Cisco Prime Optical 10.7 GateWay/CORBA Programmer Reference Guide

January 29, 2018

Americas 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: 408 527-0883

Cisco Prime Optical 10.7 GateWay/CORBA Programmer Reference Guide

Preface

This section explains the objectives, intended audience, and organization of this guide and describes the conventions that convey instructions and other information.

Objectives

This guide provides a detailed description of the northbound Common Object Request Broker Architecture (CORBA) interfaces supported by Cisco Prime Optical. This guide serves as a reference for developers of Operations Support System (OSS) applications that interface with Cisco Prime Optical GateWay/CORBA 10.7.

Use this guide to:

- Learn about the northbound CORBA interfaces supported by Cisco Prime Optical 10.7
- Understand the input parameters, output parameters, and attributes associated with Cisco Prime Optical GateWay/CORBA
- Audience

Audience

The primary audience for this guide includes network operations personnel and system administrators. This guide assumes the reader is familiar with the following products and topics:

- Cisco ONS 15000-series products
- Cisco Carrier Packet Transport (CPT) System
- Network Convergence System (NCS)
- Basic internetworking terminology and concepts
- Network topology and protocols
- Microsoft Windows Vista, Windows 2000 Professional, Windows XP Professional, and Windows
- Server 2003 Terminal Server
- Sun Microsystems Solaris administration
- Oracle database administration
- Object Management Group (OMG) CORBA, notification service, and naming service
- Telecommunication Management Network (TMN) architecture model

Conventions

This guide uses the following conventions:

Item	Convention
Commands and keywords	**boldface font**
Displayed session and system information screen font	*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**

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

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

! Means the following information will help you solve a problem.

Related Documentation

You can access the most current Prime Optical documentation online at http://www.cisco.com/c/en/us/support/cloud-systems-management/prime-optical/tsd-products-support-series-home.html.

The Prime Optical documentation set comprises the following guides:

- *Release Notes for Cisco Prime Optical 10.7*—Describes the caveats for Prime Optical.
- *Cisco Prime Optical 10.7 Installation Guide*—Explains how to install Prime Optical and how to upgrade from previous releases.
- *Cisco Prime Optical 10.7 Quick Start Guide*—Provides steps that lead you through a new installation for a single-server setup with an embedded database. The purpose of this guide is to quickly get you up and running with Prime Optical *10.7* .
- *Cisco Prime Optical 10.7 User Guide*—Describes how to use the Prime Optical software, which consists of user applications and tools for network discovery, network configuration, connection management, fault management, system administration, and security management.
- *Cisco Prime Optical 10.7 GateWay/CORBA Programmer Reference Guide*—This guide.
- *Cisco Prime Optical 10.7 Database Schema*—Describes the database schema that Prime Optical uses to store information in a Structured Query Language (SQL) database such as the Oracle database. The document is designed for users who need to create their own reports without using Prime Optical.

  This document is not available after Cisco Prime Optical 9.6.3.
- *Cisco Prime Optical 10.7 High Availability Installation Guide*—Explains how to install Prime Optical in a high availability (HA) environment.
- *Cisco Prime Optical 10.7 Basic External Authentication*—Describes how Prime Optical supports basic external authentication.
- *Cisco Prime Optical 10.7 Supported Devices*—Contains compatibility information and lists the Cisco software versions that can be used with a particular Cisco hardware product or module.
- *Cisco Prime Optical 10.7 Open Source*—Lists licenses and notices for open source software.
- *Migration Matrix for Service Pack Releases*—Describes the migration matrix for service pack releases

The following table lists the related NE hardware documentation.

Table 1: Related Cisco NE Documentation

Cisco Product	Document Title
Cisco Carrier Packet Transport (CPT) System	• *Cisco CPT Hardware Installation Guide* • *Cisco CPT Configuration Guide* • *Cisco CPT Command Reference Guide* • *Release Notes for Cisco CPT*

Cisco Product	Document Title
ONS 15216	• *Cisco ONS 15216 EDFA Operations Guide* • *Cisco ONS 15216 EDFA2 Operations Guide* • *Cisco ONS 15216 EDFA3 Operations Guide* • *Cisco ONS 15216 FlexLayer User Guide* • *Cisco ONS 15216 OSC-1510 User Guide* • *Cisco ONS 15216 System Dispersion Compensation Unit User Guide* • *Installing Cisco ONS 15216 100 GHz DWDM Filters* • *Installing Cisco ONS 15216 OADMs* • *Release Notes for the Cisco ONS 15216 EDFA* • *Release Notes for the Cisco ONS 15216 OADM* • *Upgrade Procedures for the Cisco ONS 15216 EDFA2*
ONS 15305 (non-CTC-based)	• *Cisco ONS 15305 Installation and Operations Guide* • *Cisco ONS 15305 Quick Installation Guide* • *Release Notes for the Cisco ONS 15305*
ONS 15305 (CTC-based)	• *Cisco ONS 15305 Cisco Transport Controller Operations Guide, R5.0* • *Release Notes for Cisco ONS 15305 Release 3.0*
ONS 15310 CL and ONS 15310 MA	• *Cisco ONS 15310 Network Defaults* • *Cisco ONS 15310-CL and Cisco ONS 15310-MA Ethernet Card Software Feature and Configuration Guide* • *Cisco ONS 15310-CL and Cisco ONS 15310-MA Procedure Guide* • *Cisco ONS 15310-CL and Cisco ONS 15310-MA Reference Manual* • *Cisco ONS 15310-CL and Cisco ONS 15310-MA Troubleshooting Guide* • *Release Notes for the Cisco ONS 15310*
ONS 15327	• *Cisco ONS 15327 Procedure Guide* • *Cisco ONS 15327 Reference Manual* • *Cisco ONS 15327 Software Upgrade Guide* • *Cisco ONS 15327 Troubleshooting Guide* • *Cisco ONS 15454 and Cisco ONS 15327 TL1 Command Guide* • *Cisco ONS 15454 and Cisco ONS 15327 TL1 Command Quick Reference Guide* • *Release Notes for the Cisco ONS 15327*

Cisco Product	Document Title
ONS 15454 SDH	• *Cisco ONS 15454 SDH Procedure Guide* • *Cisco ONS 15454 DWDM Procedure Guide* • *Cisco ONS 15454 SDH Reference Manual* • *Cisco ONS 15454 SDH Software Upgrade Guide* • *Cisco ONS 15454 SDH Troubleshooting Guide* • *Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide* • *Release Notes for the Cisco ONS 15454 SDH*
ONS 15454 SONET	• *Cisco ONS 15454 and Cisco ONS 15327 TL1 Command Guide* • *Cisco ONS 15454 and Cisco ONS 15327 TL1 Command Quick Reference Guide* • *Cisco ONS 15454 Procedure Guide* • *Cisco ONS 15454 Reference Manual* • *Cisco ONS 15454 Software Upgrade Guide* • *Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide* • *Cisco ONS 15454 Troubleshooting Guide* • *Release Notes for the Cisco ONS 15454*
ONS 15600 SDH	• *Cisco ONS 15600 SDH Procedure Guide* • *Cisco ONS 15600 SDH Reference Manual* • *Cisco ONS 15600 SDH TL1 Test Access* • *Cisco ONS 15600 SDH Troubleshooting Guide* • *Release Notes for the Cisco ONS 15600 SDH*
ONS 15600 SONET	• *Cisco ONS 15600 Procedure Guide* • *Cisco ONS 15600 Reference Manual* • *Cisco ONS 15600 TL1 Command Guide* • *Cisco ONS 15600 Troubleshooting Guide* • *Release Notes for the Cisco ONS 15600*
ONS NCS2K	• *Cisco NCS2K Procedure Guide* • *Cisco NCS2K Reference Manual* • *Cisco NCS2K Software Upgrade Guide* • *Cisco NCS2K Troubleshooting Guide* • *Release Notes for the Cisco NCS2K* • *Cisco NCS2K EDRA 1-26C Operations Guide* • *Cisco NCS2K EDRA 1-35C Operations Guide* • *Cisco NCS2K EDRA 2-26C Operations Guide* • *Cisco NCS2K EDRA 2-35C Operations Guide*

Cisco Product	Document Title
ONS NCS2K SDH	• *Cisco NCS2K SDH Procedure Guide* • *Cisco NCS2K SDH Reference Manual* • *Cisco NCS2K SDH Software Upgrade Guide* • *Cisco NCS2K SDH Troubleshooting Guide* • *Release Notes for the Cisco NCS2K SDH* • *Cisco NCS2K SDH EDRA 1-26C Operations Guide* • *Cisco NCS2K SDH EDRA 1-35C Operations Guide* • *Cisco NCS2K SDH EDRA 2-26C Operations Guide* • *Cisco NCS2K SDH EDRA 2-35C Operations Guide*

The following related documentation may also be used as reference:

- Telcordia GR-253-CORE: Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria
- 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
- Telcordia GR-2998-CORE: Generic Requirements for Wavelength Division Multiplexing (WDM) Element Management Systems
- Telcordia GR-3000-CORE: Generic Requirements for SONET Element Management Systems
- ITU-T G.774: Synchronous Digital Hierarchy (SDH)—Management information for the network element view
- ITU-T G.784: Synchronous Digital Hierarchy (SDH) management
- 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.829: Error performance events for SDH multiplex and regenerator sections
- ITU-T G.831: Management Capabilities of Transport Networks based on the synchronous digital hierarchy (SDH)
- 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 and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to What's New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

1. Overview of Cisco Prime Optical GateWay/CORBA

This chapter provides an overview of Cisco Prime Optical GateWay/CORBA (formerly Cisco Transport Manager GateWay/CORBA). It includes the following information:

Cisco Prime Optical is a carrier-class, multitechnology management system that integrates the end-to-end management of traditional transport networks and new carrier packet transport networks. It can help maintain the integrity of existing services, plus deliver interactive, content-based services and high-bandwidth applications.

Cisco Prime Optical manages the entire Cisco optical portfolio, including:

- Metro core
- Metro dense wavelength-division multiplexing (DWDM)
- Metro edge and access products
- New Carrier Packet Transport (CPT) System products

Prime Optical also serves as a foundation for integration into a larger overall Operations Support System (OSS) environment by providing northbound gateway interfaces to higher-layer management systems. The 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.

GateWay/CORBA allows service providers to implement a programmatic interface between Prime Optical and various OSS tools in the back office. Such interfaces enable next-generation flow-through equipment and circuit management, including inventory, provisioning, and assurance functions.

For each NE that Prime Optical manages, the GateWay/CORBA option provides higher-layer management systems with fault, inventory, and performance information, plus configuration and circuit provisioning capabilities. GateWay/CORBA supports up to eight simultaneous sessions with higher-layer management systems.

GateWay/CORBA supports the following NEs in Prime Optical 10.7:

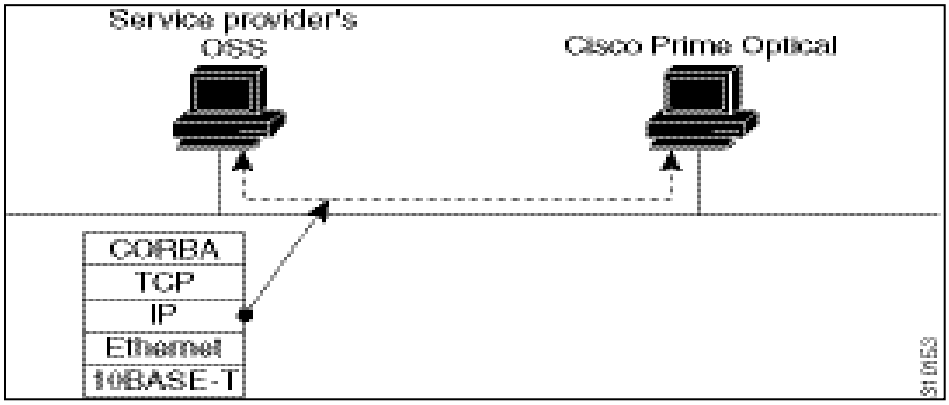
- ONS 15200
- ONS 15216
- ONS 15305
- ONS 15310 CL
- ONS 15310 MA SDH
- ONS 15310 MA SONET
- ONS 15327
- ONS 15454 SDH
- ONS 15454 SONET
- ONS 15600 SDH
- ONS 15600 SONET
- CPT 200

- CPT 200 SDH
- CPT 600
- CPT 600 SDH
- NCS2K
- NCS2k SDH

GateWay/CORBA does not support ONS 155xx family and ONS 158xx family NEs in Prime Optical 10.7.

The following figure illustrates the GateWay/CORBA communications architecture within a service provider's OSS environment.

Figure 1-1: 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. GateWay/CORBA is based on the following TMF standards:

- TMF.513 v2.0 (August 2001): Multi Technology Network Management Business Agreement
- TMF.608 v2.0 (August 2001): Multi Technology Network Management Information Agreement
- TMF.814 v3.0 (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 might use them.

1.1 Hardware Modifications

The following new hardwares have been supported in Release 10.7:

- 20-SMRFS-CV (Single Module ROADM (SMR)).
- NCS2K-MF-6RU (10-Slot Passive Shelf)
- NCS2K-MF10-6RU (14-Slot Passive Shelf)

1.2 Features of GateWay/CORBA

GateWay/CORBA supports the following TMF814 modules:

- Inventory
  - Element management system (EMS)
  - Multilayer subnetwork
  - Managed element
  - Physical termination point (PTP)
  - Connection termination point (CTP)
  - Floating termination point (FTP)
  - Subnetwork connection (SNC)
  - Protection group
  - Equipment
  - Equipment holder
  - Topological link
- Alarm notification
- Retrieval of active alarms on objects where inventory is available
- SNC provisioning
- Equipment provisioning
- Protection operations
- Performance operations
- Maintenance operations

✐ Performance and inventory application programming interfaces (APIs) do not retrieve any information for out-of-service NEs.

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

GateWay/CORBA uses jacORB 2.1.3.7. The corresponding standard is Object Management Group (OMG) CORBA 2.3.

1.3 GateWay/CORBA Fault Management

GateWay/CORBA forwards all the fault notifications received from managed NEs and all Prime Optical alarms to the network management system (NMS) through the notification service. Information about the notification service is available on the OMG website at www.omg.org.

✐ This website is copyright © 1997–2009, Object Management Group, Inc.

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.

1.4 Overview of CORBA Services

GateWay/CORBA uses two CORBA services—naming service and notification service. The naming service is required for GateWay/CORBA to function. If the 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 Prime Optical server during the GateWay/CORBA installation. However, the CORBA services can be located anywhere as long as they are accessible from the Prime Optical server on which GateWay/CORBA is installed. The OSS application(s) must also use the same instances of the naming service and the notification service.

1.4.1 Naming Service

GateWay/CORBA and the other Prime Optical server processes must use a naming service in order to operate. GateWay/CORBA includes the OpenFusion naming service product. By default,

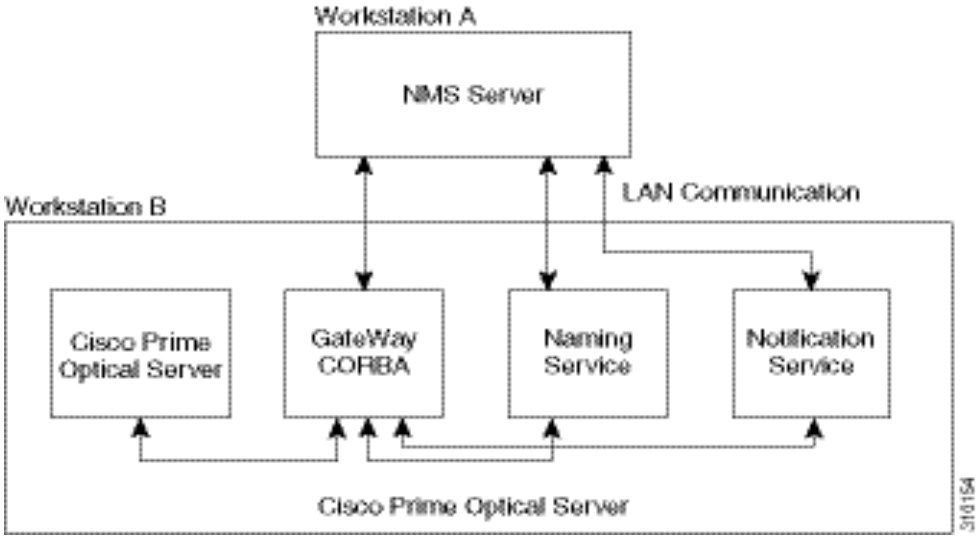
GateWay/CORBA is configured to use OpenFusion. However, you can configure GateWay/CORBA and the Prime Optical server to use a different centralized naming service. For more information, see 6.6 Using Multiple Naming Servers.

1.4.2 Notification Service

GateWay/CORBA must use a CORBA notification service to report faults and events. 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 GateWay/CORBA to use a different notification service. For more information, see 5.1 Overview of Notifications.
The following scenarios describe possible configurations of the naming and notification services.
Scenario 1: Naming and Notification Services Installed on the Prime Optical Server

The naming and notification services are installed on the Prime Optical server workstation as shown in the following figure.

Figure 1-2: Naming and Notification Services on the Prime Optical Server Workstation

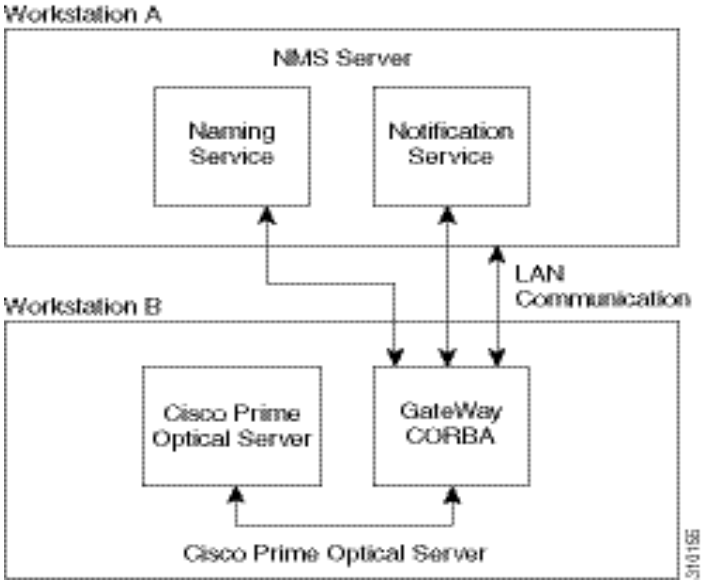


The notification service must be installed on the Prime Optical server workstation with the Prime Optical server processes. GateWay/CORBA is configured to use this notification service. The NMS processes must be configured to use the notification and naming services on the Prime Optical server workstation.

Scenario 2: Naming and Notification Services Installed on the NMS Workstation

The naming and notification services are installed on the NMS workstation as shown in the following figure.

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

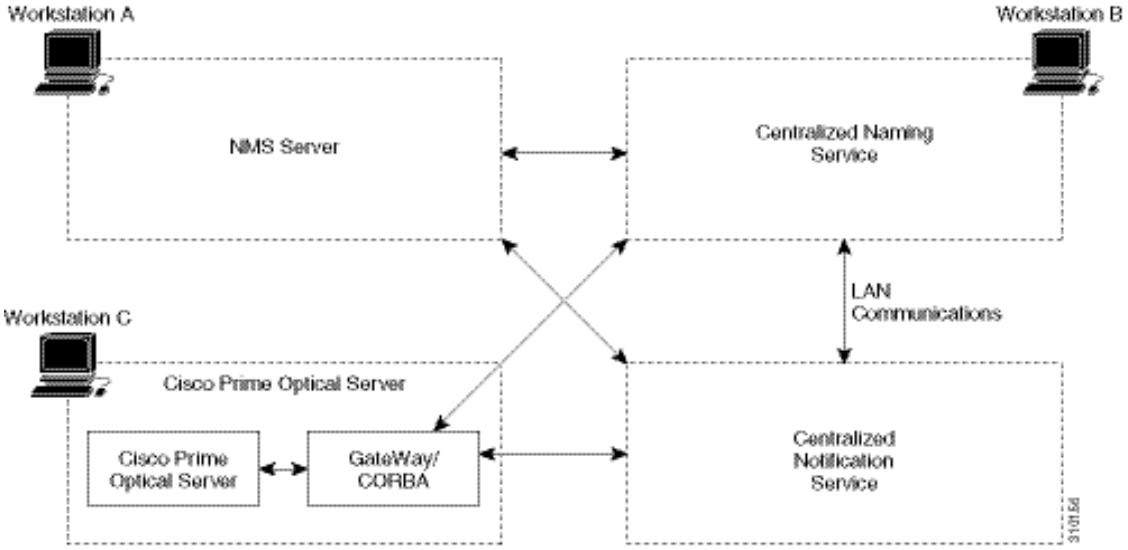


The Prime Optical server processes and GateWay/CORBA must be configured to use the naming and notification services on the NMS workstation.

Scenario 3: Naming and Notification Services Installed on Independent Servers

The naming and notification services are installed on independent servers as shown in the following figure.

Figure 1-4: Naming and Notification Services on Independent Servers



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

1.5 Information Model

The information model used by the GateWay/CORBA northbound interface is based on the TMF NML-EML CORBA interface—TMF.814 v3.0. 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 Prime Optical must use the GateWay/CORBA product, and will require specific information on what API features are implemented and how those features should be used.

1.6 IDL-Related Changes

The following table lists the IDL-related changes that occurred between two given releases.

Table 1-1: Functionalities Not Supported

Release Details	Change

Cisco Prime Optical 9.3.1 to Cisco Prime Optical 9.5	New enumerated PGT_Y_CABLE added to protection.idl
	enum ProtectionGroupType_T
	{
	PGT_MSP_1_PLUS_1,
	PGT_MSP_1_FOR_1,
	PGT_MSP_1_FOR_N,
	PGT_2_FIBER_BLSR,
	PGT_4_FIBER_BLSR,
	PGT_SPLITTER,
	PGT_L2_1_PLUS_1,
	PGT_Y_CABLE
	};

Cisco Prime Optical 9.3 to Cisco Prime Optical 9.3.1	No change
From R9.2 to Cisco Prime Optical 9.3	globaldefs.idl (added definition): typedef sequence<NamingAttributesList_T> NamingAttributesMultipleList_T; multiLayerSubnetwork.idl (added API): void getRouteAndTopologicalLinks(in globaldefs::NamingAttributes_T sncName, out subnetworkConnection::Route_T route, out topologicalLink::TopologicalLinkList_T topologicalLinkList) raises(globaldefs::ProcessingFailureException); The following new standard IDLs have been added: flowDomain.idl flowDomainFragment.idl trafficConditioningProfile.idl trailNtwProtection.idl
From R9.1 to R9.2	No change

Release Details	Change
From R9.0 to R9.1	managedElementManager.idl extended with new API void getAllCtmMEInternalPatchord(in globaldefs::NamingAttributes_T managedElementName, out globaldefs::NVSList_T patches) raises (globaldefs::ProcessingFailureException); performance.idl extended with new API void getCTMValidHistoryPMData(in nmsSession::NmsSession_I client, in PMTPSelectList_T pmTPSelectList, in PMParameterNameList_T pmParameters, in globaldefs::Time_T startTime, in globaldefs::Time_T endTime) raises(globaldefs::ProcessingFailureException);

From R8.5 to R9.0	New enumerated PC_LOCKON, PGT_SPLITTER, PGT_L2_1_PLUS_1 added to protection.idl
	New enumerated OT_NETWORK_PARTITION added to notifications.idl.
	New interface type added to multiLayerSubnetwork.idl:
	enum L2InterfaceType_T
	{
	L2IT_UNKNOWN,
	L2IT_FASTETHER,
	L2IT_GIGAETHER,
	L2IT_PORTCHANNEL
	};
	L2InterfaceType_T interfaceType;

From R10.5.2 to R10.7	New enumeration SNCS_ACTIVE_PREROUTED added to subnetworkconnection.idl
	
	enum SNCState_T
	{
	SNCS_NONEXISTENT,
	SNCS_PENDING,
	SNCS_ACTIVE,
	SNCS_PARTIAL,
	SNCS_DELETING,
	SNCS_ACTIVE_PREROUTED
	};

1.7 Functionalities Not Supported in Prime Optical 10.7

The following table lists the functionalities which are not supported in release 10.7. The cards or equipment in which the functionalities are not supported are also listed.

✎ The inventory for ECU card on M15 chassis will not list the USB ports.

Functionalities	Card/Equipment
Provisioning Optical Transmission Section (OTS) links	Passive units
Historical PM retrieval	80WXC-C MSNE 30 Shelf XP_10GE (gz) for LACP, REP 100GS-CK-C 200G-CK-C MR-MXP
Realtime PM retrieval	80WXC-C MSNE 30 Shelf ADM_10G for RMON, IPPMHDLC/LEX XP_10GE (gz) for E-OAM, LACP, REP TNC/TNCE 40G_MXP_C 40E_MXP_C EDRA1-26C EDRA1-35C EDRA2-26C EDRA2-35C 16-WXC-FS 100G-CK-C 100GS-CK-C 200G-CK-C MR-MXP
Get and set thresholds	80WXC-C
PPC links	80WXC-C
Alarm management	XP_10GE (gz) for CFM, REP, E-OAM, LACP
Card mode provisioning	AR-XP AR-MXP 100G-LC-C 10x10G-LC WSE M-CFP-LC 100G-CK-C AR-XPE MR-MXP
Thresholds and TCA settings	_ **Note**: Secuirty threshold settings for WSE card is not supported

1.8 Summary

As a state-of-the-art EMS, Prime Optical can provide a CORBA-based interface between itself and an NMS. This functionality is provided by a separately purchasable option— GateWay/CORBA.

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. GateWay/CORBA includes naming and notification services. Customers can use the bundled naming and notification services or their own services in various configurations. GateWay/CORBA brings the service provider one step closer to full OSS automation.

2 NE- and Cisco Prime Optical-Specific Details

This chapter provides specific details about each of the NEs that Cisco Prime Optical supports. It also provides details specific to Prime Optical.

The following sections include NE- and Prime Optical-specific details:

- 2.1 NE-Specific Details
- 2.2 Prime Optical-Specific Details

✐ In this document, the terms network element (NE) and managed element (ME) are interchangeable. Also, card and equipment strings are used interchangeably to refer to equipment.

2.1 NE-Specific Details

2.1.1 ONS 15216

Some NEs in the ONS 15216 family do not provide a management interface. These NEs are considered passive because the user enters all data from the Prime Optical client. After the data has been entered, Prime Optical reports these NEs. Other ONS 15216 NEs provide a management interface; these NEs are considered act

See the Release Notes for Cisco Prime Optical 10.7 for the passive and active ONS 15216 software releases that Prime Optical 10.7 supports.

✐ All ONS 15216 NEs contain a single shelf and have no removable equipment, except for the ONS 15216 DCU, which contains a chassis and a removable module. The ONS 15216 FlexLayer also has a removable module.

2.1.1.1 PTPs

Prime Optical reports PTPs for ONS 15216 NEs only if a topological link originates from or terminates on the ONS 15216 NE.

2.1.1.2 Topological Links

Topological links are unidirectional for ONS 15216 NEs. Inter-NE links are available between all NEs in the ONS 15216 family. Prime Optical supports links between an ONS 15216 and the following NEs at compatible layer rates:

- ONS 15216
- ONS 15310 CL and ONS 15310 MA SONET
- ONS 15327 (physical, SONET/SDH, and OCH)
- ONS 15454 (physical, SONET/SDH, and OCH)
- ONS 15600 (physical, SONET/SDH, and OCH)
- NCS2K (physical, SONET/SDH, and OCH)

2.1.2 ONS 15305 and ONS 15305 CTC

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15305 software releases that Prime Optical 10.7 supports.

✐ NEs with R3.x are called ONS 15305 CTC.

2.1.2.1 Slots

The ONS 15305 and ONS 15305 CTC are single-shelf NEs with four slots.

2.1.2.2 PTPs

The ONS 15305 and ONS 15305 CTC support PTPs associated to WAN, SDH, and PDH ports.

2.1.3 ONS 15310 CL

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15310 CL software releases that Prime Optical 10.7 supports.

2.1.3.1 Slots

The ONS 15310 CL is a single-shelf NE with two slots. Slot 1 is the expansion slot and contains the provisionable data card. Slot 2 is a nonreplicable unit and contains the Control Timing and Cross-Connect (CTX)-CL controller.

2.1.3.2 Equipment

All cards for the ONS 15310 CL R5.0 and later display administration and service states. For earlier software releases, these states are not applicable and Prime Optical displays them as N/A. The following tables show the attribute values for these two states.

2.1.3.2.1 Equipment Administration State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. Table 2-1 lists the attribute values for the administration state.
Table 2-2 and Table 2-3 list the attribute values for the service state.

Table 2-1: Equipment Administration State (ONS 15310 CL)

Attribute Value	Meaning
In Service	Places the entity In Service.
Auto in Service	Places the entity in Auto in Service. The circuit is out of service until it receives a valid signal for the duration of the soak period, at which time the circuit state changes to In Service. During the soak period, alarms and loopbacks are suppressed but traffic is carried.
Out of Service–Maintenance	Removes the entity from service for maintenance. Alarms are no longer generated.
Out of Service	Removes the entity from service and disables it. Alarms are no longer generated and traffic is not carried.

2.1.3.2.2 Equipment Service State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. The attribute value for the service state is the same as that shown in Prime Optical.

Table 2-2: Equipment Service State–Primary (ONS 15310 CL)

Primary States (PST-PSTQ)	
Attribute Value	**Meaning**
In Service–Normal (IS_NR)	The entity is fully operational and performs as provisioned.
Out of Service–Management (OOS_MA)	The entity has been administratively removed from service.
Out of Service–Autonomous (OOS_AU)	The entity is not operational due to an autonomous event.
Out of Service–Autonomous and Management (OOS_AUMA)	The entity is not operational due to an autonomous event and has been administratively removed from service.

Table 2-3: Equipment Service State–Secondary (ONS 15310 CL)

Secondary States (SST)	
Attribute Value	**Meaning**
Unassigned (UAS)	The entity is not provisioned in the database.
Unequipped (UEQ)	The entity has been removed, leaving an empty slot. It may or may not be assigned (provisioned).
Mismatch of Equipment and Attributes (MEA)	Incorrect equipment is installed. For example, the card plugged in is not compatible with the provisioned card or slot.
Automatic In-Service (AINS)	The entity is in a delay transition (to IS) state. The transition to IS is pending the correction of off-normal conditions on the entity.
Maintenance (MT)	The entity has been manually removed from service for maintenance. It is still capable of performing its provisioned functions. Traffic is still carried.
Disabled (DSBLD)	The entity has been manually removed from service and cannot perform its provisioned functions. All of its provisioned services are disrupted. Traffic is not carried.
Loopback (LPBK)	The loopback command is in effect (any loopback type).

2.1.3.3 Topological Links

All topological links are unidirectional or bidirectional for ONS 15310 CL NEs.

2.1.3.4 CTX Cards

The CTX card is an electrical card that consists of 21 digital signal level 1 (DS1) ports and three DS3/EC1 ports. Two optical interfaces are supported using SFP technology. OC-3/STM-1 and OC-12/STM-4 ports are available.

2.1.3.5 PTPs

All PTPs are bidirectional for the ONS 15310 CL. The ports are always in channelized mode. PTPs support In Service and Out of Service–Maintenance values for the service state attribute.

2.1.3.6 CTPs

All CTPs are in channelized mode for the ONS 15310 CL.

2.1.3.7 SNCs

CTX cards perform synchronous transport signal (STS) and VT1.5 switching. The subnetwork connection (SNC) name cannot exceed 48 characters. The NE enforces the SNC name character limit.

2.1.3.8 Autodiscovery

The ONS 15310 CL supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an attribute

value change (AVC) event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.4 ONS 15310 MA SDH

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15310 MA SDH software releases that Prime Optical 10.7 supports.

2.1.4.1 Slots

The ONS 15310 MA SDH consists of two CTX slots and four Expansion (EXP) I/O slots.

2.1.4.2 Topological Links

All topological links are unidirectional or bidirectional for ONS 15310 MASDH NEs.

2.1.4.3 Cards

The ONS 15310 MA SDH supports the following cards:

- CTX2500
  - E1_21_E3_DS3_3_LINE_CARD—The electrical ports consist of 21 DS1 and 3 DS3 or EC1 ports.
  - E1_63_E3_DS3_3_LINE_CARD—The electrical ports consist of 63 DS1 and 3 DS3 or EC1 ports.
- ML-100T-8
- CE-100T-8

2.1.4.4 PTPs

All PTPs are bidirectional for the ONS 15310 MA SDH. The ports are always in channelized mode. PTPs support In Service and Out of Service–Maintenance values for the service state attribute.
SNC provisioning on E1_21_DS3_3 and E1_63_E3_3 cards uses the following DS3/E3 ports:

- E1_21_DS3_3—Port numbers 22, 23, and 24.
- E1_63_E3_3—Port numbers 64, 65, and 66.

2.1.4.5 Auto Discovery

The ONS 15310 MA SDH supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an AVC event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.5 ONS 15310 MA SONET

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15310 MA SONET software releases that Prime Optical 10.7 supports.

2.1.5.1 Slots

The ONS 15310 MA SONET consists of two CTX slots and four EXP I/O slots.

2.1.5.2 Topological Links

All topological links are unidirectional or bidirectional for ONS 15310 MA SONET NEs.

2.1.5.3 CTX Cards

The CTX card is an electrical card that consists of 21 DS1 ports and three DS3/EC-1 ports. Two optical interfaces are supported using SFP technology. OC-3/STM-1 and OC-12/STM-4 ports are available.

- DS1_28_DS3_EC1_3_LINE_CARD—The electrical ports consist of 28 DS1 and 3 DS3 or EC1 ports.
- DS1_84_DS3_EC1_3_LINE_CARD—The electrical ports consist of 84 DS1 and 3 DS3 or EC1 ports.

2.1.5.4 PTPs

All PTPs are bidirectional for the ONS 15310 MA SONET. The ports are always in channelized mode. PTPs support In Service and Out of Service–Maintenance values for the service state attribute.
SNC provisioning on DS1_28_DS3_3 and DS1_84_DS3_3 cards uses the following DS3/EC1 ports:

- DS1_28_DS3_3—Port numbers 29, 30, and 31.
- DS1_84_DS3_3—Port numbers 85, 86, and 87.

2.1.5.5 CTPs

All CTPs are in channelized mode for the ONS 15310 MA SONET.

2.1.5.6 SNCs

CTX cards perform STS and VT1.5 switching. The SNC name cannot exceed 48 characters. The NE enforces the SNC name character limit.

2.1.5.7 Auto Discovery

The ONS 15310 MA SONET supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an AVC event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.6 ONS 15327

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15327 software releases that Prime Optical 10.7 supports.

2.1.6.1 Slots

The ONS 15327 is a single-shelf NE with eight slots. The Mechanical Interface Card (MIC) A card must occupy slot 8; the MIC B card must occupy slot 7. If an integrated Cross-Connect, Timing, and Control (XTC) card is required, it must occupy slot 5 or slot 6.

2.1.6.2 Equipment

All of the cards in the ONS 15327 R5.0 and later display administration and service states. For earlier software releases, these states are not applicable and Prime Optical displays them as N/A.

2.1.6.2.1 Equipment Administration State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. Table 2-4 lists the attribute values for the administration state. Table 2-5 and Table 2-6 list the attribute values for the service state.

Table 2-4: Equipment Administration State (ONS 15327)

Attribute Value	Meaning
In Service	Places the entity In Service.
Auto in Service	Places the entity in Auto in Service. The circuit is out of service until it receives a valid signal for the duration of the soak period, at which time the circuit state changes to In Service. During the soak period, alarms and loopbacks are suppressed but traffic is carried.
Out of Service–Maintenance	Removes the entity from service for maintenance. Alarms are no longer generated.
Out of Service	Removes the entity from service and disables it. Alarms are no longer generated and traffic is not carried.

2.1.6.2.2 Equipment Service State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. The attribute value for the service state is the same as that shown in Prime Optical.

Table 2-5: Equipment Service State–Primary (ONS 15327)

Primary States (PST-PSTQ)	
Attribute Value	**Meaning**
In Service–Normal (IS_NR)	The entity is fully operational and performs as provisioned.
Out of Service–Management (OOS_MA)	The entity has been administratively removed from service.
Out of Service–Autonomous (OOS_AU)	The entity is not operational due to an autonomous event.
Out of Service–Autonomous and Management (OOS_AUMA)	The entity is not operational due to an autonomous event and has been administratively removed from service.

Table 2-6: Equipment Service State–Secondary (ONS 15327)

Secondary States (SST)	
Attribute Value	**Meaning**
Unassigned (UAS)	The entity is not provisioned in the database.
Unequipped (UEQ)	The entity has been removed, leaving an empty slot. It may or may not be assigned (provisioned).
Mismatch of Equipment and Attributes (MEA)	Incorrect equipment is installed. For example, the card plugged in is not compatible with the provisioned card or slot.
Automatic In-Service (AINS)	The entity is in a delay transition (to IS) state. The transition to IS is pending the correction of off-normal conditions on the entity.

Secondary States (SST)	
Attribute Value	**Meaning**
Maintenance (MT)	The entity has been manually removed from service for maintenance. It is still capable of performing its provisioned functions. Traffic is still carried.
Disabled (DSBLD)	The entity has been manually removed from service and cannot perform its provisioned functions. All of its provisioned services are disrupted. Traffic is not carried.
Loopback (LPBK)	The loopback command is in effect (any loopback type).

2.1.6.3 XTC Cards

The XTC-14 card has 14 DS1 ports numbered 1 through 14. Only VT1.5 SNCs can be created on these ports. The XTC-28-3 card has 28 DS1 ports numbered 1 through 28 and 3 DS3 ports numbered 29 through 31. VT1.5 SNCs can be created on DS1 ports, and STS-1 SNCs can be created on DS3 ports.

2.1.6.4 PTPs

All PTPs are bidirectional for the ONS 15327. The ports are always in channelized mode. PTPs support In Service, Out of Service, Out of Service–Maintenance, and Auto in Service values for the service state attribute.

2.1.6.5 CTPs

All CTPs are in channelized mode for the ONS 15327. Alarm monitoring cannot be turned on or off for CTPs.

2.1.6.6 SNCs

XTC cards perform STS and VT1.5 switching. The XTC cards support the total rearrangement of 192 bidirectional STSs from the four high-speed slots (1 to 4), plus 12 bidirectional STSs for XTC module low-speed electrical interfaces. The XTC VT1.5 matrix supports the grooming of 336 bidirectional VT1.5 circuits.
The SNC name cannot exceed 48 characters.

2.1.6.7 Autodiscovery

ONS 15327 autodiscovery is identical to ONS 15454 SONET autodiscovery, described in 2.1.8.10 Autodiscovery.

2.1.7 ONS 15454 SDH

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15454 SDH software releases that Prime Optical 10.7 supports.

2.1.7.1 Shelves

All supported ONS 15454 SDH NEs have exactly one shelf each, with the exception of the ONS 15454 SDH multiservice transport platform (MSTP) R7.0, which can have up to eight shelves per NE.

2.1.7.2 Slots

The ONS 15454 SDH has one shelf view supplemented with the top subrack Electrical Facility Connection Assembly (EFCA) shelf and bottom subrack chassis. The main shelf has 17 slots numbered 1 to 17 and contains cards that carry traffic. The EFCA shelf has 12 additional slots numbered 18 to 29 and contains Front Mount Electrical Connection (FMEC) modules, the ALM/PWR/MIC (A/P/M) module, and the CRFT/TMG/MIC (C/T/M) module. The Timing Communications and Control-International (TCC-I)

module must be in slot 7 or slot 11, the A/P/M module must be in slot 23, and the C/T/M module must be in slot 24.

2.1.7.3 Equipment

All cards in the ONS 15454 SDH R5.0 and later display administration and service states. For earlier software releases, these states are not applicable and Prime Optical displays them as N/A.

2.1.7.3.1 Equipment Administration State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. Table 2-7 lists the attribute values for the administration state. Table 2-8 and Table 2-9 list the attribute values for the service state.

Table 2-7: Equipment Administration State (ONS 15454 SDH)

Attribute Value	Meaning
In Service	Places the entity In Service.
Auto in Service	Places the entity in Auto in Service. The circuit is out of service until it receives a valid signal for the duration of the soak period, at which time the circuit state changes to In Service. During the soak period, alarms and loopbacks are suppressed but traffic is carried.
Out of Service–Maintenance	Removes the entity from service for maintenance. Alarms are no longer generated.
Out of Service	Removes the entity from service and disables it. Alarms are no longer generated and traffic is not carried.

2.1.7.3.2 Equipment Service State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. The attribute value for the service state is the same as that shown in Prime Optical.

Table 2-8: Equipment Service State–Primary (ONS 15454 SDH)

Primary States (PST-PSTQ)	
Attribute Value	**Meaning**
In Service–Normal (IS_NR)	The entity is fully operational and performs as provisioned.
Out of Service–Management (OOS_MA)	The entity has been administratively removed from service.
Out of Service–Autonomous (OOS_AU)	The entity is not operational due to an autonomous event.
Out of Service–Autonomous and Management (OOS_AUMA)	The entity is not operational due to an autonomous event and has been administratively removed from service.

Table 2-9: Equipment Service State–Secondary (ONS 15454 SDH)

Secondary States (SST)	
Attribute Value	**Meaning**
Unassigned (UAS)	The entity is not provisioned in the database.
Unequipped (UEQ)	The entity has been removed, leaving an empty slot. It may or may not be assigned (provisioned).
Mismatch of Equipment and Attributes (MEA)	Incorrect equipment is installed. For example, the card plugged in is not compatible with the provisioned card or slot.

Secondary States (SST)	
Attribute Value	**Meaning**
Automatic In-Service (AINS)	The entity is in a delay transition (to IS) state. The transition to IS is pending the correction of off-normal conditions on the entity.
MT (Maintenance)	The entity has been manually removed from service for maintenance. It is still capable of performing its provisioned functions. Traffic is still carried.
DSBLD (Disabled)	The entity has been manually removed from service and cannot perform its provisioned functions. All of its provisioned services are disrupted. Traffic is not carried.
LPBK (Loopback)	The loopback command is in effect (any loopback type).

2.1.7.4 Topological Links

Topological links are unidirectional or bidirectional for ONS 15454 SDH NEs.

2.1.7.5 PTPs

All PTPs are bidirectional for the ONS 15454 SDH, except for PTPs on DWDM cards (optical amplifiers, optical multiplexers, optical demultiplexers, and optical add/drop multiplexers). The ports are always in channelized mode. PTPs support IN_SERVICE, OUT_OF_SERVICE, OUT_OF_SERVICE_MAINTENANCE, and AUTO_IN_SERVICE values for the service state attribute.

2.1.7.6 CTPs

All CTPs are in channelized mode for the ONS 15454 SDH. Alarm monitoring cannot be turned on or off for CTPs.

2.1.7.7 SNCs

One node supports a maximum of 192 bidirectional VC4 SNCs. The SNC name cannot exceed 48 characters. The NE enforces the SNC name character limit.

2.1.7.8 Equipment Protection

1:N equipment protection is supported for E1, E1-42, and DS3i cards. You must install protect cards (E1-n, E1-42, DS3i-n) in slot 3 or slot 15 on the same side of the shelf as the protected cards. Protect cards must match the cards they protect. For example, an E1-n card protects only E1 or E1-n cards. You must create the protection group. Installing E1-n, E1-42, or DS3i-n cards in slot 3 or slot 15 does not automatically protect other E1, E1-42, or DS3i cards. 1:N equipment protection is always revertive.
1:1 equipment protection is supported for E1, E1-42, E3, STM-1E-12, and DS3i cards. Install the protect card in an odd-numbered slot and install the working card in an adjacent even-numbered slot. For example, install the protect card in slot 1 and install the working card in slot 2.

2.1.7.9 Facility Protection

Facility protection is available for STM-n PTPs. Protect ports must match the working ports. For example, port 1 of an STM-1 (OC-3) card is protected only by port 1 on another STM-1 (OC-3) card.

2.1.7.10 Autodiscovery

ONS 15454 SDH autodiscovery is identical to ONS 15454 SONET autodiscovery, described in 2.1.8.10 Autodiscovery.

2.1.8 ONS 15454 SONET

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15454 SONET software releases that Prime Optical 10.7 supports.

2.1.8.1 Shelves

All supported ONS 15454 SONET NEs have exactly one shelf each, with the exception of the ONS 15454 SONET MSTP R7.0, which can have up to eight shelves per NE.

2.1.8.2 Slots

Each ONS 15454 SONET shelf has 17 slots. The Timing Communications and Control Plus (TCC+) card must occupy slot 7 or slot 11. For more information, refer to the Cisco ONS 15454 SONET user documentation.

2.1.8.3 Equipment

All the cards in the ONS 15454 SONET R5.0 and later display administration and service states. For earlier software releases, these states are not applicable and Prime Optical displays them as N/A. The following tables show the attribute values for these two states.

2.1.8.3.1 Equipment Administration State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. Table 2-10 lists the attribute values for the administration state. Table 2-11 and Table 2-12 list the attribute values for the service state.

Table 2-10: Equipment Administration State (ONS 15454 SONET)

Attribute Value	Meaning
In Service	Places the entity In Service.
Auto in Service	Places the entity in Auto in Service. The circuit is out of service until it receives a valid signal for the duration of the soak period, at which time the circuit state changes to In Service. During the soak period, alarms and loopbacks are suppressed but traffic is carried.
Out of Service–Maintenance	Removes the entity from service for maintenance. Alarms are no longer generated.
Out of Service	Removes the entity from service and disables it. Alarms are no longer generated and traffic is not carried.

2.1.8.3.2 Equipment Service State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. The attribute value for the service state is the same as that shown in Prime Optical.

Table 2-11: Equipment Service State–Primary (ONS 15454 SONET)

Primary States (PST-PSTQ)	
Attribute Value	**Meaning**
In Service–Normal (IS_NR)	The entity is fully operational and performs as provisioned.
Out of Service–Management (OOS_MA)	The entity has been administratively removed from service.
Out of Service–Autonomous (OOS_AU)	The entity is not operational due to an autonomous event.
Out of Service–Autonomous and Management (OOS_AUMA)	The entity is not operational due to an autonomous event and has been administratively removed from service.

Table 2-12: Equipment Service State–Secondary (ONS 15454 SONET)

Secondary States (SST)	
Attribute Value	**Meaning**
UAS (Unassigned)	The entity is not provisioned in the database.
UEQ (Unequipped)	The entity has been removed, leaving an empty slot. It may or may not be assigned (provisioned).
MEA (Mismatch of Equipment and Attributes)	Incorrect equipment is installed. For example, the card plugged in is not compatible with the provisioned card or is not compatible with the slot.
AINS (Automatic In-Service)	The entity is in a delay transition (to IS) state. The transition to IS is pending the correction of off-normal conditions on the entity.
MT (Maintenance)	The entity has been manually removed from service for maintenance. It is still capable of performing its provisioned functions. Traffic is still carried.
DSBLD (Disabled)	The entity has been manually removed from service and cannot perform its provisioned functions. All of its provisioned services are disrupted. Traffic is not carried.
LPBK (Loopback)	The loopback command is in effect (any loopback type).

2.1.8.4 Topological Links

Topological links are unidirectional or bidirectional for ONS 15454 SONET NEs.

2.1.8.5 PTPs

All PTPs are bidirectional for the ONS 15454 SONET, except for PTPs on DWDM cards (optical amplifiers, optical multiplexers, optical demultiplexers, and optical add/drop multiplexers). The ports are always in channelized mode. PTPs support IN_SERVICE, OUT_OF_SERVICE, OUT_OF_SERVICE_MAINTENANCE, and AUTO_IN_SERVICE values for the service state attribute.

2.1.8.6 CTPs

All CTPs are in channelized mode for the ONS 15454 SONET. Alarm monitoring cannot be turned on or off for CTPs.

2.1.8.7 SNCs

One node supports a maximum of 144 bidirectional STS SNCs with an XC or Cross-Connect Virtual Tributary (XCVT) card installed or 576 bidirectional STS SNCs with a 10 Gigabit Cross-Connect (XC10G) card installed. If an XCVT or XC10G card is present, you can create a maximum of 336 bidirectional VT1.5 SNCs.
The SNC name cannot exceed 48 characters. The NE enforces the SNC name character limit.

2.1.8.8 Equipment Protection

1:N equipment protection is supported for DS1, DS3, DS3I, and DS3E cards. You must install protect cards (DS1N, DS3N, DS3IN, and DS3NE) in slot 3 or slot 15 on the same side of the shelf as the protected cards. Protect cards must match the cards they protect. For example, a DS1N protects only DS1 or DS1N cards. You must create the protection group. Installing DS1N or DS3N cards in slot 3 or slot 15 does not automatically protect other DS1 or DS3 cards. 1:N equipment protection is always revertive.

1:1 equipment protection is supported for DS1, DS3, DS3I, DS3E, EC-1, and DS3XM cards. Install the protect card in an odd-numbered slot and install the working card in an adjacent even-numbered slot. For example, install the protect card in slot 1 and install the working card in slot 2.

2.1.8.9 Facility Protection

Facility protection is available for OC-n PTPs. Protect ports must match the working ports. For example, port 1 on an OC-3 card is protected only by port 1 on another OC-3 card.

2.1.8.10 Autodiscovery

The ONS 15454 SONET supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an AVC event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.9 ONS NCS2K SDH

See the Cisco Prime Optical 10.7 Release Notes for the ONS NCS2K SDH software releases that Prime Optical 10.7 supports. For more details, see section 2.1.7 ONS 15454 SDH.

2.1.10 ONS NCS2K SONET

See the Cisco Prime Optical 10.7 Release Notes for the ONS NCS2K SONET software releases that Prime Optical 10.7 supports. For more details, see section 2.1.8 ONS 15454 SDH.

2.1.11 ONS 15600 SONET

See the Cisco Prime Optical 10.7 Release Notes for the ONS 15600 SONET software releases that Prime Optical 10.7 supports.

2.1.11.1 Slots

The ONS 15600 SONET is a single-shelf NE with 14 slots. The Timing and Shelf Control (TSC) card must occupy slot 5 or slot 10. The Core Cross-Connect (CXC) cards are always 1+1 redundant in slots 6/7 and 8/9. Each CXC card occupies two slots. Slots 1 to 4 and 11 to 14 are reserved for optical I/O cards. Single Shelf Cross-Connect (SSXC) cards replace CXC cards from release 5.0 of the NE. However, due to a known caveat (CSCse87505), Prime Optical and CTC display SSXC cards as CXC. For more information, refer to the Cisco ONS 15600 SONET user documentation.

2.1.11.2 Equipment

All cards in the ONS 15600 SONET R5.0 and later display administration and service states. For earlier software releases, these states are not applicable and Prime Optical displays them as N/A.

2.1.11.3 Equipment Administration State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. Table 2-13 lists the attribute values for the administration state. Table 2-14 and Table 2-15 list the attribute values for the service state.

Table 2-13: Equipment Administration State (ONS 15600 SONET)

Attribute Value	Meaning
In Service	Places the entity In Service.
Auto in Service	Places the entity in Auto in Service. The circuit is out of service until it receives a valid signal for the duration of the soak period, at which time the circuit state changes to In Service. During the soak period, alarms and loopbacks are suppressed but traffic is carried.
Out of Service–Maintenance	Removes the entity from service for maintenance. Alarms are no longer generated.
Out of Service	Removes the entity from service and disables it. Alarms are no longer generated and traffic is not carried.

2.1.11.4 Equipment Service State

The attribute name is ACTUAL_EQUIPMENT_SERVICE_STATE and is displayed in getAdditionalInfo for the equipment. The attribute value for the service state is the same as that shown in Prime Optical.

Table 2-14: Equipment Service State–Primary (ONS 15600 SONET)

Primary States (PST-PSTQ)	
Attribute Value	**Meaning**
In Service–Normal (IS_NR)	The entity is fully operational and performs as provisioned.
Out of Service–Management (OOS_MA)	The entity has been administratively removed from service.
Out of Service–Autonomous (OOS_AU)	The entity is not operational due to an autonomous event.
Out of Service–Autonomous and Management (OOS_AUMA)	The entity is not operational due to an autonomous event and has been administratively removed from service.

Table 2-15: Equipment Service State–Secondary (ONS 15600 SONET)

Secondary States (SST)	
Attribute Value	**Meaning**
Unassigned (UAS)	The entity is not provisioned in the database.
Unequipped (UEQ)	The entity has been removed, leaving an empty slot. It may or may not be assigned (provisioned).
Mismatch of Equipment and Attributes (MEA)	Incorrect equipment is installed. For example, the card plugged in is not compatible with the provisioned card or slot.
Automatic In-Service (AINS)	The entity is in a delay transition (to IS) state. The transition to IS is pending the correction of off-normal conditions on the entity.
Maintenance (MT)	The entity has been manually removed from service for maintenance. It is still capable of performing its provisioned functions. Traffic is still carried.
Disabled (DSBLD)	The entity has been manually removed from service and cannot perform its provisioned functions. All of its provisioned services are disrupted. Traffic is not carried.
Loopback (LPBK)	The loopback command is in effect (any loopback type).

2.1.11.5 Topological Links
Topological links are unidirectional or bidirectional for ONS 15600 SONET NEs.

2.1.11.6 PTPs
All PTPs are bidirectional for the ONS 15600 SONET. These ports are always in channelized mode. PTPs support In Service and Out of Service–Maintenance values for the service state attribute.

2.1.11.7 CTPs
All CTPs are in channelized mode for the ONS 15600 SONET. Alarm monitoring cannot be turned on or off for CTPs.

2.1.11.8 SNCs
One ONS 15600 SONET NE supports a maximum of 1536 STS SNCs.

2.1.11.9 Protection
The ONS 15600 SONET supports 1+1 protection to create redundancy for optical ports. Protect ports must match the working ports. For example, port 1 on an OC-48 card can be protected only by port 1 on another OC-48 card.

2.1.11.10 Autodiscovery
The ONS 15600 SONET supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an AVC event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.12 ONS 15600 SDH

See the Release Notes for Cisco Prime Optical 10.7 for the ONS 15600 SDH software releases that Prime Optical 10.7 supports.

2.1.12.1 Slots
The ONS 15600 SDH is a single-shelf NE with 14 slots. The TSC card must occupy slot 5 or slot 10. The CXC cards are always 1+1 redundant in slots 6/7 and 8/9. Each CXC card occupies two slots. Slots 1 to 4 and 11 to 14 are reserved for optical I/O cards. SSXC cards replace CXC cards from release 5.0 of the NE. For more information, refer to the Cisco ONS 15600 SDH user documentation.

2.1.12.2 Topological Links
Topological links are unidirectional or bidirectional for ONS 15600 SDH NEs.

2.1.12.3 PTPs
All PTPs are bidirectional for the ONS 15600 SDH. These ports are always in channelized mode. PTPs support In Service and Out of Service–Maintenance values for the service state attribute.

2.1.12.4 CTPs

All CTPs are in channelized mode for the ONS 15600 SDH. Alarm monitoring cannot be turned on or off for CTPs.

2.1.12.5 SNCs

One ONS 15600 SDH NE supports a maximum of 1536 VC4 SNCs.

2.1.12.6 Protection

The ONS 15600 SDH supports 1+1 protection to create redundancy for optical ports. Protect ports must match the working ports. For example, port 1 on an STM-16 card can be protected only by port 1 on another STM-16 card.

2.1.12.7 Autodiscovery

The ONS 15600 SDH supports automatic discovery of connected elements. Prime Optical can connect to one node and retrieve information about all connected nodes. The first time Prime Optical connects to a node, it retrieves only the IP address of the connected node; Prime Optical does not retrieve the node name. However, Prime Optical must report this element to the users. Because the IP address is unique, Prime Optical initializes the name of the newly discovered element with the IP address. An object creation event is generated for this managed element with the IP address as the name. Later, Prime Optical connects to this element and retrieves all information, including the actual name, and an AVC event is generated for the managed element name. The NMS listens to the AVC event for the managed element name and invokes managedElementManager::ManagedElementManager_I:: getManagedElement.

2.1.13 Carrier Packet Transport 200/600 SONET/SDH

Prime Optical manages the following NE types for the CPT System:

- CPT 200
- CPT 200 SDH
- CPT 600
- CPT 600 SDH

These NEs support a subset of the ONS15454 and ONS 15454 SDH features. The features are limited to the card and port types listed in 8.12 Equipment List for CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH.

2.1.14 Unmanaged NEs

Unmanaged NEs are other vendor MEs in Prime Optical. There is no equipment for these NEs. Prime Optical reports PTPs for unmanaged NEs only if a topological link originates from or terminates on the unmanaged NE. There are no alarms reported for unmanaged NEs.

2.2 Prime Optical-Specific Details

2.2.1 Layer Rate

✎ The following table reports the layer rates supported by Prime Optical. The layer rates are used for tasks such as link and circuit management, PM retrieval, port provisioning, and alarm handling. The Comments column contains further details on usage. Layer rates numbered 0 to 91 are TMF-defined, and layer rates numbered 92 and above are Prime Optical-defined.

Table 2-16: Layer Rates Supported by Prime Optical 10.7

Layer	Type	Layer Rate	Value	Comments

Layer	Type	Layer Rate	Value	Comments
Physical	Gigabit	LR_Physical_2_5_Gigabit_ITU	108	OTU1 physical link
		LR_Physical_10_Gigabit_ITU	107	OTU2 physical link
		LR_Physical_40_Gigabit_ITU	152	OTU3 physical link
		LR_8b10b_2_5_Gigabit	121	—
		LR_Physical_100_Gigabit_ITU	10018	—
	Ethernet	LR_Gigabit_Ethernet	68	—
		LR_Fast_Ethernet	61	—
		LR_Ten_Gigabit_Ethernet	106	WDM OCHCCs
		LR_Forty_Gigabit_Ethernet	162	—
		LR_DSR_Gigabit_Ethernet	87	Physical links
		LR_LEX	164	—
		LR_OneHundred_Gigabit_Ethernet	10019	—
	SONET/ SDH	LR_DSR_OC1_STM0	72	—
		LR_DSR_OC3_STM1	73	—
		LR_DSR_OC12_STM4	74	WDM OCHCCs
		LR_DSR_OC24_STM8	75	—
		LR_DSR_OC48_STM16	76	Physical SONET/SDH
		LR_DSR_OC192_STM64	77	—
		LR_DSR_OC768_STM256	78	—
	FICON	LR_FICON_1G	112	—
		LR_FICON_2G	113	—
		LR_FICON_4G	137	—
	Fiber Channel	LR_FC_12_133M	62	120M FC
		LR_FC_25_266M	63	250M FC
		LR_FC_50_531M	64	500M FC
		LR_FC_100_1063M	65	1G FC
		LR_FC_200_2125M	111	2G FC links
		LR_FC_400_4250M	134	4G FC
		LR_FC_800_8500M	160	8G FC
		LR_FC_1000_10518M	133	10G FC
	Optical	LR_PHYSICAL_OPTICAL	47	—
	Electrical	LR_PHYSICAL_ELECTRICAL	46	—
Optical	OTS	LR_Optical_Transmission_Secti	42	OTS links
Optical Multiplex	OMS	LR_Optical_Multiplex_Section	41	—
		LR_OMS_DWDM_18_CH	125	OMS links
		LR_OMS_DWDM_18_CH	126	—
Optical Channel	OCH	LR_Optical_Channel	40	For WDM wavelength
		LR_OCH_2_5_Gigabit_ITU	110	OCH links
		LR_OCH_10_Gigabit_ITU	109	—
		LR_OCH_40_Gigabit_ITU	163	—
		LR_OCH_100_Gigabit_ITU	10020	—
		LR_OCH_Transport_Unit	10001	For OCH Trail Tunnel

Layer	Type	Layer Rate	Value	Comments
		LR_OCH_OTU1	10010	OTU PPC
		LR_OCH_OTU2	10011	
		LR_OCH_OTU3	10012	
		LR_OCH_OTU4	10013	
Optical Transport	Path, Section, FEC	LR_Optical_Transport_Path	116	—
		LR_Optical_Transport_Section	117	—
		LR_Optical_Transport_FEC	118	—
		LR_CPO_DSR_OTU1	10009	—
		LR_ENCRYPTION	10021	Encription on OTN channel
SONET/SDH	OC/STM Line/Section	LR_Line_OC1_STS1_and_MS_	24	—
		LR_Line_OC3_STS3_and_ MS_STM1	25	—
		LR_Line_OC12_STS12_and_	26	—
		LR_Line_OC24_STS24_and_M	89	—
		LR_Line_OC48_STS48_and_	27	—
		LR_Line_OC192_STS192_and_	28	—
		LR_Line_OC768_STS768_and_	91	—
		LR_Section_OC1_STS1_and_R	19	—
		LR_Section_OC3_STS3_and_R	20	—
		LR_Section_OC12_STS12_and_	21	—
		LR_Section_OC24_STS24_and_	88	SONET/SDH links
		LR_Section_OC48_STS48_and_	22	—
		LR_Section_OC192_STS192_an	23	—
		LR_Section_OC768_STS768_an	90	—
		LR_SONET_SDH_Composite	10002	For special links created automatically by CTC and used for SONET/SDH circuit routing through DWDM transponders.
	CCAT VCAT SNC	LR_STS1_and_AU3_High_Orde	14	Only SONET OC cards
		LR_Low_Order_TU3_VC3	13	Only SDH STM cards
		LR_STS3c_and_AU4_VC4	15	—
		LR_STS12c_and_VC4_4c	16	Server Trail links
		LR_STS6c_and_VC4_2c	98	SONET/SDH SNCs
		LR_STS9c_and_VC4_3c	99	—
		LR_STS192c_and_VC4_64c	161	—
		LR_VT1_5_and_TU11_VC11	10	—
		LR_VT2_and_TU12_VC12	11	—
		LR_VT6_and_TU2_VC2	12	—
Layer 2	L2	LR_L2_10_Gigabit_Ethernet	153	PPC L2 links
		LR_L2_1_Gigabit_Ethernet	154	—

Layer	Type	Layer Rate	Value	Comments
		LR_L2	159	—
		LR_Not_Applicable	1	ML/CE L2 topologies
		LR_POS_100M	119	—
		LR_POS_Gigabit	120	—
		LR_COS_100M	122	ML/CE PM retrieval
		LR_COS_Gigabit	123	—
		LR_RPR_IEEE_802_17	150	—
		LR_MPLS	165	MPLS-TP tunnels
		LR_MPLS_PATH	166	MPLS-TP tunnel paths
		LR_PW	167	Pseudowire Layer 2
		LR_EVC	168	Ethernet Virtual Circuit
		LR_LACP	169	Link Aggregation
		LR_LAG_FRAGMENT	305	Link
PDH	T*, E*	LR_T1_and_DS1_1_5M	2	—
		LR_T2_and_DS2_6M	3	—
		LR_T3_and_DS3_45M	4	—
		LR_E1_2M	5	—
		LR_E2_8M	6	—
		LR_E3_34M	7	—
		LR_E4_140M	8	—
		LR_E5_565M	9	—
ATM	VC, VP, UNI/NNI	LR_ATM_NI	43	For ATM Network
		LR_ATM_VP	44	For ATM Virtual Paths
		LR_ATM_VC	45	For ATM Virtual
Video	D1, HDTV, DV	LR_D1_Video	58	Video-capable port
		LR_HDTV	146	—
		LR_DV_6000	148	—
		LR_DVBASI	156	—
		LR_SDI_3G_VIDEO	10005	—
		LR_SD_SDI	10003	—
		LR_HD_SDI	10004	—
Data/Storag e	Escon, ETR, ISC, IB, SYSPLEX	LR_ESCON	59	IBM protocol for Mainframe
		LR_ETR	60	IBM protocol for
		LR_ISC3_PEER_1G	143	—
		LR_ISC3_PEER_2G	144	—
		LR_ISC1	155	—
		LR_IB	157	—
		LR_SYSPLEX_ISC_COMP	102	Infiniband
		LR_SYSPLEX_ISC_PEER	103	—
		LR_SYSPLEX_TIMER_ETR	104	—
		LR_SYSPLEX_TIMER_CLO	105	—
		LR_PASS_THROUGH	149	—

Layer	Type	Layer Rate	Value	Comments
		LR_ISC3_STP_1G	10006	—
		LR_ISC3_STP_2G	10007	—
		LR_AUTO	10008	—

2.2.2 Inventory

The ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, ONS 15600 SONET, and ONS 15600 SDH are single-shelf NEs. However, in future they may be expanded to multishelf, multibay NEs. Considering this, Prime Optical reports the following:

- Rack = 1 and shelf = 1 for ONS 15310, ONS 15327, ONS 15600 SONET, and ONS 15600 SDH inventory. This implementation allows Prime Optical to accommodate future changes quickly.
- ONS 15454 MSTP NEs support up to eight shelves per NE. The shelf ID identifies a specific shelf on the NE. The rack ID remains 1 for all shelves.
- For the ONS 15454 SDH, the EFCA shelf and the main shelf are considered to be a single logical shelf with slots numbering from 1 to 29.
- For the ONS NCS2K SDH and ONS NCS2K SONET, the EFCA shelf and the main shelf are considered to be a single logical shelf with slots numbering from 1 to 29.
- Rack = 1 and shelf = 1 for ONS 15216 NEs.
- For the ONS 15216 DCU, Prime Optical reports the DCU chassis as the shelf. Prime Optical reports two slots as equipment holders contained by the shelf. Any DCM module is reported as equipment under the slot.
- For the ONS 15216 FlexLayer, Prime Optical reports the FlexLayer chassis as the shelf. Prime Optical reports four slots as equipment holders contained by the shelf. Any provisioned module is reported as equipment under the slot.
- For the remaining ONS 15216 NEs, Prime Optical reports only one type of equipment under the shelf. The equipment name is the same as the NE name.
- For some supported NEs, equipment does not fit in the normal equipment holders (rack, shelf, slot, and so on) and does not have provisioning functions. For inventory purposes, Prime Optical defines a special EquipmentHolder named AdditionalHolder to hold this equipment.
- For the ONS 15454 SONET R3.4 and later, the Alarm Interface Extension (AIE) is reported in AdditionalHolder 1 (if installed). The Alarm Extension Panel (AEP) is reported in AdditionalHolder 2.
- For the ONS 15454 SDH R3.4 and later, the AIE (if installed) is reported in AdditionalHolder1.
- For the ONS 15600 SONET R5.0 and later, pluggable port modules (PPMs) and pluggable I/O modules (PIMs) on ASAP cards are reported as subslots and port holders, respectively.
- For the ONS 15600 SDH R8.0, PPMs and PIMs on ASAP cards are reported as subslots and port holders, respectively.
- For unmanaged NEs, Prime Optical reports only ManagedElement. Prime Optical does not report any equipment holder or equipment.
- For storage NEs, Prime Optical reports only ManagedElement. Prime Optical does not report any equipment holder or equipment.
- For M2 NEs, Prime Optical reports the slots with a provisioned card as equipment holders; this is applicable only to slots 1 to 3. Other equipment is not managed.

- For M6 NEs, Prime Optical reports the slots with a provisioned card as equipment holders; this is applicable only to slots 1 to 8. Other equipment is not managed.

2.2.2.1 Passive Unit Inventory

In Prime Optical 10.7, passive units are supported, but there are some restrictions in the inventory parameters.

The following table lists the available cards and inventory details.**Cards**	**Inventory Fields Availability**
PPMESH - 4	No
PPMESH - 8	No
PPMESH-SMR	Yes
15216 MD 40 ODD	Yes
15216 MD 40 EVEN	Yes
MD-ID-50	No
15216-MD-ID-50	No
15216-FLC-8-MD	No
PASSIVE OSC	No
PASSIVE DCU	No
PASSIVE MD-48-ODD	Yes
PASSIVE MD-48-EVEN	Yes
PASSIVE MD-48-CM	Yes
PASSIVE FBGDCU	Yes
MF-4x4-COFS	Yes
MF-MPO-8LC	Yes
MF-DEG-5	Yes
MF-UPG-4	Yes
MF-16AD-CFS	Yes
MF-10AD-CFS	Yes
MF-16AE-CFS	Yes
MF-MPO-16LC	Yes
MF-2MPO-ADP	Yes
MF-PPMESH8-5AD	Yes

2.2.3 Interfaces

Unless specified otherwise, all getXXX methods are synchronous.
Prime Optical supports all of the ONS 15xxx-series NEs. However, methods supported for one NE type might not be supported for another NE type. In this case, Prime Optical returns a NOT_IMPLEMENTED ProcessingFailureException for the unsupported method.

✐ Prime Optical supports TMF-defined and Cisco-proprietary application programming interfaces (APIs) and data structures. The Cisco-proprietary APIs and data structures are subject to change in future releases of Prime Optical to accommodate the introduction of new features or to maintain TMF

compliance. This is especially true if the TMF were to define the same method with a different behavior.

In such cases, the affected API name is changed without changing the behavior. The renamed API retains the signature and behavior of the original API.
A standard deprecation period of one Prime Optical release is allowed on the obsolete APIs and data structure. However, Cisco is willing to consider exceptions that minimize the business impact to customers.

2.2.4 Object Details

2.2.4.1 Multilayer Subnetwork

A subnetwork is a collection of managed elements that reflects network connectivity, in most cases. There are two ways to create a new subnetwork through the Prime Optical client:

- In the Domain Explorer window, choose File > Add Network Element(s). Choose <SELF> as the GNE ID and <System Default> as the subnetwork ID. Prime Optical creates a new subnetwork and assigns a name to the new subnetwork object. The naming convention is Subnetwork-ID, where ID = 00000001, 00000002, n.
- Create a subnetwork object from the Prime Optical client by choosing File > Subnetwork Explorer. In the Subnetwork Explorer window, select a network partition and choose File > Add New Subnetwork. Enter the subnetwork name.

There are four ways to add NEs to subnetworks:

- Use the Prime Optical client to specify a subnetwork for the new NE. If the new NE is a GNE and you choose <System Default> as the subnetwork ID, the new subnetwork is created as described in the first bullet above.
- If the new NE is not a GNE, choose an existing subnetwork as the subnetwork ID. Prime Optical adds this NE to the subnetwork.
- If the new NE is connected to an existing NE on the network, Prime Optical discovers the NE and automatically adds it to the existing NE's subnetwork. No user action is required.
- Use the Prime Optical client to move the NE from one subnetwork to another subnetwork.

SNC creation is limited within a multilayer subnetwork scope. For example, assume Subnetwork-1 contains two NEs, A and B, that are connected to each other with an autodiscovered link. Subnetwork-2 contains two NEs, C and D, that are connected to each other with an autodiscovered link. There is also a manually routable link between NEs B and C. To create an SNC between NEs A and D, you must move NEs C and D into Subnetwork-1, or move NEs A and B into Subnetwork-2.
Prime Optical includes a feature to automatically group NEs in subnetworks. This feature is enabled by default. If a routable link is created between two NEs in different subnetworks and Prime Optical merges the two subnetworks, the result is reflected in GateWay/CORBA. The two subnetworks are merged automatically when the Automatically Group NEs in Subnetworks feature is enabled in Prime Optical.

2.2.4.2 SNCs

SNCs are supported only for ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SDH, ONS NCS2K SONET, ONS 15600 SONET, and ONS 15600 SDH NEs. Prime Optical does not support SNC in pending state. However, Prime Optical SNC has a proprietary deleting state.

2.2.4.3 AVC Events

In certain situations, Prime Optical cannot provide the list of changed object attributes. In this case, Prime Optical generates an AVC event with an empty attribute list. The NMS makes a get call to retrieve the object details.

2.2.4.4 Managed Element and Object Creation Event

You can use the Prime Optical client to add new managed elements. If the name of the new element is incorrect, an object creation event is generated with the incorrect name. Later, when Prime Optical connects to the actual NE, it retrieves the correct name. Prime Optical generates an AVC event to notify the NMS of the correct name.

2.2.5 Naming Conventions

The following sections describe naming conventions for objects that Prime Optical reports.

2.2.5.1 Multilayer Subnetwork

You define the name in the Prime Optical client. If you do not specify a name, Prime Optical assigns a default name.

2.2.5.2 Managed Element

The ME reports the name.

2.2.5.3 Physical Termination Point

The PTP name represents the position of the PTP with respect to the equipment. For example, port number 4 on equipment in slot 15 is reported as /rack=1/shelf=1/slot=15/port=4.

For multishelf NEs, the rack and shelf number must be included in the PTP name. ONS 15454 MSTP NEs support up to eight shelves per NE. The shelf ID identifies a specific shelf on the NE. The rack ID remains 1 for all shelves.

This PTP naming convention is used to represent the front Ethernet ports for the ONS 15454 SONET and ONS 15454 SDH ML-series cards. The back Ethernet ports are reported as floating termination points (FTPs).

For class of service (CoS) data on data cards, the PTP name specifies the interface (POS, Fast Ethernet, Gigabit Ethernet), direction (input/output), and CoS level (0 to 7).

For example, CoS data with a POS interface, input direction, and CoS level 3 for slot 1, port 1 is reported as rack=1/shelf=1/slot=1/port=1:POS/Input/3.

Similarly, CoS data with a Fast Ethernet interface, output direction, and CoS level 5 for slot 1, port 1 is reported as /rack=1/shelf=1/slot=1/port=1:FastEthernet/Output/5.

For the ONS 15600 SONET, the ASAP card on slot 3, port 1 on PPM 3 and PIM 2 is reported as /rack=1/shelf=1/slot=3/sub_slot=2/ppm_holder=3/port=1.

For the ONS 15310 CL, ONS 15310 MA SONET, and ONS 15310 MA SDH, a port on PPM number 2 is reported as /rack=1/shelf=1/slot=2/ppm_holder=2/port=1.

For the ONS 15454 SONET and ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH, a port on PPM number 2 is reported as /rack=1/shelf=1/slot=2/ppm_holder=2/port=1.

For unmanaged NEs, the PTP native name is a free-format string.

For 100G TXP-series cards (100G-LC-C, 10x10G-LC, and M-CFP-LC) on ONS 15454 SONET and ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH, Prime Optical uses /rack=1/shelf=1/slot=3/ppm_holder=1/port=1 for main payload, and /rack=1/shelf=1/slot=3/ppm_holder=1/port=1:<Internal Port Number> for internal ports in CFP and CXP.

