>
        <SessionCount>0</SessionCount>
        <SessionSuccessCount>0</SessionSuccessCount>
        <SessionFailCount>0</SessionFailCount>
        <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
        <Capacity>100000000</Capacity>
        <BandwidthUpstream>0</BandwidthUpstream>
        <BandwidthDownstream>0</BandwidthDownstream>
        <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
        <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
      </Interface>
      <Interface>
        <InterfaceName>MARLBORO:NANJIN-Router-C-203-02/0</InterfaceName>

        <Description></Description>
        <ActiveSessionCount>811</ActiveSessionCount>
        <SessionCount>39</SessionCount>
        <SessionSuccessCount>39</SessionSuccessCount>
        <SessionFailCount>0</SessionFailCount>
        <AbnormalDisconnectCount>0</AbnormalDisconnectCount>
        <Capacity>100000000</Capacity>
        <BandwidthUpstream>0</BandwidthUpstream>
        <BandwidthDownstream>1474</BandwidthDownstream>
        <MaxBandwidthUpstream>0</MaxBandwidthUpstream>
        <MaxBandwidthDownstream>0</MaxBandwidthDownstream>
      </Interface>
    </Sample>
  </NetworkElementStats>
</Statistics>
```

Note: If the version number input tag is placed next to the XmlInterfaceRequest, then different responses will occur in the OSSI API. These are examples of request / response for this type of input.

Note: See [OSSI Response Failure Reason Code](#) for descriptions of possible errors.

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>
```

Diameter Synchronized Statistics

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

Request

This request follows the **QueryOmStats > DiameterStateSyncStatistics** 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 Version="2.2.14">
  <StartTime>2014-02-27T08:45:00Z</StartTime>
  <EndTime>2014-02-27T09:00:00Z</EndTime>
  <DiameterStateSyncStatistics/>
</QueryOmStats>
```

Response

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

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

```
<xml version="1.0" encoding="UTF-8"?>
<Statistics>
  <DiameterStateSyncStatistics>
    <Sample>
      <StartTime>2014-02-27T08:45:00Z</StartTime>
      <EndTime>2014-02-27T08:50:00Z</EndTime>
      <PolicyServer>MPE195</PolicyServer>
      <Name>MX-0050</Name>
      <NeId>MX-0050</NeId>
      <NetworkElementType>Wireline Gateway</NetworkElementType>
      <NetworkElementSubType>MX Series</NetworkElementSubType>
      <ConnectTime/>
      <DisconnectTime/>
      <ConnectAddress/>
      <ConnectPort>44192</ConnectPort>
      <MessagesInCount>5</MessagesInCount>
      <MessagesOutCount>5</MessagesOutCount>
      <JSERMessagesReceivedCount>5</JSERMessagesReceivedCount>
      <JSERMessagesSentCount>0</JSERMessagesSentCount>
      <JSEASuccessMessagesReceivedCount>0</JSEASuccessMessagesReceivedCount>
      <JSEASuccessMessagesSentCount>5</JSEASuccessMessagesSentCount>
      <JSEAFailureMessagesReceivedCount>0</JSEAFailureMessagesReceivedCount>
      <JSEAFailureMessagesSentCount>0</JSEAFailureMessagesSentCount>
      <JSDRMessagesReceivedCount>0</JSDRMessagesReceivedCount>
      <JSDRMessagesSentCount>0</JSDRMessagesSentCount>
      <JSDASuccessMessagesReceivedCount>0</JSDASuccessMessagesReceivedCount>
      <JSDASuccessMessagesSentCount>0</JSDASuccessMessagesSentCount>
      <JSDAFailureMessagesReceivedCount>0</JSDAFailureMessagesReceivedCount>
      <JSDAFailureMessagesSentCount>0</JSDAFailureMessagesSentCount>
    </Sample>
  </DiameterStateSyncStatistics>
</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>
    <CCAISuccessMessagesReceivedCount>0</CCAISuccessMessagesReceivedCount>
    <CCAISuccessMessagesSentCount>0</CCAISuccessMessagesSentCount>
    <CCAIFailureMessagesReceivedCount>0</CCAIFailureMessagesReceivedCount>
    <CCAIFailureMessagesSentCount>0</CCAIFailureMessagesSentCount>
    <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>

```

Chapter 8

Identity Management (IDM)

Topics:

- [Identity Management Overview.....73](#)
- [Add a User.....73](#)
- [Update a User.....79](#)
- [Query a User.....84](#)
- [Delete a User.....86](#)
- [Query User Role.....87](#)
- [Query User Scope.....93](#)

Identity Management (IDM) allows the CMP to configure user names, passwords and roles using the OSSI interface. It also allows querying of user scopes and roles.

Identity Management Overview

The Identity Management system connects to a CMP through an HTTP or HTTPS interface. The user must specify the operation parameter **idm** in the requested URL, similar to `http://127.0.0.1/mi/xmlInterfaceRequest.do?user=supervisor&pwd=password&operation=idm`.

The account used for connecting the OSSI to the CMP is a regular CMP user account, with the default global scope and the following privileges:

- Show Privilege for OSSI Import/Export
- Read-Write Privilege for User Management

The Identity Management feature allows users to perform the following tasks:

- Add a User - Create one or more users, or update a current user's parameters by overwriting them with new parameters.
- Update a User - Update parameters for one or more users.
- Query a User - Query one user if the Name element occurs, or all users if no Name element occurs.
- Delete a User - Delete one or more users.
- Query Role - Query one role if the Name element occurs, or all roles if no Name element occurs.
- Query Scope - Query one scope if the Name element occurs, or all scopes if no Name element occurs.

Add a User

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

Successful Request

Request

This request follows the `AddSysAdminUser` tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>camiant</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>
```

```

</AddSysAdminUser>
</XmlInterfaceRequest>

