Cisco Media Gateway Manager 5.0

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-5465-01
CONTENTS

Preface vii

Objectives vii

Audience vii

Conventions viii

Related Documentation viii

Obtaining Documentation x

Cisco.com x

Ordering Documentation x

Documentation Feedback x

Obtaining Technical Assistance xi

Cisco Technical Support Website xi

Submitting a Service Request xi

Definitions of Service Request Severity xii

Obtaining Additional Publications and Information xii

CHAPTER 1

Overview of Cisco MGM GateWay/CORBA 1-1

1.1 Features of Cisco MGM GateWay/CORBA 1-2

1.2 Cisco MGM GateWay/CORBA Fault Management 1-3

1.3 Overview of CORBA Services 1-3

1.3.1 Naming Service 1-3

1.3.2 Notification Service 1-3

1.4 Information Model 1-6

1.5 Summary 1-6

CHAPTER 2

NE- and Cisco MGM-Specific Details 2-1

2.1 NE-Specific Details 2-1

2.2 Cisco MGM-Specific Details 2-1

2.2.1 Layer Rates 2-2

2.2.2 Inventory 2-2

2.2.3 Interfaces 2-2

2.2.4 Object Details 2-3

2.2.4.1 AVC Events 2-3

2.2.4.2 Managed Element and Object Creation Event 2-3
2.2.4.3 Source of Object Attributes reported by MGM 2-3

2.2.5 Naming Conventions 2-5
  2.2.5.1 ManagedElement 2-5
  2.2.5.2 Physical Termination Point 2-5
  2.2.5.3 Cisco MGX 8880/8850 Equipment 2-6
  2.2.5.4 EquipmentHolder 2-6
  2.2.5.5 Boolean Mapping 2-7
  2.2.5.6 Resource Cleanup 2-7
  2.2.5.7 Development Environment 2-7
  2.2.5.8 Exception Handling 2-7

CHAPTER 3
Using Cisco MGM GateWay/CORBA Interfaces 3-1

3.1 equipment::EquipmentInventoryMgr_I 3-1
  3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment 3-1
  3.1.2 equipment::EquipmentInventoryMgr_I::getAllEquipmentNames 3-4
  3.1.3 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs 3-5
  3.1.4 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPNames 3-7
  3.1.5 equipment::EquipmentInventoryMgr_I::getAllSupportingEquipment 3-8
  3.1.6 equipment::EquipmentInventoryMgr_I::getAllSupportingEquipmentNames 3-9
  3.1.7 equipment::EquipmentInventoryMgr_I::getContainedEquipment 3-10
  3.1.8 equipment::EquipmentInventoryMgr_I::getEquipment 3-11

3.2 managedElementManager::ManagedElementManager_I 3-12
  3.2.1 managedElementManager::ManagedElementManager_I::getAllActiveAlarms 3-13
  3.2.2 managedElementManager::ManagedElementManager_I::getAllManagedElements 3-14
  3.2.3 managedElementManager::ManagedElementManager_I::getAllManagedElementNames 3-16
  3.2.4 managedElementManager::ManagedElementManager_I::getAllPTPs 3-17
  3.2.5 managedElementManager::ManagedElementManager_I::getAllPTPNames 3-18
  3.2.6 managedElementManager::ManagedElementManager_I::getMangedElement 3-19
  3.2.7 managedElementManager::ManagedElementManager_I::forceNEResync 3-20

APPENDIX A
Notifications, Fault Management, and Probable Cause Mapping A-1

A.1 Overview of Notifications A-1
A.2 Resolve the Notification Service A-2
A.3 Tips to Improve Notification Throughput A-2
A.4 Supported Events A-3
  A.4.1 Object Creation A-3
A.4.2 Object Deletion A-4
A.4.3 Attribute Value Change A-5
A.4.4 State Change Event A-5
A.4.5 Alarm A-6
A.4.6 Heartbeat Event A-7

A.5 Probable Cause Mapping A-7
A.5.1 TMF-Defined Probable Cause A-7
A.5.2 Alarm Probable Cause Mapping A-9

APPENDIX B

Server Administration and Configuration B-1
B.1 Naming Convention for Published CORBA Objects B-1
B.2 Using Multiple Naming Servers B-2
B.3 Configuring Cisco MGM GateWay/CORBA B-3
  B.3.1 Configuring the Notification Service Listening Port B-4
B.4 Installation B-5
B.5 Managing OSS Client Profiles for Cisco MGM GateWay/CORBA B-5
  B.5.1 Create an OSS Client Profile B-5
  B.5.2 Modify an OSS Client Profile B-6
  B.5.3 Delete an OSS Client Profile B-6
B.6 Using Notification Service B-7
  B.6.1 Develop a Cisco MGM GateWay/CORBA Client B-7
    B.6.1.1 Recommended Tools B-8
    B.6.1.2 Sample Code for Receiving Notifications B-8
    B.6.1.3 Initialize the Client Connection B-9
    B.6.1.4 Get Reference to Naming Service B-9
    B.6.1.5 Get Reference to EventChannel B-9
    B.6.1.6 Obtain ConsumerAdmin Reference B-9
    B.6.1.7 Obtain ProxyPushSupplier B-10
    B.6.1.8 Implement StructuredPushConsumer B-10
    B.6.1.9 Connect StructuredPushConsumerImpl B-10
    B.6.1.10 Sample Code for Inventory Retrieval B-11
    B.6.1.11 Get Reference to EmsSessionFactory B-11
    B.6.1.12 Log In and Retrieve EmsSession B-11
    B.6.1.13 Retrieve List of Managers B-12
    B.6.1.14 Operations on EMS Manager B-12
B.7 Encryption Between Cisco MGM and GateWay/CORBA B-12
B.8 Using the CLI to Start and Stop Cisco MGM GateWay/CORBA B-13
APPENDIX C  
OSS Client Development and Use Cases  C-1
  C.1 Use Cases to Resynchronize the NMS from the EMS  C-1
  C.2 Developing a Cisco MGM Gateway/CORBA Client  C-3
    C.2.1 Recommended Tools  C-3
    C.2.2 General Procedure  C-3
    C.2.3 Sample Code in Java  C-4
      C.2.3.1 Initialize the Client Connection  C-4
      C.2.3.2 Get Reference to Naming Service  C-4
      C.2.3.3 Sample Code for Operations on Supported Managers  C-4
      C.2.3.4 Get Reference to EventChannel  C-6
      C.2.3.5 Obtain ConsumerAdmin Reference  C-6
      C.2.3.6 Obtain ProxyPushSupplier  C-6
      C.2.3.7 Implement StructuredPushConsumer  C-6
      C.2.3.8 Connect StructuredPushConsumerImpl  C-7

APPENDIX D  
Cisco MGM Gateway/CORBA Supported IDLs  D-1
Preface

Objectives

This document describes the Cisco Media Gateway Manager GateWay/Common Object Request Broker Architecture (CORBA) Northbound Interface Release 5.0 product that is available for Cisco Media Gateway Manager (Cisco MGM) Release 5.0. This document serves as a reference for developers of operations support system (OSS) applications that work with the Cisco MGM GateWay/CORBA interface.

Use this document to:

• Learn about the northbound GateWay/CORBA interfaces supported by Cisco MGM R5.0
• Understand the input parameters, output parameters, and attributes associated with Cisco MGM GateWay/CORBA

Audience

The primary audience for this guide includes network operations personnel and system administrators who are responsible for Cisco MGX 8880 and 8850 network management. This guide assumes the reader is familiar with the following products and topics:

• Cisco Media Gateway Manager
• Basic internetworking terminology and concepts
• Network topology and protocols
• Microsoft Windows 2000 and NT 4.0
• Sun Microsystems Solaris administration
• Oracle database administration
• Object Management Group (OMG) CORBA, Notification Service, and Naming Service
Conventions

This document uses the following conventions:

<table>
<thead>
<tr>
<th>Item</th>
<th>Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commands and keywords</td>
<td><strong>boldface</strong> font</td>
</tr>
<tr>
<td>Displayed session and system</td>
<td><strong>screen</strong> font</td>
</tr>
<tr>
<td>information</td>
<td></td>
</tr>
<tr>
<td>Information that the user must</td>
<td><strong>boldface screen</strong> font</td>
</tr>
<tr>
<td>enter</td>
<td></td>
</tr>
<tr>
<td>Variables that the user must</td>
<td><strong>italic screen</strong> font</td>
</tr>
<tr>
<td>supply</td>
<td></td>
</tr>
<tr>
<td>Menu items and button names</td>
<td><strong>boldface</strong> font</td>
</tr>
<tr>
<td>Selecting a menu item</td>
<td>Option &gt; Network Preferences</td>
</tr>
</tbody>
</table>

**Caution**

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

**Note**

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

**Timesaver**

Means *the described action saves time*. You can save time by performing the action described in the paragraph.

**Tip**

Means *the information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Related Documentation

Make sure to have access to the documents listed in Table 1. These documents are available in print, on CD-ROM, and on www.cisco.com.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Related Cisco Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Product</td>
<td>Document Title</td>
</tr>
<tr>
<td>Cisco MGM</td>
<td>• <em>Cisco Media Gateway Manager 5.0 User Guide</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Cisco Media Gateway Manager 5.0 Installation Guide</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Release Notes for Cisco Media Gateway Manager Release 5.0</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Cisco Media Gateway Manager 5.0 High Availability Installation Guide</em></td>
</tr>
<tr>
<td>Cisco MGX 8880 and Cisco MGX 8850</td>
<td>All Cisco MGX 8800 documentation.</td>
</tr>
</tbody>
</table>
The following related documentation may also be used as reference:

- Telcordia GR-815-CORE: Generic Requirements for Network Elements/Network Systems (NE/NS) Security
- Telcordia GR-820-CORE: Generic Digital Transmission Surveillance
- Telcordia GR-831-CORE: Operations Applications Messages
- ITU-T G.826: Error performance parameters and objectives for international, constant bit rate digital paths at or above the primary rate
- ITU-T G.872: Architecture of optical transport networks
- ITU-T M.3010: Principles for a Telecommunications management network
- ITU-T M.3100: Generic network information model
- ITU-T M.3400: TMN Management Functions
- ITU-T X.721: Information technology—Open Systems Interconnection—Structure of management information: Definition of management information
- ITU-T X.731: Information technology—Open Systems Interconnection—Systems Management: State management function
- ITU-T X.735: Information technology—Open Systems Interconnection—Systems Management: Log control function
- ITU-T X.736: Information technology—Open Systems Interconnection—Systems Management: Security alarm reporting function
- TMF 513: TM Forum Business Agreement, version 1.02
- TMF 513: Multi-Technology Network Management Business Agreement, version 2.0
- TMF 608: Multi Technology Network Management Information Agreement, version 0.4
- TMF 608: Multi Technology Network Management Information Agreement, version 2.0
- TMF 814: Multi Technology Network Management Solution Set, version 0.2
- TMF 814: Multi-Technology Network Management Solution Set, version 2.0
- TMF 814A: Multi-Technology Network Management Solution Set, TM Forum Implementation (IS) Template Guidelines, version 2.0
Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:
http://www.cisco.com/univercd/home/home.htm
You can access the Cisco website at this URL:
http://www.cisco.com
You can access international Cisco websites at this URL:

Ordering Documentation

You can find instructions for ordering documentation at this URL:
You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:
- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 1 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.
You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:
Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883
We appreciate your comments.
Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:


Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support Website by clicking the Tools & Resources link under Documentation & Tools. Choose Cisco Product Identification Tool from the Alphabetical Index drop-down list, or click the Cisco Product Identification Tool link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting show command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts
Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- **Cisco Marketplace** provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:
  
  http://www.cisco.com/go/marketplace/

- **The Cisco Product Catalog** describes the networking products offered by Cisco Systems, as well as ordering and customer support services. Access the Cisco Product Catalog at this URL:
  
  http://cisco.com/univercd/cc/td/doc/pcat/

- **Cisco Press** publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:
  
  http://www.ciscopress.com

- **Packet magazine** is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:
  
  http://www.cisco.com/packet

- **iQ Magazine** is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:
  
  http://www.cisco.com/go/iqmagazine
- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:
  [http://www.cisco.com/ipj](http://www.cisco.com/ipj)

- World-class networking training is available from Cisco. You can view current offerings at this URL:
Overview of Cisco MGM GateWay/CORBA

In this document, the terms network element (NE) and managed element (ME) are interchangeable.

The Cisco MGM GateWay/CORBA product is a CORBA northbound interface that complies with the TMF CORBA specification for network management layer-element management layer (NML-EML) interfaces.

A programming technology for distributed computing, CORBA enables components of various application programs to communicate with one another regardless of their implementation characteristics—programming language, operating system, machine, location, and so on. Cisco MGM GateWay/CORBA allows service providers to implement a programmatic interface between Cisco MGM and various OSS tools in the back office. Such interfaces enable next-generation flow-through equipment management, including inventory, provisioning, and assurance functions.

For each NE that Cisco MGM manages, the Cisco MGM GateWay/CORBA option provides higher layer management systems with fault, and inventory information. Cisco MGM GateWay/CORBA supports up to eight simultaneous sessions with higher layer management systems.

Table 1-1 shows the NE support for Cisco MGM GateWay/CORBA in Cisco MGM R5.0.

<table>
<thead>
<tr>
<th>MGX NE</th>
<th>CORBA Support?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGX 8880</td>
<td>Yes</td>
</tr>
<tr>
<td>MGX 8850</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 1-1 illustrates the Cisco MGM GateWay/CORBA communications architecture within a service provider’s OSS environment.
The telemanagement community has adopted CORBA as the preferred distribution technology with which to integrate the various components of an OSS. The Multi Technology Network Management Working Group of the TMF has been working to create a specification for an NML-EML interface. Cisco MGM GateWay/CORBA is based on the following TMF standards:

- TMF513 v2.0 (August 2001): Multi Technology Network Management Business Agreement
- TMF608 v2.0 (August 2001): Multi Technology Network Management Information Agreement
- TMF814 v2.0 (August 2001): Multi Technology Network Management Solution Set

The TMF.608 document is a protocol-independent information model for the network resources being managed. The TMF.814 document is the CORBA Interface Data Language (IDL) component of the specification; it consists of a set of definitions for specific objects and methods that are available for use at the NML-EML interface for passing information or accomplishing specific actions. However, TMF.814 is a specific interface implementation based on a TMF high-level business agreement described by the TMF.513 document. To complete the document set, TMF.814A recognizes that different parties may provide varying degrees of compliance in their implementations. The TMF.814A document is called the Implementation Agreement, and is simply a form to be filled out by those parties implementing TMF.814 to further describe their specific implementations to those who would use them.

### 1.1 Features of Cisco MGM GateWay/CORBA

Cisco MGM GateWay/CORBA supports the following TMF 814 modules:

- **Inventory**
  - Element management system (EMS)
  - Managed element
  - Physical Termination Point (PTP)
  - Equipment
  - Equipment Holder
- **Alarm notification**
- **Retrieval of active alarms on objects where inventory is available**

The gateway authenticates the northbound OSS. Once authenticated, the OSS can perform operations and receive alarm notifications.

Cisco MGM GateWay/CORBA uses JacORB 1.4.1.8. The corresponding standard is Object Management Group (OMG) CORBA 2.3. For information about JacORB, visit [www.jacorb.org](http://www.jacorb.org).
1.2 Cisco MGM GateWay/CORBA Fault Management

Cisco MGM GateWay/CORBA forwards all the fault notifications received from managed NEs and all Cisco MGM alarms to the NMS through notification service. Information about notification service is available on the OMG website at www.omg.org.

The NMS retrieves all active alarms on the EMS and on managed element objects. The NMS filters alarms based on probable cause, perceived severity, and managed element name.

1.3 Overview of CORBA Services

Cisco MGM GateWay/CORBA uses two CORBA services—naming service and notification service. Naming service is required for Cisco MGM GateWay/CORBA to function. If notification service is not available, alarm and event notifications are not forwarded to the OSS. By default, the naming and notification services are installed on the Cisco MGM server during the GateWay/CORBA installation. However, the CORBA services can be located anywhere as long as they are accessible from the Cisco MGM server on which Cisco MGM GateWay/CORBA is installed. The OSS application(s) must also use the same instances of the naming service and the notification service.