For ONS 15454 SONET and SDH, ONS NCS2K SONET, and ONS NCS2K SDH, NEs, when 10x10G_LC is configured in FANOUT mode, the CXP sublanes provisioned on the NE use "/rack=1/shelf=1/slot=3/ppm_holder=11/port=1:<portNum>"

The provisionable <portNum> values are 2,3,4,5,6,7,8,9,10 and 11.

✐ Each of the ports (<portNum>) provisioned under ppm_holder=11 is a main PTP.

2.2.5.4 Connection Termination Point

Naming conventions for SONET, SDH, and plesiochronous digital hierarchy (PDH) CTP are defined by TMF.814 in a document named objectNaming.html. Prime Optical follows these rules.

For OC-n or STM-n cards on the ONS 15327 R3.3 or later, ONS 15454 SONET R3.2.1 or later, and ONS 15454 SDH R3.3 or later, Prime Optical supports STS6c, STS9c, and STS24c layer rates. On the ONS 15327 R5.0 or later, ONS 15454 SONET R5.0 or later, and ONS 15454 SDH R5.0 or later, Prime Optical supports STS18c and STS36c layer rates. These are not conventional SONET/SDH rates; therefore, the TMF document does not cover their naming.

The following table shows the Prime Optical CTPs for STS6c, STS9c, STS24c, STS18c, and STS36c layer rates. The number after '-r' is the difference between the STS/VC4 number and the previous edge STS/VC4 number.

Table 2-17: CTPs for STS6c, STS9c, STS18c, STS24c, and STS36c

Layer Rate	CTP Tuple	Comments
PTP: OC12		
sts6c_vc4_2c	/sts6c_vc4_2c=1, 2	CTP is on the edge of sts6c, STS#1 and STS#7.
	/sts6c_vc4_2c=1-r=[1..5]	CTP is not on the edge of sts6c. For example, if CTP is on STS#4, the name is /sts6c_vc4_2c=1-r=3.
sts9c_vc4_3c	/sts9c_vc4_3c=1	CTP is on the edge of sts9c, STS#1.
	/sts9c_vc4_3c=1-r=[1..3]	CTP is not on the edge of sts9c. For example, if CTP is on STS#4, the name is /sts9c_vc4_3c=1-r=3.
PTP: OC48		
sts6c_vc4_2c	/sts6c_vc4_2c=[1..8]	CTP is on the edge of sts6c, STS#1, STS#7… STS#43.
	/sts6c_vc4_2c=[1..7]-r=[1..5]	CTP is not on the edge of sts6c. For example, if CTP is on STS#10, the name is /sts6c_vc4_2c=2-r=3.
sts9c_vc4_3c	/sts9c_vc4_3c=[1..5]	CTP is on the edge of sts9c, STS#1, STS#10… STS#37.
	/sts9c_vc4_3c=[1..5]-r=[1..8]	CTP is not on the edge of sts9c. For example, if CTP is on STS#16, the name is /sts9c_vc4_3c=2-r=6.
sts24c_vc4_8c	/sts24c_vc4_8c=1, 2	CTP is on the edge of sts24c, STS#1, STS#25.
	/sts24c_vc4_8c=1-r=[1..23]	CTP is not on the edge of sts24c. For example, if CTP is on STS#10, the name is /sts24c_vc4_8c=1-r=9.
sts18c_vc4_6c	/sts18c_vc4_6c=1, 2	CTP is on the edge of sts18c, STS#1, STS#19.
	/sts18c_vc4_6c=1-r=[1..17]	CTP is not on the edge of sts18c. For example, if CTP is on STS#10, the name is /sts18c_vc4_6c=1-r=9.
sts36c_vc4_12c	/sts36c_vc4_12c=1	CTP is on the edge of sts36c, STS#1.
	/sts36c_vc4_12c=1-r=[1..35]	CTP is not on the edge of sts36c. For example, if CTP is on STS#10, the name is /sts36c_vc4_12c=1-r=9.
PTP: OC192		
sts6c_vc4_2c	/sts6c_vc4_2c=[1..32]	CTP is on the edge of sts6c, STS#1, STS#7… STS#187.
	/sts6c_vc4_2c=[1..31]-r=[1..5]	CTP is not on the edge of sts6c. For example, if CTP is on STS#4, the name is /sts6c_vc4_2c=1-r=3.

Layer Rate	CTP Tuple	Comments
sts9c_vc4_3c	/sts9c_vc4_3c=[1..21]	CTP is on the edge of sts9c, STS#1, STS#10… STS#181.
	/sts9c_vc4_3c=[1..20]-r=[1..8]	CTP is not on the edge of sts9c. For example, if CTP is on STS#52, the name is /sts9c_vc4_3c=6-r=6.
sts18c_vc4_6c	/sts18c_vc4_6c=1,10	CTP is on the edge of sts18c, STS#1, STS#19.
	/sts18c_vc4_6c=1-r=[1..17]	CTP is not on the edge of sts18c. For example, if CTP is on STS#10, the name is /sts18c_vc4_6c=1-r=9.
sts24c_vc4_8c	/sts24c_vc4_8c=[1..8]	CTP is on the edge of sts24c, STS#1, STS#25… STS#169.
	/sts24c_vc4_8c=[1..7]-r=[1..23]	CTP is not on the edge of sts24c. For example, if CTP is on STS#61, the name is /sts24c_vc4_8c=3-r=12.
sts36c_vc4_12c	/sts36c_vc4_12c=1,5	CTP is on the edge of sts36c, STS#1, STS#37.
	/sts36c_vc4_12c=1-r=[1..35]	CTP is not on the edge of sts36c. For example, if CTP is on STS#10, the name is /sts36c_vc4_12c=1-r=9.
PTP: STM4		
sts6c_vc4_2c	/sts6c_vc4_2c=1,2	CTP is on the edge of vc4_2c, VC4#1, and VC4#3.
	/sts6c_vc4_2c=1-r=1	CTP is on VC4#2.
sts9c_vc4_3c	/sts9c_vc4_3c=1	CTP is on the edge of vc4_3c, VC4#1.
	/sts9c_vc4_3c=1-r=1	CTP is on VC4#2.
PTP: STM16		
sts6c_vc4_2c	/sts6c_vc4_2c=[1..8]	CTP is on the edge of vc4_2c, VC4#1, VC4#3… VC4#15.
	/sts6c_vc4_2c=[1..7]-r=1	CTP is not on the edge of vc4_2c. For example, if CTP is on VC4#4, the name is /sts6c_vc4_2c=2-r=1.
sts9c_vc4_3c	/sts9c_vc4_3c=[1..5]	CTP is on the edge of vc4_3c, VC4#1, VC4#4… VC4#13.
	/sts9c_vc4_3c=[1..5]-r=[1..2]	CTP is not on the edge of vc4_3c. For example, if CTP is on VC4#6, the name is /sts9c_vc4_3c=2-r=2.
sts18c_vc4_6c	/sts18c_vc4_6c=1,2	CTP is on the edge of vc4_6c, VC4#1, VC4#7.
	/sts18c_vc4_6c=1-r=[1..5]	CTP is not on the edge of vc4_6c. For example, if CTP is on VC4#3, the name is /sts18c_vc4_6c=1-r=2.

Layer Rate	CTP Tuple	Comments
sts24c_vc4_8c	/sts24c_vc4_8c=1,2	CTP is on the edge of vc4_8c, VC4#1, VC4#9.
	/sts24c_vc4_8c=1-r=[1..7]	CTP is not on the edge of vc4_8c. For example, if CTP is on VC4#4, the name is /sts24c_vc4_8c=1-r=3.
sts36c_vc4_12c	/sts36c_vc4_12c=1	CTP is on the edge of vc4_12c, VC4#1.
	/sts36c_vc4_12c=1-r=[1..11]	CTP is not on the edge of vc4_12c. For example, if CTP is on VC4#10, the name is /sts36c_vc4_12c=1-r=9.
PTP: STM64		
sts6c_vc4_2c	/sts6c_vc4_2c=[1..32]	CTP is on the edge of vc4_2c, VC4#1, VC4#3… VC4#63.
	/sts6c_vc4_2c=[1..31]-r=1	CTP is not on the edge of vc4_2c. For example, if CTP is on VC4#20, the name is /sts6c_vc4_2c=10-r=1.
sts9c_vc4_3c	/sts9c_vc4_3c=[1..21]	CTP is on the edge of vc4_3c, VC4#1, VC4#4… VC4#61.
	/sts9c_vc4_3c=[1..21]-r=[1..3]	CTP is not on the edge of vc4_3c. For example, if CTP is on VC4#6, the name is /sts9c_vc4_3c=2-r=2.
sts18c_vc4_6c	/sts18c_vc4_6c=1,9	CTP is on the edge of vc4_6c, VC4#1, VC4#7.
	/sts18c_vc4_6c=1-r=[1..5]	CTP is not on the edge of vc4_6c. For example, if CTP is on VC4#3, the name is /sts18c_vc4_6c=1-r=2.
sts24c_vc4_8c	/sts24c_vc4_8c=[1..8]	CTP is on the edge of vc4_8c, VC4#1, VC4#9… VC4#57.
	/sts24c_vc4_8c=[1..7]-r=[1..7]	CTP is not on the edge of vc4_8c. For example, if CTP is on VC4#28, the name is /sts24c_vc4_8c=4-r=3.
sts36c_vc4_12c	/sts36c_vc4_12c=1,4	CTP is on the edge of vc4_12c, VC4#1.
	/sts36c_vc4_12c=1-r=[1..11]	CTP is not on the edge of vc4_12c. For example, if CTP is on VC4#10, the name is /sts36c_vc4_12c=1-r=9.

In addition, the following CTP name support has been added:

PTP	Layer Rate	CTP Tuple
STM1	vt2_tu12	/sts3c_au4-j=1/vt2_tu12-k=[1..3]-l=[1..7]-m=[1..3] The qualifier strings represent the following values: j—AUG index k—TUG-3 or AU-3 index l—TUG-2 index m—TU-12 or TU-11 index

For CTPs used for WDM SNCs (optical channel network connection [OCHNC], optical channel client connection [OCHCC], and OCH trail tunnels), use the naming conventions defined in TMF: /frequency=nnn.mm, where nnn.mm is a decimal representing the frequency in terahertz (THz).
For unidirectional CTP, the naming convention is to add "/direction=src or sink" in front of the name. Because the CTPs in WDM SNC are all unidirectional, the name is "/direction=src or sink/frequency=nnn.mm" for WDM SNC CTPs.
For CTPs used for WDM SNCs, the frequency notation in the naming convention of the CTP refers to the working path only. The frequency of the protected path is identified by the ProtectedFrequency attribute passed in the additionalInfo section of the SNC creation request if one of following conditions is satisfied:

- SNC type is OCHCC in a non-GMPLS circuit with splitter protection.
- SNC type is OCHCC in a GMPLS circuit with splitter protection.
- SNC type is OCHCC in a GMPLS circuit with Y cable protection.

Because the Prime Optical client uses wavelengths to represent an optical channel, enter the corresponding wavelength value in the nativeEMSName field for the CTP.
The formula for converting wavelength to frequency is:

frequency (THz) = 299792458 / wavelength (nm) / 1000

The formula for converting frequency back to wavelength is:

wavelength (nm) = 299792458 / frequency (THz) / 1000

The possible values of the frequency nnn.mm (in THz) in the CTP and the corresponding wavelength values are:

For OCHNC:

- C-band

195.90(1530.33), 195.80(1531.12), 195.70(1531.90), 195.60(1532.68), 195.40(1534.25), 195.30(1535.04), 195.20(1535.82), 195.10(1536.61), 194.90(1538.19), 194.80(1538.98), 194.70(1539.77), 194.60(1540.56), 194.40(1542.14), 194.30(1542.94), 194.20(1543.73), 194.10(1544.53), 193.90(1546.12), 193.80(1546.92), 193.70(1547.72), 193.60(1548.51), 193.40(1550.12), 193.30(1550.92), 193.20(1551.72), 193.10(1552.52), 192.90(1554.13), 192.80(1554.94), 192.70(1555.75), 192.60(1556.55), 192.40(1558.17), 192.30(1558.98), 192.20(1559.79), 192.10(1560.61).

- L-band

1577.86, 1578.69, 1579.52, 1580.35, 1581.18, 1582.02, 1582.85, 1583.69, 1584.53, 1585.36, 1586.20, 1587.04, 1587.88, 1588.73, 1589.57, 1590.41, 1591.26, 1592.10, 1592.95, 1593.79, 1594.64, 1595.49, 1596.34, 1597.19, 1598.04, 1598.89, 1599.75, 1600.60, 1601.46, 1602.31, 1603.17, 1604.03.

For OCHCC:

- C-band (odd)

1529.55 nm, 1530.33 nm, 1531.12 nm, 1531.90 nm, 1532.68 nm, 1533.47 nm, 1534.25 nm, 1535.04 nm, 1535.82 nm, 1536.61 nm, 1537.40 nm, 1538.19 nm, 1538.98 nm, 1539.77 nm, 1540.56 nm, 1541.35 nm, 1542.14 nm, 1542.94 nm, 1543.73 nm, 1544.53 nm, 1545.32 nm, 1546.12 nm, 1546.92 nm, 1547.72 nm, 1548.51 nm, 1549.32 nm, 1550.12 nm, 1550.92 nm, 1551.72 nm, 1552.52 nm, 1553.33 nm, 1554.13 nm, 1554.94 nm, 1555.75 nm, 1556.55 nm, 1557.36 nm, 1558.17 nm, 1558.98 nm, 1559.79 nm, 1560.61 nm, 1561.42 nm.

- C-band (even)

1529.94 nm, 1530.73 nm, 1531.51 nm, 1532.29 nm, 1533.07 nm, 1533.86 nm, 1534.64 nm, 1535.43 nm, 1536.22 nm, 1537.00 nm, 1537.79 nm, 1538.58 nm, 1539.37 nm, 1540.16 nm, 1540.95 nm, 1541.75 nm, 1542.54 nm, 1543.33 nm, 1544.13 nm, 1544.92 nm, 1545.72 nm, 1546.52 nm, 1547.32 nm, 1548.12 nm, 1548.92 nm, 1549.71 nm, 1550.52 nm, 1551.32 nm, 1552.12 nm, 1552.93 nm, 1553.73 nm, 1554.54 nm, 1555.34 nm, 1556.15 nm, 1556.96 nm, 1557.77 nm, 1558.58 nm, 1559.39 nm, 1560.20 nm, 1561.01 nm, 1561.83 nm.

- L-band (odd)

1570.83 nm, 1571.65 nm, 1572.48 nm, 1573.30 nm, 1574.13 nm, 1574.95 nm, 1575.78 nm, 1576.61 nm, 1577.44 nm, 1578.27 nm, 1579.10 nm, 1579.93 nm, 1580.77 nm, 1581.60 nm, 1582.44 nm, 1583.27 nm, 1584.11 nm, 1584.95 nm, 1585.78 nm, 1586.62 nm, 1587.46 nm, 1588.30 nm, 1589.15 nm, 1589.99 nm, 1590.83 nm, 1591.68 nm, 1592.52 nm, 1593.37 nm, 1594.22 nm, 1595.06 nm, 1595.91 nm, 1596.76 nm, 1597.62 nm, 1598.47 nm, 1599.32 nm, 1600.17 nm, 1601.03 nm, 1601.88 nm, 1602.74 nm, 1603.60 nm.

- L-band (even)

1571.24 nm, 1572.06 nm, 1572.89 nm, 1573.71 nm, 1574.54 nm, 1575.37 nm, 1576.20 nm, 1577.03 nm, 1577.86 nm, 1578.69 nm, 1579.52 nm, 1580.35 nm, 1581.18 nm, 1582.02 nm, 1582.85 nm, 1583.69 nm, 1584.53 nm, 1585.36 nm, 1586.20 nm, 1587.04 nm, 1587.88 nm, 1588.73 nm, 1589.57 nm, 1590.41 nm, 1591.26 nm, 1592.10 nm, 1592.95 nm, 1593.79 nm, 1594.64 nm, 1595.49 nm, 1596.34 nm, 1597.19 nm, 1598.04 nm, 1598.89 nm, 1599.75 nm, 1600.60 nm, 1601.46 nm, 1602.31 nm, 1603.17 nm, 1604.03 nm.

The supported wavelengths range is from 1528.77 nm to 1566.72 nm.
For circuit creation, the supported wavelengths range is from 1530.33 nm to 1561.42 nm.

L-Band				C-Band				S-Band			
100 GHz Grid		50 GHz Offset		100 GHz Grid		50 GHz Offset		100 GHz Grid		50 GHz Offset	
THz	nm	THz	nm	THz	nm	THz	nm	THz	Nm	THz	nm
186.00	1611.78	186.05	1611.35	191.00	1569.59	191.05	1569.18	196.00	1529.55	196.05	1529.16
186.10	1610.92	186.15	1610.48	191.10	1568.77	191.15	1568.36	196.10	1528.77	196.15	1528.38
186.20	1610.05	186.25	1609.62	191.20	1567.95	191.25	1567.54	196.20	1527.99	196.25	1527.60
186.30	1609.19	186.35	1608.76	191.30	1567.13	191.35	1566.72	196.30	1527.21	196.35	1526.82
186.40	1608.32	186.45	1607.89	191.40	1566.31	191.45	1565.90	196.40	1526.43	196.45	1526.04
186.50	1607.46	186.55	1607.03	191.50	1565.49	191.55	1565.08	196.50	1525.66	196.55	1525.27
186.60	1606.60	186.65	1606.17	191.60	1564.67	191.65	1564.27	196.60	1524.88	196.65	1524.49
186.70	1605.74	186.75	1605.31	191.70	1563.86	191.75	1563.45	196.70	1524.11	196.75	1523.72
186.80	1604.88	186.85	1604.45	191.80	1563.04	191.85	1562.63	196.80	1523.33	196.85	1522.94
186.90	1604.02	186.95	1603.59	191.90	1562.23	191.95	1561.82	196.90	1522.56	196.95	1522.17
187.00	1603.16	187.05	1602.73	192.00	1561.41	192.05	1561.01	197.00	1521.78	197.05	1521.40
187.10	1602.31	187.15	1601.88	192.10	1560.60	192.15	1560.20	197.10	1521.01	197.15	1520.63
187.20	1601.45	187.25	1601.02	192.20	1559.79	192.25	1559.38	197.20	1520.24	197.25	1519.86
187.30	1600.60	187.35	1600.17	192.30	1558.98	192.35	1558.57	197.30	1519.47	197.35	1519.09
187.40	1599.74	187.45	1599.31	192.40	1558.17	192.45	1557.76	197.40	1518.70	197.45	1518.32
187.50	1598.89	187.55	1598.46	192.50	1557.36	192.55	1556.95	197.50	1517.93	197.55	1517.55
187.60	1598.04	187.65	1597.61	192.60	1556.55	192.65	1556.15	197.60	1517.16	197.65	1516.78
187.70	1597.18	187.75	1596.76	192.70	1555.74	192.75	1555.34	197.70	1516.40	197.75	1516.01
187.80	1596.33	187.85	1595.91	192.80	1554.94	192.85	1554.53	197.80	1515.63	197.85	1515.25
187.90	1595.48	187.95	1595.06	192.90	1554.13	192.95	1553.73	197.90	1514.86	197.95	1514.48
188.00	1594.64	188.05	1594.21	193.00	1553.32	193.05	1552.92	198.00	1514.10	198.05	1513.72
188.10	1593.79	188.15	1593.36	193.10	1552.52	193.15	1552.12	198.10	1513.33	198.15	1512.95
188.20	1592.94	188.25	1592.52	193.20	1551.72	193.25	1551.31	198.20	1512.57	198.25	1512.19
188.30	1592.10	188.35	1591.67	193.30	1550.91	193.35	1550.51	198.30	1511.81	198.35	1511.43
188.40	1591.25	188.45	1590.83	193.40	1550.11	193.45	1549.71	198.40	1511.05	198.45	1510.67
188.50	1590.41	188.55	1589.98	193.50	1549.31	193.55	1548.91	198.50	1510.28	198.55	1509.90
188.60	1589.56	188.65	1589.14	193.60	1548.51	193.65	1548.11	198.60	1509.52	198.65	1509.14

L-Band				C-Band				S-Band			
100 GHz Grid		50 GHz Offset		100 GHz Grid		50 GHz Offset		100 GHz Grid		50 GHz Offset	
188.70	1588.72	188.75	1588.30	193.70	1547.71	193.75	1547.31	198.70	1508.76	198.75	1508.38
188.80	1587.88	188.85	1587.46	193.80	1546.91	193.85	1546.51	198.80	1508.01	198.85	1507.63
188.90	1587.04	188.95	1586.62	193.90	1546.11	193.95	1545.72	198.90	1507.25	198.95	1506.87
189.00	1586.20	189.05	1585.78	194.00	1545.32	194.05	1544.92	199.00	1506.49	199.05	1506.11
189.10	1585.36	189.15	1584.94	194.10	1544.52	194.15	1544.12	199.10	1505.73	199.15	1505.36
189.20	1584.52	189.25	1584.10	194.20	1543.73	194.25	1543.33	199.20	1504.98	199.25	1504.60
189.30	1583.68	189.35	1583.27	194.30	1542.93	194.35	1542.53	199.30	1504.22	199.35	1503.84
189.40	1582.85	189.45	1582.43	194.40	1542.14	194.45	1541.74	199.40	1503.47	199.45	1503.09
189.50	1582.01	189.55	1581.60	194.50	1541.34	194.55	1540.95	199.50	1502.71	199.55	1502.34
189.60	1581.18	189.65	1580.76	194.60	1540.55	194.65	1540.16	199.60	1501.96	199.65	1501.59
189.70	1580.35	189.75	1579.93	194.70	1539.76	194.75	1539.37	199.70	1501.21	199.75	1500.83
189.80	1579.51	189.85	1579.10	194.80	1538.97	194.85	1538.58	199.80	1500.46	199.85	1500.08
189.90	1578.68	189.95	1578.27	194.90	1538.18	194.95	1537.79	199.90	1499.71	199.95	1499.33
190.00	1577.85	190.05	1577.43	195.00	1537.39	195.05	1537.00	200.00	1498.96	200.05	1498.58
190.10	1577.02	190.15	1576.61	195.10	1536.60	195.15	1536.21	200.10	1498.21	200.15	1497.83
190.20	1576.19	190.25	1575.78	195.20	1535.82	195.25	1535.42	200.20	1497.46	200.25	1497.09
190.30	1575.36	190.35	1574.95	195.30	1535.03	195.35	1534.64	200.30	1496.71	200.35	1496.34
190.40	1574.54	190.45	1574.12	195.40	1534.25	195.45	1533.85	200.40	1495.97	200.45	1495.59
190.50	1573.71	190.55	1573.30	195.50	1533.46	195.55	1533.07	200.50	1495.22	200.55	1494.85
190.60	1572.88	190.65	1572.47	195.60	1532.68	195.65	1532.28	200.60	1494.47	200.65	1494.10
190.70	1572.06	190.75	1571.65	195.70	1531.89	195.75	1531.50	200.70	1493.73	200.75	1493.36
190.80	1571.23	190.85	1570.82	195.80	1531.11	195.85	1530.72	200.80	1492.99	200.85	1492.61
190.90	1570.41	190.95	1570.00	195.90	1530.33	195.95	1529.94	200.90	1492.24	200.95	1491.87

For NCS2K SONET and NCS2K SDH NEs, the supported wavelength range for circuit creation is from 1528.77 nm to 1566.72 nm. In addition, the following wavelengths in nanometers (nm) (frequency THz) are supported by legacy cards:

C-BAND odd

- 1528.77(196.10)
- 1529.55(196.00)
- 1562.23(191.90)
- 1563.05(191.80)
- 1563.86(191.70)
- 1564.68(191.60)
- 1565.50(191.50)
- 1566.31(191.40)

C-BAND even

- 1529.16(196.05)
- 1529.94(195.95)
- 1562.64(191.85)
- 1563.45(191.75)
- 1564.27(191.65)
- 1565.09(191.55)
- 1565.90(191.45)
- 1566.72(191.35)

The CTPs when retrieved may assume one of the follow frequency values:

- NA_OR_1ST_TUNABLE: No channel is associated to the CTP and no frequency is available, or it is not possible to get the frequency associated to the CTP because of the CTP configuration.
- UNKNOWN: The frequency associated to the CTP cannot be determined.

2.2.5.5 Floating Termination Point

The FTP naming contains three tuples. The first tuple is the EMS name, the second tuple is the ME name, and the third tuple is the FTP name. The value of the name field in the third tuple is “FTP.” The native FTP name is a free-format string. However, GateWay/CORBA follows the convention for PTP and CTP, if applicable.

All Ethernet termination points (TPs) for circuit provisioning on E-series, G-series, ML-series, CE-100T-8, and FCMR cards are modeled as FTPs. The convention for the FTP values for Ethernet TPs is as follows:

- For FTPs used in single-card EtherSwitch, Prime Optical uses /rack=1/shelf=1/slot=n/EtherSwitchGroup.
- For FTPs used in multicard EtherSwitch, Prime Optical uses /MulticardEtherSwitchGroup.
- For FTPs used in linear-card EtherSwitch or G-series cards, Prime Optical uses /rack=1/shelf=1/slot=n/port=m.
- For FTPs used in ML-series cards, Prime Optical uses /rack=1/shelf=1/slot=n/port=m, where m = 0,1.
- To create or report VT tunnel, Prime Optical uses /VTTunnel.
- To create or report VC tunnel, Prime Optical uses /VCTunnel.

- To create VC tunnel for VC3 port grouping and SNCs, Prime Optical uses /rack=1/shelf=1/slot=n/vc4=m, where n is from 1 to 17 and m is from 1 to 4.
- For VT aggregation point circuits used to create or report VT aggregation point (VAP) circuits, Prime Optical uses /VAPTP for SONET NEs.
- For low-order aggregation point circuits, Prime Optical uses /LAPTP for SDH NEs.
- For virtual ports used in 100G TXP-series cards (100G-LC-C, and M-CFP-LC) on ONS 15454 SONET and ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH, Prime Optical uses /rack=1/shelf=1/slot=3/vport=1.

2.2.5.6 Equipment

For a list of equipment that GateWay/CORBA reports for each NE type, see Chapter 8, "Equipment List for NEs."

2.2.5.7 Equipment Holder

The TMF defines the EquipmentHolder naming convention. Numbering is done from left to right and from top to bottom. For example, slot 15 is named /rack=1/shelf=1/slot=15.

For ONS 15xxx NEs, the numbering for EquipmentHolders starts from 0 in the NE. The TMF convention is to start numbering from 1 for EquipmentHolders. Prime Optical translates EquipmentHolder names. For example, subslot 1 in slot 0 in the NE is named /rack=1/shelf=1/slot=1/sub_slot=2.

For the ONS 15454 MSTP, ONS NCS2K SONET, and ONS NCS2K SDH NEs, shelves are numbered using their unique shelf ID, and not their physical location attributes, such as rack number and shelf position. For example, slot 1 on an MSTP shelf with an ID of 7 that is physically located in rack 1, shelf 2 is named /rack=1/shelf=7/slot=1.

For the ONS 15600 NE and ASAP cards, a PIM present in slot 3 is named /rack=1/shelf=1/slot=3/sub_slot=1. A PPM present in PIM 3 in slot 2 is named /rack=1/shelf=1/slot=2/sub_slot=3/ppm_holder=1.

For AdditionalHolder, the naming convention is defined as /additionalHolder=1; there is no subholder for AdditionalHolder. AdditionalHolders apply only to ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH NEs, when AIE or AEP cards are installed.

A peripheral_holder present on a shelf is named rack=1/shelf=1/ /peripheral_holder=17236110, where the number 17236110 is the unique peripheral ID.

A peripheral_holder in slot 1 is named /rack=1/shelf=1/slot=1/peripheral_holder=17236111, where the number 17236111 is the unique peripheral ID.

2.2.5.8 Remote Unit Holder

- CPT System is also referred to as "PT System" or "PTS" in the documentation.
- From Prime Optical 9.6 onward, the remote unit holder is supported on CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH NEs.

In Prime Optical 10.7, Carrier Packet Transport (CPT) System is supported. The remote unit holder manages remote equipment as an extension to the local equipment being modeled. This new holder applies to PTSA cards connected to the PT System.

The remote unit holder naming convention is "/remote_unit=1". The remote unit holder is at the same level as the rack. For example, to address a PTSA card, the holder slot must be addressed as "/remote_unit=1/shelf=1/slot=36". The remote unit ID value is 1 and the PTSA card slot can be any number from 36 to 128.

- For the PTSA card, the ONS 15454 "FOG" identifier is translated into the slot number.

Example

```
Equipment Type     = EQT_HOLDER

Name         = [0] EMS=Cisco Systems/Prime Optical; [1] ManagedElement=M6-65-
117;
```

```
[2] EquipmentHolder=/remote_unit=1/shelf=1
User Label              = Native EMS Name = shelf1
Owner                     =
Alarm Reporting Indicator = false
Holder Type                        = shelf
Expected Or Installed Equipment    = [0] EMS=Cisco Systems/Prime Optical; [1]
ManagedElement=M6-65-117;
[2] EquipmentHolder=/remote_unit=1/shelf=1;
[3] Equipment=PTSA-50
acceptable Equipment Types  = [00] FILLER
Holder State                       = INSTALLED_AND_EXPECTED
Additional Info                  =
```

2.2.5.9 Passive Unit Holder

In Prime Optical 10.7, passive units are supported. The new passive unit holder manages passive units as extensions of local equipment being modeled.
The syntax is "unit=1" and the range is from 1 to 128.

Example

```
Equipment Type                      = EQT_HOLDER
Name                                 = [0] EMS=Cisco Systems/Prime Optical;
             [1] ManagedElement=15454-ANSI-68-6;
             [2] EquipmentHolder=/rack=1/unit=1
User Label                          =
Native EMS Name               = unit1
Owner                                     =
Alarm Reporting Indicator     = false
Holder Type                         = unit
Expected Or Installed Equipment     =
[0] EMS=Cisco Systems/Prime Optical;
[1] ManagedElement=15454-ANSI-68-6;
[2] EquipmentHolder=/rack=1/unit=1;
[3] Equipment=Passive MD 40 Odd
acceptable Equipment Types  = [00] FILLER
Holder State                            = UNKNOWN
Additional Info                       =
```

The passive unit holder applies to the following cards:

- PPMESH - 4
- PPMESH - 8
- 15216 MD 40 EVEN
- 15216 MD 40 ODD
- MD-ID-50
- 15216-MD-ID-50

- 15216-FLD-4
- 15216-FLC-8-MD
- PASSIVE OSC
- PASSIVE DCU
- PPMESH-SMR
- MF-4x4-COFS
- MF-MPO-8LC
- MF-DEG-5
- MF-UPG-4
- MF-16AD-CFS
- MF-10AD-CFS
- MF-16AE-CFS
- MF-PPMESH8-5AD
- MF-2MPO-ADP
- MF-MPO-16L

2.2.5.10 Passive Shelf Unit Holder

In Prime Optical 10.7, passive Shelf units are supported. The new passive shelf unit holder manages passive cards and is an EquipmentHolder for the passive cards. The syntax used for passive Shelf unit is "**psunit=1**".

Example

```
Equipment Type = EQT_HOLDER
Name = [0] EMS=Cisco Systems/PrimeOptical; [1] ManagedElement=ncs2006-239-58; [2] EquipmentHolder=/rack=1/psunit=6/slot=4
User Label =
Native EMS Name = psunit6
Owner =
Alarm Reporting Indicator = false
Holder Type = psunit
Expected Or Installed Equipment = [0] EMS=Cisco Systems/PrimeOptical; [1] ManagedElement=ncs2006-239-58; [2] EquipmentHolder =/rack=1/psunit=6/slot=4; [3] Equipment=MF-4x4-COFS
acceptable Equipment Types = [00] FILLER
Holder State = INSTALLED_AND_EXPECTED
Additional Info =
```

Passive Shelf PID	Passive Modules Supported
NCS2K-MF-6RU	• MF-2MPO-ADP • MF-4X4-COFS

Passive Shelf PID	Passive Modules Supported
	• MF-6AD-CFS • MF-DEG-5 • MF-MPO-8LC • MF-UPG-4
NCS2K-MF10-6RU	• MF-10AD-CFS • MF-16AD-CFS • MF-16AE-CF5 • MF-8X10G-FO • MF-MPO-16LC • MF-MPO-20LC

2.2.5.11 Topological Links

Prime Optical discovers topological links and assigns a name with the following default format:
MEName A:slot number/port number-ME Name B:slot number/port number
Integer values are used for slot and port number. You can use the Prime Optical client to change the link name.
For example:
NE454A:Wdm0/0—NE454B:Wdm1/0
Prime Optical reports two topological links for a Y-cable link. Both topological links have the same nativeEMSName, but different values in the second tuple of the topological link name. The names "<nativeEMSName>:::1" and "<nativeEMSName>:::2" identify the two legs of the Y-cable link.

✐ Whenever you create a new topological link, the object returned by the API reflects the topological link and the corresponding state at that particular moment. To obtain a definitive state, you should execute the inventory APIs only after the topological link creation is complete. You should also wait until you receive a corresponding notification.

2.2.5.12 Passive Unit Links

The following table lists the cards and termination availability for passive unit links.

Table 2-18: Cards and Termination Availability

Cards	Termination Availability
PPMESH - 4	No
PPMESH - 8	No
PPMESH-SMR	No
15216 MD 40 ODD	Yes
15216 MD 40 EVEN	Yes
MD-ID-50	No
15216-MD-ID-50	No
15216-FLD-4	Yes
15216-FLC-8-MD	No
PASSIVE OSC	No
PASSIVE DCU	No
MF-4x4-COFS	Yes
MF-MPO-8LC	Yes

Cards	Termination Availability
MF-DEG-5	Yes
MF-UPG-4	Yes
MF-16AD-CFS	Yes

2.2.5.13 Protection Group

The protection group name reported by the NE is used as the PGP native name for 1_FOR_1, 1_FOR_N, and 1_PLUS_1 and Y_CABLE protection groups. The ring ID is used as the PGP native name for 2_FIBER_BLSR and 4_FIBER_BLSR protection groups. “<Ring Id>-EAST” and “<Ring Id>-WEST” are used as the native names for the two component groups of 4_FIBER_BLSR PGP.

2.2.5.14 L2 Topology

An L2 topology represents a point-to-point, hub-and-spoke, or Resilient Packet Ring (RPR) Layer 2 network topology. These topologies exist over the underlying Layer 1 topology, which is formed by physical topological links. Each L2 topology name is identified in the context of an EMS and has two tuples. The first tuple represents the EMS name, and the second tuple represents the nativeEMSName of the L2 topology. The L2 topology name is unique within an EMS. Each L2 topology can be associated with multiple ML VLANs.

2.2.5.15 ML VLAN

Each ML VLAN is associated with an L2 topology and is unique for a given L2 topology. The fully qualified ML VLAN name is identified by three tuples. The first tuple represents the EMS name, the second tuple represents the L2 topology name, and the last tuple represents the nativeEMSName of the ML VLAN (which is the VLAN ID). A maximum of 255 ML VLANs can be created per L2 topology, ranging from 1 to 4095.

2.2.5.16 QoS Template

The QoS template is defined in the scope of an EMS domain. The fully qualified QoS template name is identified by two tuples. The first tuple represents the EMS name and the second tuple represents the QoS template name.

2.2.5.17 VCAT

The fully qualified VCAT name is identified by two tuples. The first tuple represents the EMS name and the second tuple represents the VCAT name, which is the same as the nativeEMSName of the VCAT.

2.2.5.18 AID

AID is an array of the NameAndStringValue type. NamingAttributes represent the hierarchical name structure of an object. The structure of the name is hierarchical and reflects the containment relationship between objects in a simple way. The following convention is used for the field name. AID is used for alarm purposes only.

Name	Type	Description
AID	globaldefs::NamingAttributes_T	meName[0] = new NameAndStringValue_T("EMS", "Cisco Systems/Prime Optical”) meName [1] = new NameAndStringValue_T("ManagedElement", "CTM-454-141"); meName [2] = new NameAndStringValue_T("AID",

Name	Type	Description
		"notificationId=10");

2.2.6 Programming Details

2.2.6.1 Boolean Mapping

The following boolean parameters apply:

- On = True
- Off = False

2.2.6.2 Resource Cleanup

Many TMF interfaces that Prime Optical implements return iterator objects for large volumes of data. These iterators are allocated Prime Optical resources. Prime Optical supports a maximum of 128 iterators. If the iterator limit is reached, the NMS receives an EXCPT_TOO_MANY_OPEN_ITERATORS exception.

By default, Prime Optical cleans up all iterator objects if they are not accessed within six hours.

2.2.6.3 Development Environment

GateWay/CORBA has been developed with Java Development Kit (JDK) 1.6.0_23 and jacORB 2.x1.3.7 on Sun Solaris 10 and is compliant with the CORBA 2.3 specification. The OSS can use Java or the C++ IDL compiler to compile IDL files.

✎ If you use Java and jacORB, the Prime Optical server installation provides JAR files for notification IDLs and TMF IDLs.

The thread_pool_max property indicates the maximum number of concurrent threads that Prime Optical can handle. By default, the thread_pool_max property is set to 200 in the */Prime Optical-installation-directory/bin/jcorbagw.sh* file. If the value is exceeded, the following exception is returned:

```
org.omg.CORBA.TRANSIENT: Server-side Exception: resource limit reached
```

2.2.6.4 Exception Handling

All interfaces that GateWay/CORBA implements raise the exception globaldefs::ProcessingFailureException. The following sections discuss each exception type in detail.

2.2.6.4.1 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.

2.2.6.4.2 EXCPT_INTERNAL_ERROR

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

2.2.6.4.3 EXCPT_INVALID_INPUT

This exception indicates an incorrect parameter format, such as a three-level naming attribute 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.

2.2.6.4.4 EXCPT_ENTITY_NOT_FOUND

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

2.2.6.4.5 EXCPT_UNABLE_TO_COMPLY

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

2.2.6.4.6 EXCPT_NE_COMM_LOSS

This exception is used as a generic value when the 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.

2.2.6.4.7 EXCPT_ACCESS_DENIED

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

2.2.6.4.8 EXCPT_TOO_MANY_OPEN_ITERATORS

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

2.2.6.4.9 EXCPT_USERLABEL_IN_USE

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

2.2.6.4.10 EXCPT_STRICT_MERGE_FAILED

This exception is raised when an OSS issues an upgrade request to merge circuits with a STRICT MERGE operation type. The OSS uses this exception to issue the upgrade request with a LOOSE MERGE operation type.

2.2.7 Provisioning Subnetwork Connections

SNC provisioning must be within a multilayer subnetwork scope, which means the aEnd and the zEnd of an SNC must be in the same multilayer subnetwork.
Prime Optical defines the method multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createAndActivateSNCFromUserLabel for SNC provisioning.

2.2.7.1 SNC Provisioning on OC-n or Electrical Equipment

Prime Optical supports SNC provisioning on OC-n or electrical cards for ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS 15600 SONET, ONS NCS2K SONET, and ONS NCS2K SDH NEs.
CTPs are used as the endpoints to create the SNC.

2.2.7.2 SNC Provisioning on Ethernet Equipment

ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, CPT 200/600 SONET, CPT 200/600 SDH, ONS NCS2K SONET, and ONS NCS2K SDH managed elements support Ethernet equipment. SNCs can be created on this equipment to carry Ethernet traffic. The TMF standard supports only SONET, SDH, DWDM, and ATM; it does not support Ethernet at this time.
Provisioning SNCs on E-series, G-series, and ML-series equipment supported by the ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, ONS 15454 SONET, and ONS 15454 SDH is different from creating SNCs on SONET or PDH equipment. For SNCs on Ethernet equipment, Prime Optical uses FTP to model the endpoint. See Floating Termination Point for naming conventions.
The three modes supported for SNC provisioning on Ethernet equipment are:

- Single-card mode—E-series cards; VLANs can be associated during creation.
- Multicard mode—E-series cards; VLANs can be associated during creation.
- Linear-card mode—E-series, G-series, and ML-series cards.

To obtain available source and destination FTPs for SNC provisioning, use the following methods:

- managedElementManager::ManagedElementManager_I::
  getAllSrcFTPsForSNCProvisioning
- managedElementManager::ManagedElementManager_I::
  getAllDestFTPsForSNCProvisioning

2.2.7.2.1 Single-Card Mode and Multicard Mode

By default, all E-series cards are in the multicard mode EtherSwitch group. Each node has only one multicard mode EtherSwitch group; the group exists even if the node has no Ethernet cards. A single-card EtherSwitch group can exist only if a node has at least one Ethernet card.

- To create SNCs in multicard mode, both aEnd and zEnd must be in the multicard EtherSwitch group.
- To create SNCs in single-card mode, one end can be an OC-n CTP. You must create a bidirectional SNC.

The following table shows the maximum bandwidth allowed for single- and multicard EtherSwitches.

Table 2-19: Single-Card and Multicard Mode

ONS 15327 Single Card	ONS 15454 SONET/SDH Single Card	ONS 15327 Multicard	ONS 15454 SONET/SDH Multicard
Six STS1s	Six STS1s and two STS3c (SONET only)	Three STS1s	Six STS1s (SONET only)
Two STS3c	Four STS3c (VC4)	One STS3c	Two STS3c (VC4)
One STS6c	One STS6c (VC4_2c) and six STS1s (SONET only)	—	One STS6c (VC4_2c)
One STS12c	One STS6c (VC4_2c) and two STS3c (VC4)	—	—
—	Two STS6c (VC4_2c)	—	—
—	One STS12c (VC4_4c)	—	—

2.2.7.2.2 Linear-Card Mode

You can apply linear-card mode SNC provisioning to all types of Ethernet cards. One end of the SNC must be an FTP; the other end can be a CTP on any OC-n port, or an FTP. The FTPs must use linear-mode naming conventions.

Only single source and single destination are supported for the SNC creation, and the SNC must be bidirectional.

For E-series and G-series cards, the port number in the FTP name is the number of the front Ethernet port. For ML-series cards, the port number is the number of the virtual back-end port.

2.2.7.2.3 Layer Rates

Prime Optical supports the following layer rates on E-series cards:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4
- LR_STS6c_and_VC4_2c
- LR_STS12c_and_VC4_4c

Prime Optical supports the following layer rates on G-series cards:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4
- LR_STS6c_and_VC4_2c
- LR_STS9c_and_VC4_3c

- LR_STS12c_and_VC4_4c
- LR_STS24c_and_VC4_8c
- LR_STS48c_and_VC4_16c

Prime Optical supports the following layer rates on ML-series cards:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4
- LR_STS6c_and_VC4_2c
- LR_STS9c_and_VC4_3c
- LR_STS12c_and_VC4_4c
- LR_STS24c_and_VC4_8c

Prime Optical supports the following layer rates on FCMR cards:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4

Prime Optical supports the following layer rate on CE-100T-8 and ML-100T-8 cards:

- LR_STS1_and_AU3_High_Order_VC3

2.2.7.2.4 SNC Provisioning for ONS 15454 CE-MR-10 Cards

The ONS 15454 CE-MR-10 card implements two different working modes:

- Automatic mode—Available only for ONS 15454 SONET
- Manual (or Exposed) mode—Available for ONS 15454 SONET and ONS 15454 SDH

Automatic mode is similar to the provisioning of existing ML-series cards, where FTPs must be used for CCAT and VCAT circuit provisioning, and the FTP name is the number of the virtual back-end port.
Manual mode requires PTPs/CTPs to be specified for the aEnd and zEnd on CCAT circuits, and for aEnd(s) and zEnd(s) on VCAT member circuits. For VCAT circuit endpoints, FTPs must be used.
Use one of the following APIs to retrieve available CTPs:
3.8.20 getAllSrcCTPsForVCATMemberSNCProvisioning
3.8.23 getAllDestCTPsForVCATMemberSNCProvisioning
Use the API getAllSrcPTP/CTP/FTPForCircuitProvisioning to retrieve the possible endpoints, taking into account the current card mode.

✐ You cannot change the card mode from the GateWay/CORBA interface if there is a circuit on the card.

To determine whether a CE-MR-10 card is configured for Manual or Automatic mode, use the getEquipment API, which returns the CardMode parameter.
The following layer rates are supported for CCAT circuits:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4
- LR_STS6c_and_VC4_2c
- LR_STS9c_and_VC4_3c
- LR_STS12c_and_VC4_4c
- LR_STS24c_and_VC4_8c
- LR_STS48c_and_VC4_16c

The following layer rates are supported for VCAT member circuits:

- LR_STS1_and_AU3_High_Order_VC3
- LR_STS3c_and_AU4_VC4
- LR_VT1_5_and_TU11_VC11

2.2.7.2.5 SNC Provisioning for ONS 15454 ADM-10G Cards

The ADM_10G card ports carry the OCN and GE traffic. You can provision each of the first eight ports as 1GE ports that carry GE traffic. These ports are, by default, GFP-encapsulation enabled. However, if required, you can configure them as HDLC.

GateWay/CORBA can manage CCAT bidirectional SNCs with FTP termination points on ONE_GE_PORT ports ranging from 1 to 8 in the framing protocol types previously listed.

The following table lists the supported layer rates along with their corresponding values:

Layer Rates	Values
STS1	LR_STS1_and_AU3_High_Order_VC3
STS3C	LR_STS3c_and_AU4_VC4
STS6C	LR_STS6c_and_VC4_2c
STS9C	LR_STS9c_and_VC4_3c
STS12C	LR_STS12c_and_VC4_4c
STS24C	LR_STS24c_and_VC4_8c

As opposed to the MSPP data cards and the OC cards, FTPs on ADM_10G client ports contain reference to ppm_holder as well. For example:

```
15454-ANSI-68-8
/rack=1/shelf=1/slot=5/ppm_holder=2/port=1
```

2.2.7.3 Tunnel SNC Provisioning

Prime Optical supports tunnel SNC provisioning on ONS 15310, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH NEs. The tunnel SNC must be bidirectional. The tunnel SNC for the ONS 15310, ONS 15327, ONS 15454 SONET, ONS NCS2K SONET, and ONS NCS2K SDH is called VT tunnel, and the layer rate must be 14 (LR_STS1_and_AU3_High_Order_VC3). The tunnel SNC for the ONS 15454 SDH is called VC tunnel, and the layer rate must be 15 (LR_STS3c_and_AU4_VC4). See Naming Conventions for naming conventions.

Only one aEnd and zEnd are specified for the VT/VC tunnel SNC provisioning.

Prime Optical supports only fully automatically routed VT/VC tunnel SNCs. Prime Optical does not support manually routed and constraint-based automatically routed VT/VC tunnel SNCs.

Prime Optical supports VC tunnel for VC3 port grouping and SNC provisioning on ONS 15454 SDH. For VC tunnel for VC3 port grouping and SNC provisioning, if the source or destination end is an STM-n card, it is modeled as CTP; if the source or destination end is on an E3, DS3I, or DS3IN card, it is modeled as FTP.

The following table shows the combinations of cards that can be selected for creating the VC4 tunnel for VC3 port grouping circuits.

Table 2-20: Tunnel SNC Provisioning

A-End	Z-End
DS3I or DS3IN	DS3I, DS3IN, or STM-n
E3	E3 or STM-n
STM-n	DS3I, DS3IN, or E3

When VC tunnel for VC3 port grouping and SNC provisioning is complete:

- One VC4 tunnel SNC is created with the SNC name name:PGT1
- Three VC3 SNCs are created with the SNC names name:PGC1, name:PGC2, and name:PGC3

Deleting one PGC SNC results in the deletion of all four SNCs. You cannot delete a PGT SNC if any PGC SNCs exist. See Naming Conventions for the naming conventions.

2.2.7.4 SNC Provisioning on WDM Equipment

2.2.7.4.1 OCHCC Circuit Provisioning

Prime Optical supports OCHCC circuit provisioning on transponders, muxponders, and ITU-T line cards for ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH NEs.
The OCHCC SNC is created between client ports on transponder or muxponder cards, or on trunk ports on ITU-T line cards.
OCHCC is an extension of OCHNC. OCHCC is a part of the OCHCC SNC. OCHCC can have only one source and one destination.
CTP naming is identical to OCHNC SNCs: GateWay/CORBA determines whether to create an OCHNC or OCHCC, depending on the layer. See Connection Termination Point for the CTP details.
OCHCC must be bidirectional.
Prime Optical supports different sizes corresponding to possible payload types of transponder and muxponder client ports. A specific layer must be defined for each payload type.
OCHCC is either *unprotected or protected*. Protected is created only on protected transponder or muxponder cards where there is a protected trunk port.

- To create OCHCC, you must create patchcord links at an optical channel layer between the transponders, muxponders, or ITU-T line cards and DWDM ports.
- OSPF-detected links between GCC terminations on transponder or muxponder cards are indicated at the Optical_Transport_OTUk layer.

When an OCHCC circuit is created, unless transponder cards are directly connected by a trunk-to-trunk patchcord link, an additional OCH trail tunnel circuit is created automatically.
The following method is defined for protected OCHCC circuits when it is necessary to specify constraints for both working and protected routes:

- multiLayerSubnetwork::MultiLayerSubnetworkMgr_I::
- createAndActivateProtectedOchccFromUserLabel

It is not supported the creation of non-GMPLS PSM and Y-Cable Protected OCHCCs.

2.2.7.4.2 OCHNC Circuit Provisioning

WDM SNCs are either unidirectional or bidirectional. GateWay/CORBA supports the creation of bidirectional OCHNC circuits on Release 5.0 and later NEs. You can also create two unidirectional OCHNC circuits on the same frequency in opposite directions.

The CTPs used in WDM SNCs are different from the CTPs used in SONET/SDH SNCs in terms of naming and nature. See the Connection Termination Point section for the CTP naming conventions. The layer rate for the CTPs and WDM SNC must be LR_Optical_Channel.

For OCHNC on ONS 15454 SONET and ONS 15454 SDH Release 7.0 and later, you do not have to specify the east-to-west or west-to-east direction; instead, UniDir and BiDir must be used. For Release 8.0 NEs, you can specify route constraints (NE include/exclude constraints) in the SNCCreateData_T parameter.

For OCHNC on NCS2K SONET and NCS2K SDH Release 10.7 and later, you can create gmpls circuits only.

When creating OCHNC bidirectional circuits on R5.x or R6.x NEs, you must specify all four circuit endpoints: source, destination, secondary source (the source in the opposite direction), and secondary

destination (the destination in the opposite direction). For R7.x or R8.x NEs you must specify only the source and destination in one direction, because the endpoints for the opposite direction are calculated automatically.
However, with the 80-WXC card (introduced in R9.2), more secondary sources and secondary destinations could be available. In such a case, you must also specify the secondary TP. If it does not belong to the actual list of available PTPs, an EXCPT_UNABLE_TO_COMPLY exception is thrown.
For example:

```
The secondary destination does not belong to the list of available PTPs:
/rack=1/shelf=1/slot=4/port=1, /rack=1/shelf=1/slot=4/port=2,
/rack=1/shelf=1/slot=4/port=3, /rack=1/shelf=1/slot=4/port=4,
/rack=1/shelf=1/slot=4/port=5, /rack=1/shelf=1/slot=4/port=6,
/rack=1/shelf=1/slot=4/port=7, /rack=1/shelf=1/slot=4/port=8,
/rack=1/shelf=1/slot=4/port=9
```

Or:
The secondary source does not belong to the list of available PTPs:

```
/rack=1/shelf=1/slot=2/port=9
```

A mixed configuration, such as an OCHNC bidirectional circuit between an 80WXC card (configured as bidirectional) and a MUX/DMX card, requires only the secondary drop on the 80WXC card. Prime Optical still automatically calculates the secondary source on the MUX/DMX card.
For the ONS 15454 SONET and ONS 15454 SDH R9.0 and later, NCS2K SONET, and NCS2K SDH, OCHNC circuits support protected optical paths. You can specify the following property within the creation data structure: staticProtectionLevel=StaticProtectionLevel_T.FULLY_PROTECTED.
The availability of protected paths and the possibility of provisioning protected OCHNC circuits depend on the presence of the Protection Switch Module (PSM). When configured correctly, the PSM provides one of the available protection schemas.
If the PSM card mode is configured as STANDALONE, you cannot create OCHNC or OCHCC protected circuits that involve the PSM card. The STANDALONE mode can be set and retrieved from the Prime Optical client GUI only. (In the Prime Optical GUI, launch the NE Explorer, select the PSM card, click the Card tab, and configure the Card Mode attribute.)
GateWay/CORBA distinguishes the mode that the NE version uses.

2.2.7.5 Specifying Route Constraints for OCHxx Circuit Provisioning

For OCHCC, OCHNC, and OCH trail tunnel circuits, you can specify route constraints (NE include/exclude constraints) in the SNCCreateData_T parameter.
For protected OCHCC circuits, constraints for the working path are specified in the SNCCreateData_T parameter; constraints for the protected path are specified in a separate parameter. See details in the createAndActivateProtectedOchccFromUserLabel API description.

2.2.7.6 GMPLS Provisioning

Creating an OCHCC GMPLS SNC requires a background request validation. The API performs a first parsing level of passed information. A successful response does not mean that the SNC has been created. The SNC is created only after a successful network validation.
Notification of OCHCC GMPLS SNC creation occurs through an attribute value change event. If the API executes successfully but the network validation fails for any reason, a new notification of an attribute value change event is sent. The notification contains the SNC name and the reason for the failure.
It is also possible to create OCHNC GMPLS circuits using GateWay/CORBA. However, the mandatory set and management of alien wavelengths is not supported by GateWay/CORBA and must be requested from other interfaces.
For OCHCC, OCHNC, and OCH trail circuits, you can provide additional GMPLS information about the associated SNCs.

Table 2-21: GMPLS Information on SNCs

Name	Type	Description/Valid Values
isGmpls	String	Optional. Valid Values are: • true • false Specifies whether or not GMPLS applies to the SNC that is being created. If isGmpls is not specified or is specified and set to false, the following parameters cannot be specified: • gmplsOptQual • gmplsProtType • gmplsAcptThreshold • gmplsProtAcptThreshold
gmplsOptQual	String	Optional. Valid values are: • None—No optical validation is applied while creating the SNC; the attribute gmplsAcptThreshold must be set to Red. • Full—The SNC is created if the optical validation of the required thresholds is satisfied.
gmplsProtType	String	Optional. gmplsProtType is necessary only if isGmpls has been set to true. Represents the protection type associated to the GMPLS SNC. • Valid values for OCHCC SNC are: – NotDefined – PSM – Y Cable – Splitter • Valid values for OCHNC SNC are: – NotDefined – PSM • Valid value for the OCH Trail SNC is NotDefined. gmplsProtType is ignored if isGmpls is not specified or is specified and set to false.
gmplsAcptThreshold	String	Optional. gmplsAcptThreshold is necessary only if isGmpls is set to true. This threshold acceptance value is checked against the actual optical value once the OCHCC SNC is created and discovered. Valid values are: • Green • Yellow • Orange • Red • NotApplicable gmplsAcptThreshold is ignored if isGmpls is not specified or is specified and set to false.

Name	Type	Description/Valid Values
		gmplsAcptThreshold cannot be set to NotApplicable if isGmpls is set to true. gmplsAcptThreshold must be set to Red if isGmpls is set to true and gmplsOptQual is set to None. ✎ In the return message of getSNC or getAllSNC API, gmplsAcptThreshold may assume one of the following values: • Not valid • Out • Not applicable
gmplsProtAcptThreshold	String	gmplsProtAcptThreshold is necessary only if isGmpls is set to true and gmplsProtType is set to a value other than NotDefined. gmplsProtAcptThreshold cannot be set to NotApplicable if the isGmpls value is set to true and gmplsProtType is set to a value other than NotDefined. gmplsProtAcptThreshold must be set to Red if isGmpls is set to true, gmplsProtType is set to a value other than NotDefined, and attribute gmplsOptQual is set to None.
gmplsUserLabel	String	(Required parameter) Only letters, numbers, and underscores (_) are allowed. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsPriority	String	Required Parameter. Valid values are: • N/A • High • Low • 1 • 2 • 3 • 4 • 5 • 6 • Unknown For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsCircuitDiversity	String	(Required parameter) Only letters and numbers are allowed and represent the circuit ID. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsPowerOffsetUp	float	(Required parameter). The range is from -15.0 to 15.0. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsPowerOffsetDown	float	(Required parameter). The range is from -15.0 to 15.0. For more information, see Cisco Prime Optical 10.7 User Guide.

Name	Type	Description/Valid Values
gmplsAllowRegen	String	Required parameter. Valid values are: • true • false gmplsAllowRegen is necessary only if isGmpls is set to true. If set to true, the far-end CTPs will have different frequencies. For more information, see Cisco Prime Optical 10.7 User Guide.

For a create SNC operation or modifySNC API invocation, in the additionalInfo section you can specify the GMPLS Revert/Restore parameters listed in the following table.

For SNC creation, the attributes can be part of additionalInfo only if isGmpls has been specified and its value is true. For SNC modification these attributes can be specified only if the applied SNC is GMPLS.

Table 2-21: GMPLS Revert/Restore Parameters

Name	Type	Description/Valid Values
gmplsIgnorePathAlarms	String	Optional. Valid Values are: • True • False If true, all restoration path alarms are ignored. This attribute is required only while creating SNC. If specified during modifySNC, the request is rejected.
gmplsRestoration	String	Optional. Valid Values are: • True • False If true, the circuit is restorable and the the system searches for a new path when the circuit fails. If false, the system does not look for a new path on circuit failure. If gmplsIgnorePathAlarms is true, this parameter cannot be set to true, since the restoration mechanism is based on receipt of path alarms.
gmplsRevertiveParameters	String	Optional. Valid Values are: • None • Manual • Automatic If the attribute gmplsRestoration is not specified or is set to false, this parameter must not be specified.
gmplsSoakTime	String	Optional. The possible range is between 1 and 86399; that is, 23h:59m:59s. This parameter must not be specified if: • gmplsRestoration is not specified or is set to false • gmplsRevertiveParameters is not specified or is set to none If gmplsRevertiveParameters is specified and set to Manual or Automatic, this parameter must be specified.
gmplsRestoreThresold	String	Similar to gmplsAcptThreshold, but applied to restoration path.
gmplsProtectedRestoreThresold	String	Similar to gmplsProtAcptThreshold, but applied to restoration path.

Name	Type	Description/Valid Values
gmplsRestoreOptQual	String	Similar to gmplsOptQual, but applied to restoration path.

The OCHCC SNC Protection values and the corresponding gmplsProtType values must match. If these values do not match, the request is rejected.
The following table lists the gmplsProtType values and the corresponding SNC Protection values.

gmplsProtType	SNC Protection
Y Cable	Unprotected
PSM	Fully protected
Splitter	Fully protected
NotDefined	Unprotected

✐ The SNC creation for GMPLS circuits that are Y-cable protected require that SNC protection be set to Unprotected.

2.2.7.7 SubnetworkConnection_T structure information

Prime Optical support for the SubnetworkConnection_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported. This field contains two tuples. The first tuple contains values for the EMS. The second tuple contains the SNC name, which is the same as nativeEMSName.
- string userLabel
  - Supported.
- string nativeEMSName
  - Supported. The name supplied by the user at creation time is reported.
- string owner
  - Not supported.
- SNCState_T sncState
  - Prime Optical supports SNCS_NONEXISTENT, SNCS_ACTIVE, SNCS_ACTIVE_PREROUTED, SNCS_DELETING, and SNCS_PARTIAL.

✐ When the SNC is not in SNCS_ACTIVE state, this structure might not have complete information about the SNC. For example, transmissionParameters for TPs reported might not be initialized. When the SNC goes into SNCS_ACTIVE state, a state change event is generated (see section 5.4.4 State Change Event). Then the NMS invokes the multiLayerSubnetwork::MultiLayerSubnetworkMgr_I::getSNC method to retrieve complete information about the SNC.

- globaldefs::ConnectionDirection_T direction
  - Supported.

- transmissionParameters::LayerRate_T rate
  - Supported.
- StaticProtectionLevel_T staticProtectionLevel
  - Supported. For PCA circuits, the value is set to PREEMPTIBLE.
- SNCType_T sncType
  - Not supported.
- TPDataList_T aEnd
  - Supported.
- TPDataList_T zEnd
  - Supported.
- Reroute_T rerouteAllowed
  - Not supported.
- NetworkRouted_T networkRouted
  - Not supported.
- globaldefs::NVSList_T additionalInfo
  - Supported. The customer ID and service ID are reported if they are available. For an SNC on Ethernet equipment, Prime Optical reports all associated VLAN IDs. For a WDM SNC, the sncSize is reported.

For OCHCC SNCs, the frequency value associated to the OCHCC circuit refers to the active path. Each OCH trail reports its own frequency value. It can be different for the OCH trail subtending to the same OCHCC if the implemented protection is either “splitter” or “Y cable” and the SNC is GMPLS.

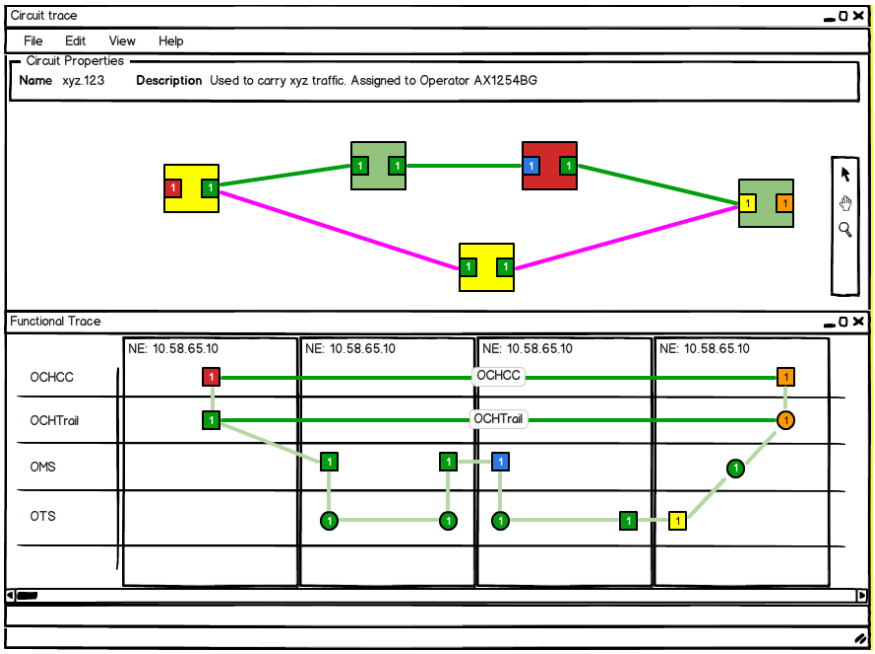
For OCHCC SNCs, GMPLS information, if any, is also reported. The following table lists the GMPLS information that could be reported.

Attribute Name	Type	Description
gmplsAcptThreshold	String	This threshold acceptance value is checked against the actual optical value. The threshold acceptance value can assume one of the following values: • Green • Yellow • Orange • Red
gmplsOptValid	String	The current optical valid value. The optical valid value can assume one of the following values: • Green • Yellow • Orange • Red

Attribute Name	Type	Description
gmplsRestorationState	String	Valid values are: • None • Restored and Revertible • Restoring • Restored • Failed
gmplsIgnorePathAlarms	String	Optional. Valid Values are: • True • False If true, all restoration path alarms are ignored. This attribute is required only while creating SNC. If specified during modifySNC, the request is rejected.
gmplsRestoration	String	Optional. Valid Values are: • True • False If true, the circuit is restorable and the the system searches for a new path when the circuit fails. If false, the system does not look for a new path on circuit failure. If gmplsIgnorePathAlarms is true, this parameter cannot be set to true, since the restoration mechanism is based on receipt of path alarms.
gmplsRevertiveParameters	String	Optional. Valid Values are: • None • Manual • Automatic If the attribute gmplsRestoration is not specified or is set to false, this parameter must not be specified.
isGmpls	String	Valid Values are: • True, if the retrieved SNC is a GMPLS SNC • False, if the retrieved SNC is not a GMPLS SNC
CircuitAdminState	String	Valid SDH values are: • UNLOCKED • LOCKED,DISABLED • LOCKED,MAINTENANCE • UNLOCKED,AUTO_IN_SERVICE • LOCKED,OUT_OF_GROUP • Valid SONET values are: • IN_SERVICE • OUT_OF_SERVICE • OUT_OF_SERVICE_BY_MAINTENANCE • AUTO_IN_SERVICE • OUT_OF_GROUP • Valid for ALL • UNKNOWN_ADMIN_STATE
gmplsUserLabel	String	(Required parameter) Only letters, numbers, and underscores (_) are allowed. For more information, see the Cisco Prime Optical 10.7 User

Attribute Name	Type	Description
		Guide.
gmplsPriority	String	Required Parameter. Valid values are: • N/A • High • Low • 1 • 2 • 3 • 4 • 5 • 6 • Unknown For more information, see the Cisco Prime Optical 10.7 User Guide.
gmplsCircuitDiversity		(Required parameter) Only letters and numbers are allowed and represent the circuit ID. For more information, see the Cisco Prime Optical 10.7 User Guide.
gmplsPowerOffsetUp	float	(Required parameter) The range is from -15.0 to 15.0. For more information, see the Cisco Prime Optical 10.7 User Guide.
gmplsPowerOffsetDown	float	(Required parameter) The range is from -15.0 to 15.0. For more information, see the Cisco Prime Optical 10.7 User Guide.
gmplsAllowRegen	String	Required parameter. Valid values are: • true • false gmplsAllowRegen is necessary only if isGmpls is set to true. If set to true, the far-end CTPs will have different frequencies. For more information, see the Cisco Prime Optical 10.7 User Guide.
Trail_0 and Trail_1	String	OCHCC circuit has read only attributes. Values report relation between OCHCC and subtending trail circuit.

For OCHCC or Trail relation see figure below:



2.2.7.8 Low-Order Circuit Provisioning Using VAP/LAP

Prime Optical supports low-order circuit provisioning using VAP circuits on ONS 15310 CL, ONS 15310 MA SONET, ONS 15310 MA SDH, ONS 15327, and ONS 15454 SONET NEs, and low-order aggregation point (LAP) circuits on ONS 15454 SDH NEs. The VAP/LAP circuits must be bidirectional. When provisioning VAP/LAP circuits, the source is treated as the STS grooming end; the destination is treated as the VT grooming end and is used later to connect VT circuits.

The source is a CTP and the destination is an FTP. See Connection Termination Point for naming conventions.

3 Using GateWay/CORBA Interfaces

This chapter describes each interface that Cisco Prime Optical supports, including TeleManagement Forum (TMF)-specific and Cisco-proprietary interfaces. This chapter includes the following information:

✎ The iterator interfaces defined by the TMF are not listed here. Prime Optical implementation of these interfaces does not deviate from the TMF definition.

3.1 Overview of APIs Supported by NE Type

The following table lists each API and the NEs that support it.