```

Table 3: AddSysAdminUser Operation Parameters

Parameter Name	Optional/Required	Limitation
User Name	Required	250 characters (string)
Description	Optional	250 characters (string)
Password	Required	A clear text
LockedStatus	Required	0 = unlocked; 1 = locked
RoleRef	Required	Can specify only one role
ScopeRef	Required	Can specify multiple scopes

Response

This request follows the AddSysAdminUser tag defined in the XSDs.

```

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

```

The errors that a user may receive are shown below.

Table 4: AddSysAdminUser Operation Errors

Error	Message	Description
ERROR_USER_INVALID_NAME	Invalid User Name for: '{\${UserName}'	UserName is an empty string.
ERROR_USER_NAME_TOO_LONG	User Name exceeds max length for: '{\${UserName}'	UserName exceeds max length (250 characters).
ERROR_USER_DESCRIPTION_TOO_LONG	User Description exceeds max length. User: '{\${UserName}'	Description exceeds max length (250 characters)
ERROR_USER_NOROLE_OR_NOSCOPE	User: '{\${Username}' must have an associated Role and Scope.	Non-Role or non-Scope is defined in OSSI command.
ERROR_USER_INVALID_ROLE	Invalid Role Name for: '{\${RoleName}'	RoleName is an empty string, or

Error	Message	Description
NAME		RoleName exceeds max length (250 characters), or There is more than one Role defined.
ERROR_USER_INVALID_SCOPE_NAME	Invalid Scope Name for: '{\$ScopeName}'	ScopeName is an empty string, or ScopeName is exceeds max length (250 characters).
ERROR_USER_ROLE_LINK	The user: {\$UserName}'s associated role does not exist. Please create it first.	RoleName is not defined in CMP.
ERROR_USER_SCOPE_LINK	The user: {\$UserName}'s associated scope does not exist. Please create it first.	ScopeName is not defined in CMP.
ERROR_SCHEMA_INVALID	N/A	If the input OSSI command cannot match the schema specification, there will be an error-message response to the operator. For example, if you define a LockStatus with a string, the following message will be reported: "Error parsing import file: Error parsing import file: cvc-datatype-valid.1.2.1: 'ABC' is not a valid value for 'integer'."

ERROR_USER_INVALID_NAME

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name></Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>oracle</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>
```

```

    </RoleRef>
    <ScopeRef>
      <Name>Global</Name>
    </ScopeRef>
  </SysAdminUser>
</AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user name is invalid.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 User(s).

      Invalid User Name for: ''</Failure>
    </Command>
</Response>

```

ERROR_USER_INVALID_ROLE_NAME

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>oracle</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name></Name>
      </RoleRef>
      <ScopeRef>
        <Name>Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the role name is invalid.

```

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

```

```

    <Failure count="1">Failed to import 1 User(s).

