



## Cisco Media Gateway Manager 5.0 GateWay/CORBA User Guide and Programmer Manual

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



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCSP, the Cisco Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Access Registrar, Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, Packet, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StrataView Plus, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0501R)

*Cisco Media Gateway Manager 5.0 GateWay/CORBA User Guide and Programmer Manual*  
Copyright © 1998-2005 Cisco Systems, Inc. All rights reserved.



<b>Preface</b>	<b>vii</b>
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

<b>Overview of Cisco MGM GateWay/CORBA</b>	<b>1-1</b>
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

<b>NE- and Cisco MGM-Specific Details</b>	<b>2-1</b>
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::getAllIPTPs 3-17
  - 3.2.5 managedElementManager::ManagedElementManager\_I::getAllIPTPNames 3-18
  - 3.2.6 managedElementManager::ManagedElementManager\_I::getManagedElement 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**

---

INDEX



## 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:

Item	Convention
Commands and keywords	<b>boldface font</b>
Displayed session and system information	screen font
Information that the user must enter	boldface screen font
Variables that the user must supply	italic screen font
Menu items and button names	boldface font
Selecting a menu item	Option > Network Preferences



## 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](http://www.cisco.com).

**Table 1** *Related Cisco Documentation*

Cisco Product	Document Title
Cisco MGM	<ul style="list-style-type: none"> <li>• <i>Cisco Media Gateway Manager 5.0 User Guide</i></li> <li>• <i>Cisco Media Gateway Manager 5.0 Installation Guide</i></li> <li>• <i>Release Notes for Cisco Media Gateway Manager Release 5.0</i></li> <li>• <i>Cisco Media Gateway Manager 5.0 High Availability Installation Guide</i></li> </ul>
Cisco MGX 8880 and Cisco MGX 8850	All Cisco MGX 8800 documentation.



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.733: Information technology–Open Systems Interconnection–Systems Management: Alarm reporting function
- ITU-T X.734: Information technology–Open Systems Interconnection–Systems Management: Event report 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:

[http://www.cisco.com/public/countries\\_languages.shtml](http://www.cisco.com/public/countries_languages.shtml)

## Ordering Documentation

You can find instructions for ordering documentation at this URL:

[http://www.cisco.com/univercd/cc/td/doc/es\\_inpk/pdi.htm](http://www.cisco.com/univercd/cc/td/doc/es_inpk/pdi.htm)

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:  
<http://www.cisco.com/en/US/partner/ordering/index.shtml>
- 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](mailto: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:

<http://tools.cisco.com/RPF/register/register.do>



### 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>

- World-class networking training is available from Cisco. You can view current offerings at this URL:

<http://www.cisco.com/en/US/learning/index.html>





# Overview of Cisco MGM GateWay/CORBA



## Note

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.

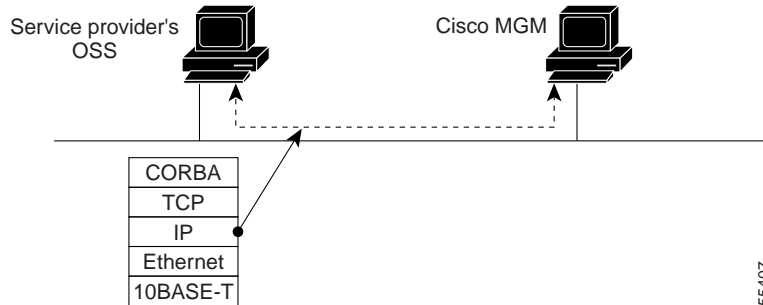
[Table 1-1](#) shows the NE support for Cisco MGM GateWay/CORBA in Cisco MGM R5.0.

**Table 1-1** Cisco MGM GateWay/CORBA Option for MGX NEs

MGX NE	CORBA Support?
MGX 8880	Yes
MGX 8850	Yes

[Figure 1-1](#) illustrates the Cisco MGM GateWay/CORBA communications architecture within a service provider's OSS environment.

**Figure 1-1 Cisco MGM GateWay/CORBA Communications Architecture**



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](http://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

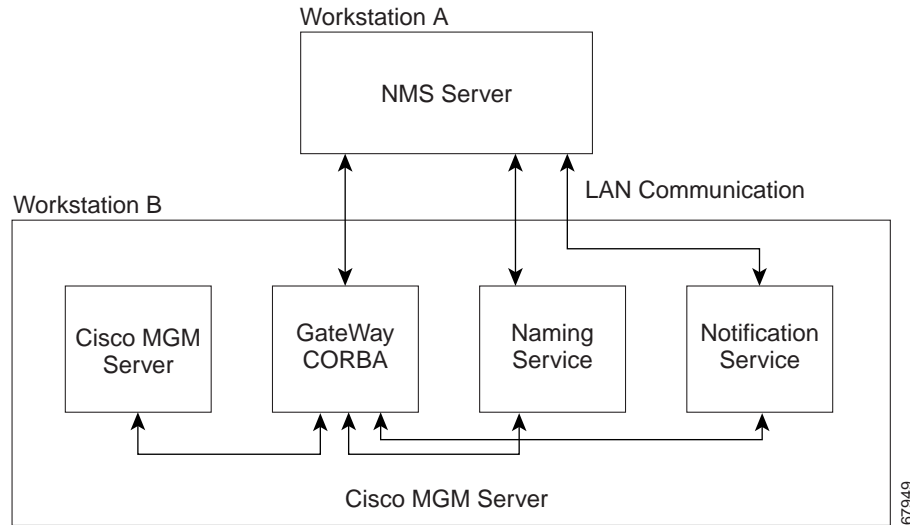
Cisco MGM GateWay/CORBA must use a CORBA notification service to report faults and events. Cisco MGM GateWay/CORBA includes the PrismTech OpenFusion notification service product, and is configured by default to use it. As with the naming service, you can configure Cisco MGM GateWay/CORBA to use a different 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](#).)