1.3.1 Naming Service

Cisco MGM GateWay/CORBA and the other Cisco MGM server processes must use a naming service in order to operate. Cisco MGM GateWay/CORBA includes the OpenFusion naming service product. By default, Cisco MGM GateWay/CORBA is configured to use OpenFusion. However, you can configure Cisco MGM GateWay/CORBA and the Cisco MGM server to use a different centralized naming service. For more information, see the “B.2 Using Multiple Naming Servers” section on page B-2.

1.3.2 Notification Service

Cisco MGM GateWay/CORBA must use a CORBA notification service to report faults and events. Cisco MGM GateWay/CORBA includes the PrismTech OpenFusion notification service product, and is configured by default to use it. As with the naming service, you can configure Cisco MGM GateWay/CORBA to use a different notification service. For more information, see the “A.1 Overview of Notifications” section on page A-1.

The following scenarios describe possible configurations of the naming and notification services.
Scenario 1: Naming and Notification Services Installed on Cisco MGM Server

The naming and notification services are located on the Cisco MGM server workstation. (See Figure 1-2.)

The notification service must be installed on the Cisco MGM server workstation with the Cisco MGM server processes. Cisco MGM GateWay/CORBA will be configured to use this notification service. The NMS processes must be configured to use the notification and naming services on the Cisco MGM server workstation.
Scenario 2: Naming and Notification Services Installed on NMS Workstation

The naming and notification services are located on the NMS workstation. (See Figure 1-3.)

Figure 1-3  Naming and Notification Services on the NMS Workstation

The Cisco MGM server processes and Cisco MGM GateWay/CORBA must be configured to use the naming and notification services on the NMS workstation.
Scenario 3: Naming and Notification Services Installed on Independent Servers

The naming and notification services are installed on independent servers. (See Figure 1-4.)

Figure 1-4  Naming and Notification Services on Independent Servers

The Cisco MGM server processes, Cisco MGM GateWay/CORBA, and the NMS processes must be configured to use the naming and notification services on the independent workstations.

1.4 Information Model

The information model used by the Cisco MGM GateWay/CORBA northbound interface is based on the TMF NML-EML CORBA interface—TMF.814 Version 2.0. This standard is available to TMF members from the TMF website. The information model allows the EMS to provide an abstraction layer between the EMS and the NMS. This standard is generic and is, therefore, applicable to many different EMS and NE implementations. Customers implementing a TMF-compliant CORBA interface between a specific NMS and Cisco MGM must use the Cisco MGM GateWay/CORBA product, and will require specific information on what application programming interface (API) features are implemented and how these features are to be used.

1.5 Summary

As a state-of-the-art EMS, Cisco MGM can provide a CORBA-based interface between itself and an NMS. This functionality is provided by a separately purchasable option—Cisco MGM GateWay/CORBA. Cisco MGM GateWay/CORBA provides a TMF-compliant CORBA interface in accordance with the TMF NML-EML specification composed of TMF.513, TMF.608, and TMF.814. Cisco MGM GateWay/CORBA includes naming and notification services. Customers can use the bundled naming and notification services or their own services in various configurations. Cisco MGM GateWay/CORBA brings the service provider one step closer to full OSS automation.
NE- and Cisco MGM-Specific Details

This chapter provides specific details about Cisco MGM and supported network elements.

2.1 NE-Specific Details

Cisco MGM supports Cisco MGX 8880/8850 software release 5.0.20:

- Rack—The Cisco MGX 8880/8850 are single rack NEs.
- Shelf—Each Cisco MGX 8880/8850 NE has one shelf per rack.
- Slots—The Cisco MGX 8880/8850 are single shelf NEs with 32 slots. Slots seven and eight are reserved for the controller (PXM45) front card. Slots 15/31 and 16/32 are reserved for SRME front cards.
- Sub-slot—Cisco MGM 8880/8850 slots are divided into sub-slots (for example, bays). Full height cards (AXSM/VXSM/PXM/RPM) will have two sub-slots and half height cards (VISM and SRME) will have one sub-slot. The sub-slot will contain the back card for the corresponding front card.
- PTP—For the Cisco MGX 8880/8850 NEs, all PTPs are bi-directional. PTPs do not support the Service State attribute. PTPs will support the Layer Rate attribute.

2.2 Cisco MGM-Specific Details

This section contains the following Cisco MGM-specific information:

- 2.2.1 Layer Rates, page 2-2
- 2.2.2 Inventory, page 2-2
- 2.2.3 Interfaces, page 2-2
- 2.2.4 Object Details, page 2-3
- 2.2.5 Naming Conventions, page 2-5
2.2.1 Layer Rates

The type of layer rate is a short number. The following layer rates are supported by Cisco MGM:

0 = LR_Not_Applicable, the layer is not relevant,
1 = LR_T1_and_DSI1_1_5M, 1.5Mbit/s async/PDH signal,
2 = LR_T3_and_DS3_45M, 45Mbit/s async/PDH signal,
5 = LR_E1_2M, 2Mbit/s PDH signal,
7 = LR_E3_34M, 34Mbit/s PDH signal,
25 = LR_Line_OC3_STS3_and_MS_STM1,
26 = LR_Line_OC12_STS12_and_MS_STM4,
27 = LR_Line_OC48_STS48_and_MS_STM16,
28 = LR_Line_OC192_STS192_and_MS_STM64,
61 = LR_Fast_Ethernet, Fast Ethernet (legacy),
68 = LR_Gigabit_Ethernet, Gigabit ethernet,
72 = LR_DSR_OC1_STM0,

2.2.2 Inventory

The Cisco MGM 8880/8850 is a single shelf NE. Future expansion may include multi-shelf, multi-bay NEs. Considering this, for Cisco MGX 8880/8850 inventory, Cisco MGM always reports Rack = 1 and Shelf = 1, which will allow Cisco MGM to quickly accommodate future changes.

For the Cisco MGX 8880/8850 NE, Cisco MGM reports slots and peripheral holders as equipment holders contained by the shelf. Any provisioned card is reported as equipment under the slot. Any provisioned peripheral is reported as equipment under the peripheral holder. Cisco MGM does not report empty slots as equipment holders; only slots that have provisioned cards are reported as equipment holders.

For unmanaged NEs, Cisco MGM does not report any equipment holder or equipment.

2.2.3 Interfaces

Unless specified otherwise, all getXXX methods are synchronous.

In addition to implementing TMF-defined interfaces, Cisco MGM defines the following new methods on different interfaces:

- `managedElementMgr::ManagedElementMgr_I::forceNEResync`
- `emsSessionFactory::EmsSessionFactory_I::getEmsPublicKey`

Note: These are proprietary methods.
2.2.4 Object Details

2.2.4.1 AVC Events

In certain situations, Cisco MGM can not provide the list of changed object attributes. In this case, Cisco MGM generates an AVC event with an empty attribute list. The NMS needs to retrieve the object details by making an explicit get call.

2.2.4.2 Managed Element and Object Creation Event

If you add a new managed element using the Cisco MGM client, and the name of the new element is incorrect, an object creation event is generated with the incorrect name. However, when Cisco MGM connects to the actual network element, it will also retrieve the correct name. Cisco MGM then generates an attribute value change event to notify the NMS of the change.

2.2.4.3 Source of Object Attributes reported by MGM

This section lists the attributes of objects that Cisco MGM GateWay/CORBA reports. Each attribute is a constant value or is retrieved from the cache or from the NE. The term cache refers to “in-memory” and includes the Cisco MGM database repository and configuration files. Cisco MGM relies on notifications from the NE to update the value of these attributes.

2.2.4.3.1 EMS_T

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reported From</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cache</td>
</tr>
<tr>
<td>nativeEMSName</td>
<td></td>
</tr>
<tr>
<td>emsVersion</td>
<td>constant value</td>
</tr>
<tr>
<td>type</td>
<td></td>
</tr>
</tbody>
</table>

2.2.4.3.2 Equipment_T

Note: For NE-specific details, refer to 3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment, page 3-1.
Table 2-2  Equipment_T Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reported From</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cache</td>
</tr>
<tr>
<td>nativeEMSName</td>
<td></td>
</tr>
<tr>
<td>expectedEquipmentObjectType</td>
<td></td>
</tr>
<tr>
<td>installedEquipmentObjectType</td>
<td></td>
</tr>
<tr>
<td>installedPartNumber</td>
<td></td>
</tr>
<tr>
<td>installedVersion</td>
<td></td>
</tr>
<tr>
<td>installedSerialNumber</td>
<td></td>
</tr>
<tr>
<td>additionalInfo</td>
<td></td>
</tr>
<tr>
<td>serviceState</td>
<td></td>
</tr>
</tbody>
</table>

2.2.4.3.3 EquipmentHolder_T

Note: For NE-specific details, refer to 3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment, page 3-1.

Table 2-3  EquipmentHolder_T Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reported From</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cache</td>
</tr>
<tr>
<td>nativeEMSName</td>
<td></td>
</tr>
<tr>
<td>holderType</td>
<td></td>
</tr>
<tr>
<td>expectedOrInstalledEquipment</td>
<td></td>
</tr>
<tr>
<td>acceptableEquipmentTypeList</td>
<td></td>
</tr>
<tr>
<td>holderState</td>
<td></td>
</tr>
</tbody>
</table>

2.2.4.3.4 TerminationPoint_T

Note: For NE-specific details, refer to 3.1.3 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs, page 3-5.

Table 2-4  TerminationPoint_T Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reported From</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cache</td>
</tr>
<tr>
<td>type</td>
<td></td>
</tr>
<tr>
<td>direction</td>
<td>constant value</td>
</tr>
</tbody>
</table>
2.2.4.3.5 ManagedElement_T

**Note**
For NE-specific details, refer to 3.2.2 `managedElementManager::ManagedElementManager_I::getAllManagedElements`, page 3-14.

### Table 2-5 ManagedElement_T Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reported From</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cache</td>
</tr>
<tr>
<td>nativeEMSName</td>
<td></td>
</tr>
<tr>
<td>location</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td></td>
</tr>
<tr>
<td>productName</td>
<td></td>
</tr>
<tr>
<td>communicationState</td>
<td></td>
</tr>
<tr>
<td>supportedRates</td>
<td></td>
</tr>
<tr>
<td>additionalInfo</td>
<td></td>
</tr>
</tbody>
</table>

2.2.5 Naming Conventions

The following sections describe naming conventions for objects that Cisco MGM reports.

2.2.5.1 ManagedElement

Name is reported by ManagedElement.

2.2.5.2 Physical Termination Point

The PTP name is representative of the position of the PTP with respect to the equipment. For example, port number four on equipment which is in slot 10 and bay 1 will be reported as follows:

/rack=1/shelf=1/slot=10/subslot=1/port=4

For unmanaged NEs, PTP nativeName is a free format string.
2.2.5.3 Cisco MGX 8880/8850 Equipment

The following equipment is supported in Cisco MGM:

- MGX Route Processor Module -XF 512M
- Double-height ATM SM, 16 T3s or E3s
- Double-height ATM SM, 32 T1/E1 Ports
- Double-height ATM SM, 16 OC-3c/STM-1
- Double-height ATM SM, 4 OC-12c/STM-4
- 45 Gbps processor/fabric
- Cisco VXSM T1/E1 Front Card
- Cisco VXSM OC-3/STM-1 Front Card
- Voice Interworking Service Module -PR: 8 T1 Ports
- Voice Interworking Service Module -PR: 8 E1 Ports
- Service Redundancy Module Enhanced
- Ethernet Interface
- SRMEB_STS3
- DC level sensor
- Fan speed sensor
- Power supply
- MGX8880-MG backplane
- Temperature sensor
- Voice Switch Service Module (VXSM) front card, 4 OC3/STM-1
- Voice Switch Service Module (VXSM) front card, 48 T1/E1
- Double-height ATM SM, 1 OC-48c/STM-16
- RPM-PR
- 45 Gbps, switch/cntlr, PXM45 ver C
- Service Redundancy Module Enhanced-B (SRME-B)
- 1.2v, 1.8v, 2.5v, and 3.3v sensors
- Daughter card temperature sensors, 1-8
- Host temperature sensors, 1-8

2.2.5.4 EquipmentHolder

Naming convention is defined by TMF. Numbering is from from left to right and top to bottom. For example, slot 10 will be named as follows:

/rack=1/shelf=1/slot=10

A new equipment holder has been created for the Cisco MGX 8880/8850 family, called peripheral_holder. Since TMF defines that equipment must be in an equipment holder, peripheral_holder has been created to hold the peripherals.
For example, a peripheral_holder present on the shelf will be named as follows:
/rack=1/shelf=1/ /peripheral_holder=17236110

where number 17236110 is the unique ID of the peripheral.
Similarly, a peripheral_holder present on slot 1 will be named as follows:
/rack=1/shelf=1/slot=1/peripheral_holder=17236111

where number 17236111 is the unique ID of the peripheral.

2.2.5 Boolean Mapping

The following Boolean parameters apply:

- On = True
- Off = False

2.2.6 Resource Cleanup

Many TMF interfaces that Cisco MGM implements return iterator objects for large volumes of data. These iterators are allocated Cisco MGM resources. Cisco MGM supports a maximum of 128 iterators. If the iterator limit is reached, the NMS receives an EXCPT_TOO_MANY_OPEN_ITERATORS exception. To avoid this exception, the NMS must invoke the destroy method on the iterator.

2.2.7 Development Environment

Cisco MGM GateWay/CORBA has been developed with Java Development Kit (JDK) 1.4.1 and JacORB object request broker (ORB) 1.4 on Sun Solaris 2.8. JacORB1.4.1.8 is portable object adapter (POA)-based and ORB-compliant with the CORBA 2.3 specification. You can use Java or the C++ IDL compiler to compile IDL files.

**Note**
If you use Java and JacORB, the Cisco MGM server installation provides JAR files for notification and TMF IDLs.

2.2.8 Exception Handling

All interfaces implemented by Cisco MGM GateWay/CORBA raise the exception globaldefs::ProcessingFailureException. The following sections discuss each exception type in detail.

2.2.8.1 EXCPT_NOT_IMPLEMENTED

This exception indicates whether some IDL operations are optional or are not implemented in this release. If the operation itself is not supported, the errorReason is an empty string.

2.2.8.2 EXCPT_INTERNAL_ERROR

This exception indicates an internal EMS error and applies to all methods.
2.2.5.8.3 EXCPT_INVALID_INPUT

This exception indicates an incorrect parameter format, such as a three-level naming termination point (TP) name that is passed as a single-level name. If a parameter is out of range, this exception is also used. The reason field contains the incorrect parameter.

2.2.5.8.4 EXCPT_ENTITY_NOT_FOUND

This exception indicates that the NMS has supplied an object name as a parameter to an operation and the EMS cannot find the object with the given name. The reason field contains the name that was passed as a parameter.

2.2.5.8.5 EXCPT_UNABLE_TO_COMPLY

This exception is used as a generic value when a server cannot respond to the request.

2.2.5.8.6 EXCPT_NE_COMM_LOSS

This exception is used as a generic value when a server cannot communicate with the NE, preventing the successful completion of the operation. All operations that involve communication with the NE might return this exception type.

2.2.5.8.7 EXCPT_ACCESS_DENIED

This exception indicates that an operation has resulted in a security violation. Verify that you have the required access to invoke the operation.

2.2.5.8.8 EXCPT_TOO_MANY_OPEN_ITERATORS

This exception indicates that the EMS has exceeded the number of iterators it can support. Cisco MGM supports 128 iterators. If you receive this exception, it is possible that other sessions are retrieving data in large volume. Wait several minutes; then, retry. Alternately, close other sessions. To avoid this exception, the NMS must invoke the destroy method on iterators.

2.2.5.8.9 EXCPT_USERLABEL_IN_USE

This exception indicates that the userLabel uniqueness constraint cannot be met.
CHAPTER 3

Using Cisco MGM GateWay/CORBA Interfaces

The following sections describe each interface that Cisco MGM supports, including TeleManagement Forum (TMF)-specific and Cisco-proprietary interfaces.

**Note**

Iterator interfaces defined by the TMF are not listed. Cisco MGM implements iterator interfaces according to the TMF definition.

### 3.1 equipment::EquipmentInventoryMgr_I

This interface accesses operations that deal with equipment. A handle to an instance of this interface is gained through the getManager operation in emsSession. This interface is inherited from common::Common_I.

```c
interface EquipmentInventoryMgr_I:common::Common_I
```

This interface implements the getCapabilities method. Operations described in the following sections are returned to the NMS when it invokes getCapabilities.