    Invalid Role Name for: 'admin1'</Failure>
</Command>
</Response>

```

ERROR_USER_INVALID_SCOPE_NAME

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>oracle</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name></Name>
      </ScopeRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the scope name is invalid.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 User(s).

    Invalid Scope Name for: 'admin1'</Failure>
  </Command>
</Response>

```

ERROR_USER_ROLE_LINK

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>

```

```

    <Name>admin1</Name>
    <Description>The default administrator user with all privileges</Description>

    <Password>oracle</Password>
    <LockedStatus>0</LockedStatus>
    <ScopeRef>
      <Name>Not Exist Global</Name>
    </ScopeRef>
  </SysAdminUser>
</AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user's associated role does not exist.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 User(s).

    User: 'admin1' must have an associated Role and Scope.</Failure>
  </Command>
</Response>

```

ERROR_USER_NOROLE_OR_NOSCOPE

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>oracle</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Not Exits Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user's role does not exist.

```

<?xml version='1.0' ?>

```

```

<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 User(s).

    The user: admin1's associated role does not exist. Please create it
    first.</Failure>
  </Command>
</Response>

```

Request

This request follows the AddSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <AddSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>The default administrator user with all privileges</Description>

      <Password>oracle</Password>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Not Exist Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </AddSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user's scope does not exist.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to import 1 User(s).

    The user: admin1's associated scope does not exist. Please create it
    first.</Failure>
  </Command>
</Response>

```

Update a User

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

Request

This request follows the UpdateSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>New The default administrator user with all
        privileges</Description>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </UpdateSysAdminUser>
</XmlInterfaceRequest>
```

Table 5: UpdateSysAdminUser Operation Parameters

Parameter Name	Optional/Required	Limitation
User Name	Required	250 characters (string)
Description	Optional	250 characters (string)
Password	Optional	A clear text
LockedStatus	Optional	0 = unlocked; 1 = locked
RoleRef	Optional	Can specify only one role
ScopeRef	Optional	Can specify multiple scopes

The probable errors a user may receive are shown below.

Table 6: UpdateSysAdminUser Operation Errors

Error	Message	Description
ERROR_USER_NAME_NOT_EXIST	Specified User does not exist: '{UserName}'	The specified user is not in system.
ERROR_USER_INVALID_NAME	Invalid User Name for: '{UserName}'	UserName is an empty string.
ERROR_USER_NAME_TOO_LONG	User Name exceeds max length for: '{UserName}'	UserName exceeds max length (250 characters).
ERROR_USER_DESCRIPTION	User Description exceeds max length. User: '{UserName}'	Description exceeds max length (250 characters)

Error	Message	Description
_TOO_LONG		
ERROR_USER_NOROLE_OR_NOSCOPE	User: '{\$Username}' must have an associated Role and Scope.	Non-Role or non-Scope is defined in OSSI command.
ERROR_USER_INVALID_ROLE_NAME	Invalid Role Name for: '{\$RoleName}'	RoleName exceeds max length (250 characters), or There is more than one Role defined.
ERROR_USER_INVALID_SCOPE_NAME	Invalid Scope Name for: '{\$ScopeName}'	ScopeName exceeds max length (250 characters).
ERROR_USER_ROLE_LINK	The user: {\$UserName}'s associated role does not exist. Please create it first.	RoleName is an empty string, or RoleName is not defined in CMP.
ERROR_USER_SCOPE_LINK	The user: {\$UserName}'s associated scope does not exist. Please create it first.	ScopeName is an empty string, or ScopeName is not defined in CMP.
ERROR_SCHEMA_INVALID	N/A	If the input OSSI command cannot match the schema specification, there will be an error-message response to the operator. For example, if you define a LockStatus with a string, the following message will be reported: "Error parsing import file: Error parsing import file: cvc-datatype-valid.1.2.1: 'ABC' is not a valid value for 'integer'."

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 User(s).</Success>
    <Failure count="0"></Failure>
```

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

ERROR_USER_NAME_NOT_EXIST

Request

This request follows the UpdateSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateSysAdminUser>
    <SysAdminUser>
      <Name>NotExistadmin1</Name>
      <Description>New The default administrator user with all
        privileges</Description>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </UpdateSysAdminUser>
</XmlInterfaceRequest>
```

Response

The following error response occurs when the user name does not exist.

```
<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to update 1 User(s).