**Figure 1-2** Naming and Notification Services on the Cisco MGM Server Workstation

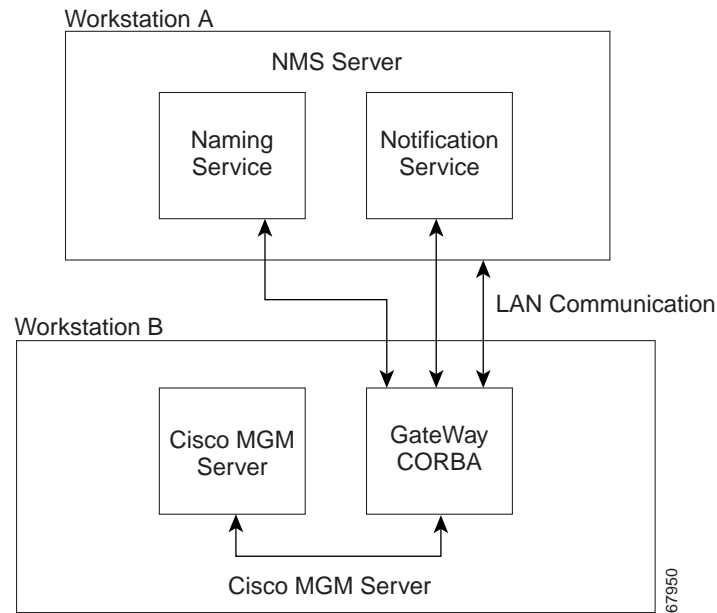


The notification service must be installed on the Cisco MGM server workstation with the Cisco MGM server processes. Cisco MGM GateWay/CORBA will be configured to use this notification service. The NMS processes must be configured to use the notification and naming services on the Cisco MGM server workstation.

## 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

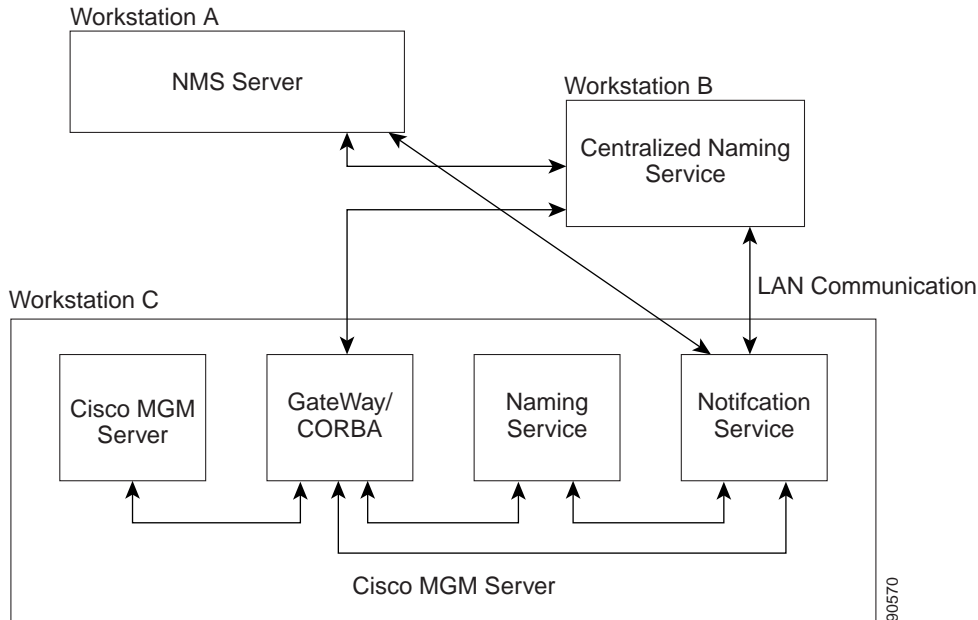


The Cisco MGM server processes and Cisco MGM GateWay/CORBA must be configured to use the naming and notification services on the NMS workstation.

### 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 MGM 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 Not used,
1 = LR_Not_Applicable, the layer is not relevant,
2 = LR_T1_and_DS1_1_5M, 1.5Mbit/s async/PDH signal,
4 = 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 2-1** EMS\_T Attributes

Attribute	Reported From
name	cache
nativeEMSName	
emsVersion	constant value
type	

#### 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*

Attribute	Reported From
name	cache
nativeEMSName	
expectedEquipmentObjectType	
installedEquipmentObjectType	
installedPartNumber	
installedVersion	
installedSerialNumber	
additionalInfo	
service state	

**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*

Attribute	Reported From
name	cache
nativeEMSName	
holderType	
expectedOrInstalledEquipment	
acceptableEquipmentTypeList	
holderState	

**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*

Attribute	Reported From
name	cache
type	
direction	constant value



**Table 2-4** *TerminationPoint\_T Attributes (continued)*

Attribute	Reported From
transmissionParameters	cache
LayeredParameters	

### 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*

Attribute	Reported From
name	cache
nativeEMSName	
location	
version	
productName	
communicationState	
supportedRates	
additionalInfo	

## 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/cntrl, 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.

### 2.2.5.5 Boolean Mapping

The following Boolean parameters apply:

- On = True
- Off = False

### 2.2.5.6 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.

### 2.2.5.7 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. JacORB 1.4.1.8 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.