#### 3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment

**Synopsis**

```c
void getAllEquipment(in globaldefs::NamingAttributes_T meOrHolderName,
                     in unsigned long how_many,
                     out EquipmentOrHolderList_T eqList,
                     out EquipmentOrHolderIterator_I eqIt)
      raises(globaldefs::ProcessingFailureException);
```

**Description**

This operation allows an NMS to request all the equipment and equipment holders contained in an ME or equipment holder.
Chapter 3     Using Cisco MGM GateWay/CORBA Interfaces

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meOrHolderName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the ME or equipment holder for which to retrieve contained equipment and equipment holders</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of EquipmentOrHolder_Ts to return in the first batch</td>
</tr>
<tr>
<td>eqList</td>
<td>EquipmentOrHolderList_T</td>
<td>out</td>
<td>First batch of EquipmentOrHolder_Ts</td>
</tr>
<tr>
<td>eqIt</td>
<td>EquipmentOrHolderIterator_I</td>
<td>out</td>
<td>Iterator used to retrieve the remaining EquipmentOrHolder_Ts</td>
</tr>
</tbody>
</table>

The EqList parameter contains a sequence of the EquipmentOrHolder_T data type, which is defined as the union of Equipment_T and EquipmentHolder_T.

Cisco MGM supports Equipment_T as follows:

- `globaldefs::NamingAttributes_T name`
  - Supported. This field has four tuples. The first tuple is for EMS, the second is for ManagedElement, the third is for EquipmentHolder, and the fourth contains Equipment. The value of the fourth tuple is the same as the nativeEMSName field.
- `string userLabel`
  - Not supported.
- `string nativeEMSName`
  - Supported. The naming convention is described in the “2.2.5 Naming Conventions” section on page 2-5.
- `string owner`
  - Not supported.
- `boolean alarmReportingIndicator`
  - Not supported.
- `ServiceState_T serviceState`
  - Not supported.
- `EquipmentObjectType_T expectedEquipmentObjectType`
  - Slot 14, 15, 31, 32 expected equipment object is SRME-B cards.
  - Slot 7, 8 PXM-45 Controller cards.

**Note**: \( \alpha \) represents the specific lambda wavelength as configured on the Client Layer Interface Port (CLIP).

- `EquipmentObjectType_T installedEquipmentObjectType`
  - Supported. The value reported by the NE is inserted in this field.
- `string installedPartNumber`
  - Supported when equipment is installed.
• string installedVersion
  - Supported when equipment is installed.
• string installedSerialNumber
  - Supported when equipment is installed.
• globaldefs::NVSList_T additionalInfo
  - Supported when equipment is installed. Common Language Equipment Identification (CLEI) code is available.

Cisco MGM supports EquipmentHolder_T as follows:
• globaldefs::NamingAttributes_T name
  - Supported. This field has three tuples: one for EMS, one for ManagedElement, and one for EquipmentHolder name.
• string userLabel
  - Not supported.
• string nativeEMSName
  - Supported. The naming convention is defined in the “2.2.5 Naming Conventions” section on page 2-5.
• string owner
  - Not supported.
• boolean alarmReportingIndicator
  - Not supported.
• EquipmentHolderType_T holderType
  - Supported. Valid values are rack, shelf, and slot.
• globaldefs::NamingAttributes_T expectedOrInstalledEquipment
  - Supported. If the slot is empty and not preprovisioned for any equipment, this field is empty.
• EquipmentObjectTypeList_T acceptableEquipmentTypeList
  - Supported for all systems.
• HolderState_T holderState
  - Supported.
• globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR

This exception indicates a nonspecific internal EMS failure.

EXCPT_INVALID_INPUT
This exception indicates that managedElementOrEquipmentName does not reference an ME or an equipment holder.

EXCPT_ENTITY_NOT_FOUND

This exception indicates that managedElementOrEquipmentName references an object that does not exist.

EXCPT_NE_COMM_LOSS

This exception indicates a loss of communication with the NE.

EXCPT_TOO_MANY_OPEN_ITERATORS

This exception indicates that the maximum number of iterators (128) has been reached.

Compliance

TMF-defined.

3.1.2 equipment::EquipmentInventoryMgr_I::getAllEquipmentNames

Synopsis

```c
void getAllEquipmentNames(in globaldefs::NamingAttributes_T meOrHolderName, 
in unsigned long how_many, 
out globaldefs::NamingAttributesList_T nameList, 
out globaldefs::NamingAttributesIterator_I nameIt)
  raises(globaldefs::ProcessingFailureException);
```

Description

This operation has the same behavior as
3.1.2 equipment::EquipmentInventoryMgr_I::getAllEquipmentNames, page 3-4, but returns the object names instead of returning the entire object.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meOrHolderName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>The name of the ME or equipment holder for which to retrieve contained equipment and equipment holders</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>The maximum number of EquipmentOrHolder_Ts to return in the first batch</td>
</tr>
<tr>
<td>nameList</td>
<td>globaldefs::NamingAttributesList_T</td>
<td>out</td>
<td>The first batch of names</td>
</tr>
<tr>
<td>nameIt</td>
<td>globaldefs::NamingAttributesIterator_I</td>
<td>out</td>
<td>The iterator to retrieve the remaining names</td>
</tr>
</tbody>
</table>
Chapter 3      Using Cisco MGM GateWay/CORBA Interfaces

Throses

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR
This exception is raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT
This exception is raised when meOrHolderName does not reference an ME or an equipment holder.

EXCPT_ENTITY_NOT_FOUND
This exception is raised when managedElementOrEquipmentName references object which does not exist.

EXCPT_NE_COMM_LOSS
This exception is raised when communications to managedElement is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS
This exception is raised when maximum number of iterators (128) has been reached.

Compliance
TMF-defined.

3.1.3 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs

Synopsis

void getAllSupportedPTPs(
   in globaldefs::NamingAttributes_T equipmentName,
   in unsigned long how_many,
   out terminationPoint::TerminationPointList_T tpList,
   out terminationPoint::TerminationPointIterator_I tpIt)
   raises(globaldefs::ProcessingFailureException);

Description

This operation allows an NMS to request the PTPs that are directly supported by specific equipment. The PTPs that are returned are those that share their physical layer with the primary equipment (that is, they represent a port on the equipment or are connected by a fiber, wire, and so on).
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equipmentName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the equipment for which to retrieve supported PTPs</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of PTPs to be reported in the first batch</td>
</tr>
<tr>
<td>tpList</td>
<td>terminationPoint::TerminationPointList_T</td>
<td>out</td>
<td>First batch of PTPs</td>
</tr>
<tr>
<td>tpIt</td>
<td>terminationPoint::TerminationPointIterator_I</td>
<td>out</td>
<td>Iterator used to retrieve the remaining PTPs</td>
</tr>
</tbody>
</table>

Cisco MGM supports TerminationPoint_T as follows:

- globaldefs::NamingAttributes_T name
  - Supported. It contains three tuples for PTP as defined by the TMF. The third tuple contains positional information for the PTP. The naming convention is described in the “2.2.5 Naming Conventions” section on page 2-5.

- string userLabel
  - Not supported.

- string nativeEMSName
  - Supported.

- string owner
  - Not supported.

- globaldefs::NamingAttributes_T ingressTrafficDescriptorName
  - Not supported.

- globaldefs::NamingAttributes_T egressTrafficDescriptorName
  - Not supported.

- TPTType_T type
  - Supported. Cisco MGM supports TPT_PTP.

- TPConnectionState_T connectionState
  - Supported. For PTP it is TPCS_NA.

- TerminationMode_T tpMappingMode
  - Not supported.

- Directionality_T direction
  - Supported. PTPs always report D_BIDIRECTIONAL.

- transmissionParameters::LayeredParameterList_T transmissionParams
  - Supported.

- TPProtectionAssociation_T tpProtectionAssociation
  - Not supported.
• boolean edgePoint
  - Supported.
• globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException
EXCPT.INTERNAL_ERROR
This exception indicates a nonspecific internal EMS failure.
EXCPT.INVALID_INPUT
This exception indicates that equipmentName does not reference an equipment object.
EXCPT.ENTITY_NOT_FOUND
This exception indicates that equipmentName references an object that does not exist.
EXCPT.NE_COMM_LOSS
This exception indicates a loss of communication with the NE.
EXCPT.TOO_MANY_OPEN_ITERATORS
This exception indicates that the maximum number of iterators (128) has been reached.

Compliance
TMF-defined.

3.1.4 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPNames

Synopsis

void getAllSupportedPTPNames(
    in globaldefs::NamingAttributes_T equipmentName,
    in unsigned long how_many,
    out globaldefs::NamingAttributesList_T nameList,
    out globaldefs::NamingAttributesIterator_I nameIt)
raises(globaldefs::ProcessingFailureException);

Description

This operation has the same behavior as
3.1.3 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs, page 3-5, but returns the object names instead of returning the entire objects.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equipmentName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>The name of the equipment for which to retrieve supported PTPs</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>The maximum number of PTPs to be reported in the first batch</td>
</tr>
<tr>
<td>nameList</td>
<td>globaldefs::NamingAttributesList_T</td>
<td>out</td>
<td>The first batch of PTP names</td>
</tr>
<tr>
<td>nameIt</td>
<td>globaldefs::NamingAttributesIterator_I</td>
<td>out</td>
<td>The iterator to retrieve the remaining PTP names</td>
</tr>
</tbody>
</table>

### Throws

- **globaldefs::ProcessingFailureException**
  - **EXCPT_INTERNAL_ERROR**
    This exception is raised in case of nonspecific EMS internal failure.
  - **EXCPT_INVALID_INPUT**
    This exception is raised when equipmentName does not reference an equipment object.
  - **EXCPT_ENTITY_NOT_FOUND**
    This exception is raised when equipmentName references object that does not exist.
  - **EXCPT_NE_COMM_LOSS**
    This exception is raised when communications to managedElement is lost.
  - **EXCPT_TOO_MANY_OPEN_ITERATORS**
    This exception is raised when maximum number of iterators (128) has been reached.

### Compliance

TMF-defined.

### 3.1.5 equipment::EquipmentInventoryMgr_I::getAllSupportingEquipment

#### Synopsis

```c
void getAllSupportingEquipment(
    in globaldefs::NamingAttributes_T ptpName,
    out EquipmentOrHolderList_T eqList
) raises(globaldefs::ProcessingFailureException);
```

#### Description

This operation allows an NMS to request the equipment that contains this PTP.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptpName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the PTP for which to retrieve supporting equipment</td>
</tr>
<tr>
<td>eqList</td>
<td>EquipmentOrHolderList_T</td>
<td>out</td>
<td>List of equipment (not equipment holders) that directly implements the PTP</td>
</tr>
</tbody>
</table>

EquipmentOrHolderList_T is a list of EquipmentOrHolder_T, which is defined as the union of Equipment_T and EquipmentHolder_T structures. For more information about both structures, see the “3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment” section on page 3-1.

Throws

- globaldefs::ProcessingFailureException
- EXCPT_INTERNAL_ERROR
  This exception indicates a nonspecific internal EMS failure.
- EXCPT_INVALID_INPUT
  This exception indicates that ptpName does not reference a PTP object.
- EXCPT_ENTITY_NOT_FOUND
  This exception indicates that ptpName references a PTP object that does not exist.
- EXCPT_NE_COMM_LOSS
  This exception indicates a loss of communication with the NE.
- EXCPT_TOO_MANY_OPEN_ITERATORS
  This exception indicates that the maximum number of iterators (128) has been reached.

Compliance

TMF-defined.

3.1.6 equipment::EquipmentInventoryMgr_I::getAllSupportingEquipmentNames

Synopsis

```java
void getAllSupportingEquipmentNames(
    in globaldefs::NamingAttributes_T ptpName,
    out globaldefs::NamingAttributesList_T nameList)
raises(globaldefs::ProcessingFailureException);
```

Description

This operation performs the same function as 3.1.5 equipment::EquipmentInventoryMgr_I::getAllSupportingEquipment, page 3-8, but returns the object names instead of returning the entire objects.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptpName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>The name of the PTP for which to retrieve the supporting equipment</td>
</tr>
<tr>
<td>nameList</td>
<td>globaldefs::NamingAttributesList_T</td>
<td>out</td>
<td>The list of equipment names (not equipment holders) directly implementing the PTP</td>
</tr>
</tbody>
</table>

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR
This exception is raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT
This exception is raised when ptpName does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND
This exception is raised when ptpName references a PTP object that does not exist.

EXCPT_NE_COMM_LOSS
This exception is raised when communication to managedElement is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS
This exception is raised when maximum number of iterators (128) has been reached.

Compliance
TMF-defined.

3.1.7 equipment::EquipmentInventoryMgr_I::getContainedEquipment

Synopsis

```c
void getContainedEquipment(
    in globaldefs::NamingAttributes_T equipmentHolderName,
    out EquipmentOrHolderList_T equipmentOrHolderList)
raises(globaldefs::ProcessingFailureException);
```

Description

This operation returns the equipment and equipment holders that a specific equipment holder contains. This method differs from getAllEquipment in that it looks only at the next level of the containment hierarchy.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equipmentHolderName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the equipment holder for which to retrieve the directly contained</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>equipment and equipment holders. Cisco MGM also accepts the ME name as a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>value for this parameter.</td>
</tr>
<tr>
<td>equipmentOrHolderList</td>
<td>EquipmentOrHolderList_T</td>
<td>out</td>
<td>Directly contained equipment and equipment holders.</td>
</tr>
</tbody>
</table>

EquipmentOrHolderList_T is a list of EquipmentOrHolder_T, which is defined as the union of Equipment_T and EquipmentHolder_T structures. For information about both structures, see the “3.1.1 equipment::EquipmentInventoryMgr_I::getAllEquipment” section on page 3-1.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR

This exception indicates a nonspecific internal EMS failure.

EXCPT_INVALID_INPUT

This exception indicates that equipmentHolderName does not reference an equipment holder object.

EXCPT_ENTITY_NOT_FOUND

This exception indicates that equipmentHolderName references an equipment holder that does not exist.

EXCPT_NE_COMM_LOSS

This exception indicates a loss of communication with the NE.

Compliance

TMF-defined.

3.1.8 equipment::EquipmentInventoryMgr_I::getEquipment

Synopsis

void getEquipment
(in globaldefs::NamingAttributes_T equipmentOrHolderName,
out EquipmentOrHolder_T equip)
raises(globaldefs::ProcessingFailureException);

Description

This service returns the equipment or equipment holder for the given equipment or equipment holder name.
managedElementManager::ManagedElementManager_I

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equipmentOrHolderName</td>
<td>globaldefs::Naming Attributes_T</td>
<td>in</td>
<td>Name of the equipment or equipment holder to retrieve</td>
</tr>
<tr>
<td>equip</td>
<td>EquipmentOrHolder_T</td>
<td>out</td>
<td>The returned equipment or equipment holder</td>
</tr>
</tbody>
</table>

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR

Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT

Raised when equipmentOrHolderName does not reference an equipment or an equipment holder.

EXCPT_ENTITY_NOT_FOUND

Raised when equipmentOrHolderName references an equipment or equipment holder that does not exist.

EXCPTНЕ_COMM_LOSS

Raised when communication to managedElement is lost.

Compliance

TMF-defined.

3.2 managedElementManager::ManagedElementManager_I

This interface accesses operations that deal with MEs and TPs. A handle to an instance of this interface is gained through the getManager operation in emsSession. This interface is inherited from common::Common_I.

interface EquipmentInventoryMgr_I:common::Common_I

This interface also implements the getCapabilities method. Operations described in the following sections are returned to the NMS when it invokes getCapabilities.
3.2.1 managedElementManager::ManagedElementManager_I::
getAllActiveAlarms

Synopsis

void getAllActiveAlarms(
  in globaldefs::NamingAttributes_T meName,
  in notifications::ProbableCauseList_T excludeProbCauseList,
  in notifications::PerceivedSeverityList_T excludeSeverityList,
  in unsigned long how_many,
  out notifications::EventList_T eventList,
  out notifications::EventIterator_I eventIt)
raises(globaldefs::ProcessingFailureException);

Description

This operation allows an NMS to request all active alarms for the specified ME, excluding alarms that
the ME does not report to Cisco MGM. Some alarms can be filtered out (excluded) by specifying their
probable causes or severities. The result of this operation is independent of the filtering set up by the
NMS for the notification service.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of ME for which to retrieve alarms</td>
</tr>
<tr>
<td>excludeProbCauseList</td>
<td>notifications::ProbableCauseList_T</td>
<td>in</td>
<td>List of probable causes to exclude (for which events should not be reported)</td>
</tr>
<tr>
<td>excludeSeverityList</td>
<td>notifications::PerceivedSeverityList_T</td>
<td>in</td>
<td>List of severities to exclude from the output event list</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of alarms to report in the first batch</td>
</tr>
<tr>
<td>eventList</td>
<td>notifications::EventList_T</td>
<td>out</td>
<td>First batch of alarms</td>
</tr>
<tr>
<td>eventIt</td>
<td>notifications::EventIterator_I</td>
<td>out</td>
<td>Iterator to retrieve the remaining alarms</td>
</tr>
</tbody>
</table>

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR

This exception indicates a nonspecific internal EMS failure.