    Specified User does not exist: 'NotExistadmin1'</Failure>
  </Command>
</Response>
```

ERROR_USER_NOROLE_OR_NOSCOPE

Request

This request follows the UpdateSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>New The default administrator user with all
        privileges</Description>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
```

```

    <Name>not exist Administrator</Name>
  </RoleRef>
  <ScopeRef>
    <Name>Global</Name>
  </ScopeRef>
</SysAdminUser>
</UpdateSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user's specified role does not exist.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to update 1 User(s).

    The user: admin1's associated role does not exist. Please create it
    first.</Failure>
  </Command>
</Response>

```

Request

This request follows the UpdateSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>New The default administrator user with all
        privileges</Description>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>Not exist Global</Name>
      </ScopeRef>
    </SysAdminUser>
  </UpdateSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the user's specified scope does not exist.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to update 1 User(s).

```

```

    The user: admin1's associated scope does not exist. Please create it
first.</Failure>
</Command>
</Response>

```

ERROR_USER_INVALID_ROLE_NAME

Request

This request follows the UpdateSysAdminUser tag defined in the XSDs.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <UpdateSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
      <Description>New The default administrator user with all
privileges</Description>
      <LockedStatus>0</LockedStatus>
      <RoleRef>
        <Name>123</Name>
      </RoleRef>
      <RoleRef>
        <Name>Administrator</Name>
      </RoleRef>
      <ScopeRef>
        <Name>123</Name>
      </ScopeRef>
    </SysAdminUser>
  </UpdateSysAdminUser>
</XmlInterfaceRequest>

```

Response

The following error response occurs when the role name either exceeds the maximum length of 250 characters, or there is more than one Role defined.

```

<?xml version='1.0' ?>
<Response>
  <Result>0</Result>
  <Command type="XmlInterfaceResponse">
    <Success count="0"></Success>
    <Failure count="1">Failed to update 1 User(s).

    Invalid Role Name for: 'admin1'</Failure>
  </Command>
</Response>

```

Query a User

The following examples show the request and response that are defined in the XSDs for the QuerySysAdminUser tag. If the Name element occurs, the CMP will query only the corresponding user; otherwise, all users are queried.

Request

This request follows the QuerySysAdminUser tag defined in the XSDs.

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

Table 7: QuerySysAdminUser Operation Parameters

Parameter Name	Optional/Required	Limitation
Name	Optional	250 characters (string)

If the query is successful, the list of users will be returned in XML as a response with the following fields for each user.

Table 8: Output Fields for QuerySysAdminUser Operation

Field Name	Description
Name	The user's name
Description	The user's description
Password	The user's password, encrypted
LockedStatus	0 = locked, 1 = unlocked
RoleRef	The associated role
ScopeRef	The associated scope; there can be multiple scopes.

Response

The response to this request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="7.5.0">
  <SysAdminUser>
    <Name>admin</Name>
    <Description>The default administrator user with all
privileges</Description>
    <Password>835154D6D3FA2C3575AA700A54AB9F6492E7ABB7</Password>
    <LockedStatus>0</LockedStatus>
    <RoleRef>
      <Name>Administrator</Name>
    </RoleRef>
    <ScopeRef>
      <Name>Global</Name>
    </ScopeRef>
```

```
</SysAdminUser>
</ConfigurationData>
```

Delete a User

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

Successful Deletion

Request

This request follows the DeleteSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
    </SysAdminUser>
  </DeleteSysAdminUser>
</XmlInterfaceRequest>
```

Table 9: DeleteSysAdminUser Operation Parameters

Parameter Name	Optional/Required	Limitation
Name	Required	250 characters (string)

The probable errors a user may receive are shown below.

Table 10: DeleteSysAdminUser Operation Errors

Error	Message	Description
ERROR_DELETE_FAILURE	Error deleting User Name: {\${UserName}}	The specified user does not exist, or "Admin" user is specified.

Response

The response to this request:

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

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

If deletion is successful, the number of deleted users will be reported.