---

### 2.2.5.8 Exception Handling

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

#### 2.2.5.8.1 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.

#### 2.2.5.8.2 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 namingAttribute 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.



## 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`.

```
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

```
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.

**Parameters**

Name	Type	Input/Output	Description
meOrHolder Name	globaldefs::NamingAttribute s_T	in	Name of the ME or equipment holder for which to retrieve contained equipment and equipment holders
how_many	unsigned long	in	Maximum number of EquipmentOrHolder_Ts to return in the first batch
eqList	EquipmentOrHolderList_T	out	First batch of EquipmentOrHolder_Ts
eqIt	EquipmentOrHolderIterator_ I	out	Iterator used to retrieve the remaining EquipmentOrHolder_Ts

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** *xx* 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

```
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

Name	Type	Input/Output	Description
meOrHolderName	globaldefs::NamingAttributes_T	in	The name of the ME or equipment holder for which to retrieve contained equipment and equipment holders
how_many	unsigned long	in	The maximum number of EquipmentOrHolder_Ts to return in the first batch
nameList	globaldefs::NamingAttributesList_T	out	The first batch of names
nameIt	globaldefs::NamingAttributesIterator_I	out	The iterator to retrieve the remaining names



**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 `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**

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

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**

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

**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**

```
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**

Name	Type	Input/Output	Description
ptpName	globaldefs::NamingAttributes_T	in	Name of the PTP for which to retrieve supporting equipment
eqList	EquipmentOrHolderList_T	out	List of equipment (not equipment holders) that directly implements the PTP

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**

```
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**

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

**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**

```
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**

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

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.

**Parameters**

Name	Type	Input/Output	Description
equipmentOrHolderName	globaldefs::NamingAttributes_T	in	Name of the equipment or equipment holder to retrieve
equip	EquipmentOrHolder_T	out	The returned equipment or equipment holder

**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.

`EXCPT_NE_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

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

### 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

```
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

Name	Type	Input/Output	Description
how_many	unsigned long	in	Maximum number of MEs to report in the first batch
meList	managedElement::ManagedElementList_T	out	First batch of MEs
meIt	managedElement::ManagedElementIterator_I	out	Iterator to retrieve the remaining MEs

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.

OperationStates	EmsInSyncState
InServiceSyncConfiguration	false
OutOfService	false
UnderMaintenance	false
InServiceIntializing	false
InService	true

### 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

```
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

Name	Type	Input/Output	Description
how_many	unsigned long	in	Maximum number of MEs to report in the first batch
nameList	globaldefs::NamingAttributesList_T	out	First batch of MEs
nameIt	globaldefs::NamingAttributesIterator_I	out	Iterator to retrieve the remaining MEs

### 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

```
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

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

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

Name	Type	Input/Output	Description
managedElementName	globaldefs::NamingAttributes_T	in	The name of the ME for which to retrieve PTPs
tpLayerRateList	transmissionParameters::LayerRateList_T	in	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
connectionLayerRateList	transmissionParameters::LayerRateList_T	in	Cisco MGM does not use this parameter
how_many	unsigned long	in	Maximum number of PTPs to report in the first batch
nameList	globaldefs::NamingAttributesList_T	out	First batch of PTP names
nameIt	globaldefs::NamingAttributesIterator_I	out	Iterator to retrieve remaining PTP names

**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 `managedElementName` does not reference an ME object, or `tpLayerRateList` or `connectionLayerRateList` contains undefined rates.

`EXCPT_ENTITY_NOT_FOUND`

This exception is raised when `managedElementName` references an ME 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 the maximum number of iterators (128) has been reached.

**Compliance**

TMF-defined.

## 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**

Name	Type	Input/Output	Description
<code>managedElementName</code>	<code>globaldefs::NamingAttributes_T</code>	in	Name of the ME to retrieve
<code>me</code>	<code>managedElement::ManagedElement_T</code>	out	The retrieved ME

For information about the `ManagedElement_T` structure, see the [“3.2.2 managedElementManager::ManagedElementManager\\_I: getAllManagedElements”](#) section on page 3-14.

**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.

**Compliance**

TMF-defined.

## 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**

Name	Type	Input/Output	Description
meName	globaldefs::NamingAttributes_T	in	Name of the ManagedElement to be resynched



## 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\_NE\_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 A-1** QoS Parameters for the Notification Channel

Parameter	Value
ConnectionReliability	BestEffort
StartTimeSupported	false
OrderPolicy	FifoOrder
DiscardPolicy	FifoOrder
MaxEventsPerConsumer <sup>1</sup>	10000

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 A-2** QoS Parameters for Notifications

Parameter	Value
EventReliability	BestEffort
Priority	DefaultPriority
Timeout	30 minutes for alarms and TCAs; 24 hours for all other notifications

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:

```
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 A-3** Object Creation

Attribute Name	Type	Supported by Cisco MGM?
notificationId	string	Yes. The uniqueness and sequence of the notificationID are not guaranteed.
objectName	globaldefs::NamingAttributes_T	Yes
objectType	notifications::ObjectType_T	Yes
emsTime	globaldefs::Time_T	Yes
neTime	globaldefs::Time_T	No
edgePoint	boolean	No

Object creation notifications are sent to the NMS for the objects in [Table A-4](#).