EXCPT_INVALID_INPUT

This exception indicates that meName does not reference an ME object or that excludeProbCauseList
contains undefined values.

EXCPT_ENTITY_NOT_FOUND
This exception indicates that meName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS

This exception indicates a loss of communication with the NE.

EXCPT_TOO_MANY_OPEN_ITERATORS

This exception indicates that the maximum number of iterators (128) has been reached.

Compliance

TMF-defined.

### 3.2.2 managedElementManager::ManagedElementManager_I::
getAllManagedElements

**Synopsis**

```c++
void getAllManagedElements(
    in unsigned long how_many,
    out managedElement::ManagedElementList_T meList,
    out managedElement::ManagedElementIterator_I meIt)
raises(globaldefs::ProcessingFailureException);
```

**Description**

This operation allows an NMS to request details about all MEs that are under the control of ManagedElementMgr_I.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of MEs to report in the first batch</td>
</tr>
<tr>
<td>meList</td>
<td>managedElement::ManagedElementList_T</td>
<td>out</td>
<td>First batch of MEs</td>
</tr>
<tr>
<td>meIt</td>
<td>managedElement::ManagedElementIterator_I</td>
<td>out</td>
<td>Iterator to retrieve the remaining MEs</td>
</tr>
</tbody>
</table>

Cisco MGM supports ManagedElement_T as follows:
- globaldefs::NamingAttributes_T name
  - Supported.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. The NE reports the name.
- string owner
  - Not supported.
• string location  
  – Supported.
• string version  
  – Supported.
• string productName  
  – Supported.
• CommunicationState_T communicationState  
  – Supported.
• boolean emsInSyncState  
  – Supported only in forceNEResync activity.
• globaldefs::NVSList_T additionalInfo  
  – Supported. There is a Name-Value-Pair for IPAddress with the name “IPAddress.” There is a Name-Value-Pair for Operational State with the name “OperationalState” with following values. The emsInSyncState value is determined by the OperationalState value. A StateChangeEvent is generated whenever the state of emsInSyncState changes.

<table>
<thead>
<tr>
<th>OperationStates</th>
<th>EmsInSyncState</th>
</tr>
</thead>
<tbody>
<tr>
<td>InServiceSyncConfiguration</td>
<td>false</td>
</tr>
<tr>
<td>OutOfService</td>
<td>false</td>
</tr>
<tr>
<td>UnderMaintenance</td>
<td>false</td>
</tr>
<tr>
<td>InServiceIntializing</td>
<td>false</td>
</tr>
<tr>
<td>InService</td>
<td>true</td>
</tr>
</tbody>
</table>

**Throws**

globaldefs::ProcessingFailureException

**EXCPT_INTERNAL_ERROR**

This exception indicates a nonspecific internal EMS failure.

**EXCPT_TOO_MANY_OPEN_ITERATORS**

This exception indicates that the maximum number of iterators (128) has been reached.

**Compliance**

TMF-defined.
3.2.3 managedElementManager::ManagedElementManager_I::
getAllManagedElementNames

Synopsis

```c
void getAllManagedElementNames(
    in unsigned long how_many,
    out globaldefs::NamingAttributesList_T nameList,
    out globaldefs::NamingAttributesIterator_I nameIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This operation has the same behavior as
3.2.2 managedElementManager::ManagedElementManager_I:: getAllManagedElements, page 3-14,
but instead of returning the entire object structures, this operation returns their names.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of MEs to report in the first batch</td>
</tr>
<tr>
<td>nameList</td>
<td>globaldefs::NamingAttributesList_T</td>
<td>out</td>
<td>First batch of MEs</td>
</tr>
<tr>
<td>nameIt</td>
<td>globaldefs::NamingAttributesIterator_I</td>
<td>out</td>
<td>Iterator to retrieve the remaining MEs</td>
</tr>
</tbody>
</table>

Throws

- globaldefs::ProcessingFailureException
- EXCPT_INTERNAL_ERROR
  - This exception is raised in case of nonspecific EMS internal failure.
- EXCPT_TOO_MANY_OPEN_ITERATORS
  - This exception is raised when maximum number of iterators (128) has been reached.

Compliance

TMF-defined.
### 3.2.4 managedElementManager::ManagedElementManager_I::getAllPTPs

#### Synopsis

```c
void getAllPTPs(
    in globaldefs::NamingAttributes_T managedElementName,
    in transmissionParameters::LayerRateList_T tpLayerRateList,
    in transmissionParameters::LayerRateList_T connectionLayerRateList,
    in unsigned long how_many,
    out terminationPoint::TerminationPointList_T tpList,
    out terminationPoint::TerminationPointIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

#### Description

This operation allows an NMS to request all PTPs on a specific ME that contain one or more NMS-specified PTP layer rates, and that are capable of supporting one or more NMS-specified connection layer rates. See 2.2.1 Layer Rates. If no PTPs match the layer constraints, an empty list is returned.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>managedElementName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the ME for which to retrieve PTPs</td>
</tr>
<tr>
<td>tpLayerRateList</td>
<td>transmissionParameters::LayerRateList_T</td>
<td>in</td>
<td>Cisco MGM does not use this parameter</td>
</tr>
<tr>
<td>connectionLayerRateList</td>
<td>transmissionParameters::LayerRateList_T</td>
<td>in</td>
<td>Cisco MGM does not use this parameter</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of PTPs to report in the first batch</td>
</tr>
<tr>
<td>tpList</td>
<td>terminationPoint::TerminationPointList_T</td>
<td>out</td>
<td>First batch of PTPs</td>
</tr>
<tr>
<td>tpIt</td>
<td>terminationPoint::TerminationPointIterator_I</td>
<td>out</td>
<td>Iterator to retrieve remaining PTPs</td>
</tr>
</tbody>
</table>

For information about the TerminationPoint_T structure, see the “3.1.3 equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs” section on page 3-5.

#### Throws

- globaldefs::ProcessingFailureException

  EXCPT_INTERNAL_ERROR

  This exception indicates a nonspecific internal EMS failure.

  EXCPT_INVALID_INPUT

  This exception indicates that managedElementName does not reference an ME object.

  EXCPT_ENTITY_NOT_FOUND
This exception indicates that managedElementName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS

This exception indicates a loss of communication with the NE.

EXCPT_TOO_MANY_OPEN_ITERATORS

This exception indicates that the maximum number of iterators (128) has been reached.

Compliance

TMF-defined.

3.2.5 managedElementManager::ManagedElementManager_I::getAllPTPNames

Synopsis

void getAllPTPNames(
  in globaldefs::NamingAttributes_T managedElementName,
  in transmissionParameters::LayerRateList_T tpLayerRateList,
  in transmissionParameters::LayerRateList_T connectionLayerRateList,
  in unsigned long how_many,
  out globaldefs::NamingAttributesList_T nameList,
  out globaldefs::NamingAttributesIterator_I nameIt)
   raises(globaldefs::ProcessingFailureException);

Description

This operation has the same behavior as 3.2.4 managedElementManager::ManagedElementManager_I::getAllPTPs, page 3-17, but instead of returning the entire object structures, this operation returns their names.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>managedElementName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>The name of the ME for which to retrieve PTPs</td>
</tr>
<tr>
<td>tpLayerRateList</td>
<td>transmissionParameters::LayerRateList_T</td>
<td>in</td>
<td>List of PTP layer rates for which the PTPs are to be retrieved. A PTP must contain at least one of the layer rates specified to be reported. If the list is empty then all PTPs (of all rates) are returned</td>
</tr>
<tr>
<td>connectionLayerRateList</td>
<td>transmissionParameters::LayerRateList_T</td>
<td>in</td>
<td>Cisco MGM does not use this parameter</td>
</tr>
<tr>
<td>how_many</td>
<td>unsigned long</td>
<td>in</td>
<td>Maximum number of PTPs to report in the first batch</td>
</tr>
<tr>
<td>nameList</td>
<td>globaldefs::NamingAttributesList_T</td>
<td>out</td>
<td>First batch of PTP names</td>
</tr>
<tr>
<td>nameIt</td>
<td>globaldefs::NamingAttributesIterator_I</td>
<td>out</td>
<td>Iterator to retrieve remaining PTP names</td>
</tr>
</tbody>
</table>
Chapter 3  Using Cisco MGM GateWay/CORBA Interfaces

managedElementManager::ManagedElementManager_I

3.2.6 managedElementManager::ManagedElementManager_I::
getManagedElement

Synopsis

void getManagedElement(
    in globaldefs::NamingAttributes_T managedElementName,
    out managedElement::ManagedElement_T me)
raises(globaldefs::ProcessingFailureException);

Description

This operation returns the ME for a specific ME name.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>managedElementName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the ME to retrieve</td>
</tr>
<tr>
<td>me</td>
<td>managedElement::ManagedElement_T</td>
<td>out</td>
<td>The retrieved ME</td>
</tr>
</tbody>
</table>

For information about the ManagedElement_T structure, see the “3.2.2 managedElementManager::ManagedElementManager_I:: getAllManagedElements” section on page 3-14.
3.2.7 managedElementManager::ManagedElementManager_I::forceNEResync

Synopsis

void forceNEResync(
    in globaldefs::NamingAttributes_T meName
); raises(globaldefs::ProcessingFailureException);

Description

This method forces Cisco MGM to resynchronize the inventory and alarms of the specified ME. This method returns after basic validation. The ME is then queued for resynchronization. Cisco MGM generates a StateChange event indicating that the emsInSyncState has changed to False. Upon successful completion of resynchronization, Cisco MGM generates a StateChange event indicating that the emsInSyncState attribute has changed to True. An entry is made into Cisco MGM error log if the resynchronization fails.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Input/Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>in</td>
<td>Name of the ManagedElement to be resynched</td>
</tr>
</tbody>
</table>
Throws

globaldefs::ProcessingFailureException:

EXCPT_UNABLE_TO_COMPLY
This exception indicates that an ME is in the initial poll state, or that the ME is already in the
resynchronization state.

EXCPT_ENTITY_NOT_FOUND
This exception indicates that the ME does not exist.

EXCPT_INVALID_INPUT
This exception indicates an invalid ME name.

EXCPT_COMM_LOSS
This exception indicates a loss of communication to the managedElement.

EXCPT_INTERNAL_ERROR
This exception indicates a nonspecific internal EMS failure.

Compliance
Cisco MGM-defined.
managedElementManager::ManagedElementManager_I
Notifications, Fault Management, and Probable Cause Mapping

A.1 Overview of Notifications

The notification service forwards all fault notifications received from managed NEs to the network management system (NMS). The notification service also forwards object creation, object deletion, attribute change, protection switch, and threshold crossing alert (TCA) events. TMF 814 defines the structure and format of attributes for all events and alarms.

*Note*
The Cisco MGM server and the Operations Support System (OSS) host must have Domain Name System (DNS) enabled in order to receive notifications.

Cisco MGM creates the notification channel with the quality of service (QoS) parameters listed in Table A-1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionReliability</td>
<td>BestEffort</td>
</tr>
<tr>
<td>StartTimeSupported</td>
<td>false</td>
</tr>
<tr>
<td>OrderPolicy</td>
<td>FifoOrder</td>
</tr>
<tr>
<td>DiscardPolicy</td>
<td>FifoOrder</td>
</tr>
<tr>
<td>MaxEventsPerConsumer</td>
<td>10000</td>
</tr>
</tbody>
</table>

1. The MaxEventsPerConsumer value is configurable. To modify this value, see the “B.3 Configuring Cisco MGM GateWay/CORBA” section on page B-3. All other parameter values are defined by the TMF and cannot be changed.

Cisco MGM sets the RejectNewEvents admin property of the channel to true, as defined by the TMF.
Every notification that Cisco MGM generates has the QoS parameters listed in Table A-2. These parameter values are defined by the TMF and cannot be changed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventReliability</td>
<td>BestEffort</td>
</tr>
<tr>
<td>Priority</td>
<td>DefaultPriority</td>
</tr>
<tr>
<td>Timeout</td>
<td>30 minutes for alarms and TCAs; 24 hours for all other notifications</td>
</tr>
</tbody>
</table>

The NMS retrieves all active alarms on the Element Management System (EMS) and on managed element objects. The NMS filters alarms based on probable cause and managed element name. Cisco MGM GateWay/CORBA forwards Cisco MGM-specific alarms to the NMS.

For NE-related alarms, Cisco MGM inserts the original probable cause in the alarm’s nativeProbableCause field. The TMF-mapped probable cause is available in the probableCause field.

### A.2 Resolve the Notification Service

Cisco MGM GateWay/CORBA can integrate with external notification services. When Cisco MGM GateWay/CORBA starts, a reference to EventNotifyChannelFactory is resolved. To resolve this object, Cisco MGM GateWay/CORBA attempts the following:

1. Calls resolve_initial_references with the default service name NotificationService. You can change the service name by setting the corbagw.notification.ServiceName property in the corbagw.properties file. The file location is /mgm_server_install_dir/cfg/corbagw.properties.
2. Resolves the object through the naming service. The default naming service entry is services.NotifyChannelFactory. You can change the naming service entry by setting the corbagw.Naming-Context property in the corbagw.properties file.
3. Resolves the object directly through the Interoperable Object Reference (IOR). You can change the URL of the IOR file by setting the corbagw.notification.FactoryIORFile property in the corbagw.properties file. The default location for the IOR file is /mgm_server_install_dir/openfusion/domains/OpenFusion/localhost/NotificationService/NotificationSingleton.ior. You can specify a URL that points to a remote web server.

### A.3 Tips to Improve Notification Throughput

While Cisco MGM GateWay/CORBA and the bundled notification server are developed and tuned to provide high throughput, you might want to take additional steps to improve throughput and avoid any loss of notifications. You should implement the StructuredPushConsumer interface defined by the Object Management Group (OMG) and implement the push_structured_event method. The notification server invokes this method on the NMS.

⚠️ **Caution**

Because all CORBA calls are blocking, the notification server thread will block until the push_structured_event method completes. If you try to perform a time-consuming task before the method completes, you will block the notification server.
To improve throughput, you should receive the event and forward it to another thread for processing. For example:

```java
void push_structured_event(StructuredEvent notification) {
  try {
    some_other_thread_queue.addNotification(notification);
  } catch (Exception ex) {
    // catch all exception and ignore
    ex.printStackTrace();
  }
}
```

Caution
As shown in the example, make sure to catch all exceptions. If you do not catch all exceptions, you will stop receiving notifications.

## A.4 Supported Events

### A.4.1 Object Creation

The following tables list the object creation types and attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationID are not guaranteed.</td>
</tr>
<tr>
<td>objectName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>Yes</td>
</tr>
<tr>
<td>objectType</td>
<td>notifications::ObjectType_T</td>
<td>Yes</td>
</tr>
<tr>
<td>emsTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>neTime</td>
<td>globaldefs::Time_T</td>
<td>No</td>
</tr>
<tr>
<td>edgePoint</td>
<td>boolean</td>
<td>No</td>
</tr>
</tbody>
</table>

Object creation notifications are sent to the NMS for the objects in Table A-4.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Type Sent in remainder_of_body for Object Creation Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>OT_MANAGED_ELEMENT</td>
<td>managedElement::ManagedElement_T</td>
</tr>
<tr>
<td>OT_EQUIPMENT</td>
<td>equipment::Equipment_T</td>
</tr>
<tr>
<td>OT_PTP</td>
<td>terminationPoint::TerminationPoint_T</td>
</tr>
</tbody>
</table>
Appendix A      Notifications, Fault Management, and Probable Cause Mapping

The number of equipment holders for NEs supported by Cisco MGM is constant. To reduce traffic between Cisco MGM and the NMS, Cisco MGM does not generate object creation events for equipment holders. The number of PTPs on EQUIPMENT is also constant. Therefore, Cisco MGM generates only OBJECT_CREATION events for EQUIPMENT.