ERROR_DELETE_FAILURE

Request

This request follows the DeleteSysAdminUser tag defined in the XSDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlInterfaceRequest>
  <DeleteSysAdminUser>
    <SysAdminUser>
      <Name>admin1</Name>
    </SysAdminUser>
  </DeleteSysAdminUser>
</XmlInterfaceRequest>
```

Response

The response to this request when the specified user name does not exist:

```
<?xml version='1.0' ?>
<Response>
<Result>0</Result>
<Command type="XmlInterfaceResponse">
<Success count="0"></Success>
<Failure count="1">Failed to delete 1 user.
Error deleting User Name: admin1</Failure>
</Command>
</Response>
```

Query User Role

The following examples show the request and response that are defined in the XSDs for the QueryRole tag. If the Name element occurs, the CMP will query the corresponding role with the name; otherwise, it will query all roles.

Request

This request follows the QueryRole tag defined in the XSDs.

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

Table 11: QueryRole Operation Parameters

Parameter Name	Optional/Required	Limitation
Name	Optional	250 characters (string)

If the query is successful, the list of users will be returned in XML as a response with the following fields for each user.

Table 12: Output Fields for QueryRole Operation

Field Name	Description
Name	The role's name
Description	The role's description
RolePrivilege	The privilege with privilege name and access level assigned to the Role. There can be multiple privileges.

Response

The response to this request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="2.1.6">
  <Role>
    <Name>Administrator</Name>
    <Description>The default role with all privileges</Description>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
      <PrivilegeUid>10</PrivilegeUid>
      <Level>40</Level>
    </RolePrivilege>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
      <PrivilegeUid>11</PrivilegeUid>
      <Level>40</Level>
    </RolePrivilege>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
      <PrivilegeUid>12</PrivilegeUid>
      <Level>40</Level>
    </RolePrivilege>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
      <PrivilegeUid>13</PrivilegeUid>
      <Level>40</Level>
    </RolePrivilege>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
      <PrivilegeUid>14</PrivilegeUid>
      <Level>40</Level>
    </RolePrivilege>
    <RolePrivilege>
      <RoleUid>42</RoleUid>
    </RolePrivilege>
  </Role>
</ConfigurationData>
```

```

    <PrivilegeUid>15</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>16</PrivilegeUid>
    <Level>0</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>17</PrivilegeUid>
    <Level>50</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>18</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>19</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>20</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>21</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>22</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>23</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>24</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>25</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>26</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>27</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>

```

```

    <RoleUid>42</RoleUid>
    <PrivilegeUid>28</PrivilegeUid>
    <Level>10</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>29</PrivilegeUid>
    <Level>20</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>30</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>31</PrivilegeUid>
    <Level>70</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>32</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>33</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>34</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>35</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>36</PrivilegeUid>
    <Level>10</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>37</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>38</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>39</PrivilegeUid>
    <Level>0</Level>
  </RolePrivilege>
  <RolePrivilege>
    <RoleUid>42</RoleUid>
    <PrivilegeUid>40</PrivilegeUid>
    <Level>40</Level>
  </RolePrivilege>

```

```

<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>41</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>42</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>43</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>44</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>45</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>46</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>47</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>48</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>49</PrivilegeUid>
  <Level>10</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>50</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>51</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>52</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>53</PrivilegeUid>
  <Level>40</Level>

```

```

</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>54</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>55</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>56</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>57</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>58</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>59</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>60</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>61</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>62</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>63</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>64</PrivilegeUid>
  <Level>40</Level>
</RolePrivilege>
<RolePrivilege>
  <RoleUid>42</RoleUid>
  <PrivilegeUid>70</PrivilegeUid>
  <Level>10</Level>
</RolePrivilege>
</Role>
</ConfigurationData>

```

A subset of the RolePrivilege element will be outputted in XML based on the mode selected by the operator. The output XML cannot be imported to the CMP again.

Query User Scope

The following examples show the request and response that are defined in the XSDs for the QueryScope tag. If the Name element occurs, the CMP will query only the corresponding scope, otherwise all scopes are queried.

Request

This request follows the QueryScope tag defined in the XSDs.

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

If query is successful, the list of the scopes will be returned in XML as a response. If no Ref element occurs, it means the scope is a global scope in system.

Table 13: QueryScope Operation Parameters

Parameter Name	Optional/Required	Limitation
Name	Optional	250 characters (string)

If the query is successful, the list of the roles will be returned in XML as a response.

Table 14: Output Fields for QueryScope Operation

Field Name	Description
Name	The scope's name
Description	The scope's description
ResourceControllerGroupRef	The manageable MPEs' name set.
NetworkElementGroupRef	The manageable Network Elements' name set.

Response

The response to this request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConfigurationData version="7.5.0">
  <Scope>
```

```
<Name>Global</Name>  
<Description>The default scope that automatically contains all items in  
the system</Description>  
</Scope>  
</ConfigurationData>
```

A subset of the Ref element will be outputted in XML based on the mode selected by the operator. The output XML cannot be imported to the CMP again.