**Table A-4** Object Creation Notifications

Object Type	Type Sent in remainder_of_body for Object Creation Event
OT_MANAGED_ELEMENT	managedElement::ManagedElement_T
OT_EQUIPMENT	equipment::Equipment_T
OT_PTP	terminationPoint::TerminationPoint_T

**Note**

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 A-5** Object Deletion

Attribute Name	Type	Supported by Cisco MGM?
notificationId	string	Yes. The uniqueness and sequence of the notificationID are not guaranteed.
objectName	globaldefs::NamingAttributes_T	Yes
objectType	notifications::ObjectType_T	Yes
emsTime	globaldefs::Time_T	Yes
neTime	globaldefs::Time_T	No
edgePoint	boolean	No

Object deletion notifications are sent to the NMS for the objects in [Table A-6](#).

**Table A-6** Object Deletion Notifications

Object Type	Type Sent in remainder_of_body for Object Deletion Event
OT_MANAGED_ELEMENT	managedElement::ManagedElement_T
OT_EQUIPMENT	equipment::Equipment_T
OT_PTP	terminationPoint::TerminationPoint_T

**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*

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

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*

Object Type	Attributes
OT_EMS	nativeEMSName
OT_MANAGED_ELEMENT	nativeEMSName location version productName
OT_EQUIPMENT	equipment::Equipment_T
OT_PTP	terminationPoint::TerminationPoint_T

## A.4.4 State Change Event

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

**Table A-9** *State Change Event*

Attribute Name	Type	Supported by Cisco MGM?
notificationId	string	Yes. The uniqueness and sequence of the notificationID are not guaranteed.
objectName	globaldefs::NamingAttributes_T	Yes
objectType	notifications::ObjectType_T	Yes
emsTime	globaldefs::Time_T	Yes
neTime	globaldefs::Time_T	No

**Table A-9** State Change Event (continued)

Attribute Name	Type	Supported by Cisco MGM?
edgePoint	boolean	No
attributeList	notifications::NVList_T	Yes

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

Object Type	Attributes
OT_MANAGED_ELEMENT	<p>communicationState—Cisco MGM supports following values:</p> <ul style="list-style-type: none"> <li>• CS_AVAILABLE</li> <li>• CS_UNAVAILABLE</li> </ul> <p>emsInSyncState—Cisco MGM will support following values:</p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>

## A.4.5 Alarm

[Table A-11](#) lists the most common alarm messages.

**Table A-11** Alarm

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



Table A-11 Alarm (continued)

Attribute Name	Type	Supported by Cisco MGM?
affectedTPList	globaldefs::NamingAttributesList_T	No
additionalText	string	Yes
additionalInfo	globaldefs::NVSList_T	Yes Cisco MGM provides affectedObject when the object type is OT_EMS and the nativeProbableCause is "Server Monitor Threshold Crossed."
isPropagated	boolean	yes
aresFDN	string	yes

## 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

Attribute Name	Type	Supported by Cisco MGM?
notificationId	string	Yes. The uniqueness and sequence of the notificationID are not guaranteed.