A.4.2 Object Deletion

The following table lists the object deletion types and attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationID are not guaranteed.</td>
</tr>
<tr>
<td>objectName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>Yes</td>
</tr>
<tr>
<td>objectType</td>
<td>notifications::ObjectType_T</td>
<td>Yes</td>
</tr>
<tr>
<td>emsTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>neTime</td>
<td>globaldefs::Time_T</td>
<td>No</td>
</tr>
<tr>
<td>edgePoint</td>
<td>boolean</td>
<td>No</td>
</tr>
</tbody>
</table>

Object deletion notifications are sent to the NMS for the objects in Table A-6.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Type Sent in remainder_of_body for Object Deletion Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>OT_MANAGED_ELEMENT</td>
<td>managedElement::ManagedElement_T</td>
</tr>
<tr>
<td>OT_EQUIPMENT</td>
<td>equipment::Equipment_T</td>
</tr>
<tr>
<td>OT_PTP</td>
<td>terminationPoint::TerminationPoint_T</td>
</tr>
</tbody>
</table>

Note

The number of equipment holders for all NEs supported by Cisco MGM is constant. To reduce traffic between Cisco MGM and the NMS, Cisco MGM does not generate object deletion events for equipment holders. The number of PTPs on EQUIPMENT is also constant. Therefore, Cisco MGM generates only OBJECT_DELETION events for EQUIPMENT.
A.4.3 Attribute Value Change

The following tables list the attribute value change (AVC) types and attributes.

**Table A-7** Attribute Value Change

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationID are not guaranteed.</td>
</tr>
<tr>
<td>objectName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>Yes</td>
</tr>
<tr>
<td>objectType</td>
<td>notifications::ObjectType_T</td>
<td>Yes</td>
</tr>
<tr>
<td>emsTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>neTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>edgePoint</td>
<td>boolean</td>
<td>No</td>
</tr>
<tr>
<td>attributeList</td>
<td>notifications::NVList_T</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Attribute value change notifications are sent to the NMS for the objects and attributes in Table A-8.

**Table A-8** Attribute Value Change Notifications

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OT_EMS</td>
<td>nativeEMSName</td>
</tr>
<tr>
<td>OT_MANAGED_ELEMENT</td>
<td>nativeEMSName</td>
</tr>
<tr>
<td></td>
<td>location</td>
</tr>
<tr>
<td></td>
<td>version</td>
</tr>
<tr>
<td></td>
<td>productName</td>
</tr>
<tr>
<td>OT_EQUIPMENT</td>
<td>equipment::Equipment_T</td>
</tr>
<tr>
<td>OT_PTP</td>
<td>terminationPoint::TerminationPoint_T</td>
</tr>
</tbody>
</table>

A.4.4 State Change Event

The following tables list the state change event types and attributes.

**Table A-9** State Change Event

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationID are not guaranteed.</td>
</tr>
<tr>
<td>objectName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>Yes</td>
</tr>
<tr>
<td>objectType</td>
<td>notifications::ObjectType_T</td>
<td>Yes</td>
</tr>
<tr>
<td>emsTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>neTime</td>
<td>globaldefs::Time_T</td>
<td>No</td>
</tr>
</tbody>
</table>
Table A-9  State Change Event (continued)

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgePoint</td>
<td>boolean</td>
<td>No</td>
</tr>
<tr>
<td>attributeList</td>
<td>notifications::NVList_T</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Attribute state change notifications are sent to the NMS for the objects and attributes in Table A-10.

Table A-10  State Change Event Notifications

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OT_MANAGED_ELEMENT</td>
<td>communicationState—Cisco MGM supports following values:</td>
</tr>
<tr>
<td></td>
<td>• CS_AVAILABLE</td>
</tr>
<tr>
<td></td>
<td>• CS_UNAVAILABLE</td>
</tr>
<tr>
<td></td>
<td>emsInSyncState—Cisco MGM will support following values:</td>
</tr>
<tr>
<td></td>
<td>• True</td>
</tr>
<tr>
<td></td>
<td>• False</td>
</tr>
</tbody>
</table>

A.4.5 Alarm

Table A-11 lists the most common alarm messages.

Table A-11  Alarm

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationId are not guaranteed.</td>
</tr>
<tr>
<td>objectName</td>
<td>globaldefs::NamingAttributes_T</td>
<td>Yes</td>
</tr>
<tr>
<td>nativeEMSName</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>nativeProbableCause</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>objectType</td>
<td>notifications::ObjectType_T</td>
<td>Yes</td>
</tr>
<tr>
<td>emsTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>neTime</td>
<td>globaldefs::Time_T</td>
<td>Yes</td>
</tr>
<tr>
<td>isClearable</td>
<td>boolean</td>
<td>Yes. The value is always set to True.</td>
</tr>
<tr>
<td>layerRate</td>
<td>transmissionParameters::LayerRate_T</td>
<td>No</td>
</tr>
<tr>
<td>probableCause</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>probableCauseQualifier</td>
<td>string</td>
<td>Yes. Cisco MGM provides the same value as in the ProbableCause field.</td>
</tr>
<tr>
<td>perceivedSeverity</td>
<td>notifications::PerceivedSeverity_T</td>
<td>Yes</td>
</tr>
<tr>
<td>serviceAffecting</td>
<td>notifications::ServiceAffecting_T</td>
<td>Yes</td>
</tr>
</tbody>
</table>
A.4.6 Heartbeat Event

Cisco MGM generates a periodic heartbeat event and pushes the event to the notification channel. By default, this function is disabled. You can configure the interval value through Cisco MGM Client Control Panel. If the interval value is set to 0, the heartbeat event is disabled. By monitoring this event, the NBI client knows whether the notification service is active. The type_name field contains the NT_HEART_BEAT value under the fixed_header for this type of event. Filterable_data has only one field, which is shown in Table A-12.

### Table A-12 Heartbeat Event

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Supported by Cisco MGM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>notificationId</td>
<td>string</td>
<td>Yes. The uniqueness and sequence of the notificationID are not guaranteed.</td>
</tr>
</tbody>
</table>

A.5 Probable Cause Mapping

A.5.1 TMF-Defined Probable Cause

The following report lists the probable causes defined by the TMF:

- "UNIDENTIFIED": for alarms that do not match any other string below. EMS shall in this case fill out the additional text field as much as possible.
- "AIS": alarm indication signal
- "AMS": Alternate modulation signal
- "AU-AIS": AU alarm indication signal
- "BER_SD": signal degrade (includes receiver degrade)
- "BER_SF": signal fail (includes receiver failure and excessive BER)
- "DCC_FAILURE": Data Communication Channel Failure
- "EMS": EMS system alarm
- "EMS_ALM_LOSS": The 1st notification that the EMS may supply after 1 or more notifications for protection switch, TCA, alarm, or file transfer status have been discarded by the EMS while other events have not been discarded
- "EMS_LIFECYCLE_LOSS": The 1st notification that an EMS may supply after 1 or more events of type OC/OD/AVC/SC/RC have been discarded by the EMS
Probable Cause Mapping

“EMS_ALM_AND_LIFECYCLE_LOSS”: The 1st notification that an EMS may supply after 1 or more
notifications for protection switch, TCA, alarm, or file transfer status, and 1 or more
events of type OC/OD/AVC/SC/RC have been discarded by the EMS
“EQPT”: equipment alarm
“ENV” Environmental/external cause
“FOP_APS”: failure of APS protocol
“LCD”: Loss of Cell Delineation (from TC Adaptor part of ATM NI)
“LOF”: loss of frame (when distinguished from LOS)
“LOM”: loss of multiframe (SDH only, since not an alarm in GR-253)
“LOP”: loss of pointer
“LOS”: loss of signal
“MS_AIS”: MS alarm indication signal
“OSC_AIS”: Optical Supervisory Channel alarm indication signal
“OSC_BER_SF”: Optical Supervisory Channel signal fail/RX fail/excessive BER
“OSC_PERF”: Optical Supervisory Channel Far End Receive Failure
“OSC_LOF”: Optical Supervisory Channel Loss of Frame
“OSC_LOS”: Optical Supervisory Channel Loss of Signal
“OSC_SD”: Optical Supervisory Channel signal degrade
“PLM”: payload label mismatch (when reported as an alarm)
“RAI”: remote alarm indication (sometimes reported as RDI)
“SECURITY_VIOLATION”: security violation
“SSF”: server signal fail
“TCM_AIS”: Tandem Connection Sink - Incoming Alarm Indication Signal
“TCM_LOS”: Tandem Connection Sink - Loss of Tandem Connection Signal
“TCM_OAI”: Tandem Connection Sink - Outgoing Defect Indication (same / similar to Alarm
Indication)
“TCM_RAI”: Tandem Connection Sink - Remote Defect Indication (same / similar to Alarm
Indication)
“TCM_SD”: Tandem Connection Sink - Signal Degrad
“TCM_SSF”: Tandem Connection Sink - Server Signal Fail
“TCM_TIM”: Tandem Connection Sink - Trace Identifier Mismatch
“TCM_UNEQ”: Tandem Connection Sink - Unequipped
“TIM”: trace identifier mismatch (when reported as an alarm)
“TU_AIS”: TU alarm indication signal
“TX_DEGRADE”: transmitter degrade, including laser degrade
“TX_FAIL”: transmitter failure, including laser failure
“UAT”: Unavailable Time
“UNEQ”: payload unequipped
“VC_AIS”: VCL/VCC TP Alarm Indication Signal
“VC_RDI”: VCL/VCC TP Remote Defect Indication
“VP_AIS”: VPL/VPC TP Alarm Indication Signal
“VP_RDI”: VPL/VPC TP Remote Defect Indication
## A.5.2 Alarm Probable Cause Mapping

### Table A-13  Alarm Probable Cause Mapping

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMS Alarms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMS</td>
<td>Loss of Communication</td>
<td>EMS</td>
<td>EMS</td>
</tr>
<tr>
<td></td>
<td>Memory Auto or Manual Backup Failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY_VIOLATION</td>
<td>Maximum Login Attempts Exceeds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMS</td>
<td>Server Monitor Threshold Crossed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A Critical Process is Hanging. Cisco MGM will shutdown in 5 minute</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Communication through Secondary IP Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sync-Up has not started yet</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Currently in Sync-Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Partial Sync-Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sync-Up Failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server in Partial Sync-Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server Sync-Up Failed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Probable Cause Mapping

Following EMS Alarms needs to be cleared manually, so “isClearable” will be set to “False” in corresponding CORBA Alarm Event.

<table>
<thead>
<tr>
<th>Table A-13 Alarm Probable Cause Mapping (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TMF Mapping</strong></td>
</tr>
<tr>
<td>EMS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Shelf Level Peripheral Alarms**

<table>
<thead>
<tr>
<th><strong>EQPT</strong></th>
<th><strong>SHELF</strong></th>
<th><strong>SHELF</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Above normal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Below normal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC power failure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQPT</td>
<td>Front Card not present</td>
<td>CARD</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>Front Card failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in self test</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card held in reset</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in boot</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card unknown alarm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card core card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card failed, no backup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in Update mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Unavailable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Downloading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Downloader</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Downloaded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Locked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in Program mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Upgrading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Upgraded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Frozen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in Hold mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card not responding</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card in initialization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card blocked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Removed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Back Card not present</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Back Card in mismatch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQPT (continued)</td>
<td>Secondary Back Card not present</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Back Card in mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card not present</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in self test</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card held in reset</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in boot</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card unknown alarm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card core card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card failed, no backup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in Update mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Unavailable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Downloading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Downloader</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Downloaded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Locked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in Program mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Upgrading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Upgraded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card Frozen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in Hold mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card not responding</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card in initialization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Card blocked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Front Card Removed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card not present</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in self test</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card held in reset</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in boot</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card unknown alarm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQPT (continued)</td>
<td>Secondary Card core card mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card failed, no backup</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in Update mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Unavailable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Downloading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Downloader</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Downloaded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Locked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in Program mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Upgrading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Upgraded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Frozen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in Hold mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card not responding</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card in initialization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Redundant blocked</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Card Removed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controller in standby mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>License Expired</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>License Missing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Card Sync-Up Failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Redundant Card Switch Over</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor Alarms: VXSM</td>
<td>CARD</td>
<td>CARD</td>
<td></td>
</tr>
<tr>
<td>EQPT</td>
<td>Sensor less than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor greater than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor equal to threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor less than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor greater than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor equal to threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor less than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor greater than threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor equal to threshold value</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tabular data representing probable cause mappings for various scenarios. The table includes specific conditions and their corresponding alarm/event levels.
## Probable Cause Mapping

### Table A-13   Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Level Alarms</td>
<td></td>
<td>LINE</td>
<td>LINE</td>
</tr>
</tbody>
</table>

For **DS1 Line: AXSM, VXSM, VISM, SRME**

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAI</td>
<td>Receiving RAI</td>
</tr>
<tr>
<td></td>
<td>Transmitting RAI</td>
</tr>
<tr>
<td>AIS</td>
<td>Receiving AIS</td>
</tr>
<tr>
<td></td>
<td>Transmitting AIS</td>
</tr>
<tr>
<td>OOF</td>
<td>Receiving OOF</td>
</tr>
<tr>
<td>LOS</td>
<td>Receiving LOS</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Near-End local loop in effect</td>
</tr>
<tr>
<td></td>
<td>Near-End remote loop in effect</td>
</tr>
<tr>
<td></td>
<td>Near-End remote Payload loop in effect</td>
</tr>
<tr>
<td>OOF</td>
<td>Receiving CRC MF OOF</td>
</tr>
<tr>
<td></td>
<td>Receiving Sig MF OOF</td>
</tr>
<tr>
<td>RAI</td>
<td>Receiving MF RAI</td>
</tr>
<tr>
<td>AIS</td>
<td>Receiving MF AIS</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Far-End remote loopback</td>
</tr>
<tr>
<td></td>
<td>Detected Near-End remote loopback</td>
</tr>
</tbody>
</table>

For **DS3 Line: AXSM, VXSM, SRME**

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIDENTIFIED</td>
<td>Receiving RAI</td>
</tr>
<tr>
<td>RAI</td>
<td>Transmitting RAI</td>
</tr>
<tr>
<td>AIS</td>
<td>Receiving AIS</td>
</tr>
<tr>
<td></td>
<td>Transmitting AIS</td>
</tr>
<tr>
<td>OOF</td>
<td>Receiving OOF</td>
</tr>
<tr>
<td>LOS</td>
<td>Receiving LOS</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Near-End local loop in effect</td>
</tr>
<tr>
<td></td>
<td>Near-End remote loop in effect</td>
</tr>
<tr>
<td></td>
<td>Receiving test pattern</td>
</tr>
<tr>
<td></td>
<td>Receiving LOCD</td>
</tr>
<tr>
<td></td>
<td>Receiving C-bit parity pattern</td>
</tr>
</tbody>
</table>

For **SONET Line: AXSM, VXSM, SRME**

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS</td>
<td>Line Receiving AIS</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Line Receiving RDI</td>
</tr>
<tr>
<td>AIS</td>
<td>Line Transmitting AIS</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Line Transmitting RDI</td>
</tr>
<tr>
<td>LOS</td>
<td>Section LOS</td>
</tr>
<tr>
<td>LOF</td>
<td>Section LOF</td>
</tr>
</tbody>
</table>
### Probable Cause Mapping

#### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIDENTIFIED</td>
<td>Section trace ID mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOP</td>
<td>Path Receiving STS LOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIS</td>
<td>Path Receiving STS AIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Path Receiving STS RDI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNEQ</td>
<td>Path Unequipped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Path Signal Label mismatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Path Trace failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIS</td>
<td>Path Transmitting AIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Path Transmitting RDI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Path Signal Receiving LOCD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line Stats Alarm</td>
<td></td>
<td>LINE</td>
<td>LINE</td>
</tr>
<tr>
<td>DS1/DS3: AXSM, VXSM, VISM, SRME</td>
<td>Stats Alarm, PES current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, PES 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Stats Alarm, PSES current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, PSES 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, SEFS current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, SEFS 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, UAS current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, UAS 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, LCV current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, LCV 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, PCV current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, PCV 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, LES current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, LES 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CCV current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CCV 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CES current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CES 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CSES current 15 minute threshold exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stats Alarm, CSES 24 hour threshold exceeded</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SONET: AXSM, VXSM, SRME</td>
<td>UNIDENTIFIED</td>
<td>Stats Alarm, SONET line, Total ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Total SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Total CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Total UAs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Current ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Current SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Current CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Current UAs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Current ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Total ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Total SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Total CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Total UAs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Current SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Current CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET line, Far End, Current UAs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Total ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Total SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Total SEFSs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Total CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Current ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Current SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Current SEFSs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET section, Current CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Total ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Total SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Total CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Total UAs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Current ESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Current SESs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Current CVs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stats Alarm, SONET path, Current UAs</td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TM F Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
</table>
| **UNIDENTIFIED** (continued) | Stats Alarm, SONET path, Far End, Current ESs  
Stats Alarm, SONET path, Far End, Total ESs  
Stats Alarm, SONET path, Far End, Total SESs  
Stats Alarm, SONET path, Far End, Total CVs  
Stats Alarm, SONET path, Far End, Total UAs  
Stats Alarm, SONET path, Far End, Current SESs  
Stats Alarm, SONET path, Far End, Current CVs  
Stats Alarm, SONET path, Far End, Current UASs | LINE              | LINE               |
| Line APS Alarms: AXSM, VXSM, SRME  | APS Alarm, Channel Mismatch  
APS Alarm, ProtectionByte Fail  
APS Alarms, FEProtection Failure  
APS Alarms, Mode Mismatch  
APS Alarms, Signal Degrade  
APS Alarms, Signal Failure  
APS Alarms, Line Alarm  
APS Alarms, Line Loopback  
APS Alarms, Stdby signal Degrade  
APS Alarms, Stdby signal Failure  
APS Alarms, Stdby Line Alarm  
APS Alarms, Stdby Line Loopback  
APS Alarms, Direction Mismatch  
APS Alarms, Wrong Request | LINE              | LINE               |
| EQPT                  | APS Alarms, Protect BC missing  
APS Alarms, Protect DC missing  
APS Alarms, Working BC missing  
APS Alarms, Working DC missing  
APS Alarms, Working Card mismatch  
APS Alarms, Protect Card mismatch |                  |                   |
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOP_APS</td>
<td>APS Alarms, apsRevertiveSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsManulSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsSDLowSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsSDHighSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsSFLowSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsSFHighSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsForceSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apsLockOutSwitchFail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APS Alarms, apslineStatusOk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port Alarms: AXSM</td>
<td></td>
<td>PORT CARD</td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Port, not configured</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port, down/failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port, remote loopback</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port, signaling failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>port, out of cell delineation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPM Port Alarms: RPM, RPM-XF</td>
<td></td>
<td>PORT CARD</td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>RPM Port, Down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPM Port, Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPM Port, Unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPM Port, lower layer down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVC Port Alarms: AXSM, VXSM, RPM-XF, RPM</td>
<td></td>
<td>PORT CARD</td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>SVC Port, failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVC Port, Inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVC Port, lower layer down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISM Port Alarms: VISM</td>
<td></td>
<td>PORT CARD</td>
<td></td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>VISM Port, failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VISM Port, Degraded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VISM Port, lower layer down</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate Port Alarms: AXSM, RPM, RPM-XF</td>
<td>Aggregate Port alarm, One or more connections on this port are in primary failure</td>
<td>PORT</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>Aggregate Port alarm, One or more connections on this port are in secondary failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Aggregate Port alarm, One or more connections on this port are incomplete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection Alarm: AXSM, VXSM, VISM, RPM, RPM-XF</td>
<td>Connection Alarm, Primary Failure</td>
<td>CONNECTION</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>Connection Alarm, Secondary Failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incomplete Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card Level VXSM and VISM Alarms</td>
<td>CAC Resources, Above high threshold value</td>
<td>CARD</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CAC Resources, Below low threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CAC Resources, Above medium threshold value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media Gateway Link state, down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media Gateway Link state, testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media Gateway Link state, unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUDP Session state, Out Of Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUDP Session state, StandbyIs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUDP Session state, FullIs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUDP Session state, Unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Group state, Out Of Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Group state, Unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Set state, Out Of Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Set state, StandbyIs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Set state, FullI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session Set state, Unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MGC in Redundancy group, inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MGC communication state, inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTP Connection state, failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CID state, failed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Probable Cause Mapping

- `isPropagated = true`
Table A-13  Alarm Probable Cause Mapping (continued)

<table>
<thead>
<tr>
<th>TMF Mapping</th>
<th>NE Native Probable Cause</th>
<th>Alarm/Event Level</th>
<th>Propagated to Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Level VISM and VXSM Alarms</td>
<td></td>
<td>LINE</td>
<td>LINE</td>
</tr>
<tr>
<td>UNIDENTIFIED</td>
<td>Endpoint state, failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Endpoint state, degraded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAPD state, inactive</td>
<td>isPropagated = true for LAPD on VXSM-OC3 card</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAPD Trunk state, inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAPD Trunk state, unknown</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Server Administration and Configuration

B.1 Naming Convention for Published CORBA Objects

Cisco MGM GateWay/CORBA publishes two top-level objects: EMSSessionFactory and NotificationChannel. Cisco MGM creates these objects and registers them with the CORBA name server.

Cisco MGM GateWay/CORBA creates naming contexts under the root as shown in Figure B-1. The last context in the tree must have a different name. To change this value in the Cisco MGM client GUI:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Log into the Cisco MGM client with the appropriate Cisco MGM user access profile.</td>
</tr>
<tr>
<td>Step 2</td>
<td>In the Domain Explorer window, click the Cisco MGM Domain node.</td>
</tr>
<tr>
<td>Step 3</td>
<td>In the Management Domain Properties sheet, click the Identification tab.</td>
</tr>
<tr>
<td>Step 4</td>
<td>In the EMS Domain section, look for EMS ID. The value of this field should be used as the “id” field for context, where “kind” equals “EMS.” The default value is Cisco MGM.</td>
</tr>
</tbody>
</table>
B.2 Using Multiple Naming Servers

Cisco MGM registers with multiple naming servers. Add the following parameters to the
MGM_server_installation_directory/cfg/corbagw.properties config file:

```
corbagw.namingservice.ServerList=ctmc4-u80,ctm7-u60
```
(default is empty value)

```
corbagw.namingservice.RootIORLoc=/namingroot.ior
```
(default value)

The first parameter lists all hosts on which naming service is running. These hosts should be reachable
from the Cisco MGM server host. In addition, the Hypertext Transfer Protocol (HTTP) server must be
running on all naming service hosts. The naming service root Interoperable Object Reference (IOR)
must be published in a file. The location and name of the file are defined by the second parameter.

In addition to these naming service hosts, Cisco MGM registers itself with the local naming service. This
naming service is bundled with Cisco MGM.
B.3 Configuring Cisco MGM GateWay/CORBA

You can configure Cisco MGM GateWay/CORBA properties by modifying the MGM_server_installation_directory/cfg/corbagw.properties config file.

If Cisco MGM GateWay/CORBA is running, changes to the config file do not take effect dynamically. You must restart Cisco MGM GateWay/CORBA for the changes to take effect. The following properties can be changed to modify Cisco MGM GateWay/CORBA functionality:

- **corbagw.notification.ServiceName=NotificationService**
  This property defines the service name that the resolve_initial_reference function uses to get a reference to the notification service.
  The Cisco MGM GateWay/CORBA installation installs the notification service automatically. To use your own notification service, modify this parameter.

  **Tip**
  You do not need to change this parameter if you plan to use the notification service that is bundled with Cisco MGM GateWay/CORBA.

- **corbagw.notification.NamingContext=services/NotifyChannelFactory**
  NamingContext defines the naming context of NotificationService. This property is used when resolve_initial_reference fails to resolve NotificationService. Cisco MGM GateWay/CORBA contacts namingservice to resolve the name context defined in this property. The value of this property must match the value published by your notification server.

  **Tip**
  You do not need to change this parameter if you plan to use the notification service that is bundled with Cisco MGM GateWay/CORBA.

- **corbagw.notification.FactoryIORFile = file://localhost/opt/MGM_install_dir/cfg/NotificationSingleton.ior**
  The FactoryIORFile property defines the path to a text file that contains the IOR of NotificationService. This property is used only after resolve_initial_reference and naming service fail. Cisco MGM GateWay/CORBA opens the file as defined by the URL format in this property and attempts to retrieve the IOR from this file. This parameter lets you run your notification service on a different host to improve performance.

  **Tip**
  You do not need to change this parameter if you plan to use the notification service that is bundled with Cisco MGM GateWay/CORBA.

- **corbagw.namingservice.ServerList=**
  The ServerList property defines where the name servers are running. This property takes a comma-separated list of host names.

- **corbagw.namingservice.RootIORLoc=/namingroot.ior**
  The RootIORLoc property defines the path used to find the naming service IOR on each host defined in ServerList. The complete path is constructed as http://<item_of_Server> <List RootIORLoc>.
Corbagw.notification.QoS.MaxEventsPerConsumer=10000
Cisco MGM GateWay/CORBA uses this property to set the MaxEventsPerConsumer administrative QoS parameter of the notification channel. The notification server uses this property to bound the maximum number of events in a given channel that are allowed to queue at any given time. The default value is 0, where the notification server does not impose a limit on the maximum number of events that can be queued. If no limits are imposed on the queue, the notification server might run out of memory if a client behaves incorrectly. The server must keep all events in memory until they are consumed by all registered consumers.

Caution
Any change to this value should be made with extreme caution. If you set the value too low, the NMS will not receive all notifications. If you set the value too high, the Cisco MGM notification server will run out of memory. The current value is set to handle alarm bursts of 10,000 events per minute.

Corbagw.CTP.getLayeredParameters=false
This property by default is not enabled.

Corbagw.session.listen.port=<port_number>
This property configures the IIOP listening port. The Cisco MGM GateWay/CORBA service listens to CORBA requests on this port. If this property is not specified in the corbagw.properties file, the listener port number is allocated by the Operating System.

B.3.1 Configuring the Notification Service Listening Port
To configure the notification service listening port, use the OpenFusion Manager GUI. Complete the following steps to use the OpenFusion Manager GUI:

Step 1  Change to csh and verify that the DISPLAY environment variable is set.
Step 2  Shut down the Cisco MGM GateWay/CORBA service from the Cisco MGM client Control Panel window. Alternately, use the CLI tool to shut down the Cisco MGM GateWay/CORBA service.
Step 3  Enter the following command from the CLI to stop the notification service process:

```
/opt/MGM_install_dir/openfusion/bin/server -stop NotificationService
```
Step 4  Enter the following command from the CLI to launch the OpenFusion Manager GUI:

```
/opt/MGM_install_dir/openfusion/bin/manager
```
Step 5  The OpenFusion Manager Object Viewer window opens. In the left pane of the Administration Manager tab, select the Domains/OpenFusion/localhost/Notification Service tree node.
Step 6  Select the CORBA tab in the right pane.
Step 7  Enter the port number in the Value field for the Server Port property.
Step 8  Save the change and exit the OpenFusion Manager GUI.
Step 9  Restart Cisco MGM GateWay/CORBA from the Control Panel window. Alternately, use the CLI tool to restart Cisco MGM GateWay/CORBA.
Even after completing Step 1 to Step 9, the port used to establish the initial connection to the notification channel will be different. This is because the host/port information that is encoded in the IOR of the notification channel is that of the JacORB implementation repository (ImR). Therefore, even though the service is starting on the assigned port, it is not visible to the clients because the calls are routed from the ImR. By default, the ImR port is assigned by the ORB. If you want to disable ImR, complete the following steps:

---

**Step 1** Shut down Cisco MGM GateWay/CORBA and the notification service.

**Step 2** Run OpenFusion Manager. From the top menu, click **Tools > JacorbProperties Manager**. On the right side, select the **Implementation Repository Configuration** tab. Try one of following:

1. Uncheck the check box for `jacorb.use_imr`. If this property is off, the ImR is not used.
2. If the ImR is on, but you do not want the ImR address encoded in the server IORs, uncheck the `jacorb.use_imr_endpoint` property.

**Step 3** Save the changes, exit OpenFusion Manager, and restart Cisco MGM GateWay/CORBA.

---

**B.4 Installation**

The Cisco MGM installation program installs the Cisco MGM GateWay/CORBA component, which includes OpenFusion 3.0.4 Notification Service from PrismTech.

IDL files are installed under the `/opt/MGM_install_dir/idl` directory. Refer to the *Cisco Media Gateway Manager Release 5.0 Installation Guide* for more information. Appendix D, “Cisco MGM Gateway/CORBA Supported IDLs” lists the interfaces and IDL methods supported in Cisco MGM.

**B.5 Managing OSS Client Profiles for Cisco MGM GateWay/CORBA**

The following sections describe how to create, modify, and delete an OSS client profile for Cisco MGM GateWay/CORBA.

**B.5.1 Create an OSS Client Profile**

The CORBA gateway authenticates the OSS against a previously created user profile before allowing access to Cisco MGM. You can create up to sixteen OSS client profiles for Cisco MGM GateWay/CORBA sessions. Each OSS profile defines Cisco MGM GateWay/CORBA parameters, such as the OSS profile name, password, and IP address.

OSS client profiles are stored in the Cisco MGM GateWay/CORBA Client Configuration Table.

To create an OSS client profile for Cisco MGM GateWay/CORBA:

---

**Step 1** Log into the Cisco MGM client with the appropriate user access profile.

**Step 2** In the Domain Explorer window, choose **Administration > Control Panel**.

**Step 3** Click the **GateWay/CORBA Service** icon to open the GateWay/CORBA Service property sheet.
Step 4  Click the **OSS Clients** tab and the **New** subtab. Enter the following OSS client information:

- **OSS Profile Name**—The name of the OSS profile.
- **Password**—The password the OSS client uses to log into the Cisco MGM server. Confirm the password in the Confirm Password field.
- **OSS IP Address**—The OSS IP address.

Step 5  Click **Save**; then, click **Yes** in the confirmation dialog box. Changes take effect immediately. A subtab with the new OSS profile name appears.

Step 6  In the Control Panel window, choose **Administration > GW/CORBA Client Configuration Table**. The GW/CORBA Client Configuration Table displays a profile for each OSS client that uses a Cisco MGM GateWay/CORBA service.

**B.5.2 Modify an OSS Client Profile**

To modify an OSS client profile for Cisco MGM GateWay/CORBA:

**Step 1**  Log into the Cisco MGM client with the appropriate user access profile.

**Step 2**  In the Domain Explorer window, choose **Administration > Control Panel**.

**Step 3**  Click the **GateWay/CORBA Service** icon to open the GateWay/CORBA Service property sheet.

**Step 4**  Click the **OSS Clients** tab and click the subtab for the OSS client you want to modify. Modify the following OSS client information, as appropriate:

- **OSS Profile**—Modify the OSS profile.
- **Password**—Modify the password that the OSS client uses to log into the Cisco MGM server.
- **Confirm Password**—Enter the password again for confirmation.
- **OSS IP Address**—Modify the OSS IP server.

**Step 5**  Click **Save**. Changes take effect immediately.

**B.5.3 Delete an OSS Client Profile**

To delete an OSS client profile for Cisco MGM GateWay/CORBA:

**Step 1**  Log into the Cisco MGM client with the appropriate user access profile.

**Step 2**  In the Domain Explorer window, choose **Administration > Control Panel**.

**Step 3**  Click the **GateWay/CORBA Service** icon to open the GateWay/CORBA Service property sheet.

**Step 4**  Click the **OSS Clients** tab and click the subtab for the OSS client you want to delete.
Step 5 Click **Delete** to delete the OSS profile from the Cisco MGM database.

Step 6 Click **OK** in the confirmation dialog box.

**Note** If the OSS is connected to Cisco MGM when the profile is deleted, Cisco MGM does not terminate the OSS session.

---

### B.6 Using Notification Service

Cisco MGM GateWay/CORBA can integrate with external notification services. When Cisco MGM GateWay/CORBA starts, a reference to EventNotifyChannelFactory is resolved. Cisco MGM GateWay/CORBA makes the following attempts to resolve the object:

**Note** If you want to use the notification service supplied with Cisco MGM, do not change the parameters as described in the following procedure.

---

**Step 1** Invoke resolve_initial_references with the default service name of NotificationService to resolve the object. You can change the service name by setting the corbagw.notification.ServiceName property in the corbagw.properties file.

**Step 2** Resolve the object through the naming service. The default naming service entry is services.NotifyChannelFactory. You can change the naming service entry by setting the corbagw.notification.NamingContext property in the corbagw.properties file.