Table 3-1: APIs Supported by NE Type

API No.	GateWay/CORBA API	Supported NEs
1	common::Common_I:: getCapabilities	—
2	emsMgr::EMSMgr_I:: getAllEMSAndMEActiveAlarms	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
3	emsMgr::EMSMgr_I:: getAllEMSSystemActiveAlarms	—
4	emsMgr::EMSMgr_I:: getAllTopLevelSubnetworks	—

API No.	GateWay/CORBA API	Supported NEs
5	emsMgr::EMSMgr_I:: getAllTopLevelSubnetworkNames	—
6	emsMgr::EMSMgr_I:: getAllTopLevelTopologicalLinks	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
7	emsMgr::EMSMgr_I:: getAllTopLevelTopologicalLinkNames	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
8	emsMgr::EMSMgr_I:: getTopLevelTopologicalLink	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
9	emsMgr::EMSMgr_I:: createTopologicalLink	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH,

API No.	GateWay/CORBA API	Supported NEs
		ONS 15600 SONET, unmanaged NE, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
10	emsMgr::EMSMgr_I:: createTopologicalLinkWithYCableProtection	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
11	emsMgr::EMSMgr_I:: deleteTopologicalLink	ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, unmanaged NE, ONS NCS2K SDH, ONS NCS2K SONET
12	emsMgr::EMSMgr_I:: modifyTopologicalLink	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2, ONS NCS2K SDH, ONS NCS2K SONET SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH
13	emsMgr::EMSMgr_I:: getEMS	—
14	emsMgr::EMSMgr_I:: getAllL2Topologies	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH,

API No.	GateWay/CORBA API	Supported NEs
		ONS 15454 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
15	emsMgr::EMSMgr_I:: getAllQoSTemplateNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
16	emsMgr::EMSMgr_I:: getAllQoSTemplates	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
17	emsMgr::EMSMgr_I:: getQoSTemplate	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
18	emsMgr::EMSMgr_I:: getAllEMSandMEClearedAlarms	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15600 SDH, ONS 15600 SONET,CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
19	emsMgr::EMSMgr_I:: getAllEMSSystemClearedAlarms	All
20	emsMgr::EMSMgr_I:: createServerTrailLink	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH,

API No.	GateWay/CORBA API	Supported NEs
		ONS NCS2K SDH, ONS NCS2K SONET
21	emsMgr::EMSMgr_I:: deleteServerTrailLink	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
22	emsMgr::EMSMgr_I:: addNetworkPartition	—
23	emsMgr::EMSMgr_I:: addMultiLayerSubnetwork	All
24	emsMgr::EMSMgr_I:: deleteNetworkPartition	—
25	emsMgr::EMSMgr_I:: deleteMultiLayerSubnetwork	All
26	emsMgr::EMSMgr_I:: getAllSupportedMEModels	All
27	emsSession::EmsSession_I:: getEventChannel	—
28	emsSession::EmsSession_I:: getManager	All
29	emsSession::EmsSession_I:: getSupportedManagers	All
30	emsSessionFactory::EmsSessionFactory_I:: getEmsSession	—
31	emsSessionFactory::EmsSessionFactory_I:: getEmsPublicKey	—
32	emsSessionFactory::EmsSessionFactory_I:: getEmsPublicKeyPair	—
33	equipment::EquipmentInventoryMgr_I:: getAllEquipment	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
34	equipment::EquipmentInventoryMgr_I:: getAllEquipmentNames	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-

API No.	GateWay/CORBA API	Supported NEs
		M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET,CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
35	equipment::EquipmentInventoryMgr_I:: getAllSupportedPTPs	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
36	equipment::EquipmentInventoryMgr_I:: getAllSupportedPTPNames	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
37	equipment::EquipmentInventoryMgr_I:: getAllSupportingEquipment	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET,CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET

API No.	GateWay/CORBA API	Supported NEs
38	equipment::EquipmentInventoryMgr_I:: getAllSupportingEquipmentNames	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
39	equipment::EquipmentInventoryMgr_I:: getContainedEquipment	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2, ONS 15454-M6, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
40	equipment::EquipmentInventoryMgr_I:: getEquipment	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
41	equipment::EquipmentInventoryMgr_I:: provisionEquipment	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET,

API No.	GateWay/CORBA API	Supported NEs
		ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
42	equipment::EquipmentInventoryMgr_I:: setAlarmReportingOff	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
43	equipment::EquipmentInventoryMgr_I:: setAlarmReportingOn	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
44	equipment::EquipmentInventoryMgr_I:: unprovisionEquipment	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
45	maintenanceOps::MaintenanceMgr_I:: getActiveMaintenanceOperations	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454 SDH,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
46	maintenanceOps::MaintenanceMgr_I:: performMaintenanceOperation	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
47	managedElementManager::ManagedElementManager_I:: getAllActiveAlarms	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
48	managedElementManager::ManagedElementManager_I:: getAllSNCs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200

API No.	GateWay/CORBA API	Supported NEs
		SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
49	managedElementManager::ManagedElementManager_I:: getAllManagedElements	All
50	managedElementManager::ManagedElementManager_I:: getAllManagedElementNames	All
51	managedElementManager::ManagedElementManager_I:: getAllPTPs	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
52	managedElementManager::ManagedElementManager_I:: getAllPTPNames	ONS 15216, ONS 15305, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
53	managedElementManager::ManagedElementManager_I:: getContainedInUseTPs	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200

API No.	GateWay/CORBA API	Supported NEs
		SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
54	managedElementManager::ManagedElementManager_I:: getContainedInUseTPNames	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
55	managedElementManager::ManagedElementManager_I:: getContainedPotentialTPs	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
56	managedElementManager::ManagedElementManager_I:: getContainedPotentialTPNames	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
57	managedElementManager::ManagedElementManager_I:: getContainingSubnetworkNames	All
58	managedElementManager::ManagedElementManager_I::	All

API No.	GateWay/CORBA API	Supported NEs
	getManagedElement	
59	managedElementManager::ManagedElementManager_I:: getTP	ONS 15216, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 , ONS NCS2K SDH, ONS NCS2K SONET SDH, CPT 600, CPT 600 SDH
60	managedElementManager::ManagedElementManager_I:: getAssociatedTopologicalLinks	ONS 15216, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
61	managedElementManager::ManagedElementManager_I::setTPData	ONS 15216, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
62	managedElementManager::ManagedElementManager_I:: getAllSrcPTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA

API No.	GateWay/CORBA API	Supported NEs
		SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
63	managedElementManager::ManagedElementManager_I:: getAllSrcCTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
64	managedElementManager::ManagedElementManager_I:: getAllDestPTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, OONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
65	managedElementManager::ManagedElementManager_I:: getAllDestCTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600

API No.	GateWay/CORBA API	Supported NEs
		SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
66	managedElementManager::ManagedElementManager_I:: getAllSrcFTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
67	managedElementManager::ManagedElementManager_I:: getAllDestFTPsForSNCProvisioning	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
68	managedElementManager::ManagedElementManager_I:: forceNEResync	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
69	managedElementManager::ManagedElementManager_I:: getAllTopologicalLinksForME	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454-M2 SONET, ONS 15454-M6 SONET, ASR 9000, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
70	managedElementManager::ManagedElementManager_I:: getAllRolls	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
71	managedElementManager::ManagedElementManager_I:: getAllRollNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
72	managedElementManager::ManagedElementManager_I:: getRoll	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
73	managedElementManager::ManagedElementManager_I:: getRollToPTPs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
74	managedElementManager::ManagedElementManager_I:: getRollToCTPs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
75	managedElementManager::ManagedElementManager_I:: getRollFromCTPs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
76	managedElementManager::ManagedElementManager_I:: setPortName	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
77	managedElementManager::ManagedElementManager_I:: getPortName	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH,

API No.	GateWay/CORBA API	Supported NEs
		ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
78	managedElementManager::ManagedElementManager_I:: getAllClearedAlarms	ONS 15305, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
79	managedElementManager::ManagedElementManager_I:: getAllCrossConnections	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
80	managedElementManager:: getAllSoftwareDownloadJobs	—
81	managedElementManager::ManagedElementMgr_I:: setMEAdminState	All
82	managedElementManager::ManagedElementManager_I:: getAllSrcCTPsForVCATMemberSNCProvisioning	ONS 15454 SDH, ONS 15454 SONET
83	managedElementManager::ManagedElementManager_I:: getAllDestCTPsForVCATMemberSNCProvisioning	ONS 15454 SDH, ONS 15454 SONET
84	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllEdgePoints	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 , ONS NCS2K SDH, ONS NCS2K

API No.	GateWay/CORBA API	Supported NEs
		SONET SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH
85	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllEdgePointNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
86	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllManagedElement	All
87	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllManagedElementNames	All
88	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllSubnetworkConnections	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
89	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllSubnetworkConnectionsWithTP	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
90	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllSubnetworkConnectionNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
91	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllSubnetworkConnectionNamesWithTP	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
92	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllTopologicalLinks	ONS 15216, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
93	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllTopologicalLinkNames	ONS 15216, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
94	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getMultiLayerSubnetwork	All
95	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getCTPsAndTopologicalLinks	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
96	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getSNC	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
97	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getSNCsByUserLabel	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, , ONS NCS2K SDH, ONS NCS2K SONET, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200

API No.	GateWay/CORBA API	Supported NEs
		SDH, CPT 600, CPT 600 SDH
98	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getTopologicalLink	ONS 15216, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
99	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createAndActivateSNCFromUserLabel	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
100	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createAndActivateSNC	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
101	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createAndActivateProtectedOchccFromUserLabel	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
102	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: addDrops	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
103	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deactivateAndDeleteSNC	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
104	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getVCATSNC	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
105	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVCATSNCs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
106	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVCATSNCNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K

API No.	GateWay/CORBA API	Supported NEs
		SONET
107	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVCATMemberSNCs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
108	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVCATMemberSNCNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
109	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createVCAT	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
110	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteVCAT	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
111	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: addMemberSNCsToVCAT	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
112	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteMemberSNCsFromVCAT	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
113	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteAllMemberSNCFromVCAT	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
114	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getValidSubnetworkConnectionsForMerge	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA

API No.	GateWay/CORBA API	Supported NEs
		SONET, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS NCS2K SDH, ONS NCS2K SONET
115	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: upgrade	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET , ONS NCS2K SDH, ONS NCS2K SONET
116	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createRoll	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
117	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteRoll	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
118	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: cancelRoll	Release 6.0 of Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
119	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: forceValidSignal	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
120	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: completeRoll	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
121	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: finishRoll	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
122	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllRolls	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
123	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllRollNames	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
124	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAssociatedRollNamesForSNC	Release 6.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
125	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAssociatedSNCNamesForRoll	Release 5.0 of ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
126	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createAndActivateSNCOnDRI	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15600 SDH, ONS 15600 SONET, ONS NCS2K SDH, ONS NCS2K SONET
127	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: addManagedElement	All
128	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteManagedElement	All
129	nmsSession::NmsSession_I::historyPMDataCompleted	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
130	nmsSession::NmsSession_I::historyPMDataFailed	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
131	nmsSession::NmsSession_I:: alarmLossOccurred	—
132	performance::PerformanceManagementMgr_I:: disablePMData	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH,

API No.	GateWay/CORBA API	Supported NEs
		ONS NCS2K SDH, ONS NCS2K SONET
133	performance::PerformanceManagementMgr_I:: enablePMData	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
134	performance::PerformanceManagementMgr_I:: setTCATPParameter	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
135	performance::PerformanceManagementMgr_I:: getTCATPParameter	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
136	performance::PerformanceManagementMgr_I:: getCTMHistoryPMData	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2

API No.	GateWay/CORBA API	Supported NEs
		SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
137	performance::PerformanceManagementMgr_I:: deleteHistoryPMDataFile	ONS 15310 CL, ONS 15327, ONS 15454 SDH, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
138	performance::PerformanceManagementMgr_I:: getTPHistoryPMData	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
139	performance::PerformanceManagementMgr_I:: getHoldingTime	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET

API No.	GateWay/CORBA API	Supported NEs
140	performance::PerformanceManagementMgr_I:: getAllCurrentPMData	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS NCS2K SDH, ONS NCS2K SONET
141	performance::PerformanceManagementMgr_I:: getMEPMcapabilities	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
142	protection::ProtectionMgr_I::getAllProtectedTPNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
143	protection::ProtectionMgr_I::getAllProtectionGroups	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200

API No.	GateWay/CORBA API	Supported NEs
		SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
144	protection::ProtectionMgr_I::getProtectionGroup	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
145	protection::ProtectionMgr_I::performProtectionCommand	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327,ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
146	protection::ProtectionMgr_I::createProtectionGroup	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
147	protection::ProtectionMgr_I::deleteProtectionGroup	ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS

API No.	GateWay/CORBA API	Supported NEs
		15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
148	protection::ProtectionMgr_I:: getAssociatedProtectionGroup	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS 15600 SDH, ONS 15600 SONET, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SDH, ONS NCS2K SONET
149	session::Session_I::endSession	—
150	session::Session_I::ping	—
151	mtnmVersion::Version_I::getVersion	—
152	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createMLVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
153	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyMLVLANAddFrontPorts	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
154	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyMLVLANDropFrontPorts	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
155	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I::	ONS 15310 CL, ONS 15310

API No.	GateWay/CORBA API	Supported NEs
	modifyMLVLANPortState	MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
156	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyMLVLANRSTPState	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
157	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyMLVLANTPQOS	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
158	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyMLVLANCustomerIDServiceID	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET
159	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getMLVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
160	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteMLVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
161	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllMLVLANs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
162	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllMLVLANNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET

API No.	GateWay/CORBA API	Supported NEs
163	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllAssociatedSNCs	ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET,, ONS NCS2K SDH, ONS NCS2K SONET
164	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getL2Topology	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
165	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAssociatedL2TopologyForSNC	ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET , ONS NCS2K SDH, ONS NCS2K SONET
166	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAssociatedL2TopologyForMLVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET
167	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createL2Topology	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
168	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteL2Topology	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
169	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: modifyL2Topology	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET
170	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllAvailableMLEquipmentOrTPNames	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K

API No.	GateWay/CORBA API	Supported NEs
		SONET
171	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: createVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS NCS2K SDH, ONS NCS2K SONET
172	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: deleteVLAN	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS NCS2K SDH, ONS NCS2K SONET
173	multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVLANs	ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15454 SDH, ONS 15454-M2 SDH, ONS 15454-M6 SDH, ONS 15454 SONET, ONS 15454-M2 SONET, ONS 15454-M6 SONET, ONS NCS2K SDH, ONS NCS2K SONET
174	softwareAndDataManager::SoftwareAndDataMgr_I:: backupME	All
175	softwareAndDataManager::SoftwareAndDataMgr_I:: getMEBackupStatus	All
176	softwareAndDataManager::SoftwareAndDataMgr_I:: abortMEBackup	All
177	softwareAndDataManager::SoftwareAndDataMgr_I:: getBackupList	All
178	softwareAndDataManager::SoftwareAndDataMgr_I:: activateSoftwareOnME	All
179	softwareAndDataManager::SoftwareAndDataMgr_I:: revertSoftwareOnME	All
180	softwareAndDataManager::SoftwareAndDataMgr_I:: restoreME	All
181	softwareAndDataManager::SoftwareAndDataMgr_I:: getMERestoreStatus	All
182	softwareAndDataManager::SoftwareAndDataMgr_I:: abortMERestore	All
183	softwareAndDataManager::SoftwareAndDataMgr_I:: getAllSupportedMESoftwareVersions	All

API No.	GateWay/CORBA API	Supported NEs
184	softwareAndDataManager::SoftwareAndDataMgr_I:: downloadMESoftware	All
185	softwareAndDataManager::SoftwareAndDataMgr_I:: addSupportedMESoftwareVersion	All
186	softwareAndDataManager::SoftwareAndDataMgr_I:: getSWDownloadStatus	All
187	performance::PerformanceManagementMgr_I:: getCTMValidHistoryPMData	All

3.2 common::Common_I

The Common_I interface is a set of services and utilities that each manager interface inherits. Prime Optical supports only one method from this interface.

3.2.1 common::Common_I:: getCapabilities

Synopsis

```
void getCapabilities(out CapabilityList_T capabilities)
     raises(globaldefs::ProcessingFailureException);
```

Description

This interface retrieves the capabilities of the manager. All nonspecified capabilities are assumed to be unsupported.

Parameters

Name	Type	Input/Output	Description
capabilities	CapabilityList_T	Out	List of capabilities of this manager object. It is a name-value pair, where the name represents the feature or capability and the value represents the support or nonsupport of the specified feature or capability.

A Capability_T value is used to identify a functionality supported by the EMS across the network management layer-element management layer (NML-EML) interface. It is a name/value pair, in which the name represents the feature/capability name and the value represents the support or nonsupport of the specified feature/capability.

The EMS capabilities for this release include individual IDL operation support. The feature/capability name part is used to identify an IDL operation using the following convention:

module_name::interface_name::operation_name

The currently defined values are:

- Supported—The specified feature/capability is fully or partially supported across the NML-EML interface; an operation may be partially supported if not all values of the parameters are supported.
- Unsupported—The specified feature/capability is not supported across the NML-EML interface.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
```

Compliance

TMF-defined.

3.3 emsMgr::EMSMgr_I

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

```
interface EMSMgr_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.

3.3.1 TMF 3.0 Alarm Data Structures and Parameters

Prime Optical supports the following data structures:

- 3.3.2 AlarmAndTCAIDList_T
- 3.3.3 TCAId_T
- 3.3.4 AlarmOrTCAIdentifier_T
- 3.3.5 AcknowledgeAlarms
- 3.3.6 unacknowledgeAlarms
- 3.3.7 getAllEMSAndMEUnacknowledgedActiveAlarms
- 3.3.8 getAllEMSSystemUnacknowledgedActiveAlarms
- 3.3.9 setAdditionalInfo

3.3.2 AlarmAndTCAIDList_T

Synopsis

```
typedef sequence<AlarmOrTCAIdentifier_T>  AlarmAndTCAIDList_T;
```

Description

AlarmAndTCAIDList_T is a sequence of identifiers for alarms and TCAs.

Compliance

TMF-defined.

3.3.3 TCAId_T

Synopsis

```
struct TCAId_T
    {
     globaldefs::NamingAttributes_T objectName;
     transmissionParameters::LayerRate_T layerRate;
     performance::PMParameterName_T pmParameterName;
     performance::PMLocation_T pmLocation;
     performance::Granularity_T granularity;
    };
```

Parameters

Name	Type	Input/ Output	Description
objectName	globaldefs::NamingAttributes_T	In	Identifies the alarm that must be an AID. This parameter is used as a key to search the alarm, so it must exist and must be consistent.
additionalInfo	globaldefs::NVSList_T	In	List of additional information for the alarms. This parameter must not be null or empty.
failedAcknowledgeIDList	notifications::AlarmAndTCAIDList_T	Out	List of failed acknowledged alarms, if available.

3.3.4 AlarmOrTCAIdentifier_T

Synopsis

```
union AlarmOrTCAIdentifier_T switch (AlarmTypeQualifier_T)
    {
     case ALARM:  AlarmId_T alarmId;
     case TCA:    TCAId_T tcaId;
    };
```

Description

AlarmOrTCAIdentifier_T is used as a unique identifier of a threshold crossing alert.

Compliance

TMF-defined.

3.3.5 acknowledgeAlarms

Synopsis

```
void acknowledgeAlarms(
        in notifications::AlarmAndTCAIDList_T acknowledgeIDList,
        in globaldefs::NVSList_T additionalInfo,
        out notifications::AlarmAndTCAIDList_T failedAcknowledgeIDList)
       raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows an NMS to acknowledge all of the alarms that are under the control of this EMS.
This operation is restricted to allow only 100 alarms per call.

Parameters

Name	Type	Input/ Output	Description
acknowledgeIDList	notifications::AlarmAndTCAIDList_T	In	List of target alarms. Each alarm must be identified with its AID.
additionalInfo	globaldefs::NVSList_T	In	List of additional information for the alarms. This parameter can be null or empty if the alarms to acknowledge are non-GMPLS

Name	Type	Input/ Output	Description
			alarms. The list can assume the value 'gmpls', if GMPLS alarms are acknowledged. ✎ GMPLS non-GMPLS alarms cannot be acknowledged together.
failedAcknowledgeIDList	notifications::AlarmAndTCAIDList_T	Out	List of failed acknowledged alarms, if available.

3.3.6 unacknowledgeAlarms

Synopsis

```
void unacknowledgeAlarms(
            in notifications::AlarmAndTCAIDList_T unacknowledgeIDList,
            in globaldefs::NVSList_T additionalInfo,
            out notifications::AlarmAndTCAIDList_T failedunAcknowledgeIDList)
           raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows you to acknowledge all of the alarms that are under the control of this EMS. This operation is restricted to allow only 100 alarms per call.

Parameters

Name	Type	Input/ Output	Description
acknowledgeIDList	notifications::AlarmAndTCAIDList_T	In	List of target alarms. Each alarm must be identified with its AID.
additionalInfo	globaldefs::NVSList_T	In	List of additional information for the alarms. This parameter must not be null or empty.
failedAcknowledgeIDList	notifications::AlarmAndTCAIDList_T	Out	List of failed acknowledged alarms, if available.

Throws

globaldefs::ProcessingFailureException:

EXCPT_INTERNAL_ERROR - Raised in case of non-specific EMS internal failure.

EXCPT_UNABLE_TO_COMPLY - Raised when this method is being used by another OSS client.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators have reached 100.

To address EMS alarms, AID must be used without ManagedElement:

objectName [0] = new NameAndStringValue_T("EMS", "Cisco Systems/Prime Optical")

```
objectName [1] = new NameAndStringValue_T("AID", "notificationId=10");
The EMS generates an AVC event each time the API is called.
```

Compliance
TMF-defined.

3.3.7 getAllEMSAndMEUnacknowledgedActiveAlarms

Synopsis

```
void getAllEMSAndMEUnacknowledgedActiveAlarms(
        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
getAllEMSAndMEUnacknowledgedActiveAlarms returns all EMS and ManagetElements Alarms that are active and unacknowledged.

Parameters

Name	Type	Input/ Output	Description
excludeProbCauseList	notifications::ProbableCauseList_T	In	List of probable causes to exclude from the the output event list.
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.

3.3.8 getAllEMSSystemUnacknowledgedActiveAlarms

Synopsis

```
void getAllEMSSystemUnacknowledgedActiveAlarms(
        in notifications::PerceivedSeverityList_T excludeSeverityList,
        in unsigned long how_many,
        out notifications::EventList_T eventList,
        out notifications::EventIterator_I eventIt)
       raises(globaldefs::ProcessingFailureException);
```

Description
getAllEMSSystemUnacknowledgedActiveAlarms returns all EMS System Alarms that are active and unacknowledged.

Parameters

Name	Type	Input/ Output	Description
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.

3.3.9 setAdditionalInfo

Synopsis

```
void setAdditionalInfo(
        in globaldefs::NamingAttributes_T objectName,
        inout globaldefs::NVSList_T additionalInfo)
       raises (globaldefs::ProcessingFailureException);
```

Description

This interface allows you to change the note of an alarm. You can use setAdditionalInfo to perform an append, replace, or delete operation for an alarm. This is applicable for alarms at the EMS/NE level.

Name	Type	Input/ Output	Description
objectName	globaldefs::NamingAttributes_T	In	Alarm AID that is the target for note changes.
additionalInfo	globaldefs::NVSList_T	In	List of additional information for the alarm. This parameter must not be null or empty.

additionalInfo must contain one of the parameters in the following table, where Key and Value are the NameAndStringValue_T fields:

Key	Value	Description
note_append	String	Note to append to an existing note.
note_replace	String	Note to replace.
note_delete	String	Note to delete.

To address EMS alarms, AID must be used without ManagedElement:

- objectName [0] = new NameAndStringValue_T("EMS", "Cisco Systems/Prime Optical")
- objectName [1] = new NameAndStringValue_T("AID", "notificationId=10");

The EMS generates an AVC event each time this API is called.

3.3.10 getAllEMSAndMEActiveAlarms

Synopsis

```
void getAllEMSAndMEActiveAlarms(
        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 interface enables an NMS to request active alarms that are under the control of the EMS (alarms raised by the NEs and alarms raised by the EMS itself). Some alarms might be filtered out (excluded) by specifying their probable causes or severities.
Due to performance concerns, only one OSS client is allowed to invoke this operation at a time. Otherwise, Prime Optical throws the EXCPT_UNABLE_TO_COMPLY exception.
The result of this operation is independent of the filtering set up by the NMS for the notification service.
ONS 15310, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET, ONS 15600 SONET, and ONS 15600 SDH

ONS 15310, ONS 15327, ONS 15454 SDH, ONS 15454 SONET, ONS NCS2K SDH, ONS NCS2K SONET**,** ONS 15600 SONET, and ONS 15600 SDH do not send a "clear" for TCA events. TCAs are not reported in the eventList returned by this method.

Name	Type	Input/ Output	Description
excludeProbCauseList	notifications::ProbableCauseList_T	In	List of probable causes to exclude (for which events are not 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 - Raised in case of nonspecific EMS internal failure.

EXCPT_UNABLE_TO_COMPLY - Raised when this method is being used by another OSS client.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.3.11 getAllEMSSystemActiveAlarms

Synopsis

```
void getAllEMSSystemActiveAlarms(
        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 interface enables an NMS to request all active alarms for the EMS. Some alarms can be filtered out (excluded) by specifying their 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
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 - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.3.12 getAllTopLevelSubnetworks

Synopsis

```
void getAllTopLevelSubnetworks(
        in unsigned long how_many,
        out multiLayerSubnetwork::SubnetworkList_T sList,
        out multiLayerSubnetwork::SubnetworkIterator_I sIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request all subnetworks that are under the control of the EMS.

Parameters

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Maximum number of subnetworks to return in the first batch.
sList	multiLayerSubnetwork::SubnetworkList_T	Out	First batch of subnetworks.
sIt	multiLayerSubnetwork::SubnetworkIterator_I	Out	Iterator to retrieve the remaining subnetworks.

Prime Optical support for MultiLayerSubnetwork_T is as follows:

- globaldefs::NamingAttributes_T name

  - Supported. This field contains two tuples. The first tuple identifies the EMS. The second tuple is the same as nativeEMSName. The naming convention is described in 2.2.5.1 Multilayer Subnetwork.
- string userLabel
  - Supported.
- string nativeEMSName
  - Supported.
- string owner
  - Not supported.
- Topology_T subnetworkType
  - Supports TOPO_SINGLETON, TOPO_CHAIN, TOPO_PSR, TOPO_OPEN_PSR, TOPO_SPRING, TOPO_OPEN_SPRING, and TOPO_MESH.
- transmissionParameters::LayerRateList_T supportedRates
  - Not supported.

- globaldefs::NVSList_T additionalInfo
  - Supported.

For the Prime Optical client subnetwork types, the following mapping applies:

CORBA Subnetwork Type	Prime Optical Client Subnetwork Type
TOPO_SINGLETON	SINGLETON
TOPO_CHAIN	LINEAR
TOPO_PSR	UPSR
TOPO_PSR	SNC
TOPO_OPEN_PSR	OPEN_UPSR
TOPO_OPEN_PSR	OPEN_SNC
TOPO_SPRING	BLSR
TOPO_SPRING	MSSP
TOPO_OPEN_SPRING	OTHER
TOPO_MESH	MESH
TOPO_MESH	UNKNOWN
TOPO_MESH	DEFAULT

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.3.13 getAllTopLevelSubnetworkNames

Synopsis

```
void getAllTopLevelSubnetworkNames(
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
```

```
raises(globaldefs::ProcessingFailureException);
```

Description

This interface has the same behavior as the interface described in 3.3.12 getAllTopLevelSubnetworks, but instead of returning entire object structures, this interface returns their names.

Parameters

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Maximum number of subnetworks to return in the first batch.
sList	globaldefs::NamingAttributesList_T	Out	First batch of subnetwork names.
sIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining subnetwork names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.3.14 getAllTopLevelTopologicalLinks

Synopsis

```
void getAllTopLevelTopologicalLinks(
        in unsigned long how_many,
        out topologicalLink::TopologicalLinkList_T topoList,
        out topologicalLink::TopologicalLinkIterator_I topoIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request all of the top-level topological links that are under the control of the EMS. A top-level topological link is a link in which the link endpoints exist in two different multilayer subnetworks.

If a topological link that is under the control of the EMS is removed from the NE, the provType parameter is reported as INVALID.

Parameters

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Maximum number of top-level topological links to return in the first batch.
topoList	topologicalLink::TopologicalLinkList_T	Out	First batch of top-level topological links.
topoIt	topologicalLink::TopologicalLinkIterator_I	Out	Iterator to retrieve the remaining top-level topological links.

Prime Optical support for TopologicalLink_T structure is as follows:

- globaldefs::NamingAttributes_T name

  - Supported. This field has two tuples. The first tuple is for the EMS. The second is for the topological link. The value in the second tuple is the same as the nativeEMSName field except for the Y-cable links, which have "::1" and "::2" appended to the nativeEMSName to identify the two legs of the Y-cable link.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. For more information, see 2.2.5 Naming Conventions.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction. Prime Optical supports CD_BI for bidirectional links and CD_UNI for unidirectional links.
- transmissionParameters::LayerRate_T rate. Supported. For a list of possible values, see 2.2.1 Layer Rate.
- globaldefs::NamingAttributes_T aEndTP
  - Supported.
- globaldefs::NamingAttributes_T zEndTP
  - Supported.
- globaldefs::NVSList_T additionalInfo
  - Supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support topological links	provType	AUTODISCOVERED MANUAL UNMANAGED PATCHCORD SERVER_TRAIL
All managed elements that support topological links	linkStatus	LINK_VALID LINK_INVALID
All managed elements that support topological links	linkProtectionTyp e	PROT_ONE_PLUS_ONE PROT_BLSR_2F PROT_BLSR_4F PROT_BLSR_2F_PCA PROT_BLSR_4F_PCA PROT_PCA PROT_TUNNEL PROT_UNPROTECTED PROT_UNKNOWN PROT_NOT_APPLICABLE PROT_PARTIALLY_PROTECTE D PROT_YCABLE
All managed elements that support topological links	linkCost	Integer, 0 to 1024

Throws

`globaldefs::ProcessingFailureException`

`EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.`

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.3.15 getAllTopLevelTopologicalLinkNames

Synopsis

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

Description
This interface has the same behavior as the interface described in 3.3.14 getAllTopLevelTopologicalLinks, but instead of returning entire object structures, this interface returns their names.

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Maximum number of top-level topological links to return in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of top-level topological link names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining top-level topological link names.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.3.16 getTopLevelTopologicalLink

Synopsis

```
void getTopLevelTopologicalLink(
        in globaldefs::NamingAttributes_T topoLinkName,
        out topologicalLink::TopologicalLink_T topoLink)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables an NMS to request the top-level topological link if its name is under the control of the EMS.

If a link that is under the control of the EMS is removed from the NE, the ProvType parameter is reported as INVALID.

Parameters

Name	Type	Input/Output	Description
topoLinkName	globaldefs::NamingAttributes_T	In	Name of the topological link.
topoLink	TopologicalLink::TopologicalLink_T	Out	Top-level topological link.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when topoLinkName does not reference a top level topological link object.

EXCPT_ENTITY_NOT_FOUND - Raised when topoLinkName references a top level topological link object that does not exist in the EMS.

Compliance

TMF-defined.

3.3.17 createTopologicalLink

Synopsis

```
void createTopologicalLink(
        in topologicalLink::TLCreateData_T createData,
        out topologicalLink::TopologicalLink_T topologicalLink)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to create a topological link manually between two MEs. Prime Optical supports topological link creation between MEs belonging to the same multilayer subnetwork. If Prime Optical autodiscovers a topological link between two endpoints on which a manually created link exists, Prime Optical overwrites the manually created topological link information with the autodiscovered topological link information. Prime Optical generates an attribute value change event when a manually created link is overwritten with autodiscovered topological link information. Prime Optical supports topological link creation between two unmanaged NEs, and also between a managed NE and an unmanaged NE. Prime Optical ignores the topological link layer rate if the topological link creation is between two unmanaged NEs.

Prime Optical generates an object creation event notification when a new topological link is created.

✐ Prime Optical creates topological links between two PTPs even if there are no existing physical link connections in the network. To enable SDCC on the topological link, the NMS must ensure the presence of physical link connectivity.

Parameters

Name	Type	Input/Output	Description
createData	topologicalLink::TLCreateData_T	In	Data structure containing details for creating new topological links.
topologicalLink	topologicalLink::TopologicalLink_T	Out	Created topological link information.

Prime Optical support for TLCreateData_T structure is as follows:

- string userLabel
  - Supported. The value of this field is assigned to the name of the newly created topological link. For patchcord links, Prime Optical assigns the link name. The user label that is passed as input is ignored for patchcord links.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of topological link names.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Prime Optical supports CD_BI for bidirectional links.
  - Prime Optical supports CD_UNI for unidirectional links.
- transmissionParameters::LayerRate_T rate. For a list of supported layer rates, see 2.2.1 Layer Rate.
- globaldefs::NamingAttributes_T aEndTP
  - Supported.
- globaldefs::NamingAttributes_T zEndTP
  - Supported.
- globaldefs::NVSList_T additionalCreationInfo
  - Supported. The supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support topological links	linkProtectionType	PROT_ONE_PLUS_ONE.
All managed elements that support topological links	linkCost	Integer, 0 to 999999.
All managed elements that support topological links	provType	MANUAL (default value) and PATCHCORD.
All managed elements that support topological links	zEndSecSrc	A string that represents a sequence of namevalue pairs for the TMF physical location of a PTP. A semicolon (;) is used as separator. Format: `name='EMS';value='Cisco Systems/CTM';name='ManagedElement';value='15454-ANSI-68-8';name='PTP';value='/rack=1/shelf=1/slot=15/port=10'`
All managed elements that support topological links	aEndSecDst	A string that represents a sequence of namevalue pairs for the TMF physical location of a PTP. A semicolon (;) is used as separator.

- globaldefs::sSrlgGroupList_T srlgGroupListValue
  - Supported. This field holds the shared risk link group (SRLG) group values. This is an optional parameter. You can include the SRLG group during link creation. A maximum of five SRLG groups are supported for a given link. SRLG groups are only supported for SONET/SDH topological links.

To create unidirectional OTS links without side between 80WXC cards on the ONS15454 NEs, specify the parameter zEndSecSrc as secondary source and the parameter aEndSecDst as destination TP in the additionalCreationInfo list.

```
A End PTP
        [0]    name       = EMS                        value      = Cisco
Systems/Prime Optical
        [1]    name       = ManagedElement             value      = 15454-ANSI-
68-9
        [2]    name       = PTP                        value
=/rack=1/shelf=1/slot=15/port=12


Z End PTP
        [0]    name       = EMS                        value      = Cisco
Systems/Prime Optical
        [1]    name       = ManagedElement             value      = 15454-ANSI-
68-8
        [2]    name       = PTP                        value      =
/rack=1/shelf=1/slot=13/port=13


Additional CreationInfo
        [0]    name       = linkProtectionType         value      =
PROT_UNPROTECTED
        [1]    name       = linkCost                   value      = 1024
        [2]    name       = provType                   value      = PATCHCORD
        [3]    name       = zEndSecSrc                 value      =
name='EMS';value='Cisco Systems/’Prime
Optical’;name='ManagedElement';value='15454-ANSI-68-
8';name='PTP';value='/rack=1/shelf=1/s lot=13/port=10'
        [4]    name       = aEndSecDst                 value      =
name='EMS';value='Cisco Systems/’Prime
Optical’;name='ManagedElement';value='15454-ANSI-68-
9';name='PTP';value='/rack=1/shelf=1/s lot=15/port=1'
```

In both of these examples, if the operator supplies any wrong parameters for secondary source/drop ports, the Cisco Prime Optical GateWay/CORBA interface returns an exception that reports the list of actual available ports retrieved from the ME, such as the following:

```
java.lang.String errorReason=The secondary PTP /rack=1/shelf=1/slot=15/port=10
does not belong to the list of available PTPs: /rack=1/shelf=1/slot=15/port=1,
/rack=1/shelf=1/slot=15/port=2, /rack=1/shelf=1/slot=15/port=3,
/rack=1/shelf=1/slot=15/port=4, /rack=1/shelf=1/slot=15/port=5,
/rack=1/shelf=1/slot=15/port=6, /rack=1/shelf=1/slot=15/port=7,
/rack=1/shelf=1/slot=15/port=8
```

If you do not provide a secondary source PTP, the primary link between the AendTP and the ZendTP is successfully created. However, while creating the reverse link, the GateWay/CORBA interface tries using the secondary source as the first element from the list of available PTPs returned by the ME/NCP layer (for example, /rack=1/shelf=1/slot=15/port=1). If such a PTP port is still available, the reverse link is created properly; otherwise, Prime Optical throws an exception.

It is recommended that you specify a secondary source PTP. If you do not specify the secondary source PTP, GateWay/CORBA, by default, uses the first item from the list of available PTPs. However, the element may also be invalid.

You have to specify the desired PTP in the additional creation info list in case of the availability of multiple choices.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_NOT_IMPLEMENTED - Raised if input createData is not valid.
EXCPT_INVALID_INPUT - Raised if the EMS cannot support this service.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
```

Compliance

TMF-defined.

3.3.18 createTopologicalLinkWithYCableProtection

Synopsis

```
void createTopologicalLinkWithYCableProtection(
        in topologicalLink::MTLCreateData_T createData,
        out topologicalLink::TopologicalLink_T topologicalLink)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to create a Y-cable topological link manually between two managed elements. A Y-cable link is a bidirectional link with one PTP source and two PTP destinations. Because of the limitation in TMF for topological links that have one PTP source and one PTP destination, Prime Optical reports two topological links for a Y-cable link. These two topological links have the same nativeEMSName. See 2.2.5.10 Topological Links for the naming convention for the Y-cable topological link object name. The destination of the Y-cable link must be on a DWDM card on an ONS 15454 SONET/SDH or ONS NCS2K SONET/SDH.

Prime Optical generates two object creation event notifications when a new Y-cable topological link is created.

The topological link returned by this method is one leg of the Y-cable.

Parameters

Name	Type	Input/ Output	Description
createData	topologicalLink:MTLCreateData_T	In	Data structure containing details for creating a new Y-cable topological link.
topologicalLink	topologicalLink::TopologicalLink_T	Out	One leg of the created Y-cable topological link information.

Prime Optical support for MTLCreateData_T structure is as follows:

- string userLabel
  - Supported. The value of this field is assigned to the nativeEMSName of the newly created topological link.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of topological link names.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Supports only CD_BI for bidirectional links.
- transmissionParameters::LayerRate_T rate. Supported layer rates are:

  - LR_Physical_2_5_Gigabit_ITU
  - LR_Physical_10_Gigabit_ITU
  - LR_DSR_OC3_STM1
  - LR_DSR_OC12_STM4
  - LR_DSR_OC48_STM16
  - LR_DSR_OC192_STM64
- globaldefs::NamingAttributesList_T aEndTPs
  - Supported. It contains only one PTP.
- globaldefs::NamingAttributesList_T zEndTPs
  - Supported. It contains two PTPs on the same ME.

- globaldefs::NVSList_T additionalInfo
  - Supported. The supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support topological links	linkCost	Integer, 0 to 999999.
All managed elements that support topological links	provType	MANUAL (default value) and PATCHCORD.

- globaldefs::sSrlgGroupList_T srlgGroupListValue
  - Supported. This field holds the SRLG group values. This is an optional parameter. You can include the SRLG group during link creation. A maximum of five SRLG groups are supported for a given link. SRLG groups are only supported for SONET/SDH topological links.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if input createData is not valid.

EXCPT_INVALID_INPUT - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.3.19 deleteTopologicalLink

Synopsis

```
void deleteTopologicalLink(
        in globaldefs::NamingAttributes_T topoLinkName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to delete a topological link manually between two managed elements. Prime Optical throws the EXCPT_UNABLE_TO_COMPLY exception if an attempt is made to delete an autodiscovered topological link or if the topological link cannot be deleted in the current state. Prime Optical generates an object deletion event notification when a topological link is deleted.

✎ Prime Optical deletes topological links between two PTPs even if there is a physical link connection in the network. The NMS must ensure the disabling of the SDCC, the deletion of all circuits carried through the link, and the removal of the physical link connectivity.

✎ Patchcord links are deleted twice. The first delete operation forces the link to become invalid, but the link information is still present in the Prime Optical database. The second delete operation removes the link information from the Prime Optical database.

Parameters

Name	Type	Input/Output	Description
topoLinkName	globaldefs::NamingAttributes_T	In	Name of the topological link to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised if input topological link name is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised if the specified topological link name does not exist in Prime Optical.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_UNABLE_TO_COMPLY - Raised if an attempt is made to delete an auto-discovered topological link.

Compliance

Prime Optical-defined. This interface is already proposed in TMF and will be included TMF Release 3.0.

3.3.20 modifyTopologicalLink

Synopsis

```
void modifyTopologicalLink(
                    in topologicalLink::TLCreateData_T    newTLCreateData,
                    out topologicalLink::TopologicalLink_T newTopologicalLink )
                raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to modify the following parameters of existing topological links:

- Cost attribute (modifiable only for SONET and SDH links)
- SRLG values (modifiable only for SONET and SDH links)

✎ The modifyTopologicalLink API is not supported for links with patchcord provisioning. Also, you cannot use the modifyTopologicalLink API to modify the link description on any supported NEs.

Parameters

Name	Type	Input/ Output	Description
createData	topologicalLink::TLCreateData_T	In	Data structure containing details for creating a new topological link.
topologicalLink	topologicalLink::TopologicalLink_T	Out	Created topological link information.

Prime Optical support for TLCreateData_T structure is as follows:

- string userLabel
  - Supported. The value of this field is assigned to the name of the newly created topological link.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of topological link names.
- string owner

  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Prime Optical supports CD_BI for bidirectional links and CD_UNI for unidirectional links.
- transmissionParameters::LayerRate_T rate. For a list of supported layer rates, see the layer rates supported in 3.3.17 createTopologicalLink.
- globaldefs::NamingAttributes_T aEndTP
  - Supported.
- globaldefs::NamingAttributes_T zEndTP
  - Supported.
- globaldefs::NVSList_T additionalCreationInfo
  - Supported. The supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support topological links	linkProtectionType	PROT_ONE_PLUS_ONE.
All managed elements that support topological links	linkCost	Integer, 0 to 999999.

- globaldefs::sSrlgGroupList_T srlgGroupListValue
  - Supported. This field holds the SRLG group values. This is an optional parameter. You can include the SRLG group during link creation. A maximum of five SRLG groups are supported for a given link. SRLG groups are supported only for SONET/SDH topological links.

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised if input createData is not valid.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
```

Compliance

Prime Optical-defined.

3.3.21 getEMS

Synopsis

```
void getEMS(out EMS_T emsInfo)
     raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request EMS information.

Parameters

Name	Type	Input/Output	Description
emsInfo	EMS_T	Out	EMS information.

Prime Optical supports EMS_T is as follows:

- globaldefs::NamingAttributes_T name
  - EMS ID is returned as the same name that is displayed on the Prime Optical client EMS property sheet.
- string userLabel
  - Not supported.
- string nativeEMSName

  - Prime Optical returns the EMS ID that is displayed on the Prime Optical client EMS property sheet.
- string owner
  - Not supported.
- string emsVersion
  - Prime Optical returns "10.7."
- string type
  - Prime Optical returns "Cisco Systems/PrimeOptical."
- globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
```

Compliance

TMF-defined.

3.3.22 getAllL2Topologies

Synopsis

```
void getAllL2Topologies(in multiLayerSubnetwork::L2TopoType_T l2TopoType,
        in unsigned long how_many,
        out multiLayerSubnetwork::L2TopologyList_T l2TopoList,
        out multiLayerSubnetwork::L2TopologyIterator_I l2TopoIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface lists all L2 topologies of a particular topology type in the EMS. The L2 topology has a scope in the EMS. Prime Optical supports only Resilient Packet Ring (RPR) and point-to-point L2 topology types.

Parameters

Name	Type	Input/ Output	Description
l2TopoType	MultiLayerSubnetwork::L2TopoType_T	In	Type of L2 topologies to return.
how_many	unsigned long	In	Maximum number of L2 topologies to return in the first batch.
l2TopoList	multiLayerSubnetwork::L2TopologyList_T	Out	List of available L2 topologies in this multilayer subnetwork.
l2TopoIt	multiLayerSubnetwork::L2TopologyIterator_I	Out	Iterator used to retrieve the remaining L2 topologies.

Prime Optical support for L2Topology_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported. The field contains two tuples. The first tuple contains a value for EMS. The second tuple contains the L2 topology name.
- string userLabel
  - Not supported.
- string nativeEMSName

  - Supported. The L2 topology name is reported.
- string owner
  - Not supported.
- L2TopoLayerRate_T layerRate
  - Supported. This represents the total capacity of the L2 topology. L2TopoLayerRate_T contains the following information:
  - LayerRate_T layerRate—Represents the layer rate of the SNCs. For VCAT, represents the layer rate of the member SNCs.
  - Unsigned long numberOfVcatMembers—Represents the number of members in each SNC of an L2 topology consisting of all VCAT SNCs. In all other cases, it contains a value less than one and should be ignored.
  - Supported layer rates are:
  - LR_STS1_and_AU3_High_Order_VC3
  - LR_STS3c_and_AU4_VC4
  - LR_STS6c_and_VC4_2c
  - LR_STS9c_and_VC4_3c
  - LR_STS12c_and_VC4_4c
  - LR_STS24c_and_VC4_8c
  - LR_Not_Applicable (if the L2 topology contains L1 SNCs with different layer rates or if the L2 topology contains a mix of CCAT and VCAT L1 SNCs)
- multiLayerSubnetwork::L2TopoType l2TopoType
  - The L2 topology type. Valid values are L2TopoType.PT2PT, L2TopoType.RPR, and L2TopoType.DOT17_RPR.
- multiLayerSubnetwork::L2TopoState topoState
  - GateWay/CORBA handles the L2 topology states L2TopoState.COMPLETE and L2TopoState.INCOMPLETE.

✐ The L2TopoState API does not take into consideration the L2 protection information (WRAPPING or STEERING). Consequently, the COMPLETE_WRAPPING and COMPLETE_STEERING states are remapped into COMPLETE. Likewise, the INCOMPLETE_WRAPPING and INCOMPLETE_STEERING states are remapped into INCOMPLETE.

- multiLayerSubnetwork::L2TopoOperState
  - The operational state of the L2 topology. Valid values are L2TopoOperState.IS, L2TopoOperState.ISPARTIAL, and L2TopoOperState.OOS.
- multiLayerSubnetwork::L2TopoSyncState topoSyncState
  - The current L2 topology synchronization state. Valid values for L2TopoSyncState are:
  - L2TopoSyncState. NEEDED
  - L2TopoSyncState. INPROGRESS
  - L2TopoSyncState. COMPLETE
  - L2TopoSyncState. PARTIALLY_COMPLETE
  - L2TopoSyncState. FAILED
  - L2TopoSyncState. L2SERVICENOTREADY

- globaldefs::NVSList_T additionalInfo
  - Supported.
  - If the topology type is DOT17_RPR, the bandwidth parameter values return the absolute bandwidth reserved for the class of traffic. The return value is similar to 250 (Class A). The number 250 indicates 250 Mbps bandwidth reserved for Class A-type traffic. Multicast

bandwidth parameters are not supported for this type of topology. If no values are defined for class types, the following default values are returned:

- ClassSpMgmt: C
- ClassCommitted: C
- ClassBestEffort: C
- ClassAvvidControl: B
- ClassAvvidVoiceVideo: A
- ClassAReservedBandwidth: 0
- ClassA1Bandwidth: 10
- ClassBCommittedBandwidth: 10
- SpMgmtCoS: 6, 7
- CommittedCoS: 1, 2
- AvvidVoiceVideoCoS: 5
- AvvidControlCoS: 3
- CosCommit: 2

- The bandwidth parameters SpMgmtBandwidth, CommittedBandwidth, BestEffortBandwidth, AvvidControlBandwidth, SpMgmtCoS, CommittedCoS, AvvidVoiceVideoCoS, AvvidControlCoS, and the CoS value CosCommit are supported. Either all or none of these parameters are passed. Each of the bandwidth parameters (SpMgmtBandwidth, CommittedBandwidth, BestEffortBandwidth, and AvvidControlBandwidth) is from 1 to 96; the sum of all four bandwidth parameters is from 1 to 99. The CoS commit value is from 0 to 7. If none of these parameters is passed, Prime Optical takes the following default values:
  - SpMgmtBandwidth: 5
  - CommittedBandwidth: 80
  - BestEffortBandwidth: 10
  - AvvidControlBandwidth: 4
  - SpMgmtCoS: 6, 7
  - CommittedCoS: 1, 2
  - AvvidVoiceVideoCoS: 5
  - AvvidControlCoS: 3
  - CosCommit: 2

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised when an l2TopoType is invalid.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

Prime Optical-defined.

3.3.23 getAllQoSTemplateNames

Synopsis

```
void getAllQoSTemplateNames( in unsigned long how_many,
        out globaldefs::NamingAttributesList_T qosTempNameList,
        out globaldefs::NamingAttributesIterator_I qosTempNameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface lists all QoS template names in the EMS. The QoSTemplate has a scope in the EMS.

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Maximum number of qosTempNames to return in the first batch.
qosTempNameList	globaldefs::NamingAttributesList_T	Out	First batch of names.
qosTempNameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator used to retrieve the remaining names.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance
Prime Optical-defined.

3.3.24 getAllQoSTemplates

Synopsis

```
void getAllQoSTemplates
        in unsigned long how_many,
        out QoSTemplateList_T qosTempList
        out QoSTemplateIterator_I qosTempIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface lists all QoS templates in the EMS. The QoSTemplate has a scope in the EMS.
Parameters

Name	Type	Input/Output	Description
how_many	unsigned long	In	Maximum number of qosTempNames to return in the first batch.
qosTempList	QoSTemplateList_T	Out	First batch of QoS templates.
qosTempIt	QoSTemplateIterator_I	Out	Iterator used to retrieve the remaining QoS templates.

Prime Optical support for QoSTemplate structure is as follows:

- globaldefs::NamingAttributes_T templateName—Supported. This field specifies the template name.
- string qos_template_description—Supported. This field specifies the description of the template.
- emsMgr::TemplateType_T—TemplateType specifies the type of template. Prime Optical supports the following types:
  - TemplateType_T.BEST_EFFORT
  - TemplateType_T.CIR_PIR
  - TemplateType_T.ADVANCED
- QOSPolicy_T policy—Supported. This field specifies the QoS policy for the template.
- QOSPolicy_T is defined as follows:

  - string policy_name—Supported. This field specifies the policy name for the QoS.
  - QoSClassList_T qosClass—Supported. This field provides a list of QoS classes for the policy. Each QoSClass_T has the following members:
    - multiLayerSubnetwork::MatchProtocolList_T classification—Supported. Each MatchProtocol_T is defined as:
    - Classification_T matchTraffic—Specifies the traffic classification. Prime Optical supports the following classifications:
      Classification_T.MATCH_ANY
      Classification_T.MATCH_DSCP
      Classification_T.MATCH_IP
      Classification_T.MATCH_COS
- string value—This field is a string value. Multiple values are separated by spaces.
- string class_name—Supported. This field specifies the name of the QoS class.
- QoSPolicer_T policer—Supported. This field specifies the policer to use.
- QoSPolicer_T structure has the following members:
  - multiLayerSubnetwork::CIRType_T cir_type—Supported. This field specifies the type of CIR (line rate or rate limited). Prime Optical defines the following values:
    - CIRType_T.CIR_RATE
    - CIRType_T.CIR_LINE
    - long cir—Supported. This field specifies the CIR value in the range of 96 to 800,000 kilobits per second.
    - long cir_burst—Supported. This field specifies the CIR burst value in the range of 8000 to 64000 bytes.
  - multiLayerSubnetwork::COSMarking_T cir_cos—Supported. Specifies the type of CoS marking for the CIR. Prime Optical supports the following values:
    - COSMarking_T.COSMARKING_MARK_COS
    - COSMarking_T.COSMARKING_TRUST
    - long cir_cos_val—Supported. This field specifies the CIR CoS value.
  - multiLayerSubnetwork::ExcessAction_T excess_act—Supported. Prime Optical supports the following values:
    - ExcessAction_T.EXCESS_ACTION_DISCARD
    - ExcessAction_T.EXCESS_ACTION_ALLOW
  - long pir—Supported. This field specifies the PIR value in the range of 96 to 800,000 kilobits per second. The PIR value is equal to or greater than the CIR value.
  - long pir_burst—Supported. This field specifies the PIR burst value in the range of 8000 to 64000 bytes. The PIR burst value is equal to or greater than the CIR burst value.
  - multiLayerSubnetwork::COSMarking_T pir_cos—Supported. This field specifies the type of CoS marking for PIR. Prime Optical supports the following values:
    - COSMarking_T.COSMARKING_MARK_COS
    - COSMarking_T.COSMARKING_TRUST
  - long pir_cos_value—Supported. This field specifies the PIR COS value.
  - multiLayerSubnetwork::ViolateAction_T violate_act—Supported. Prime Optical supports the following values:
    - ViolateAction_T.VIOLATE_ACTION_DISCARD
    - ViolateAction_T.VIOLATE_ACTION_ALLOW
  - long violate_cos_val—Supported. This field specifies the violated COS value.
  - multiLayerSubnetwork::BesteffortType_T beff_type—Supported. Prime Optical supports the following values for best effort type:

  - BestEffortType_T.BEST_EFFORT_RATE
  - BestEffortType_T.BEST_EFFORT_LINE
- long beff_rate—Supported. This field specifies the maximum rate of best effort in the range of 96 to 800,000 kilobits per second.
- long beff_rate_burst—Supported. This field is in the range of 8000 to 64000 bytes.
- multiLayerSubnetwork::MatchOption_T operator—Supported. Prime Optical supports the following Match option:
  - MatchOption_T.MATCH_OPTION_OR
  - MatchOption_T.MATCH_OPTION_AND

✎ Prime Optical always uses MatchOption_T.MATCH_OPTION_AND.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

Prime Optical-defined.

3.3.25 getQoSTemplate

Synopsis

```
void getAllQoSTemplates(
        in globaldefs::NamingAttribute_T QoSTemplateName,
        out QoSTemplate_T qosTemp )
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface lists the complete QoSTemplate information for each QoSTemplate name. The QoSTemplate has a scope in the EMS.

Parameters

Name	Type	Input/Output	Description
qosTemplateName	globaldefs::NamingAttributes_T	In	QoS template name.
qosTemp	QoSTemplate_T	Out	QoS template with the complete information.

For details about the QoSTemplate_T structure, see 3.3.23 getAllQoSTemplateNames.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the QoSTemplate name is not valid.

EXCPT_ENTITY_NOT_FOUND - Raised when the QoSTemplate name does not exist on the EMS.

Compliance

Prime Optical-defined.

3.3.26 getAllEMSandMEClearedAlarms

Synopsis

```
void getAllEMSandMEClearedAlarms(
```

```
        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 interface enables an NMS to request all of the cleared and acknowledged alarms that are under the control of the EMS (both alarms raised by the NEs and alarms raised by the EMS itself). Some alarms may be filtered out (excluded) by specifying their probable causes or severities.

Due to performance concerns, this operation is restricted to enable only one OSS client to access it at one time. Otherwise, Prime Optical throws the EXCPT_UNABLE_TO_COMPLY exception.

The result of this operation is independent of the filtering set up by the NMS for the notification service.

ONS 15310, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, ONS NCS2K SDH, ONS 15600 SONET, and ONS 15600 SDH

The ONS 15310, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, ONS NCS2K SDH, ONS 15600 SONET, and ONS 15600 SDH do not send a “clear” for TCA events. TCAs are not reported in the eventList that is returned by this method.

Parameters

Name	Type	Input/Output	Description
excludeProbCauseList	notifications::ProbableCauseList_T	In	List of probable alarm causes to exclude from the output event list.
excludeSeverityList	notifications::PerceivedSeverityList_T	In	List of alarm 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	Out	First batch of alarms.
EventIt	notifications::EventIterator_I	Out	Iterator to retrieve the remaining alarms.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_UNABLE_TO_COMPLY - Raised when this method is being used by another OSS Client.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.3.27 getAllEMSSystemClearedAlarms

Synopsis

```
void getAllEMSSystemClearedAlarms(
        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 interface enables an NMS to request all of the cleared and acknowledged alarms that are for the EMS itself. Some alarms may be filtered out (excluded) by specifying their 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
excludeSeverityList	notifications::PerceivedSeverityList_T	In	List of alarm 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	Out	First batch of alarms.
EventIt	notifications::EventIterator_I	Out	Iterator to retrieve the remaining alarms.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.3.28 Passive Unit EMS Alarms

In Prime Optical 10.7, passive units are supported. See 2.2.2.1 Passive Unit Inventory for the list of passive units that are supported. For passive units, no alarms are supported.

3.3.29 createServerTrailLink

Synopsis

```
void createServerTrailLink(
        in serverTrailLink::STLCreateData_T createData
        raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to create a server trail link manually between two MEs. Prime Optical supports the server trail link creation between MEs belonging to the same multilayer subnetwork. Prime Optical generates an object creation event notification when a new server trail link is created.

- STLCreateData_T is a Prime Optical-defined data structure. The TMF proposed this structure and included it in the TMF 3.0 specification.
- Prime Optical creates the server trail link between two CTPs even if there are no existing physical link connections in the network.

Parameters

Name	Type	Input/Output	Description
createData	serverTrailLink::STLCreateData_T	In	Data structure containing details for creating a new server trail link.

Prime Optical support for STLCreateData_T structure is as follows:

- string userLabel
  Not supported. The value of this field is assigned to the name of the newly created server trail link.
- boolean forceUniqueness
  Not supported.
- string owner
  Not supported.
- globaldefs::ConnectionDirection_T direction
  Prime Optical supports only CD_BI for bidirectional links.
- transmissionParameters::LayerRate_T rate
  Supported layer rates are:
  - LR_STS1_and_AU3_High_Order_VC3
  - LR_STS3c_and_AU4_VC4
  - LR_STS12c_and_VC4_4c
  - LR_STS24c_and_VC4_8c
  - LR_STS48c_and_VC4_16c
  - LR_STS192c_and_VC4_64c
- globaldefs::NamingAttributes_T aEndCTP
  Supported.
- globaldefs::NamingAttributes_T zEndCTP
  Supported.
- globaldefs::NVSList_T additionalCreationInfo
  Supported. The supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support ServerTrailLink	serverTrailLinkProtectionType	PREEMPTIBLE, UNPROTECTED, FULLY_PROTECTED.
All managed elements that support ServerTrailLink	linkCost	Integer, 0 to 999999.
All managed elements that support ServerTrailLink	linkCount	Integer, 1 to 1024.

- globaldefs::sSrlgGroupList_T srlgGroupListValue
  Supported. This field holds the SRLG group values. This is an optional parameter that you can include in the SRLG group during the link creation. A maximum of five SRLG groups are supported for a given link. SRLG groups are only supported for SONET/SDH server trail links.
  Prime Optical support for ServerTrailLink_T structure is as follows:
- globaldefs::NamingAttributes_T name
  Supported. This field has two tuples. The first tuple is for EMS. The second one is for ServerTrailLink. The value in the second tuple is the same as the nativeEMSName field.
- string userLabel
  Not supported.
- boolean forceUniqueness

Not supported

- string owner
  Not supported.
- globaldefs::ConnectionDirection_T direction
  Prime Optical supports only CD_BI for bidirectional links.
- transmissionParameters::LayerRate_T rate
  Supported layer rates are:
  - LR_STS1_and_AU3_High_Order_VC3
  - LR_STS3c_and_AU4_VC4
  - LR_STS12c_and_VC4_4c
  - LR_STS24c_and_VC4_8c
  - LR_STS48c_and_VC4_16c
  - LR_STS192c_and_VC4_64c
- globaldefs::NamingAttributes_T aEndTP
  Supported. The CTP for the serverTrailLink aEnd.
- globaldefs::NamingAttributes_T zEndTP
  Supported. The CTP for the serverTrailLink zEnd.
- globaldefs::NVSList_T additionalCreationInfo
  Supported. The supported parameters are listed in the following table.

Managed Element Type	Parameter Name	Supported Parameter Values
All managed elements that support ServerTrailLink	serverTrailLinkProtectionType	PREEMPTIBLE, UNPROTECTED, FULLY_PROTECTED.
All managed elements that support ServerTrailLink	linkCost	Integer, 0 to 999999.
All managed elements that support ServerTrailLink	linkCount	Integer, 1 to 1024.

- globaldefs::sSrlgGroupList_T srlgGroupListValue
  Supported. This field holds the SRLG group values. This is an optional parameter that you can include in the SRLG group during the link creation. A maximum of five SRLG groups are supported for a given link. SRLG groups are only supported for SONET/SDH server trail links.

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised if input createData is not valid.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.3.30 deleteServerTrailLink

Synopsis

```
void deleteServerTrailLink(
        in globaldefs::NamingAttributes_T serverTrailLinkName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to delete a server trail link manually between two MEs. Prime Optical throws the EXCPT_UNABLE_TO_COMPLY exception if the server trail link cannot be deleted in the

current state. Prime Optical generates an object deletion event notification when a server trail link is deleted.

Parameters

Name	Type	Input/ Output	Description
serverTrailLinkName	globaldefs::NamingAttributes_T	In	Name of the server trail link to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised if input the server trail link name is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised if the specified server trail link name does not exist in Prime Optical.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_UNABLE_TO_COMPLY - Raised if attempt is made to delete a server trail which cannot be deleted.

Compliance

Prime Optical-defined.

3.3.31 addNetworkPartition

Synopsis

```
void addNetworkPartition(
in string name, in string description)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the OSS to add a new network partition.

Parameters

Name	Type	Input/Output	Description
name	string	In	Network partition name.
description	string	In	Description of the new partition.

Throws

globaldefs::ProcessingFailureException

Compliance

Prime Optical-defined.

3.3.32 addMultiLayerSubnetwork

Synopsis

```
void addMultiLayerSubnetwork(in MultiLayerSubnetworkCreateData _T
subnetCreateData)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows the NMS to add a new multilayer subnetwork. Upon successful addition, the EMS generates an object creation event for the multilayer subnetwork object.

Parameters

Name	Type	Input/ Output	Description
subnetCreateData	MultiLayerSubnetworkCreateData _T	In	MultiLayerSubnetwork input data.

The structure for MultiLayerSubnetworkCreateData _T is as follows:

```
struct MultiLayerSubnetworkCreateData_T
{
        globaldefs::NamingAttributes_T subnetworkName;
        string userLabel;
        string nativeEMSName;
        string owner;
        multiLayerSubnetwork::Topology_T subnetworkType;
        globaldefs::NVSList_T additionalInfo ;
};
```

The description of each parameter is as follows:

Name	Type	Description
subnetworkName	globaldefs::NamingAttributes_T	New subnet partition name.
userLabel	String	User label.
nativeEMSName	String	EMS name.
owner	String	Owner.
subnetworkType	multiLayerSubnetwork::Topology_T	Subnetwork topology.
additionalInfo	globaldefs::NVSList_T	The NMS uses the additionalInfo attribute to communicate to the EMS any other information required to create the multiLayerSubnetwork.

```
enum Topology_T  { TOPO_SINGLETON,  TOPO_CHAIN,  TOPO_PSR,
    TOPO_OPEN_PSR,  TOPO_SPRING,  TOPO_OPEN_SPRING,  TOPO_MESH  };
```

The structure of Additional Info is as follows:

Name	Value	Description	Mandatory or Optional
NETWORK PARTITION	subnetwork name	Name of the network partition where the subnet is added.	Mandatory
Type	subnetwork type	Subnet type.	Mandatory (see the following values)

The parameter name “Type” in additionalInfo can have the following values:

- SONET
- SDH
- OTHER
- UNKNOWNSUBNETTYPE (the default value if a value other than the three preceding values is specified)

Throws

globaldefs::ProcessingFailureException

Compliance
Prime Optical-defined.

3.3.33 deleteNetworkPartition

Synopsis
```
void deleteNetworkPartition(in string name)
raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the OSS to delete an existing network partition.
Parameters

Name	Type	Input/Output	Description
name	string	In	Network partition name to delete.

Throws
```
globaldefs::ProcessingFailureException
```

Compliance
Prime Optical-defined.

3.3.34 deleteMultiLayerSubnetwork

Synopsis
```
void deleteMultiLayerSubnetwork(in globaldefs::NamingAttributes_T
subnetworkName) raises(globaldefs::ProcessingFailureException);
```

Description
This interface allows the NMS to delete an existing subnetwork. Successful operation requires the EMS to generate an object deletion event.
Parameters

Name	Type	Input/Output	Description
subnetworkName	globaldefs::NamingAttributes_T	In	Subnet name to delete.

Throws
```
globaldefs::ProcessingFailureException
EXCPT_ENTITY_NOT_FOUND- Raised if the multilayer subnetwork is not found.
EXCPT_UNABLE_TO_COMPLY- Raised if the EMS cannot comply.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
```

Compliance
Prime Optical-defined.

3.3.35 getAllSupportedMEModels

Synopsis
```
void getAllSupportedMEModels(out globaldefs::NVSList_T supportedMEs)
raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of all of the models that the EMS supports. A model's structure has the following form:

- name = "ModelName"
- value = meModel

Parameters

Name	Type	Input/Output	Description
supportedMEs	NVSList_THolder	Out	List of all of the supported model names.

Throws

```
globaldefs::ProcessingFailureException
```

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.4 EmsSession::EmsSession_I

The emsSession module enables the client to query the EMS to determine the manager interfaces that the EMS supports. The NMS retrieves an instance of the required manager interface objects. This interface is inherited from session::Session_I.
interface EmsSession_I:session::Session_I

3.4.1 getEventChannel

Synopsis

```
void getEventChannel(
        out CosNotifyChannelAdmin::EventChannel eventChannel)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to access the event channel to receive notifications. Prime Optical publishes the name of the notification channel through the naming service. You can use the naming service to get the corresponding object reference.

Parameters

Name	Type	Input/ Output	Description
eventChannel	CosNotifyChannelAdmin::EventChannel	Out	Event channel that the NMS uses.

Throws

```
globaldefs::ProcessingFailureException
```

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

TMF-defined.

3.4.2 GetManager

Synopsis

```
void getManager(
        in string managerName,
        out common::Common_I managerInterface)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to access the specified manager interface.

Parameters

Name	Type	Input/Output	Description
managerName	String	In	Class or type of manager object that the client wants.

Name	Type	Input/Output	Description
managerInterface	common::Common_I	Out	The actual object returned implements the specified manager interface. However, it is returned as a Common_I object, so the operation can be generic. The client narrows the returned object to the correct object type.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - This exception indicates that the EMS does not support the manager.

EXCPT_INTERNAL_ERROR - This exception indicates a nonspecific internal EMS failure.

Compliance

TMF-defined.

3.4.3 getSupportedManagers

Synopsis

```
void getSupportedManagers(out managerNames_T supportedManagerList)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request the manager interfaces that the EMS implements.

Parameters

Name	Type	Input/ Output	Description
supportedManagerList	managerNames_T	Out	List of manager names supported in the form managerName, where managerName is one of the following defined manager strings for Prime Optical: • EMS • ManagedElement • MultiLayerSubnetwork • EquipmentInventory • PerformanceManagement • Maintenance • Protection • Software

Throws

None.

Compliance

TMF-defined.

3.5 emsSessionFactory::EmsSessionFactory_I

A single instance of the EmsSessionFactory_I is the entry point to the EMS. The NMS uses this object reference to connect to Prime Optical. This interface implements the version interface and returns the server IDL version. It is inherited from mtnmVersion::Version_I.

```
interface EmsSessionFactory_I:mtnmVersion::Version_I
```

3.5.1 getEmsSession

Synopsis

```
void getEmsSession(in string user,
        in string password,
        in nmsSession::NmsSession_I client,
        out emsSession::EmsSession_I emsSessionInterface)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the NMS to obtain the EmsSession_I object, which contains all managers of the EMS.

Prime Optical allows 4 to 25 concurrent login sessions and as many user profiles as required for configuring the number of concurrent login sessions. See 6.8 Location of the Naming Service IOR File to configure the number of concurrent login sessions. The username and password must contain from 1 to 53 characters.

This interface can also accept the encrypted username and password. To enable encryption on the username and password, check the **Enable encryption for username and password** check box in the Prime Optical client Control Panel for the GateWay/CORBA service. Prime Optical then decrypts the username and password accordingly.

To encrypt the username and password, you must use the RSA algorithm with the public key or the public key pair provided by Prime Optical. To obtain the Prime Optical public key, use the emsSessionFactory::EmsSessionFactory_I::getEmsPublicKey API. To obtain the Prime Optical public key pair, use the emsSessionFactory::EmsSessionFactory_I::getEmsPublicKeyPair API.

Parameters

Name	Type	Input/ Output	Description
user	String	In	User or application that is trying to access the server. This user must have a profile created in Prime Optical.
password	String	In	Password for the user application.
client	nmsSession::NmsSession_I	In	Handle to the NMS NmsSession_I object, to which the returned EmsSession_I object must be associated.
emsSessionInterface	emsSession::EmsSession_I	Out	CORBA Interoperable Object Reference (IOR) for the EmsSession_I interface.

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of a nonspecific internal EMS failure.

EXCPT_INVALID_INPUT - Raised when the client is invalid or when the length of
the username or password passed is not within the permissible limits (1-53).
```

EXCPT_ACCESS_DENIED - Raised in the following cases:

When all the eight login sessions are used up.

When the user name specified does not exist.

Or in case of any other security violation.

Compliance
TMF-defined.

3.5.2 getEmsPublicKey

Synopsis

```
void getEmsPublicKey(out string publicKey)
     raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the NMS to obtain the string representation of the RSA public key encoded in the Base64 encoding scheme. The operation encrypts the username and password from the string representation of the RSA public key.
The OSS client must use the 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; in this case, X.509 SubjectPublicKeyInfo.
You can create the RSA public key using this byte array and cryptographic libraries. Use this key to encrypt the username and password. Bouncy Castle Provider is an example of a security provider that you can use to create the RSA public key.
The public key is then used to encrypt the username and password. Before passing the encrypted password to Prime Optical for login, the OSS client must encode the encrypted username and password using the Base64 encoder to get the string equivalent of the encrypted data.

Parameters

Name	Type	Input/Output	Description
publicKey	string	Out	Public key string returned by Prime Optical.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.5.3 getEmsPublicKeyPair

Synopsis

```
void getEmsPublicKeyPair(out string publicKeyModulus,
        out string publicKeyExponent)
     raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the NMS to obtain the string representation of the RSA public key pair (in the modulus and exponent forms) from Prime Optical for encrypting the username and password.
The OSS client must convert the string representation of the public key modulus and public key exponent into a nonnegative integer and create an RSA public key using cryptographic libraries. Crypto++ is one of the security providers available.

Before passing the encrypted password to Prime Optical for login, the OSS client must use the Base64 encoder to encode the encrypted username and password. This provides the string equivalent of the encrypted data.

Parameters

Name	Type	Input/Output	Description
publicKeyModulus	string	Out	Public key modulus string returned by Prime Optical.
publicKeyExponent	string	Out	Public key exponent string returned by Prime Optical.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.6 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.

3.6.1 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 interface enables an NMS to request all of the equipment and equipment holders contained in a managed element or equipment holder.

Parameters

Name	Type	Input/ Output	Description
meOrHolderName	globaldefs::NamingAttributes_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.

Prime Optical support for Equipment_T is 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 value in the nativeEMSName field.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. The naming convention is described in 2.2.5.7 Equipment Holder.
- string owner
  - Not supported.
- boolean alarmReportingIndicator
  - Supported for ONS 15327 and ONS 15454 and ONS NCS2K only.
- ServiceState_T serviceState
  - Not supported.
- EquipmentObjectType_T expectedEquipmentObjectType
  - ONS 15216—All values listed in Table 8-1 are supported.
  - ONS 15305 and ONS 15305 CTC—All values listed in Table 8-2 and Table 8-3 are supported.
  - ONS 15310 CL—All values listed in Table 8-4 are supported.
  - ONS 15310 MA SDH—All values listed in Table 8-5 are supported.
  - ONS 15310 MA SONET—All values listed in Table 8-6 are supported.
  - ONS 15327—All values listed in Table 8-7 are supported.
  - ONS 15454 SDH—All values listed in Table 8-8 are supported.
  - ONS 15454 SONET—All values listed in Table 8-9 are supported.
  - ONS 15600 SDH—All values listed in Table 8-13 are supported.
  - ONS 15600 SONET—All values listed in Table 8-14 are supported.
  - CPT 200 SONET—All values listed in Table 8-10 and Table 8-11 are supported.
  - CPT 200 SDH—All values listed in Table 8-10 and Table 8-11 are supported.
  - CPT 600 SONET—All values listed in Table 8-10 and Table 8-11 are supported.
  - CPT 600 SDH—All values listed in Table 8-10 and Table 8-11 are supported.
  - NCS 2000 SONET —All values listed in Table 8-17 and Table 8-18 are supported.
  - NCS 2000 SDH —All values listed in Table 8-17 and Table 8-18 are supported.
  - NCS 2006 SONET —All values listed in Table 8-17 and Table 8-18 are supported.
  - NCS 2006 SDH —All values listed in Table 8-17 and Table 8-18 are supported.
  - NCS 2015 SONET/SDH —All values listed in Table 8-17 and Table 8-18 are supported.

- 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.

Prime Optical support for EquipmentHolder_T is 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 2.2.5.7 Equipment Holder.
- string owner
  - Not supported.
- boolean alarmReportingIndicator

Not supported.

- EquipmentHolderType_T holderType
  - Supported values are additionalHolder, rack, shelf, and slot.
  - For the ONS 15600 SONET and ONS 15600 SDH, valid values are rack, shelf, slot, sub_slot, and ppm_holder.
  - For the ONS 15310 CL, ONS 15310 MA SONET, and ONS 15310 MA SDH, valid values are rack, shelf, slot, and ppm_holder.
  - For the ONS 15454 SONET,ONS 15454 SDH, ONS NCS2K SONET and ONS NCS2K SDH valid values are rack, shelf, slot, ppm_holder, and port.
  - For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, valid values are rack, shelf, slot, ppm_holder, and port.
  - For all equipment supporting SFP modules, valid values are rack, shelf, slot, sub_slot, and port_holder.
- globaldefs::NamingAttributes_T expectedOrInstalledEquipment
  - Supported. If the slot is empty and not preprovisioned for equipment, this field is empty.
- EquipmentObjectTypeList_T acceptableEquipmentTypeList
  - Supported.

Table 3-2: Acceptable Equipment for ONS 15454 SONET

Acceptable Equipment	Classic Shelf	M2 Shelf	M6 Shelf
OPT_AMP_17_C, OPT_AMP_L, OPT_BST, OPT_BST_E, OPTBST_L, OPT_PRE, OPT_AMP_C,OPT_RAMP_C, OPT_RAMP_CE, 40-SMR1-C, 40-SMR2-C, TDCU-C, TDCU-F	1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 17	2, 3	2, 3, 4, 5, 6, 7
OPT_EDFA_17, OPT_EDFA_24, PPMESH - 4, 15216 MD 40 EVEN, MD-ID-50, 15216-FLD-4, PASSIVE OSC, PPMESH-SMR, PPMESH - 8, 15216 MD 40 ODD, 15216-MD-ID-50, 15216-FLC-8-MD, PASSIVE DCU			
10GE_XP, GE_XP, 10DME, 2.5G_DM, 2.5G_DMP, ADM_10G, MXP_2.5G_10E, MXP_2.5G_10G, TXPP_MR_2.5G, TXP_MR_2.5G, TXP_MR_10E, TXP_MR_10G, TXP_MR_2.5G,OTU2_XP			

Acceptable Equipment	Classic Shelf	M2 Shelf	M6 Shelf
32DMX, 32DMX_L, 32DMX_O, 32MUX_O, 32_WSS, 32WSS_L, 4MD, 40DMX, 40MUX, 40WSS, 4MD			
40WXC			
AD_1B, AD_1C, AD_2C, AD_4B, AD_4C			
MMU, OSC-CSM,PSM			
80WXC-C, 40G_MXP_C, 40E_MXP_C			
ML1000, ML100T, ML-100X-8, ML-MR-10, CE-MR-10, CE-100T-8, CE-1000-4 (Only when XC10G or XCVXL-10G is installed in slot 8 or 10.)	1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 17	N/A	N/A
DS1, DS3, DS3E, DS3XM, DS3XM_12, DS3I, ETH100, ETH1000, MRC_12, MRC25G_4, OC12, OC3_4, OC12_4, OC3_8, OC48	1, 2, 4, 14, 16, 17	N/A	N/A
DS1, DS3, DS3E, DS3XM, DS3XM_12, DS1N, DS3IN, DS3N, DS3NE, ETH100, ETH1000, MRC_12, MRC25G_4, OC12, OC3_4, OC12_4, OC3_8, OC48	3, 15	N/A	N/A
DS1, DS3, DS3E, DS3XM, DS3XM_12, DS3I, ETH100, ETH1000, MRC_12, MRC25G_4, OC12, OC3_4, OC48, OC192, OC192_XFP	5, 16, 12, 13	N/A	N/A
TCC, TCC2, TCC3	7, 11	N/A	N/A
XC10G, OSCM	8, 10	N/A	N/A
AIC, AICI	9	N/A	N/A
TCC3	N/A	N/A	1, 8

Table 3-3: Acceptable Equipment for ONS 15454 SDH

Acceptable Equipment	Classic Shelf	M2 Shelf	M6 Shelf

Acceptable Equipment	Classic Shelf	M2 Shelf	M6 Shelf
OPT_AMP_17_C, OPT_AMP_L, OPT_BST, OPT_BST_E, OPTBST_L, OPT_PRE, OPT_AMP_C,OPT_RAMP_C, OPT_RAMP_CE, 40-SMR1-C, 40-SMR2-C, TDCU-C, TDCU-F OPT_EDFA_17, OPT_EDFA_24, PPMESH-4, 15216 MD 40 EVEN, MD-ID-50,15216-FLD-4, PASSIVE OSC, PPMESH-SMR, PPMESH - 8, 15216 MD 40 ODD,15216-MD-ID-50, 15216-FLC-8-MD, PASSIVE DCU 10GE_XP, GE_XP, 10DME, 2.5G_DM, 2.5G_DMP, ADM_10G, MXP_2.5G_10E, MXP_2.5G_10G, TXPP_MR_2.5G, TXP_MR_2.5G, TXP_MR_10E, TXP_MR_10G, TXP_MR_2.5G,OTU2_XP 32DMX, 32DMX_L, 32DMX_O, 32MUX_O, 32_WSS, 32WSS_L, 4MD, 40DMX, 40MUX, 40WSS, 4MD 40WXC AD_1B, AD_1C, AD_2C, AD_4B, AD_4C MMU, OSC-CSM,PSM	1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 17	2, 3	2, 3, 4, 5, 6, 7
ML1000, ML100T, ML-100X-8, ML-MR-10, CE-MR-10, CE-100T-8, CE-1000-4 (Only when XC10G or XCVXL-10G is installed in slot 8 or 10.)	1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 17	N/A	N/A
DS1, DS3, DS3E, DS3XM, DS3XM_12, DS3I, ETH100, ETH1000, MRC_12, MRC25G_4, OC12, OC3_4, OC12_4, OC3_8, OC48	1, 2, 4, 14, 16, 17	N/A	N/A
E3, DS3IN, E1_42, STM1E_12, STM1_8, STM4_4, ETH100, ETH1000, MRC12, MRC25G_12, STM1_4, STM16, STM4	3, 15	N/A	N/A

Acceptable Equipment	Classic Shelf	M2 Shelf	M6 Shelf
DS1, DS3, DS3E, DS3XM, DS3XM_12, DS1N, DS3IN, DS3N, DS3NE, ETH100, ETH1000, MRC_12, MRC25G_4, OC12, OC3_4, OC12_4, OC3_8, OC48	3, 15	N/A	N/A
DS31, E3, ETH100, ETH1000,MRC_12,MRC25G_1 2, STM1_4, STM16, STM4, STM64, STM64_XFP	5, 13	N/A	N/A
ETH100, ETH1000, MRC12, MRC25G_12, STM1_4, STM16, STM4, STM64, STM64_XFP	6, 12	N/A	N/A
TCC, TCC2, TCC3	7, 11	N/A	N/A
XCVXC_10G, XC10G, XCVXL_10G, XCVXL_2.5, XCVXL_10G, OSCM	8, 10	N/A	N/A
AICI	9	N/A	N/A
TCC3	N/A	1	1, 8

Table 3-4: Acceptable Equipment for ONS NCS2K SONET and SDH

Acceptable Equipment	M2 Shelf	M6 Shelf	M15 Shelf
OPT_EDFA_17, OPT_EDFA_24 EDRA1-26C, EDRA1-35C, EDRA2-26C, EDRA2-35C	2, 3 2,4	2, 3, 4, 5, 6, 7 3, 4, 6	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
10GE_XP, 10DME, MXP_2.5G_10E, TXP_MR_10E, OTU2_XP			
40G_MXP_C 40E_MXP_C			
16-WXC-FS			
RAMAN_CTP,RAMAN_COP	N/A	2,3,4,5,6,7	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
100G-LC-C,100GS-CK-LC,100G-CK-C,200G-CK-LC,MR-MXP,17-SMR9-FS,24-SMR9-FS,34-SMR9-FS, 20-SMR-FS,12-AD-FS,16-AD-FS,AR_XPE,AR_MXP	N/A	2,3,4,5,6,7	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
TNC	1	1, 8	1, 17

ONS 15216 FlexLayer Slot Number	Acceptable Equipment
1, 2, 3, 4	FLA-8-*channel ID* *Channel ID* = 36.6, 44.5, 52.5, 60.6
	FLB-2-*channel* *Channel ID* = 31.1, 32.6, 35.0, 36.6, 38.9, 40.5, 42.9, 44.5, 46.9, 48.5, 50.9, 52.5, 54.9, 56.5, 58.9, 60.6
	CS-2
	CS-3
	CS-4
	CS-SM-Y
	CS-MM-Y
	SC-4B
1, 2, 3	VOA-4

ONS 15216 DCU R2.0 Slot Number	Acceptable Equipment
1, 2	DCU-100 (100 ps/nm)
1, 2	DCU-350 (350 ps/nm)
1, 2	DCU-450 (450 ps/nm)
1, 2	DCU-550 (550 ps/nm)x
1, 2	DCU-750 (750 ps/nm)
1, 2	DCU-950 (ps/nm)
1, 2	DCU-1150 (1150 ps/nm)
1, 2	DCU-E-200 (200 ps/nm)
1, 2	DCU-E-350 (350 ps/nm)
1, 2	DCU-L-300 (300 ps/nm)
1, 2	DCU-L-600 (600 ps/nm)
1, 2	DCU-L-700 (700 ps/nm)
1, 2	DCU-L-800 (800 ps/nm)
1, 2	DCU-L-1000 (1000 ps/nm)

ONS 15216 DCU R2.0 Slot Number	Acceptable Equipment
1, 2	DCU-L-1100 (1100 ps/nm)
1, 2	DCU-DS-L-100 (100 ps/nm)
1, 2	DCU-DS-L-200 (200 ps/nm)
1, 2	DCU-DS-L-300 (300 ps/nm)

ONS 15305 and ONS 15305 CTC Slot Number	Acceptable Equipment
1, 2, 3, 4	All equipment listed in 8.2 Equipment List for ONS 15305

ONS 15310 CL Slot Number	Acceptable Equipment
1	CTX-2500, CTX-FILLER
2	CE-100T-8, ML-100T-8, CE-MR-6

ONS 15310 MA SDH Slot Number	Acceptable Equipment
3, 4	CTX-2500, CTX-FILLER
1, 2, 5, 6	E1-21/E3-DS3-3, E1-63/E3-DS3-3
1, 2, 5, 6	CE-100T-8, ML-100T-8, CE-MR-6, EXP-FILLER

ONS 15310 MA SONET Slot Number	Acceptable Equipment
3, 4	CTX-2500, CTX-FILLER
1, 2, 5, 6	DS1-84/DS3-EC1-3, DS1-28/DS3-EC1-3, DS3-EC-6
1, 2, 5, 6	CE-100T-8, ML-100T-8, CE-MR-6, EXP-FILLER

ONS 15327 Slot Number (All Releases)	Acceptable Equipment
7, 8	MIC
5, 6	XTC
1, 2, 3, 4	ETH100, OC12, OC48

ONS 15600 SDH Slot Number	Acceptable Equipment
5,10	TXC
6/7, 8/9	SSXC

1, 2, 3, 4, 11, 12, 13, 14	OC48_16, OC192_4, OC48_32
1, 2, 3, 4, 11, 12, 13, 14	ASAP, STM64_4_DWDM

ONS 15600 SONET Slot Number	Acceptable Equipment
5, 10	TXC
6/7, 8/9	SSXC
1, 2, 3, 4, 11, 12, 13, 14	OC48_16, OC192_4, OC48_32
1, 2, 3, 4, 11, 12, 13, 14	ASAP, OC192_4_DWDM

CPT 200 SONET Slot Number	Acceptable Equipment
2, 3	PT_10GE_4, PTF_10GE_4
1	TNC, TSC
36 onwards	PTSA_GE

CPT 200 SDH Slot Number	Acceptable Equipment
2, 3	PT_10GE_4, PTF_10GE_4
1	TNC, TSC
36 onwards	PTSA_GE

CPT 600 SONET Slot Number	Acceptable Equipment
2, 3, 4, 5, 6, 7	PT_10GE_4, PTF_10GE_4, PTSA_GE
1, 8	TNC, TSC
36 onwards	PTSA_GE

CPT 600 SDH Slot Number	Acceptable Equipment
2, 3, 4, 5, 6, 7	PT_10GE_4, PTF_10GE_4, PTSA_GE
1, 8	TNC, TSC
36 onwards	PTSA_GE

Compliance
TMF-defined.

3.6.2 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 interface has the same behavior as the interface described in 3.6.1 getAllEquipment, but returns the object name instead of returning the entire object.

Parameters

Name	Type	Input/ Output	Description
meOrHolderName	globaldefs::NamingAttributes_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.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining names.

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when meOrHolderName does not reference a managed
element or an equipment holder.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementOrEquipmentName references
an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators
reaches 128.
```

Compliance

TMF-defined.