## 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
```

"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\_FERF" : 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 Degrade

"TCM-SSF": Tandem Connection Sink - Server Signal Fail

"TCM-TIM" 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

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

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
	Following EMS Alarms needs to be cleared manually, so "isClearable" will be set to "False" in corresponding CORBA Alarm Event.		
EMS	FTP transfer failure		
	FTP file size mismatch		
	FTP transfer failed		
	Upload File Error		
	Minus 2 Trap		
	Card Lost Trap		
	SNMP Retry exceeded		
	FTP Retry exceeded		
	Stats File Error		
	Stats File Transfer Error		
	SNMP Throttle Error		
	Backoff failed		
	SNMP timeout		
	FTP Session timeout		
	FTP Transfer timeout		
	TFTP Transfer timeout		
	Database Error, DELETE failed		
	Database Error, SELECT failed		
	Unknown Error		
	Initialization Error		
	Configuration Error		
	Communication Error in ILOG		
	Communication Error in shared memory		
	Communication Error in CORBA		
	NTS Registration failed		
	NTS Trap loss		
	DataBroker Sync-Up Complete		
<b>Shelf Level Peripheral Alarms</b>		<b>SHELF</b>	<b>SHELF</b>
EQPT	Above normal		
	Below normal		
	Down		
	AC power failure		

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
Card Level Alarms: For all Supported Cards on MGX8880		CARD	CARD
EQPT	Front Card not present		
	Front Card failed		
	Front Card in self test		
	Front Card held in reset		
	Front Card in boot		
	Front Card mismatch		
	Front Card unknown alarm		
	Front Card core card mismatch		
	Front Card failed, no backup		
	Front Card reserved		
	Front Card down		
	Front Card in Update mode		
	Front Card Unavailable		
	Front Card Downloading		
	Front Card Downloader		
	Front Card Downloaded		
	Front Card Locked		
	Front Card in Program mode		
	Front Card Upgrading		
	Front Card Upgraded		
	Front Card Frozen		
	Front Card in Hold mode		
	Front Card not responding		
	Front Card in initialization		
	Front Card blocked		
	Front Card Removed		
Primary Back Card not present			
Primary Back Card in mismatch			

Table A-13 Alarm Probable Cause Mapping (continued)

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

Table A-13 Alarm Probable Cause Mapping (continued)

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

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
Line Level Alarms		LINE	LINE
For DS1 Line: AXSM, VXSM, VISM, SRME			
RAI	Receiving RAI		
	Transmitting RAI		
AIS	Receiving AIS		
	Transmitting AIS		
OOF	Receiving OOF		
LOS	Receiving LOS		
UNIDENTIFIED	Near-End local loop in effect		
	Near-End remote loop in effect		
	Near-End remote Payload loop in effect		
OOF	Receiving CRC MF OOF		
	Receiving Sig MF OOF		
RAI	Receiving MF RAI		
AIS	Receiving MF AIS		
UNIDENTIFIED	Far-End remote loopback		
	Detected Near-End remote loopback		
For DS3 Line: AXSM, VXSM, SRME			
UNIDENTIFIED	Receiving RAI		
RAI	Transmitting RAI		
AIS	Receiving AIS		
	Transmitting AIS		
OOF	Receiving OOF		
LOS	Receiving LOS		
UNIDENTIFIED	Near-End local loop in effect		
	Near-End remote loop in effect		
	Receiving test pattern		
	Receiving LOCD		
	Receiving C-bit parity pattern		
For SONET Line: AXSM, VXSM, SRME			
AIS	Line Receiving AIS		
UNIDENTIFIED	Line Receiving RDI		
AIS	Line Transmitting AIS		
UNIDENTIFIED	Line Transmitting RDI		
LOS	Section LOS		
LOF	Section LOF		



Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
UNIDENTIFIED	Section trace ID mismatch		
LOP	Path Receiving STS LOP		
AIS	Path Receiving STS AIS		
UNIDENTIFIED	Path Receiving STS RDI		
UNEQ	Path Unequipped		
UNIDENTIFIED	Path Signal Label mismatch		
	Path Trace failure		
AIS	Path Transmitting AIS		
UNIDENTIFIED	Path Transmitting RDI		
	Path Signal Receiving LOCD		
Line Stats Alarm		LINE	LINE
DS1/DS3: AXSM, VXSM, VISM, SRME			
UNIDENTIFIED	Stats Alarm, PES current 15 minute threshold exceeded		
	Stats Alarm, PES 24 hour threshold exceeded		
	Stats Alarm, PSES current 15 minute threshold exceeded		
	Stats Alarm, PSES 24 hour threshold exceeded		
	Stats Alarm, SEFS current 15 minute threshold exceeded		
	Stats Alarm, SEFS 24 hour threshold exceeded		
	Stats Alarm, UAS current 15 minute threshold exceeded		
	Stats Alarm, UAS 24 hour threshold exceeded		
	Stats Alarm, LCV current 15 minute threshold exceeded		
	Stats Alarm, LCV 24 hour threshold exceeded		
	Stats Alarm, PCV current 15 minute threshold exceeded		
	Stats Alarm, PCV 24 hour threshold exceeded		
	Stats Alarm, LES current 15 minute threshold exceeded		
	Stats Alarm, LES 24 hour threshold exceeded		
	Stats Alarm, CCV current 15 minute threshold exceeded		
	Stats Alarm, CCV 24 hour threshold exceeded		
	Stats Alarm, CES current 15 minute threshold exceeded		
	Stats Alarm, CES 24 hour threshold exceeded		
	Stats Alarm, CSES current 15 minute threshold exceeded		
	Stats Alarm, CSES 24 hour threshold exceeded		