**Step 3** Resolve the object through the Interoperable Object Reference (IOR) directly. You can change the URL of the IOR file by setting the corbagw.notification.FactoryIORFile property in the corbagw.properties file. The default location for the IOR file is file://mgm_server_install_dir/openfusion/domains/OpenFusion/localhost/NotificationService/NotificationSingleton.ior. You can specify a URL that points to a remote web server.

**Note** Naming service uses port 14005 for all Cisco MGM-related naming service instances. The port is hard-coded and can be changed only by changing the startup script jcorbagw.sh.

---

### B.6.1 Develop a Cisco MGM GateWay/CORBA Client

Cisco MGM GateWay/CORBA implements interfaces defined by the TeleManagement Forum’s Multi Technology Network Management (MTNM) group. These interfaces are defined for communication between an NMS and the EMS. The NMS must develop a client application that uses these application programming interfaces (APIs). The following sections describe the tools required for development and provide sample code for writing a client application.
B.6.1.1 Recommended Tools

You can develop the CORBA client on UNIX or PC platforms. Cisco recommends that developers have the following tools:

- Sun Microsystems Java Development Kit (JDK) 1.4.2 (available on www.javasoft.com)
- JacORB 1.4.1.8 or VisiBroker 4.5.1

B.6.1.2 Sample Code for Receiving Notifications

A typical CORBA client application involves the following steps:

Step 1  Initialize the client connection to the object request broker (ORB).
Step 2  Get a reference to an object.
Step 3  Invoke operations on this object.

If you want your client to receive notifications from Cisco MGM GateWay/CORBA, the following additional steps are required:

Step 4  Obtain a reference to an EventChannel object in the notification server in Step 2.
Step 5  Obtain a ConsumerAdmin object reference.
Step 6  Invoke obtain_notification_push_consumer() on the SupplierAdmin object, specifying CosNotifyChannelAdmin::STRUCTURED_EVENT as a parameter. This operation returns a reference to StructuredProxyPushSupplier.
Step 7  Implement an instance of StructuredPushConsumer (defined by OMG).
Step 8  Invoke the operation connect_structured_push_consumer() on the StructuredProxyPushSupplier object to connect the StructuredPushConsumer implementation object to the notification channel.
Step 9  Monitor incoming notifications.

The following subsections show sample Java code for each step listed in the preceding procedure.

Note  The following sample code is for example purposes only; the code might not compile as shown. The sample code does not handle all exceptions.
Appendix B  Server Administration and Configuration

B.6.3 Initialize the Client Connection

// Import OMG packages
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import org.omg.CosNotifyChannelAdmin.*;

public static void main(String[] args) {
    try {
        orb = ORB.init(args, null);
    } catch (SystemException ex) {} // Exception handling
    catch (SystemException ex) {} // Exception handling

B.6.4 Get Reference to Naming Service

// Get Nameservice reference
NamingContext nsRootContext = null;
try {
    org.omg.CORBA.Object objRef =
        orb.resolve_initial_references("NameService");
    nsRootContext = NamingContextHelper.narrow(objRef);
} catch (org.omg.CORBA.ORBPackage.InvalidName inEx) {
    // Exception handling
}

B.6.5 Get Reference to EventChannel

The current release uses the following TMF-defined scheme:

EventChannel notifChannel;
EventChannelHolder chanHolder = new EventChannelHolder();

Try {
    EmsSession.getEventChannel(chanHolder);
} catch (Exception ex){
    // handle exceptions
} notifChannel = chanHolder.value;

B.6.6 Obtain ConsumerAdmin Reference

//retrieve default consumer admin
try {
    ConsumerAdmin cadmin = notifChannel.get_consumeradmin(0);
} catch (AdminNotFound anfSe) {
    // Exception handling
}
Using Notification Service

B.6.1.7 Obtain ProxyPushSupplier

try {
    ProxySupplier baseSupplier =
        cadmin.obtain_notification_push_supplier(
            ClientType.STRUCTURED_EVENT, id);
    structuredProxyPushSupplier =
        StructuredProxyPushSupplierHelper.narrow(baseSupplier);
} catch (AdminLimitExceeded aleEx) {
    // Exception handling
}

B.6.1.8 Implement StructuredPushConsumer

class StructuredPushConsumerImpl extends _StructuredPushConsumerImplBase
{
    StructuredPushConsumerImpl() {
        super();
        System.out.println("StructuredPushConsumerImpl created.");
    }

    public void disconnect_structured_push_consumer() {
        System.out.println("Disconnect structured push consumer.");
    }

    public void push_structured_event(StructuredEvent notification) {
        System.out.println("Received notification.");
    }

    public void offer_change(EventType[] added,
        EventType[] removed)
    throws InvalidEventType {
        System.out.println("Offer changed.");
    }
}

B.6.1.9 Connect StructuredPushConsumerImpl

try {
    structuredProxyPushSupplier.connect_structured_push_consumer(
        new StructuredPushConsumerImpl());
    System.out.println(  "Connected to structured proxy push supplier."");
} catch (AlreadyConnected acEx) {
    // Exception handling
} catch (TypeError teEx) {
    // Exception handling
}
B.6.10 Sample Code for Inventory Retrieval

To retrieve inventory from Cisco MGM GateWay/CORBA, your client must log in using a username and password created on the Cisco MGM client. You can create a maximum of eight user profiles for Cisco MGM GateWay/CORBA. See the “B.6 Using Notification Service” section on page B-7.

You can use the username and password in your code to retrieve inventory, as follows:

- Resolve reference to EMSSessionFactory.
- Retrieve EMSSession by supplying the username and password.
- Query EMSSession and get a list of managers available for inventory. Cisco MGM GateWay/CORBA supports four managers:
  - EMS
  - ManagedElement
  - EquipmentInventory
- Invoke a method to retrieve contained objects.

B.6.11 Get Reference to EMSSessionFactory

Follow the example in the “B.6.1.4 Get Reference to Naming Service” section on page B-9 to get a reference of Naming Service.

```java
name[0] = new NameComponent(classString,"Class");
name[1] = new NameComponent(vendorString,"Vendor");
name[2] = new NameComponent(emsInstanceString,"EMSInstance");
name[4] = new NameComponent(ctime_sys_id,"EMS");
name[5] = new NameComponent("SessionFactory","");
try {
    org.omg.CORBA.Object obj = nsRootContext.resolve(name);
} catch (InvalidName inEx) {
    // Exception handling
} catch (NotFound nfEx) {
    // Exception handling
}
```

B.6.12 Log In and Retrieve EmsSession

```java
EmsSessionFactory_I m_emsFactory = EmsSessionFactory_IHelper.narrow(obj);
NmsSession_I csession = new _tie_Session_I(new NmsSessionImpl());
orb.connect(ccession);
EmsSession_IHolder sessionHolder = new EmsSession_IHolder();
m_emsFactory.getEmsSession("<userid>",
"<password>",
ccession,
sessionHolder);
EmsSession_I emsSession = sessionHolder.value;
```
B.6.1.13 Retrieve List of Managers

```java
managerNames_THolder names = new managerNames_THolder();
emsSession.getSupportedManagers(names);
managers = names.value;
for (i = 0; i < managers.length; i++)
{
    System.out.print("Manager ");
    System.out.print(i);
    System.out.println(" + managers[i]);
}
```

B.6.1.14 Operations on EMS Manager

```java
EMS_T m_ems;
EMS_THolder m_emsHolder = new EMS_THolder();
try {
    Common_IHolder mgrHolder = new Common_IHolder();
    emsSession.getManager("EMS", mgrHolder);
    EMSMgr_I emsMgr = EMSMgr_IHelper.narrow(mgrHolder.value);
    emsMgr.getEMS(m_emsHolder);
} catch (ProcessingFailureException pfe) {
    System.out.println("Processing Exception" + pfe.getMessage());
    pfe.printStackTrace();
}

m_ems = m_emsHolder.value;
System.out.println("Native EMS Name" + m_ems.nativeEMSName);
```

B.7 Encryption Between Cisco MGM and GateWay/CORBA

Cisco MGM R5.0 uses improved encryption of usernames and passwords for network security.
You can set the Cisco MGM Client Control Panel to send encrypted usernames and passwords to
Cisco MGM GateWay/CORBA:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log into the Cisco MGM client with the appropriate user access profile.</td>
</tr>
<tr>
<td>2</td>
<td>In the Domain Explorer window, choose Administration &gt; Control Panel.</td>
</tr>
<tr>
<td>3</td>
<td>Click the GateWay/CORBA Service icon to open the GateWay/CORBA Service property sheet.</td>
</tr>
<tr>
<td>4</td>
<td>Click the Global tab.</td>
</tr>
<tr>
<td>5</td>
<td>Check the Enable Encryption for Username and Password check box.</td>
</tr>
<tr>
<td>6</td>
<td>Click Save; then, click Yes in the confirmation dialog box. Changes take effect immediately.</td>
</tr>
</tbody>
</table>
If the OSS clients enable the encryption feature, they must provide implementation for RSA-based encryption by retrieving the RSA Public Key or the Public Key Pair from GateWay/CORBA and by using cryptographic libraries.

- To obtain the RSA Public Key from Cisco MGM, use the `emsSessionFactory::EmsSessionFactory_I::getEmsPublicKey` API.
- To obtain the RSA Public Key Pair from Cisco MGM, use the `emsSessionFactory::EmsSessionFactory_I::getEmsPublicKeyPair` API.

Cisco MGM returns the string representation of the RSA Public Key or Public Key Pair, encoded in the Base64 encoding scheme. OSS clients should use Base64 decoders to decode the public key and get the byte[] of the public key from the decoded public key string. The byte[] corresponding to the public key represents the key in its primary encoded format (X.509 SubjectPublicKeyInfo). Using this byte[] and cryptographic libraries, the RSA Public Key can be created.

One example of the security provider is Bouncy Castle Provider.

Use the public key to encrypt the username and password. Before passing the encrypted username and password to Cisco MGM for login, OSS clients should encode the encrypted username and password by using Base64 encoders to obtain the string equivalent of the encrypted data.

**Note**

Use cryptographic libraries implementing RSA Public Key encryption supporting the “PKCS #1 v2.0 EME-PKCS1-v1_5 (PKCS #1 v1.5 block type 2), PKCS1Padding” encoding scheme. Cisco MGM does not provide these cryptographic libraries.

### B.8 Using the CLI to Start and Stop Cisco MGM GateWay/CORBA

If you are a Cisco MGM user with administrative privileges, you can start and stop Cisco MGM GateWay/CORBA services from the command line. You need a Cisco MGM username and password for the SysAdmin or SuperUser profile.

- To start a GateWay/CORBA service, enter the following on the command line:
  
  `/opt/MGM_install_dir/bin/gwcorba-start`

- To start a GateWay/CORBA service, enter the following on the command line:
  
  `/opt/MGM_install_dir/bin/gwcorba-stop`

If the GateWay/CORBA service is already running when you enter the `gwcorba-start` script, the script simply exits with the message “GWCORBA already running.”

If the GateWay/CORBA service is already stopped when you enter `gwcorba-stop` script, the script simply exists with the message “GW CORBA not running.”

**Note**

The GateWay/CORBA Service can take up to 60 seconds to initialize after the GUI status has changed to indicate that the service is up. The status is an indication of the successful initiation of the service startup, not successful initialization.
This appendix lists interceptors and use cases to resynchronize the NMS from the EMS.

### C.1 Use Cases to Resynchronize the NMS from the EMS

Table C-1, and Table C-2 list use cases to resynchronize the NMS from the EMS.

#### Table C-1 NMS Retrieves Cisco MGM Information When the GateWay/CORBA Service Is Started or Restarted

<table>
<thead>
<tr>
<th>Information</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The NMS retrieves Cisco MGM information.</td>
</tr>
<tr>
<td>Summary</td>
<td>The NMS retrieves all EMS and ManagedElement alarms and information for existing ManagedElements, and PTPs from GateWay/CORBA interfaces.</td>
</tr>
<tr>
<td>Actor(s)</td>
<td>NMS</td>
</tr>
<tr>
<td>Pre-Conditions</td>
<td>The NMS locates the EmsSessionFactory object and obtains references to EMSMgr_I, ManagedElementMgr_I, and EquipmentInventoryMgr_I interfaces.</td>
</tr>
<tr>
<td>Begins When</td>
<td>The NMS sends a request to retrieve all Cisco MGM information.</td>
</tr>
<tr>
<td>Description</td>
<td>The NMS requests the following information from the GateWay/CORBA interface:</td>
</tr>
<tr>
<td></td>
<td>1. The NMS uses the emsMgr::EMSMgr_I::getAllEMSSystemActiveAlarms interface to request all EMS alarms.</td>
</tr>
<tr>
<td></td>
<td>2. The NMS uses the emsMgr::EMSMgr_I::getAllTopLevelSubnetworks interface to request all multilayer subnetworks in Cisco MGM.</td>
</tr>
<tr>
<td></td>
<td>Cisco MGM returns a list of MultiLayerSubnetwork_T objects with detailed information about a MultiLayerSubnetwork.</td>
</tr>
<tr>
<td></td>
<td>3. For each MultiLayerSubnetwork_T object, the NMS uses the multiLayerSubnetwork::MultiLayerSubnetworkMgr_I::getAllManagedElements interface to request all MEs that belong to the same multilayer subnetwork.</td>
</tr>
<tr>
<td></td>
<td>Cisco MGM returns a list of ManagedElement_T objects with detailed information about a ManagedElement.</td>
</tr>
<tr>
<td></td>
<td>For each ManagedElement object, the NMS executes the use case “NMS (Re)synchronizes information specific to a Managed Element.”</td>
</tr>
<tr>
<td>Ends When</td>
<td>The NMS retrieves all Cisco MGM information.</td>
</tr>
</tbody>
</table>
Use Cases to Resynchronize the NMS from the EMS

Table C-1  NMS Retrieves Cisco MGM Information When the GateWay/CORBA Service Is Started or Restarted (continued)

<table>
<thead>
<tr>
<th>Information</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptions</td>
<td>Refer to the exceptions thrown by the individual interface method.</td>
</tr>
</tbody>
</table>
| Post-Conditions | • The NMS synchronizes with Cisco MGM.  
                   • The NMS registers with Cisco MGM to retrieve notifications related to changes in the managed object on Cisco MGM, new alarms, and TCAs. |
| Traceability | —     |

Table C-2  NMS Resynchronizes Information Specific to a Managed Element

<table>
<thead>
<tr>
<th>Information</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The NMS retrieves Cisco MGM information specific to the ME.</td>
</tr>
<tr>
<td>Summary</td>
<td>The NMS retrieves all PTP information for the specific ME.</td>
</tr>
<tr>
<td>Actor(s)</td>
<td>NMS</td>
</tr>
<tr>
<td>Pre-Conditions</td>
<td>The NMS obtains references to ManagedElementMgr_I, and EquipmentInventoryMgr_I interfaces. The NMS also obtains the ManagedElement name.</td>
</tr>
<tr>
<td>Begins When</td>
<td>The NMS sends a request to retrieve all information about an ME. The NMS identifies the name of the ME for which information is to be retrieved.</td>
</tr>
</tbody>
</table>
| Description | The NMS requests the following information from the GateWay/CORBA interface:  
1. The NMS uses the managedElementManager::ManagedElementManager_I::getAllActiveAlarms interface to request all current alarms on the ME.  
2. The NMS uses the equipment::EquipmentInventoryMgr_I::getAllEquipment interface to request all existing equipment on the ME.  
   Cisco MGM returns a list of EquipmentHolder_T objects with detailed information about equipment holders and equipment on the ME.  
3. For the equipment information retrieved in EquipmentorHolder_T object, the NMS uses the equipment::EquipmentInventoryMgr_I::getAllSupportedPTPs interface to request all the supported PTPs on the equipment.  
   Cisco MGM returns a list of TerminationPoint_T objects with detailed PTP information, such as SDH_SONET port, ServiceState, LineCode, FrameFormat, and so on. |
| Ends When    | The NMS retrieves all ME-specific information. |
| Exceptions   | Refer to the exceptions thrown by the individual interface method. |
| Post-Conditions | The NMS synchronizes information about an ME with Cisco MGM. |
| Traceability | —     |
C.2 Developing a Cisco MGM Gateway/CORBA Client

Cisco MGM GateWay/CORBA implements Application Programming Interfaces (APIs) defined by the TeleManagement Forum's Multi-Technology Network Management (MTNM) group. These APIs are defined for communication between an NMS and the EMS. The NMS must develop a client application that uses these APIs.