3.6.3 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 interface enables 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).
If the equipment is protection equipment in a 1_FOR_N protection group, Prime Optical reports only the AlarmReporting for the LayeredParameters of the supported PTPs.

- ONS 15454 SONET: Prime Optical returns OC-n, DS1, DS3, EC-1, fibre channel, and front Ethernet ports.
- ONS 15454 SDH: Prime Optical returns STM-n (OC-n), E1, E3, DS3I, STM1E, fibre channel, and front Ethernet ports.
- ONS 15327: Prime Optical returns OC-n, DS1, DS3, and Ethernet ports.
- ONS 15216: Prime Optical returns all the supported PTPs for ONS 15216, if there is a topological link associated with the NE.
- ONS 15600 SONET: Prime Optical returns OC-n ports.
- ONS 15600 SDH: Prime Optical returns STM-n ports.
- ONS 15305 and ONS 15305 CTC: Prime Optical returns WAN, SDH, and PDH ports.
- ONS 15310 CL and ONS 15310 MA: Prime Optical returns OC-n, DS1, DS3, and Ethernet ports.
- CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH: Prime Optical returns OTU and front Ethernet ports.

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 report 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.

Prime Optical support for TerminationPoint_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported. This field contains three tuples for PTP and four tuples for CTP as defined by the TMF. The third tuple contains positional information for the PTP. The naming convention is described in 2.2.5.3 Physical Termination Point, 2.2.5.4 Connection Termination Point, and 2.2.5.5 Floating Termination Point.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported.

- string owner
  - Not supported.
- globaldefs::NamingAttributes_T ingressTrafficDescriptorName
  - Not supported.
- globaldefs::NamingAttributes_T egressTrafficDescriptorName
  - Not supported.
- TPType_T type
  - Supported. Prime Optical supports TPT_PTP, TPT_CTP, and TPT_FTP.
- TPConnectionState_T connectionState
  - Supported. For PTP, it is TPCS_NA; for CTP and floating termination point (FTP), it is TPCS_BI_CONNECTED.
- TerminationMode_T tpMappingMode
  - Not supported.
- Directionality_T direction
  - Supported. PTPs always report D_BIDIRECTIONAL.
- transmissionParameters::LayeredParameterList_T transmissionParams
  - Supported. Layer rate is supported for PTP, CTP, and FTP.
  - globaldefs::NamingAttributes_T[] additionalInfo
  - list of additionalInfo parameters, contains the “dataRate” attribute value, for more details refer to 11 PTP dataRate attribute.

Prime Optical supports the following parameters for PTP.

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
AlarmReporting			
PTP	ONS 15310 CL	On, Off	All equipment (except equipment that does not have PTP, such as XC, TCC2, CXC, and so on).
	ONS 15310 MA		
	ONS 15327		
	ONS 15454 SONET		
	ONS 15454 SDH		
	ONS 15600 SONET		
	ONS NCS2K SDH		
	ONS NCS2K SONET		
	ONS 15600 SDH		
	CPT 200 SONET		
	CPT 200 SDH		
	CPT 600 SONET		
	CPT 600 SDH		
OperationalState			
PTP	CPT 200 SONET	—	All equipment (except ML and CE
	CPT 200 SDH		

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
	CPT 600 SONET		cards, FCMR, and equipment that does not have PTP, such as XC, TCC2, and so on).
	CPT 600 SDH		
	ONS 15310 CL		
	ONS 15310 MA		
	ONS 15454 SONET		
	ONS 15454 SDH		
	ONS NCS2K SONET		
	ONS NCS2K SDH		
	ONS 15600 SONET		
	ONS 15600 SDH		
AdminState			
PTP	ONS 15310 CL	IN_SERVICE, OUT_OF_SERVICE, OUT_OF_SERVICE_BY_MAINTENANCE (for NE R3.4.1 or later), AUTO_IN_SERVICE (for NE R3.4.1 or later) Down, Up	All equipment (except ML and CE cards, FCMR, and equipment that does not have PTP, such as XC, TCC2, and so on). G1000_4, G1000_2, and FCMR do not support the AUTO_IN_SERVICE value. For ML-series cards, Down and Up are valid values, and are read-only. For CE-series and E-series Ethernet cards, valid values are In Service, Auto In Service, Out of Service, and Out of Service by Maintenance. SERV_NA is not a valid value for any card.
	ONS 15310 MA		
	ONS 15327		
	ONS 15454 SONET		
	ONS NCS2K SONET		
	CPT 200 SONET		
	CPT 600 SONET		
	ONS 15600	IN_SERVICE, OUT_OF_SERVICE_BY_MAINTENANCE, OUT_OF_SERVICE (for NE R5.0 or later), AUTO_IN_SERVICE (for NE R5.0 or later)	All equipment (except equipment that

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
			does not have PTP, such as CXC and so on).
	ONS 15454 SDH ONS 15600 SDH CPT 200 SONET CPT 600 SONET ONS NCS2K SDH	UNLOCKED LOCKED, DISABLED LOCKED, MAINTENANCE UNLOCKED, AUTO_IN_SERVICE	All equipment (except ML and CE cards, and equipment that does not have PTP, such as XC, TCC2, and so on). FCMR does not support the UNLOCKED, AUTO_IN_SERVICE value. The ONS 15600 SDH does not support LOCKED, DISABLED and UNLOCKED, AUTO IN SERVICE.
LineCode			
PTP	ONS 15310 CL ONS 15310 MA ONS 15327	ami, b8zs	DS1 port on XTC and CTX cards.
	ONS 15454 SONET ONS NCS2K SONET	ami, b8zs	DS1, DS1N.
	ONS 15310 CL ONS 15310 MA ONS 15454 SONET ONS NCS2K SONET	b3zs	DS3E, DS3NE, DS3XM, DS3I, DS3IN, DS3, DS1-84/DS3-EC1-3, DS1-28/DS3-EC1-3, ONS 15310 CL and ONS 15310 MA DS3 ports on CTX cards.
	ONS 15454 SDH ONS NCS2K SDH	hdb3	E1, E1N, E1_42.
	ONS 15454 SDH ONS NCS2K SDH	b3zs	DS3I, DS31N.
FrameFormat			

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
PTP	ONS 15310 CL ONS 15310 MA ONS 15327	esf, d4, unframed, auto provision fmt (only on ONS 15310)	DS1 port on XTC cards.
	ONS 15454 SONET ONS NCS2K SONET	esf, d4, unframed	DS1, DS1N.
	ONS 15454 SONET ONS NCS2K SONET	m13, cbit, unframed, auto provision fmt	DS3E, DS3NE, DS3I, DS31N, DS3.
	ONS 15454 SONET ONS NCS2K SONET	m13, cbit	DS3XM.
	ONS 15454 SDH ONS NCS2K SDH	e1_mf, e1_crcmf, e1_unframed	E1, E1N, E1_42.
	ONS 15310 CL ONS 15310 MA	m13, cbit, unframed	DS3 port on CTX cards.
	ONS 15454 SDH ONS NCS2K SDH	m13, cbit, unframed, auto provision fmt	DS3I, DS31N.
SDH_SONET_SS_BITS			
PTP	ONS 15454 SONET ONS NCS2K SONET	SDH, SONET	OC-n equipment.
	ONS 15454 SDH ONS NCS2K SDH	SDH	STM-n equipment.
EnableSyncMsg			
PTP	ONS 15310 CL ONS 15310 MA ONS 15327 ONS 15454 SONET ONS NCS2K SONET ONS 15600 SONET	On, Off	OC-n equipment.
	ONS 15454 SDH ONS NCS2K SDH ONS 15600 SDH,	On, Off	STM-n equipment.
ALSMode			
PTP	ONS 15454 SONET ONS NCS2K SONET	DISABLED, AUTO, MANUAL, MANUAL_RESTART_FOR_TEST	OC3_8, OC48 ELR, OC192 equipment only.
	ONS 15454 SDH ONS NCS2K SDH	DISABLED, AUTO, MANUAL, MANUAL_RESTART_FOR_TEST	STM1_8, STM16 ELR, STM64 equipment only.
PulseWidth			
PTP	ONS 15454 SONET ONS NCS2K SONET	Float value from 2.0 to 100.0	OC3_8, OC48 ELR, OC192, 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC equipment only.

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
	ONS 15454 SDH ONS NCS2K SDH	Float value from 2.0 to 100.0	STM1_8, STM16 ELR, STM64, 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC equipment only.
RecoveryInterval			
PTP	ONS 15454 SONET ONS NCS2K SONET	Integer value from 100 to 300	OC3_8, OC48 ELR, OC192, 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC equipment only.
	ONS 15454 SDH ONS NCS2K SDH	Integer value from 100 to 300	STM1_8, STM16 ELR, STM64, 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC equipment only.
HasLaserBeenShutdown			
PTP	ONS 15454 SONET ONS NCS2K SONET	True, False (read-only value)	OC3_8, OC48 ELR, OC192, 10x10G-LC, WSE, 100G-LC-C, CFP-LC, 100G-LC-C, M-CFP-LC equipment only.
	ONS 15454 SDH ONS NCS2K SDH	True, False (read-only value)	STM1_8, STM16 ELR, STM64, 10x10G-LC, WSE, 100G-LC, M-CFP-LC equipment only.
RestartLaser			
PTP	ONS 15454 SONET ONS NCS2K SONET	True, False (write-only value)	OC3_8, OC48 ELR, OC192, 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC equipment only.
	ONS 15454 SDH ONS NCS2K SDH	True, False (write-only value)	STM1_8, STM16 ELR, STM64 equipment only.
IPPMMonitor			

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
CTP	ONS 15310 CL ONS 15310 MA ONS 15327 ONS 15454 SONET ONS NCS2K SONET ONS 15600 SONET ONS 15600 SDH	On, Off	STS CTPs on OC-n cards.
	ONS 15454 SDH ONS NCS2K SDH	On, Off	VC4 CTPs on STM-n cards.
TrailTraceActualTx			
CTP	ONS 15454 SONET ONS NCS2K SONET	Free-format string	DS3E, DS3NE, DS3XM, DS3I, DS31N.
	ONS 15454 SDH ONS NCS2K SDH	Free-format string	E3, DS3I, DS3IN, STM1E_12.
TrailTraceExpectedRx			
CTP	ONS 15310 CL ONS 15310 MA ONS 15327	Free-format string	OC3.
	ONS 15454 SONET ONS NCS2K SONET	Free-format string	DS3E, DS3NE, DS3I, DS3IN, DS3XM, EC1, OC3, OC3_8, OC12_4, OC48AS, OC192.
	ONS 15454 SDH ONS NCS2K SDH	Free-format string	E3, DS3I, DS3IN, STM1E_12, OC3, OC3_8, OC12_4, OC48AS, OC192.
	ONS 15600 SONET ONS 15600 SDH	Free-format string	OC48_16, OC192_4.
FTP	ONS 15327	Free-format string	G1000_2.
	ONS 15310 CL ONS 15310 MA	Free-format string	ML2_MAPPER and ML2_L2L3.
	ONS 15454 SONET ONS NCS2K SONET ONS 15454 SDH ONS NCS2K SDH	Free-format string	G1000_4, ML-series, and CE-series cards (except G1000; for all other cards it is supported only for R5.0 and later).
TrailTraceMonitor			

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
CTP	ONS 15310 CL ONS 15310 MA ONS 15327	On, Off	OC3.
	ONS 15454 SONET ONS NCS2K SONET	On, Off	DS3E, DS3NE, DS3I, DS3IN, DS3XM, EC1, OC3, OC3_8, OC12_4, OC48AS, OC192.
	ONS 15454 SDH ONS NCS2K SDH	On, Off	E3, DS3I, DS3IN, STM1E_12, OC3, OC3_8, OC12_4, OC48AS, OC192.
	ONS 15600 SONET ONS 15600 SDH	On, Off	OC48_16, OC192_4.
FTP	ONS 15327	On, Off	G1000_2.
	ONS 15310 CL ONS 15310 MA	On, Off	ML2_L2L3 and ML2_MAPPER.
	ONS 15454 SONET ONS NCS2K SONET ONS 15454 SDH ONS NCS2K SDH	On, Off	G1000_4, ML-series, and CE-series cards (except G1000; for all other cards it is supported only for R5.0 and later).
XCLoopBack			
CTP	ONS 15327 ONS 15454 SONET ONS NCS2K SONET ONS 15600 SONET ONS 15600 SDH	On, Off	STS CTPs on OC-n cards when line loopback is not set on the containing PTP.
	ONS 15454 SDH ONS NCS2K SDH	On, Off	VC4 CTPs on STM-n cards when line loopback is not set on the containing PTP.
PortMediaType			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	UNDEFINED, FibreChannel-1 Gbps, FibreChannel-2 Gbps, FICON-1 Gbps, and FICON-2 Gbps	FCMR (for NE R5.0 and later).
Link Recovery			
PTP	ONS 15454 SONET ONS 15454 SDH	On, Off	FCMR (for NE R5.0 and later).

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
	ONS NCS2K SONET ONS NCS2K SDH		
No. of Superblocks			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	From 4 to 32	FCMR (for NE R5.0 and later).
Enable Distance Extension			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	On, Off	FCMR (only in distance extension mode and for NE R5.0 and later).
Auto Detect Credits			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	On, Off	FCMR (only in distance extension mode and for NE R5.0 and later).
Number Credits			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	From 2 to 256, in multiples of 2	FCMR (only in distance extension mode and for NE R5.0 and later).
Autoadjust GFP Buffer Threshold			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	On, Off	FCMR (only in distance extension mode and for NE R5.0 and later).
GFP Buffers Available			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	From 16 to 1200, in multiples of 16	FCMR (only in distance extension mode and for NE R5.0 and later).
Ingress Idle Filtering			
PTP	ONS 15454 SONET ONS 15454 SDH ONS NCS2K SONET ONS NCS2K SDH	On, Off	FCMR (only in distance extension mode and for NE R5.0 and later).
MaxFrameSize			
PTP	ONS 15454 ONS NCS2K	From 2148 to 2172	FCMR (only in distance extension mode).
FrameFormat			
CTP	ONS 15454	Auto provision fmt, esf, d4, and unframed	DS3XM_12.

TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
	ONS NCS2K		
FDLMode			
CTP	ONS 15454 ONS NCS2K	T1.403 and BFDL	DS3XM_12.
DetectedLineType			
PTP	ONS 15454 ONS NCS2K	Read-only value	DS3E.

The following table lists additional parameters that are supported by OTU1 and OTU2 PTP.

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
RequestRestart	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• True	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C
EnableSyncMsgIn	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• True • False	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C
SendDoNotUse	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• Enabled • Disabled	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C
AINSSquelch	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• DISABLE • SQUELCH	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, CFP-LC-C
FEC	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K	• Disable • Standard • Enhanced-I.4 • Enhanced-I.7	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, WSE

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
		SDH		
TraceLevel	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• SECTION • PATH	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, CFP-LC-C, WSE
CurrentTransmitString	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• String	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, WSE
CurrentExpectedString	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• String	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, WSE
CurrentReceivedString	PTP	ONS15454 SONET, ONS15454 SDH	• String	OTU PTP on AR-XP, AR-MXP, AR-XPE, WSE
ProvideSync	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• True • False	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C
AsyncSyncMapping	—	—	• SYNC • ASYNC • ODU_MULTIPLE X • NOTUSED	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE
TraceMode	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• OFF • MANUAL • MANUALNOAIS	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, WSE
DisableFDIonTTIM	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K	• True • False	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, 100G-LC-C, WSE

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
		SONET, ONSNCS2K SDH		
SFBer	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• 1E-3 • 1E-4 • 1E-5	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 100G-LC-C
			• • 1E-5	Read-only PTP on 10x10G-LC, WSE
SDBer	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• 1E-5 • 1E-6 • 1E-7 • 1E-8 • 1E-9	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C
OTUMapping	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• None • CBR10G • ODU2E • ODU1E • ODU1MUX • ODU1	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE
VideoType	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• N/A • PAL • NTSC	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
G709OTN	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• Enabled • Disabled	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
AINSSoak	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Format: <hours>h:<minutes> • min	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
AdminSSMIn	PTP	ONS15454 SONET, ONS15454 SDH,	• PRS • STU • ST2	OTU PTP on AR-XP, AR-MXP, AR-XPE

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
		ONSNCS2K SONET, ONSNCS2K SDH	• TNC • ST3E • ST3 • SMC • ST4 • DUS • RES • G811 • STU_SDH • G812T • G812L • SETS DUS_SDH	PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
PulseWidth	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• Number of sec	OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
ODUTransparency	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• N/A • TRANSPARENT	OTU PTP on AR-XP, AR-MXP, AR-XPE
TerminationMode	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• TRANSPARENT • LINE • SECTION	Read-only OTU PTP on AR-XP, AR-MXP, AR-XPE PTP on 10x10G-LC, WSE, 100G-LC-C, M-CFP-LC
Overclock	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• YES • NO • N/A	PTP on 10x10G-LC, WSE, 100G-LC-C
MTU	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K	• 1548 • Jumbo	PTP on 10x10G-LC, WSE, 100G-LC-C for Ethernet Ports
			[1..9700]	Read and write OTU PTP on AR-XP, AR-MXP, AR-XPE for

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
		SDH		Ethernet ports
IncomingMACAddress	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• —	Read-only PTP on 10x10G-LC, WSE, 100G-LC-C
Mapping	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• GFP • WIS • CBR • TRP • NON	PTP on 10x10G-LC, WSE, 100G-LC-C
Type	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• SONET • SDH	Read-only PTP on 10x10G-LC, WSE, 100G-LC-C
SyncMsgIn	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• TRUE • FALSE	PTP on 10x10G-LC, WSE, 100G-LC-C
DisableAISRDIonTIMS	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• TRUE • FALSE	PTP on 10x10G-LC, WSE, 100G-LC-C
TransmitLength	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	• 1 byte • 16 bytes	PTP on 10x10G-LC, WSE, 100G-LC-C
AIS_Squelch_Configuration	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K	• DISABLE • SQUELCH	Read and write OTU PTP on AR-XP, AR-MXP, AR-XPE

Parameter	TP Type	NE Type	Valid Values	Supporting Equipment and Restrictions
		SONET, ONSNCS2K SDH		

Keep the following in mind:

- All parameters have the same function as defined for the Prime Optical Client Interface.
- All values are encoded as strings. Numeric values are also represented in string form.
- If the parameter value is an empty string, the parameter is not applicable for the current configuration.

The following table lists additional parameters for Active Protection Regeneration that are supported by OTU2 PTP.

Parameter	TP Type	NE Type	Valid Value	Supporting Equipment or Restrictions
EnableProactiveProtection	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Enable, Disable, EnablePreFECPSM	OTU2 PTP on AR-XP, AR-MXP, AR-XPE, OTU4 Trunk PTP on 100G-LC-C and 100G-CK-C. ✐ EnablePreFECPSM is only supported by OTU4 Trunk PTP.
RevertWindow	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Integer ms	OTU2 PTP on AR-XP, AR-MXP, AR-XPE
RevertThreshold	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	1E-4, 9E-5, 8E-5, 7E-5, 6E-5, 5E-5, 4E-5, 3E-5, 2E-5, 1E-5, 9E-6, 8E-6, 7E-6, 6E-6, 5E-6, 4E-6, 3E-6, 2E-6, 1E-6, 9E-7, 8E-7, 7E-7, 6E-7, 5E-7, 4E-7, 3E-7, 2E-7, 1E-7, 9E-8, 8E-8, 7E-8, 6E-8, 5E-8	OTU2 PTP on AR-XP, AR-MXP, AR-XPE OTU4 Trunk PTP on 100G-LC-C and 100G-CK-C
TriggerWindow	PTP	ONS15454 SONET,	Integer ms	OTU2 PTP on AR-XP,

Parameter	TP Type	NE Type	Valid Value	Supporting Equipment or Restrictions
		ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH		AR-MXP, AR-XPE OTU4 Trunk PTP on 100G-LC-C and 100G-CK-C
TriggerThreshold	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	1E-3, 9E-4, 8E-4, 7E-4, 6E-4, 5E-4, 4E-4, 3E-4, 2E-4, 1E-4, 9E-5, 8E-5, 7E-5, 6E-5, 5E-5, 4E-5, 3E-5, 2E-5, 1E-5, 9E-6, 7E-6, 6E-6, 5E-6, 4E-6, 3E-6, 2E-6, 1E-6, 9E-7, 8E-7, 7E-7, 6E-7, 5E-7, 4E-7, 3E-7, 2E-7, 1E-7	OTU2 PTP on AR-XP, AR-MXP, AR-XPE OTU4 Trunk PTP on 100G-LC-C and 100G-CK-C

Keep the following in mind:

- All parameters have the same function as defined for the Prime Optical Client Interface.
- All values are encoded as strings. Numeric values are also represented in string form.
- If the parameter value is an empty string, the parameter is not applicable for the current configuration.

The following table lists an additional parameter that is supported by AUTO PTP.

Parameter	TP Type	NE Type	Valid Value	Supporting Equipment and Restrictions
AutoSensing	PTP	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	On, Off	AUTO PTP on AR-XP, AR-MXP, AR-XPE

Keep the following in mind:

- All parameters have the same function as defined for the Prime Optical Client Interface.
- All values are encoded as strings. Numeric values are also represented in string form.
- If the parameter value is an empty string, the parameter is not applicable for the current configuration.

✐ If TrailTraceMonitor is off for a given CTP or FTP, Prime Optical does not return TrailTraceActualRx in transmissionParams.

- TPProtectionAssociation_T tpProtectionAssociation
  - Not supported.
- boolean edgePoint
  - Supported. For CTP and FTP, the value must be False.

- globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when equipmentName does not reference an equipment object.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.6.4 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 interface has the same behavior as the interface described in 3.6.3 getAllSupportedPTPs, but returns the object names instead of returning the entire object.

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 report in the first batch.
nameList	obaldefs::NamingAttributesList_T	Out	First batch of PTP names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining PTP names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when equipmentName does not reference an equipment object.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.6.5 getAllSupportingEquipment

Synopsis

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

Description
This interface enables an NMS to request the equipment that contains a given 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 a union of Equipment_T and EquipmentHolder_T structures. For details about both structures, see 3.6.1 getAllEquipment.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when ptpName does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when ptpName references a PTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.6.6 getAllSupportingEquipmentNames

Synopsis

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

Description
This interface has the same behavior as the operation described in 3.6.5 getAllSupportingEquipment, but returns the object name instead of returning the entire object.

Parameters

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributes_T	In	Name of the PTP for which to retrieve the supporting equipment.
nameList	globaldefs::NamingAttributesList_T	Out	List of equipment names (not equipment holders) directly implementing the PTP.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure. EXCPT_INVALID_INPUT - Raised when ptpName does not reference a PTP object. EXCPT_ENTITY_NOT_FOUND - Raised when ptpName references a PTP object that does not exist. EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost. EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.6.7 getContainedEquipment

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

Description
This interface returns the equipment and equipment holders that a specific equipment holder contains. This interface 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. Prime Optical 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 a union of Equipment_T and EquipmentHolder_T structures. For details about both structures, see 3.6.1 getAllEquipment.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when equipmentHolderName does not reference an equipment holder object.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentHolderName references an equipment holder that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.6.8 getEquipment

Synopsis

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

Description

getEquipment retrieves the equipment or equipment holder for a 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	Returned equipment or equipment holder.

The following table lists the additional information that is reported.

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
CLEICODE	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	As reported by Prime Optical client interface	—

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
ACTUAL_EQUIPMENT_ADMIN_STATE	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	As reported by Prime Optical client interface	—
ACTUAL_EQUIPMENT_SERVICE_STATE	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	As reported by Prime Optical client interface	—
PRODUCT_ID	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	As reported by Prime Optical client interface	—

The following table lists additional parameters that are supported.

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
TotalBandwidthUsed	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Total bandwidth used in GB	AR-XP, AR-MXP, AR-XPE
BandwidthUtilization	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Percentage of bandwidth supported by the card	AR-XP, AR-MXP, AR-XPE

The following table lists operational modes.

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
Card.Counter	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Any integer.	AR-XP, AR-MXP, AR-XPE
Card.OperationalModeNumber	ONS15454 SONET,	Operational mode associated to a	AR-XP, AR-

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
	ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	group of ports, sequential numbers.	MXP, AR-XPE
Card.CardMode	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	TXP_MR, TXPP_MR, MXP_DME, MXPP_DME, MXP_MR, MXPP_MR, MXP_4x25_10G, MXPP_4x25_10G, MXP_VD_10G, RGN, FANOUT-10x10G	AR-XP, AR-MXP, AR-XPE
Card.Rate	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Low rate, High rate	AR-XP, AR-MXP, AR-XPE
Card.ClientTrunkMapping	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	<nr client port>:<nr trunk port>	AR-XP, AR-MXP, AR-XPE
Card.ClientPorts	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	String of client ports; comma-separated list of number groups. The numbers in each group are hyphenated.	AR-XP, AR-MXP, AR-XPE
Card.TrunkPorts	ONS15454 SONET, ONS15454 SDH	String of trunk ports; comma-separated list of number groups. The numbers in each group are hyphenated.	AR-XP, AR-MXP, AR-XPE
Card.BandwidthUsed	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Comma-seperated list of bandwidths, in gigabytes, used by the operational mode.	AR-XP, AR-MXP, AR-XPE
Card. ODUUtilization	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	List of ODU Utilization Matrix comma separated. ODUUtilization_<ID> where ID represents the related operational mode number.	AR-XP, AR-MXP,AR-XPE

The following table lists the ODU time slot utilization for the card operational modes described in the previous table.

Parameter	NE Type	Valid Values	Supporting Equipment and Restrictions
ODUUtilization_<Id>.Counter	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Number of rows in the ODUUtilization Matrix; any integer.	AR-XP, AR-MXP, AR-XPE
ODUUtilization_<Id>. ID	ONS 15454 SONET, ONS 15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Sequential list of ODU time slots associated with the card comma separated.	AR-XP, AR-MXP, AR-XPE
ODUUtilization_<Id>.TimeSlots	ONS 15454 SONET, ONS 15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	List of used time slots comma separated. Each position corresponds to the number of time slots. If the value is 0, the time slot is free. If the value is not 0, the number of associated ports, when used, is reported; for example, 4-4-4-4-4-0-0-0-0-0-0-0-0-0-0;	AR-XP, AR-MXP, AR-XPE

Example Card Table

```
Card.Count=3
Card.OperationalModeNumber=1,2,3
Card.CardMode=MXP_MR,TXP_MR,RGN
Card.Rate=HighRate,LowRate,LowRate
Card.ClientTrunkMapping=N:1,1:1,1:1
Card.ClientPorts=1-2-3-4,5,7
Card.TrunkPorts=9,6,8
Card.BandwidthUsed=10.0G,0.0G,5.0G
Card. ODUUtilization =ODUUtilization_1,
```

The card configuration in this example supports only one ODUUtilization Matrix related to an OperationalModeNumber with a value of 1.. The others do not support the ODU.
The ODU Matrix is:

```
ODUUtilization_1.Counter=4
ODUUtilization_1 .ID=1,2,3,4
ODUUtilization_1 .TimeSlots=1-1-1-1-3-3-3-3-3-3-3-3-4-4-0-0, 5-5-5-5-5-5-5-5-
5-5-5-5-5-5-5-5, 2-2-2-2-2-2-2-2-2-2-2-2-2-2-0-0, 0-0-0-0-0-0-0-0-0-0-0-0-0-0-
0-0
```

The preceding data represents in TMF format the following matrix.

Operational ModeNumber	CardMode	Rate	ClientTrunk Mapping	Client Ports	Trunk Ports	BandwidthUsed	ODUUtilization
1	TXP-10G	High Rate	N:1	1-2-3-4	9	10.0G	ODUUtilization_1
2	MXP_MR	LowRate	1:1	5	6	0.0G	Empty
3	RGN	LowRate	1:1	7	8	5.0G	Empty

The following table lists the time slot for each ID.

ID	Time Slot
1	1-1-1-1-3-3-3-3-3-3-3-3-4-4-0-0
2	5-5-5-5-5-5-5-5-5-5-5-5-5-5-5-5
3	2-2-2-2-2-2-2-2-2-2-2-2-2-2-0-0
4	0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0

Additional information for Virtual ODUk Object/Client port mapping is supported by AR_XPE cards. In all the operational modes (except MXP-MR-LOW/HIGH), whenever an ODU0-mapped payload (GE/1G FC) is created, a corresponding Virtual ODUk Object is created. In the MXP-MR-LOW/HIGH modes, whenever an OCHCC circuit is created on an ODU0-mapped payload (GE/1G FC), a corresponding Virtual ODUk Object is created.

Parameter	NE Type	Valid Value	Supporting Equipment or Restriction
ODUInterface.Count	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Any integer	AR_XPE
ODUInterface.ODUObjects	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Comma-separated list of all ODUk(k=0) AIDs used to raise the ODU0 level alarms. Each AID is in the form ODU0[-SH]-SL-PPM-PORT-ODU1-ODU0.	AR_XPE
ODUInterface.TrunkPorts	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K	Comma-separated list of all physical trunk port details that map to the virtual ODUk(k=0) port/object.	AR_XPE

Parameter	NE Type	Valid Value	Supporting Equipment or Restriction
	SDH		
ODUInterface.ODU1	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Comma-separated list of all ODU1 frame numbers within the ODU2 frame that contains the corresponding ODUk(k=0) frame.	AR_XPE
ODUInterface.ODU0	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Comma-separated list of all ODUk(k=0) frame numbers within an ODU1 frame.	AR_XPE
ODUInterface.ClientPorts	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Comma-separated list of all physical client port numbers that map to the virtual ODUk(k=0) port/object.	AR_XPE

Example Virtual ODUk Object Interface Table:

```
ODUInterface.Count=2
ODUInterface.ODUObjects=ODU0-1-3-5-1-1-1, ODU0-1-3-5-1-1-2
ODUInterface.TrunkPorts=5,5
ODUInterface.ODU1=1,1
ODUInterface.ODU0=1,2
ODUInterface.ClientPorts=3,4
```

The following table lists the additional attributes supported for 10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C and WSE cards.

Parameter	NE Type	Valid Value	Supporting Equipment or Restrictions
Card.Counter	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Any integer	10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C, WSE
Card.CardMode	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	TXP_MR, TXPP_MR, MXP_DME, MXPP_DME, MXP_MR, MXPP_MR,	10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C, WSE

Parameter	NE Type	Valid Value	Supporting Equipment or Restrictions
		MXP_4x25_10G, MXPP_4x25_10G, MXP_VD_10G, RGN	
Card.ClientPorts	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	List of client ports, string, list of numbers comma-separated	10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C, WSE
Card.TrunkPorts	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	List of trunk ports, string, list of numbers comma-separated	10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C, WSE
Card.PeerList	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	Cards peered with the current card	10x10G_LC, M-CFP-LC, 100G-LC-C, 100G-CK-C , WSE

Each additional information table is characterized by a sequence of attributes. The information on all of the card classes is thus modeled as a table. As per the TMF standard, tables must contain the following:

- An attribute (Card.Count) reporting the number of rows in the table.
- An attribute consisting of a string of comma-separated values for every column in the table. The number of comma-separated values in each parameter must be equal to the Card.Count value.

For example:

```
Card.Count=4;
Card.ClientPorts=2,4,6,8;
Card.CardMode=TXP-10G,TXP-10G,TXP-10G,TXP-10G;
Card.TrunkPorts=1,3,5,7;
Card.PeerList=N/A,N/A,N/A,N/A
```

The following table lists the additional attributes supported by the shelf equipment holder.

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
SERIAL_NUMBER	CPT 200 SONET, CPT 200 SDH, CPT 600 SONET, CPT 600 SDH, ONS 15305 CTC, ONS 15310 CL, ONS 15310 MA, ONS 15310 MA SDH, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, ONS NCS2K SDH, ONS 15600 SONET, ONS 15600SDH	Serial number or empty string	Shelf equipment holder only
VOLTAGE	ONS 15454 SONET, ONS 15454 SDH,ONS NCS2K SONET, ONS NCS2K SDH	Voltage or empty string	Shelf equipment holder only. Available for getEquipment only.
TEMPERATURE	ONS 15454 SONET, ONS 15454 SDH,ONS NCS2K SONET, ONS NCS2K SDH	Temperature or empty string	Shelf equipment holder only. Available for getEquipment only.

Keep the following in mind:

- All parameters have the same function as defined for the Prime Optical Client Interface.
- All values are encoded as strings. Numeric values are also represented in string form.
- If the parameter value is an emptry string, the parameter is not applicable for the current configuration.

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 the managed element is lost.

Compliance

TMF-defined.

3.6.9 provisionEquipment

Synopsis

```
void provisionEquipment(
        in EQTCreateData_T equipmentCreateData,
        out Equipment_T createdEquipment)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the NMS to permanently provision a piece of equipment in an equipment holder in an ME. The result of this operation is the creation of the equipment object and all of its related objects, such as TPs. The resulting object is returned.

The equipment holder that contains the equipment must exist. The equipment object can already exist, but the equipment holder must not have expected equipment already provisioned.

If the equipment is provisioned successfully, Prime Optical generates a state change event on the holder state change and an object creation event for the equipment, and pushes them to the event channel.

It is important to note that the ONS 15454 SONET and ONS NCS2K SONET detects new equipment and provisions it automatically, provided that the equipment holder (slot) is empty and is not preprovisioned for another module type.

The equipment must be from the acceptableEquipmentTypeList of the slots. If these slots are selected to provision any other type of card, Prime Optical returns the EXCPT_UNABLE_TO_COMPLY exception.

PPMs and PIMs can also be provisioned using this interface.

It is not necessary to know the subtypes of the OC-n card at the time of provisioning. When equipment is actually installed, Prime Optical reports the subtype in the name field of the equipment.

For the ONS 15600 SONET and ONS 15600 SDH, each CXC or SSXC card occupies two slots (slots 6 and 7 or slots 8 and 9), and the NMS can specify any one of the slots to provision a CXC or SSXC card.

Parameters

Name	Type	Input/ Output	Description
equipmentCreateData	EQTCreateData_T	In	Data that describes the equipment to create.
createdEquipment	Equipment_T	Out	Resulting created equipment.

Prime Optical support for EQTCreateData_T structure is as follows:

- string userLabel
  - Not supported.
- boolean forceUniqueness
  - Not supported.
- string owner
  - Not supported.
- EquipmentObjectType_T expectedEquipmentObjectType
  - This parameter must take one of the values for equipment described in 3.6.1 getAllEquipment.
- globaldefs::NamingAttributes_T equipmentHolderName
  - Supported.
- globaldefs::NVSList_T additionalInfo
  - Supported only for client port on 100G-CK-C card provisioned in TXP-100G mode.

Parameter	NE Type	Valid Value	Supporting Equipment and Restrictions
nrOfLanes	ONS15454 SONET, ONS15454 SDH, ONSNCS2K SONET, ONSNCS2K SDH	4 or 10, integer value.	100G-CK-C card provisioned in TXP-100G mode, write only parameter

This interface is not applicable for configuring payloads for client ports in TXP_MR_10G.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_OBJECT_IN_USE - Raised when the equipment holder already has an expected equipment.

EXCPT_INVALID_INPUT - Raised when equipmentHolder does not reference an equipmentHolder object.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentHolder references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the equipment cannot be created at the NE.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.6.10 setAlarmReportingOff

Synopsis

```
void setAlarmReportingOff(
        in globaldefs::NamingAttributes_T equipmentOrHolderName)
```

```
raises(globaldefs::ProcessingFailureException);
```

Description
This interface disables alarm reporting on a piece of equipment. If successful, Prime Optical sends a warning alarm notification. For optical NEs, the interface supports the node names as equipment only when the NE alarms are configured with "off."

Parameters

Name	Type	Input/ Output	Description
equipmentOrHolderName	Globaldefs::NamingAttributes_T	In	Equipment on which to deactivate alarm reporting.

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 holder.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentOrHolderName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when alarm reporting cannot be disabled for the equipment or holder.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.6.11 setAlarmReportingOn

Synopsis

```
void setAlarmReportingOn(
        in globaldefs::NamingAttributes_T equipmentOrHolderName)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables alarm reporting on a piece of equipment.
The default setting for alarmReporting is “on.” If the alarmReporting status is off, it indicates that a warning alarm has been raised for it. Prime Optical generates a cleared warning alarm if successful. For optical NEs, the interface supports the node names as equipment only when the NE alarms are configured with "off."

Parameters

Name	Type	Input/Output	Description
equipmentOrHolderName	Globaldefs::NamingAttributes_T	In	Equipment on which to activate alarm reporting.

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 holder.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentOrHolderName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when alarm reporting cannot be disabled for the equipment or holder.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.6.12 unprovisionEquipment

Synopsis

```
void unprovisionEquipment(
        in globaldefs::NamingAttributes_T equipmentName)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the NMS to permanently unprovision equipment from the managed element. The successful result of this operation is the potential deletion of the equipment object and all of its related objects, such as termination points.
The ONS 15454 SONET and, ONS NCS2000 SONET does not allow a user to unprovision a piece of equipment that is in use (SNC is created on this equipment). However, if the equipment is present in the slot and is not in use, it can be unprovisioned. If the equipment is not physically removed from the slot and new equipment is not provisioned in the same slot, the NE redetects the equipment in approximately 1 minute. If the equipment is not physically removed from the slot and new equipment of a different type is provisioned in the same slot, the NE generates an equipment mismatch alarm.
If the equipment is unprovisioned successfully, Prime Optical generates a state change event on the holder state change and an object deletion event for the equipment, and pushes them to the event channel.
The ONS 15454 reserves slots 7 and 11 for TCC. One of these slots has an in-service TCC card. If the NMS invokes unprovisionEquipment on this equipment, Prime Optical throws the EXCPT_UNABLE_TO_COMPLY exception.
In case of NCS 2000 SONET and NCS 2000 SDH, when unprovisionEquipment is invoked on slot 18 of M15 chassis on ECU equipment, Prime optical throws the EXCPT_UNABLE_TO_COMPLY exception.

Parameters

Name	Type	Input/Output	Description
equipmentName	globaldefs::NamingAttributes_T	In	Name of the equipment to unprovision.

This interface is not applicable for configuring payloads for client ports in TXP_MR_10G.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when equipmentName does not reference an equipment object.

EXCPT_OBJECT_IN_USE - Deletion is rejected because equipment resources are in use.

EXCPT_ENTITY_NOT_FOUND - Raised when equipmentName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the equipment cannot be unprovisioned at the NE.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.7 maintenanceOps::MaintenanceMgr_I

MaintenanceMgr_I is used as a handle to gain access to the maintenance operation. A handle to an instance of this interface is gained through the getManager operation in emsSession.
interface MaintenanceMgr_I:common::Common_I

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

3.7.1 getActiveMaintenanceOperations

Synopsis

```
void getActiveMaintenanceOperations(
        in globaldefs::NamingAttributes_T tpOrMeName,
        in unsigned long how_many,
        out CurrentMaintenanceOperationList_T
            currentMaintenanceOpeationList,
        out CurrentMaintenanceOperationIterator_I cmoIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the NMS to query the EMS to determine if any persistent maintenance commands have been invoked. This query is supported for PTP, CTP, and ME objects.

Parameters

Name	Type	Input/ Output	Description
tpOrMeName	globaldefs::NamingAttributes_T	In	Name of the PTP, CTP, or ME.
how_many	unsigned long	In	Maximum number of maintenance operations to return in the first batch.
rrentMaintenanceOpeationList	CurrentMaintenanceOperationList_T	Out	First batch of maintenance operations.
cmoIt	CurrentMaintenanceOperationIterator_I	Out	Iterator to access the remaining maintenance operations.

Prime Optical support for CurrentMaintenanceOperation structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - Supported.
- MaintenanceOperation_T maintenanceOperation
  - Prime Optical supports FACILITY_LOOPBACK, TERMINAL_LOOPBACK, and
- PAYLOAD_LOOPBACK.
- transmissionParameters::LayerRate_T layerRate
  - Supported for CTPs only.
- globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpOrMeName does not reference a valid object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpOrMeName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.7.2 performMaintenanceOperation

Synopsis

```
void performMaintenanceOperation(
        in CurrentMaintenanceOperation_T maintenanceOperation,
        in MaintenanceOperationMode_T maintenanceOperationMode)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the NMS to operate and release the maintenance commands that are supported by a TP.

ONS 15310 CL and ONS 15310 MA

Prime Optical supports only terminal and facility loopback operations for electrical and OC-n PTPs.
Prime Optical does not support loopback operations for CTPs.

ONS 15327

Prime Optical supports only terminal and facility loopback operations for DS-n and OC-n PTPs.
Prime Optical does not support loopback operations for CTPs.
For the ONS 15327 R3.4 or later, the loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.

ONS 15454 SDH

Prime Optical supports only terminal and facility loopback operations for PTPs.
Prime Optical does not support loopback operations for CTPs.
For the ONS 15454 SDH R3.4 or later, the loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.

ONS NCS2000 SDH
Prime Optical supports only terminal and facility loopback operations for PTPs.
Prime Optical does not support loopback operations for CTPs.
The loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.
ONS 15454 SONET
Prime Optical supports only terminal and facility loopback operations for DS-n, OC-n, EC1, 10G-transponder, 10G-muxponder, WSE and G1000_4 PTPs.
Only facility loopback is supported for DS1 CTPs on DS3XM cards.
Loopbacks will not be supported on encrypted trunk ports of WSE card.
For the ONS 15454 SONET R3.4 or later, the loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.
ONS NCS2K SONET
Prime Optical supports only terminal and facility loopback operations for DS-n, OC-n, EC1, 10G-transponder, 10G-muxponder, WSE and G1000_4 PTPs.
Only facility loopback is supported for DS1 CTPs on DS3XM cards.
Loopbacks will not be supported on encrypted trunk ports of WSE card.
The loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.

ONS 15600 SDH
Prime Optical supports only terminal and facility loopback operations for PTPs.
Prime Optical does not support loopback operations for CTPs.
Loopback is supported only in the Out of Service–Maintenance state of the PTP.
Prime Optical ignores the MaintenanceOperation_T value in CurrentMaintenanceOperation_T structure, if MaintenanceOperationMode_T is set to MOM_RELEASE.
ONS 15600 SONET
Prime Optical supports only facility and payload loopback operations for PTPs.
Prime Optical does not support loopback operations for CTPs.
Loopback is supported only in the Out of Service–Maintenance state of the PTP.
Prime Optical ignores the MaintenanceOperation_T value in CurrentMaintenanceOperation_T structure, if MaintenanceOperationMode_T is set to MOM_RELEASE.
CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH
Prime Optical supports only terminal and facility loopback operations for PTPs.
The loopback state can be set only if the TP state is either Out of Service–Maintenance or Auto in Service.

Parameters

Name	Type	Input/ Output	Description
maintenanceOperation	CurrentMaintenanceOperation_T	In	Information on the maintenance operation to perform. This structure contains the field Maintenance operation. Valid values for this field are: • FACILITY_LOOPBACK • TERMINAL_LOOPBACK • PAYLOAD_LOOPBACK
maintenanceOperation	MaintenanceOperationMode_	In	Whether the maintenance

Name	Type	Input/ Output	Description
Mode	T		operation is completed or released.

Prime Optical support for CurrentMaintenanceOperation_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - Supported.
- MaintenanceOperation_T maintenanceOperation
  - Only FACILITY_LOOPBACK, TERMINAL_LOOPBACK, and PAYLOAD_LOOPBACK are supported.
- transmissionParameters::LayerRate_T layerRate
  - Supported for CTPs only.
- globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a TP.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the operation is denied by the ME.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.8 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.

interface ManagedElementManager_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.

3.8.1 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 interface enables an NMS to request all active alarms for the specified ME, excluding alarms that the ME does not report to Prime Optical. 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.
ONS 15310 CL, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS 15600 SONET, ONS 15600 SDH, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SONET and ONS NCS2K SDH
The ONS 15310 CL, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS 15600 SONET, ONS 15600 SDH CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SONET, and ONS NCS2K SDH do not send a "clear" for TCA events. TCAs are not reported in the eventList that is returned by this method.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the ME for which to retrieve alarms.
excludeProbCauseList	notifications::ProbableCauseList_T	In	List of probable causes to exclude (for which events are not 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 - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when meName does not reference a managed element object or excludeProbCauseList contains undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.8.2 getAllSNCs

Synopsis
```
void getAllSNCs(
        in globaldefs::NamingAttributes_T managedElementName,
        in transmissionParameters::LayerRateList_T connectionRateList,
```

```
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T sncList,
        out subnetworkConnection::SNCIterator_I sncIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request all SNCs that originate on, terminate on, or transit through a given ME.

- For the ONS 15310 CL, this method does not report STS-1 SNC on CTX using DS1 ports.
- For the ONS 15310 MA SONET, this method does not report STS-1 SNC on DS1 ports of DS1_xx cards.
- For the ONS 15310 MA SDH, this method does not report VC4 SNC on E1 ports of E1_63_E3_3 cards.
- For the ONS 15327, this method does not report STS-1 SNC on XTC using DS1 ports.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.
- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, Using GateWay/CORBA Connectionless Interfaces for the supported SNCs.

This interface does not report VCAT SNCs.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME whose SNCs are required.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the SNCs to report. If an empty list is specified, all SNCs of all rates are reported.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
sncList	subnetworkConnection::SubnetworkConnectionList_T	Out	First batch of SNCs.
sncIt	subnetworkConnection::SNCIterator_I	Out	Iterator to retrieve the remaining SNCs.

Prime Optical support for SubnetworkConnection_T structure, for more details refer to section 2.2.7.7 SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a managed element.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
Prime Optical-defined.

3.8.3 getAllManagedElements

Synopsis

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

Description
This interface enables 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.

Prime Optical support for ManagedElement_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. The name is reported by the NE.
- string owner
  - Not supported.
- string location
  - Supported.
- string version
  - Supported.
- string productName
  - Supported.
- CommunicationState_T communicationState
  - Supported.
- boolean emsInSyncState
  - Supported.
- transmissionParameters::LayerRateList_T supportedRates
  - For a list of all supported layer rates, see 2.2.1 Layer Rate.
- globaldefs::NVSList_T additionalInfo
  Supported. Three parameters are sent in this field:
  - A name/value pair for the IP address with the name “IPAddress.”

– A name/value pair for the operational state with the name “OperationalState,” containing the following values. The emsInSyncState value is determined by the OperationalState value. A state change event is generated whenever the state of emsInSyncState changes.

OperationStates	EmsInSyncState
PreProvisioned	False
InServiceSyncConfiguration	False
OutOfService	False
UnderMaintenance	False
InServiceIntializing	False
InService	True

– A name/value pair for the vendor with the name “VendorName.”

In addtion to the preceding, the following parameters are available for a restricted group of MEs:

- A name/value pair for the ME Description field with the name “Description.” It is an editable, free text field. When modified, AVC notifications are not sent to GateWay/CORBA clients.
- A name/value pair for the ME Latitude field with the name “Latitude.” It is an editable field in the format “North/South <num> Deg <num> Min” (for example, “North 38 Deg 13 Min”). When modified, AVC notifications are not sent to GateWay/CORBA clients.
- A name/value pair for the ME Longitude field with the name “Longitude.” It is an editable field in the format “West/East <num> Deg <num> Min” (for example, “West 122 Deg 38 Min”). When modified, AVC notifications are not sent to GateWay/CORBA clients.

The following table lists the MEs that support these additional parameters:

Attribute Name	Supporting ME
Description Latitude Longitude	15454
	15454 SDH
✎ These attributes are read-only in GateWay/CORBA. You must use the Prime Optical client to edit them.	CPT 200
	NCS2K SONET
	NCS2K SDH
	CPT 200 SDH
	CPT 600
	CPT 600 SDH
	ONS 15305 CTC
	ONS 15310 CL
	ONS 15310 MA
	ONS 15310 MA SDH
	ONS 15327
	ONS 15600
	ONS 15600 SDH

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.8.4 getAllManagedElementNames

Synopsis

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

Description
This interface has the same behavior as 3.8.3 getAllManagedElements, but instead of returning 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 - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.8.5 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 interface 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. If no PTPs match the layer constraints, an empty list is returned. A PTP is returned regardless of connectivity to other MEs or position in the subnetwork. (Both PTPs at the subnetwork edge and PTPs that are internal to the subnetwork are reported.)

For PTPs on protection equipment in a 1_FOR_N protection group, Prime Optical reports only the AlarmReporting for the LayeredParameters of the PTPs.

For a list of layered parameters returned for each layer, see "Layered Parameters and Layer Rates".

Layered parameters are not returned for passive units. MS-ISC cards do not have layered parameters.

For Precision Time Protocol (PTP), the additionalInfo attribute list contains the dataRate attribute value. For more information, see the "11 PTP dataRate Attribute" chapter.

The OTL lines/internal ports of CXP Port are also listed along with the main PTPs of CXP port in the 100G_LC cards, and they do not have any layered parameters.

This interface also returns a list of front Ethernet ports on an ML-series card on the specified managed element.

- ONS 15216: Prime Optical returns all the supported PTPs for ONS 15216.
- ONS 15305 and ONS 15305 CTC: Prime Optical returns WAN, SDH, and PDH ports.
- ONS 15310 CL, ONS 15310 MA SDH, and ONS 15310 MA SONET: Prime Optical returns OC-n, DS1, DS3, and Ethernet ports.
- ONS 15327: Prime Optical returns OC-n, DS1, DS3, and Ethernet ports.
- ONS 15454 SONET: Prime Optical returns OC-n, DS1, DS3, EC-1, fibre channel, and Ethernet ports.
- ONS 15454 SDH: Prime Optical returns STM-n (OC-n), E1, E3, DS3I, STM1E, fibre channel, and Ethernet ports.
- ONS 15600 SONET: Prime Optical returns OC-n ports.
- ONS 15600 SDH: Prime Optical returns STM-n ports.
- Unmanaged NE: Prime Optical returns PTPs only when a topological link exists, and returns only those PTPs that participate in the topological link.
- CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH: Prime Optical returns OTN and Ethernet ports.

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	List of PTP layer rates for which the PTPs are retrieved. To be reported, a PTP must contain at least one of the layer rates specified. If the list is empty, all PTPs (of all rates) are returned.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	Prime Optical does

Name	Type	Input/ Output	Description
			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 details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.
For a list of all supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.8.6 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 interface has the same behavior as the interface described in 3.8.5 getAllPTPs but returns the object names instead of the object structures.

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	List of PTP layer rates for which PTPs are

Name	Type	Input/ Output	Description
			retrieved. To be reported, a PTP must contain at least one of the specified layer rates. If the list is empty, all PTPs (of all rates) are returned.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	Prime Optical 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 - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.8.7 getContainedInUseTPs

Synopsis

```
void getContainedInUseTPs(
        in globaldefs::NamingAttributes_T tpName,
        in transmissionParameters::LayerRateList_T layerRateList,
        in unsigned long how_many,
        out terminationPoint::TerminationPointList_T tpList,
        out terminationPoint::TerminationPointIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface retrieves the in-use CTPs contained in a specific PTP, at specific layer rates. An in-use CTP is a CTP used by an SNC in any state (including pending) or a CTP that is terminated and mapped. Example for SONET: The NMS invokes this method on an OC3 PTP (and assumes that the NE does not have an XCVT card, so VT-level cross-connects are not possible) with an empty layerRateList.

- If there are no SNCs created, Prime Optical returns zero CTPs.

- If there is one SNC created at the STS-1 level, Prime Optical returns one CTP at the STS-1 layer rate.

Example for SDH: The NMS invokes this method on an STM-1 PTP (and assumes that the NE does not have an XC10G card, so VC12-level cross-connects are not possible) with an empty layerRateList.

- If there are no SNCs created, Prime Optical returns zero CTPs.
- If there is one SNC created at the VC4 level, Prime Optical returns one CTP at the VC4 layer rate.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the PTP.
layerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the contained actual CTPs to return. An empty list tells the EMS to retrieve all contained actual CTPs (of all rates).
how_many	unsigned long	In	Maximum number of CTPs to return in the first batch.
tpList	terminationPoint::TerminationPointList_T	Out	First batch of contained in-use CTPs.
tpIt	terminationPoint::TerminationPointIterator_I	Out	Iterator to retrieve the remaining contained in-use CTPs.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.
For a list of all supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a PTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

This API is not supported for MSTP OCHCC and OCHTrail layers.

3.8.8 getContainedInUseTPNames

Synopsis

```
void getContainedInUseTPNames(
        in globaldefs::NamingAttributes_T tpName,
        in transmissionParameters::LayerRateList_T layerRateList,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
```

```
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface has the same behavior as the operation described in 3.8.7 getContainedInUseTPs, but instead of returning entire object structures, this operation returns their names.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the PTP.
layerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the contained actual CTPs to return. An empty list indicates to the EMS to get all contained actual CTPs (for all rates).
how_many	unsigned long	In	Maximum number of CTPs to return in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of contained in-use CTP names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining contained in-use CTP names.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a PTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.8.9 getContainedPotentialTPs

Synopsis

```
void getContainedPotentialTPs(
        in globaldefs::NamingAttributes_T tpName,
        in transmissionParameters::LayerRateList_T layerRateList,
        in unsigned long how_many,
        out terminationPoint::TerminationPointList_T tpList,
        out terminationPoint::TerminationPointIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables an NMS to request all of the CTPs it can support in all possible mapping configurations, at specified rates and TPs. This interface returns all potential contained CTPs for a specific TP. The TP might be a PTP or a CTP. If the layerRateList is empty, contained CTPs at all flexible and static LayerRates are returned.

For example, on an ONS 15454 SONET NE, the NMS invokes this method on an OC-3 PTP (assuming that the NE does not have an XCVT card, so VT-level cross-connects are not possible) with an empty layerRateList. Regardless of the number of SNCs, Prime Optical returns four CTPs: three at the STS-1 layer rate and one at the STS3c layer rate.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the PTP or CTP for which to retrieve contained CTPs.
layerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the contained CTPs to report. An empty list tells the EMS to report all contained CTPs (of all rates).
how_many	unsigned long	In	Maximum number of contained CTPs to report in the first batch.
tpList	terminationPoint::TerminationPointList_T	Out	First batch of contained in-use CTPs.
tpIt	terminationPoint::TerminationPointIterator_I	Out	Iterator to retrieve the remaining contained in-use CTPs.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.
For a list of all supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a PTP or CTP object.

EXCPT_TP_INVALID_ENDPOINT - Raised when tpName refers to a PTP (and all contained potential CTPs) that is part of a topological link.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a PTP or CTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.8.10 getContainedPotentialTPNames

Synopsis

```
void getContainedPotentialTPNames(
in globaldefs::NamingAttributes_T tpName,
in transmissionParameters::LayerRateList_T layerRateList,
in unsigned long how_many,
out globaldefs::NamingAttributesList_T nameList,
out globaldefs::NamingAttributesIterator_I nameIt)
```

```
raises(globaldefs::ProcessingFailureException);
```

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the PTP or CTP for which to retrieve contained CTPs.
layerRateList	transmissionParameters::LayerRateList_T	In	List of the rates of the contained CTPs to report. An empty list indicates to the EMS to report all contained CTPs (of all rates).
how_many	unsigned long	In	Maximum number of contained CTPs to report in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of contained in-use CTP names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining contained in-use CTP names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a PTP or CTP object.

EXCPT_TP_INVALID_ENDPOINT - Raised when tpName refers to a PTP (and all contained potential CTPs) that is part of a topological link.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a PTP or CTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.8.11 getContainingSubnetworkNames

Synopsis

```
void getContainingSubnetworkNames(
        in globaldefs::NamingAttributes_T managedElementName,
        out globaldefs::NamingAttributesList_T subnetNames)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of subnetwork names to which the ME (supplied as an input parameter) belongs.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME for which to retrieve the containing subnetwork names.
subnetNames	globaldefs::NamingAttributesList_T	Out	Names of the subnetworks to

			which the NE belongs.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.8.12 getManagedElement

Synopsis

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

Description

This interface returns the managed element for a specific managed element name.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME to retrieve.
me	managedElement::ManagedElement_T	Out	Retrieved ME.

For details about the ManagedElement_T structure, see 3.8.3 getAllManagedElements.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.8.13 getAllCTMMEInternalPatchcord

Synopsis

```
void getAllCtmMEInternalPatchcord(
        in globaldefs::NamingAttributes_T managedElementName,
        out globaldefs::NVSList_T patches)
```

```
raises (globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of internal Prime Optical patchcords for a specific managed element name. For PTP, the additionalInfo attribute list contains the OTU2 protection regeneration attributes value if the selected card is one of the following:

- AR-XP (ONS 15454 SONET/SDH, ONS NCS2000 SONET/SDH)
- AR-MXP (ONS 15454 SONET/SDH, ONS NCS2000 SONET/SDH))

For PTP, the additionalInfo attribute list contains the AUTO attributes value if the selected card is one of the following:

- AR-XP (ONS 15454 SONET/SDH, ONS NCS2000 SONET/SDH))
- AR-MXP (ONS 15454 SONET/SDH, ONS NCS2000 SONET/SDH))

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME to retrieve.
patches	globaldefs::NVSList_T	Out	The retrieved patchcords.

For details about the ManagedElement_T structure, see 3.8.3 getAllManagedElements.

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when managedElementName does not reference a
managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME
object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.8.14 setAdditionalInfo

Synopsis

```
void setAdditionalInfo(

in globaldefs::NamingAttributes_T tpName,

      inout globaldefs::NVSList_T  additionalInfo)

    raises(globaldefs::ProcessingFailureException);
```

Description

This interface sets attributes using the additionalInfo structure. This interface is supported only for:

- Physical termination point (PTP)
- Cards and NE types listed in the following table

Card	ONS 15310 CL	ONS 15310 MA SONET	ONS 15310 MA SDH	ONS 15454 SONET	ONS 15454 SDH
G1000_4				X	X
CE-1000-4				X	X

Card	ONS 15310 CL	ONS 15310 MA SONET	ONS 15310 MA SDH	ONS 15454 SONET	ONS 15454 SDH
CE-100T-8	X	X	X	X	X
CE-MR-10				X	X
CE-MR-6		X	X		

This interface is also supported for the attribute LinkIntegrityTimer for the cards and the NE types listed in the following table.

Card	ONS 15310 CL	ONS 15310 MA SONET	ONS 15310 MA SDH	ONS 15454 SONET	ONS 15454 SDH
CE-100T-8	X	X	X	X	X
CE-MR-10				X	X
CE-MR-6		X	X		

This interface is also supported for the attribute LinkIntegrityTimer for the cards and the NE types listed in the following table.

Card	ONS 15454 SONET	ONS 15454 SDH
G1000_4	X	X
CE-1000-4	X	X

For CE-100T-8, CE-MR-6, and CE-MR-10 cards, the LinkIntegrityTimer attribute value cannot be set to 0.
For G1000_4 and CE-1000-4 cards, the LinkIntegrityTimer attribute value cannot be set to 0 if the LinkIntegrityDisable attribute is set to false.
For G1000_4 and CE-1000-4 cards, the LinkIntegrityTimer attribute value must be set to 0 if the LinkIntegrityDisable attribute is set to true.
For G1000_4 and CE-1000-4 cards, the LinkIntegrityDisable attribute can be set only to true or false.
For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, additional connectionless TP information can be set. For more information, see Chapter 3, Using GateWay/CORBA Interfaces.
This operation returns a list of modified attributes with the new related values applied.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the TP to which to apply the list of modified attribute values.
additionalInfo	globaldefs::NVSList_T	In/out	List of attributes that must be modified, as well as the new related values to apply.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service, or if the service is not supported for the selected NE type or card.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a termination point object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a TP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.8.15 getTP

Synopsis

```
void getTP(
        in globaldefs::NamingAttributes_T tpName,
        out terminationPoint::TerminationPoint_T tp)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns the TP structure for a specific TP name.
For the PTP on protection equipment in a 1_FOR_N protection group, Prime Optical reports only the AlarmReporting for the LayeredParameters of that PTP.
For PTPs, the additionalInfo attribute list contains the LinkIntegrityTimer attribute value if the selected card is one of the following (see also 3.8.14 setAdditionalInfo):

- CE-100T-8 or CE-MR-6 on ONS 15310 MA SONET or ONS 15310 MA SDH
- CE-100T-8 on ONS 15310 CL
- CE-100T-8 or CE-MR-10 on ONS 15454 SONET or ONS 15454 SDH

For PTPs, the additionalInfo attribute list contains the LinkIntegrityTimer and LinkIntegrityDisable attribute values if the selected card is one of following:

- CE-1000-4 and G1000_4 on ONS 15454 SONET or ONS 15454 SDH

For Precision Time Protocol (PTP), the additionalInfo attribute list contains the dataRate attribute value. For more information, see the "11 PTP dataRate Attribute" chapter.
For PTPs, the layeredParameters and additionalInfo attribute lists contain the OTU1 and OTU2 attribute values if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2000 SONET and ONS NCS2000 SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2000 SONET and ONS NCS2000 SDH)

For PTPs, the additionalInfo attribute list contains the OTU2 protection regeneration attribute value if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2000 SONET and ONS NCS2000 SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2000 SONET and ONS NCS2000 SDH)

For PTPs, the additionalInfo attribute list contains the AUTO attribute value if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)

For more information on AR-XP and AR-MXP cards, see 3.6.3 getAllSupportedPTPs.
For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, additional connectionless TP information can be set. For more information, see Chapter 4, Using GateWay/CORBA Connectionless Interfaces.
For a list of layered parameters returned for each layer, see Layered Parameters and Layer Rates. Layered parameters are not returned for passive units. The MS-ISC card does not have layered parameters.

The OTL lines/internal ports of CXP Port on the 100G_LC cards do not have layered parameters.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of the TP to retrieve.
TP	terminationPoint::TerminationPoint_T	Out	Retrieved TP.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a termination point object, or the parameter is not correct.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a TP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.8.16 getAssociatedTopologicalLinks

Synopsis

```
void getAssociatedTopologicalLinks(
        in globaldefs::NamingAttributes_T ptpName,
        out topologicalLink::TopologicalLink_T topologicalLinks)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the NMS to pass the TP name. Prime Optical supports multilayer topological links and returns a list of topological links through the TP.

Prime Optical reports all the physical optical and manual topological links.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Name of TP whose topological link is required.
topologicalLinks	topologicalLink::TopologicalLink_T	Out	Topological link structure whose end is a specific TP.

For details about the TopologicalLink_T structure, see 3.3.14 getAllTopLevelTopologicalLinks.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when ptpName does not reference a physical termination point object.

EXCPT_ENTITY_NOT_FOUND - Raised when ptpName references a PTP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.8.17 setTPData

Synopsis

```
void setTPData(
in subnetworkConnection::TPData_T tpInfo,
out terminationPoint::TerminationPoint_T modifiedTP)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows the NMS to set parameters on a specific TP. The results tell the NMS which modifications succeeded.

Prime Optical supports the following parameters for CTPs:

- IPPMMonitor
- XCLoopBack
- TrailTraceActualTx
- TrailTraceExpectedRx
- TrailTraceMonitor

To completely remove the path trace, the NMS must set TrailTraceMonitor to Off and TrailTraceExpectedRx to an empty string in one call.

If the NMS sends only TrailTraceExpectedRx, Prime Optical sets TrailTraceMonitor to On and sets the value of the expected string. If TrailTraceExpectedRx contains an empty string as the only parameter in the call, Prime Optical changes TrailTraceMonitor to On and sets the expected string to an empty string.

To completely remove the path trace, the NMS must set TrailTraceMonitor to Off and set TrailTraceExpectedRx to an empty string in one call.

Prime Optical supports the following parameters for PTPs:

- AlarmReporting
- AdminState
- LineCode
- FrameFormat
- SDH_SONET_SS_BITS
- EnableSyncMsg
- TraceMode

The SDH_SONET_SS_BITS parameter can be modified only when EnableSyncMsg is Off.

EnableSyncMsg can be set to On only if SDH_SONET_SS_BITS is set to SONET.

The SDH_SONET_SS_BITS and EnableSyncMsg parameters must be modified by invoking two separate method calls. If you try to use the same method to modify SDH_SONET_SS_BITS and EnableSyncMsg, Prime Optical generates a processing failure exception.

The TraceMode parameter can be modified. The DisableFDIonTTIM parameter is disabled when TraceMode is set to MANUAL and enabled when TraceMode is MANUALNOAIS.

The setTPData interface does not apply to the following ML-series card parameters that are configurable through Cisco IOS:

- Admin state, MTU size, and COS accounting for POS and Ethernet interfaces.
- Speed, duplex, and flow control for Ethernet interfaces.

Prime Optical supports the following automatic laser shutdown parameters on PTPs. These parameters are not defined by the TMF.

- ALSMode

- PulseWidth
- RecoveryInterval
- RestartLaser

If the equipment is the protecting equipment in a 1_For_N protection group, Prime Optical sets the PTPs on the equipment to AlarmReporting.

Prime Optical supports only the following parameters for FTPs:

- TrailTraceExpectedRx
- TrailTraceMonitor

For valid values for each parameter, see Table 3-4 in 3.6.3 getAllSupportedPTPs.

For PTPs, the layeredParameters and additionalInfo attribute lists contain the OTU1 and OTU2 attribute values if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)

For PTPs, the additionalInfo attribute list contains the OTU2 protection regeneration attribute value if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)

For PTPs, the additionalInfo attribute list contains the AUTO attribute value if the selected card is one of the following:

- AR-XP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)
- AR-MXP (ONS15454 SONET and ONS15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH)

For more information on AR-XP and AR-MXP cards, see 3.6.3 getAllSupportedPTPs.

For CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, you can set the attributes the same as those of the ONS 15454 SONET and ONS 15454 SDH.

You must use the Prime Optical client to set other TP parameters.

Invoking this method from the NMS is a synchronous call.

AVC notifications are not sent when PTP attributes are modified by a user.

Each layer rate is associated to a specific list of layered parameters (see Layered Parameters and Layer Rates for more details).

When a layered parameter is modified using the setTPData API, the parameter name must be compliant with the referred layer rate, failing which the command is rejected. To ensure compliance, a check is applied to the following cards of ONS 15454 SONET and ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH:

- AR_XP
- AR_MXP
- AR_XPE
- TNC
- 100G_LC
- 10x10G_LC
- WSE
- M-CFP-LC

Parameters

Name	Type	Input/ Output	Description
tpInfo	subnetworkConnection::TPData_T	In	Details about required modifications.
modifiedTP	terminationPoint::TerminationPoint_T	Out	Result of modification.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.
Prime Optical support for TPData_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - Supported.
- terminationPoint::TerminationMode_T tpMappingMode
  - Not supported.
- transmissionParameters::LayeredParameterList_T transmissionParams
  - Supported.
- globaldefs::NamingAttributes_T ingressTrafficDescriptorName
  - Not supported.
- globaldefs::NamingAttributes_T egressTrafficDescriptorName
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the TP referred to in tpInfo does not exist.

EXCPT_INVALID_INPUT - Raised when tpInfo does not reference a valid CTP object, or a wrong parameter.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.8.18 getAllSrcPTPsForSNCProvisioning

Synopsis

```
void getAllSrcPTPsForSNCProvisioning(
        in globaldefs::NamingAttributes_T managedElementName,
        in transmissionParameters::LayerRate_T layerRate,
        in globaldefs::ConnectionDirection_T direction,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T tpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of PTP names that can be used as the source of an SNC to create on a given ME. The NMS must specify the layer rate and the direction of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns a PTP name list based on current resource availability. The actual SNC creation might still fail.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME that is the source of the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. All PTP names available as the source of the SNC at this layer rate are returned.
direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_UNI (unidirectional) or CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of PTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of PTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining PTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.8.19 getAllSrcCTPsForSNCProvisioning

Synopsis

```
void getAllSrcCTPsForSNCProvisioning(
        in globaldefs::NamingAttributes_T ptpName,
        in transmissionParameters::LayerRate_T layerRate,
        in globaldefs::ConnectionDirection_T direction,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T tpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of CTP names that can be used as the source of an SNC to create on a given ME. The NMS must specify the layer rate and the direction of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns a CTP name list based on current resource availability. The actual SNC creation might still fail. Values returned by this list can be used as aEnd in

the SNCCreateData_T parameter of the createAndActivateSNCFromUserLabel method described in 3.9.16 createAndActivateSNCFromUserLabel.

Parameters

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributes_T	In	Name of the PTP that is the source of the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. All CTP names available at this layer rate are returned.
direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_UNI (unidirectional) or CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of CTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of CTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining CTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.
EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.8.20 getAllSrcCTPsForVCATMemberSNCProvisioning

Synopsis

```
void getAllSrcCTPsForVCATMemberSNCProvisioning(
in globaldefs::NamingAttributes_T ptpName,
in transmissionParameters::LayerRate_T layerRate,
in globaldefs::ConnectionDirection_T direction,
in unsigned long how_many,
out globaldefs:: NamingAttributesList_T tpList,
out globaldefs:: NamingAttributesIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of CTP names that can be used as the source of a VCAT member SNC to create on a given PTP. The NMS must specify the layer rate and the direction of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns a CTP name list based on current resource availability. The actual SNC creation might still fail. Values returned by this list can be used as the aEnd in each SNCCreateData_T listed from the sncCreateDataList field of the type VCAT

SNCCreateData_T, which is required as a parameter of the createVCATl method described in 3.9.26 createVCAT.
This interface must be used according to 2.2.7.2.4 SNC Provisioning for ONS 15454 CE-MR-10 Cards.
Restrict the use of this API to VCAT circuits involving CE-MR-10 cards in Manual mode.

Parameters

See the Parameters section of 3.8.19 getAllSrcCTPsForSNCProvisioning.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.
EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.8.21 getAllDestPTPsForSNCProvisioning

Synopsis

```
void getAllDestPTPsForSNCProvisioning(
in globaldefs::NamingAttributes_T managedElementName,
in globaldefs::NamingAttributes_T srcCTPName,
in transmissionParameters::LayerRate_T layerRate,
in globaldefs::ConnectionDirection_T direction,
in unsigned long how_many,
out globaldefs:: NamingAttributesList_T tpList,
out globaldefs:: NamingAttributesIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of PTP names that can be used as the destination of an SNC to create on a given ME. The NMS must specify the source CTP name, layer rate, and the direction of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns a PTP name list based on current resource availability. The actual SNC creation might still fail.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the managed element that is the destination of the SNC to create.
srcCTPName	globaldefs::NamingAttributes_T	In	Name of the CTP that is the source of the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. This value must match the layer rate of the source CTP specified in

Name	Type	Input/ Output	Description
			the srcCTPName field. This parameter returns all PTP names that are available as the destination of the SNC at this layer rate.
Direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_UNI (unidirectional) or CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of PTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of PTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining PTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the PTP does not exist.

EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.8.22 getAllDestCTPsForSNCProvisioning

Synopsis

```
void getAllDestCTP namesForSNCProvisioning(
        in globaldefs::NamingAttributes_T ptpName,
        in globaldefs::NamingAttributes_T srcCTPName,
        in transmissionParameters::LayerRate_T layerRate,
        in globaldefs::ConnectionDirection_T direction,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T ctpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of CTP names that can be used as the destination of an SNC to create on a given PTP. The NMS must specify the source CTP name (or source FTP name, in the case of VC3 port grouping circuits that have the source on E3, DS3I, or DS3IN cards), layer rate, and the direction of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns a CTP list based on current resource availability. The actual SNC creation might still fail. Values returned by this list can be used as the zEnd in the SNCCreateData_T parameter of 3.9.16 createAndActivateSNCFromUserLabel.

Parameters

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributes_T	In	Name of the PTP.
srcCTPName	globaldefs::NamingAttributes_T	In	Name of the CTP selected as the source CTP name of the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. This value must match the layer rate of srcCTP specified in the srcCTPName field. All CTP names available at this layer rate are returned.
direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_UNI (unidirectional) or CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of CTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of CTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining CTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised if the PTP does not exist.
EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.8.23 getAllDestCTPsForVCATMemberSNCProvisioning

Synopsis

```
void getAllDestCTPsForVCATMemberSNCProvisioning(
in globaldefs::NamingAttributes_T ptpName,
in globaldefs::NamingAttributes_T srcTPName,
in transmissionParameters::LayerRate_T layerRate,
in globaldefs::ConnectionDirection_T direction,
in unsigned long how_many,
out globaldefs:: NamingAttributesList_T tpList,
out globaldefs:: NamingAttributesIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of CTP names that can be used as the destination of a VCAT member SNC to create on a given PTP. The NMS must specify the source CTP name (or source FTP name for VC3 port grouping circuits whose source is on E3, DS3I, or DS3IN cards), layer rate, and direction of the SNC to

create. This interface does not reserve resources for SNC creation; rather, this method returns a CTP name list based on current resource availability. The actual SNC creation might still fail.
Values returned by this list can be used as the zEnd in each SNCCreateData_T listed from the sncCreateDataList field of the type VCAT SNCCreateData_T, which is required as a parameter of the createVCATl method described in 3.9.26 createVCAT.
This interface must be used according to 2.2.7.2.4 SNC Provisioning for ONS 15454 CE-MR-10 Cards.
Restrict the use of this API to VCAT circuits involving CE-MR-10 cards in Manual mode.

Parameters

See the Parameters section of 3.8.22 getAllDestCTPsForSNCProvisioning.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised if the PTP does not exist.
EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.8.24 getAllSrcFTPsForSNCProvisioning

Synopsis

```
void getAllSrcFTPsForSNCProvisioning(
in globaldefs::NamingAttributes_T meName,
in transmissionParameters::LayerRate_T layerRate,
in globaldefs::ConnectionDirection_T direction,
in unsigned long how_many,
in unsigned long members,
out globaldefs:: NamingAttributesList_T tpList,
out globaldefs:: NamingAttributesIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of FTP names that can be used as the source of an SNC to create on a given managed element. The NMS must specify the layer rate of the SNC that is created. This interface does not reserve resources for SNC creation; rather, this method returns the FTP name list based on current resource availability. The actual SNC creation might still fail. Values returned by this list can be used as the aEnd in the SNCCreateData_T parameter of the createAndActivateSNCFromUserLabel method described in 3.9.16 createAndActivateSNCFromUserLabel.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the ME that is the source of the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. All FTPs available at this layer rate are returned.

Name	Type	Input/ Output	Description
direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of FTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of FTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining FTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.8.25 getAllDestFTPsForSNCProvisioning

Synopsis

```
void getAllDestFTPsForSNCProvisioning(
in globaldefs::NamingAttributes_T meName,
in globaldefs::NamingAttributes_T srcTPName,
in transmissionParameters::LayerRate_T layerRate,
in globaldefs::ConnectionDirection_T direction,
in unsigned long how_many,
in unsigned long no_of_member_circuits,
out globaldefs:: NamingAttributesList_T tpList,
out globaldefs:: NamingAttributesIterator_I tpIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of FTP names that can be used as the destination of an SNC to create on a specific managed element. The NMS must specify the source TP (CTP or FTP) name and layer rate of the SNC to create. This interface does not reserve resources for SNC creation; rather, this method returns the FTP name list based on current resource availability. The actual SNC creation might still fail. Values returned by this list can be used as the zEnd in the SNCCreateData_T parameter of the createAndActivateSNCFromUserLabel method described in 3.9.16 createAndActivateSNCFromUserLabel.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the ME that is the source of the SNC to create.

Name	Type	Input/ Output	Description
srcTPName	globaldefs::NamingAttributes_T	In	Name of the TP that is selected as the source TP for the SNC to create.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate at which the SNC is created. All FTPs available at this layer rate are returned.
direction	globaldefs::ConnectionDirection_T	In	Prime Optical supports CD_BI (bidirectional).
how_many	unsigned long	In	Maximum number of FTP names to report in the first batch.
tpList	globaldefs::NamingAttributesList_T	Out	First batch of FTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining FTP names.

For a list of supported layer rates, see 2.2.1 Layer Rate.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised when layerRate or direction are invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.8.26 forceNEResync

Synopsis

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

Description

This interface forces Prime Optical to resynchronize the inventory and alarms of the specified managed element. This interface is an asynchronous call and returns after basic validation. The managed element is then queued for resynchronization. Prime Optical generates a state change event indicating that the emsInSyncState has changed to False. Upon successful completion of resynchronization, Prime Optical generates a state change event indicating that the emsInSyncState attribute has changed to True. An entry is made in the Prime Optical error log if the resynchronization fails.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the managed element to resynchronize.

Throws

globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or if the managed element is already in the resynchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.8.27 getAllTopologicalLinksForME

Synopsis

```
void getAllTopologicalLinksForME(
        in globaldefs::NamingAttributes_T meName,
        in unsigned long how_many,
        out topologicalLink::TopologicalLinkList_T topoList,
        out topologicalLink::TopologicalLinkIterator_I topoIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of topological links that originate or terminate on the specified managed element.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the managed element.
how_many	unsigned long	In	Maximum number of topological links to return in the first batch.
topoList	topologicalLink::TopologicalLinkList_T	Out	First batch of topological links.
tpIt	topologicalLink::TopologicalLinkIterator_I	Out	Iterator to retrieve the remaining topological links.

For details about the TopologicalLink_T structure, see 3.3.14 getAllTopLevelTopologicalLinks.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
Prime Optical-defined.

3.8.28 getAllRolls

Synopsis

```
void getAllRolls(
        in globaldefs::NamingAttributes_T meName,
        in unsigned long how_many,
```

```
        out multiLayerSubnetwork::RollList_T rollList,
        out multiLayerSubnetwork::RollIterator_I rollIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of rolls within the managed element whose name is passed as a parameter.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of MEs with rolls that are being queried.
how_many	unsigned long	In	Maximum number of rolls to report in the first batch.
rollList	multiLayerSubnetwork::RollList_T	Out	First batch of rolls.
rollIt	multiLayerSubnetwork::RollIterator_I	Out	Iterator to retrieve the remaining rolls.

Prime Optical supports Roll_T as follows:

- globaldefs::NamingAttributes_T rollName
  - Supported. It contains three tuples: the first tuple contains the EMS name, the second tuple contains theManagedElement name, and the third tuple contains the nativeEMSName of this roll.
- string::userLabel
  - Not supported.
- string::nativeEMSName
  - Supported. Each roll has a unique name assigned by the Cisco MSM.
- string::owner
  - Not supported.

- multiLayerSubnetwork::RollMode_T rollMode
  - Supported. Possible values are AUTO or MANUAL.
- globaldefs::NamingAttributes_T rollFromSNCName
  - Supported. The name of the rollFrom SNC.
- globaldefs::NamingAttributes_T rollToSNCName
  - Supported. The name of the rollTo SNC.
- globaldefs::NamingAttributes_T rollFromCTPName
  - Supported. The name of the CTP that is rolled over.
- globaldefs::NamingAttributes_T rollToCTPName
  - Supported. The CTP to which the rollFromCTPName CTP is rolled over.
- multiLayerSubnetwork::RollState_T rollState

Supported. The following values are supported:

  - ROLL_PENDING
  - ROLL_COMPLETED
  - ROLL_CANCELLED
  - ROLL_INCOMPLETE
- boolean rollValidSignal
  - Supported. Values are TRUE or FALSE.

Throws

globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or in config synchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.8.29 getAllRollNames

Synopsis

```
void getAllRollNames(
        in globaldefs::NamingAttributes_T meName,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of roll names within the managed element whose name is passed as a parameter.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the managed element with rolls that are being queried.
how_many	unsigned long	In	Maximum number of rolls to report in the first batch.
nameList	globaldefs::NamingAttributesList	Out	First batch of roll names.
nameIt	globaldefs::namingAttributesIterator_I	Out	Iterator to retrieve the remaining roll names.

Throws
globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or in config synchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.8.30 getRoll

Synopsis

```
void getRoll(
        in globaldefs::NamingAttributes_T rollName,
```

```
        out multiLayerSubnetwork::Roll_T roll,
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns the Roll_T structure for a specific roll name.
ParameterNormals

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll.
Roll	multiLayerSubnetwork::Roll_T	Out	Roll object.

Prime Optical support for Roll_T is described in 3.8.28 getAllRolls.
Throws
globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or in config synchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the roll does not exist.

EXCPT_INVALID_INPUT - Raised if the roll name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.8.31 getRollToPTPs

Synopsis

```
void getRollToPTPs (
        in globaldefs::NamingAttributes_T meName,
        in globaldefs::NamingAttributesList_T sncNames,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T tpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of PTPs on a managed element that can be used for provisioning a roll on a specific SNC. The returned PTPs can be used to contain the rollToCTP. The SNC must be associated with the managed element on which the interface is invoked.
Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Name of the managed element.
sncNames	globaldefs::NamingAttributesList_T	In	Name of the SNC to roll over. If more than one SNC is specified, the first SNC becomes the RollFrom SNC and the second SNC becomes the RollToSNC.
how_many	Unsigned	In	Maximum number of PTPs to report in the first batch.
tpList	globaldefs::namingAttributesList_T	Out	First batch of PTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining PTP

Name	Type	Input/ Output	Description
			names.

Throws

globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or in config synchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.8.32 getRollToCTPs

Synopsis

```
void getRollToCTPs(
        in globaldefs::NamingAttributes_T ptpName,
        in globaldefs::NamingAttributesList_T sncNames,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T tpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of CTPs contained in the PTP. This list can be used for roll provisioning as the rollToCTP for a given SNC. The list of PTPs can be obtained by using the operation defined in 3.8.31 getRollToPTPs.

Parameters

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributes_T	In	Name of the PTP for which the list of CTPs is obtained for roll provisioning.
how_many	Unsigned	In	Maximum number of PTPs to report in the first batch.
sncNames	globaldefs::NamingAttributesList_T	In	Name of the SNC to roll over. If more than one SNC is specified, the first SNC becomes the RollFrom SNC and the second SNC becomes the RollToSNC.
tpList	globaldefs::namingAttributesList_T	Out	First batch of CTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve remaining CTP names.

Throws

globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or in config synchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.8.33 getRollFromCTPs

Synopsis

```
void getRollFromCTPs(
        in globaldefs::NamingAttributes_T meName,
        in globaldefs::NamingAttributesList_T sncNames,
        in globaldefs::NamingAttributes_T fixCTPName,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T tpList,
        out globaldefs::NamingAttributesIterator_I tpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of rollFromCTPNames after selecting the fixed point of the roll. This is needed for path-protection SNCs and SNCs that have a secondary source, because multiple rollFromCTP values are possible.
This interface also returns a list of CTPs that can be used to populate the rollFromCTP data structure in the RollCreateData_T data structure.

Parameters

Name	Type	Input/ Output	Description
sncNames	globaldefs::NamingAttributesList_T	In	Name of the SNC to roll over. If more than one SNC is specified, the first SNC becomes the RollFrom SNC and the second SNC becomes the RollToSNC.
fixCTPName	globaldefs::NamingAttributes_T	In	Fixed point on the roll.
how_many	Unsigned	In	Maximum number of CTPs to report in the first batch.
tpList	globaldefs::namingAttributesList_T	Out	First batch of CTP names.
tpIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining CTP names.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the cancellation of the rolling operation.

EXCPT_UNABLE_TO_COMPLY - Raised when the roll cannot be created.

Compliance
Prime Optical-defined.

3.8.34 setPortName

Synopsis

```
void setPortName(
        in globaldef::NamningAttributes_T ptpName,
        in string portName,
        out string newPortName)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables an NMS to set a name for the selected port. The PTP specified as input identifies the port. The port name is set to the value passed in through the portName parameter.

Parameters

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributesList_T	In	Name of the PTP whose name will be set.
portName	String	In	Port name value to set for the specified PTP.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the ptpName does not refer to a valid PTP name or when it contains undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when ptpName references an object that does not exist.

Compliance
Prime Optical-defined.

3.8.35 getPortName

Synopsis

```
void getPortName (
in globaldefs:: NamingAttributes_T ptpName,
out string portName )
raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables an NMS to get a name for the selected port. The PTP specified as input identifies the port. The resulting port name is used as the argument parameter.

Parameters

Name	Type	Input/ Output	Description

Name	Type	Input/ Output	Description
ptpName	globaldefs::NamingAttributesList_T	In	Name of the PTP whose name will be set.
portName	String	Out	Port name specified for the input PTP.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the ptpName does not refer to a valid PTP name or when it contains undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when ptpName references an object that does not exist.

Compliance

Prime Optical-defined.

3.8.36 getAllClearedAlarms

Synopsis

```
void getAllClearedAlarms(
        in globaldefs::NamingAttributes_T meName,
        in notifications::ProbablyCauseList_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 interface enables an NMS to request all of the cleared and acknowledged alarms for the specified ME. Alarms that are not reported by the ME to the EMS are not reported by this operation. Some alarms may be filtered out by specifying the probable cause of the severities.

The result of this operation is independent of the filtering set up by the NMS for the notification service.

ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS 15600 SONET, ONS 15600 SDH, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SONET and ONS NCS2K SDH

The ONS 15310 CL, ONS 15310 MA SDH, ONS 15310 MA SONET, ONS 15327, ONS 15454 SONET, ONS 15454 SDH, ONS 15600 SONET, ONS 15600 SDH, CPT 200, CPT 200 SDH, CPT 600, CPT 600 SDH, ONS NCS2K SONET, and ONS NCS2K SDH do not send a “clear” for TCA events. TCAs are not reported in the eventList returned by this method.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributesList_T	In	Name of the ME for which to retrieve alarms.
excludeProbCauseList	notifications::ProbableCauseList_T	In	List of probable causes to exclude (for which events are not reported).
exludeSeverityList	notifications::PerceivedSeverityList_T	In	List of severities to exclude

Name	Type	Input/ Output	Description
			from the output event list.
how_many	Unsigned long	In	Maximum number of alarms to report in the first batch.
eventList	notifications::EventList	Out	First batch of alarms.
eventIt	notifications::EventIterator_I	Out	Iterator to retrieve the remaining alarms.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when meName does not reference a managed element object or excludeProbCauseList contains undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.8.37 getAllCrossConnections

Synopsis

```
void getAllCrossConnections(in globaldefs::NamingAttributes_T
managedElementName,
      in transmissionParameters::LayerRateList_T connectionRateList,
      in unsigned long how_many,
      out subnetworkConnection::CrossConnectList_T ccList,
      out subnetworkConnection::CCIterator_I ccIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of cross-connections on the managed element.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the managed element.
connectionRateList	transmissionParameters::LayerRateList_T	In	Layer rate list.
how_many	unsigned long	In	Maximum number of CrossConnect_T to report in the first batch.
ccList	subnetworkConnection::CrossConnectList_T	Out	First batch of cross-connects.
ccIt	subnetworkConnection::CCIterator_I	Out	Iterator to retrieve the remaining cross-

Name	Type	Input/ Output	Description
			connects.

A cross-connect represents a connection within a single managed element. This structure is used primarily in the specification of routes.
Prime Optical support for the CrossConnection_T structure is as follows:

- boolean active:
  - Always true. Prime Optical only returns cross-connections for SNCs that are in SNCState_T.SNCS_ACTIVE or SNCState_T.SNCS_PARTIAL. This attribute indicates whether the cross-connect is active in the ME.
- globaldefs::ConnectionDirection_T direction:
  - Direction of the cross-connection. Prime Optical returns CD_UNI or CD_BI.
- SNCType_T ccType:
  - Prime Optical always returns ST_SIMPLE.
- globaldefs::NamingAttributesList_T aEndNameList:
  - Names of TPs at the aEnd of the cross-connection.
- globaldefs::NamingAttributesList_T zEndNameList:
  - Names of TPs at the zEnd of the cross-connection.
- globaldefs::NVSList_T additionalInfo:
  - This attribute allows the EMS to communicate to the NMS additional information that is not explicitly modeled. This attribute is read-only. Prime Optical does not return any additional information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME object that does not exist.

EXCPT_INVALID_INPUT - Raised if managedElementName does not reference a managed element.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

TMF-defined.

3.8.38 getAllSoftwareDownloadJobs

Synopsis

```
void getAllSoftwareDownloadJobs(
        in unsigned long how_many,
        out JobMonitorList_T jobList)
        raises(globaldefs::ProcessingFailureException);
```

Description

This interface retrieves the status of all scheduled and completed software download jobs.

Parameters

Name	Type	Input/ Output	Description

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Number of download jobs to retrieve.
jobList	JobMonitorList_T	Out	List of jobs scheduled (see the definition that follows this table).

```
typedef sequence<JobMonitorData_T> JobMonitorList_T;
struct JobMonitorData_T{
        unsigned long jobId;
        unsigned long taskId;
        string taskType;
        string taskOwner;
        string neId;
        string taskStatus;
        string creationTime;
        string scheduledTime;
        string startTime;
        string endTime;
        string displayInformation;
        string userComments;
        string additionalComments;
};
```

The following table lists descriptions of each individual parameter.

Name	Type	Description
jobId	string	The job ID.
taskId	string	The task ID.
taskType	string	The kind of task.
taskOwner	string	The task owner.
neId	string	The managed element name.
taskStatus	string	The status of the task. Valid values are: 1: QUEUED 2: RUNNING 3: CANCELLED 4: FAILED 5: SUCCEEDED 6: WAITING
creationTime	string	The time when the task was created.
scheduledTime	string	The scheduled time for the task.
startTime	string	The actual start time for the task.

Name	Type	Description
endTime	string	The actual time of completion.
displayInformation	string	Display information.
userComments	string	User comments.
additionalComments	string	Additional comments.

Throws
globaldefs::ProcessingFailureException

Compliance
Prime Optical-defined.

3.8.39 setMEAdminState

Synopsis

```
void setMEAdminState  (in globaldefs::NamingAttributes_T meName,
    in AdminState_T adminState)
raises         (globaldefs::ProcessingFailureException)
```

Description
This interface is used to force the EMS to change the administrative state of the managed element to the specified state.

Parameters

Name	Type	Input/ Output	Description
meName	NamingAttributes_T	In	Name of the managed element to resynchronize. It must contain a name/value pair, where the name is ManagedElement and the value is ME Name.
adminState	AdminState_T	In	Admin state to which the managed element name changes.

Throws
globaldefs::ProcessingFailureException

EXCPT_UNABLE_TO_COMPLY - Raised if the managed element is in the initial poll state or if the managed element is already in the resynchronization state.

EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.

EXCPT_INVALID_INPUT - Raised if the managed element name is invalid.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

The following enum is used to define the various NE admin states:

```
enum  AdminState_T {
IN_SERVICE,
OUT_OF_SERVICE
 OUT_OF_SERVICE_BY_MAINTENANCE
```

```
}
```

Compliance
Prime Optical-defined.

3.9 multiLayerSubnetwork::MultiLayerSubnetworkMgr_I

The multiLayerSubnetworkMgr_I operation is used to gain access to subnetworks and their respective operations. 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 ManagedElementManager_I:common::Common_I
```

multiLayerSubnetworkMgr_I also implements the getCapabilities method. Operations described in following subsections are returned to the NMS when it invokes getCapabilities.

3.9.1 getAllEdgePoints

Synopsis

```
void getAllEdgePoints(
        in globaldefs::NamingAttributes_T subnetName,
        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 interface enables an NMS to request a list of edge termination points for the specified subnetwork at one or more of the NMS-specified layers that are capable of containing CTPs that can be connected at one or more of the NMS-specified connection layer rates.
Prime Optical reports the PTPs that are not part of the topological link.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
tpLayerList	transmissionParameters::LayerRateList_T	In	List of TP layer rates for which edge points are retrieved. To be reported, an edge point must contain at least one of the layer rates specified. If the list is empty, edge points of all rates are returned.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	Prime Optical does not use this parameter.
how_many	unsigned long	In	Maximum number of edge points to return in the first batch.
tpList	terminationPoint::TerminationPointList_T	Out	First batch of edge points.
tpIt	terminationPoint::TerminationPointIterator_I	Out	Iterator to retrieve the remaining edge points.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object, or tpLayerRateList or connectionLayerRateList contain undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.2 getAllEdgePointNames

Synopsis
getAllEdgePointNames

```
  void getAllEdgePointNames(in globaldefs::NamingAttributes_T subnetName,
      in transmissionParameters::LayerRateList_T layerRateList,
      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 interface enables an NMS to request a list of edge termination points for the specified subnetwork at one or more of the NMS-specified layers that are capable of containing CTPs that can be connected at one or more of the NMS-specified connection layer rates.
Prime Optical reports the PTPs that are not part of the topological link.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
tpLayerList	transmissionParameters::LayerRateList_T	In	List of TP layer rates for which the edge points are fetched. To be reported, an edge point must contain at least one of the layer rates specified. If the list is empty, edge points of all rates are returned.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	Prime Optical does not use this parameter.
how_many	unsigned long	In	Maximum number of

Name	Type	Input/ Output	Description
			edge points to return in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of edge point names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining edge point names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object or tpLayerRateList or connectionLayerRateList contain undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.9.3 getAllManagedElement

Synopsis

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

Description

This interface enables an NMS to request a list of managed elements that are associated with the specified subnetwork.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
how_many	unsigned long	In	Maximum number of MEs to report in the first batch.
meList	managedElement::ManagedElementList_T	Out	First batch of ME names.
MeIt	managedelement::ManagedElementIterator_I	Out	Iterator to retrieve the remaining ME names.

For details about the ManagedElement_T structure, see 3.8.3 getAllManagedElements.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.4 getAllManagedElementNames

Synopsis

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

Description
This interface has the same behavior as the operation described in 3.9.3 getAllManagedElement, but instead of returning entire object structures, this operation returns their names.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
how_many	unsigned long	In	Maximum number of MEs to report in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of ME names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining ME names.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.5 getAllSubnetworkConnections

Synopsis

```
void getAllSubnetworkConnections(
        in globaldefs::NamingAttributes_T subnetName,
```

```
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T sncList,
        out subnetworkConnection::SNCIterator_I sncIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request a list of SNCs for the specified subnetwork at the specified connection rates.

- For the ONS 15310 MA SONET, this method does not report STS-1 SNC on DS1 ports of DS1_xx cards.
- For the ONS 15310 MA SDH, this method does not report VC4 SNC on E1 ports of E1_63_E3_3 cards.
- For the ONS 15327, this method does not report STS-1 SNC on XTC using DS1 ports.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.
- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, "Using GateWay/CORBA Connectionless Interfaces" for supported SNCs.
- This interface does not report SNCs with duplicate names or VCAT SNCs.

If the Prime Optical subnetwork contains multiple circuits with the same name, the getAllSubnetworkConnections call returns the circuit name and details for only one of the circuits. It is not possible to predict which one of the multiple circuits will have its name and details returned.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	in	Name of the subnetwork.
connectionLayerRateList	transmissionParameters::LayerRateList_T	in	List of rates of the SNCs to report. If an empty list is specified, all SNCs of all rates are reported.
how_many	unsigned long	in	Maximum number of SNCs to report in the first batch.
sncList	subnetworkConnection::SubnetworkConnectionList_T	out	First batch of SNCs.
sncIt	subnetworkConnection::SNCIterator_I	out	Iterator to retrieve the remaining SNCs.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.6 getAllSubnetworkConnectionsWithTP

Synopsis

```
void getAllSubnetworkConnectionsWithTP(
        in globaldefs::NamingAttributes_T tpName,
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T sncList,
        out subnetworkConnection::SNCIterator_I sncIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables an NMS to request a list of the SNCs using the specified TP at the specified connection rates. A TP might be a PTP that requires a full list of SNCs using any of its contained CTPs. A TP might be a CTP or FTP that only requires the SNCs using that CTP or FTP. Both end CTPs and intermediate CTPs at the same layer are considered. In all cases, only SNCs that also match the specified connection rate filter are returned.

- For the ONS 15310 MA SONET, this method does not report STS-1 SNC on DS1 ports of DS1_xx cards.
- For the ONS 15310 MA SDH, this method does not report VC4 SNC on E1 ports of E1_63_E3_3 cards.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.
- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, "Using GateWay/CORBA Connectionless Interfaces" for supported SNCs.
- This interface does not report SNCs with duplicate names.

For ONS 15600 SONET and ONS 15600 SDH NEs, if bridge-and-roll is applied, one CTP may be associated with two SNCs during a transient phase. If Prime Optical invokes this method during this phase, it reports both SNCs.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	In	Termination point for which to report SNCs.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the SNCs to report. If an empty list is specified, all SNCs of all rates are reported.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
sncList	subnetworkConnection::SubnetworkConnectionList_T	Out	First batch of SNCs.
sncIt	subnetworkConnection::SNCIterator_I	Out	Iterator to retrieve the remaining SNCs.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised if tpName does not reference a termination point object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.9.7 getAllSubnetworkConnectionNames

Synopsis

```
void getAllSubnetworkConnectionNames(
        in globaldefs::NamingAttributes_T subnetName,
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface has the same behavior as the operation described in 3.9.5 getAllSubnetworkConnections, but instead of returning entire object structures, this operation returns their names.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
connectionLayerRateList	transmissionParameters::LayerRateList_T	In	List of rates of the SNCs to report. If an empty list is specified, all SNCs of all rates are reported.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
nameList	globaldefs::NamingAttributesList	Out	First batch of SNC names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining SNC names.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.9.8 getAllSubnetworkConnectionNamesWithTP

Synopsis

```
void getAllSubnetworkConnectionNamesWithTP(
        in globaldefs::NamingAttributes_T tpName,
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface has the same behavior as the operation described in 3.9.6 getAllSubnetworkConnectionsWithTP, but instead of returning entire object structures, this operation returns their names.

Parameters

Name	Type	Input/ Output	Description
tpName	globaldefs::NamingAttributes_T	in	Termination point for which to report SNCs.
connectionLayerRateList	transmissionParameters::LayerRateList_T	in	List of rates of the SNCs to report. If an empty list is specified, all SNCs of all rates are reported.
how_many	unsigned long	in	Maximum number of SNCs to report in the first batch.
nameList	globaldefs::NamingAttributesList	out	First batch of SNC names.
nameIt	globaldefs::NamingAttributesIterator_I	out	Iterator to retrieve the remaining SNC names.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised if tpName does not reference a terminationPoint object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.9.9 getAllTopologicalLinks

Synopsis

```
void getAllTopologicalLinks(
        in globaldefs::NamingAttributes_T subnetName,
        in unsigned long how_many,
        out topologicalLink::TopologicalLinkList_T topoList,
        out topologicalLink::TopologicalLinkIterator_I topoIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of topological links that exist within the subnetwork whose name is passed as a parameter.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
how_many	unsigned long	In	Maximum number of topological links to return in the first batch.
topoList	topologicalLink::TopologicalLinkList_T	Out	First batch of topological links.
topoIt	topologicalLink::TopologicalLinkIterator_I	Out	Iterator to retrieve the remaining topological links.

For details about the TopologicalLink_T structure, see 3.3.14 getAllTopLevelTopologicalLinks.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.10 getAllTopologicalLinkNames

Synopsis

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

Description
This interface has the same behavior as the operation described in 3.9.9 getAllTopologicalLinks, but instead of returning entire object structures, this operation returns their names.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
how_many	unsigned long	In	Maximum number of topological links to return in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of topological link names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining topological link names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.9.11 getMultiLayerSubnetwork

Synopsis

```
void getMultiLayerSubnetwork(
        in globaldefs::NamingAttributes_T subnetName,
        out MultiLayerSubnetwork_T subnetwork)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a subnetwork given a subnetwork name.
Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
subnetwork	MultiLayerSubnetwork_T	Out	Subnetwork structure returned.

For details about the MultiLayerSubnetwork_T structure, see 3.3.12 getAllTopLevelSubnetworks.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references a multiLayerSubnetwork object that does not exist.

Compliance
TMF-defined.

3.9.12 getCTPsAndTopologicalLinks

Synopsis

```
void getCTPsAndTopologicalLinks(
        in globaldefs::NamingAttributes_T sncName,
        out terminationPoint::TerminationPointList_T ctpList,
        out topologicalLink::TopologicalLinkList_T topologicalLinkList)
    raises(globaldefs::ProcessingFailureException);
```

Description
The NMS provides the SNC name, and Prime Optical returns ordered lists: one list for all CTPs/FTPs and another for all topological links. Prime Optical reports topological links and CTPs/FTPs of the working path for a 1+1, 2-fiber BLSR, or 4-fiber BLSR protected SNC. Prime Optical reports topological links and CTPs/FTPs of both the working and protect paths for a path-protection SNC. An SNC with the source or destination on Ethernet equipment is also supported.

- For the ONS 15310, this method does not report STS-1 SNC on CTX using DS1 ports.
- For the ONS 15327, this method does not report STS-1 SNC on XTC using DS1 ports.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.
- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, “Using GateWay/CORBA Connectionless Interfaces” for supported CTPs and links.

This interface does not report VCAT SNCs.
The lists are unordered for circuits for which:

- The circuit state is not active.
- The circuit has DRI protection.

A processing failure exception is returned with an EXCPT_UNABLE_TO_COMPLY exception in all of the preceding conditions.

Parameters

Name	Type	Input/ Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the subnetwork connection.
ctpList	terminationPoint::TerminationPointList_T	Out	Ordered list of CTP/FTPs for this SNC.
topoList	topologicalLink::TopologicalLinkList_T	Out	Ordered topological links through which this SNC traverses.

For details about the TerminationPoint_T structure, see 3.6.3 getAllSupportedPTPs.
For details about the TopologicalLink_T structure, see 3.3.14 getAllTopLevelTopologicalLinks.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_UNABLE_TO_COMPLY – Raised when invoked on SNC on Ethernet equipment or STS-1 SNC on DS1 equipment.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

Note:

The och-cc circuit from now on will report src/dst ctps, the ott link and all ctps that belong to ott link. That means that to obntain the same route as before, the end user must perform almost 3 calls the first to get the ochcc ochtails relationship, the second to get the ochcc route an then from ochtail circuit get the trails route

3.9.13 getSNC

3.9.13 getSNC

Synopsis

```
void getSNC(
        in globaldefs::NamingAttributes_T sncName,
        out subnetworkConnection::SubnetworkConnection_T snc)
        raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the SNC structure for the SNC whose name is supplied as a parameter.

- For the ONS 15310 CL, this method does not report STS-1 SNC on CTX using DS1 ports.
- For the ONS 15310 MA SONET, this method does not report STS-1 SNC on DS1 ports of DS1_xx cards.
- For the ONS 15310 MA SDH, this method does not report VC4 SNC on E1 ports of E1_63_E3_3 cards.
- For the ONS 15327, this method does not report STS-1 SNC on XTC using DS1 ports.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.

- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, Using GateWay/CORBA Connectionless Interfaces for supported SNCs.

This interface does not report VCAT SNCs.

A processing failure exception is returned with an EXCPT_UNABLE_TO_COMPLY exception in all of the preceding conditions.

Parameters

Name	Type	Input/ Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the subnetwork connection.
snc	subnetworkConnection::SubnetworkConnection_T	Out	SNC structure retrieved.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not reference an SNC object or when sncName is not unique.

EXCPT_UNABLE_TO_COMPLY - Raised when invoked on SNC on Ethernet equipment or STS-1 SNC on DS1 equipment.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.9.14 getSNCsByUserLabel

Synopsis

```
void getSNC(
        in string userLabel,
        out subnetworkConnection::SubnetworkConnectionList_T sncList)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the SNC structure for the SNC whose name is supplied as a parameter.

- For the ONS 15310 CL, this method does not report STS-1 SNC on CTX using DS1 ports.
- For the ONS 15310 MA SONET, this method does not report STS-1 SNC on DS1 ports of DS1_xx cards.
- For the ONS 15310 MA SDH, this method does not report VC4 SNC on E1 ports of E1_63_E3_3 cards.
- For the ONS 15327, this method does not report STS-1 SNC on XTC using DS1 ports.
- For the ONS 15454 SONET, this method does not report STS-1 SNC on DS1 equipment.
- For the ONS 15454 SDH, this method does not report VC-4 SNC on E1 or E1_42 equipment.
- For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, “Using GateWay/CORBA Connectionless Interfaces” for supported SNCs.

This interface does not report VCAT SNCs.

A processing failure exception is returned with an EXCPT_UNABLE_TO_COMPLY exception in all of the preceding conditions.

Parameters

Name	Type	Input/ Output	Description
userLabel	String	In	Name of the subnetwork connection.
snc	subnetworkConnection::SubnetworkConnection_T	Out	SNC structure retrieved.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not reference an SNC object or when sncName is not unique.

EXCPT_UNABLE_TO_COMPLY – Raised when invoked on SNC on Ethernet equipment or STS-1 SNC on DS1 equipment.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.9.15 getTopologicalLink

Synopsis

```
void getTopologicalLink(
        in globaldefs::NamingAttributes_T topoLinkName,
        out topologicalLink::TopologicalLink_T topoLink)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a topological link. Prime Optical reports all autodiscovered and manual topological links.

Parameters

Name	Type	Input/ Output	Description
topoLinkName	Globaldefs::NamingAttributes_T	In	Name of the subnetwork topological link to retrieve.
topoLink	topologicalLink::TopologicalLink_T	Out	Subnetwork topological link returned.

For details about the TopologicalLink_T structure, see 3.3.14 getAllTopLevelTopologicalLinks.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when topoLinkName does not reference a subnetwork topological link object.

```
EXCPT_ENTITY_NOT_FOUND - Raised when topoLinkName references a subnetwork
topological link object that does not exist.
```

Compliance
TMF-defined.

3.9.16 createAndActivateSNCFromUserLabel

Synopsis

```
void createAndActivateSNCFromUserLabel(
        in subnetworkConnection::SNCCreateData_T createData,
        in subnetworkConnection::GradesOfImpact_T tolerableImpact,
        in EMSFreedomLevel_T emsFreedomLevel,
        inout subnetworkConnection::TPDataList_T tpsToModify,
        out subnetworkConnection::SubnetworkConnection_T theSNC,
        out string errorReason)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface provides a way to create and activate a subnetwork connection in one command. It is Prime Optical-specific and not TMF-defined. According to the TMF, the SNC name is assigned by the EMS when it creates the SNC. However, if the NMS is to control the SNC name, it must use this method. The NMS provides the name of the SNC in the userLabel field of the SNCCreateData input parameter. Prime Optical uses the userLabel value and assign it to the name of the new SNC. Prime Optical also verifies the uniqueness of the name. The operation fails if an SNC with the same name as the userLabel exists. This is an asynchronous operation. Successful completion of this operation only guarantees delivery of the creation request to Prime Optical. Operation on CTPs of this SNC might still fail.
Optionally, you can also provide a description for the SNC by entering it in the additionalCreationInfo field.
Prime Optical cleans up resources if SNC creation fails. For example, if Prime Optical fails to create the SNC on the destination node, Prime Optical also cleans up resources on the source node.
If the OC-n port is configured for SDH, Prime Optical can create the SNC. Prime Optical supports only those layer rates supported by the NE. However, IPPM and J1 path tracing are not supported for CTPs of this SNC.
FTPs can be used as A ends and Z ends of the SNCs. A typical use of FTP is to create an SNC on Ethernet equipment, create a tunnel SNC, or create a VAP/LAP circuit. See 2.2.7 Provisioning Subnetwork Connections for more information.
Creating SNCs between ONS 15327 nodes or ONS 15310, and between ONS 15600 nodes is identical to creating SNCs between ONS 15454 or ONS NCS2K nodes. It is also possible to create SNCs among ONS 15327, ONS 15454, ONS 15600, ONSNCS2K nodes.
Prime Optical supports protection channel access (PCA). PCA SNCs are unprotected. For details about PCA, see the NE documentation.
You can provide a description for the SNC. Enter the description in the additionalCreationInfo field.
To create a dual-ring interconnect SNC, set the staticProtectionLevel value in createData to HIGHLY_PROTECTED. Use the CTC or Prime Optical client to modify the Hold-off Timer attribute for this SNC.
The following types of circuit creation are not supported:

- Monitor circuits
- STS-1 on DS1 cards and VC-4 on E1 and E1_42 cards

- DRI circuits with constraint routing

In addition to specifying SNC creation parameters using the SNCCreateData_T structure, the NMS can provide a partial or complete list of MEs or both end PTPs of a topological link to include or exclude in the SNC route during SNC creation. The NMS can specify only one type of object name list during the SNC creation. Prime Optical returns a processing failure exception if the NMS-specified list is not homogeneous. Prime Optical returns a processing failure exception with EXCPT_ENTITY_NOT_FOUND as the error reason if any one of the specified MEs or PTPs does not exist. If the NMS specifies a PTP list, Prime Optical verifies whether PTPs are specified in pairs and each PTP of a PTP pair is associated with each end PTP of the same topological link. Prime Optical uses the first PTP of a PTP pair as the aEnd of the topological link, and uses the second PTP of the same PTP pair as the zEnd of the topological link. Prime Optical attempts to route the SNC from the aEnd PTP of a PTP pair to the zEnd PTP of the same pair on a topological link. Prime Optical returns a processing failure exception if the SNC cannot be routed from the specified aEnd PTP to the zEnd PTP of a topological link. The NMS then must interchange the aEnd PTP and zEnd PTP specified, or provide a different PTP pair, or delete the PTP pair from the include or exclude list. Prime Optical returns a processing failure exception if at least one PTP is not associated with a topological link. The NMS must ensure that the sequence of the object name list conforms to the sequence of the objects that exist in the network. Prime Optical returns a processing failure exception if the sequence is nonconforming.

In addition to specifying SNC creation parameters using the SNCCreateData_T structure, the NMS can provide a complete list of CTP pairs representing the complete route of the SNC. Each CTP pair represents the two endpoints of one span in the SNC route. A span represents a partial route of an SNC with the endpoints specified on the endpoints of a topological link. Prime Optical processes the input list in CTP pairs. Prime Optical assumes each pair represents the two endpoints of a span, and the CTP native names are the same for the two endpoints (the STS numbers are the same for each CTP pair). The NMS must ensure that the sequence of spans represented by the CTP pairs conforms to the sequence existing in the network. Prime Optical returns a processing failure exception if the sequence is nonconforming.

If you create an SNC with at least one endpoint on ML-series equipment, Prime Optical identifies the SNC for a new Layer 2 point-to-point topology. Prime Optical creates a new Layer 2 point-to-point topology and an object creation event is generated to notify for the creation of a new L2 point-to-point topology in Prime Optical.

Prime Optical provides support for the creation of open-ended path-protection circuits. There are two types of SNCs that can be created for open-ended path-protection circuits.

- For one source point and two destination endpoints, the NMS must specify the source point in SNCCreateData_T.aEnd, and the two destination endpoints in SNCCreateData_T.zEnd.
- For two source points and one destination endpoint, the NMS must specify two source points in SNCCreateData_T.aEnd and one destination endpoint in SNCCreateData_T.zEnd.

Prime Optical supports optical channel SNC creation on WDM cards through this method. See the Parameters table in this section for detailed information. The sncSize field in additionalCreationInfo is used to specify the size of the optical channel SNC.

Prime Optical supports OCH trail SNC creation on DWDM transponder/muxponder cards or ITU-T line cards through a DWDM network. . See 2.2.7.4 SNC Provisioning on WDM Equipment.

For the ONS 15454 SONET and NCS2K SONET, this operation enables you to create an end-to-end STS-1 circuit using the portless transmux interfaces of the transmux card as an intermediate connection point. Information about the high-density portless transmux card (DS3XM-12) can be provided using the usePortlessTransmux, portlessTransmuxMeName, portlessTransmuxSlot, and isSourceDS3MappedSTS attributes in the additionalCreationInfo field.

For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, see Chapter 4, Using GateWay/CORBA Connectionless Interfaces for supported SNCs.

For GMPLS SNCs, see the GMPLS Provisioning section.

For CTPs used for WDM SNCs, the frequency notation in the naming convention of the CTP refers to the working path only. The frequency of the protected path is identified by the ProtectedFrequency attribute passed in the additionalInfo section of the SNC creation request if one of following conditions is satisfied:

- SNC type is OCHCC in a non-GMPLS circuit with splitter protection.
- SNC type is OCHCC in a GMPLS circuit with splitter protection.
- SNC type is OCHCC in a GMPLS circuit with Y cable protection.

✐ GateWay/CORBA cannot create manually routed OCHNC, OCHTRAIL, and OCHCC links.

Parameters

Name	Type	Input/ Output	Description
createData	subnetworkConnection::SNCCreateData_T	In	Structure describing the SNC to create and activate.
tolerableImpact	subnetworkConnection::GradesOfImpact_T	In	Maximum tolerable impact allowed.
emsFreedomLevel	EMSFreedomLevel_T	In	Maximum level of freedom allowed to the EMS to perform the creation and activation.
tpsToModify	subnetworkConnection::TPDataList_T	In/out	Prime Optical ignores this parameter.
theSNC	subnetworkConnection::SubnetworkConnection_T	Out	Resulting SNC, with the sncState and name set. The name is set to the user label supplied in the createData structure.
errorReason	String	Out	Specifies the creation and/or activation error(s), if any.

Prime Optical does not support locking.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Prime Optical support for the SNCCreateData_T structure is as follows:

- string userLabel
  - Supported. The value in this field is assigned to the name of the newly created SNC.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of SNCs if they are all created by Prime Optical. Prime Optical cannot guarantee the uniqueness if any other tool has been used instead.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Prime Optical supports CD_UNI (unidirectional), CD_BI (bidirectional), CD_UNI_EW (unidirectional east-to-west), CD_UNI_WE (unidirectional west-to-east), CD_BI_EW (bidirectional east-to-west), and CD_BI_WE (bidirectional west-to-east).
- StaticProtectionLevel_T staticProtectionLevel

  - Prime Optical supports UNPROTECTED, FULLY_PROTECTED, and HIGHLY_PROTECTED. Prime Optical does not support any other value.
- ProtectionEffort_T protectionEffort
  - Prime Optical supports all values defined by the TMF.
  - If the NMS requests an unprotected SNC on a BLSR ring, Prime Optical creates the SNC. The same is true for an unprotected SNC with 1+1 protection.
  - If the NMS specifies EFFORT_SAME_OR_WORSE or EFFORT_WHATEVER, Prime Optical expects the value of staticProtectionLevel to be UNPROTECTED. These values indicate that Prime Optical must create the PCA.
- Reroute_T rerouteAllowed
  - Not supported.
- NetworkRouted_T networkRouted
  - Not supported.
- SNCType_T sncType;
  - Not supported.
- transmissionParameters::LayerRate_T layerRate
  - Prime Optical supports the following layer rates:

Layer Rate Category	**Layer Rates**	**SNC Sizes**	**Comments**
Video	LR_D1_Video LR_HDTV LR_DV_6000 LR_DVBASI LR_SD_SDI LR_HD_SDI LR_SDI_3G_VIDEO	27=SD1/D1 26=HDTV 28=DV 6000 31=Video /DVB ASI 95=SD SDI 96=HD SDI 90=SDI-3G-VIDEO	For WDM OCHCC
Data/Storage	LR_ESCON LR_ETR LR_ISC3_PEER_1G LR_ISC3_PEER_2G LR_ISC1 LR_IB LR_SYSPLEX_ISC_COMP LR_SYSPLEX_ISC_PEER LR_ISC3_STP_1G LR_ISC3_STP_2G	20=ESCON 21=ETR-CLO 22=ISC PEER 24=ISC3 PEER 2Gb/s 23=ISC3 PEER 1Gb/s 25=ISC COMPAT 155=ISC1 30=data Storage /Video - ISC Channel 32=IB 22=ISC STP 24=ISC3 STP 2Gb/s	For WDM OCHCC
STS/VC	LR_STS1_and_AU3_High_Or d er_VC3 LR_STS3c_and_AU4_VC4 LR_STS12c_and_VC4_4c LR_STS48c_and_VC4_16c LR_STS192c_and_VC4_64c LR_STS6c_and_VC4_2c LR_STS9c_and_VC4_3c LR_STS24c_and_VC4_8c LR_STS18c_and_VC4_6c LR_STS36c_and_VC4_12c LR_VT1_5_and_TU11_VC11		Only on ONS15454 SDH NSC2K SDH

Layer Rate Category	Layer Rates	SNC Sizes	Comments
	LR_Low_Order_TU3_VC3 LR_VT2_and_TU12_VC12		
FICON	LR_FICON_1G LR_FICON_2G LR_FICON_4G LR_FC_100_1063M LR_FC_200_2125M	19=1Gb/s FiCon 18=2Gb/s FiCon 17=4Gb/s FiCon	For WDM OCHCC
FC	LR_FC_100_1063M LR_FC_200_2125M LR_FC_400_4250M LR_FC_800_8500M LR_FC_1000_10518M	16=1Gb/s FC 15=2Gb/s FC 14=4Gb/s FC 86= 8Gb/s FC 13=10Gb/s FC	For WDM OCHCC
SONET/SDH	LR_DSR_OC3_STM1 LR_DSR_OC12_STM4 LR_DSR_OC48_STM16 LR_DSR_OC192_STM64 LR_DSR_OC768_and_STM256	7=OC 192 8=OC 48 9=OC 12 10= OC 3 33=OC 768	For WDM OCHCC
PDH	LR_T1_and_DS1_1_5M LR_T3_and_DS3_45M LR_E1_2M LR_E3_34M	—	For WDM OCHCC Only on ONS15454 SDH NCS2K SDH
Ethernet	LR_Fast_Ethernet LR_Gigabit_Ethernet LR_Ten_Gigabit_Ethernet LR_Forty_Gigabit_Ethernet	38=FE 11=10Gb/s E 12=1Gb/s E 89=40Gb/s E 1=2.5Gb/s FEC 2=2.5Gb/s No FEC 3=10 Gb/s FEC 4=10 Gb/s No FEC 5=Multirate 46=100Gb/s E	For WDM OCHCC
OTU	LR_Optical_Channel LR_OCH_10_Gigabit_ITU LR_OCH_40_Gigabit_ITU LR_CPO_DSR_OTU1 LR_OCH_100_Gigabit_ITU	35=OCH OTU2 36=OCH OTU3 92=OCH OTU1 98=”ochOTU4	For WDM OCHCC
Other	LR_PASS_TROUGH LR_AUTO	29=Pass-through 6=Equipped not specific 40=OCH AUTO	Other

- CrossConnectList_T ccInclusions

Not supported.

- ResourceList_T neTpInclusions

Supported. You must provide a list of managed element names or the PTP names at the endpoints of the topological link. For complete manual routing, you must provide the list of all CTPs of the links (OCH trail links, if any) to which the circuit must be routed.
For protected OCHNC and OCHCC circuits, this parameter defines only the inclusion routing constraints for the working path of the circuit. To define the constraints for the protected path, Prime Optical provides the additional info parameters "includedProtectedME" and "excludedProtectedME."

- boolean fullRoute

Not supported.

- ResourceList_T neTpSncExclusions

Supported. Prime Optical expects a list of managed element names or the PTP names at the endpoints of the TopologicalLink to exclude from the SNC route.
For protected OCHNC and OCHCC circuits, this parameter defines only the exclusion routing constraints for the working path of the circuit. To define the constraints for the protected path, Prime Optical provides the additional info parameters "includedProtectedME" and "excludedProtectedME."

- globaldefs::NamingAttributesList_T aEnd

Supported. Prime Optical expects at least one valid CTP or FTP in this list. At most, two valid CTPs can be in the list.

- globaldefs::NamingAttributesList_T zEnd

Supported. Prime Optical expects at least one valid CTP or FTP in this list. At most, two valid CTPs can be in the list.

- globaldefs::NVSList_T additionalCreationInfo

This list can be used to provide additional parameters for SNC creation. The following table provides additional Prime Optical-supported parameters for SNC creation. For more information, refer to the ONS 15454 SONET or ONS NCS2K SONET user documentation.

Table 3-5: Prime Optical-Supported Parameters for SNC Creation

Name	Value	Description
upsrRevertiveMode	Revertive or nonrevertive	If the value is set to revertive, traffic reverts back to the working path when the conditions that diverted it to the protect path are repaired; otherwise, traffic remains on the protect path after the switch.
upsrRevertiveTime	Valid float value from 0.5 to 12, in increments of 12	If the SNC is revertive, you must set this parameter. This is the amount of time that elapses before traffic reverts back to the working path. The default value is 5 minutes.
		Prime Optical does not validate this value.
upsrSfBerLevel	3, 4, or 5	Set the thresholds for path protection, path-level signal failure, and bit error rate (only for STS SNCs).
		Prime Optical does not validate this value.
upsrSdBerLevel	5, 6, 7, 8, 9	Set the thresholds for path protection, path-level signal degrade, and bit error rate (only for STS SNCs).
		Prime Optical does not validate this value.
upsrSwitchOnPdiP	True or false	Set this to true if you want traffic to switch when an STS payload defect indicator is received (only

Name	Value	Description
		for STS SNCs).
goAndReturnRouting	True or false	Set this to true if you want to provision SNCP/path-protection routes in Go and Return fashion (ITU-T G.841); otherwise, SNCP/path-protection routes are provisioned in unidirectional fashion following GR-1400.
sncDescription	String	Description of the SNC. This value is represented in the userLabel field of the SubnetworkConnection_T structure.
VLAN	VLAN ID	NMS must provide one or more VLAN names. The value indicates the VLAN for Prime Optical to use for SNC creation on Ethernet equipment. An exception is thrown if the VLAN specified does not exist in the multilayer subnetwork of the source TP.
CustomerID	String	User-specified string representing the customer ID.
ServiceID	String	User-specified string representing the service ID.
sncSize	Int	User-specified size for WDM SNCs. Values from 1 to 6 apply to OCHNC and OCH trail SNCs; the default value is 6. Values from 7 to 46 apply to OCHCC SNCs; the value corresponds to that of the layerRate field.
createTunnelOnTransitNodes	True or false	Applicable for VT and VC SNCs with one or more transit nodes that do not have a tunnel or the existing tunnel is exhausted. This parameter specifies whether to create a tunnel on the transit nodes. For SNCs without transit nodes or with an existing tunnel on a transit node that has available bandwidth, this value is ignored. If the value specified is true, a tunnel is created and the SNC is routed using this tunnel. If the value specified is false, a new tunnel is not created. In this case, if a tunnel that is not full is available on the transit node, the circuit is routed through the tunnel; otherwise, the circuit is not routed through a tunnel.
createVTAggregation	True or false	Applicable for VT or VC SNCs with one or more transit nodes that do not have a VAP circuit or if the existing VAP circuit is exhausted. This parameter specifies whether to create a new VAP circuit on the transit nodes. If the value specified is true, a VAP circuit is created and SNC is routed using this new VAP circuit. If the value specified is false, a new VAP circuit

Name	Value	Description
		is not created.
sourceIsSTSGrooming	True or false	If the value is true, the source drop point of the VT or VC circuit is STS groomed for the VAP circuit.
		This attribute is valid only when the createVTAggregation value is set to true. sourceIsSTSGrooming and destinationIsSTSGrooming cannot both be set to true.
destinationIsSTSGrooming	True or false	If the value is true, the destination drop point of the VT or VC circuit is STS groomed for the VAP circuit.
		This attribute is valid only when the createVTAggregation value is set to true. sourceIsSTSGrooming and destinationIsSTSGrooming cannot both be set to true.
vapDestination	String	You can specify the string (ME name) representing the VAP destination drop point.
usePortlessTransmux	True or false	If the value is true, the portless transmux card is used for the circuit. The NMS provides information for portlessTransmuxMeName, portlessTransmuxSlot, and isSourceDS3MappedSTS; otherwise, an INVALID INPUT exception is thrown. If the value is false, the portless transmux card is not used as the intermediate point for the end-to-end STS-1 circuit.
portlessTransmuxMeName	String	ME name containing the high-density portless transmux card, which is used as a circuit constraint.
		This attribute is applicable only if usePortlessTransmux is true.
portlessTransmuxSlot	"/rack=1/shelf=1/slot=[1..6][12..17]"	NMS provides the physical location of the high-density portless transmux card.
		Information is required up to the slot level. This attribute is applicable only if usePortlessTransmux is true.
isSourceDS3MappedSTS	True or false	Indicates whether the source or destination is DS3 mapped.
		This attribute is applicable only if usePortlessTransmux is true.
vcatMemberId	Integer, 1 to 256	Used for VCAT SNC creation to identify every VCAT member L1 SNC uniquely. For normal CCAT circuits, this

Name	Value	Description
		value is ignored, if specified.
timeSlotRestriction	Integer, 1 to 192 for SONET and 1 to 64 for SDH.	You can select a specific STS/VC-4 when using autoroute with constraints. The STS/VC-4 is maintained around the ring. If the same STS/VC-4 is not available for the entire circuit route, a warning message is displayed.
includedProtectedME	String	The parameter must match the name of any ManagedElement which you have requested to be included in the protected path of an OCHNC or OCHCC circuit. For example: EMS = Cisco Systems/CTM ManagedElement = myNE Add the following parameter to configure a routing constraint: includedProtectedME = myNE
excludedProtectedME	String	Must match the name of any ManagedElement which you have requested to be excluded from the protected path of an OCHNC or OCHCC circuit. See the parameter includedProtectedME for the scenario.
overlayRing	String	Values can be: • True—Indicates that an Overlay Ring SNC is allowed. You can create an Overlay Ring SNC only by using CTP Inclusions to pass the complete route from the SNC to the Prime Optical Server. • False—Indicates that an Overlay Ring SNC is not allowed.
startOdu1	Integer	Number of initial ODUs used by the circuit (for AR-XP, AR-XPE, and AR-MXP cards).
endOdu1	Integer	Number of final ODUs used by the circuit (for AR-XP, AR-XPE, and AR-MXP cards).
startTs	Integer	Number of initial time slots in the startOdu number used by AR-XP and AR-MXP cards. For AR-XPE, startTs represents Start Odu0.
endTs	Integer	Number of final time slot in the endOdu number used by AR-XP and AR-MXP cards. For AR-XPE, endTs represents End Odu0.
ProtectedFrequency	String	AutoProvisioned or one of the CTP frequency values. AutoProvisioned forces the system to calculate and assign a frequency value automatically. CTP frequency values force the system to provision the specified frequency for the protected path. This attribute can be passed only if one of the

Name	Value	Description
		following conditions is satisfied: • SNC type is OCHCC in a non-GMPLS circuit with splitter protection. • SNC type is OCHCC in a GMPLS circuit with splitter protection. • SNC type is OCHCC in a GMPLS circuit with Y cable protection.
gmplsPowerOffsetUp	float	(Required parameter). The range is from -15.0 to 15.0. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsPowerOffsetDown	float	(Required parameter). The range is from -15.0 to 15.0. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsUserLabel	String	Required parameter) Only letters, numbers, and underscores (_) are allowed. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsPriority	String	Required Parameter. Valid values are: • N/A • High • Low • 1 • 2 • 3 • 4 • 5 • 6 • Unknown For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsCircuitDiversity	String	(Required parameter) Only letters and numbers are allowed and represent the circuit ID. For more information, see Cisco Prime Optical 10.7 User Guide.
gmplsWaveReq	booleaen	(Optional) Instructs the network control plane operation to force the provided CTP frequency. By default, the gmplsWaveReq parameter is False. ✎ The gmplsWaveReq parameter is not supported in Release 10.7 or Release 10.3.

Prime Optical does not support tolerableImpact (GradesOfImpact_T). Circuits created on the ONS 15327 and ONS 15454 and ONS NCS2K do not impact existing traffic.
Prime Optical does not support emsFreedomLevel (EMSFreedomLevel_T). Prime Optical has full control of SNC creation.

Throws

`globaldefs::ProcessingFailureException`

`EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.`

`EXCPT_INVALID_INPUT - Raised when a field of createData is invalid.`

EXCPT_ENTITY_NOT_FOUND - Raised when fields of createData reference objects that do not exist.

EXCPT_UNABLE_TO_COMPLY – Raised if the SNC cannot be created and activated because CTP conflicts with other SNC or NE does not support this SNC creation. This exception is also raised when Prime Optical cannot meet protection levels or there is no bandwidth available.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents creation of the SNC.

EXCPT_USERLABEL_IN_USE - Raised when the user label uniqueness constraint is not met.

Compliance
Prime Optical-defined.

3.9.17 CreateAndActivateSNC

Synopsis

```
void createAndActivateSNCFromUserLabel(
        in subnetworkConnection::SNCCreateData_T createData,
        in subnetworkConnection::GradesOfImpact_T tolerableImpact,
        in EMSFreedomLevel_T emsFreedomLevel,
        inout subnetworkConnection::TPDataList_T tpsToModify,
        out subnetworkConnection::SubnetworkConnection_T theSNC,
        out string errorReason)
     raises(globaldefs::ProcessingFailureException);
```

Description
This interface is identical to 3.9.16 createAndActivateSNCFromUserLabel except that Prime Optical assigns the name of the created subnetwork connection. The userLabel field in SNCCreateData_T can remain blank. Even if the user label is specified, Prime Optical ignores it and assigns a user label to the SNC.
For GMPLS SNCs, see the GMPLS Provisioning section.
Compliance
TMF-defined.

3.9.18 (Deprecated) createAndActivateProtectedOchccFromUserLabel

Synopsis

```
void createAndActivateProtectedOchccFromUserLabel(
in subnetworkConnection::SNCCreateData_T createData,
in OCHCCProtectedConstraints_T  ochccProtectedConstraints,
      in subnetworkConnection::GradesOfImpact_T tolerableImpact,
      in EMSFreedomLevel_T emsFreedomLevel,
      inout subnetworkConnection::TPDataList_T tpsToModify,
      out subnetworkConnection::SubnetworkConnection_T theSNC,
      out string errorReason)
    raises(globaldefs::ProcessingFailureException);
```

where:

```
  struct OCHCCProtectedConstraints_T {
```

```
        subnetworkConnection::ResourceList_T neTpProtectedInclusions;
        subnetworkConnection::ResourceList_T neTpSncProtectedExclusions;
    };
```

Description
This interface has the same behavior as createAndActivateSNCFromUserLabel, but is used only to create protected OCHCC circuits when it is necessary to specify route constraints for the protected path.
This API has been deprecated from CTM R9.0.
Parameters

Name	Type	Input/ Output	Description
ochccProtectedConstraints	multiLayerSubnetwork::OCHCCProtectedConstraints_T	In	Structure containing ME include/exclude route constraints.
other	—	—	Same as for createAndActivateSNCFromUserLabel.

Prime Optical supports the following layer rates:
- ONS 15454 SONET/ONS NCS2K SONET: LR_OCH_Transport_Unit
- ONS 15454 SDH/ONS NCS2K SDH: LR_OCH_Transport_Unit

Compliance
Prime Optical-defined.

3.9.19 addDrops

Synopsis

```
void addDrops(
        in globaldefs::NamingAttributes_T sncName,
        in globaldefs::NamingAttributesList_T zEnd)
    raises(globaldefs::ProcessingFailureException);
```

Description
This proprietary interface supports SNCs with multiple drops for the ONS 15310, ONS 15327, and ONS 15454. There are two steps to create an SNC with multiple drops:
Invoke the operation described in 3.9.16 createAndActivateSNCFromUserLabel to create a unidirectional SNC between two nodes.
Invoke this operation to add new drop points to an existing SNC.
The layer rate must match the SNC layer rate. This interface performs on a best-effort basis. If Prime Optical fails to add one or more TPs to the SNC, it generates an UNABLE_TO_COMPLY exception. The errorReason contains a list of CTPs that failed.
Depending on the card type, the terminology is CTP or FTP. For example, termination points on CE-MR-6 cards are CTPs when the cards are configured for manual mode. For CE-MR-6 cards configured for automatic mode, the termination points are FTPs.
Parameters

Name	Type	Input/ Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the subnetwork connection to which to add termination points (either CTPs or FTPs).
zEnd	globaldefs::NamingAttributesList_T	In	List of additional termination points (either CTPs or FTPs) to add to the SNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not refer to an SNC object, or layer rate of any CTP does not match the SNC layer.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY – Raised if the CTP is in use by other SNC. This exception is also raised when there is no bandwidth available.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost, which prevents the deactivation of the SNC.

Compliance

Prime Optical-defined.

3.9.20 deactivateAndDeleteSNC

Synopsis

```
void deactivateAndDeleteSNC(
      in globaldefs::NamingAttributes_T sncName,
      in subnetworkConnection::GradesOfImpact_T tolerableImpact,
      in EMSFreedomLevel_T emsFreedomLevel,
      inout subnetworkConnection::TPDataList_T tpsToModify,
      out subnetworkConnection::SubnetworkConnection_T theSNC,
      out string errorReason)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface provides a way to deactivate and then delete a subnetwork connection in one operation. The ports admin state is forced to ‘OOS, DSBLD’ (for SDH nodes ‘LOCKED, DISABLED’) upon successful deletion of non MPLS SNCs.

Prime Optical removes resources allocated to this SNC from each node. Successful completion of this asynchronous operation guarantees only delivery of the deletion request to Prime Optical. The NMS waits for an OBJECT_DELETION event for the SNC.

Prime Optical performs this operation in two steps:

- The first step is validation. This is performed synchronously. If successful, this operation returns successfully. If validation fails, the NMS receives an exception.
- After a successful validation, the second step is the actual deletion of resources from all NEs. This step is performed asynchronously. Prime Optical changes the SNC state to SNCS_DELETING and sends a notification to the NMS. Upon successful completion, Prime Optical generates an OBJECT_DELETION event. If Prime Optical loses connectivity to one or more nodes on the SNC, it aborts the cleanup and generates an SNCS_PARTIAL state change event. The NMS must reinvoke this method when connectivity to the node is restored. If the deletion fails but the state remains SNCS_ACTIVE, Prime Optical generates an AVC event. These SNC state change events are generated as part of the Prime Optical operation only. If another interface is used, Prime Optical cannot generate these events.

Parameters

Name	Type	Input/ Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the subnetwork connection to deactivate and delete.
tolerableImpact	subnetworkConnection::GradesOfImpact_T	In	Prime Optical ignores this parameter.
emsFreedomLevel	EMSFreedomLevel_T	In	Prime Optical ignores this parameter.
tpsToModify	subnetworkConnection::TPDataList_T	In/out	Prime Optical ignores this parameter.
theSNC	subnetworkConnection::SubnetworkConnection_T	Out	Deactivated and deleted subnetwork connection.
errorReason	string	Out	Specifies the deactivation and/or deletion error(s), if any.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not refer to an SNC object, or any field in tpsToModify is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the deactivation of the SNC.

Compliance

TMF-defined.

3.9.21 getVCATSNC

Synopsis

```
void MultiLayerSubnetworkMgr_I::getVCATSNC(
        in globaldefs::NamingAttributes_T vcatName,
        out subnetworkConnection::VCAT_T vcat)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the VCAT structure for the VCAT SNC whose name is supplied as an argument. The VCAT SNCs are supported on the following equipment types:

- ML equipment (ML1000 and ML100T)
- FCMR
- ADM_10G

Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
vcat	subnetworkConnection::VCAT_T	Out	VCAT structure.

Prime Optical support for VCAT_T is as follows:

- globaldefs::NamingAttributes_T vcatName
  Supported. This field will contain two tuples. The first tuple contains the value for the EMS. The second tuple contains the VCAT name, which is the same as the nativeEMSName. The value of the name field in the last tuple is VCATSubnetworkConnection.
- string userLabel
  - Supported.
- string nativeEMSName
  - Supported. Name supplied by the user at creation time will be reported.
- string owner
  - Not supported.
- VCATState_T vcatState
  - Supported. The state of the VCAT circuit is derived from the states of the member circuits in the same way as it is derived for CCAT SNCs. The values supported are:
  - VCATState_T.VCATS_NONEXISTENT
  - VCATState_T.VCATS_PENDING
  - VCATState_T.VCATS_ACTIVE
  - VCATState_T.VCATS_PARTIAL
  - VCATState_T.VCATS_DELETING
- globaldefs::ConnectionDirection_T direction
  - Supported. This field represents the direction of the VCAT and not the member L1 SNCs.
  - Prime Optical supports CD_UNI (unidirectional) and CD_BI (bidirectional). Current release supports only CD_BI.
- boolean isSymmetrical
  - Supported. Current release supports only symmetric VCAT SNCs.
  - True if the rate/capacity is same in both directions.
  - False if the rate/capacity is different in each direction.
- transmissionParameters::VCATLayerRate_T txRate
  - Supported. This field represents the total capacity the VCAT SNC can carry in the forward direction. If the VCAT is symmetrical, it represents the capacity both in the forward and backward directions. To represent the total capacity of a VCAT SNC, VCATLayerRate_T contains the following information:
  - LayerRate_T memberRate: Layer rate of the member SNCs.
  - Unsigned long numberOfMembers: Number of members for the given VCAT SNC.

  For a list of supported layer rates, see 2.2.1 Layer Rate.
- transmissionParameters::VCATLayerRate_T rxRate
  - Supported. This field represents the total capacity the VCAT SNC can carry in the backward direction. If the VCAT is symmetrical, this field is ignored. For a list of supported layer rates, see 2.2.1 Layer Rate.
- transmissionParameters::LayerRate_T memberRate
  - Supported. This field represents the rate of each member circuit.
- globaldefs::NamingAttributes_T aEnd
  - Supported. This is an FTP representing the port at the aEnd.
- globaldefs::NamingAttributes_T zEnd
  - Supported. This is an FTP representing the port at the zEnd. This field is empty for open-ended VCAT.
- SNCType_T sncType
  - Not supported.

- subnetworkConnection::LCASMode_T mode
  - Supported. Prime Optical supports the following values:
  - LCASMode_T. LCASMODE _NONE
  - LCASMode_T. LCASMODE _SWLCAS
  - LCASMode_T. LCASMODE _LCAS

- subnetworkConnection::RoutingMode_T routingMode
  - Supported. Prime Optical supports the following values:
  - RoutingMode_T. ROUTE_COMMONFIBER
  - RoutingMode_T. ROUTE_INDEPENDENTFIBER
- globaldefs::NVSList_T additionalInfo
  - Supported. CustomerID and ServiceID are reported if they are available. The openVcat field reports values of true or false to show whether the SNC is an open-ended VCAT.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not refer to a VCAT SNC object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

Compliance

Prime Optical-defined.

3.9.22 getAllVCATSNCs

Synopsis

```
void MultiLayerSubnetworkMgr_I::getAllVCATSNCs(
        in globaldefs::NamingAttributes_T subnetName,
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out subnetworkConnection::VCATList_T vcatList,
        out subnetworkConnection::VCATIterator_I vcatIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns all the VCAT SNCs in the specified multilayer subnetwork. This interface returns only the VCAT SNCs.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
connectionLayerRateList	transmissionParameters::VCATLayerRateList_T	In	List of the VCAT SNCs to report with specified number of members and memberRate. If

Name	Type	Input/ Output	Description
			an empty list is specified, all VCAT SNCs of bandwidth are reported.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
vcatList	subnetworkConnection::VCATList_T	Out	First batch of VCAT SNCs.
vcatIt	subnetworkConnection::VCATIterator_I	Out	Iterator to retrieve the remaining VCAT SNCs.

For details about the VCAT_T structure, see 3.9.21 getVCATSNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.9.23 getAllVCATSNCNames

Synopsis

```
void MultiLayerSubnetworkMgr_I::getAllVCATSNCNames(
        in globaldefs::NamingAttributes_T subnetName,
        in transmissionParameters::LayerRateList_T connectionRateList,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the names of all the VCAT SNCs in the specified multilayer subnetwork. This interface returns only the VCAT SNC names.

Parameters

Name	Type	Input/ Output	Description

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnetwork.
connectionLayerRateList	transmissionParameters::VCATLayerRateList_T	In	List of the VCAT SNCs to report with specified number of members and member rate. If an empty list is specified, all VCAT SNCs of bandwidth are reported.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of VCAT SNC names.
nameIt	globaldefs::NamingAttributesIterator_T	Out	Iterator to retrieve the remaining VCAT SNC names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetName does not reference a multiLayerSubnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.9.24 getAllVCATMemberSNCs

Synopsis

```
void MultiLayerSubnetworkMgr_I::getAllMemberVCATSNCs(
        in globaldefs::NamingAttributes_T vcatName,
        in boolean isTransmit,
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T sncList,
```

```
        out subnetworkConnection::SubnetworkConnectionIterator_I sncIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns all of the member SNCs for the VCAT SNC.
Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
isTransmit	Boolean	In	If the isTransmit boolean is set to true, only transmit-side member SNCs are returned. If set to false, only receive-side member SNCs are returned. If the VCAT is symmetrical, the isTransmit value is ignored and all bidirectional SNCs are returned.
how_many	unsigned long	In	Maximum number of SNCs to report in the first batch.
sncList	subnetworkConnection::SubnetworkConnectionList_T	Out	First batch of SNCs.
sncIt	subnetworkConnection::SNCIterator_I	Out	Iterator to retrieve the remaining SNCs.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not reference a VCAT object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
Prime Optical-defined.

3.9.25 getAllVCATMemberSNCNames

Synopsis

```
void MultiLayerSubnetworkMgr_I::getAllVCATMemberSNCNames(
        in globaldefs::NamingAttributes_T vcatName,
        boolean isTransmit,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
```

```
raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns the names of all member SNCs for a given VCAT SNC.
Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
isTransmit	boolean	In	If the isTransmit boolean is set to true, only transmit-side member SNCs are returned. If set to false, only receive-side member SNCs are returned. If the VCAT is symmetrical, the isTransmit value is ignored and all bidirectional SNCs are returned.
how_many	unsigned long	In	Maximum number of member VCAT SNCs to report in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of member VCAT SNC names.
nameIt	globaldefs::NamingAttributesIterator_T	Out	Iterator to retrieve the remaining member VCAT SNC names.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not reference a VCAT object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
Prime Optical-defined.

3.9.26 createVCAT

Synopsis

```
void MultiLayerSubnetworkMgr_I::createVCAT(
        in subnetworkConnection::VCATSNCCreateData_T createData,
        out subnetworkConnection::VCAT_T vcat,
        out string errorReason)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface attempts to create a VCAT SNC.
Prime Optical generates an object creation event for the VCAT SNC that was created and for the individual Layer 1 SNCs.
Parameters

Name	Type	Input/ Output	Description
createData	subnetworkConnection::VCATSNCCreateData_T	In	Structure describing the VCAT SNC to create.

Name	Type	Input/ Output	Description
vcat	subnetworkConnection::VCAT_T	Out	VCAT SNC created.
errorReason	String	Out	Reason for the failure.

Prime Optical support for VCATSNCCreateData_T structure is as follows:

- string userLabel
  - Supported. Value assigned to this field is the name of the new VCAT.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of all VCATs created by Prime Optical. Prime Optical does not guarantee uniqueness if any other tool is used for VCAT SNC creation.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Prime Optical supports only CD_BI (bidirectional).
- boolean isSymmetrical
  - Supported. In the current release, only symmetric VCAT SNCs are supported. If set to true, all members are bidirectional L1 SNCs, and the VCAT layer rate is the same for both directions. If set to false, all members are unidirectional L1 SNCs, and the supported VCAT layer rate on either direction can be different.
- subnetworkConnection::LCASMode_T mode
  - Supported. Prime Optical supports the following values:
  - LCASMode_T.MODE_NONE
  - LCASMode_T.MODE_SWLCAS
  - LCASMode_T.MODE_LCAS
- subnetworkConnection::RoutingMode_T routingMode
  - Supported. Prime Optical supports the following values:
  - RoutingMode_T.COMMONFIBER_ROUTED
  - RoutingMode_T.INDEPENDENTFIBER_ROUTED
- subnetworkConnection::TPData_T aEnd
  - Supported. Prime Optical expects this field to be an FTP representing the information up to port level of the card involved in VCAT SNC.
- subnetworkConnection::TPData_T zEnd
  - Supported. Prime Optical expects this field to be an FTP representing the information up to the port level of the card involved in the VCAT SNC. This field is empty for open-ended VCAT.
- subnetworkConnection::SNCCreateDataList_T sncCreateDataList
  - Supported. At least one member L1 SNC must be created at the time of VCAT SNC creation. Prime Optical expects user to specify the create data information required to create the L1 SNC(s) in this data structure. For complete information on SNCCreateData_T structure, see 3.9.16 createAndActivateSNCFromUserLabel. For every L1 SNC, you must specify an additionalCreationParamemter vcatMemberId, which must contain a unique value from 1 to 256.
- globaldefs::NVSList_T additionalCreationInfo
  - This list can be used to provide additional parameters for VCAT creation. Prime Optical supports the following values:

Name	Value	Description
vcatDescription	String	Description of the VCAT.
CustomerID	String	User-specified string representing the customer ID.

Name	Value	Description
ServiceID	String	User-specified string representing the service ID.
useExtendedBuffer	Boolean (True/False)	For differential delay.
openVcat	Boolean (True/False)	Indicates that an open-ended VCAT must be created.

- subnetworkConnection::ResourceList_T neTpInclusions
  - Supported. You must provide a list of managed element names or the PTP names at the endpoints of the topological link. For complete manual routing, you must provide the list of all CTPs of the links (OCH trail links, if any) to which the circuit must be routed.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when a field of createData is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when fields of createData reference objects that do not exist.

EXCPT_UNABLE_TO_COMPLY – Raised if the VCAT SNC cannot be created and activated because FTP conflicts with another VCAT SNC or NE does not support this VCAT SNC creation. This exception is also raised when Prime Optical cannot meet protection levels or there is no bandwidth available.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents creation of the VCAT SNC.

EXCPT_USERLABEL_IN_USE - Raised when the user label uniqueness constraint is not met.

Compliance

Prime Optical-defined.

3.9.27 deleteVCAT

Synopsis

```
void deleteVCAT(
        in globaldefs::NamingAttributes_T vcatName)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface deletes the VCAT specified in the input. Prime Optical generates an object deletion event for the VCAT SNC that was deleted and for the individual Layer 1 SNCs that were deleted.

Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not reference a VCAT object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when deletion of the VCAT SNC fails.

Compliance

Prime Optical-defined.

3.9.28 addMemberSNCsToVCAT

Synopsis

```
void addMemberSNCsToVCAT(
        in globaldefs::NamingAttributes_T vcatName,
        in subnetworkConnection::SNCCreateDataList_T sncData,
        out subnetworkConnection::VCAT_T vcat)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface adds a member SNC to the VCAT that is specified in the input. Prime Optical generates an attribute value change event for the VCAT SNC that was modified due to the addition of the new SNC(s), and an object creation event for the member L1 SNCs that were created.

For each new member, you must specify an additional creation parameter called "vcatMemberId," which is a unique number from 1 to 256. The value that you choose cannot be in use by any other members that belong to the VCAT SNC.

Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
sncData	subnetworkConnection::SNCCreateDataList_T	In	Structure describing the SNC(s) to create and add to the VCAT.
vcat	subnetworkConnection::VCAT_T	Out	The augmented VCAT SNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not reference a VCAT object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when addition of a member SNC to the VCAT fails.

Compliance

Prime Optical-defined.

3.9.29 deleteMemberSNCsFromVCAT

Synopsis

```
void deleteMemberSNCsFromVCAT(
        in globaldefs::NamingAttributes_T vcatName,
        in globaldefs::NamingAttributesList_T sncName,
        out subnetworkConnection::VCAT_T vcat)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface deletes a member SNC from the VCAT specified in the input. Prime Optical generates an attribute value change event for the VCAT SNC that was modified due to the deletion of the member SNC, and an object deletion event for the member L1 SNC that was deleted.

Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
sncName	globaldefs::NamingAttributesList_T	In	Name of the SNC(s) to delete and remove from the VCAT.
vcat	subnetworkConnection::VCAT_T	Out	Modified VCAT SNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName or sncName does not reference a VCAT/SNC object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName or sncName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when deletion of member SNC(s) from the VCAT fails.

Compliance

Prime Optical-defined.

3.9.30 deleteAllMemberSNCFromVCAT

Synopsis

```
void deleteAllMemberSNCFromVCAT(
        in globaldefs::NamingAttributes_T vcatName,
        in boolean isForward)
     raises( globaldefs::ProcessingFailureException );
```

Description

This interface deletes all member SNCs in the VCAT group that is specified during input. If boolean value is set to true, all member SNCs in the transmit direction are deleted. If boolean value is set to false, all member SNCs in the receive direction are deleted. For symmetrical VCAT, this value is ignored; all member circuits in both transmit and receive directions are deleted. Prime Optical generates an attribute value change for the VCAT SNC modified due to the deletion of member SNC(s), and an object deletion event for the member L1 SNCs deleted.

Parameters

Name	Type	Input/ Output	Description
vcatName	globaldefs::NamingAttributes_T	In	Name of the VCAT SNC.
isForward	Boolean	In	Boolean variable used to determine the direction in which the member SNCs of the VCAT are deleted.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when vcatName does not reference a VCAT object.

EXCPT_ENTITY_NOT_FOUND - Raised when vcatName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when deletion of member SNC(s) from the VCAT fails.

Compliance
Prime Optical-defined.

3.9.31 getValidSubnetworkConnectionsForMerge

Synopsis

```
void getValidSubnetworkConnectionsForUpgrade(
        in multiLayerSubnetwork::SncUpgradeInfo_T details,
        in multiLayerSubnetwork::SncFilterCriteria_T filterCriteria,
        in globaldefs::NamingAttributes_T masterSncName,
        in globaldefs::NamingAttributesList_T candidateSncList,
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T mergeableSncList,
        out subnetworkConnection::SNCIterator_I sncIt )
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface is used to get the list of SNCs for the Merge operation. The SncFilterCriteria specifies the SNCs that are requested. These SNCs are lined up. The master SNC name specifies the name of the SNC with which the candidates will merge. The how_many parameter provides the list of SNCs that will be returned.

✎ The valid SncUpgradeInfo value supported by GateWay/CORBA is Merge. The valid SncFilterCriteria value supported by GateWay/CORBA is LINED_UP. The available values that will be supported in the future are listed in the ENUM definitions in the following table.

Parameters

Name	Type	Input/ Output	Description
details	multiLayerSubnetwork::SncUpgradeInfo_T	In	Upgrade info for which the list of SNCs is requested. Available values are: • Merge • Reconfigure ✎ Prime Optical supports only Merge.
filterCriteria	multiLayerSubnetwork::SncFilterCriteria_T	In	The filter criteria can be used to specify if the requested SNCs are LINED_UP or nonfiltered. ✎ Prime Optical supports only LINED_UP.
masterSncName	globaldefs::NamingAttributes_T	In	This parameter is used to specify the name of the SNC to merge or upgrade with the following candidate SNCs.

Name	Type	Input/ Output	Description
candidateSncList	globaldefs::NamingAttributesLis t_T	In	This parameter gives a list of SNCs to merge with the master SNC.
how_many	unsigned long	In	This parameter specifies the number of SNCs to pass and the SNCs that are valid for the merge operation with the master SNC.
mergeableSncList	subnetworkConnection::Subnetw orkConnectionList_T	Out	This parameter gives the list of SNCs that qualify for a merge.
sncIt	subnetworkConnection::SNCIter ator_I	Out	This parameter is used to iterate the list of SNCs returned by the API.

```
enum SncUpgradeType_T {
        MERGE,
        RECONFIGURE
    };

enum SncOperationType_T {
        STRICT_MERGE,
        LOOSE_MERGE,
        NONE
    };
enum SncFilterCriteria_T {
        LINED_UP,
        NO_FILTER
    };

struct SncUpgradeInfo_T {
        SncUpgradeType_T upgradeType;
        SncOperationType_T operationType;
    };
```

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

3.9.32 upgrade

Synopsis

```
void upgrade(
       in multiLayerSubnetwork::SncUpgradeInfo_T details,
       in globaldefs::NamingAttributes_T masterSncName,
       in globaldefs::NamingAttributesList_T upgradableSncNameList,
```

```
        out string resultString )
        raises(globaldefs::ProcessingFailureException);
```

Description
This interface is used to merge or configure a list of SNCs to the SNC reported in the masterSncName. This interface supports STRICT MERGE and LOOSE MERGE as operation types from the OSS. The LOOSE MERGE option in SncUpgradeInfo_T is available only if a strict operation failed and the OSS receives a processing failure exception with EXCPT_STRICT_MERGE_FAILED.
There is no restriction in GateWay/CORBA to prevent a LOOSE MERGE operation from the OSS during a normal call.

Parameters

Name	Type	Input/ Output	Description
details	multiLayerSubnetwork::Snc UpgradeInfo_T	In	Upgrade info for which the list of SNCs are requested. Available values are: • Merge • Reconfigure
masterSncName	globaldefs::NamingAttributes_T	In	This parameter is used to specify the name of the SNC to merge or reconfigure with the following
upgradableSncNameList	globaldefs::NamingAttributesList_T	In	This parameter gives the list of SNCs to merge or reconfigure
resultString	string	Out	This parameter gives the result of the operation in human-readable

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_STRICT_MERGE_FAILED – Raised when a strict merge operation from the OSS fails. The OSS uses this exception to issue a loose merge call.

Compliance
Prime Optical-defined.

3.9.33 createRoll

Synopsis

```
void createRoll(
        in multiLayerSubnetwork::RollCreateData_T rollCreateData)
    raises(globaldefs::ProcessingFailureException);
```

Description
Prime Optical supports roll creation on optical CTPs. Electrical CTPs and FTPs are not supported for bridge and roll creation.

Prime Optical generates an Object Creation event for the roll created and subnetwork connections created as part of the roll creation. The subnetwork connections that are created as part of the roll creation will be named as nativeEMSName of the SNC appended with _ROLL** where ** is a number.
The creation of bridge and roll is an asynchronous process. If the operation fails, Prime Optical cleans up all resources. For example, if Prime Optical fails to create a bridge and roll on one node, it also cleans up resources on other nodes.

Parameters

Name	Type	Input/ Output	Description
rollCreateData	multiLayerSubnetwork::RollCreateData_T	In	Structure describing the roll to create.

For details about the Roll_T structure, see 3.8.28 getAllRolls.
Prime Optical support for RollCreateData_T is as follows:

- multiLayerSubnetwork::RollType_T rollType
  - Supported.
- multiLayerSubnetwork::RollMode_T rollMode
  - Supported.
- globaldefs::NamingAttributesList_T sncNames
  - Supported. This refers to the name of the SNC to roll partially or fully. If two SNCs are specified, the first SNC will be the RollFromSNC and the second SNC will be the RollToSNC.
- globaldefs::NamingAttributesList_T fixCTPs
  - Supported. This refers to the fully qualified name of the CTP of the SNC. This CTP is the fixed point of the roll. In cases where the rollType is DUAL, there will be two fixed points in the roll. The first fixCTP indicates the first fixed point in the roll object and the second fixCTP indicates the second fixed point of the roll.
- globaldefs::NamingAttributesList_T rollFromCTP
  - Supported. This refers to the fully qualified name of the CTP of the SNC. This CTP is either the source or destination or any other associated CTP of the SNC. In cases where the rollType is DUAL, the first element in the array indicates the first rollFromCTP point and the second element in the array indicates the second rollFromCTP point. If the rollFromCTP can be determined from the fixed CTP, the value is ignored.
- globaldefs::NamingAttributesList_T rollToCTP
  - Supported. This refers to the fully qualified name of the CTP where the roll occurs. In cases where the rollType is DUAL, the first element in the array indicates the first rollToCTP point and the second element in the array indicates the second rollToCTP point.
- subnetworkConnection::ResourceList_T neTpInclusions
  - Supported. Prime Optical expects a list of managed element names or PTP names of the endpoint of a topological link or CTP name pair representing the endpoints of the span to include in the subnetwork connection route. For complete manual routing, Prime Optical expects the list of all CTPs of the links (OCH Trail circuit links, if any) to which the circuit must be routed. If the OSS specifies a CTP pair in the neTpInclusions, the entire path for the subnetwork connection must be specified.

- subnetworkConnection::ResourceList_T neTpSncExclusions
  - Supported. Prime Optical expects a list of managed element names or PTP names of the endpoint of a topological link representing the endpoints of the span to exclude in the subnetwork connection route.
- subnetworkConnection::StaticProtectionLevel_T staticProtectionLevel

  - Prime Optical supports the following for the subnetwork connection that is created as part of the roll creation:
  - UNPROTECTED
  - FULLY_PROTECTED
  - HIGHLY_PROTECTED

✎ Prime Optical does not support other values.

- subnetworkConnection::ProtectionEffort_T protectionEffort
  - If the NMS requests unprotected SNC on a BLSR ring, Prime Optical creates the SNC. The same is true for unprotected SNC on 1+1 protection.
  - If the NMS requests EFFORT_SAME_OR_WORSE or EFFORT_WHATEVER, Prime Optical expects a value of staticProtectionLevel of UNPROTECTED. These values indicate that Prime Optical must create the PCA. Prime Optical support for Roll_T is described in 3.8.28 getAllRolls.

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised when sncName does not refer to an SNC object, or any field in rollCreateData is invalid, or the data does not conform to roll creation rules.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the deactivation of the SNC.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance

Prime Optical-defined.

Scenarios: Roll Creation

Figure 3-1: Example 1: Roll Source or Destination of an SNC—Part 1

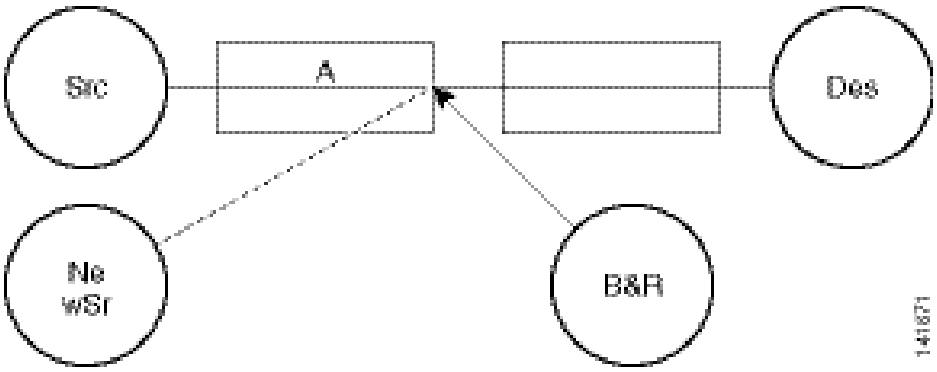


Figure 3-2: Example 2: Roll Source or Destination of an SNC—Part 2

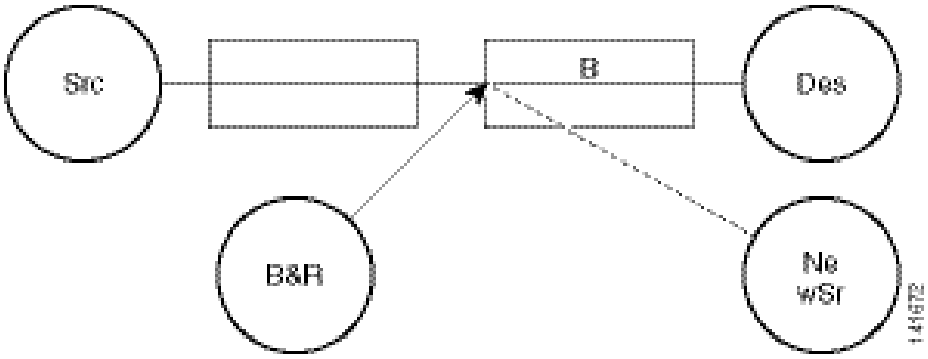
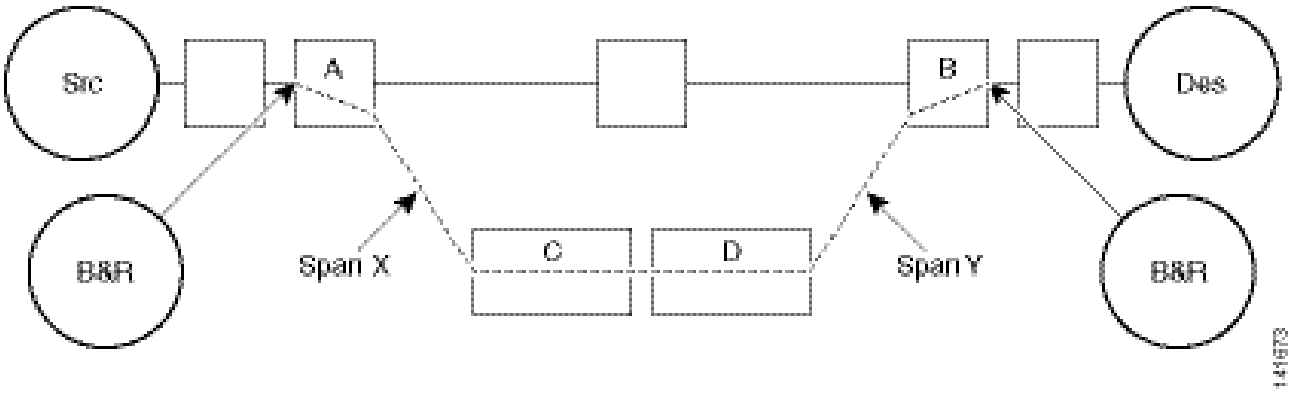


Figure 3-3: Example 3: Roll Intermediate Managed Elements or Topological Links



3.9.34 deleteRoll

Synopsis

```
void deleteRoll(
        in globaldefs::NamingAttributes_T rollname)
    raises(globaldefs::ProcessingFailureException);
```

Description

Prime Optical generates an object deletion event for the deleted roll object. Prime Optical also generates object deletion events for the SNCs that are deleted.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the deletion of the roll.

Compliance

Prime Optical-defined.

3.9.35 cancelRoll

Synopsis

```
void cancelRoll(
        in globaldefs::NamingAttributes_T rollName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface cancels a roll task when the rollValidSignal is not received. For more information on canceling a roll task, see the network element documentation.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll to cancel.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the cancellation of the roll operation.

EXCPT_UNABLE_TO_COMPLY – Raised when the roll cannot be created.

Compliance

Prime Optical-defined.

3.9.36 forceValidSignal

Synopsis

```
void forceValidSignal(
        in globaldefs::NamingAttributes_T rollName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface runs on both automatic and manual rolls.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll on which to force a valid signal.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.9.37 completeRoll

Synopsis

```
void completeRoll(
        in globaldefs::NamingAttributes_T rollName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface must run on manual rolls (RollMode_T.MANUAL).
Prime Optical generates a state change event upon completion of the roll.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll to complete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the deactivation of the SNC.

Compliance

Prime Optical-defined.

3.9.38 finishRoll

Synopsis

```
void completeRoll(
        in globaldefs::NamingAttributes_T rollName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface is invoked to finish rolls. When the roll is finished, it is cleared from the rolls list and the new rolled SNC becomes active.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of the roll to finish.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost and this prevents the deactivation of the SNC.

Compliance

Prime Optical-defined.

3.9.39 getAllRolls

Synopsis

```
void getAllRolls(
        in globaldefs::NamingAttributes_T subnetName,
        in unsigned long how_many,
        out multiLayerSubnetwork::RollList_T rollList,
        out multiLayerSubnetwork::RollIterator_I rollIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns a list of rolls within the subnetwork whose name is passed as a parameter.

Parameters

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnet for which all rolls are reported.
how_many	unsigned	In	Maximum number of rolls to report in the first batch.
rollList	multiLayerSubnetwork::RollList_T	Out	First batch of rolls.
rollIt	multiLayerSubnetwork::RollIterator_I	Out	Iterator to retrieve remaining rolls.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetwork name does not refer to a subnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetwork references an object that does not exist.

Compliance

Prime Optical-defined.

3.9.40 getAllRollNames

Synopsis

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

Description

This interface returns a list of roll names within the subnetwork whose name is passed as a parameter.

Parameters

Name	Type	Input/ Output	Description

Name	Type	Input/ Output	Description
subnetName	globaldefs::NamingAttributes_T	In	Name of the subnet for which all rolls are reported.
how_many	unsigned	In	Maximum number of rolls to report in the first batch.
nameList	globaldefs::NamingAttributesList	Out	First batch of roll names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining roll names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when subnetwork name does not refer to a subnetwork object.

EXCPT_ENTITY_NOT_FOUND - Raised when subnetwork references an object that does not exist.

Compliance

Prime Optical-defined.

3.9.41 getAssociatedRollNamesForSNC

Synopsis

```
void getAssociatedRollNamesForSNC(
        in globaldefs::NamingAttributes_T sncName,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList rollNameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns all the rolls associated with the SNC.

Parameters

Name	Type	Input/Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the SNC whose associated roll objects are required.
how_many	unsigned long	In	Maximum number of rolls to report in the first batch.
rollNameList	globaldefs::NamingAttributesList	Out	First batch of roll names.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator to retrieve the remaining roll names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not refer to an SNC object.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an object that does not exist.

Compliance

Prime Optical-defined.

3.9.42 getAssociatedSNCNamesForRoll

Synopsis

```
void getAssociatedSNCNamesForRoll(
        in globaldefs::NamingAttributes_T rollName,
        out globaldefs::NamingAttributes_T rollFromSnc,
        out globaldefs::NamingAttributes_T rollToSnc)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the rollFrom SNC name and rollTo SNC name for a roll whose name is passed as a parameter. For circuit bridge-and-roll operations, the Prime Optical client Rolls table contains roll-from circuit and roll-to circuit attributes. The getAssociatedSNCNamesForRoll method returns these attributes for a given circuit roll.

Parameters

Name	Type	Input/ Output	Description
rollName	globaldefs::NamingAttributes_T	In	Name of roll whose associated SNC will be reported.
rollFromSnc	globaldefs::NamingAttributes_T	Out	Name of the rollFrom SNC.
rollToSnc	globaldefs::NamingAttributes_T	Out	Name of the rollTo SNC.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when rollName does not refer to a roll object.

EXCPT_ENTITY_NOT_FOUND - Raised when rollName references an object that does not exist.

Compliance

Prime Optical-defined.

3.9.43 createAndActivateSNCOnDRI

Synopsis

```
void createAndActivateSNCOnDRI(
        in subnetworkConnection::DRISNCCreateData_T createData,
        in subnetworkConnection::GradesOfImpact_T tolerableImpact,
        in EMSFreedomLevel_T emsFreedomLevel,
        inout subnetworkConnection::TPDataList_T tpsToModify,
        out subnetworkConnection::SubnetworkConnection_T theSNC,
        out string errorReason)
    raises(globaldefs::ProcessingFailureException);
```

Description

This structure was introduced to provide BLSR-DRI protection support. This structure is not defined by TMF. This structure is similar to createAndActivateSNCOnUserLabel (see 3.9.14 getSNCsByUserLabel, for more details) except that a new field, driNode, has been added.

Parameters

Name	Type	Input/ Output	Description
createData	subnetworkConnection::DRISNCCreateData_T	In	New structure describing the SNC to create and activate. This is similar to SNCCreateData_T except that DRI-specific information has been added.
tolerableImpact	subnetworkConnection::GradesOfImpact_T	In	Maximum tolerable impact allowed.
emsFreedomLevel	EMSFreedomLevel_T	In	Maximum level of freedom allowed to the EMS to perform the creation and activation.
tpsToModify	subnetworkConnection::TPDataList_T	In/out	Prime Optical ignores this parameter.
theSNC	subnetworkConnection::SubnetworkConnection_T	Out	Resulting SNC, with sncState and name set. The name is set to the user label supplied in createData structure.
errorReason	String	Out	Specifies the creation and/or activation error(s), if any.

The structure of DRISNCCreateData_T is as follows:

```
struct DRISNCCreateData_T
{
    subnetworkConnection::SNCCreateData_T createData,
    subnetworkConnection::DRINodesList_T nodeList,
};

typedef sequence <DRINodes_T> DRINodesList_T;
```

The structure of DRINodes_T is as follows:

```
struct DRINodes_T
{
    globalDefs::DRIType_T type,
    boolean isSecPCA;
    DriNodePairList_T nodePairList;
    globaldefs::NamingAttributesList requiredRings;
```

```
}

typedef sequence <DRINodePair_T> DRINodePairList_T;
struct DRINodePair_T
{
    globaldefs::NamingAttributes_T PrimaryME;
    globaldefs::NamingAttributes_T SecondaryME
}
```

Prime Optical support for DriNode_T structure is as follows:

- type
  - This field defines integer values that categorize the BLSR-DRI as traditional or nontraditional BLSR-DRI. In Prime Optical type = 0 means it is a dummy node, 1 = traditional. and 2 = nontraditional.
- isSecPCA
  - This field indicates whether or not the RIP option is supported.
- DriNodePair_T
  - This structure holds the primary and secondary nodes. The sDriNodePair structure is as follows:
  - globaldefs::NamingAttributes_T PrimaryME—Defines the primary DRI node.
  - globaldefs::NamingAttributes_T SecondaryME —Defines the secondary DRI node.
  - globaldefs::NamingAttributesList requiredRings—Holds the ring information. The first name value tuple represents EMS instance and the second one holds the BLSR ring ID.

Compliance

Prime Optical-defined.

3.9.44 addManagedElement

Synopsis

```
void addManagedElement(in MEAdditionData_T meData)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows the NMS to add a new managed element to the specified subnetwork.
If the operation fails, an exception is returned, and the managed element is not added to the subnetwork.
If the operation succeeds, a managed element is added to the specified subnetwork. The EMS populates the ME communication state, its supported layer rates, and emsInSyncState when connectivity to the ME is established.
Prime Optical generates an NT_OBJECT_CREATION event upon successful addition of a managed element.

Parameters

Name	Type	Input/ Output	Description
meData	MEAdditionData_T	In	Input data required for the addition of the NE.

The structure of MEAdditionalData_T is as follows:

```
struct MEAdditionData_T
{
globaldefs::NamingAttributes_T managedElementName;
globaldefs::NamingAttributes_T containingSubnetworkName;
in string userLabel;
```

```
in string nativeEMSName;
in string owner;
in string location;
in string version;
in string productName;
globaldefs::NVSList_T additionalInfo;
};
```

The description of each field is as follows:

- globaldefs::NamingAttributes_T managedElementName—Name of the managed element to add.
  - name="ManagedElement"
  - value=Managed Element
- containingSubnetworkName—Name of the subnetwork that contains the managed element.
  - name="subnetworkName"
  - value=subnetwork name
- string userLabel—User-specified label of the managed element.
- string nativeEMSName—nativeEMSName of the managed element to add.
- string owner—Owner of the managed element.
- string location—Location of the managed element.
- string version—The NMS can specify the ME version; however, this value is overwritten when the ME is put in service and the EMS retrieves the actual version information.
- string productName—Product name of the managed element.
- globaldefs::NVSList_T additionalInfo—Parameters that must be specified to the EMS to add the ME; for example, IP address, ME username, password, and any other vendor-specific proprietary data.

Prime Optical support for addManagedElement is as follows:

- globaldefs::NamingAttributes_T managedElementName: Not supported
- globaldefs::NamingAttributes_T containingSubnetworkName: Supported
- string userLabel: Not supported
- string nativeEMSName: Not supported
- string owner: Not supported
- string location (location of the managed element): Supported
- string version: Not supported
- string productName: Not supported
- globaldefs::NVSList_T additionalInfo: Supported

The structure of Additional Info is as follows:

Name	Value	Description	Mandatory or Optional
NEModel	modelname	NE model name	Mandatory (see the following values)
IPAddress	ipaddress	IP address of the network element	Mandatory (for non-ONS 15216 NEs)
NPName	network partition name	Name of the network partition under which to add the NE	Mandatory

Name	Value	Description	Mandatory or Optional
NESYSID	nesysid	NE ID for the NE	Mandatory (for ONS 15216 NEs)
NEUsername	username	Username	Mandatory (for non-ONS 15216 NEs)
NEPassword	password	Password	Mandatory (for non-ONS 15216 NEs)
Description	description	Description	Optional
Community String	private/public	SNMP community string	Mandatory (see the following values)
Vendor	vendor	NE version name	Optional

The parameter name “Community String” in additionalInfo can have the following values:

- For NE types ONS 15216 EDFA2 and ONS 15216 EDFA3, the Community String value is private.
- For OADM NE types, the Community String parameter is not required.
- For the remaining NE types, the Community String value is public.

The parameter name “NEModel” in additionalInfo can have the following values:

- "CISCO_ONS_15327_UPSR_ADM"
- "CISCO_ONS_15327_TSA_TERMINAL"
- "CISCO_ONS_15327_BLSR_ADM"
- "CISCO_ONS_15327_HYBRID"
- "CISCO_ONS_15327_Other"
- "CISCO_ONS_15327_Unknown"
- "CISCO_ONS_15454_UPSR_ADM"
- "CISCO_ONS_15454_BLSR_ADM"
- "CISCO_ONS_15454_TSA_TERMINAL"
- "CISCO_ONS_15454_OPTICAL_ADM"
- "CISCO_ONS_15454_HUB"
- "CISCO_ONS_15454_DWDM_HYBRID"
- "CISCO_ONS_15454_TERMINAL_HUB_EAST"
- "CISCO_ONS_15454_TERMINAL_HUB_WEST"
- "CISCO_ONS_15454_DWDM_TERMINAL"
- "CISCO_ONS_15454_PASSIVE_LINE_SITE"
- "CISCO_ONS_15454_ACTIVE_LINE_SITE"
- "CISCO_ONS_15454_ROADM"
- "CISCO_ONS_15454_TERMINAL_OADM_EAST"
- "CISCO_ONS_15454_TERMINAL_OADM_WEST"
- "CISCO_ONS_15454_DWDM_HUB_NODE"
- "CISCO_ONS_15454_OSC_REGENERATION_SITE"
- "CISCO_ONS_15454_L_LINE_SITE"
- "CISCO_ONS_15454_L_ROADM"
- "CISCO_ONS_15454_L_TERMINAL"
- "CISCO_ONS_15454_L_HUB"
- "CISCO_ONS_15454_Other"
- "CISCO_ONS_15454_Unknown"
- "CISCO_ONS_15454_SDH_MSSPRING"
- "CISCO_ONS_15454_SDH_SNC_ADM"

- "CISCO_ONS_15454_SDH_TSA_TERMINAL"
- "CISCO_ONS_15454_SDH_TSA_TERMINAL_EAST"
- "CISCO_ONS_15454_SDH_TSA_TERMINAL_WEST"
- "CISCO_ONS_15454_SDH_OPTICAL_ADM"
- "CISCO_ONS_15454_SDH_HUB"
- "CISCO_ONS_15454_SDH_HYBRID"
- "CISCO_ONS_15454_SDH_DWDM_HUB_NODE"
- "CISCO_ONS_15454_SDH_PASSIVE_LINE_SITE"
- "CISCO_ONS_15454_SDH_ACTIVE_LINE_SITE"
- "CISCO_ONS_15454_SDH_LINE_AMPLIFIER"
- "CISCO_ONS_15454_SDH_DWDM_TERMINAL"
- "CISCO_ONS_15454_SDH_ROADM"
- "CISCO_ONS_15454_SDH_L_LINE_SITE"
- "CISCO_ONS_15454_SDH_L_ROADM"
- "CISCO_ONS_15454_SDH_L_TERMINAL"
- "CISCO_ONS_15454_SDH_L_HUB"
- "CISCO_ONS_15454_SDH_Other"
- "CISCO_ONS_15454_SDH_Unknown"
- "CISCO_ONS_15600_UPSR_ADM"
- "CISCO_ONS_15600_TSA_TERMINAL"
- "CISCO_ONS_15600_BLSR_ADM"
- "CISCO_ONS_15600_LINEAR_ADM"
- "CISCO_ONS_15600_DCS"
- "CISCO_ONS_15600_HYBRID"
- "CISCO_ONS_15600_Other"
- "CISCO_ONS_15600_Unknown"
- "CISCO_ONS_15600_SDH_MSSPRING"
- "CISCO_ONS_15600_SDH_TSA_TERMINAL"
- "CISCO_ONS_15600_SDH_LINEAR_ADM"
- "CISCO_ONS_15600_SDH_DCS"
- "CISCO_ONS_15600_SDH_HYBRID"
- "CISCO_ONS_15600_SDH_Other"
- "CISCO_ONS_15600_SDH_Unknown"
- "CISCO_ONS_15305"
- "CISCO_ONS_15305_CTC_SNCP"
- "CISCO_ONS_15305_CTC_Unknown"
- "CISCO_ONS_15305_CTC_Other"
- "Not_Managed_Other_Vendor"
- "CISCO_ONS_15216_EDFA1"
- "CISCO_ONS_15216_EDFA2"
- "CISCO_ONS_15216_EDFA3_POST_AMPLIFIER"
- "CISCO_ONS_15216_EDFA3_LINE_AMPLIFIER"
- "CISCO_ONS_15216_EDFA3_PRE_AMPLIFIER"
- "CISCO_ONS_15216_EDFA3_Other"
- "CISCO_ONS_15216_EDFA3_Unknown"
- "CISCO_ONS_15216_OSC_1510"

- "CISCO_ONS_15216_DCU"
- "CISCO_ONS_15216_FlexLayer"
- "CISCO_ONS_15216_200_GHZ_OADM1"
- "CISCO_ONS_15216_200_GHZ_OADM2"
- "CISCO_ONS_15216_100_GHZ_OADM4"
- "CISCO_ONS_15216_100_GHZ_OADM2"
- "CISCO_ONS_15216_100_GHZ_OADM1"
- "CISCO_ONS_15216_DWDM_Filter_RED_200_GHZ_DWDM_FILTER"
- "CISCO_ONS_15216_DWDM_Filter_BLUE_200_GHZ_DWDM_FILTER"
- "CISCO_ONS_15216_DWDM_Filter_RED_100_GHZ_DWDM_FILTER"
- "CISCO_ONS_15216_DWDM_Filter_BLUE_100_GHZ_DWDM_FILTER"
- "CISCO_ONS_15310_CL_UPSR_ADM"
- "CISCO_ONS_15310_CL_TSA_TERMINAL"
- "CISCO_ONS_15310_CL_BLSR_ADM"
- "CISCO_ONS_15310_CL_HYBRID"
- "CISCO_ONS_15310_CL_Other"
- "CISCO_ONS_15310_CL_Unknown"
- "CISCO_ONS_15310_MA_UPSR_ADM"
- "CISCO_ONS_15310_MA_TSA_TERMINAL"
- "CISCO_ONS_15310_MA_BLSR_ADM"
- "CISCO_ONS_15310_MA_HYBRID"
- "CISCO_ONS_15310_MA_Other"
- "CISCO_ONS_15310_MA_Unknown"
- "CISCO_ONS_15310_MA_SDH_UPSR_ADM"
- "CISCO_ONS_15310_MA_SDH_TSA_TERMINAL"
- "CISCO_ONS_15310_MA_SDH_BLSR_ADM"
- "CISCO_ONS_15310_MA_SDH_HYBRID"
- "CISCO_ONS_15310_MA_SDH_Other"
- "CISCO_ONS_15310_MA_SDH_Unknown"
- "CISCO_CARRIER_PACKET_TRANSPORT_200"
- "CISCO_CARRIER_PACKET_TRANSPORT_200_SDH"
- "CISCO_CARRIER_PACKET_TRANSPORT_600"
- "CISCO_CARRIER_PACKET_TRANSPORT_600_SDH"

Throws

globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised when containingSubnetworkName does not reference a valid subnetwork name.

EXCPT_OBJECT_IN_USE - Raised if the managed element already exists.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

Compliance

Prime Optical-defined.

3.9.45 deleteManagedElement

Synopsis

void deleteManagedElement (

```
in globaldefs::NamingAttributes_T managedElementName)
raises(globaldefs::ProcessingFailureException)
```

Description
This interface allows the NMS to delete a managed element. Prime Optical generates an NT_OBJECT_DELETION event upon successful deletion of the managed element.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME to delete.

Throws

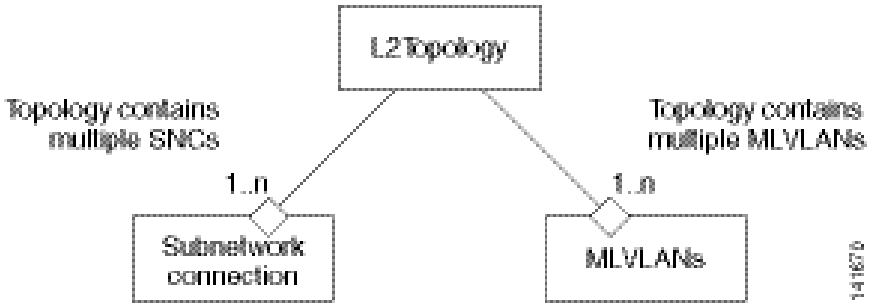
```
globaldefs::ProcessingFailureException
EXCPT_ENTITY_NOT_FOUND - Raised if the managed element is not found.
EXCPT_UNABLE_TO_COMPLY - Raised if the EMS is unable to comply.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
```

Compliance
Prime Optical-defined.

3.10 L2 Topology and ML VLAN for ML-Series Ethernet Cards

Prime Optical supports L2 topology and ML VLAN operations for the ML-series Ethernet cards in ONS 15454 SONET and ONS 15454 SDH NEs. You can create an ML VLAN in an L2 topology. An L2 topology has a collection of ML VLANs and underlying subnetwork connections to support the traffic at the optical layer.
The following figure shows the relationship diagram for various L2 service objects.

Figure 3-4: L2 Service Object Relationship Diagram



Prime Optical supports L2 service interfaces for the ONS 15454 SONET and ONS 15454 SDH R4.1 and later.

3.10.1 createMLVLAN

Synopsis

```
void createMLVLAN(in String protocol,in globaldefs::NamingAttributes_T
l2TopologyName,
in multiLayerSubnetwork::MLVLANCreateData_T mlVlanCreateData,
out multiLayerSubnetwork::MLVLAN_T mlVlan)
raises(globaldefs::ProcessingFailureException);
```

Description
This interface creates a new ML VLAN for a given L2 topology. An object creation event is generated when an ML VLAN is created successfully. Prime Optical supports a maximum of 4095 ML VLANs for an L2 topology.

Parameters

Name	Type	Input/ Output	Description
protocol	string	In	This VLAN runs the set protocol. If you want to run the default protocol, pass an empty string. Otherwise, you can set different protocols; for example, RSTP. You must set the RSTP protocol on the VLAN to enable RSTP on the port that belongs to the VLAN.
l2TopologyName	globaldefs::NamingAttributes_T	In	The L2 topology name in which this ML VLAN is created.
mlVlanCreateData	multiLayerSubnetwork::MLVLANCreateData_T	In	The createData for the ML VLAN to create.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	The ML VLAN object created by Prime Optical.

Prime Optical support for MLVLANCreateData_T structure is as follows:

- int mlVlanId
  - Supported. The ML VLAN ID used to create the ML VLAN. If it is –1, Prime Optical assigns the ID for the ML VLAN created. Prime Optical supports ML VLAN IDs from 1 to 4095.
- string userLabel
  - Not supported.
- boolean forceUniqueness
  - Supported. Prime Optical always enforces the uniqueness of the ML VLAN ID.
- string owner
  - Not supported.
- multilayersubnetwork::MLVLANTPDataList_T tpList
  - Supported. A list of front Ethernet ports associated with the ML VLAN.
  - Each MLVLANTPData_T structure has the following members:
  - globaldefs::NamingAttributes_T tpName
  - Specifies the front Ethernet port name.
  - multiLayerSubnetwork::L2PortType_T portType
  - Specifies the port type to configure. Prime Optical supports the following port types:
  - L2PortType_T.L2PT_UNI
  - L2PortType_T.L2PT_NNI
- multiLayerSubnetwork::L2ConnectionType_T connectionType
  - Specifies the connection type to configure for the specified port type. If the port type is L2PortType_T.UNI, the following connection types are supported:
  - L2ConnectionType_T.L2CT_QINQ
  - L2ConnectionType_T.L2CT_DOT1Q
  - L2ConnectionType_T.L2CT_UNTAGGED

  - If the port type is L2PortType_T.L2PT_NNI, the following connection type is supported:
    - L2ConnectionType_T.L2CT_DOT1Q
  - If the connection type is not L2ConnectionType_T.L2CT_QINQ, you can specify the following RSTP values.
- multiLayerSubnetwork::L2RSTPState_T rstpState
  - The NMS can enable or disable the RSTP state for a non-QinQ drop. Prime Optical supports two port states:
    - L2RSTPState_T.L2RSTP_ENABLE
    - L2RSTPState_T.L2RSTP_DISABLE
  - For a QinQ drop, this value (if specified) is ignored.
- multiLayerSubnetwork::L2PortState_T portState
  - NMS can enable a service or disable a port. Prime Optical support two port states:
    - L2PortState_T.L2PS_ENABLE
    - L2PortState_T.L2PS_DISABLE
- multiLayerSubnetwork::L2InterfaceType_T interfaceType
  - Specifies the interface type to be configured. Prime Optical supports the following interface types:
    - L2InterfaceType_T. L2IT_FASTETHER
    - L2InterfaceType _T. L2IT_GIGAETHER
    - L2InterfaceType _T. L2IT_PORTCHANNEL
- multiLayerSubnetwork::QOSPolicyData_T qosPolicyData
  - Prime Optical support for QOSPolicyData_T structure is described in QOSPolicyData_T.
- int portVlan
  - The user VLAN ID to associate with this ML VLAN. If the connection type is L2ConnectionType_T.L2CT_DOT1Q, the valid port VLAN range is from 2 to 4096. If the connection type is L2ConnectionType_T.L2CT_UNTAGGED, the port VLAN is ignored.
- globaldefs::NVSList_T additionalInfo
  - Supported. You can specify the customerID and service ID.

Prime Optical support for MLVLAN_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported. This field contains three tuples. The first tuple has the EMS value. The second tuple has the L2 topology name. The third tuple has the ML VLAN ID that is the same as nativeEMSName.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. The ML VLAN ID is reported.
- string owner
  - Not supported.
- multilayersubnetwork::MLVLANTPDataList_T tpList
  - Supported. Specifies the list of all front Ethernet ports and associated service type.
  - If the port type on a port cannot be determined, Prime Optical represents the port type as L2PortType_T.L2PT_UNKNOWN.
  - If the connection type for an ML VLAN on a port cannot be determined, Prime Optical represents the connection type as L2ConnectionType_T.L2CT_UNKNOWN.
- multiLayerSubnetwork::MLVLANState_T vlanState

  - Supported. Valid values are MLVLANState_T.COMPLETE and MLVLANState_T.INCOMPLETE. A state change event is generated if the VLAN state changes.
- globaldefs::NVSList_T additionalInfo
  - Supported. The customer ID and service ID are reported.

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when one of the following conditions occurs:

- l2TopologyName does not reference an L2 topology object.

- MLVLAN name does not follow the naming convention.

- QOSPolicy is specified incorrectly.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology
object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be created due to the
restriction that an L2 topology can have a maximum of 4095 ML VLAN IDs.
```

Compliance

Prime Optical-defined.

3.10.2 modifyMLVLANAddFrontPorts

Synopsis

```
void modifyMLVLANAddFrontPorts(in globaldefs::NamingAttributes_T mlVlanName,

in multiLayerSubnetwork::MLVLANTPDataList_T tpList,

out multiLayerSubnetwork::MLVLAN_T mlVlan, String protocol)

raises(globaldefs::ProcessingFailureException);
```

Description

This interface adds additional front ports to the existing MVLAN. An attribute change event is reported when the front ports are successfully added. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain updated MLVLAN_T structure information. If the addition of any of the front ports to the ML VLAN fails, Prime Optical generates a processing failure exception. The NMS user must requery the ML VLAN information to obtain the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/ Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which to modify the ML VLAN.
tpList	multiLayerSubnetwork::MLVLANTPDataList_T	In	List of front ports to include for the specified ML VLAN.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN object returned after modification.
protocol	String	In	Bridge group protocol of the ML VLAN.

For details about the MLVLAN_T, MLVLANTPData_T structure, see 3.10.1 createMLVLAN.

Throws

```
globaldefs::ProcessingFailureException
```

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object or when one or more PTPs in the PTP list does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist or when one or more PTPs in the PTP list does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be modified.

Compliance
Prime Optical-defined.

3.10.3 modifyMLVLANDropFrontPorts

Synopsis

```
void modifyMLVLANDropFrontPorts(
        in globaldefs::NamingAttributes_T mlVlanName,
        in globaldefs::NamingAttributesList_T tpList,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface drops front ports associated with an existing ML VLAN. An attribute change event is reported when the front ports are dropped from the ML VLAN. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain updated MLVLAN_T structure information. If removal of any front ports associated with the ML VLAN fails, Prime Optical generates a processing failure exception. The NMS user must requery the ML VLAN information to obtain the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which this ML VLAN is created.
tpList	globaldefs::NamingAttributesList_T	In	List of front PTP names to drop from the specified ML VLAN.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN object returned after modification.

For details about the MLVLAN_T structure, see 3.10.1 createMLVLAN.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object or when one or more PTPs in the PTP list does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist or when one or more PTPs in the PTP list does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be modified.

Compliance

Prime Optical-defined.

3.10.4 modifyMLVLANPortState

Synopsis

```
void modifyMLVLANPortState(
        in globaldefs::NamingAttributes_T mlVlanName,
        in multiLayerSubnetwork::MLVLANPortStateList_T tpPortStateList,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables or disables specified ML VLAN traffic on the specified list of front ports. An attribute change event is reported when one of the ML VLAN parameters is modified. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain updated MLVLAN_T structure information. If modification of any of the ML VLAN port states fails, Prime Optical generates a processing failure exception. The NMS user must requery the ML VLAN information to obtain the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/ Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which to create the ML VLAN.
tpPortStateList	multiLayerSubnetwork::MLVLANTPPortStateList_T	In	List of front ports on which to modify the port state for the specified ML VLAN.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN object returned after modification.

For details about the MLVLAN_T structure, see 3.10.1 createMLVLAN.

Prime Optical support for MLVLANTPPortState_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - The front Ethernet port name used by this ML VLAN.
- L2PortState_T portState
  - The portState is applicable only to the specified ML VLAN. The supported states are:
  - L2PortState_T.L2PS_ENABLE
  - L2PortState_T.L2PS_DISABLE

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object or when one or more PTPs in the PTP list does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist or when one or more PTPs in the PTP list does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be modified.

Compliance
Prime Optical-defined.

3.10.5 modifyMLVLANRSTPState

Synopsis

```
void modifyMLVLANRSTPState(
        in globaldefs::NamingAttributes_T mlVlanName,
        in multiLayerSubnetwork::MLVLANTPrstpStateList_T tpRstpStateList,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables or disables specific ML VLAN traffic on the specified list of drop ports. An attribute change event is reported when an ML VLAN parameter is modified. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain updated MLVLAN_T structure information. If the modification of an ML VLAN RSTP state fails, Prime Optical generates a processing failure exception. You must issue another query for the ML VLAN information to obtain the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which to create the ML VLAN.
tpRstpStateList	multiLayerSubnetwork:: MLVLANTPrstpStateList_T	In	List of front ports on which the RSTP state must be modified for the specified ML VLAN.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN object returned after modification.

For details about the MLVLAN_T structure, see 3.10.1 createMLVLAN.
Prime Optical support for MLVLANRSTPState_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - The front Ethernet port name used by this ML VLAN.
- L2RSTPState_T rstpState
  - The rstpState is applicable only to the specified non-QinQ drops of the ML VLAN. The supported states are:
  - L2RSTPState_T.L2RSTP_ENABLE
  - L2RSTPState_T.L2RSTP_DISABLE

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object or when one or more PTPs in the PTP list does not reference a PTP object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist or when one or more PTPs in the PTP list does not exist.

EXCPT_QINQ - Raised when the drop port for which the RSTP state must be enabled or disabled uses the QinQ connection type.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be modified.

Compliance
Prime Optical-defined.

3.10.6 modifyMLVLANTPQOS

Synopsis

```
void modifyMLVLANTPQOS(
        in globaldefs::NamingAttributes_T mlVlanName,
        in multiLayerSubnetwork::MLVLANTPQOSPolicyList_T tpQosPolicyList,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface modifies the QoS policies of a list of termination points in the specified ML VLAN. An attribute change event is reported for each call to this API, even if more than one TP QoS policy has been modified. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain updated MLVLAN_T structure. Modification of any QoS policy is done on a best effort basis. If modification of any part of the QOSPolicy fails, Prime Optical generates a processing failure exception. The NMS user must requery the ML VLAN information to get the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which this ML VLAN is created.
tpqosPolicyList	multiLayerSubnetwork::MVLANTPQOSPolicyList_T	In	New set of QOSPolicy_T to set on the specified ML VLAN TPs.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN TP object returned after modification.

For details about the MLVLAN_T and QOSPolicy_T structures, see 3.10.1 createMLVLAN.
Prime Optical support for MLVLANTPQOSPolicy_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - The front Ethernet port name used by this ML VLAN to modify the QoS policy.
- QOSPolicyData_T qosPolicyData
  - Refer to QOSPolicyData_T.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when one of the following conditions occurs:
- The mlVlanName passed does not have the correct format for ML VLAN name.
- The tpQosPolicyList contains the incorrect format for TP or QOS policy.
- The tpQosPolicyList contains the same TP more than once.
EXCPT_ENTITY_NOT_FOUND - Raised when one of the following conditions occurs:
- The mlVlanName passed does not reference an ML VLAN object.
- The tpQosPolicyList contains TPs that do not reference valid objects.
EXCPT_UNABLE_TO_COMPLY - Raised when the QOS policy cannot be modified.
```

Compliance
Prime Optical-defined.

3.10.7 modifyMLVLANCustomerIDServiceID

Synopsis

```
void modifyMLVLANCustomerIDServiceID(
        in globaldefs::NamingAttributes_T mlVlanName,
        in multiLayerSubnetwork::NVSList_T ids,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface modifies the customer ID and/or service ID associated with an existing ML VLAN. An attribute change event is reported when the customer ID or service ID is modified. Prime Optical does not specify which attribute of the ML VLAN has changed. The NMS must obtain an updated MLVLAN_T structure information. If modification of the customer ID or service ID fails, Prime Optical generates a processing failure exception. The NMS user must requery the ML VLAN information to get the new set of parameters associated with the ML VLAN.

Parameters

Name	Type	Input/ Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which this ML VLAN is created.
ids	globaldefs::NVSList_T	In	Name and value string. The name string is CustomerID and/or ServiceID, and the value string is the new customer ID or service ID.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	ML VLAN object returned after modification.

For details about the MLVLAN_T structure, see 3.10.1 createMLVLAN.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2
topology object.
```

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the ML VLAN cannot be modified.

Compliance
Prime Optical-defined.

3.10.8 getMLVLAN

Synopsis

```
void getMLVLAN(
        in globaldefs::NamingAttributes_T mlVlanName,
        out multiLayerSubnetwork::MLVLAN_T mlVlan)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface reports the complete ML VLAN information for a specified ML VLAN name.

Parameters

Name	Type	Input/Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name with which to obtain the ML VLAN.
mlVlan	MultiLayerSubnetwork::MLVLAN_T	Out	MLVLAN_T structure.

For details about the MLVLAN_T structure, see 3.10.1 createMLVLAN.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the ML VLAN name is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when the ML VLAN name does not exist on the L2 topology.

Compliance
Prime Optical-defined.

3.10.9 deleteMLVLAN

Synopsis

```
void deleteMLVLAN(
        in globaldefs::NamingAttributes_T mlVlanName)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface deletes an existing ML VLAN. Prime Optical generates a processing failure exception if the ML VLAN deletion fails. Prime Optical generates an object deletion event when the ML VLAN is successfully deleted.

Parameters

Name	Type	Input/Output	Description
mlVlanName	mlVlan:MLVLAN_T	In	ML VLAN name to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mlVlanName does not reference an ML VLAN object.

EXCPT_ENTITY_NOT_FOUND - Raised when mlVlanName references an ML VLAN object that does not exist.

EXCPT_UNABLE_TO_COMPLY – Raised when mlVlan deletion fails.

Compliance
Prime Optical-defined.

3.10.10 getAllMLVLANs

Synopsis

```
void getAllMLVLANs(
        in globaldefs::NamingAttributes_T l2TopologyName,
        in unsigned long how_many,
        out multiLayerSubnetwork::MLVLANList_T mlVlanList,
        out multiLayerSubnetwork::MLVLANIterator_I mlVlanIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface lists all ML VLANs for the L2 topology. The NMS user must specify the L2 topology name. Prime Optical returns a list of all ML VLANs associated with the L2 topology specified.

Parameters

Name	Type	Input/Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	L2 topology name for which to obtain all the ML VLANs.
how_many	unsigned long	In	Maximum number of ML VLANs to return in the first batch.
mlVlanList	MLVLANList_T	Out	List of available ML VLAN names in the L2 topology.
mlVlanIt	MultiLayerSubnetwork::MLVLANIterator_I	Out	Iterator used to retrieve the remaining ML VLANs.

For details about the MLVLAN_T structure, see 3.10 L2 Topology and ML VLAN for ML-Series Ethernet Cards.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance
Prime Optical-defined.

3.10.11 getAllMLVLANNames

Synopsis

```
void getAllMLVLANNames(
        in globaldefs::NamingAttributes_T l2TopologyName,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface lists all the ML VLAN names for the L2 topology.

Parameters

Name	Type	Input/ Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	L2 topology name for which to obtain all the ML VLANs.
how_many	unsigned long	In	Maximum number of ML VLANs to return in the first batch.
nameList	globaldefs::NamingAttributesList_T	Out	List of available ML VLAN names in this L2 topology.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator used to retrieve the remaining ML VLAN names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

Prime Optical-defined.

3.10.12 getAllAssociatedSNCs

Synopsis

```
void getAllAssociatedSNCs(
        in globaldefs::NamingAttributes_T l2TopologyName,
        in unsigned long how_many,
        out subnetworkConnection::SubnetworkConnectionList_T sncList,
        out subnetworkConnection::SNCIterator_I sncIt,
        out subnetworkConnection::VCATList_T vcatList,
        out subnetworkConnection::VCATIterator_I vcatIt)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface allows Prime Optical to report all the underlying Layer 1 SONET/SDH SNCs that are being used by the specified L2 topology.
Parameters

Name	Type	Input/ Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	L2 topology name for which to obtain all the SNCs.
how_many	unsigned long	In	Maximum number of SNCs to return in the first batch.
sncList	subnetworkConnection::SubnetworkConnectionList_T	Out	First batch of SNCs.
sncIt	subnetworkConnection::SNCIterator_I	Out	Iterator to retrieve the remaining SNCs.
vcatList	subnetworkConnection::VCATList_T	Out	First batch of VCAT SNCs.
vcatIt	subnetworkConnection::VCATIterator_I	Out	Iterator to retrieve the remaining VCAT SNCs.

For details about the SubnetworkConnection_T structure, see 2.2.7.7. SubnetworkConnection_T structure information.

For details about the VCAT_T structure, see 3.9.21 getVCATSNC.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
Prime Optical-defined.

3.10.13 getL2Topology

Synopsis
```
void getL2Topology(
        in globaldefs::NamingAttributes_T l2TopologyName,
        out multiLayerSubnetwork::L2Topology_T l2Topo)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns the L2 topology structure for a given L2 topology name.
Parameters

Name	Type	Input/ Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	Name of the L2 topology to retrieve.
l2Topo	multiLayerSubnetwork::L2Topology_T	Out	Retrieved L2 topology.

For details about the L2Topology_T structure, see 3.3.21 getEMS.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an L2 topology object that does not exist.

Compliance
Prime Optical-defined.

3.10.14 getAssociatedL2TopologyForSNC

Synopsis
```
void getAssociatedL2TopologyForSNC(
        in globaldefs::NamingAttributes_T sncName,
        out multiLayerSubnetwork::L2Topology_T l2Topo)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns the L2 topology structure associated with a given SNC name. Prime Optical reports the L2 topology using the specified Layer 1 SNC. Prime Optical returns an empty L2Topology_T structure if the SNC name specified is currently not associated with any L2 topology in the EMS.

Parameters

Name	Type	Input/ Output	Description
sncName	globaldefs::NamingAttributes_T	In	Name of the SNC to retrieve the associated L2 topology.
l2Topo	multiLayerSubnetwork::L2Topology_T	Out	Retrieved L2 topology.

For details about the L2Topology_T structure, see 3.3.22 getAllL2Topologies.
Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not reference a subnetwork connection object.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that does not exist.

Compliance
Prime Optical-defined.

3.10.15 getAssociatedL2TopologyForMLVLAN

Synopsis

```
void getAssociatedL2TopologyForMLVLAN(
        in globaldefs::NamingAttributes_T mlVlanName,
        out multiLayerSubnetwork::L2Topology_T l2Topo)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface returns the L2 topology structure associated with a given ML VLAN name. Prime Optical returns an empty L2Topology_T structure if the ML VLAN name specified is currently not associated with any L2 topology in the EMS.

Parameters

Name	Type	Input/Output	Description
mlVlanName	globaldefs::NamingAttributes_T	In	Name of the ML VLAN to retrieve containing the L2 topology.
l2Topo	multiLayerSubnetwork::L2Topology_T	Out	Retrieved L2 topology.

For details about the L2Topology_T structure, see 3.3.22 getAllL2Topologies.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mlVlanName does not reference an ML VLAN object.

EXCPT_ENTITY_NOT_FOUND - Raised when mlVlanName references an ML VLAN object that does not exist.

Compliance

Prime Optical-defined.

3.10.16 createL2Topology

Synopsis

```
void createL2Topology(
        in multiLayerSubnetwork::L2TopoCreateData_T createData,
        out multiLayerSubnetwork::L2Topology_T l2Topo)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface provides a way to create an L2 topology, including all the optical SNCs associated with it. The user must associate ML VLANs using the interface method defined in 3.10.1 createMLVLAN. Prime Optical creates an L2 topology within the same MultiLayerSubnetwork. If the NMS specifies the L2 topology createData with MEs belonging to a different MultiLayerSubnetwork, Prime Optical returns the ProcessingFailureException EXCPT_UNABLE_TO_COMPLY.

Prime Optical supports point-to-point, hub-and-spoke, Resilient Packet Ring (RPR), and 802.17 RPR topologies. However, Prime Optical supports the creation of only RPR, 802.17 RPR, and point-to-point topologies. Prime Optical treats a hub-and-spoke topology as multiple point-to-point topologies.

This is an asynchronous operation. Prime Optical returns this method after all requests for creating the underlying SNCs are complete. Prime Optical autodiscovers an L2 topology based on the L1 SNCs created. Prime Optical generates OCE for all the L1 SNCs created. Prime Optical also generates L2 topology OCE after the L2 Topology is autodiscovered.

To create an RPR or 802.17 RPR topology, you must specify the complete list of ML-series and/or ML2 (L2/L3) equipment names to involve in the topology. RPR consists of all ML-series equipment or all ML2 (L2/L3) equipment or a combination of ML and ML2 (L2/L3) equipment. To create a point-to-point topology, you must specify the end TPs of the underlying L1 SNC.

During the L2 topology creation, if any part of the underlying L1 SNC creation fails, Prime Optical generates a processing failure exception. However, Prime Optical autodiscovers the L2 topology in INCOMPLETE state if any SNC was created before the failure occurred. Either the NMS creates the remaining L1 SNCs that failed, using the interface described in 3.9.16 createAndActivateSNCFromUserLabel to complete the L2 topology, or the NMS deletes the L2 topology and recreates it using this operation.

For RPR topologies, the NMS must specify all the equipment to involve, in cyclical order and in pairs. There are three different scenarios:

- All SNCs in the RPR are CCAT. In this case, only SNCCreateDataList_T is populated and VCATSNCCreateDataList_T is empty. For example, if the NMS is trying to create an RPR topology involving three ML-series cards (M1, M2, and M3), the NMS must specify a three-element array of SNCCreateData_T. The first element's aEnd will specify the M1 equipment name and the zEnd will specify the M2 equipment name. The second element's aEnd will specify the M2 equipment name and zEnd will specify the M3 equipment name. The third element's aEnd will specify the M3 equipment name and zEnd will specify the M1 equipment name.
- All SNCs in the RPR are VCAT. In this case, only VCATSNCCreateDataList_T is populated and SNCCreateDataList_T is empty. For example, if the NMS is trying to create an RPR topology involving three ML-series cards (M1, M2, and M3), the NMS must specify a three-element array of VCATSNCCreateData_T. The first element's aEnd will specify the M1 equipment name and zEnd will specify the M2 equipment name. The second element's aEnd will specify the M2 equipment name and zEnd will specify the M3 equipment name. The third element's aEnd will specify the M3 equipment name and zEnd will specify the M1 equipment name.
- Mix of CCAT and VCAT SNCs in the RPR. In this case, both VCATSNCCreateDataList_T and SNCCreateDataList_T are populated and all the equipment in the two arrays is in cyclical order.

  For example, if the NMS is trying to create an RPR topology involving three ML-series cards (M1, M2, and M3) and the SNC between M1-M2 and M3-M1 is of type CCAT and M2-M3 is of type VCAT, the NMS must specify:
  - A two-element array of SNCCreateData_T with the first element's aEnd specifying the M1 equipment name and the zEnd specifying the M2 equipment name, and the second element's aEnd specifying the M3 equipment name and zEnd specifying the M1 equipment name.
  - A one-element array of VCATSNCCreateData_T with the first element's aEnd specifying the M2 equipment name and zEnd specifying the M3 equipment name.

For point-to-point topologies, the NMS specifies either CCATSNCCreateData_T or VCATSNCCreatedata_T. For creating CCAT point-to-point topologies, CCATSNCCreateData_T is populated and VCATSNCCreateData_T is empty. For creating VCAT point-to-point topologies, VCATSNCCreateData_T is populated and CCATSNCCreateData_T is empty.

In addition to specifying SNC creation parameters using the SNCCreateData_T structure for each SNC in the L2 topology, the NMS can provide a partial or complete list of managed elements, or of both end PTPs of a topological link that is included or excluded in the SNC route during SNC creation. The NMS can specify only one type of object name list during the SNC creation. Prime Optical returns a processing failure exception if the NMS-specified list is not homogeneous. Prime Optical returns a processing failure exception with EXCPT_ENTITY_NOT_FOUND as the error reason if any one of the specified managed

elements or PTPs does not exist. If the NMS specifies a PTP list, Prime Optical verifies whether the PTPs are specified in pairs and each PTP in a pair is associated with each end PTP of the same topological link. Prime Optical uses the first PTP of a pair as the aEnd of the topological link and uses the second PTP of the same pair as the zEnd of the topological link. Prime Optical attempts to route the SNC from aEnd PTP of a PTP pair to zEnd PTP of the same pair on a topological link. Prime Optical returns a processing failure exception if the SNC cannot be routed from the specified aEnd PTP to the zEnd PTP of a topological link. The NMS then must interchange the aEnd and zEnd PTPs specified, provide a different PTP pair, or delete the PTP pair from the include or exclude list. Prime Optical returns a processing failure exception if at least one PTP is not associated with a topological link. The NMS must ensure that the sequence of the object name list conforms to the sequence of the objects existing in the network. Prime Optical returns a processing failure exception if the sequence is nonconforming.
Prime Optical does not support the creation of L2 topologies in which SNCs have more than one source point and destination endpoint in SNCCreateData_T.aEnd or SNCCreateData_T.zEnd.
Prime Optical does not support 802.17 RPR topology creation if the circuit size entered is not compatible with the bandwidth values entered. Prime Optical returns a processing failure exception if the circuit size and bandwidth values are incompatible. See the Cisco Prime Optical 10.7 User Guide for compatibility between circuit size and bandwidth.
Prime Optical supports the creation of L2 topologies containing VCAT SNCs. For point-to-point topologies, either the CCAT SNC or VCAT SNC is specified. RPR topologies can consist of both CCAT and VCAT SNCs.

Parameters

Name	Type	Input/Output	Description
L2TopoCreateData_T	multiLayerSubnetwork::L2TopoCreateData_T	In	Data structure defining the L2 topology to create.
L2Topo	multiLayerSubnetwork::L2Topology_T	Out	Created L2 topology.

Prime Optical support for L2TopologyCreateData_T structure is as follows:

- string nativeEMSName
  - Supported. Used as the L2 topology name.
- string userLabel
  - Not supported
- boolean forceUniqueness
  - Guarantees the uniqueness if all L2 topologies are created by Prime Optical only. If any other tool is used for L2 topology creation, Prime Optical does not guarantee uniqueness.
- string owner
  - Not supported.

- transmissionParameters::LayerRate_T
  - Not supported. LayerRate is specified as part of SNCCreateData_T an and VCAT SNCCreateData_T structure
- multiLayerSubnetwork::L2TopoType l2TopoType
  - The type of the L2 topology. Valid values are L2TopoType.PT2PT, L2TopoType.RPR, and L2TopoType.DOT17_RPR.
- subnetworkConnection::SNCCreateDataList_T sncInfoList
  - Supported.

  - Prime Optical support for SNCCreateData_T structure for the L2 topology creation is as follows:
    - string userLabel
    - Not supported.
    - boolean forceUniqueness
    - Not supported.
    - string owner
    - Not supported.
    - globaldefs::ConnectionDirection_T direction
    - Not supported. Prime Optical always creates CD_BI (bidirectional) circuit.
    - StaticProtectionLevel_T staticProtectionLevel
    - Supported. Prime Optical supports UNPROTECTED and FULLY_PROTECTED.
    - ProtectionEffort_T protectionEffort
    - Prime Optical supports all values defined by TMF.
    - If the NMS requests unprotected SNC on a BLSR ring, Prime Optical creates the SNC. The same is true for unprotected SNC on 1+1 protection.
    - If the NMS specifies EFFORT_SAME_OR_WORSE or EFFORT_WHATEVER, Prime Optical expects the value of staticProtectionLevel to be UNPROTECTED. These values indicate that Prime Optical must create the PCA.
    - Reroute_T rerouteAllowed
    - Not supported.
    - NetworkRouted_T networkRouted
    - Not supported.
    - SNCType_T sncType
    - Not supported.
    - transmissionParameters::LayerRate_T layerRate
    - Supported. For a list of supported layer rates, see 2.2.1 Layer Rate.
- CrossConnectList_T ccInclusions
  - Not supported.
- ResourceList_T neTpInclusions
  - Supported. You must provide a list of managed element names or the PTP names at the endpoints of the topological link. For complete manual routing, you must provide the list of all CTPs of the links (OCH trail links, if any) to which the circuit must be routed.
- boolean fullRoute
  - Not supported.
- ResourceList_T neTpSncExclusions
  - Supported. Prime Optical expects a list of ME or PTP names at the endpoints of the topological link to exclude from the SNC route.
- globaldefs::NamingAttributesList_T aEnd
  - Supported. If the topology is RPR, you specify an ML-series equipment name to use to create the L2 topology. If the topology is point-to-point, you specify the aEnd TP name. Point-to-point topologies can be created with one endpoint on the ML-series equipment and the other endpoint on ML-series, G-series, or OC-n equipment.
- globaldefs::NamingAttributesList_T zEnd
  - Supported. If the topology is RPR, you specify an ML-series equipment name to use to create the L2 topology. If the topology is point-to-point, you specify the zEnd TP name. Point-to-

point topologies can be created with one endpoint on the ML-series equipment and the other endpoint on ML-series, G-series, or OC-n equipment.

- globaldefs::NVSList_T additionalCreationInfo
  - Supported. This list can be used to provide additional parameters for SNC creation. Prime Optical supports the following values.

Name	Value	Description
upsrRevertiveMode	Revertive or nonrevertive	Supported only for point-to-point L2 topology. If value is set to revertive, traffic reverts back to the working path when the conditions that diverted it to the protect path are repaired. Otherwise it remains on the protect path after the switch.
upsrRevertiveTime	Valid float value from 0.5 to 12, in increments of 12	Supported only for point-to-point L2 topology. If SNC is revertive, set this parameter. This is the amount of time that will elapse before the traffic reverts to the working path. The default value is 5 minutes. Prime Optical does not validate any of these values.
upsrSfBerLevel	3, 4, or 5	Supported only for point-to-point L2 topology. Set the thresholds for path protection, path-level signal failure, and bit error rate (only for STS SNCs). Prime Optical does not validate any of these values.
upsrSdBerLevel	5, 6, 7, 8, and 9	Supported only for point-to-point L2 topology. Set the thresholds for path protection, path-level signal degrade, and bit error rate (only for STS SNCs). Prime Optical does not validate any of these values.
upserSwitchOnPdiP	True or false	Supported only for point-to-point L2 topology. Set this to “true” if you want traffic to switch when an STS payload defect indicator is received (only for STS SNCs).
goAndReturnRouting	True or false	Not supported.
sncDescription	String	Not supported.
VLAN	VLAN ID	Not supported.
CustomerID	String	Not supported.
ServiceID	String	Not supported.

Name	Value	Description
createTunnelOnTransitNodes	True or false	Applicable for VC SNCs with one or more transit nodes that do not have a tunnel or the existing tunnel is exhausted. This parameter specifies whether to create a tunnel on the transit nodes. For SNCs without transit nodes or with an existing tunnel on a transit node that has available bandwidth, this value is ignored. If the value specified is true, a tunnel is created and the SNC is routed using this tunnel. If the value specified is false, a new tunnel is not created. In this case, if a tunnel that is not full is available on the transit node, the circuit is routed through the tunnel; otherwise, the circuit is not routed through the tunnel.
createVTAggregation	True or false	Applicable for VC SNCs with one or more transit nodes that do not have a VAP circuit or the existing VAP circuit is exhausted. This parameter specifies whether to create a new VAP circuit on the transit node(s). If the value specified is true, a VAP circuit is created and SNC is routed using this new VAP circuit. If the value specified is false, a new VAP circuit is not created.
sourceIsSTSGrroming	True or false	When set to true, the source drop point of the VC circuit is STS groomed for the VAP circuit. This attribute is valid only when createVTAggregation is set to true. sourceIsSTSGrooming and destinationIsSTSGrooming cannot both be set to true.
destinationIsSTSGrooming	True or false	When set to true, the destination drop point of the VT or VC circuit is STS groomed for the VAP circuit. This attribute is valid only when createVTAggregation is set to true. sourceIsSTSGrooming and destinationIsSTSGrooming cannot both be set to true.
vapDestination	String	User-specified string (ME name) representing the VAP destination drop point.

Name	Value	Description
usePortlessTransmux	True or false	When set to true, the portless transmux card is used for the circuit. The NMS provides information for portlessTransmuxMeName, portlessTransmuxSlot, and isSourceDS3MappedSTS; otherwise, an INVALID INPUT exception is thrown. When set to false, the portless transmux card is not used as the intermediate point for the end-to-end STS-1 circuit.
portlessTransmuxMeName	String	Not supported.
portlessTransmuxSlot	/rack=1/shelf=1/slot=[1..6][12..17]	Not supported.
isSourceDS3MappedSTS	True or false	Not supported.
vcatMemberId	Integer from 1 to 256	Used for VCAT SNC creation, to identify every VCAT member L1 SNC uniquely. For normal CCAT circuits, this value is ignored if specified.

- subnetworkConnection::VCATSNCCreateDataList_T vcatSncInfoList
  - Supported.

Prime Optical support for VCATSNCCreateData_T structure for the L2 topology creation is as follows:

- string userLabel
  - Not supported.
- boolean forceUniqueness
  - Prime Optical guarantees the uniqueness of all VCATs created in Prime Optical. It is not guaranteed if other tools were used for VCAT SNC creation.
- string owner
  - Not supported.
- globaldefs::ConnectionDirection_T direction
  - Not supported. Prime Optical supports CD_BI (bidirectional).
- boolean isSymmetrical
  - Supported. Only symmetric VCAT SNCs are supported. If set to true, all members are bidirectional L1 SNCs. The VCAT layer rate is the same in both directions. If set to false, all members are unidirectional L1 SNCs.
- subnetworkConnection::LCASMode_T mode
  - Supported. Prime Optical supports the following values:
  - LCASMode_T.MODE_NONE
  - LCASMode_T.MODE_SWLCAS
  - LCASMode_T.MODE_LCAS
- subnetworkConnection::RoutingMode_T routingMode
  - Supported. Prime Optical supports the following values:
  - RoutingMode_T.COMMONFIBER_ROUTED
  - RoutingMode_T.INDEPENDENTFIBER_ROUTED
- globaldefs::NamingAttributes_T aEnd;

  - Supported. If the topology is an RPR topology, you specify an ML-series equipment name to use to create the L2 topology. If the topology is a point-to-point topology, Prime Optical expects this field to be an FTP representing the information up to the slot level of the card involved in the VCAT SNC.
- globaldefs::NamingAttributes_T zEnd;
  - Supported. If the topology is an RPR topology, you specify an ML-series equipment name to use to create the L2 topology. If the topology is a point-to-point topology, Prime Optical expects this field to be an FTP representing the information up to the slot level of the card involved in the VCAT SNC.
- subnetworkConnection::SNCCreateDataList_T sncCreateDataList
  - Supported. At least one L1 SNC is created at the time of VCAT SNC creation. Prime Optical expects you to specify the create data information required to create the L1 SNCs in the data structure.
- globaldefs::NVSList_T additionalCreationInfo
  - Not supported.
- subnetworkConnection::StaticProtectionLevel_T staticProtectionLevel
  - Not supported. StaticProtectionLevel is specified as part of SNCCreateData_T and VCATSNCCreateData_T structure.
- globaldefs::NVSList_T additionalInfo
  - Supported. You can choose to specify bandwidth parameters. By default, bandwidth parameters have a zero value. If you choose to enter a specific bandwidth value, you must also specify its corresponding CoS value. The default value for all CoS values is N/A. The sum of all the bandwidth parameters must be from 1 to 99. The CoS commit value must be from 0 to 7.
  - The following are the supported bandwidth parameters (Group1Bandwidth and Group2Bandwidth are multicast bandwidth parameters, and their corresponding CoS values are Group1CoS and Group2CoS):
    - Group1Bandwidth
    - Group2Bandwidth
    - SpMgmtBandwidth
    - CommittedBandwidth
    - BestEffortBandwidth
    - AvvidControlBandwidth
    - ClassSpMgmt
    - ClassCommitted
    - ClassBestEffort
    - ClassAvvidControl
    - ClassAvvidVoiceVideo
    - ClassAReservedBandwidth
    - ClassA1Bandwidth
    - ClassBCommittedBandwidth
    - SpMgmtCoS
    - CommittedCoS
    - AvvidVoiceVideoCoS
    - AvvidControlCoS
    - CosCommit
    - Group1CoS
    - Group2CoS

> Group2Bandwidth parameters must be entered along with Group1Bandwidth parameters. Entering Group2Bandwidth alone creates inconsistency in the L2 topology table.