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
SONET: AXSM, VXSM, SRME			
UNIDENTIFIED	Stats Alarm, SONET line, Total ESs		
	Stats Alarm, SONET line, Total SESs		
	Stats Alarm, SONET line, Total CVs		
	Stats Alarm, SONET line, Total UAs		
	Stats Alarm, SONET line, Current ESs		
	Stats Alarm, SONET line, Current SESs		
	Stats Alarm, SONET line, Current CVs		
	Stats Alarm, SONET line, Current UAs		
	Stats Alarm, SONET line, Far End, Current ESs		
	Stats Alarm, SONET line, Far End, Total ESs		
	Stats Alarm, SONET line, Far End, Total SESs		
	Stats Alarm, SONET line, Far End, Total CVs		
	Stats Alarm, SONET line, Far End, Total UAs		
	Stats Alarm, SONET line, Far End, Current SESs		
	Stats Alarm, SONET line, Far End, Current CVs		
	Stats Alarm, SONET line, Far End, Current UAs		
	Stats Alarm, SONET section, Total ESs		
	Stats Alarm, SONET section, Total SESs		
	Stats Alarm, SONET section, Total SEFSs		
	Stats Alarm, SONET section, Total CVs		
	Stats Alarm, SONET section, Current ESs		
	Stats Alarm, SONET section, Current SESs		
	Stats Alarm, SONET section, Current SEFSs		
	Stats Alarm, SONET section, Current CVs		
	Stats Alarm, SONET path, Total ESs		
	Stats Alarm, SONET path, Total SESs		
	Stats Alarm, SONET path, Total CVs		
	Stats Alarm, SONET path, Total UAs		
	Stats Alarm, SONET path, Current ESs		
	Stats Alarm, SONET path, Current SESs		
	Stats Alarm, SONET path, Current CVs		
	Stats Alarm, SONET path, Current UAs		

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
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 APS Alarms: AXSM, VXSM, SRME		LINE	LINE
FOP_APS	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, Stdbby signal Degrade		
	APS Alarms, Stdbby signal Failure		
	APS Alarms, Stdbby Line Alarm		
	APS Alarms, Stdbby Line Loopback		
	APS Alarms, Direction Mismatch		
	APS Alarms, Wrong Request		
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)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
FOP_APS	APS Alarms, apsRevertiveSwitchFail		
	APS Alarms, apsManulSwitchFail		
	APS Alarms, apsSDLowSwitchFail		
	APS Alarms, apsSDHighSwitchFail		
	APS Alarms, apsSFLowSwitchFail		
	APS Alarms, apsSFHighSwitchFail		
	APS Alarms, apsForceSwitchFail		
	APS Alarms, apsLockOutSwitchFail		
	APS Alarms, apslineStatusOk		
Port Alarms: AXSM		PORT	CARD
		isPropagated = true	
UNIDENTIFIED	Port, not configured		
	Port, down/failure		
	Port, remote loopback		
	Port, signaling failure		
	port, out of cell delineation		
	port, lower layer down		
RPM Port Alarms: RPM, RPM-XF		PORT	CARD
		isPropagated = true	
UNIDENTIFIED	RPM Port, Down		
	RPM Port, Testing		
	RPM Port, Unknown		
	RPM Port, lower layer down		
SVC Port Alarms: AXSM, VXSM, RPM-XF, RPM		PORT	CARD
		isPropagated = true	
UNIDENTIFIED	SVC Port, failed		
	SVC Port, Inactive		
	SVC Port, lower layer down		
VISM Port Alarms: VISM		PORT	CARD
		isPropagated = true	
UNIDENTIFIED	VISM Port, failed		
	VISM Port, Degraded		
	VISM Port, lower layer down		

Table A-13 Alarm Probable Cause Mapping (continued)

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

Table A-13 Alarm Probable Cause Mapping (continued)

TMF Mapping	NE Native Probable Cause	Alarm/Event Level	Propagated to Level
Line Level VISM and VXSM Alarms		LINE	LINE
UNIDENTIFIED	Endpoint state, failed		
	Endpoint state, degraded		
	LAPD state, inactive	isPropagated = true for LAPD on VXSM-OC3 card	
	LAPD Trunk state, inactive		
	LAPD Trunk state, unknown		



## Server Administration and Configuration

---

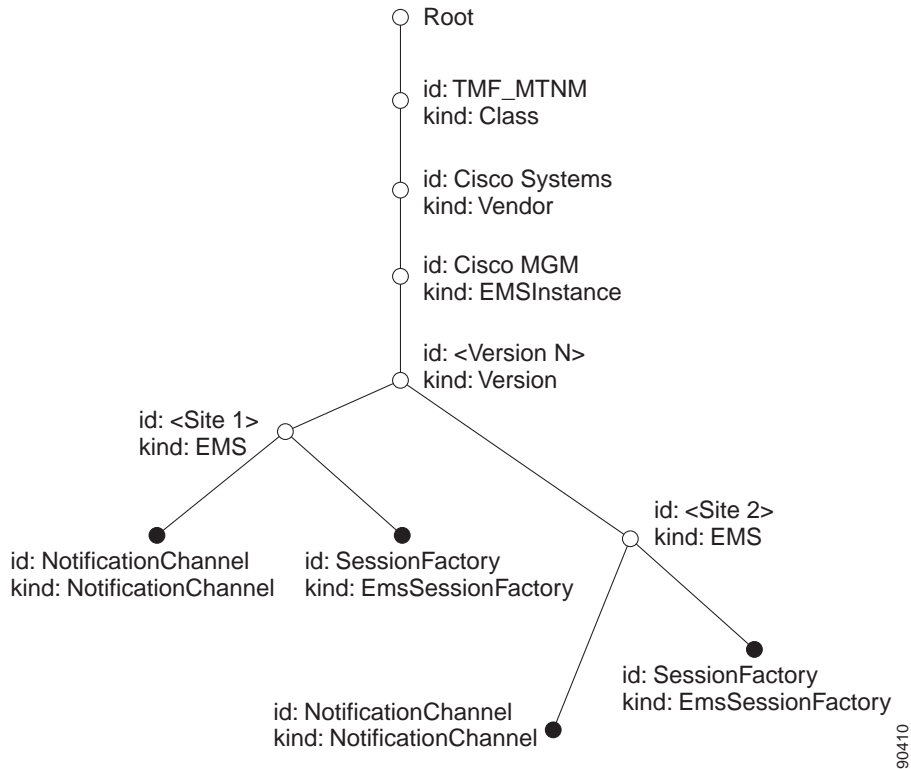
### B.1 Naming Convention for Published CORBA Objects

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:

- 
- Step 1** Log into the Cisco MGM client with the appropriate Cisco MGM user access profile.
  - Step 2** In the Domain Explorer window, click the **Cisco MGM Domain** node.
  - Step 3** In the Management Domain Properties sheet, click the **Identification** tab.
  - Step 4** 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*.
-

Figure B-1 Naming Scheme for Cisco MGM GateWay/CORBA Objects



## 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:
- Uncheck the check box for `jacorb.use_imr`. If this property is off, the ImR is not used.
  - 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](http://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.

---

### B.6.1.3 Initialize the Client Connection

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

import org.omg.CosNotifyChannelAdmin.*;

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

### B.6.1.4 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.6.1.5 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.6.1.6 Obtain ConsumerAdmin Reference

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

### 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.6.1.10 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.6.1.11 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.