The following sections describe the tools required for developing a client application. Sample code is provided.

C.2.1 Recommended Tools

You can develop the CORBA client on a UNIX or PC platform using the following tools:

- Sun Microsystems Java Development Kit (JDK) 1.4.2 (available on www.javasoft.com)
- JacORB 1.4.1.8 or VisiBroker 4.5.1

C.2.2 General Procedure

A typical CORBA client application involves the following steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialize the client connection to the Object Request Broker (ORB).</td>
</tr>
<tr>
<td>2</td>
<td>Get a reference to an object, and invoke operations on this object.</td>
</tr>
<tr>
<td>3</td>
<td>If you want your client to receive notifications from Cisco MGM GateWay/CORBA, the following additional steps are required:</td>
</tr>
<tr>
<td></td>
<td>c. Obtain a ConsumerAdmin object reference.</td>
</tr>
<tr>
<td></td>
<td>d. Invoke <code>obtain_notification_push_consumer()</code> on the SupplierAdmin object, specifying <code>CosNotifyChannelAdmin::STRUCTURED_EVENT</code> as a parameter. This operation returns a reference to StructuredProxyPushSupplier.</td>
</tr>
<tr>
<td></td>
<td>e. Implement an instance of StructuredPushConsumer (defined by OMG).</td>
</tr>
<tr>
<td></td>
<td>f. Invoke the operation <code>connect_structured_push_consumer()</code> on the StructuredProxyPushSupplier object to connect the StructuredPushConsumer implementation object to the notification channel.</td>
</tr>
<tr>
<td></td>
<td>g. Monitor incoming notifications.</td>
</tr>
</tbody>
</table>
C.2.3 Sample Code in Java

The following sections show sample Java code for the steps required to develop a client application.

\[\text{Note}\]

The sample code shown is for example purposes only; the code might not compile as shown. The sample code does not handle all exceptions.

C.2.3.1 Initialize the Client Connection

```java
// Import OMG packages
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import org.omg.CosNotifyChannelAdmin.*;
public static void main(String[] args)
{
    try {
        orb = ORB.init(args, null);
    }
    catch (SystemException ex) {
        // Exception handling
    }
}
```

C.2.3.2 Get Reference to Naming Service

```java
// Get Nameservice reference
NamingContext nsRootContext = null;
try {
    org.omg.CORBA.Object objRef =
    orb.resolve_initial_references("NameService");
    nsRootContext = NamingContextHelper.narrow(objRef);
} catch (org.omg.CORBA.ORBPackage.InvalidName inEx) {
    // Exception handling
}
```

C.2.3.3 Sample Code for Operations on Supported Managers

To perform operations from Cisco MGMGateWay/CORBA, your client must log in using a username and password created on the Cisco MGM client. You can create a maximum of eight user profiles for Cisco MGMGateWay/CORBA. See the “B.6 Using Notification Service” section on page B-7 for more information about client profiles.

You can use the username and password in your code to retrieve inventory. The general procedure follows:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Resolve reference to EMSSessionFactory.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Retrieve EMSSession by supplying the username and password.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Query EMSSession and get a list of managers available for operations.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Invoke the desired method on one of the managers in the list.</td>
</tr>
</tbody>
</table>
C.2.3.3.1 Get Reference to EMSSessionFactory

```java
NameComponent name = new NameComponent[6];
name[0] = new NameComponent(classString,"Class");
name[1] = new NameComponent(vendorString, "Vendor");
name[2] = new NameComponent(emsInstanceString, "EMSInstance");
name[4] = new NameComponent(ctm_sys_id, "EMS");
name[5] = new NameComponent("SessionFactory", "EmsSessionFactory");
try {
    org.omg.CORBA.Object obj = nsRootContext.resolve(name);
} catch (InvalidName inEx) {
    // Exception handling
} catch (NotFound nfEx) {
    // Exception handling
}
```

Follow the example in the “C.2.3.2 Get Reference to Naming Service” section on page C-4 to get a reference to Naming Service.

C.2.3.3.2 Log In and Retrieve EmsSession

```java
EmsSessionFactory_I m_emsFactory = EmsSessionFactory_IHelper.narrow(obj);
NmsSession_I csession = new _tie_Session_I(new NmsSessionImpl());
orb.connect(csession);
EmsSessionFactory_IHelper.sessionHolder = new EmsSessionFactory_IHolder();
m_emsFactory.getEmsSession(\"userid\", \"password\", csession,sessionHolder);
EmsSession_I emsSession = sessionHolder.value;
```

C.2.3.3.3 Retrieve List of Managers

```java
managerNames_THolder names = new managerNames_THolder();
emsSession.getSupportedManagers(names);
managers = names.value;
for (i = 0; i < managers.length; i++)
{
    System.out.print("Manager ");
    System.out.print(i);
    System.out.println(" + managers[i]");
}
```

C.2.3.3.4 Perform getEMS Operation on EMS Manager

```java
EMS_T m_ems;
EMS_THolder m_emsHolder = new EMS_THolder();
try {
    Common_IHolder mgrHolder = new Common_IHolder();
    emsSession.getManager("EMS", mgrHolder);
    EMSMgr_I emsMgr = EMSMgr_IHelper.narrow(mgrHolder.value);
    emsMgr.getEMS(m_emsHolder);
}
```
C.2.3.4 Get Reference to EventChannel

```java
EventChannel notifChannel;
EventChannelHolder chanHolder = new EventChannelHolder();
try {
    emsSession.getEventChannel(chanHolder);
} catch (Exception ex) {
    // handle exceptions
}
notifChannel = chanHolder.value;
```

C.2.3.5 Obtain ConsumerAdmin Reference

```java
// retrieve default consumer admin
try {
    ConsumerAdmin cadmin = notifChannel.get_consumeradmin(0);
} catch (AdminNotFound anfSe) {
    // Exception handling
}
```

C.2.3.6 Obtain ProxyPushSupplier

```java
try {
    ProxySupplier baseSupplier =
    cadmin.obtain_notification_push_supplier(
        ClientType.STRUCTURED_EVENT, id);
    structuredProxyPushSupplier =
    StructuredProxyPushSupplierHelper.narrow(baseSupplier);
} catch (AdminLimitExceeded aleEx) {
    // Exception handling
}
```

C.2.3.7 Implement StructuredPushConsumer

```java
class StructuredPushConsumerImpl extends _StructuredPushConsumerImplBase
{
    StructuredPushConsumerImpl() {
        super();
        System.out.println("StructuredPushConsumerImpl created.");
    }
    public void disconnect_structured_push_consumer() {
        System.out.println("Disconnect structured push consumer.");
    }
    public void push_structured_event(StructuredEvent notification) {
        System.out.println("Received notification.");
    }
    public void offer_change(EventType[] added,
                             EventType[] removed)
    throws InvalidEventType
    {
        System.out.println("Offer changed.");
    }
}```
C.2.3.8 Connect StructuredPushConsumerImpl

```java
try {
structuredProxyPushSupplier.connect_structured_push_consumer(
    new StructuredPushConsumerImpl());
System.out.println("Connected to structured proxy push supplier.");
} catch (AlreadyConnected acEx) {
    // Exception handling
} catch (TypeError teEx) {
    // Exception handling
}
```
Cisco MGM Gateway/CORBA Supported IDLs

This appendix lists the interfaces and IDL methods supported in Cisco MGM.

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>IDL Methods</th>
<th>Supported in Cisco MGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common_I</td>
<td>setNativeEMSName</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>setUserLabel</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>setOwner</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getCapabilities</td>
<td>Yes</td>
</tr>
<tr>
<td>EMSMgr_I</td>
<td>getEMS</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllTopLevelSubnetworks</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllTopLevelSubnetworkNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllTopLevelTopologicalLinks</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllTopLevelTopologicalLinkNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getTopLevelTopologicalLink</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllEMSAndMEActiveAlarms</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllEMSSystemActiveAlarms</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>createTopologicalLink</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>createTopologicalLinkWithYCableProtection</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>deleteTopologicalLink</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllL2Topologies</td>
<td>No</td>
</tr>
<tr>
<td>Interface Name</td>
<td>IDL Methods</td>
<td>Supported in Cisco MGM</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>EquipmentInventoryMgr_I</td>
<td>provisionEquipment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>unprovisionEquipment</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>setAlarmReportingOn</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>setAlarmReportingOff</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainedEquipment</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getEquipment</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllEquipment</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllEquipmentNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllSupportedPTPs</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllSupportedPTPNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllSupportingEquipment</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllSupportingEquipmentNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getEthernetPTPs</td>
<td>No</td>
</tr>
<tr>
<td>Version_I</td>
<td>getVersion</td>
<td>Yes</td>
</tr>
<tr>
<td>EmsSessionFactory_I</td>
<td>getEmsSession</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getEmsPublicKey</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getEmsPublicKeyPair</td>
<td>Yes</td>
</tr>
<tr>
<td>Session_I</td>
<td>ping</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>endSession</td>
<td>Yes</td>
</tr>
<tr>
<td>EmsSession_I</td>
<td>getSupportedManagers</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getManager</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getEventChannel</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>releaseResource</td>
<td>Yes</td>
</tr>
<tr>
<td>Interface Name</td>
<td>IDL Methods</td>
<td>Supported in Cisco MGM</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>ManagedElementMgr_I</td>
<td>getAllManagedElements</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllManagedElementNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getContainingSubnetworkNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllPTPs</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllPTPNames</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getTP</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getManagedElement</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getContainedPotentialTPs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainedPotentialTPNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainingInUseTPs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainingInUseTPNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainedCurrentTPs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainedCurrentTPNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainingTPs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getContainingTPNames</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllActiveAlarms</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>setTPData</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllCrossConnections</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllSNCS</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAssociatedTopologicalLinks</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllSrcFTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllDestFTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllSrcPTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllSrcCTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllDestPTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>getAllDestCTPsForSNCProvisioning</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>forceNEResync</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>getAllTopologicalLinksForME</td>
<td>No</td>
</tr>
<tr>
<td>NmsSession_I</td>
<td>eventLossOccurred</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>eventLossCleared</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>historyPMDataCompleted</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>historyPMDataFailed</td>
<td>No</td>
</tr>
<tr>
<td>MultiLayerSubnetworkMgr_I</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Interface Name</td>
<td>IDL Methods</td>
<td>Supported in Cisco MGM</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>EquipmentOrHolderIterator_I</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ManagedElementIterator_I</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>CurrentMaintenanceOperationIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>SubnetworkIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>L2TopologyIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>MLVLANIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>EventIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>PMDataIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>PerformanceManagementMgr_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ProtectionGroupIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ProtectionMgr_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>EmsSessionPolicy</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>CCIIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>SNCIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>TerminationPointIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>TopologicalLinkIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>TrafficDescriptorIterator_I</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>TrafficDescriptorMgr_I</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
INDEX

A

accessing
equipment inventory 3-1
alarms
list of common messages (table) A-6 to A-7
probable causes
TMF-defined A-7 to A-8
retrieving
managed 3-13 to 3-14
synchronizing 3-20 to 3-21
attributes, object. See object attributes
audience for this document vii
AVC events
information about A-5

B

Boolean mapping 2-7

C

cautions
regarding
blocked notification server threads A-2
changing QoS.MaxEventsPerConsumer value B-4
exceptions A-3
significance of viii
Cisco MGM GateWay/CORBA
communications architecture (figure) 1-2
fault management 1-3
features 1-2
information model 1-6
overview 1-1 to 1-6
services 1-3 to 1-6
supported NEs (table) 1-1
Cisco MGM GateWay/CORBA client, developing
connecting StructuredPushConsumerImpl B-10
getting reference to
ConsumerAdmin B-9
EMSSessionFactory B-11
EventChannel B-9
naming service B-9
ProxyPushSupplier B-10
implementing StructuredPushConsumer B-10
initializing the client connection B-9
logging in and retrieving EmsSession B-11
operations on EMS Manager B-12
overview B-7
recommended tools B-8
retrieving a list of managers B-12
sample code
for inventory retrieval B-11
for receiving notifications B-8
Cisco MGM-specific details 2-1
client, initializing connection B-9
configuring Cisco MGM GateWay/CORBA B-3
connecting
StructuredPushConsumerImpl B-10
ConsumerAdmin reference, obtaining B-9
creating
objects A-3 to A-4
OSS client profile for Cisco
MGM GateWay/CORBA B-7
D

deleting
  objects  A-4

development platform for Cisco MGM GateWay/CORBA  2-7
documentation
  audience for this vii
  conventions, typographical viii
  related, obtaining viii
  related to Cisco MGM GateWay/CORBA  1-6

E

EMS
  performing operations on EMS manager  B-12
  session
    logging into  B-11
    retrieving  B-11
  equipment holders
    retrieving  3-10 to 3-12
  equipment inventory manager interface  3-1
  event channels
    getting reference to  B-9
  exceptions
    access denied  2-8
    EMS error  2-7
    IDL operations not implemented  2-7
    invalid input  2-8
    object not found  2-8
    server/NE communications lost  2-8
    server cannot respond  2-8
    too many iterators  2-8
    userLabel in use  2-8

F

fault management, Cisco MGM GateWay/CORBA  1-3

G

getting references
  EMSSessionFactory  B-11
  to event channel  B-9
  to naming service  B-9

I

implementing StructuredPushConsumer  B-10
initializing Cisco MGM GateWay/CORBA client connection  B-9
interfaces, TMF. See TMF interfaces
  inventory
    information  2-2
    retrieving sample code  B-11
    synchronizing  3-20 to 3-21

L

layer rates, description  2-2
logging into an EMS session  B-11

M

managed element manager interface  3-12 to 3-21
managed elements
  naming conventions  2-5
  requesting
    information about  3-14 to 3-16
    resynchronization information  C-2
    retrieving  3-19 to 3-20
  managers, retrieving list of  B-12
  ME. See managed elements

N

naming conventions
  for published CORBA objects  B-1 to B-2
Index

ManagedElement 2-5
physical termination points 2-5
naming services
description 1-3
getting reference to  B-9
on Cisco MGM server 1-4
on independent servers 1-6
on NMS workstation 1-5
notification channel, QoS parameters (table) A-1
notifications
AVC events A-5
improving throughput A-2 to A-3
object creation A-3 to A-4
object deletion A-4
overview A-1
QoS parameters (table) A-2
receiving sample code B-8
resolution with external services A-7 to B-12
state change events A-5
notification services
description 1-3
on Cisco MGM server 1-4
on independent servers 1-6
on NMS workstation 1-5
using B-7

P
physical termination points
naming conventions 2-5
requesting all 3-17 to 3-19
retrieving
all supported by specific equipment 3-5 to 3-10
ProxyPushSupplier, obtaining B-10
PTP. See physical termination points

Q
QoS parameters
for notifications (table) A-2
for the notification channel (table) A-1
quality of service parameters. See QoS parameters

R
receiving notifications (sample code) B-8
references, getting
ConsumerAdmin B-9
EMSSessionFactory B-11
EventChannel B-9
naming service B-9
requesting
managed element information 3-14 to 3-16
physical termination points 3-17 to 3-19
See also retrieving
resource cleanup 2-7
resynchronizing the NMS C-1 to C-3
retrieving
ConsumerAdmin reference B-9
EMS session B-11
equipment holders 3-10 to 3-12
inventory (sample code) B-11
list of managers B-12
managed alarms 3-13 to 3-14
managed elements 3-19 to 3-20
physical termination points
   all supported by specific equipment 3-5 to 3-10
ProxyPushSupplier  B-10
See also requesting
RSA public key, obtaining from Cisco MGM  B-13
RSA public key pair, obtaining from Cisco MGM  B-13

S
services
   naming
      on Cisco MGM server 1-4, 1-5, 1-6
      overview 1-3
   notification
      description 1-3
      on Cisco MGM server 1-4
      on independent servers 1-6
      on NMS workstation 1-5
   state change events A-5
StructuredPushConsumer, implementing B-10
StructuredPushConsumerImpl, connecting B-10
synchronizing
   alarms 3-20 to 3-21
   inventory 3-20 to 3-21

T
Timesavers, significance of viii
Tips, significance of viii
TMF interfaces
   Cisco MGM-specific 2-2
   equipment inventory manager 3-1
   managed element manager 3-12 to 3-21
   using 3-1
TPs. See termination points
typographical conventions used in this document viii