If the topology is DOT17_RPR, multicast bandwidth parameters are not supported; Group1Bandwidth and Group2Bandwidth values are not needed. Group1CoS and Group2CoS do not apply to DOT17_RPR. For DOT17_RPR topologies, you can specify the absolute bandwidth values for the three possible classes of traffic (A Reserved, A1, and B Committed). For the remaining bandwidth parameters, you specify to which class of traffic (Class A, B, or C) the parameter belongs. For example, ClassSpMgmt belongs to Class C-type traffic.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the input tpList is invalid or when the specified layer rate is an invalid layer rate. This exception is also raised when the user does not specify the corresponding CoS values for the bandwidth parameters specified.

EXCPT_ENTITY_NOT_FOUND - Raised when the input tpList does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when a layer rate is not supported for L2 topology creation or when Prime Optical cannot create the L2 topology for the specified layer rate. Raised if the VCAT SNC cannot be created and activated because the CTP conflicts with another VCAT SNC or the NE does not support the VCAT SNC creation.

EXCPT_OBJECT_IN_USE - Raised when the specified L2 topology already exists.

EXCPT_NE_COMM_LOSS - Raised when one of the managed elements specified as part of the L2 topology creation is not reachable.

Compliance

Prime Optical-defined.

3.10.17 deleteL2Topology

Synopsis

```
void deleteL2Topology(
        in globaldefs::NamingAttributes_T l2TopologyName)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface provides a method to delete an L2 topology and all the SNCs associated with it. If there are any ML VLANs associated with the L2 topology when this interface method is invoked, Prime Optical generates a processing failure exception. The NMS must ensure that there are no ML VLANs associated with the L2 topology, to ensure that the L2 topology is successfully deleted.

Parameters

Name	Type	Input/ Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	Name of the L2 topology to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the l2TopologyName specified has an ML VLAN associated with it.

EXCPT_UNABLE_TO_COMPLY – Raised when one or more of the managed elements associated with the specified L2 topology are not accessible.

Compliance
Prime Optical-defined.

3.10.18 modifyL2Topology

Synopsis

```
void modifyL2Topology(
        in globaldefs::NamingAttributes_T l2TopologyName,
        in globaldefs::NVList_T l2ModifyData,
        out multiLayerSubnetwork::L2Topology_T l2Topo)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface provides a way to modify an L2 topology. It enables the OSS to modify the L2 topology for the bandwidth allocation parameters. For more information on the bandwidth allocation and CoS commit parameters, see 3.10.16 createL2Topology. This interface is used when the L2 topology is discovered, but it cannot be used when the L2 topology synchronization state is L2TopoSyncState.L2NOTREADY.

Parameters

Name	Type	Input/Output	Description
l2TopologyName	globaldefs::NamingAttributes_T	In	Name of the L2 topology to modify.
l2Modifydata	globaldefs::NVList_T	In	Parameters for modification of the L2 topology; the bandwidth allocation parameters.
l2Topo	multiLayerSubnetwork::L2Topology_T	Out	Modified L2 topology.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when l2TopologyName does not reference an L2 topology object.

EXCPT_ENTITY_NOT_FOUND - Raised when l2TopologyName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY – Raised when one or more of the managed elements associated with the specified L2 topology are not accessible.

Compliance
Prime Optical-defined.

3.10.19 getAllAvailableMLEquipmentOrTPNames

Synopsis

```
void getAllAvailableMLEquipmentOrTPNames(
```

```
        in globaldefs::NamingAttributesList_T managedElementList,
        in multiLayerSubnetwork::L2TopoType_T topoType,
        in unsigned long how_many,
        out globaldefs::NamingAttributesList_T nameList,
        out globaldefs::NamingAttributesIterator_I nameIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface provides a list of all available ML-series equipment names that can be used in creating an RPR L2 Topology, or a list of all available ML-series TP names that can be used in creating a point-to-point L2 topology. The returned list of ML-series equipment or TP names does not guarantee capability to create an L2 Topology. The list of ML-series equipment or TP names indicates the equipment or TP that is not participating in an L2 topology at this time.

Parameters

Name	Type	Input/ Output	Description
managedElementList	globaldefs::NamingAttributesList_T	In	List of MEs from which available ML-series cards must be determined.
topoType	multiLayerSubnetwork::L2TopoType_T	In	Topology type that is created using the list. Prime Optical supports L2TopoType_T.L2TT_RPR or L2TopoType_T.L2TT_PT2PT.
how_many	unsigned long	In	Maximum number of equipment or TP names to return in the first batch.
eqorTPNamesList	NamingAttributesList_T	Out	First batch of equipment or TP names.
eqorTPNamesIt	NamingAttributesIterator_I	Out	Iterator used to retrieve the remaining equipment or TP names.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any of the managed element names do not refer to a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when any of the managed elements do not exist in Prime Optical.

Compliance

Prime Optical-defined.

3.11 E-Series VLAN Interfaces

VLAN is required for single-card mode or multicard mode Ethernet SNC creation on E-series cards for ONS 15327 and ONS 15454 NEs. Prime Optical provides methods to create and delete VLAN. In Prime Optical, VLAN is created within a subnetwork and is available to all objects of a subnetwork in EMS.

Each VLAN is described by a unique name, an ID, and the name of the multilayer subnetwork. A network partition can contain more than one multilayer subnetwork. VLAN names and IDs are unique within the network partition.

```
typedef long VlanId;
struct Vlan_T {
        VlanId id;
        string name;
        globaldefs::NamingAttributes_T subnetname;
};
```

3.11.1 createVLAN

Synopsis

```
void createVLAN in globaldefs::NamingAttributes_T subnetname,
        in VlanId vlanId,
        in string name,
        out Vlan_T vlan)
     raises(globaldefs::ProcessingFailureException);
```

Description

Creates a new VLAN with an ID and name associated with the multilayer subnetwork.
The scope of the VLAN is network partition wide. A network partition is a collection of the multilayer subnetwork.

Parameters

Name	Type	Input/ Output	Description
subnetname	globaldefs::NamingAttributes_T	In	Multilayer subnetwork name to which to associate this VLAN.
vlanId	VlanId	In	ID of the VLAN to create.
name	String	In	Name of the VLAN to create.
vlan	Vlan_T	Out	VLAN object created by Prime Optical.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INVALID_INPUT - Raised when a VLAN ID or subnet name is invalid.
EXCPT_ENTITY_NOT_FOUND - Raised when a subnet name does not exist in the EMS.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
```

Compliance

Prime Optical-defined.

3.11.2 deleteVLAN

Synopsis

```
void deleteVLAN(
        in globaldefs::NamingAttributes_T subnetname,
        in VlanId vlanId)
     raises(globaldefs::ProcessingFailureException);
```

Description
This interface deletes an existing VLAN with a VLAN ID associated with the multilayer subnetwork.
Parameters

Name	Type	Input/Output	Description
subnetname	globaldefs::NamingAttributes_T	In	Multilayer subnetwork name to which this VLAN is associated.
vlanId	VlanId	In	ID of the VLAN to delete.

Throws
globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised when a VLAN ID or subnet name is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when a subnet name does not exist in the EMS.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.11.3 getAllVLANs

Synopsis
```
void getAllVLANs(
        in globaldefs::NamingAttributes_T subnetname,
        out VlanList_T vlanList)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface returns a list of all VLANs for the multilayer subnetwork. All the VLANs for the network partition that contain the multilayer subnetwork are reported.
Parameters

Name	Type	Input/ Output	Description
subnetname	globaldefs::NamingAttributes_T	In	Multilayer subnetwork name for which to obtain all the VLANs.
vlanList	VlanList_T	Out	List of available VLANs for this EMS.

Throws
globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised when a subnet name is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when a subnet name does not exist in the EMS.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.11.4 modifySNC

Synopsis
```
public void modifySNC(
NameAndStringValue_T[]        sncName,
                        String                        routeId,
                        SNCModifyData_T               SNCModifyData,
                        GradesOfImpact_T              tolerableImpact,
```

```
                        ProtectionEffort_T            tolerableImpactEffort,
                        EMSFreedomLevel_T             emsFreedomLevel,
                        TPDataList_THolder            tpsToModify,
                        SubnetworkConnection_THolder  modifiedSNC,
                        StringHolder                  errorReason)
    throws ProcessingFailureException
```

Description

This API:

- Allows the modification of SNC name (for all SNC types).
- Allows the modification of a GMPLS SNC Restore/Revert configuration.
- Allows the modification of GMPLS SNC routing constraints.
- Forces a specific procedure for a GMPLS SNC.

The modification of an OCHCC GMPLS SNC, when Restore/Revert attributes are modified or when an operation is invoked specifying the gmplsAction attribute, requires a background request validation. The API dedicated to modification of OCHCC GMPLS SNCs performs a validation check. The SNC is modified after a network validation check and the circuit is updated only if all checks are successfully cleared.

The modification of the routing constraints cannot be applied to an OCHCC GMPLS SNC.

The modification of the routing constraints of protected OCHNC SNCs (PSM protection is the only protection type valid in this case) requires an additional parameter from the additionalInfo attribute list called “gmplsAppliedPath”.

Parameters

Name	Type	Input/ Output	Description
sncName	NameAndStringValue_T[]	in	The MultiLayerSubnetwork name.
SNCModifyData	SNCModifyData_T	in	Contains a new SNC name and a list of additional info attributes that need to be set.
modifiedSNC	SubnetworkConnection_THolder	out	Modified SNC

Relevant data structures:

```
SNCModifyData_T
    String                  userLabel
    boolean                 forceUniqueness
    String                  owner
    ConnectionDirection_T   direction
```

String	*modifyType*
boolean	*retainOldSNC*
boolean	*modifyServers_allowed*
StaticProtectionLevel_T	*staticProtectionLevel*
ProtectionEffort_T	*protectionEffort*
Reroute_T	*rerouteAllowed*
NetworkRouted_T	*networkRouted*
SNCType_T	*sncType*
short	*layerRate*
RouteDescriptor_T[]	*addedOrNewRoute*
RouteDescriptor_T[]	*removedRoute*
NameAndStringValue_T[][]	**neTpInclusions**
boolean	*fullRoute*
NameAndStringValue_T[][]	**neTpSncExclusions**
NameAndStringValue_T[][]	*aEnd*
NameAndStringValue_T[][]	*zEnd*
NameAndStringValue_T[]	**additionalCreationInfo**

In the preceding example, the managed input parameters are highlighted in Bold font.
The modifySNC API does one of the following:

- Changes the name of an SNC passed in userLabel
- Applies the list of attributes specified in additionalCreationInfo
- Modifies the SNC routing that specifies the nodes that must be included or excluded.

✎ Only one operation can be applied at a time. If more information is passed, an exception is returned.

The userLabel attribute must be empty in order to leave the SNC name unchanged and perform one of the allowed operations – modify routing contraints, modify Revert/Restore configurations, or perform procedure.

String **userLabel***:*

If the userLabel attribute is not empty, specify a new SNC name:

NameAndStringValue_T[][] **neTpInclusions***:*

This is applicable for GMPLS SNCs only and lists out the ManagedElement names that must be included. This parameter can only define the inclusion routing constraints for the input SNC.

NameAndStringValue_T[][] **neTpSncExclusions**

This is applicable for GMPLS SNCs only and lists out the ManagedElement names that must be excluded. This parameter can only define the exclusion routing constraints for the input SNC.

NameAndStringValue_T[] **additionalCreationInfo**

The additionalCreationInfo can contain the list of attributes listed in the following table.

Name	Value	Comment/Example
gmplsAction	String	Identifies the name of the procedure that must be applied to the SNC. Valid values are: • Validate • Promote • ManualRevert • UpgradeRestored ✐ If the value is Validate, ManualRevert, or UpgradeRestored, no more attributes are required in the additionalCreationInfo structure, and the applied SNC must be a GMPLS SNC. ✐ If the value is Promote, two additional parameters must be specified: gmplsOptQual and gmplsAcptThreshold. The applied SNC cannot be a GMPLS SNC.
gmplsOptQual	String	See the GMPLS Provisioning section.
gmplsAcptThreshold	String	
gmplsIgnorePathAlarms	String	See Table 2-21, GMPLS Revert/Restore Parameters, in the "GMPLS Provisioning"section. ✐ The gmplsIgnorePathAlarms, gmplsRestoration, gmplsRevertiveParameters, and gmplsSoakTime attributes are not available if the gmplsAction attribute is specified.
gmplsRestoration	String	
gmplsRevertiveParameters	String	
gmplsSoakTime	String	
gmplsAppliedPath	String	Valid values are: • Working • Protected This parameter identifies the applied path of the circuit and is required if all of the following conditions are met: • The modify request is for routing constraints (neTpInclusions or neTpSncExclusions are specified) • The applied circuit type is OCHNC • The applied circuit is protected (the only valid protection type for OCHNC circuits is PSM)
CircuitAdminState	String	Required parameter. The CircuitAdminState attribute is applied to all the ports involved by the circuit. Valid SDH values are: • UNLOCKED • LOCKED,DISABLED • LOCKED,MAINTENANCE • UNLOCKED,AUTO_IN_SERVICE • LOCKED,OUT_OF_GROUP Valid SONET values are: • IN_SERVICE • OUT_OF_SERVICE • OUT_OF_SERVICE_BY_MAINTENANCE • AUTO_IN_SERVICE • OUT_OF_GROUP Valid for ALL

		• UNKNOWN_ADMIN_STATE
gmplsUserLabel	String	See the GMPLS Provisioning section.
gmplsPriority	String	
gmplsPowerOffsetUp	float	
gmplsPowerOffsetDown	float	

✐ The GMPLS Revert/Restore attributes (gmplsIgnorePathAlarms, gmplsRestoration, gmplsRevertiveParameters, gmplsSoakTime) cannot be present if gmplsAction has been specified.

✐ The TestRevert operation in the Prime Optical client can be implemented by setting the gmplsRestoration attribute to false.

3.12 nmsSession::NmsSession_I

The nmsSession module enables the EMS to inform the NMS of problems with notifications. When Prime Optical fails to push an event, it notifies all connected NMSs by invoking the eventLossOccurred method on every active NmsSession_I. When the event loss period is over, Prime Optical invokes the eventLossCleared method on NmsSession_I.
Both methods are implemented by the NMS.
NmsSession also implements all methods defined in 3.15 session::Session_I.
Prime Optical requires that NmsSession implement two new methods that are used by Prime Optical to notify NmsSession that the call to the getCTMHistoryPMData method has failed or succeeded.

3.12.1 historyPMDataCompleted

Synopsis

```
void historyPMDataCompleted(
        in string fileName)
```

Description
When getCTMHistoryPMData completes successfully, Prime Optical invokes this method on the NMS session. At that time, NMS can retrieve the performance monitoring (PM) file from the Prime Optical server.

Parameters

Name	Type	Input/ Output	Description
fileName	string	In	Name of the file created by Prime Optical in response to the getCTMHistoryPMData call made by the NMS. This name is the absolute path on Prime Optical server.

Throws
None.
Compliance
Prime Optical-defined.

3.12.2 historyPMDataFailed

Synopsis

```
void historyPMDataFailed(
        out string errorReason)
```

Description
When getCTMHistoryPMData fails, Prime Optical invokes this method on the NMS session. Errors encountered by Prime Optical are stored in the errorReason parameter.
Parameters

Name	Type	Input/Output	Description
errorReason	string	Out	Error string to indicate to the NMS the reason for the failure.

Throws
None.

Compliance
Prime Optical-defined.

3.12.3 operationInfo

Synopsis
```
void operationInfo(in string opname,
      in globaldefs::NVSList_T additionalInfo);
```

Description
The EMS invokes this interface on the NMS session when configConnLessInfos or configConnLessInfos is complete. The error encountered by the EMS is included in the errorReason parameter.
Parameters

Name	Type	Input/Output	Description
opname	string	in	Name of the operation performed or completed.
additionalInfo	NVSList_T	in	List of names and values that reports the operation status.

Throws
None.
Compliance
Prime Optical-defined.

3.12.4 alarmLossOccurred

Synopsis
```
      void alarmLossOccurred(
            in globaldefs::Time_T startTime,
              in string notificationId)
```

Description
When the EMS fails to discard an alarm, TCA, file transfer status, or protection switch notification, it notifies all connected NMSs by invoking this method on every active NmsSession_I. This interface is also invoked on any new NmsSession_I set up during the event loss period.
When the EMS invokes this method on the NmsSession_I, it sets an internal flag to indicate that it has informed the NMSs of the alarm loss. As long as this flag is set, the EMS does not invoke this method again. However, the EMS might invoke eventLossOccurred if it fails to push a different type of event.
When this method is invoked on an NmsSession_I, the NMS learns whether the EMS failed to push one or more alarms, TCAs, file transfers, or protection switch notifications. The NMS is out-of-sync with the EMS for these types of notifications. The NSM waits until the EMS calls eventLossCleared before resynchronizing alarms, TCAs, file transfers, and protection switch data with the EMS.
Parameters

Name	Type	Input/Output	Description
startTime	globaldefs::Time_T	In	Time of the first lost notification.
notificationId	string	In	ID of the first lost notification.

Throws
None.
Compliance
TMF-defined.

3.13 performance::PerformanceManagementMgr_I

This interface accesses operations that deal with performance element monitoring. A handle to an instance of this interface is gained through the getManager operation in emsSession. This interface is inherited from 3.2 common::Common_I.

```
interface PerformanceManagementMgr_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.
Whole number PM values are converted to floating point values in CORBA. In the floating point format, we can have numbers with rounded off values greater than 23 bit (> 8388608). Hence a PM value greater than 23 bits may look different from the floating point format.

For example, a PM value of 857224230528, when converted to a floating point value, becomes 8.5722425E11.

3.13.1 disablePMData

Synopsis

```
void disablePMData(
in PMTPSelectList_T pmTPSelectList,
out PMTPSelectList_T failedTPSelectList)
raises(globaldefs::ProcessingFailureException);
```

Description
This interface instructs the EMS to turn off PM data collection for a list of measurement points. Disabling is done on a best-effort basis. If PM could not be disabled for a subset of cases in the PMTPSelectList, a list identifying this subset is returned. PM collection involves collection of historical PM data. If this method is called between PM collection intervals, PM data for that interval is not collected. There are no incomplete collection periods.
Example
For PM collection intervals T-15min, T, T + 15min and so on, and a disablePMData call at T + 3min, the PM data is only collected up to and including interval T. T + 15min interval PM data is not collected.
If PM collection was never started for this TP, the operation is considered successful.
Parameters

Name	Type	Input/ Output	Description
pmTPSelectList	PMTPSelectList_T	In	This structure contains the relevant data for the disablePMData request. This value cannot be empty.
failedTPSelectList	PMTPSelectList_T	Out	List of points that were not completely disabled. An empty list indicates that the total request was successful.

Prime Optical provides the following support for the PMTPSelect_T structure for this method:

- globaldefs::NamingAttributes_T name
  - The name of the object to which this selection applies. Prime Optical only supports ManagedElement name in this field.
- transmissionParameters::LayerRateList_T layerRateList

  - Not supported.
- PMLocationList_T pMLocationList
  - Not supported.
- GranularityList_T granularityList
  - Supported only for enablePMData. 15min and 24h are the only valid values. If the list is empty, both 15min and 24h are considered.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised if pmTPSelectList is empty.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

TMF-defined.

3.13.2 enablePMData

Synopsis

```
void enablePMData(in PMTPSelectList_T pmTPSelectList,
out PMTPSelectList_T failedTPSelectList)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface instructs the EMS to enable (turn on) PM data collection for a list of measurement points. Enabling is done on a best-effort basis. If PM could not be enabled for a subset of cases in the pmTPSelectList, a list identifying this subset is returned. PM collection does not start immediately as required by the TMF. There are no incomplete collection periods. If PM collection was already on for a TP, the operation is considered successful.

Parameters

Name	Type	Input/ Output	Description
pmTPSelectList	PMTPSelectList_T	In	This structure contains the relevant data for the enablePMData request. This value cannot be empty.
failedTPSelectList	PMTPSelectList_T	Out	List of points that were not enabled completely. An empty list indicates that the total request succeeded.

For details about the PMTPSelect_T structure, see 3.13.1 disablePMData.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised if pmTPSelectList is empty.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

TMF-defined.

3.13.3 setTCATPParameter

Synopsis

```
void setTCATPParameter( in globaldefs::NamingAttributes_T tpName,
```

```
    inout TCAParameters_T tcaParameters)
  raises(globaldefs::ProcessingFailureException);
```

Description

This interface modifies the TCA/alarm threshold values on a TP-layer rate measurement point. This interface can be applied to a PTP or CTP. The NMS specifies the TP and layer rate, the granularity, and the values of one or more TCA/alarm thresholds to modify. The operation mode is best-effort. If no TCA/alarm value is set, the default value is active.

A set of TCA/alarm thresholds is supported for Optical, OTN, SONET, and SDH layer rates.

If Prime Optical fails to set TCA/alarm parameters, an exception is generated. A successful return of this operation guarantees that all values are set. If a parameter fails, the remaining parameters are unchanged. Prime Optical does not send events or updates when TCA/alarm parameters are changed. It also does not validate the range of PM parameter values.

The NMS always makes synchronous calls to this method.

Parameters

Name	Type	Input/Output	Description
TpName	globaldefs::NamingAttributes_T	In	Identification of the TP for which the values of one or more TCA parameters are set.
TcaParameters	TCAParameters_T	In/out	List of TCA IDs and corresponding values to apply to a specific layer rate. The settings that are actually applied are returned in this parameter.

Prime Optical support for TCAParameters_T structure is as follows:

- transmissionParameters::LayerRate_T layerRate
  - For a list of supported layer rates, see 2.2.1 Layer Rate.
- Granularity_T granularity
  - Prime Optical supports 15 minute (15min) and 24 hour (24h).
- PMThresholdValueList_T tcaTypeValues
  - This is a sequence of PMThresholdValue_T. It contains the PMParameterName_T pmParameterName field.

Prime Optical supports the NE PM values listed in Table 3-5, Table 3-6, Table 3-7, Table 3-9, Table 3-10, and Table 3-11.

Table 3-6: ONS 15310 CL and ONS 15310 MA SONET TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_POS_100M LR_POS_GIGABIT			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	—
	PMP_RX_PACKETS		
	PMP_IFINDISCARDS		
	PMP_RX_TOTAL_ERRORS		
	PMP_IFOUTOCTETS		
	PMP_TX_PACKETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSCSFRAISED		
	PMP_ETHERSTATSDROPEVENTS		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
	PMP_HDLCRXABORTS		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_RXPKTSDROPINTERNALCONGESTION		
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RXPACKETS		
	PMP_IFINUCASTPKTS		
	PMP_IFINMULTICASTPKTS		
	PMP_IFINBROADCASTPKTS		
	PMP_IFINDISCARDS		
	PMP_IFINERRORS		
	PMP_IFOUTOCTETS		
LR_Fast_Ethernet **LR_Gigabit_Ethernet**			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RXPACKETS		
	PMP_IFINUCASTPKTS		
	PMP_IFINMULTICASTPKTS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_IFINBROADCASTPKTS		
	PMP_IFINDISCARDS		
	PMP_IFINERRORS		
	PMP_IFOUTOCTETS		
PTP	PMP_TXPACKETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_IFOUTUCASTPKTS		
	PMP_IFOUTMULTICASTPKTS		
	PMP_IFOUTBROADCASTPKTS		
	PMP_DOT3STATSALIGNMENTERRORS		
	PMP_DOT3STATSFCSERRORS		
	PMP_DOT3STATSSINGLECOLLISIONFRAMES		
	PMP_DOT3STATSFRAMETOOLONG		
	PMP_ETHERSTATSUNDERSIZEPKTS		
	PMP_ETHERSTATSFRAGMENTS		
	PMP_ETHERSTATSPKTS64OCTETS		
	PMP_ETHERSTATSPKTS65TO127OCTETS		
	PMP_ETHERSTATSPKTS128TO255OCTETS		
	PMP_ETHERSTATSPKTS256TO511OCTETS		
	PMP_ETHERSTATSPKTS512TO1023OCTETS		
	PMP_ETHERSTATSPKTS1024TO1518OCTETS		
	PMP_ETHERSTATSBROADCASTPKTS		
	PMP_ETHERSTATSMULTICASTPKTS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ETHERSTATSOVERSIZEPKTS		
	PMP_ETHERSTATSJABBERS		
	PMP_ETHERSTATSOCTETS		
	PMP_ETHERSTATSCOLLISIONS		
	PMP_ETHERSTATSCRCALIGNERRORS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_PORTDROPCOUNTS		
	PMP_ETHERSTATSCOLLISIONFRAMES		
	PMP_IFOUTDISCARDS		
	PMP_MEDIAINDSTATSOVERSIZEDROPPED		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
LR_COS_100M **LR_COS_Gigabit**			
PTP	PMP_INTERFACE_NAME	PML_NEAR_END_RX	15 min, 24 hr
	PMP_INTERFACE_DIRECTION		
	PMP_CLASS_OF_SERVICE_LEVEL		
	PMP_POST_POLICY_PACKETS		
	PMP_POST_POLICY_BYTES		
	PMP_DROP_PACKETS		
	PMP_DROP_BYTES		
LR_T1_and_DS1_1_5M			
✐ VT1.5 data is retrieved using this layer rate. STS data retrieval is not supported.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
LR_GFP_Multirate			
PTP	PMP_GFPSTATSCSFRAISED	PML_NEAR_END_RX	15 min, 24 hr
	PMP_GFPSTATSLFDRAISED		
	PMP_GFPSTATSROUNDTRIPLATENCY		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSRXFRAME		

PTP or CTP	pmParameterName	Location	Interval
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXSBLKCRCERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
LR_T3_and_DS3_45M LR_STS1_and_AU3_High_Order_VC3 LR_STS3c_and_AU4_VC4 LR_STS12c_and_VC4_4c LR_STS6c_and_VC4_2c LR_STS9c_and_VC4_3c LR_STS48c_and_VC4_16c (ONS 15310 MA)			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_FC		
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_PPJC_PDET	PML_NEAR_END_RX	15 min, 24 hr
	PMP_NPJC_PDET		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PJDIFF		
	PMP_PJCS_PDET		
	PMP_PJCS_PGEN		
LR_Section_O C3_STS3c_an d_RS_STM1 LR_Section_O C12_STS12c_ and_RS_STM4 LR_Section_O C48_STS48c_ and_ RS_STM16 (ONS 15310 MA)			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_SEFS		

PTP or CTP	pmParameterName	Location	Interval
LR_Line_OC3_STS3_and_ MS_STM1 LR_Line_OC12_STS12_and_ MS_STM4 LR_Line_OC48_STS48_and_ MS_STM16 (ONS 15310 MA)			
PTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_AISS_PBIT		
	PMP_CVP_PBIT		
	PMP_ESP_PBIT		
	PMP_SASP_PBIT		
	PMP_SESP_PBIT		
	PMP_UASP_PBIT		
	PMP_CVP_CPBIT	PML_NEAR_END_RX, PML_FAR_END_RX	
	PMP_ESP_CPBIT		
	PMP_SESP_CPBIT		
	PMP_UASP_CPBIT		
	PMP_SASP_CPBIT	PML_FAR_END_RX	
LR_T1_and_DS1_1_5M			
PTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_CV_P	PML_NEAR_END_RX, PML_NEAR_END_TX	
	PMP_ES_P		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SES_P		
	PMP_SAS_P		
	PMP_UAS_P		
	PMP_AISS_P		
	PMP_FCAISS_P		
	PMP_CV_P	PML_FAR_END_RX	
	PMP_ES_P		
	PMP_ESA_P		
	PMP_ESB_P		
	PMP_SES_P		
	PMP_SEFS_P		
	PMP_CSS_P		
	PMP_UAS_P		
	PMP_ES		
	PMP_FC_P		
	PMP_ES		
LR_VT1_5_and_TU11_VC11			
CTP	Same as ONS 15327	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr

Table 3-7: ONS 15327 TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_Fast_Ethernet LR_Gigabit_Ethernet			
PTP	PMP_DOT3STATSALIGNMENTERRORS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_DOT3STATSCARRIERSENSEERRORS		
	PMP_DOT3STATSDEFERREDTRANSMISSIONS		
	PMP_DOT3STATSEXCESSIVECOLLISIONS		
	PMP_DOT3STATSFCSERRORS		
	PMP_DOT3STATSFRAMETOOLONG		

PTP or CTP	pmParameterName	Location	Interval
	PMP_DOT3STATSLATECOLLISIONS		
	PMP_DOT3STATSMULTIPLECOLLISIONFRAMES		
	PMP_DOT3STATSSINGLECOLLISIONFRAMES		
	PMP_DOT3STATSSQETESTERRORS		
	PMP_ETHERSTATSBROADCASTPKTS		
	PMP_ETHERSTATSCOLLISIONFRAMES		
	PMP_ETHERSTATSCOLLISIONS		
	PMP_ETHERSTATSCRCALIGNERRORS		
	PMP_ETHERSTATSFRAGMENTS		
	PMP_ETHERSTATSJABBERS		
	PMP_ETHERSTATSMULTICASTPKTS		
	PMP_ETHERSTATSOCTETS		
	PMP_ETHERSTATSOVERSIZEPKTS		
	PMP_ETHERSTATSPKTS1024TO1518OCTETS		
	PMP_ETHERSTATSPKTS128TO255OCTETS		
	PMP_ETHERSTATSPKTS256TO511OCTETS		
	PMP_ETHERSTATSPKTS512TO1023OCTETS		
	PMP_ETHERSTATSPKTS64OCTETS		
	PMP_ETHERSTATSPKTS65TO127OCTETS		
	PMP_ETHERSTATSUNDERSIZEPKTS		
	PMP_HDLCPKTDROPS		
	PMP_IFINBROADCASTPKTS		
	PMP_IFINDISCARDS		
	PMP_IFINERRORS		
	PMP_IFINMULTICASTPKTS		
	PMP_IFINOCTETS		
	PMP_IFINUCASTPKTS		
	PMP_IFOUTBROADCASTPKTS		
	PMP_IFOUTDISCARDS		
	PMP_IFOUTMULTICASTPKTS		
	PMP_IFOUTOCTETS		
	PMP_IFOUTUCASTPKTS		
	PMP_RXETHERUTILIZATIONSTATS		
	PMP_RXPAUSEFRAMES		
	PMP_RXPKTSDROPPEDINTERNALCONGESTION		
	PMP_RXPACKETS		
	PMP_TXETHERUTILIZATIONSTATS		
	PMP_TXPAUSEFRAMES		
	PMP_TXPKTSDROPPEDINTERNALCONGESTION		

PTP or CTP	pmParameterName	Location	Interval
	PMP_TXPACKETS		
PTP	PMP_ETHERSTATSPKTS1519TOMAXOCTETS	PML_NEAR_END_Tx	15 min, 24 hr
LR_T1_and_DS1_1_5M			
✐ VT1.5 data is retrieved using this layer rate. STS data retrieval is not supported.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
LR_T3_and_DS3_45M LR_STS1_and_AU3_High_Order_VC3 LR_STS3c_and_AU4_VC4 LR_STS12c_and_VC4_4c LR_STS6c_and_VC4_2c LR_STS9c_and_VC4_3c LR_STS24c_and_VC4_8c LR_STS48c_and_VC4_16c			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_FC		
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_PPJ	PML_NEAR_END_RX	
	PMP_NPJ		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PJ_DIFF		
	PMP_PJP_SEC		
	PMP_PJN_SEC		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
LR_Section_OC3_STS3c_and_RS_STM1 LR_Section_OC12_STS12c_and_RS_STM4 LR_Section_OC48_STS48c_and_RS_STM16			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_SEFS		
LR_Line_OC3_STS3_and_MS_STM1 LR_Line_OC12_STS12_and_MS_STM4 LR_Line_OC48_STS48_and_MS_STM16			
PTP	PMP_CV	PML_NEAR_END_RX,	15 min,

PTP or CTP	pmParameterName	Location	Interval
	PMP_ES	PML_FAR_END_RX	24 hr
	PMP_SES		
	PMP_FC		
	PMP_UAS		
	PMP_PSC	PML_NEAR_END_RX	15 min, 24 hr
	PMP_PSD		
	PMP_PPJ		
	PMP_NPJ		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PSC_W		
	PMP_PSD_W		
	PMP_PSC_S		
	PMP_PSD_S		
	PMP_PSC_R		
	PMP_PSD_R		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
	PMP_PJ_DIFF		
LR_T3_and_DS3_45M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
LR_T1_and_DS1_1_5M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_CV_P	PML_NEAR_END_RX, PML_NEAR_END_TX	
	PMP_ES_P		
	PMP_SES_P		
	PMP_SAS_P		
	PMP_UAS_P		
	PMP_AISS_P		
	PMP_CV_P	PML_FAR_END_RX	15 min, 24 hr
	PMP_ES_P		
	PMP_ESA_P		
	PMP_ESB_P		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SES_P		
	PMP_SEFS_P		
	PMP_CSS_P		
	PMP_UAS_P		
	PMP_ES		
LR_VT1_5_and_TU11_VC11			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		

For CPT 200 and CPT 600, a subset of ONS 15454 SONET and ONS 15454 SDH parameters is supported.
The supported parameters match the parameters associated to the port equipment supported by CPT.

Table 3-8: ONS 15454 SONET, ONS NCS2K SONET, CPT 200, and CPT 600 TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_Fast_Ethernet* LR_Gigabit_Ethernet* LR_Ten_Gigabit_Ethernet * LR_OneHundred_Gigabit_Ethernet*			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_IFINUCASTPKTS		
	PMP_IFINMULTICASTPKTS		
	PMP_IFINBROADCASTPKTS		
	PMP_IFINDISCARDS		
	PMP_IFOUTDISCARDS		
	PMP_IFOUTMULTICASTPKTS		
	PMP_IFOUTBROADCASTPKTS		
	PMP_ETHERSTATSBROADCASTPKTS		
	PMP_ETHERSTATSMULTICASTPKTS		
	PMP_PORTDROPCOUNTS		
	PMP_IFINERRORS		
	PMP_IFOUTOCTETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_IFOUTUCASTPKTS		
	PMP_TXPACKETS		
	PMP_RXPACKETS		
	PMP_RXPAUSEFRAMES		
	PMP_TXPAUSEFRAMES		
	PMP_RXPKTSDROPPEDINTERNALCONGESTION		
	PMP_TXPKTSDROPPEDINTERNALCONGESTION		
	PMP_HDLCPKTDROPS		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSTXFRAMESBADCRC		
	PMP_RXCONTROLFRAMES		
	PMP_RXUNKNOWNOPCODEFRAMES		
	PMP_IFINERRORBYTEPKTS		
	PMP_IFINFRAMINGERRORPKTS		
	PMP_IFINJUNKINTERPKTS		
PTP	PMP_ETHERSTATSPKTS1519TOMAXOCTETS	PML_NEAR_END_Tx	15 min, 24 hr
	PMP_DOT3STATSSYMBOLERRORS	PML_BIDIRECTIONAL	
LR_Fast_Ethernet LR_Gigabit_Ethernet (continued)			
PTP	PMP_MEDIAINDSTATSOVERSIZEDROPPED	PML_NEAR_END_RX	15 min, 24 hr
	PMP_DOT3STATSALIGNMENTERRORS		
	PMP_DOT3STATSFCSERRORS		
	PMP_DOT3STATSSINGLECOLLISIONFRAMES		
	PMP_DOT3STATSCARRIERSENSEERRORS		
	PMP_DOT3STATSSQETESTERRORS		
	PMP_DOT3STATSMULTIPLECOLLISIONFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_DOT3STATSDEFERREDTRANSMISSIONS		
	PMP_DOT3STATSLATECOLLISIONS		
	PMP_DOT3STATSEXCESSIVECOLLISIONS		
	PMP_DOT3STATSFRAMETOOLONG		
	PMP_ETHERSTATSUNDERSIZEPKTS		
	PMP_ETHERSTATSFRAGMENTS		
	PMP_ETHERSTATSPKTS64OCTETS		
	PMP_ETHERSTATSPKTS65TO127OCTETS		
	PMP_ETHERSTATSPKTS128TO255OCTETS		
	PMP_ETHERSTATSPKTS256TO511OCTETS		
	PMP_ETHERSTATSPKTS512TO1023OCTETS		
	PMP_ETHERSTATSPKTS1024TO1518OCTETS		
	PMP_ETHERSTATSOVERSIZEPKTS		
	PMP_ETHERSTATSJABBERS		
	PMP_ETHERSTATSOCTETS		
	PMP_ETHERSTATSCOLLISIONS		
	PMP_ETHERSTATSCOLLISIONFRAMES		
	PMP_ETHERSTATSCRCALIGNERRORS		
	PMP_RXETHERUTILIZATIONSTATS		
	PMP_TXETHERUTILIZATIONSTATS		
	PMP_RXGIANTS		
	PMP_RXGMACDROPCOUNTS		
	PMP_RXRUNTS		
	PMP_RXTHRESHOLDOVERSIZES		
	PMP_RXTOTALERRORS		
	PMP_TXCOLLISIONS		
	PMP_TXGIANTS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_DOT3STATSCONTROLINUNKNOWNOPCODES		
	PMP_DOT3STATSINPAUSEFRAMES		
	PMP_DOT3STATSOUTPAUSEFRAMES		
	PMP_ETHERSTATSPKTS1519TO1522OCTETS		
LR_POS_100M LR_POS_Gigabit			
PTP	PMP_RX_INPUT_DROP_PACKETS	PML_NEAR_END_RX	15 min, 24 hr
	PML_NEAR_END_RX		
	15 min, 24 hr		
	PMP_RX_INPUT_ABORT_PACKETS		
	PMP_TX_PRE_HDLC_BYTES		
	PMP_TX_POST_HDLC_BYTES		
	PMP_TX_PACKETS		
	PMP_PORT_DROP_COUNTS		
	PMP_RX_PRE_HDLC_BYTES		
	PMP_RX_POST_HDLC_BYTES		
	PMP_RX_PACKETS		
	PMP_RX_NORMAL_PACKETS		
	PMP_RX_SHORTS		
	PMP_RX_RUNTS		
	PMP_RX_LONGS		
	PMP_RX_TOTAL_ERRORS		
	PMP_RX_CRC_ERRORS		
	PMP_ETHERSTATSDROPEVENTS		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSRXSHORTPKTS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_HDLCINOCTETS		
	PMP_HDLCRXABORTS		
	PMP_HDLCOUTOCTETS		
	PMP_IFINOCTETS		
	PMP_IFOUTOCTETS		
	PMP_IFINDISCARDS		
	PMP_RXPKTSDROPINTERNALCONGESTION		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSCSFRAISED		
	PMP_GFPSTATSLFDRAISED		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
LR_COS_100M **LR_COS_Gigabit**			
PTP	PMP_INTERFACE_NAME	PML_NEAR_END_RX	15 min, 24 hr
	PMP_INTERFACE_DIRECTION		
	PMP_CLASS_OF_SERVICE_LEVEL		
	PMP_POST_POLICY_PACKETS		
	PMP_POST_POLICY_BYTES		
	PMP_DROP_PACKETS		
	PMP_DROP_BYTES		
LR_T1_and_DS1_1_5M			

PTP or CTP	pmParameterName	Location	Interval
✐ VT1.5 data is retrieved using this layer rate. STS data retrieval is not supported.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
LR_GFP_Multirate			
PTP	PMP_GFPSTATSCSFRAISED	PML_NEAR_END_RX	15 min, 24 hr
	PMP_GFPSTATSLFDRAISED		
	PMP_GFPSTATSROUNDTRIPLATENCY		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXSBLKCRCERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
	PMP_GFPSTATSRXDISTANCEEXTBUFFERS		
	PMP_GFPSTATSTXDISTANCEEXTBUFFERS		
	PMP_GFPSTATSCHECRXMBITERRORS		
	PMP_GFPSTATSTHECRXMBITERRORS		
LR_E1_2M			
✐ VT2 data is retrieved using this layer rate.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_UAS		
LR_E1_2M			
✐ E1 data is retrieved using this layer rate.			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_CV_P		
	PMP_BBE_P		
	PMP_ES_P		
	PMP_SES_P		
	PMP_UAS_P		
	PMP_SAS_P		
	PMP_FC_P		
	PMP_AISS_P		
LR_OCH_ISC_PEERCOMPAT			
PTP	PMP_RXOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RXPACKETS		
	PMP_TXOCTETS		
	PMP_TXPACKETS		
	PMP_INVLD_ORDRD_SET_DISP_ERROR_SUM		
LR_ESCON			
PTP	PMP_IFINERRORS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_TXPACKETS		
	PMP_IFOUTDISCARDS		
	PMP_IFINDISCARDS		
	PMP_IFOUTOCTETS		
	PMP_RXPACKETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_IFINOCTETS		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_TXETHERUTILIZATIONSTATS		
	MP_RXETHERUTILIZATIONSTATS		
	PMP_RXOCTETS		
	PMP_RXPACKETS		
	PMP_TXOCTETS		
	PMP_TXPACKETS		
LR_LEX			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_UAS		
	PMP_SES		
	PMP_ES		
	PMP_FC		
	PMP_NPJC_PGEN		
	PMP_NPJC_PDET		
	PMP_PJCS_PDET		
	PMP_PJCS_PGEN		
	PMP_PJCDIFF		
LR_RPR_IEEE_802_17			
PTP	PMP_SPANOUTMCASTCLASSAOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_SPANOUTCTRLFRAMES		
	PMP_SPANOUTOAMECHOFRAMES		
	PMP_SPANOUTOAMFLUSHFRAMES		
	PMP_SPANOUTOAMORGFRAMES		
	PMP_SPANOUTTOPOATDFRAMES		
	PMP_SPANOUTTOPOCHKSUMFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SPANOUTTOPOTPFRAMES		
	PMP_CLIENTINUCASTCLASSCFRAMES		
	PMP_CLIENTINUCASTCLASSCOCTETS		
	PMP_CLIENTINMCASTCLASSCFRAMES		
	PMP_CLIENTINMCASTCLASSCOCTETS		
	PMP_CLIENTINUCASTCLASSBEIRFRAMES		
	PMP_CLIENTINUCASTCLASSBEIROCTETS		
	PMP_CLIENTINMCASTCLASSBEIRFRAMES		
	PMP_CLIENTINMCASTCLASSBEIROCTETS		
	PMP_CLIENTINUCASTCLASSBCIRFRAMES		
	PMP_CLIENTINUCASTCLASSBCIROCTETS		
	PMP_CLIENTINMCASTCLASSBCIRFRAMES		
	PMP_CLIENTINMCASTCLASSBCIROCTETS		
	PMP_CLIENTINUCASTCLASSAFRAMES		
	PMP_CLIENTINUCASTCLASSAOCTETS		
	PMP_CLIENTINMCASTCLASSAFRAMES		
	PMP_CLIENTINMCASTCLASSAOCTETS		
	PMP_CLIENTINBCASTFRAMES		
	PMP_CLIENTOUTUCASTCLASSCFRAMES		
	PMP_CLIENTOUTUCASTCLASSCOCTETS		
	PMP_CLIENTOUTMCASTCLASSCFRAMES		
	PMP_CLIENTOUTMCASTCLASSCOCTETS		
	PMP_CLIENTOUTUCASTCLASSBEIRFRAMES		
	PMP_CLIENTOUTUCASTCLASSBEIROCTETS		
	PMP_CLIENTOUTMCASTCLASSBEIRFRAMES		
	PMP_CLIENTOUTMCASTCLASSBEIROCTETS		
	PMP_CLIENTOUTUCASTCLASSBCIRFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_CLIENTOUTUCASTCLASSBCIROCTETS		
	PMP_CLIENTOUTMCASTCLASSBCIRFRAMES		
	PMP_CLIENTOUTMCASTCLASSBCIROCTETS		
	PMP_CLIENTOUTUCASTCLASSAFRAMES		
	PMP_CLIENTOUTUCASTCLASSAOCTETS		
	PMP_CLIENTOUTMCASTCLASSAFRAMES		
	PMP_CLIENTOUTMCASTCLASSAOCTETS		
	PMP_CLIENTOUTBCASTFRAMES		
	PMP_ERRORBADPARITYFRAMES		
	PMP_ERRORBADHECFRAMES		
	PMP_ERRORTTLEXPFRAMES		
	PMP_ERRORTOOLONGFRAMES		
	PMP_ERRORTOOSHORTFRAMES		
	PMP_ERRORBADFCSFRAMES		
	PMP_ERRORSELFSRCUCASTFRAMES		
	PMP_ERRORPMDABORTFRAMES		
	PMP_ERRORBADADDRFRAMES		
	PMP_ERRORCONTAINEDFRAMES		
	PMP_ERRORSCFFERRORS		
	PMP_ERROROVERSIZEFRAMES		
	PMP_PORTCOUNTERERROR		
	PMP_SPANINUCASTCLASSCFRAMES		
	PMP_SPANINUCASTCLASSCOCTETS		
	PMP_SPANINMCASTCLASSCFRAMES		
	PMP_SPANINMCASTCLASSCOCTETS		
	PMP_SPANINUCASTCLASSBEIRFRAMES		
	PMP_SPANINUCASTCLASSBEIROCTETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SPANINMCASTCLASSBEIRFRAMES		
	PMP_SPANINMCASTCLASSBEIROCTETS		
	PMP_SPANINUCASTCLASSBCIRFRAMES		
	PMP_SPANINUCASTCLASSBCIROCTETS		
	PMP_SPANINMCASTCLASSBCIRFRAMES		
	PMP_SPANINMCASTCLASSBCIROCTETS		
	PMP_SPANINUCASTCLASSAFRAMES		
	PMP_SPANINUCASTCLASSAOCTETS		
	PMP_SPANINMCASTCLASSAFRAMES		
	PMP_SPANINMCASTCLASSAOCTETS		
	PMP_SPANINCTRLFRAMES		
	PMP_SPANINOAMECHOFRAMES		
	PMP_SPANINOAMFLUSHFRAMES		
	PMP_SPANINOAMORGFRAMES		
	PMP_SPANINTOPOATDFRAMES		
	PMP_SPANINTOPOCHKSUMFRAMES		
	PMP_SPANINTOPOTPFRAMES		
	PMP_SPANOUTUCASTCLASSCFRAMES		
	PMP_SPANOUTUCASTCLASSCOCTETS		
	PMP_SPANOUTMCASTCLASSCFRAMES		
	PMP_SPANOUTMCASTCLASSCOCTETS		
	PMP_SPANOUTUCASTCLASSBEIRFRAMES		
	PMP_SPANOUTUCASTCLASSBEIROCTETS		
	PMP_SPANOUTMCASTCLASSBEIRFRAMES		
	PMP_SPANOUTMCASTCLASSBEIROCTETS		
	PMP_SPANOUTUCASTCLASSBCIRFRAMES		
	PMP_SPANOUTUCASTCLASSBCIROCTETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SPANOUTMCASTCLASSBCIRFRAMES		
	PMP_SPANOUTMCASTCLASSBCIROCTETS		
	PMP_SPANOUTUCASTCLASSAFRAMES		
	PMP_SPANOUTUCASTCLASSAOCTETS		
	PMP_SPANOUTMCASTCLASSAFRAMES		
LR_T3_and_DS3_45M (on DS3XM, DS3E, DS3NE ports when supported by the NE)			
PTP	PMP_CV_P	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES_P		
	PMP_SES_P		
	PMP_SAS_P		
	PMP_AISS_P		
	PMP_UAS_P		
	PMP_CV_CP	PML_NEAR_END_RX, PML_FAR_END_RX	
	PMP_ES_CP		
	PMP_SES_CP		
	PMP_SAS_CP		
	PMP_AISS_CP		
	PMP_UAS_CP		
LR_T3_and_DS3_45M LR_STS1_and_AU3_High_Order_VC3 LR_STS3c_and_AU4_VC4 LR_STS12c_and_VC4_4c LR_STS6c_and_VC4_2c LR_STS9c_and_VC4_3c LR_STS24c_and_VC4_8c LR_STS48c_and_VC4_16c LR_STS192c_and_VC4_64c			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_FC		
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_PPJ		

PTP or CTP	pmParameterName	Location	Interval
	PMP_NPJ		
	PMP_PPJC_PGEN	PML_NEAR_END_RX	
	PMP_NPJC_PGEN		
	PMP_PJ_DIFF		
	PMP_PJP_SEC		
	PMP_PJN_SEC		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
LR_Section_OC3_STS3c_and_ RS_STM1 LR_Section_OC12_STS12c_and_RS_STM4 LR_Section_OC48_STS48c_and_RS_STM16 LR_Section_OC192_STS192c_and_RS_STM64 LR_Section_OC768_STS768c_and_ RS_STM256			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_SEFS		
LR_Line_OC3_STS3_and_MS_STM1 LR_Line_OC12_STS12_and_MS_STM4 LR_Line_OC48_STS48_and_MS_STM16 LR_Line_OC192_STS192_and_MS_STM64 LR_Section_OC768_STS768c_and_ RS_STM256			
PTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_FC		
	PMP_UAS		
	PMP_PSC	PML_NEAR_END_RX	
	PMP_PSD		
	PMP_PPJ		
	PMP_NPJ		

PTP or CTP	pmParameterName	Location	Interval
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PSC_W		
	PMP_PSD_W		
	PMP_PSC_S		
	PMP_PSD_S		
	PMP_PSC_R		
	PMP_PSD_R		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
LR_Physical_10_Gigabit_ITU LR_Physical_2_5_Gigabit_ITU LR_PHYSICAL_OPTICAL			
PTP	PMP_MAX_LASER_BIAS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ALARM_FAIL		
	PMP_ALARM_DEGRADE_HIGH		
	PMP_ALARM_DEGRADE_LOW		
	PMP_OSC_LOW		
	PMP_OSC_HIGH		
	PMP_OSC_ALARM_FAIL_LOW		
	PMP_OSC_ALARM_FAIL_HIGH		
	PMP_OSC_ALARM_DEGRADE_LOW		
	PMP_OSC_ALARM_DEGRADE_HIGH		
LR_Physical_10_Gigabit_ITU LR_Physical_2_5_Gigabit_ITU LR_Optical_Channel LR_Optical_Multiplex_Section LR_Optical_Transmission_Section LR_PHYSICAL_OPTICAL			
PTP	PMP_MIN_TX_POWER	PML_NEAR_END_RX	15 min, 24 hr
	PMP_AVG_TX_POWER		

PTP or CTP	pmParameterName	Location	Interval
	PMP_MAX_TX_POWER		
	PMP_MIN_RX_POWER		
	PMP_AVG_RX_POWER		
	PMP_MAX_RX_POWER		
	PMP_MIN_LASER_BIAS		
	PMP_AVG_LASER_BIAS		
	PMP_MAX_LASER_BIAS		
	PMP_MIN_TX_OSC_POWER		
	PMP_AVG_TX_OSC_POWER		
	PMP_MAX_TX_OSC_POWER		
LR_Optical_Channel **LR_Optical_Multiplex_Section** **LR_Optical_Transmission_Section**			
PTP	PMP_MIN_PT_POWER PMP_AVG_PT_POWER PMP_MAX_PT_POWER PMP_TX_POWER_HIGH PMP_LASERBIAS_HIGH PMP_TX_POWER_LOW PMP_RX_POWER_LOW PMP_RX_POWER_HIGH PMP_ALARM_TX_POWER_LOW PMP_ALARM_TX_POWER_HIGH PMP_ALARM_LASERBIAS_HIGH PMP_ALARM_RX_POWER_HIGH PMP_ALARM_RX_POWER_LOW PMP_TX_OSNR_HIGH PMP_TX_OSNR_LOW PMP_TX_DGD_HIGH PMP_TX_DGD_LOW PMP_ALARM_TX_CD_HIGH PMP_ALARM_TX_CD_LOW	PML_NEAR_END_RX	15 min, 24 hr
LR_Optical_Channel			
PTP	PMP_MAX_OSNR PMP_AVG_OSNR PMP_MIN_OSNR PMP_MIN_PMD* PMP_MAX_PMD*	PML_BIDIRECTIONAL	15min, 24hr

PTP or CTP	pmParameterName	Location	Interval
	PMP_AVG_PMD* PMP_MIN_DGD* PMP_MAX_DGD* PMP_AVG_DGD* PMP_CD PMP_MIN_SOPMD PMP_MAX_SOPMD PMP_AVG_SOPMD PMP_MIN_PCR PMP_MAX_PCR PMP_AVG_PCR PMP_MIN_PDL PMP_MAX_PDL PMP_AVG_PDL		
LR_T3_and_DS3_45M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_AISSP_PBIT	PML_NEAR_END_RX	
	PMP_CVP_PBIT		
	PMP_ESP_PBIT		
	PMP_SASP_PBIT		
	PMP_SESP_PBIT		
	PMP_UASP_PBIT		
	PMP_CVP_CPBIT	PML_NEAR_END_RX, PML_FAR_END_RX	
	PMP_ESP_CPBIT		
	PMP_SESP_CPBIT		
	PMP_UASP_CPBIT		
	PMP_SASP_CPBIT	PML_FAR_END_RX	
LR_T1_and_DS1_1_5M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min,

PTP or CTP	pmParameterName	Location	Interval
	PMP_ES		24 hr
	PMP_SES		
	PMP_LOSS		
PTP	PMP_CV_P	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES_P		
	PMP_SES_P		
	PMP_SAS_P		
	PMP_UAS_P		
	PMP_AISS_P		
	PMP_CV_P	PML_FAR_END_RX	
	PMP_ES_P		
	PMP_ESA_P		
	PMP_ESB_P		
	PMP_SES_P		
	PMP_SEFS_P		
	PMP_CSS_P		
	PMP_UAS_P		
	PMP_ES		
LR_Optical_Transport_Section			
PTP	PMP_ES	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_SES		
	PMP_UAS		
	PMP_BBE		
	PMP_FC		
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		

PTP or CTP	pmParameterName	Location	Interval
LR_Optical_Transport_Path			
PTP	PMP_ES_P	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_SES_P		
	PMP_UAS_P		
	PMP_BBE_P		
	PMP_FC_P		
	PMP_ESR_P		
	PMP_SESR_P		
	PMP_BBER_P		
LR_Optical_Transport_FEC			
PTP	PMP_BIT_ERRORS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_UNCORRECTABLEWORDS		
	PMP_PREFECBER		
LR_FC_100_1063_M* LR_FC_200_2125M* LR_FICON_1G* LR_FICON_2G*			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RXTOTALPKTS		
	PMP_IFINDISCARDS		
	PMP_IFINERRORS		
	PMP_IFOUTOCTETS		
	PMP_TXTOTALPKTS		
	PMP_IFOUTDISCARDS		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSRXSBLKCRCERRORS		
	PMP_GFPSTATSCSFRAISED		

PTP or CTP	pmParameterName	Location	Interval
	PMP_GFPSTATSROUNDTRIPLATENCY		
	PMP_FCINGRESSRXDISTANCEEXTBUFFERS		
	PMP_FCEGRESSTXDISTANCEEXTBUFFERS		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSTXFRAMESBADCRC		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_FCSTATSLINKRECOVERIES		
	PMP_FCSTATSRXCREDITS		
	PMP_FCSTATSTXCREDITS		
	PMP_FCSTATSZEROTXCREDITS		
	PMP_8B10BINVALIDORDEREDSETS		
	PMP_8B10BSTATSENCODINGDISPERRORS		
	PMP_RXUTILIZATION		
	PMP_TXUTILIZATION		
LR_8b10b_2_5_Gigabit*			
PTP	PMP_DATA_PAYLOAD	PML_NEAR_END_RX	15 min, 24 hr
	PMP_VALID_PACKETS		
	PMP_INVALID_PACKETS		
	PMP_CODE_GROUP_VIOLATIONS		
	PMP_IDLE_ORDERED_SETS		
	PMP_NON_IDLE_ORDERED_SETS		
	PMP_DATA_CODE_GROUPS		
	PMP_RXTOTALPACKETS		
	PMP_IFINERRORS		
	PMP_STATS_ENCODING_DISPERRORS		
	PMP_DATA_ORDERED_SETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_TX8B10BWORDS		
	PMP_INVLD_ORDRD_SET_DISP_ERROR_SUM		
	PMP_RX8B10BWORDS		
LR_10_GFC * LR_4_GFC *			
PTP	PMP_RXOCTETS PMP_TXFRAMES PMP_TXDISCARDS PMP_FCSTATSLINKRECOVERIES PMP_RXPACKETS PMP_FCSTATSRXFRAMESTRUNCATED PMP_RXDISCARDS PMP_FIBRESTATSTXRECVRREADY PMP_FCSTATSTXCREDITS PMP_PCS49RXERRBER PMP_PCS49RXERRDEC PMP_TXPACKETS PMP_MEDIAINDSTATSTXLCVERRORS PMP_TXOCTETS PMP_FCSTATSRXCREDITS PMP_IFOUTERRORS PMP_FCSTATSZEROTXCREDITS PMP_IFOUTOVERSIZEPACKETS PMP_FCSTATSRXFRAMESTOOLONG PMP_FCSTATSLINKRESETS PMP_FIBRESTATSTXFRAMESTOOLONG PMP_FIBRESTATSRXRECVRREADY PMP_FCSTATSRXFRAMESBADCRC PMP_RXFRAMES PMP_FCSTATSTXFRAMESBADCRC PMP_MEDIAINDSTATSTXFRAMESTRUNCATED PMP_MEDIAINDSTATSRXLCVERRORS	PML_NEAR_END_Rx	15min, 24hr
	PMP_TXUTILIZATION PMP_RXUTILIZATION PMP_IFINERRORS	PML_FAR_END_Rx	15min, 24hr
LR_VT1_5_and_TU11_VC11			
CTP	Same as ONS 15327	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
LR_E1_2M (only on DS1_E1_56)			
PTP	PMP_CV	PML_NEAR_END_RX	15 min,

PTP or CTP	pmParameterName	Location	Interval
	PMP_ES		24 hr
	PMP_SES		
	PMP_LOSS		
	PMP_CV_P		
	PMP_ES_P		
	PMP_SES_P		
	PMP_SAS_P		
	PMP_AISS_P		
	PMP_UAS_P		
LR_FC_100_1063M * LR_FC_200_2125M * LR_8_GFC *			
PTP	PMP_MEDIAINDSTATSTXLCVERRORS PMP_RXPACKETS PMP_TXPACKETS PMP_FCSTATSRXFRAMESBADCRC PMP_FCSTATSTXFRAMESBADCRC PMP_MEDIAINDSTATSRXLCVERRORS	PML_FAR_END_Rx	15min, 24hr
LR_ISC3_STP_1G * LR_ISC3_STP_2G *			
PTP	PMP_TXPACKETS PMP_TXOCTETS PMP_RXPACKETS PMP_RXOCTETS PMP_MEDIAINDSTATSTXLCVERRORS PMP_MEDIAINDSTATSRXLCVERRORS	PML_NEAR_END_Rx	15min, 24hr
LR_ISC3_STP_1G * LR_ISC3_STP_2G *			
PTP	PMP_RXUTILIZATION PMP_TXUTILIZATION	PML_FAR_END_Rx	15min, 24hr
LR_ENCRYPTION			
PTP	PMP_SEQMISM	PML_NEAR_END_Rx	15min, 24hr

** TCA settings are not supported for data layers.*

✎ PMP_ALARM_TX_POWER_LOW, PMP_ALARM_TX_POWER_HIGH, PMP_ALARM_LASERBIAS_HIGH, PMP_ALARM_RX_POWER_HIGH, PMP_ALARM_RX_POWER_LOW are alarm threshold parameters. These alarm threshold parameters are retrieved in the same query associated to the TCA of 15-minute granularity of

parameters PMP_TX_POWER_HIGH, PMP_LASERBIAS_HIGH, PMP_TX_POWER_LOW, PMP_RX_POWER_LOW, and PMP_RX_POWER_HIGH.

- A set of Alarm and TCA parameters cannot be changed in the same call, and must be done as two separate instances of setTCATPParameter invocation, always using 15-minute granularity.
- For CPT 200 SDH and CPT 600 SDH, a subset of ONS 15454 SDH parameters is supported. The supported parameters match the parameters associated to the port equipment supported by CPT.
- PMP_MIN_PMD,PMP_MAX_PMD, PMP_AVG_PMD parameters are retrieved for ONS15454 NE ver. < 9.8.
- PMP_MIN_DGD,PMP_MAX_DGD, PMP_AVG_DGD parameters are retrieved for ONS15454 NE ver. >= 9.8.

Table 3-9: ONS 15454 SDH and ONS NCS2K SDH TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_Fast_Ethernet **LR_Gigabit_Ethernet**			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_IFINUCASTPKTS		
	PMP_IFINMULTICASTPKTS		
	PMP_IFINBROADCASTPKTS		
	PMP_IFINDISCARDS		
	PMP_IFOUTDISCARDS		
	PMP_IFOUTMULTICASTPKTS		
	PMP_IFOUTBROADCASTPKTS		
	PMP_ETHERSTATSBROADCASTPKTS		
	PMP_ETHERSTATSMULTICASTPKTS		
	PMP_PORTDROPCOUNTS		
	PMP_IFINERRORS		
	PMP_IFOUTOCTETS		
	PMP_IFOUTUCASTPKTS		
	PMP_TXPACKETS		
	PMP_RXPACKETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_RXPAUSEFRAMES		
	PMP_TXPAUSEFRAMES		
	PMP_RXPKTSDROPPEDINTERNALCONGESTION		
	PMP_TXPKTSDROPPEDINTERNALCONGESTION		
	PMP_HDLCPKTDROPS		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSTXFRAMESBADCRC		
	PMP_RXCONTROLFRAMES		
	PMP_RXUNKNOWNOPCODEFRAMES		
	PMP_IFINERRORBYTEPKTS		
	PMP_IFINFRAMINGERRORPKTS		
	PMP_IFINJUNKINTERPKTS		
PTP	PMP_MEDIAINDSTATSOVERSIZEDROPPED	PML_NEAR_END_RX	15 min, 24 hr
	PMP_DOT3STATSALIGNMENTERRORS		
	PMP_DOT3STATSFCSERRORS		
	PMP_DOT3STATSSINGLECOLLISIONFRAMES		
	PMP_DOT3STATSCARRIERSENSEERRORS		
	PMP_DOT3STATSSQETESTERRORS		
	PMP_DOT3STATSMULTIPLECOLLISIONFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_DOT3STATSDEFERREDTRANSMISSIONS		
	PMP_DOT3STATSLATECOLLISIONS		
	PMP_DOT3STATSEXCESSIVECOLLISIONS		
	PMP_DOT3STATSFRAMETOOLONG		
	PMP_ETHERSTATSUNDERSIZEPKTS		
	PMP_ETHERSTATSFRAGMENTS		
	PMP_ETHERSTATSPKTS64OCTETS		
	PMP_ETHERSTATSPKTS65TO127OCTETS		
	PMP_ETHERSTATSPKTS128TO255OCTETS		
	PMP_ETHERSTATSPKTS256TO511OCTETS		
	PMP_ETHERSTATSPKTS512TO1023OCTETS		
	PMP_ETHERSTATSPKTS1024TO1518OCTETS		
	PMP_ETHERSTATSOVERSIZEPKTS		
	PMP_ETHERSTATSJABBERS		
	PMP_ETHERSTATSOCTETS		
	PMP_ETHERSTATSCOLLISIONS		
	PMP_ETHERSTATSCOLLISIONFRAMES		
	PMP_ETHERSTATSCRCALIGNERRORS		
	PMP_RXETHERUTILIZATIONSTATS		
	PMP_TXETHERUTILIZATIONSTATS		
	PMP_RXGIANTS		
	PMP_RXGMACDROPCOUNTS		
	PMP_RXRUNTS		
	PMP_RXTHRESHOLDOVERSIZES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_RXTOTALERRORS		
PTP	PMP_TXCOLLISIONS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_TXGIANTS		
	PMP_DOT3STATSCONTROLINUNKNOWNOPCODES		
	PMP_DOT3STATSINPAUSEFRAMES		
	PMP_DOT3STATSOUTPAUSEFRAMES		
	PMP_ETHERSTATSPKTS1519TO1522OCTETS		
LR_POS_100M **LR_POS_Gigabit**			
PTP	PMP_RX_INPUT_DROP_PACKETS		15 min, 24 hr
	PML_NEAR_END_RX		
	PMP_RX_INPUT_ABORT_PACKETS		
	PMP_TX_PRE_HDLC_BYTES		
	PMP_TX_POST_HDLC_BYTES		
	PMP_TX_PACKETS		
	PMP_PORT_DROP_COUNTS		
	PMP_RX_PRE_HDLC_BYTES		
	PMP_RX_POST_HDLC_BYTES		
	PMP_RX_PACKETS		
	PMP_RX_NORMAL_PACKETS		
	PMP_RX_SHORTS		
	PMP_RX_RUNTS		
	PMP_RX_LONGS		
	PMP_RX_TOTAL_ERRORS		
	PMP_RX_CRC_ERRORS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_ETHERSTATSDROPEVENTS		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_MEDIAINDSTATSRXSHORTPKTS		
	PMP_HDLCINOCTETS		
	PMP_HDLCRXABORTS		
	PMP_HDLCOUTOCTETS		
	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_IFOUTOCTETS		
	PMP_IFINDISCARDS		
	PMP_RXPKTSDROPINTERNALCONGESTION		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSCSFRAISED		
	PMP_GFPSTATSLFDRAISED		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
LR_COS_100M			

PTP or CTP	pmParameterName	Location	Interval
LR_COS_Gigabit			
PTP	PMP_INTERFACE_NAME	PML_NEAR_END_RX	15 min, 24 hr
	PMP_INTERFACE_DIRECTION		
	PMP_CLASS_OF_SERVICE_LEVEL		
	PMP_POST_POLICY_PACKETS		
	PMP_POST_POLICY_BYTES		
	PMP_DROP_PACKETS		
	PMP_DROP_BYTES		
LR_T1_and_DS1_1_5M ✎ VT1.5 data is retrieved using this layer rate. STS data retrieval is not supported.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
LR_GFP_Multirate			
PTP	PMP_GFPSTATSCSFRAISED	PML_NEAR_END_RX	15 min, 24 hr
	PMP_GFPSTATSLFDRAISED		
	PMP_GFPSTATSROUNDTRIPLATENCY		
	PMP_GFPSTATSRXCIDINVALID		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSRXMBITERRORS		
	PMP_GFPSTATSRXOCTETS		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXSBLKCRCERRORS		
	PMP_GFPSTATSRXTYPEINVALID		

PTP or CTP	pmParameterName	Location	Interval
	PMP_GFPSTATSTXFRAME		
	PMP_GFPSTATSTXOCTETS		
	PMP_GFPSTATSRXDISTANCEEXTBUFFERS		
	PMP_GFPSTATSTXDISTANCEEXTBUFFERS		
LR_E1_2M VT2 data is retrieved using this layer rate.			
CTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
LR_E1_2M E1 data is retrieved using this layer rate.			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
	PMP_CV_P		
	PMP_BBE_P		
	PMP_ES_P		
	PMP_SES_P		
	PMP_UAS_P		
	PMP_SAS_P		
	PMP_FC_P		
	PMP_AISS_P		
LR_OCH_ISC_PEERCOMPAT			
PTP	PMP_RXOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RXPACKETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_TXOCTETS		
	PMP_TXPACKETS		
	PMP_INVLD_ORDRD_SET_DISP_ERROR_SUM		
LR_ESCON			
PTP	PMP_IFINERRORS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_TXPACKETS		
	PMP_IFOUTDISCARDS		
	PMP_IFINDISCARDS		
	PMP_IFOUTOCTETS PMP_RXPACKETS		
	PMP_IFINOCTETS		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_TXETHERUTILIZATIONSTATS		
	MP_RXETHERUTILIZATIONSTATS		
	PMP_RXOCTETS		
	PMP_RXPACKETS		
	PMP_TXOCTETS		
	PMP_TXPACKETS		
LR_LEX			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_UAS		
	PMP_SES		
	PMP_ES		
	PMP_FC		
	PMP_NPJC_PGEN		
	PMP_NPJC_PDET		

PTP or CTP	pmParameterName	Location	Interval
	PMP_PJCS_PDET		
	PMP_PJCS_PGEN		
	PMP_PJCDIFF		
LR_RPR_IEEE_802_17			
PTP	PMP_SPANOUTMCASTCLASSAOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_SPANOUTCTRLFRAMES		
	PMP_SPANOUTOAMECHOFRAMES		
	PMP_SPANOUTOAMFLUSHFRAMES		
	PMP_SPANOUTOAMORGFRAMES		
	PMP_SPANOUTTOPOATDFRAMES		
	PMP_SPANOUTTOPOCHKSUMFRAMES		
	PMP_SPANOUTTOPOTPFRAMES		
	PMP_CLIENTINUCASTCLASSCFRAMES		
	PMP_CLIENTINUCASTCLASSCOCTETS		
	PMP_CLIENTINMCASTCLASSCFRAMES		
	PMP_CLIENTINMCASTCLASSCOCTETS		
	PMP_CLIENTINUCASTCLASSBEIRFRAMES		
	PMP_CLIENTINUCASTCLASSBEIROCTETS		
	PMP_CLIENTINMCASTCLASSBEIRFRAMES		
	PMP_CLIENTINMCASTCLASSBEIROCTETS		
	PMP_CLIENTINUCASTCLASSBCIRFRAMES		
	PMP_CLIENTINUCASTCLASSBCIROCTETS		
	PMP_CLIENTINMCASTCLASSBCIRFRAMES		
	PMP_CLIENTINMCASTCLASSBCIROCTETS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_CLIENTINUCASTCLASSAFRAMES		
	PMP_CLIENTINUCASTCLASSAOCTETS		
	PMP_CLIENTINMCASTCLASSAFRAMES		
	PMP_CLIENTINMCASTCLASSAOCTETS		
	PMP_CLIENTINBCASTFRAMES		
	PMP_CLIENTOUTUCASTCLASSCFRAMES		
	PMP_CLIENTOUTUCASTCLASSCOCTETS		
	PMP_CLIENTOUTMCASTCLASSCFRAMES		
	PMP_CLIENTOUTMCASTCLASSCOCTETS		
	PMP_CLIENTOUTUCASTCLASSBEIRFRAMES		
	PMP_CLIENTOUTUCASTCLASSBEIROCTETS		
	PMP_CLIENTOUTMCASTCLASSBEIRFRAMES		
	PMP_CLIENTOUTMCASTCLASSBEIROCTETS		
	PMP_CLIENTOUTUCASTCLASSBCIRFRAMES		
	PMP_CLIENTOUTUCASTCLASSBCIROCTETS		
	PMP_CLIENTOUTMCASTCLASSBCIRFRAMES		
	PMP_CLIENTOUTMCASTCLASSBCIROCTETS		
	PMP_CLIENTOUTUCASTCLASSAFRAMES		
	PMP_CLIENTOUTUCASTCLASSAOCTETS		
	PMP_CLIENTOUTMCASTCLASSAFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_CLIENTOUTMCASTCLASSAOCTETS		
	PMP_CLIENTOUTBCASTFRAMES		
	PMP_ERRORBADPARITYFRAMES		
	PMP_ERRORBADHECFRAMES		
	PMP_ERRORTTLEXPFRAMES PMP_ERRORTOOLONGFRAMES		
	PMP_ERRORTOOSHORTFRAMES		
	PMP_ERRORBADFCSFRAMES		
	PMP_ERRORSELFSRCUCASTFRAMES		
	PMP_ERRORPMDABORTFRAMES		
	PMP_ERRORBADADDRFRAMES		
	PMP_ERRORCONTAINEDFRAMES		
	PMP_ERRORSCFFERRORS		
	PMP_ERROROVERSIZEFRAMES		
	PMP_PORTCOUNTERERROR		
	PMP_SPANINUCASTCLASSCFRAMES		
	PMP_SPANINUCASTCLASSCOCTETS		
	PMP_SPANINMCASTCLASSCFRAMES		
	PMP_SPANINMCASTCLASSCOCTETS		
	PMP_SPANINUCASTCLASSBEIRFRAMES		
	PMP_SPANINUCASTCLASSBEIROCTETS		
	PMP_SPANINMCASTCLASSBEIRFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SPANINMCAST CLASSBEIROCTETS		
	PMP_SPANINUCASTC LASSBCIRFRAMES		
	PMP_SPANINUCASTC LASSBCIROCTETS		
	PMP_SPANINMCAST CLASSBCIRFRAMES		
	PMP_SPANINMCAST CLASSBCIROCTETS		
	PMP_SPANINUCASTC LASSAFRAMES		
	PMP_SPANINUCASTC LASSAOCTETS		
	PMP_SPANINMCAST CLASSAFRAMES		
	PMP_SPANINMCAST CLASSAOCTETS		
	PMP_SPANINCTRLFR AMES		
	PMP_SPANINOAMEC HOFRAMES		
	PMP_SPANINOAMFL USHFRAMES		
	PMP_SPANINOAMOR GFRAMES		
	PMP_SPANINTOPOAT DFRAMES		
	PMP_SPANINTOPOCH KSUMFRAMES		
	PMP_SPANINTOPOTP FRAMES		
	PMP_SPANOUTUCAS TCLASSCFRAMES		
	PMP_SPANOUTUCAS TCLASSCOCTETS		
	PMP_SPANOUTMCAS TCLASSCFRAMES		
	PMP_SPANOUTMCAS TCLASSCOCTETS		
	PMP_SPANOUTUCAS TCLASSBEIRFRAMES		
	PMP_SPANOUTUCAS TCLASSBEIROCTETS		
	PMP_SPANOUTMCAS TCLASSBEIRFRAMES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_SPANOUTMCASTCLASSBEIROCTETS		
	PMP_SPANOUTUCASTCLASSBCIRFRAMES		
	PMP_SPANOUTUCASTCLASSBCIROCTETS		
	PMP_SPANOUTMCASTCLASSBCIRFRAMES		
	PMP_SPANOUTMCASTCLASSBCIROCTETS		
	PMP_SPANOUTUCASTCLASSAFRAMES		
	PMP_SPANOUTUCASTCLASSAOCTETS		
	PMP_SPANOUTMCASTCLASSAFRAMES		
LR_T3_and_DS3_45M			
PTP	PMP_CV_P	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES_P		
	PMP_SES_P		
	PMP_SAS_P		
	PMP_AISS_P		
	PMP_UAS_P		
	PMP_CV_CP		
	PMP_ES_CP		
	PMP_SES_CP		
	PMP_SAS_CP		
	PMP_UAS_CP		
	PMP_CV		
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
LR_E1_2M LR_E3_34M LR_T3_and_DS3_45M			
CTP	PMP_EB	PML_NEAR_END_R	15 min, 24 hr

PTP or CTP	pmParameterName	Location	Interval
	PMP_BBE	X, PML_FAR_END_RX	
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		
LR_STS3c_and_AU4_VC4 **LR_STS12c_and_VC4_4c** **LR_STS6c_and_VC4_2c** **LR_STS9c_and_VC4_3c** **LR_STS24c_and_VC4_8c** **LR_STS48c_and_VC4_16c** **LR_STS192c_and_VC4_64c**			
CTP	PMP_EB PML_NEAR_END_RX 15 min, 24 hr PMP_BBE PMP_SES PMP_ES PMP_UAS PMP_ESR PMP_SESR PMP_BBER PMP_PPJ PMP_NPJ PMP_PPJC_PGEN PMP_NPJC_PGEN PMP_PJ_DIFF PMP_PJP_SEC PMP_PJN_SEC	PML_NEAR_END_RX	15 min, 24 hr
LR_Section_OC3_STS3c_and_RS_STM1 **LR_Section_OC12_STS12c_and_RS_STM4** **LR_Section_OC48_STS48c_and_RS_STM16** **LR_Section_OC192_STS192c_and_RS_STM64** **LR_Section_OC768_STS768c_and_ RS_STM256 (STM1E card does not support PMP_OFS)**			
PTP	PMP_EB	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_BBE		

PTP or CTP	pmParameterName	Location	Interval
	PMP_UAS		
	PMP_OFS		
	PMP_UAS	PML_FAR_END_Rx	
LR_Line_OC3_STS3_and_MS_STM1 LR_Line_OC12_STS12_and_MS_STM4 LR_Line_OC48_STS48_and_MS_STM16 LR_Line_OC192_STS192_and_MS_STM64 LR_Line_OC768_STS768c_and_ RS_STM256			
PTP	PMP_EB	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_BBE		
	PMP_UAS		
	PMP_FC		
	PMP_PSC	PML_NEAR_END_RX	15 min, 24 hr
	PMP_PSD		
	PMP_PPJ		
	PMP_NPJ		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PSC_W		
	PMP_PSD_W		
	PMP_PSC_S		
	PMP_PSD_S		
	PMP_PSC_R		
	PMP_PSD_R		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
	PMP_NPJC_PGEN		
	PMP_PPJC_PGEN		

PTP or CTP	pmParameterName	Location	Interval
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		
LR_Physical_10_Gigabit_ITU **LR_Physical_2_5_Gigabit_ITU** **LR_PHYSICAL_OPTICAL**			
PTP	PMP_MAX_LASER_BIAS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ALARM_FAIL		
	PMP_ALARM_DEGRADE_HIGH		
	PMP_ALARM_DEGRADE_LOW		
	PMP_OSC_LOW		
	PMP_OSC_HIGH		
	PMP_OSC_ALARM_FAIL_LOW		
	PMP_OSC_ALARM_FAIL_HIGH		
	PMP_OSC_ALARM_DEGRADE_LOW		
	PMP_OSC_ALARM_DEGRADE_HIGH		
LR_Physical_10_Gigabit_ITU **LR_Physical_2_5_Gigabit_ITU** **LR_Optical_Channel** **LR_PHYSICAL_OPTICAL** **LR_Optical_Multiplex_Section** **LR_Optical_Transmission_Section**			
PTP	PMP_MIN_TX_POWER	PML_NEAR_END_RX	15 min, 24 hr
	PMP_AVG_TX_POWER		
	PMP_MAX_TX_POWER		
	PMP_MIN_RX_POWER		
	PMP_AVG_RX_POWER		
	PMP_MAX_RX_POWER		
	PMP_MIN_LASER_BIAS		
	PMP_AVG_LASER_BIAS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_MAX_LASER_BIAS		
	PMP_MIN_TX_OSC_POWER		
	PMP_AVG_TX_OSC_POWER		
	PMP_MAX_TX_OSC_POWER		
LR_Optical_Channel **LR_Optical_Multiplex_Section** **LR_Optical_Transmission_Section**			
PTP	PMP_MIN_PT_POWER PMP_AVG_PT_POWER PMP_MAX_PT_POWER PMP_TX_POWER_HIGH PMP_LASERBIAS_HIGH PMP_TX_POWER_LOW PMP_RX_POWER_LOW PMP_RX_POWER_HIGH PMP_ALARM_TX_POWER_LOW PMP_ALARM_TX_POWER_HIGH PMP_ALARM_LASERBIAS_HIGH PMP_ALARM_RX_POWER_HIGH PMP_ALARM_RX_POWER_LOW PMP_TX_OSNR_HIGH PMP_TX_OSNR_LOW PMP_TX_DGD_HIGH PMP_TX_DGD_LOW PMP_ALARM_TX_CD_HIGH PMP_ALARM_TX_CD_LOW	PML_NEAR_END_RX	15 min, 24 hr
LR_Optical_Channel			
PTP	PMP_MAX_OSNR PMP_AVG_OSNR PMP_MIN_OSNR PMP_MIN_PMD* PMP_MAX_PMD*	PML_BIDIRECTIONAL	15 min, 24 hr

PTP or CTP	pmParameterName	Location	Interval
	PMP_AVG_PMD* PMP_MIN_DGD* PMP_MAX_DGD* PMP_AVG_DGD* PMP_CD PMP_MIN_SOPMD PMP_MAX_SOPMD PMP_AVG_SOPMD PMP_MIN_PCR PMP_MAX_PCR PMP_AVG_PCR PMP_MIN_PDL PMP_MAX_PDL PMP_AVG_PDL		
LR_LOW_ORDER_TU_VC3 LR_VT2_AND_TU12_VC12			
CTP	PMP_EB PMP_BBE PMP_SES PMP_ES PMP_UAS PMP_ESR PMP_SESR	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_BBER		
LR_E4_140M			
PTP	PMP_CV PMP_ES PMP_SES PMP_UAS PMP_BBE PMP_ESR PMP_SESR	PML_NEAR_END_RX,	15 min, 24 hr
	PMP_BBER		
LR_E3_34M			
PTP	PMP_CV PMP_ES PMP_SES	PML_NEAR_END_RX	15 min, 24 hr
	PMP_LOSS		
LR_E1_2M			
PTP	PMP_CV PMP_ES PMP_SES	PML_NEAR_END_RX	15 min, 24 hr
	PMP_LOSS		
	PMP_EB_P PMP_BBE_P PMP_ES_P PMP_SES_P PMP_UAS_P PMP_ESR_P PMP_SESR_P	PML_NEAR_END_RX, PML_FAR_END_RX, PML_NEAR_END_TX, PML_FAR_END_TX	
	PMP_BBER_P		

PTP or CTP	pmParameterName	Location	Interval
	PMP_AISS	PML_NEAR_END_RX, PML_NEAR_END_TX	
OTN SECTION			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
OTN PATH			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
OTN FEC			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
SAN			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
8B10B			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
LR_ENCRYPTION			
PTP	PMP_SEQMISM	PML_NEAR_END_Rx	15min, 24hr

- PMP_ALARM_TX_POWER_LOW, PMP_ALARM_TX_POWER_HIGH, PMP_ALARM_LASERBIAS_HIGH, PMP_ALARM_RX_POWER_HIGH, PMP_ALARM_RX_POWER_LOW are alarm threshold parameters. These alarm threshold parameters are retrieved in the same query associated to the TCA of 15-minute granularity of parameters PMP_TX_POWER_HIGH, PMP_LASERBIAS_HIGH, PMP_TX_POWER_LOW, PMP_RX_POWER_LOW, and PMP_RX_POWER_HIGH.
- A set of Alarm and TCA parameters cannot be changed in the same call, and must be done as two separate instances of setTCATPParameter invocation, always using 15-minute granularity.
- PMP_MIN_PMD,PMP_MAX_PMD, PMP_AVG_PMD parameters are retrieved for ONS15454 NE ver. < 9.8
- PMP_MIN_DGD,PMP_MAX_DGD, PMP_AVG_DGD parameters are retrieved for ONS15454 NE ver. >= 9.8

Table 3-10: ONS 15454 SDH and ONS NCS2k SDH TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_T3_and_DS3_45M			
PTP	PMP_CV_P PMP_ES_P	PML_NEAR_END_RX	15 min, 24 hr

PTP or CTP	pmParameterName	Location	Interval
	PMP_SES_P		
	PMP_SAS_P		
	PMP_AISS_P		
	PMP_UAS_P		
	PMP_CV_CP		
	PMP_ES_CP		
	PMP_SES_CP		
	PMP_SAS_CP		
	PMP_UAS_CP		
	PMP_CV		
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
LR_E1_2M **LR_E3_34M** **LR_T3_and_DS3_45M**			
	PMP_EB		
	PMP_BBE	PML_NEAR_END_RX, PML_FAR_END_RX	
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_ESR		
	PMP_SESR		
CTP	PMP_BBER		15 min, 24 hr
LR_STS3c_and_AU4_VC4 **LR_STS12c_and_VC4_4c** **LR_STS6c_and_VC4_2c** **LR_STS9c_and_VC4_3c** **LR_STS24c_and_VC4_8c** **LR_STS48c_and_VC4_16c** **LR_STS192c_and_VC4_64c**			
CTP	PMP_EB	PML_NEAR_END_RX	15 min, 24 hr
	PMP_BBE		
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		

PTP or CTP	pmParameterName	Location	Interval
	PMP_PPJ		
	PMP_NPJ		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		
	PMP_PJ_DIFF		
	PMP_PJP_SEC		
	PMP_PJN_SEC		
LR_Section_OC3_STS3c_and_RS_STM1 LR_Section_OC12_STS12c_and_RS_STM4 LR_Section_OC48_STS48c_and_RS_STM16 LR_Section_OC192_STS192c_and_RS_STM64 LR_Section_OC768_STS768c_and_ RS_STM256 (STM1E card does not support PMP_OFS)			
PTP	PMP_EB	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_BBE		
	PMP_UAS		
	PMP_OFS		
	PMP_BBER		
	PMP_ESR		
	PMP_SESR		
	PMP_UAS	PML_FAR_END_Rx	
LR_Line_OC3_STS3_and_MS_STM1 LR_Line_OC12_STS12_and_MS_STM4 LR_Line_OC48_STS48_and_MS_STM16 LR_Line_OC192_STS192_and_MS_STM64 LR_Line_OC768_STS768c_and_ RS_STM256			
PTP	PMP_EB	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_BBE		
	PMP_UAS		
	PMP_FC		
	PMP_PSC		
	PMP_PSD		
	PMP_PPJ		
	PMP_NPJ		
	PMP_PPJC_PGEN		

PTP or CTP	pmParameterName	Location	Interval
	PMP_NPJC_PGEN		
	PMP_PSC_W		
	PMP_PSD_W		
	PMP_PSC_S		
	PMP_PSD_S		
	PMP_PSC_R		
	PMP_PSD_R		
	PMP_PPJC_PDET		
	PMP_NPJC_PDET		
	PMP_NPJC_PGEN		
	PMP_PPJC_PGEN		
	PMP_ESR	PML_FAR_END_RX	
	PMP_SESR		
	PMP_BBER		
LR_Physical_10_Gigabit_ITU **LR_Physical_2_5_Gigabit_ITU** **LR_PHYSICAL_OPTICAL**			
PTP	PMP_MAX_LASER_BIAS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ALARM_FAIL		
	PMP_ALARM_DEGRADE_HIGH		
	PMP_ALARM_DEGRADE_LOW		
	PMP_OSC_LOW		
	PMP_OSC_HIGH		
	PMP_OSC_ALARM_FAIL_LOW		
	PMP_OSC_ALARM_FAIL_HIGH		
	PMP_OSC_ALARM_DEGRADE_LOW		
	PMP_OSC_ALARM_DEGRADE_HIGH		
LR_Physical_10_Gigabit_ITU **LR_Physical_2_5_Gigabit_ITU** **LR_Optical_Channel** **LR_PHYSICAL_OPTICAL** **LR_Optical_Multiplex_Section** **LR_Optical_Transmission_Section**			
PTP	PMP_MIN_TX_POWER	PML_NEAR_END_RX	15 min, 24 hr
	PMP_AVG_TX_POWER		
	PMP_MAX_TX_POWER		
	PMP_MIN_RX_POWER		
	PMP_AVG_RX_POWER		
	PMP_MAX_RX_POWER		

PTP or CTP	pmParameterName	Location	Interval
	PMP_MIN_LASER_BIAS		
	PMP_AVG_LASER_BIAS		
	PMP_MAX_LASER_BIAS		
LR_Optical_Channel **LR_Optical_Multiplex_Section** **LR_Optical_Transmission_Section**			
PTP	PMP_MIN_PT_POWER	PML_NEAR_END_RX	15 min, 24 hr
	PMP_AVG_PT_POWER		
	PMP_MAX_PT_POWER		
LR_LOW_ORDER_TU_VC3 **LR_VT2_AND_TU12_VC12**			
CTP	PMP_EB	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_BBE		
	PMP_SES		
	PMP_ES		
	PMP_UAS		
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		
LR_E4_140M			
PTP	PMP_CV	PML_NEAR_END_RX,	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
	PMP_BBE		
	PMP_ESR		
	PMP_SESR		
	PMP_BBER		
LR_E3_34M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_LOSS		
LR_E1_2M			
PTP	PMP_CV	PML_NEAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		

PTP or CTP	pmParameterName	Location	Interval
	PMP_LOSS		
	PMP_EB_P	PML_NEAR_END_RX, PML_FAR_END_RX, PML_NEAR_END_TX, PML_FAR_END_TX	
	PMP_BBE_P		
	PMP_ES_P		
	PMP_SES_P		
	PMP_UAS_P		
	PMP_ESR_P		
	PMP_SESR_P		
	PMP_BBER_P		
	PMP_AISS	PML_NEAR_END_RX, PML_NEAR_END_TX	
OTN SECTION			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
OTN PATH			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
OTN FEC			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
SAN			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr
8B10B			
PTP	Same as SONET	PML_NEAR_END_RX	15 min, 24 hr

Table 3-11: ONS 15600 SONET TCA TP ParameterName Values

PTP or CTP	pmParameterName	Location	Interval
LR_Gigabit_Ethernet			
PTP	PMP_IFINOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_IFINUCASTPKTS		
	PMP_IFINMULTICASTPKTS		
	PMP_IFINBROADCASTPKTS		
	PMP_ETHERSTATSOVERSIZEPKTS		
	PMP_DOT3STATSFCSERRORS		
	PMP_ETHERSTATSUNDERSIZEPKTS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_ETHERSTATSJABBERS		
	PMP_DOT3STATSALIGNMENTERRORS		
	PMP_IFOUTOCTETS		
	PMP_IFOUTUCASTPKTS		
	PMP_IFOUTMULTICASTPKTS		
	PMP_IFOUTBROADCASTPKTS		
	PMP_ETHERSTATSDROPEVENTS		
	PMP_IFINDISCARDS		
	PMP_ETHERSTATSOCTETS		
	PMP_ETHERSTATSPKTS		
	PMP_ETHERSTATSBROADCASTPKTS		
	PMP_ETHERSTATSMULTICASTPKTS		
	PMP_ETHERSTATSMULTICASTPKTS		
	PMP_ETHERSTATSFRAGMENTS		
	PMP_ETHERSTATSPKTS64OCTETS		
	PMP_ETHERSTATSPKTS65TO127OCTETS		
	PMP_ETHERSTATSPKTS128TO255OCTETS		
	PMP_ETHERSTATSPKTS256TO511OCTETS		
	PMP_ETHERSTATSPKTS512TO1023OCTETS		
	PMP_ETHERSTATSPKTS1024TO1518OCTETS		
	PMP_IFOUTDISCARDS		
	PMP_IFINERRORS		
	PMP_IFOUTERRORS		
	PMP_DOT3STATSINTERNALMACTXERRORS		
	PMP_DOT3STATSFRAMETOOLONG		
	PMP_DOT3STATSINTERNALMACRXERRORS		
	PMP_DOT3STATSSYMBOLERRORS		
	PMP_RX_PAUSE_FRAMES		
	PMP_TX_PAUSE_FRAMES		
	PMP_RXUTILIZATION		
	PMP_TXUTILIZATION		
LR_POS_Gigabit			
PTP	PMP_GFPSTATSRXOCTETS	PML_NEAR_END_RX	15 min, 24 hr
	PMP_RX_PACKETS		
	PMP_GFPSTATSRXCRCERRORS		
	PMP_GFPSTATSTXOCTETS		
	PMP_TX_PACKETS		
	PMP_GFPSTATSRXSBITERRORS		
	PMP_GFPSTATSRXMBITERRORS		

PTP or CTP	pmParameterName	Location	Interval
	PMP_GFPSTATSRXTYPEINVALID		
	PMP_HDLCRXABORTS		
	PMP_MEDIAINDSTATSRXSHORTPKTS		
	PMP_MEDIAINDSTATSRXFRAMESTOOLONG		
	PMP_RXPKTSDROPPEDINERNALCONGESTION		
	PMP_MEDIAINDSTATSRXFRAMESTRUNCATED		
	PMP_MEDIAINDSTATSRXFRAMESBADCRC		
	PMP_GFPSTATSRXFRAME		
	PMP_GFPSTATSTXFRAME		
	PMP_HDLCINOCTETS		
	PMP_HDLCOUTOCTETS		
	PMP_RXUTILIZATION		
LR_PHYSICAL_OPTICAL			
PTP	PMP_LBC_HIGH	PML_NEAR_END_RX	15 min, 24 hr
	PMP_LBC_LOW		
	PMP_OPT_HIGH		
	PMP_OPR_HIGH		
	PMP_OPR_LOW		
LR_Section_OC48_STS48c_and_RS_STM16 **LR_Section_OC192_STS192_and_RS_STM64**			
PTP	PMP_ES	PML_NEAR_END_RX	15 min, 24 hr
	PMP_SES		
	PMP_SEFS		
	PM_CV		
LR_Line_OC48_STS48_and_MS_STM16 **LR_Line_OC192_STS192_and_MS_STM64**			
PTP	PMP_CV	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hr
	PMP_ES		
	PMP_SES		
	PMP_UAS		
	PMP_FC		
	PMP_PSC	PML_NEAR_END_RX	
	PMP_PSD		
	PMP_PSC_W		
	PMP_PSD_W		
	PMP_PSC_S		
	PMP_PSD_S		
	PMP_PSC_R		
	PMP_PSD_R		

PTP or CTP	pmParameterName	Location	Interval
LR_STS1_and_AU3_High_Order_VC3 **LR_STS3c_and_AU4_VC4** **LR_STS12c_and_VC4_4c** **LR_STS24c_and_VC4_8c** **LR_STS48c_and_VC4_16c** **LR_STS192c_and_VC4_64c**			
CTP	PMP_CV_P	PML_NEAR_END_RX, PML_FAR_END_RX	15 min, 24 hour
	PMP_ES_P		
	PMP_SES_P		
	PMP_UAS_P		
	PMP_FC_P		
	PMP_PPJC_PDET	PML_NEAR_END_RX	
	PMP_NPJC_PDET		
	PMP_PPJC_PGEN		
	PMP_NPJC_PGEN		

- PMLocation_T pmLocation
  - Prime Optical supports PML_NEAR_END_RX and PML_FAR_END_RX.
- PMThresholdType_T thresholdType
  - Prime Optical supports only TWM_HIGH.
- boolean triggerFlag
  - Not supported.
- float value
  - Supported. The NE accepts only integer values. Therefore, Prime Optical rounds off floating point values to the biggest lower integer. The maximum value supported is 9999999.
- string unit
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised if tpName does not reference a terminationPoint object or at least one field of tcaParameters is not valid.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.13.4 getTCATPParameter

Synopsis