```
name[0] = new NameComponent(classString, "Class");
name[1] = new NameComponent(vendorString, "Vendor");
name[2] = new NameComponent(emsInstanceString, "EMSInstance");
name[3] = new NameComponent(versionString, "Version");
name[4] = new NameComponent(ctm_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.6.1.12 Log In and Retrieve EmsSession

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

### B.6.1.13 Retrieve List of Managers

```

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

```

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:

- 
- 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 **Global** tab.
  - Step 5 Check the **Enable Encryption for Username and Password** check box.
  - Step 6 Click **Save**; then, click **Yes** in the confirmation dialog box. Changes take effect immediately.
-

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 stop 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 exits with the message “GWCORBA 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.





## OSS Client Development and Use Cases

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*

Information	Detail
Name	The NMS retrieves Cisco MGM information.
Summary	The NMS retrieves all EMS and ManagedElement alarms and information for existing ManagedElements, and PTPs from GateWay/CORBA interfaces.
Actor(s)	NMS
Pre-Conditions	The NMS locates the EmsSessionFactory object and obtains references to EMSMgr_I, ManagedElementMgr_I, and EquipmentInventoryMgr_I interfaces.
Begins When	The NMS sends a request to retrieve all Cisco MGM information.
Description	<p>The NMS requests the following information from the GateWay/CORBA interface:</p> <ol style="list-style-type: none"> <li>1. The NMS uses the <code>emsMgr::EMSMgr_I::getAllEMSSystemActiveAlarms</code> interface to request all EMS alarms.</li> <li>2. The NMS uses the <code>emsMgr::EMSMgr_I::getAllTopLevelSubnetworks</code> interface to request all multilayer subnetworks in Cisco MGM. Cisco MGM returns a list of <code>MultiLayerSubnetwork_T</code> objects with detailed information about a <code>MultiLayerSubnetwork</code>.</li> <li>3. For each <code>MultiLayerSubnetwork_T</code> object, the NMS uses the <code>multiLayerSubnetwork::MultiLayerSubnetworkMgr_I::getAllManagedElements</code> interface to request all MEs that belong to the same multilayer subnetwork. Cisco MGM returns a list of <code>ManagedElement_T</code> objects with detailed information about a <code>ManagedElement</code>. For each <code>ManagedElement</code> object, the NMS executes the use case “NMS (Re)synchronizes information specific to a Managed Element.”</li> </ol>
Ends When	The NMS retrieves all Cisco MGM information.

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

Information	Detail
Exceptions	Refer to the exceptions thrown by the individual interface method.
Post-Conditions	<ul style="list-style-type: none"> <li>The NMS synchronizes with Cisco MGM.</li> <li>The NMS registers with Cisco MGM to retrieve notifications related to changes in the managed object on Cisco MGM, new alarms, and TCAs.</li> </ul>
Traceability	—

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

Information	Detail
Name	The NMS retrieves Cisco MGM information specific to the ME.
Summary	The NMS retrieves all PTP information for the specific ME.
Actor(s)	NMS
Pre-Conditions	The NMS obtains references to ManagedElementMgr_I, and EquipmentInventoryMgr_I interfaces. The NMS also obtains the ManagedElement name.
Begins When	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.
Description	<p>The NMS requests the following information from the GateWay/CORBA interface:</p> <ol style="list-style-type: none"> <li>The NMS uses the managedElementManager::ManagedElementManager_I::getAllActiveAlarms interface to request all current alarms on the ME.</li> <li>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.</li> <li>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.</li> </ol>
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](http://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:

- 
- Step 1** Initialize the client connection to the Object Request Broker (ORB).
  - Step 2** Get a reference to an object, and invoke operations on this object.
  - Step 3** 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.
    - c. Obtain a ConsumerAdmin object reference.
    - d. Invoke `obtain_notification_push_consumer()` on the SupplierAdmin object, specifying `CosNotifyChannelAdmin::STRUCTURED_EVENT` as a parameter. This operation returns a reference to StructuredProxyPushSupplier.
    - e. Implement an instance of StructuredPushConsumer (defined by OMG).
    - f. Invoke the operation `connect_structured_push_consumer()` on the StructuredProxyPushSupplier object to connect the StructuredPushConsumer implementation object to the notification channel.
    - g. Monitor incoming notifications.
-

## C.2.3 Sample Code in Java

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



### 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

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

```
// 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 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](#) for more information about client profiles.

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

- 
- Step 1** Resolve reference to EMSSessionFactory.
  - Step 2** Retrieve EMSSession by supplying the username and password.
  - Step 3** Query EMSSession and get a list of managers available for operations.
  - Step 4** Invoke the desired method on one of the managers in the list.
-



### C.2.3.3.1 Get Reference to EMSSessionFactory

```

        NameComponent name = new NameComponent[6];
name[0] = new NameComponent(classString, "Class");
name[1] = new NameComponent(vendorString, "Vendor");
name[2] = new NameComponent(emsInstanceString, "EMSInstance");
name[3] = new NameComponent(versionString, "Version");
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

```

        EmsSessionFactory_I m_emsFactory = EmsSessionFactory_IHelper.narrow(obj);
NmsSession_I csession = new _tie_Session_I(new NmsSessionImpl());
orb.connect(csession);
EmsSession_IHolder sessionHolder = new EmsSession_IHolder();
m_emsFactory.getEmsSession("<userid>", "<password>", csession, sessionHolder);
EmsSession_I emsSession = sessionHolder.value;

```

### C.2.3.3.3 Retrieve List of Managers

```

        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

```

        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);

```

### C.2.3.4 Get Reference to EventChannel

```

    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

```

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

```

### C.2.3.6 Obtain ProxyPushSupplier

```

    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

```

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

```
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 D-1** *Interfaces and IDL Methods Supported in Cisco MGM*

Interface Name	IDL Methods	Supported in Cisco MGM
Common_I	setNativeEMSName	No
	setUserLabel	No
	setOwner	No
	getCapabilities	Yes
EMSMgr_I	getEMS	Yes
	getAllTopLevelSubnetworks	Yes
	getAllTopLevelSubnetworkNames	Yes
	getAllTopLevelTopologicalLinks	No
	getAllTopLevelTopologicalLinkNames	No
	getTopLevelTopologicalLink	No
	getAllEMSAndMEActiveAlarms	No
	getAllEMSSystemActiveAlarms	No
	createTopologicalLink	No
	createTopologicalLinkWithYCableProtection	No
	deleteTopologicalLink	No
	getAllL2Topologies	No

**Table D-1** Interfaces and IDL Methods Supported in Cisco MGM (continued)

Interface Name	IDL Methods	Supported in Cisco MGM
EquipmentInventoryMgr_I	provisionEquipment	No
	unprovisionEquipment	No
	setAlarmReportingOn	No
	setAlarmReportingOff	No
	getContainedEquipment	Yes
	getEquipment	Yes
	getAllEquipment	Yes
	getAllEquipmentNames	Yes
	getAllSupportedPTPs	Yes
	getAllSupportedPTPNames	Yes
	getAllSupportingEquipment	Yes
	getAllSupportingEquipmentNames	Yes
	getEthernetPTPs	No
Version_I	getVersion	Yes
EmsSessionFactory_I	getEmsSession	Yes
	getEmsPublicKey	Yes
	getEmsPublicKeyPair	Yes
Session_I	ping	Yes
	endSession	Yes
EmsSession_I	getSupportedManagers	Yes
	getManager	Yes
	getEventChannel	Yes
	releaseResource	Yes

Table D-1 Interfaces and IDL Methods Supported in Cisco MGM (continued)

Interface Name	IDL Methods	Supported in Cisco MGM
ManagedElementMgr_I	getAllManagedElements	Yes
	getAllManagedElementNames	Yes
	getContainingSubnetworkNames	No
	getAllPTPs	Yes
	getAllPTPNames	Yes
	getTP	No
	getManagedElement	Yes
	getContainedPotentialTPs	No
	getContainedPotentialTPNames	No
	getContainedInUseTPs	No
	getContainedInUseTPNames	No
	getContainedCurrentTPs	No
	getContainedCurrentTPNames	No
	getContainingTPs	No
	getContainingTPNames	No
	getAllActiveAlarms	Yes
	setTPData	No
	getAllCrossConnections	No
	getAllSNCs	No
	getAssociatedTopologicalLinks	No
	getAllSrcFTPsForSNCProvisioning	No
	getAllDestFTPsForSNCProvisioning	No
	getAllSrcPTPsForSNCProvisioning	No
	getAllSrcCTPsForSNCProvisioning	No
	getAllDestPTPsForSNCProvisioning	No
	getAllDestCTPsForSNCProvisioning	No
forceNEResync	Yes	
getAllTopologicalLinksForME	No	
NmsSession_I	eventLossOccurred	No
	eventLossCleared	No
	historyPMDDataCompleted	No
	historyPMDDataFailed	No
MultiLayerSubnetworkMgr_I		No

**Table D-1** *Interfaces and IDL Methods Supported in Cisco MGM (continued)*

Interface Name	IDL Methods	Supported in Cisco MGM
<b>Iterator Interfaces</b>		
EquipmentOrHolderIterator_I		Yes
ManagedElementIterator_I		Yes
CurrentMaintenanceOperationIterator_I		No
SubnetworkIterator_I		No
L2TopologyIterator_I		No
MLVLANIterator_I		No
EventIterator_I		No
PMDDataIterator_I		No
PerformanceManagementMgr_I		No
ProtectionGroupIterator_I		No
ProtectionMgr_I		No
EmsSessionPolicy		No
CCIterator_I		No
SNCIterator_I		No
TerminationPointIterator_I		No
TopologicalLinkIterator_I		No
TrafficDescriptorIterator_I		No
TrafficDescriptorMgr_I		No





---

## 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](#)

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](#)

---

## O

object attributes, source of [C-1](#)

objects, published CORBA naming conventions  
 for [B-1 to B-2](#)

OSS  
 client profile, creating for Cisco  
 MGM GateWay/CORBA [B-7](#)

overview of Cisco MGM GateWay/CORBA  
 creating an OSS client profile for [B-7](#)  
 developing a Cisco MGM GateWay/CORBA  
 client [B-7 to B-12](#)

---

## 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](#)

---

## U

using notification services [B-7](#)