```
void getTCATPParameter(
in globaldefs::NamingAttributes_T tpName,
in transmissionParameters::LayerRate_T layerRate,
```

```
in Granularity_T granularity,
out TCAParameters_T tcaParameter)
raises(globaldefs::ProcessingFailureException);
```

Description
The purpose of this operation is to retrieve the values of 15-minute or 24-hour TCA/alarm thresholds on a TP/layer rate measurement point. The operation is best-effort. Results are returned in the out parameter of this operation. The operation can be applied to either a PTP or CTP. The NMS requests the TCA parameters for the particular TP and granularity specified.
A set of TCA/alarm thresholds is supported for Optical, OTN, SONET and SDH layer rates.

Parameters

Name	Type	Input/Output	Description
tpName	globaldefs::NamingAttributes_T	In	Identification of the TP for which to retrieve the values of the TCA parameters.
layerRate	transmissionParameters::LayerRate_T	In	Layer rate for which to retrieve the values of the TCA parameters.
granularity	Granularity_T	In	Granularity for which to retrieve the TCA parameters.
tcaParameter	TCAParameters_T	Out	Result of the operation.

For a list of supported layer rates and parameter values, see 3.11.3 getAllVLANs.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when tpName does not reference a terminationPoint
object or layerRateList contains undefined rates or granularity contains an
undefined value.
EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not
exist.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance
TMF-defined.

3.13.5 getCTMHistoryPMData

Synopsis

```
public void getCTMHistoryPMData(
in nmsSession::NmsSession_I    client,
in PMTPSelectList_T            pmTPSelectList,
in PMParameterNameList_T       pmParameters,
in globaldefs::Time_T          startTime,
in globaldefs::Time_T          endTime)
raises (globaldefs::ProcessingFailureException);
```

Description
This interface instructs the EMS to store historical PM data in a file, and to notify the NMS when the request is complete. Within the request a list of TP/layerRate measurement points and a time frame are

specified. For each measurement point, the granularity (15-minute, 24-hour) and location (PML_NEAR_END_RX, PML_FAR_END_RX) can be specified. A filtered set (scoped by the input parameter pmParameters) of PM parameters collected for a particular TP/layer rate measurement point for the specified granularity, location, and time window is made available.
Measurement intervals and the time frame are considered as closed intervals (that is, [startTime <= t endTime]).
Performance monitoring data on multiple TPs of multiple MEs is transferred in one data file. The PM file format is defined by the TMF. Prime Optical generates a unique filename and saves the file in the CTMS-home/pmData/username/ directory. This file is retained for six hours after Prime Optical finishes writing the requested PM data. After six hours, the file is deleted. When GateWay/CORBA restarts, it retrieves a list of all existing PM files and deletes them after six hours.
This is an asynchronous operation. Once Prime Optical receives the request, it validates it and the call returns. In the background, Prime Optical begins writing to the file. Upon successful completion, it notifies the NMS session by invoking nmsSession::NmsSession_I::historyPMDataCompleted. If Prime Optical fails to create this file or write into the file, the NMS is notified by invoking nmsSession::NmsSession_I::historyPMDataFailed.
Prime Optical allows only one request from each NMS session at a time.
For ONS 15454 SDH NEs, this operation is not supported on low-order STM-path, but is supported on path parameters on E3 and DS3 cards.
For the DS1 module, this operation is not supported on STS/HO path, but is supported on VT data (SONET) and VC11 data (SDH).
For the E1 module, this operation is not supported on STS/HO path, but is supported on VT data (SONET) and VC12 data (SDH).
For the E3 module, this operation is not supported on HO path, but is supported on VC3 data (SDH).
For the DS3 module, this operation is not supported on HO path, but is supported on VC3 data (SDH).
Prime Optical raises an EXCPT_UNABLE_TO_COMPLY exception if another request comes from a session for which one request is already in progress. At any given time, Prime Optical has a maximum of eight such requests. This restriction is imposed because multiple requests downgrade the Prime Optical database performance and have an adverse impact on Prime Optical.
For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, a subset of ONS 15454 SONET and ONS 15454 SDH parameters is supported. The supported parameters match the parameters associated to the port equipment supported by CPT.
Relevance of the validity status for historical PM data and the one-to-one mapping from PM data on the EMS (Prime Optical) to GateWay/CORBA:

- –1—The PM parameter is not applicable to the module in that particular physical location (N/A). The validity status is Invalid.
- –2—The PM data bucket is empty for the parameters in the module, for a time interval (blank). The validity status is Invalid.
- –3—The module in the physical location is unequipped or preprovisioned. The validity status is Invalid.

If the layer rate is L3_Optical_Channel (which requires a DWDM CTP), Prime Optical uses the naming convention defined in the TMF for the CTP.
The CTP format description differs depending on whether it is related to the DWDM.
L3_Optical_Channel is the only layer rate that requires it. If the list of requested layer rates is composed of one element only equal to LR_Optical_Channel, the CTP is considered a DWDM CTP.

Parameters

Name	Type	Input/ Output	Description
client	nmsSession::NmsSession_I	In	Reference to NMSSession. It cannot be null and it must be registered with Prime Optical using emsSessionFactory::EmsSessionFactory_I::getEmsSession. Prime Optical notifies this client about the completion or failure using methods defined in 3.12.1 historyPMDataCompleted and 3.12.2 historyPMDataFailed.
pmTPSelectList	PMTPSelectList_T	In	This parameter specifies which historical PM data to return. This list cannot be empty.
pmParameters	PMParameterNameList_T	In	This parameter specifies which PM parameters within the scope of the pmTPSelectList shall be contained in the file. An empty list means to store all supported parameters. The returned parameters are best effort; that is, among the parameters specified, only the subset of supported parameters will be stored.
startTime	globaldefs::Time_T	In	Specifies the start of the time window for collection (included).
endTime	globaldefs::Time_T	In	Specifies the end of the time window for collection (included).

Prime Optical support for PMTPSelect_T structure for this method is as follows:

- globaldefs::NamingAttributes_T name
  - The name of the object to which this selection applies. Prime Optical supports only ManagedElement, PTP, or CTP name in this field. Note that ManagedElement must be in Prime Optical's management domain. For example, if you change the name of the ME but you pass the old name to this call, no data is written to the file.
- transmissionParameters::LayerRateList_T layerRateList
  - For a list of supported layer rates, see 2.2.1 Layer Rate.
- PMLocationList_T pMLocationList
  - Prime Optical supports only PML_NEAR_END_RX and PML_FAR_END_RX.
- GranularityList_T granularityList
  - Prime Optical supports 15-minute (15min) and 24-hour (24h) PM data. If this list is empty, data for all granularities is reported.

Prime Optical supports the PM parameter values listed in Table 3-5, Table 3-6, Table 3-7, Table 3-9, Table 3-10, and Table 3-11.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when startTime or endTime are not valid, client is null, or pmTPSelectList is empty.

EXCPT_UNABLE_TO_COMPY – Raised when more than one request is sent from the same NMS client.

Compliance
Prime Optical-defined.

3.13.6 getCTMValidHistoryPMData

Synopsis

```
Public void getCTMValidHistoryPMData (
                in nmsSession::NmsSession_I client,
                in PMTPSelectList_T pmTPSelectList,
                in PMParameterNameList_T pmParameters,
                in globaldefs::Time_T startTime,
                in globaldefs::Time_T endTime)
                raises(globaldefs::ProcessingFailureException);
```

Description
This interface performs exactly like the performance::PerformanceManagementMgr_I:: getCTMHistoryPMData interface, but filters out invalid entries.
Compliance
Prime Optical-defined.

3.13.7 deleteHistoryPMDataFile

Synopsis

```
public void deleteHistoryPMDataFile(
in string fileName)
raises(globaldefs::ProcessingFailureException);
```

Description
The NMS can delete the PM data file by invoking this method explicitly. In this case, Prime Optical does not wait for six hours to delete the file. This frees up valuable disk space on the Prime Optical server. It is highly recommended that the NMS invoke this method as soon as it retrieves the PM data file from Prime Optical.

Parameters

Name	Type	Input/Output	Description
filename	string	In	Name of the file that was created by Prime Optical in response to the getCTMHistoryPMData call made by the NMS. This name must be the absolute path on the Prime Optical server.

Throws
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.13.8 getTPHistoryPMData

Synopsis

```
void getTPHistoryPMData(in PMTPSelectList_T pmTPSelectList,
      in PMParameterNameList_T pmParameters,
      in globaldefs::Time_T startTime,
      in globaldefs::Time_T endTime,
      in unsigned long how_many,
      out PMDataList_T pmDataList,
      out PMDataIterator_I pmIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables the NMS to request a filtered set (scoped by the input parameter pmParameters) of the historical PM data for a list of TP measurement points. Within the request for each measurement point, granularity (15 min, 24 hr, NA) and location (nearEnd, farEnd, bidirectional) can be specified. The user ha to specify time interval (StartTime, EndTime) in UTC. The operation mode is best effort. Prime Optical only supports 24 buckets (6 hours) for 15 min PM. If the NMS requests older historical data, Prime Optical returns an EXCPT_NOT_IMPLEMENTED exception. Prime Optical also returns this exception if the NMS requests historical data for 24-hour PM.

This interface uses an iterator so that the NMS can handle a large number of objects.

For a list of supported layer rates and PM parameters, with the exception of ONS 15305 and ONS 15305 CTC, see 3.13.5 getCTMHistoryPMData. The list for ONS 15305 and ONS 15305 CTC is shown in this section.

If the layer rate is L3_Optical_Channel (which requires a DWDM CTP), Prime Optical uses the naming convention defined in the TMF for the CTP.

The CTP format description differs depending on whether it is related to the DWDM.

L3_Optical_Channel is the only layer rate that requires it. If the list of requested layer rates is composed of one element only equal to LR_Optical_Channel, the CTP is considered a DWDM CTP.

Parameters

Name	Type	Input/ Output	Description
pmTPSelectList	PMTPSelectList_T	In	Specifies which historical PM data to return. This list cannot be empty.
pmParameters	PMParameterNameList_T	In	Specifies which PM parameters within the scope of the pmTPSelectList are contained in the file. An empty list means to store all supported parameters. The returned parameters are best effort; that is, among the parameters specified, only the subset of supported parameters is stored.
startTime	globaldefs:Time_T	In	Specifies the start of the time window for collection (included).
endTime	globaldefs:Time_T	In	Specifies the end of the time window for collection (included).
how_many	unsigned long	In	Maximum amount of PM data to return in the first batch.
pmDataList	PMDataList_T	Out	First batch of PM data.
pmIt	PMDataIterator	Out	Iterator used to retrieve the remaining PM data.

Prime Optical support for PMData_T structure is as follows:

- globaldefs::NamingAttributes_T tpName
  - The name of the termination point.

- transmissionParameters::LayerRate_T layerRate
  - Layer of the collected PM data.
  - Prime Optical supports the following layer rates for ONS 15305 and ONS 15305 CTC. For a list of supported layer rates, see 2.2.1 Layer Rate.
- Granularity_T granularity
  - Identifies whether this is 15-minute or 24-hour data. Only 15 min is supported.

Prime Optical supports the pmParamater values listed in Table 3-11 for ONS 15305 and ONS 15305 CTC. For values for other NEs, see 3.13.5 getCTMHistoryPMData.

Table 3-12: ONS 15305 and ONS 15305 CTC pmParameter Values

ONS 15305 Layer Rate	pmParameterName	Location	Interval
LR_Line_OC12_STS12_and_MS_STM4	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr
LR_Line_OC3_STS3_and_MS_STM1	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr
LR_Line_OC48_STS48_and_MS_STM16	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr
LR_Low_Order_TU3_VC3	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr
LR_Section_OC12_STS12_and_RS_STM4	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX	15 min 24 hr
LR_Section_OC3_STS3_and_RS_STM1	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX	15 min 24 hr
LR_Section_OC48_STS48_and_RS_STM16	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX	15 min 24 hr

ONS 15305 Layer Rate	pmParameterName	Location	Interval
LR_STS3c_and_AU4_VC4	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr
LR_VT2_and_TU12_VC12	PMP_BBE PMP_ES PMP_SES	PML_NEAR_END_RX PML_FAR_END_RX	15 min 24 hr

- globaldefs::Time_T retrievalTime
  - ME time at which PM measurement was obtained from the ME.
- PMMeasurementList_T pmMeasurementList:
  - PM values.
- Prime Optical support for PMMeasurement_T structure is as follows:
  - PMParameterName_T pmParameterName
  - Name of the PM measure.
  - PMLocation_T pmLocation
  - Prime Optical supports PML_NEAR_END_RX and PML_FAR_END_RX.
  - float value
  - Value of parameter.
  - string unit
  - Not supported.
  - string intervalStatus
  - Prime Optical supports the following string values:
  - Valid—The PM data value is valid.
  - Invalid—The PM data value is invalid or unavailable.

Throws

EXCPT_NOT_IMPLEMENTED - Raised when the NMS queries historical data older than 6 hours for 15-minute buckets or 24-hour PM.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is not well formed.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

Prime Optical-defined.

3.13.9 getHoldingTime

Synopsis

```
void getHoldingTime(out HoldingTime_T holdingTime)
raises(globaldefs::ProcessingFailureException);
```

Description

The NMS requests from the EMS the number of hours that it stores 15-minute and 24-hour PM data records. This interface reports the minimum time. If the EMS does not store PM data, this method reports the time supported by the NE (the minimum time of all managed NEs).

Prime Optical gives you the option to configure the holding time through the Prime Optical GUI client.

Parameters

Name	Type	Input/Output	Description
holdingTime	HoldingTime_T	Out	Time period (in hours) within which 15-minute and 24-hour PM data records can be retrieved.

Prime Optical support for the HoldingTime_T structure is as follows:

- Short storeTime24hr—Minimum store time, in hours, for 24-hour data. Prime Optical reports 0 if PM data is held indefinitely.
- Short storeTime15min—Minimum store time, in hours, for 15-minute data. Prime Optical reports 0 if PM data is held indefinitely.

Throws

EXCPT_NOT_IMPLEMENTED - Raised if the EMS is unable to support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NE_COMM_LOSS - Raised if communication to an NE is lost and the holding time cannot be retrieved.

Compliance

TMF-defined.

3.13.10 getAllCurrentPMData

Synopsis

```
void getAllCurrentPMData(in PMTPSelectList_T pmTPSelectList,
      in PMParameterNameList_T pmParameters,
      in unsigned long how_many,
      out PMDataList_T pmDataList,
      out PMDataIterator_I pmIt)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface enables an NMS to request a filtered set (scoped by the input parameter pmParameters) of the current PM data for a list of TP measurement points. Within the request for each measurement point, granularity (15 min, 24 hr, N/A) and location (nearEnd, farEnd, bidirectional) can be specified. The operation mode is best effort.

Current data is generally marked as incomplete, as the current bin is not completed. If no PM data is available as specified, an empty list is returned.

This interface uses an iterator to allow the NMS to handle a large number of objects.

For a list of supported layer rates and PM parameters, see 3.13.5 getCTMHistoryPMData.

If the layer rate is L3_Optical_Channel (which requires a DWDM CTP), Prime Optical uses the naming convention defined in the TMF for the CTP.

The CTP format description differs depending on whether it is related to the DWDM.

L3_Optical_Channel is the only layer rate that requires it. If the list of requested layer rates is composed of one element only equal to LR_Optical_Channel, the CTP is considered a DWDM CTP.

For CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH, a subset of ONS 15454 SONET and ONS 15454 SDH parameters is supported. The supported parameters match the parameters associated to the port equipment supported by CPT.

The getAllCurrentPMData interface returns values with a 15-minute granularity if the requested granularity is 24 hour. For a list of supported layer rates, see 2.2.1 Layer Rate.

Parameters

Name	Type	Input/ Output	Description
pmTPSelectList	PMTPSelectList_T	In	Specifies which real-time PM data to return. This list cannot be empty.
pmParameters	PMParameterNameList_T	In	Specifies which PM parameters within the scope of the pmTPSelectList are contained in the file. An empty list means to store all supported parameters. The returned parameters are best effort, meaning among the parameters specified, only the subset of supported parameters is stored.
how_many	unsigned long	In	Maximum number of PM data to return in the first batch.
pmDataList	PMDataList_T	Out	First batch of PM data.
pmIt	PMDataIterator_I	Out	Iterator used to retrieve the remaining PM data.

For detail of PMData_T structure, see 3.13.8 getTPHistoryPMData.

Throws

EXCPT_NOT_IMPLEMENTED - Raised when this operation is not supported by the EMS or not supported for an ME type.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is not well formed.

EXCPT_NE_COMM_LOSS - Raised when communication to the NE is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

TMF-defined.

3.13.11 getMEPMcapabilities

Synopsis

```
void performance::PerformanceManagementMgr_I::getMEPMcapabilities  (  in
globaldefs::NamingAttributes_T  meName,
  in transmissionParameters::LayerRate_T  layerRate,
  out PMParameterList_T  pmParameterList
 )  raises (globaldefs::ProcessingFailureException)
```

Description

This interface allows the NMS to request from the EMS the set of PM parameters supported by a particular ME at a specific layer.

For a list of supported layer rates, see 3.13.5 getCTMHistoryPMData.

Parameters

Name	Type	Input/ Output	Description
meName	NamingAttributes_T	In	Name of the managed element.
LayerRate	LayerRate_T	In	Layer rate at which PM parameters must be displayed.
pmParameterList	PMParameterList_T	Out	List of PMParameter_T, which returns the PM

Name	Type	Input/ Output	Description
			parameter name of the PM measure qualified by its location.

Throws

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when meName does not reference a managed element object or layerRate contains undefined rates.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

TMF-defined.

3.13.12 Passive Unit EMS PM

In Prime Optical 10.7, passive units are supported. See 2.2.2.1 Passive Unit Inventory for the list of passive units that are supported. For passive units, PM is not supported.

3.14 protection::ProtectionMgr_I

A handle to an instance of this interface is gained through the getManager operation in 3.4 EmsSession::EmsSession_I. This interface is inherited from 3.2 common::Common_I.

```
interface ProtectionMgr_I: common::Common_I
```

It also implements the getCapabilities method. Operations described in the following subsections are returned to the NMS when it invokes getCapabilities.

Prime Optical supports 1:1 equipment protection, 1:N equipment protection, 1+1 port protection, 2-fiber BLSR protection groups, and 4-fiber BLSR protection groups.

3.14.1 getAllProtectedTPNames

Synopsis

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

Description

This interface retrieves the list of PTPs that can carry protected traffic. The result is independent of the current switch status. You can use the Prime Optical client GUI to create and name protection groups. If there are multiple protection groups with the same name, Prime Optical returns the ProcessingFailureException UNABLE_TO_COMPLY.

Parameters

Name	Type	Input/ Output	Description
pgName	globaldefs::NamingAttributes_T	In	Name of the protection group.
how_many	unsigned long	In	Maximum number of PTPs to return in the first batch.

Name	Type	Input/ Output	Description
nameList	globaldefs::NamingAttributesList_T	Out	First batch of PTPs that can carry protected traffic. Protected PTPs are reported.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator used to retrieve the remaining PTPs that can carry protected traffic.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when pgName does not reference a protection group object.

EXCPT_ENTITY_NOT_FOUND - Raised when pgName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

TMF-defined.

3.14.2 getAllProtectionGroups

Synopsis

```
void getAllProtectionGroups(
in globaldefs::NamingAttributes_T meName,
      in unsigned long how_many,
      out ProtectionGroupList_T pgList,
      out ProtectionGroupIterator_I pgpIt)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface is used by the client to discover all of the protection groups currently in operation for the ME. Note that Prime Optical does not have any protection groups for path protection.
The supported protection group types are PGT_MSP_1_FOR_N, PGT_MSP_1_PLUS_1, PGT_2_FIBER_BLSR, PGT_4_FIBER_BLSR, and PGT_Y_CABLE.

The following table lists default names for the protection groups when created through CTC.

PGT_MSP_1_FOR_1	Slot n (Equipment Native Name), where n represents the slot number of the protecting equipment. For example, slot 3 (DS3IN).
PGT_MSP_1_FOR_N	Slot n (Equipment Native Name), where n represents the slot number of the protecting equipment. For example, slot 3 (DS3XM).
PGT_MSP_1_PLUS_1	Slot n (Equipment Native Name), port m, where n represents the slot number and m represents the protecting port number. For example, slot 5 (OC3), port 1.
PGT_2_FIBER_BLSR	No defaults; the name is a user-specified ring ID from 1 to 9999.
PGT_4_FIBER_BLSR	No defaults; the name is a user-specified ring ID from 1 to 9999.
PGT_Y_CABLE	SHELF N SLOT M PORT O, where N represents the shelf number, M represents the slot number, and O represents the protecting port number.

	For example, SHELF 5 SLOT 6 PORT 1-1

If the PGP configuration is 4-fiber BLSR, there are three protection groups reported for this configuration: two PGT_1_FOR_N component groups and one PGT_4_FIBER_BLSR group.
The ONS 15600 SONET and ONS 15600 SDH support only PGT_MSP_1_PLUS_1 protection group.

Parameters

Name	Type	Input/Output	Description
meName	Globaldefs::NamingAttributes_T	In	Name of the ME for which the request is made.
how_many	unsigned long	In	Maximum number of protection groups to report in the first batch.
pgList	ProtectionGroupList_T	Out	First batch of protection groups.
PgIt	ProtectionGroupIterator_I	Out	Iterator used to access the remaining protection groups (PGs), if any.

Prime Optical support for ProtectionGroup_T structure is as follows:

- globaldefs::NamingAttributes_T name
  - Supported. This field contains three tuples. The first tuple contains the EMS name; the second tuple contains the ME name; and the last tuple contains the nativeEMSName of this protection group.
- string userLabel
  - Not supported.
- string nativeEMSName
  - Supported. The name is reported by the NE. It is Ring Id for 2_FIBER_BLSR and 4_FIBER_BLSR. It is <Ring Id>-EAST or <Ring Id>-WEST for the two component groups of 4-fiber BLSR PGP.
- string owner
  - Not supported.
- ProtectionGroupType_T protectionGroupType
  - Supported.
- ProtectionSchemeState_T protectionSchemeState
  - Not supported.
- ReversionMode_T reversionMode
  - Supported.
- transmissionParameters::LayerRate_T rate
  - Line rate of the PTPs of the protection group.
- globaldefs::NamingAttributesList_T pgpTPList
  - Supported.
- globaldefs::NVSList_T pgpParameters
  - Prime Optical only supports wtrTime, the values of wtrTime are in seconds.
- globaldefs::NVSList_T additionalInfo
  - Prime Optical supports BIDIRECTIONAL for 1_PLUS_1 protection group. The values are true and false.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when meName does not reference a managed element object.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance
TMF-defined.

3.14.3 getProtectionGroup

Synopsis

```
void getProtectionGroup(
in globaldefs::NamingAttributes_T pgName,
      out ProtectionGroup_T protectionGroup)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface retrieves the current status of a protection group. This service is needed so that even if a filter is established, the NMS can query the status of a protection group. If there is more than one protection group with the same name, Prime Optical returns the ProcessingFailureException UNABLE_TO_COMPLY.

Parameters

Name	Type	Input/ Output	Description
pgName	globaldefs::NamingAttributes_T	In	Name of the protection group that the client is interested in.
protectiongroup	ProtectionGroup_T	Out	Returned protection group.

For details about the ProtectionGroup_T structure, see 3.14.2 getAllProtectionGroups.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when protectionGroupName does not reference a protection group.

EXCPT_ENTITY_NOT_FOUND - Raised when meName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.14.4 performProtectionCommand

Synopsis

```
void performProtectionCommand(
in ProtectionCommand_T protectionCommand,
      in globaldefs::NamingAttributes_T reliableSinkCtpOrGroupName,
      in globaldefs::NamingAttributes_T fromTp,
      in globaldefs::NamingAttributes_T toTp,
      out SwitchData_T switchData)
    raises(globaldefs::ProcessingFailureException);
```

Description

This interface executes a protection switch. The protection switch can be performed through a protection switch command on a protection group. Prime Optical requests the EMS to move the traffic received from the fromTP to the toTP. The same command is used to clear all existing commands. If there is more than one protection group with the same name, Prime Optical returns the ProcessingFailureException UNABLE_TO_COMPLY.

A protection switch command is used to change the flow of traffic from working to protect or from protect to working entity. All entities that are represented as working entities before the switch are still represented as working entities after the switch. Similarly, all entities that are represented as protect entities before the switch are still represented as protect entities after the switch.

For 4-fiber SONET/SDH SPrings, the span switch is performed on the PGT_MSP_1_FOR_N component protection group, and the ring switch on the PGT_4_FIBER_BLSR group. For example, if the switch is on the east span, the PGP name is ring-ID-EAST.

For 2-fiber ring switch, it can be either E -> W or W -> E.

For 4-fiber ring switch, it can be Ew (east working) -> Wp (west protecting), Ep -> Ww, Ww -> Ep, and Wp -> Ew.

For 4-fiber span switch, it can be Ew -> Ep, Ep ->Ew, Ww -> Wp, and Wp -> Ww.

Prime Optical validates whether or not fromTp and toTp are part of same protection group. If this validation fails then it throws ProcessingFailureException with EXCPT_INVALID_INPUT. Prime Optical always performs the command on the fromTp for PC_LOCKOUT and PC_EXERCISE. Prime Optical also verifies if it is possible to perform the command; if not, it throws EXCPT_UNABLE_TO_COMPLY.

Table 3-13: Protection Command Protection Groups

Protection Group	Protection Command Supported by Prime Optical
PGT_2_FIBER_BLSR	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_MANUAL_SWITCH PC_EXERCISE
PGT_4_FIBER_BLSR Protection command applies to both span and ring components.	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_MANUAL_SWITCH PC_EXERCISE
PGT_MSP_1_PLUS_1	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_MANUAL_SWITCH PC_UNLOCK
PGT_MSP_1_FOR_N	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_MANUAL_SWITCH PC_UNLOCK
PGT_MSP_1_FOR_1	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_MANUAL_SWITCH PC_UNLOCK

Protection Group	Protection Command Supported by Prime Optical
PGT_Y_CABLE	PC_CLEAR PC_LOCKOUT PC_FORCED_SWITCH PC_UNLOCK

Parameters

Name	Type	Input/ Output	Description
protectionCommand	ProtectionCommand_T	In	Command to perform.
reliableSinkCtpOrGroupName	globaldefs::NamingAttributes_T	In	Prime Optical supports only protection group name.
fromTp	globaldefs::NamingAttributes_T	In	Present source of the traffic.
toTp	globaldefs::NamingAttributes_T	In	Requested source of the traffic after the command.
switchData	SwitchData_T	Out	Protection switch status of the toTp provided after the execution of the command.

Prime Optical support for SwitchData _T structure is as follows:

- ProtectionType_T protectionType
  - Supported. Prime Optical always reports PT_MSP_APS.
- SwitchReason_T switchReason
  - Supported. Prime Optical always reports SR_MANUAL.
- transmissionParameters::LayerRate_T layerRate
  - The layer that this switch is relevant to.
- globaldefs::NamingAttributes_T groupName
  - Identifies the protection group for which protection switch status is being reported.
- globaldefs::NamingAttributes_T protectedTP
  - This is the TP that is protected.
  - For a 2-fiber BLSR, this will be from TP.
  - For a 1:1, this is always the working TP.
  - For a 1:N MSP, this is always the working TP.
  - For a 1+1 MSP, this is always the working TP.
  - For a 4-fiber BLSR, this is the working TP.
- globaldefs::NamingAttributes_T switchToTP
  - Prime Optical reports toTp.
- globaldefs::NVSList_T additionalInfo
  - Not supported.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when reliableSinkCtpOrGroupName, fromTp, or toTp references objects of the correct type, but that are invalid in the context of this operation.

EXCPT_ENTITY_NOT_FOUND - Raised when reliableSinkCtpOrGroupName references a CTP or PG object that does not exist, or when fromTp or toTp references a CTP object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised if the EMS is unable to perform the operation.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance
TMF-defined.

3.14.5 createProtectionGroup

Synopsis

```
void createProtectionGroup(
in PGPCreateData_T pgpCreateData,
      out ProtectionGroup_THolder createdPGP)
    raises(globaldefs::ProcessingFailureException);
```

Description
This interface enables the NMS to permanently create a protection group in an ME. The successful result of this operation is the creation of the protection group object. The resulting object is returned. Prime Optical generates an object creation event upon successful creation of the protection group.
Prime Optical supports only the creation of 1_PLUS_1, 1_FOR_N, 1_FOR_1 (a specific case where the cards are placed in adjacent slots), 2-fiber BLSR, 4-fiber BLSR, SPLITTER, L2_1_PLUS, and Y-CABLE protection groups.

✎ Prime Optical enables only the creation of 2-fiber and 4-fiber protection groups at the managed element level and not at the network level.

For 100G-LC-C, 10x10G_LC, and M-CFP-LC cards, the following Y-Cable protections are supported:

- TXP_10G (Carpegna card in a standalone configuration) for 10GE and 8GFC payloads only.
- MXP_10x10G (Carpegna and Titano cards in peer configuration) 10GE and 8GFC payloads only.
- CFP_MXP (Fumaiolo and Titano cards in peer configuration) 40GE payloads only.
- CFP_TXP (Fumaiolo and Titano cards in peer configuration) for 100GE payloads only.

For the ONS 15305 R3.0.x, the name of the 1+1 MSP protection group is assigned by the NE itself. The pgpName attribute that is passed as input is ignored. The object creation event reports the protection group name that is set by the NE.

Parameters

Name	Type	Input/ Output	Description
pgpCreateData	PGPCreateData_T	In	Data that describes the protection group to create.
createdPGP	ProtectionGroup_T	Out	Resulting created protection group.

Prime Optical support for PGPCreateData_T structure is as follows:

- PGPCreateData_T structure for 1+1 and 1:N protection group creation.

Type	Parameter Name	1+1	1:N	1:1
NamingAttributes_T	pgpName	Protection group name	Protection group name	Protection group name

Type	Parameter Name	1+1	1:N	1:1
string	userlabel	N/A	N/A	N/A
boolean	forceUniqueness	False	False	False
string	owner	N/A	N/A	N/A
ProtectionGroupType_T	pgpType	PGT_MSP_1_PLUS_1	PGT_MSP_1_FOR_N	PGT_MSP_1_FOR_1
ReversionMode_T	reversionMode	Could be revertive or nonrevertive ReversionMode_T. RM_REVERTIVE or ReversionMode_T. RM_NON_REVERTIVE	ReversionMode_T. RM_REVERTIVE 1:N is always revertive.	Could be revertive or nonrevertive ReversionMode_T. RM_REVERTIVE or ReversionMode_T. RM_NON_REVERTIVE
NamingAttributesList_ T	workingEquipment OrTPList	Working PTP name	Name of working equipment	Name of working equipment
NamingAttributesList_ T	protectEquipment OrTPList	Protect PTP name	Name of protect equipment	Name of protect equipment
NVList_T	pgpParameters	Name: WTR_TIME (reversion time) Value: Time in seconds	Name: WTR_TIME (reversion time) Value: Time in seconds	Name: WTR_TIME (reversion time) Value: Time in seconds
NVList_T	additionalInfo	Name: BIDIRECTIONAL Value: true or false	N/A	N/A

- PGPCreateData_T structure for 2-fiber/4-fiber BLSR ring creation.

Type	Parameter Name	2-Fiber BLSR	4-Fiber BLSR
NamingAttributes_T	pgpName	Ring name	Ring name
string	userlabel	N/A	N/A
boolean	forceUniqueness	False	False
string	owner	N/A	N/A
ProtectionGroupType_T	pgpType	PGT_2_FIBER_BLSR	PGT_4_FIBER_BLSR
ReversionMode_T	reversionMode	Always revertive. ReversionMode_T. RM_REVERTIVE	ReversionMode_T. RM_REVERTIVE
NamingAttributesList_T	workingEquipmentOrTPList	First element in the list contains the west-line PTP name Second element in the list contains the east-line PTP name	First element in the list contains the west-line PTP name Second element in the list contains the east-line PTP name

Type	Parameter Name	2-Fiber BLSR	4-Fiber BLSR
NamingAttributesList_T	protectEquipmentOrTPList	N/A. Must be an array of 0 length	First element in the list contains the west-line protect PTP Second element contains the east-line protect PTP
NVList_T	pgpParameters	Name: WTR_TIME (reversion time) Value: Time in seconds Name: SPRINGNodeId (Node ID) Value: 0–31	Name: WTR_TIME (reversion time) Value: Time in seconds Name: SPRINGNodeId (Node ID) Value: 0–31
NVList_T	additionalInfo	N/A If there is no additional info, an array of zero length is passed as the input parameter. The list cannot be null.	Name: SPAN_REVERSION Value: 0–600 seconds (in 30-second increments) If not specified, Prime Optical uses the default value of 300 seconds.

- PGPCreateData_T structure for SPLITTER, L2_1_PLUS and Y_CABLE protection.

SPLITTER	L2 -1_PLUS_1	Y_CABLE
Protection group name	Protection group name	Protection group name
N/A	N/A	N/A
False	False	False
N/A	N/A	N/A
PGT_SPLITTER	PGT_L2_1_PLUS_1	PGT_YCABLE
Could be revertive or nonrevertive ReversionMode_T. RM_REVERTIVE or ReversionMode_T. RM_NON_REVERTIVE	Could be revertive or nonrevertive ReversionMode_T. RM_REVERTIVE or ReversionMode_T. RM_NON_REVERTIVE	Could be revertive or nonrevertive ReversionMode_T. RM_REVERTIVE or ReversionMode_T. RM_NON_REVERTIVE
Working PTP name	Working PTP name	Working PTP name
Protect PTP name	Protect PTP name	Protect PTP name
Name: WTR_TIME (reversion time) Value: Time in seconds	Name: WTR_TIME (reversion time) Value: Time in seconds	Name: WTR_TIME (reversion time) Value: Time in seconds
N/A	N/A	N/A

✐ Y_CABLE protection group is supported by AR-XP, AR-MXP, 100G_LC, 10x10G_LC, and M-CFP-LC cards.

The following enum is used to define the various protection group types:

```
enum ProtectionGroup_T {
    PGT_MSP_1_PLUS_1,
    PGT_MSP_1_FOR_1,
    PGT_MSP_1_FOR_N,
    PGT_2_FIBER_BLSR,
    PGT_4_FIBER_BLSR
    PGT_SPLITTER
    PGT_L2_1_PLUS_1
    Y_CABLE
    }
```

The following enum is used to define the reversion mode:

```
enum ReversionMode_T {
RM_UNKNOWN,
RM_REVERTIVE,
RM_NON_REVERTIVE
}
```

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NOT_IMPLEMENTED - Raised if the EMS cannot support this service.
EXCPT_INVALID_INPUT - Raised when equipment or TP name does not reference an
equipment or TP object.
EXCPT_ENTITY_NOT_FOUND - Raised when equipment or TP name references an object
that does not exist.
EXCPT_UNABLE_TO_COMPLY - Raised when the protection group cannot be created.
EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.
```

Compliance

Prime Optical-defined.

3.14.6 deleteProtectionGroup

Synopsis

```
void deleteProtectionGroup(
in  globaldefs::NamingAttributes_T pgpName)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface allows the NMS to permanently delete a protection group from the ME. The successful result of this operation is the potential deletion of the protection group object. Prime Optical generates an object deletion event on successful deletion of the protection group. Prime Optical supports deletion of all protection group types. If there is more than one protection group with the same name, Prime Optical returns an UNABLE_TO_COMPLY processing failure exception.

Parameters

Name	Type	Input/ Output	Description
pgpName	globaldefs::NamingAttributes_T	In	Name of the protection group to delete.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INVALID_INPUT - Raised when pgpName does not reference a protection group object.

EXCPT_ENTITY_NOT_FOUND - Raised when pgpName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the protection group cannot be deleted from the NE.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.14.7 getAssociatedProtectionGroup

Synopsis

```
void getAssociatedProtectionGroup(
in globaldefs::NamingAttributes_T ptpName,
out protection::ProtectionGroup_T protectionGrp)
raises(globaldefs::ProcessingFailureException);
```

Description

This interface retrieves the protection group using the specified PTP name. The PTP name specified can be either the working PTP or the protect PTP.

Parameters

Name	Type	Input/Output	Description
ptpName	globaldefs::NamingAttributes_T	In	Name of the PTP to verify an association to any protection group.
protectiongroup	ProtectionGroup_T	Out	Returned protection group.

For details about the ProtectionGroup_T structure, see 3.14.2 getAllProtectionGroups.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when ptpName does not reference a PTP.

EXCPT_ENTITY_NOT_FOUND - Raised when the PTP name references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

Compliance

Prime Optical-defined.

3.15 session::Session_I

The Session_I interface provides capabilities for managing the client-server connection. It enables either a client or a server to detect communication loss with the associated party. There are two Session_I objects for each communication session between an NMS and an EMS. One Session_I object is maintained on the NMS and the other Session_I object is maintained on the EMS. The Session_I object maintained on the EMS is actually an EmsSession_I, while the Session_I object maintained on the NMS is actually an NmsSession_I (both inherit from Session_I).
Each Session_I object is responsible for pinging the other Session_I object periodically to detect communication failures. Prime Optical invokes this method on NmsSession_I every five minutes. If this call raises an exception, Prime Optical assumes the session is no longer connected and deletes all resources allocated to it.

Attributes

```
readonly attribute Session_I associatedSession;
```

This attribute contains a reference to the Session_I on the NMS/EMS associated with the object. This attribute can be checked to verify the NmsSession_I/EmsSession_I association is still valid (particularly in case of communication failures).

3.15.1 endSession

Synopsis

```
oneway void endSession();
```

Description
This interface enables a controlled disconnect between parties.
Parameters
None.
Throws
None.
Compliance
TMF-defined.

3.15.2 ping

Synopsis

```
void ping();
```

Description
This interface enables the detection of communication loss. Prime Optical pings the NMS every five minutes.
Parameters
None.
Throws
None.
Compliance
TMF-defined.

3.16 mtnmVersion::Version_I

3.16.1 getVersion

Synopsis
```
string getVersion();
```

Description
This interface returns the version of the IDL that the corresponding EMS object supports.
The format of the returned string is as follows:
Release.Major[.Minor], where Release, Major, and Minor are digit strings.
For example:

- *2.1 indicates release 2 and major release 1*
- *1.3 indicates release 1 and major release 3*

✎ *x.y* has the same meaning as x.y.0. The minor digit is used for fixing bugs in the major release. For example, 1.2.1 is a minor release on 1.2. Any changes to IDL files must be approved by the Specification Authority. For TMF 814 version 2.0, the version is 2.0.

Parameters
None.
Return Value
```
string
```

Throws
None.
Compliance
TMF-defined.

3.17 cpoEmsManager:: CpoEmsManager_I

The CpoEmsManager_I interface is used to gain access to non-TMF standard operations which deal with the EMS itself. A handle to an instance of this interface is gained via the emsSession::EmsSession_I ::getManager operation in emsSession. This interface is inherited from common::Common_I.
```
interface CpoEmsManager_I: common::Common_I
```

3.17.1 getAllEMSAndMEUnacknowledgedActiveAlarms

Synopsis
```
void getAllEMSAndMEUnacknowledgedActiveAlarms(
        in notifications::ProbableCauseList_T excludeProbCauseList,
        in notifications::PerceivedSeverityList_T excludeSeverityList,
                    in globaldefs::NVSList_T additionalInfo,
        in unsigned long how_many,
        out notifications::EventList_T eventList,
        out notifications::EventIterator_I eventIt)
       raises(globaldefs::ProcessingFailureException);
```

Description

This API returns all EMS and managed element alarms that are active and unacknowledged, or all GMPLS that are active and unacknowledged, depending on whether or not the additional information is provided.

Parameters

Name	Type	Input/ Output	Description
excludeProbCauseList	notifications::ProbableCauseList_T	In	The list of probable causes to exclude from being reported as an event.
excludeSeverityList	notifications::PerceivedSeverityList_T	In	List of severities to exclude from the output event list.
additionalInfo	globaldefs::NVSList_T	In	If filled, it allows retrieval of all GMPLS alarms. It has only one allowed value: Name = alarmType Value = gmpls If not filled, it exhibits the same behavior as the EMSManager API.
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.

3.18 cpoManagedElementManager_I:: CpoManagedElementManager_I

The CpoManagedElementManager_I interface is used to gain access to operations which deal with managed elements and termination points with CPO dedicated APIs. A handle to an instance of this interface is gained via the getManager operation in emsSession. This interface is inherited from common::Common_I.

interface cpoManagedElementManager_I : common::Common_I

3.18.1 getAllUnacknowledgedActiveAlarms

Synopsis

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

Description

This API returns all alarms that are active and unacknowledged, or all GMPLS that are active and unacknowledged for the selected managed element, depending on whether or not the additional information is provided.

Parameters

Name	Type	Input/ Output	Description
meName	globaldefs::NamingAttributes_T	In	Managed element name.
excludeProbCauseList	notifications::ProbableCauseList_T	In	The list of probable causes to exclude from being reported as an event.
excludeSeverityList	notifications::PerceivedSeverityList_T	In	List of severities to exclude from the output event list.
additionalInfo	globaldefs::NVSList_T	In	If filled, it allows retrieval of all GMPLS alarms. It has only one allowed value: Name = alarmType Value = gmpls If not filled, it exhibits the same behavior as the EMSManager API.
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.

3.19 Transmission Descriptor Interfaces

```
enum ServiceCategory_T {
     SC_CBR,
     SC_VBRRT,
     SC_VBRNRT,
     SC_ABR,
     SC_UBR,
     SC_GFR,
     SC_NA
    };
  typedef globaldefs::NVSList_T TrafficParameterList_T;

    struct TransmissionDescriptor_T {
      globaldefs::NamingAttributes_T name;
      string userLabel;
      string nativeEMSName;
```

```
  string owner;
  transmissionParameters::LayeredParameterList_T transmissionParams;
  globaldefs::NVSList_T additionalTPInfo;
  globaldefs::NamingAttributes_T containingTMDName;
  string externalRepresentationReference;
  globaldefs::NVSList_T additionalInfo;
};


struct TMDCreateData_T {
 string userLabel;
 boolean forceUniqueness;
 string owner;
 transmissionParameters::LayeredParameterList_T transmissionParams;
 globaldefs::NVSList_T additionalTPInfo;
 globaldefs::NamingAttributes_T containingTMDName;
 string externalRepresentationReference;
 globaldefs::NVSList_T additionalCreationInfo;
};

 typedef sequence<TransmissionDescriptor_T> TransmissionDescriptorList_T;
```

3.19.1 getAllTransmissionDescriptors

Synopsis

```
void getAllTransmissionDescriptors(
          in unsigned long how_many,
          out TransmissionDescriptorList_T transmissionDescList,
          out TransmissionDescriptorIterator_I transmissionDescIt)
         raises (globaldefs::ProcessingFailureException);
```

Description

This interface allows an NMS to request all of the transmissionDescriptors that are under the control of the transmissionDescriptorMgr_I. To allow the NMS to handle a large number of objects, this operation uses an iterator.

This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.

Parameters

Name	Type	Input/ Output	Description
how_many	unsigned long	In	Number of iterators to return in

Name	Type	Input/ Output	Description
			transmissionDescList.
transmissionDescList	transmissionDescriptor::TransmissionDescriptorList	Out	First batch of iterators.
transmissionDescIt	transmissionDescriptor::TransmissionDescriptorIterator_I	Out	Iterator used to access the remaining transmission descriptors.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

None.

3.19.2 getAllTransmissionDescriptorsNames

Synopsis

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

Description

This interface returns the transmission descriptor structure for the given tmdName. The transmission descriptor structure contains an NVSList of traffic parameters. The traffic parameters returned are the parameters in place on the actual transmission descriptor. If there are no traffic parameters, the NVSList is empty.

This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.

Parameters

Name	Type	Input/Output	Description
how_many	unsigned long	In	Number of iterators to return in nameList.
nameList	globaldefs::NamingAttributesList_T	Out	First batch of iterators.
nameIt	globaldefs::NamingAttributesIterator_I	Out	Iterator used to retrieve the remaining iterator names.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance
None.

3.19.3 getTransmissionDescriptor

Synopsis

```
void getTransmissionDescriptor(
          in globaldefs::NamingAttributes_T tmdName,
          out TransmissionDescriptor_T tmd)
         raises (globaldefs::ProcessingFailureException);
```

Description
This interface returns the transmission descriptor structure for the given tmdName. The transmission descriptor structure contains an NVSList of traffic parameters. The traffic parameters returned are the parameters in place on the actual transmission descriptor. If there are no traffic parameters, the NVSList is empty.
This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.
Parameters

Name	Type	Input/ Output	Description
tmdName	globaldefs::NamingAttributes_T	In	Name of the transmission descriptor.
tmd	TransmissionDescriptor_T	Out	Returned transmission descriptor.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when tmdName does not reference a transmission
descriptor object.
EXCPT_ENTITY_NOT_FOUND - Raised when tmdName references an object that does
not exist.
```

Compliance
None.

3.19.4 getAssociatedTPs

Synopsis

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

Description
This interface allows an NMS to request all of the TPs associated with the specified transmission descriptor. If there are no associated TPs, an empty list is returned.
This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.
Parameters

Name	Type	Input/ Output	Description
transmissionDescriptorName	globaldefs::NamingAttributes_T	In	Name of the transmission descriptor.
how_many	unsigned long	In	Maximum number of TPs to return in the first batch.
tpList	terminationPoint::TerminationPointList_T	Out	First batch of TPs.
tpIt	terminationPoint::TerminationPointIterator_I	Out	Iterator to access the remaining TPs.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when transmissionDescriptorName does not reference a transmission descriptor object.

EXCPT_ENTITY_NOT_FOUND - Raised when transmissionDescriptorName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance

None.

3.19.5 createTransmissionDescriptor

Synopsis

```
void createTransmissionDescriptor(
          in  TMDCreateData_T      newTMDCreateData,
          out TransmissionDescriptor_T newTransmissionDescriptor)
         raises (globaldefs::ProcessingFailureException);
```

Description

This interface is used to create a new transmission descriptor on the server. Data representing the new transmission descriptor is passed as input. The resulting transmission descriptor is returned.
This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.

Parameters

Name	Type	Input/ Output	Description
newTMDCreateData	TMDCreateData_T	In	Information about the transmission descriptor to create.
newTransmissionDescriptor	TransmissionDescriptor_T	Out	Result of the creation.

Throws
globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support creation of TMDs via this interface.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised if newTMDCreateData contains invalid data.

EXCPT_USERLABEL_IN_USE - Raised when the user label uniqueness constraint is not met.

EXCPT_CAPACITY_EXCEEDED - Raised when the EMS reaches the maximum number of transmission descriptors that it can support.

Compliance
None.

3.19.6 deleteTransmissionDescriptor

Synopsis

```
void deleteTransmissionDescriptor(
          in globaldefs::NamingAttributes_T transmissionDescriptorName)
          raises (globaldefs::ProcessingFailureException);
```

Description
This interface deletes a transmission descriptor from the server. If the service is called with the name of a nonexistent transmission descriptor, it succeeds.
This interface is introduced in Prime Optical to comply with MTNM version 3.0. The current implementation returns a globaldefs::ProcessingFailureException stating that this API is not supported.
Parameters

Name	Type	Input/ Output	Description
transmissionDescriptorName	globaldefs::NamingAttributes_T	In	Information about the transmission descriptor to delete.

Throws
globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support deletion of TMDs via this interface.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_OBJECT_IN_USE - Raised if there are TPs that are using the transmission descriptor.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the transmission descriptor to delete does not exist.

Compliance
None.

3.20 Software and Data Manager Interface

The softwareAndDataManager interface is used to gain access to operations related to backing up ME configuration data.
A handle to an instance of this interface is gained via the emsSession::EmsSession_I::getManager() operation in Manager.
This interface is inherited from interface SoftwareAndDataMgr_I:common::Common_I.
The softwareAndDataManager interface also implements the getCapabilities method. When the NMS invokes the getCapabilities method, the operations described in the following sections are returned.

3.20.1 backupME

Synopsis

```
void backupME(
        in globaldefs::NamingAttributes_T managedElementName)
        raises(globaldefs::ProcessingFailureException)
```

Description
This asynchronous operation initiates a backup request on the specified ME. This creates a backup of the current configuration on the ME. The EMS sends the results of the ME backup operation via the notification service. The NMS can obtain the status of any backup operation on demand. An NT_BACKUP_STATUS event is sent to indicate the result of the backup operation.
Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which a backup of configuration data is required.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when input parameters are syntactically
incorrect.
EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.
EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.
```

Compliance
TMF-defined.

3.20.2 getMEBackupStatus

Synopsis

```
void getMEBackupStatus (
        in globaldefs::NamingAttributes_T managedElementName,
        out BackupStatus_T  backupStatus)
        raises(globaldefs::ProcessingFailureException);
```

Description
The NMS uses this method to obtain the current backup status of an individual ME.

Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which the status of the backup configuration data is required.
backupStatus	BackupStatus_T	Out	Backup status of the ME.

BackupStatus_T is defined as:
Current_OperationStatus_T opStatus
string failureReason
This data structure identifies the status of the backup operation for a managed element. The failure reason is present if the operation status indicates a failure (in abort state).

- Current_OperationStatus_T opStatus—Indicates the current operational status of the backup. The structure of Current_OperationStatus_T is defined as:

```
enum Current_OperationStatus_T { COS_Idle, COS_Pending, COS_InProgress,
COS_Completed, COS_Aborted };
```

where:

  - COS_Idle—The backup operation is waiting.
  - COS_Pending—The backup operation is queued.
  - COS_InProgress—The backup operation is running.
  - COS_Completed—The backup operation succeeded.
  - COS_Aborted—The backup operation was canceled.

string failureReason—A free-form text string provided if the opStatus value is COS_Aborted to explain the reason for the abort. (For example, “Comms loss with NE.”)

Throws

```
globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically
incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.
```

Compliance

TMF-defined.

3.20.3 abortMEBackup

Synopsis

```
void abortMEBackup (
        in globaldefs::NamingAttributes_T managedElementName)
        raises(globaldefs::ProcessingFailureException)
```

Description

This interface aborts a pending backup of ME configuration data. If the backup operation on the ME is in progress or if there are no pending backup operations, an exception is generated.

Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which a backup of

			configuration data is required.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

EXCPT_NOT_IN_VALID_STATE - Raised when there is no pending or in-progress backup of configuration data on the ME.

Compliance

TMF-defined.

3.20.4 getBackupList

Synopsis

```
void getBackupList(
        in globaldefs::NamingAttributesList_T managedElementNameList,
        in unsigned long how_many,
        out BackupIdList_T backupList,
        out BackupIdIterator_I backupIt)
        raises(globaldefs::ProcessingFailureException)
```

Description

The NMS invokes this method to determine what configuration data backups exist in the EMS for a specified list of MEs.

This interface returns a list of IDs of all configuration data backups for the specified MEs. If the list of MEs in the request is empty, the IDs of all configuration data backups are returned for all MEs.

To allow the NMS to handle a large number of objects, this operation uses an iterator. See the TMF document Overview of Iterator Usage for information on how iterators are used in this interface.

Parameters

Name	Type	Input/ Output	Description
managedElementNameList	NamingAttributesList_T	In	List of names of MEs for which the configuration data IDs are returned. If one or more MEs do not exist, the operation stops.
how_many	long	In	Maximum number of configuration data backup IDs to report in the first batch.
backupList	BackupIdList_T	Out	First batch of configuration data backup IDs.
backupIt	BackupIdIterator_I	Out	Iterator to retrieve the remaining configuration data backup IDs.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when one or more MEs in the list do not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the EMS reaches the maximum number of iterators that it can support.

Compliance
TMF-defined.

3.20.5 activateSoftwareOnME

Synopsis

```
void activateSoftwareOnME(
        in globaldefs::NamingAttributes_T managedElementName)
        raises(globaldefs::ProcessingFailureException)
```

Description
The NMS uses this method to activate downloaded software on an ME.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T managedElementName	In	Name of the ME on which to activate software.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.

Compliance
Prime Optical-defined.

3.20.6 revertSoftwareOnME

Synopsis

```
void revertSoftwareOnME(
        in globaldefs::NamingAttributes_T managedElementName)
        raises(globaldefs::ProcessingFailureException)
```

Description
The NMS uses this method to revert to the older software version on the specified managed element.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T managedElementName	In	Name of the ME on which to revert software back to

Name	Type	Input/ Output	Description
			the older version.

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.

Compliance

Prime Optical-defined.

3.20.7 restoreME

Synopsis

```
void restoreME(
        in globaldefs::NamingAttributes_T managedElementName,
        in string downloadFileName)
        raises(globaldefs::ProcessingFailureException)
```

Description

This asynchronous operation initiates a restore request on the specified ME. The ME configuration file to restore is obtained from the server.

Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which a restore of configuration data is required.
downloadFileName	String	In	Name of the file to restore.

Throws

globaldefs::ProcessingFailureException

EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

Compliance

Prime Optical-defined.

3.20.8 getMERestoreStatus

Synopsis

```
void getMERestoreStatus (
```

```
        in globaldefs::NamingAttributes_T managedElementName,
        out RestoreStatus_T  restoreStatus)
        raises(globaldefs::ProcessingFailureException)
```

Description
The NMS uses this method to obtain the current restore status of an individual ME.
Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which a restore status of configuration data is required.
restoreStatus	RestoreStatus_T	Out	Status of the configuration data restoration.

```
struct RestoreStatus_T
{
  Current_OperationStatus_T opStatus;
  string failureReason;
}
```

RestoreStatus_T is defined as:

- Current_OperationStatus_T opStatus—Indicates the current operational status of the restore.
- string failureReason—A free-form text string that is provided if the opStatus value is COS_Aborted to explain the reason for the abort. (For example, “Comms loss with NE.”)

This data structure identifies the status of the restore operation for a managed element. The failure reason is present if the operation status indicates a failure (in abort state).
Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
EXCPT_INVALID_INPUT - Raised when input parameters are syntactically
incorrect.
EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.
```

Compliance
Prime Optical-defined.

3.20.9 abortMERestore

Synopsis

```
void abortMERestore (
        in globaldefs::NamingAttributes_T managedElementName)
        raises(globaldefs::ProcessingFailureException)
```

Description
This interface aborts the restoration of pending ME configuration data. If the restore operation on the ME is in progress or if there are no pending restore operations, an exception is generated.
Parameters

Name	Type	Input/ Output	Description
managedElementName	NamingAttributes_T	In	Name of the ME for which to stop the restoration of configuration data.

Throws
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NOT_IMPLEMENTED - Raised if the EMS does not support this service.
EXCPT_INVALID_INPUT - Raised when input parameters are syntactically incorrect.
EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.
EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.
EXCPT_NOT_IN_VALID_STATE - Raised when there is no pending or in-progress restoration of configuration data on the ME.
Compliance
Prime Optical-defined.

3.20.10 getAllSupportedMESoftwareVersions

Synopsis

```
void getAllSupportedMESoftwareVersions(in globaldefs::NamingAttributes_T
emsName ,
in unsigned long how_many,
out SupportedMESoftwareVersionList_T versionList,
out SupportedMESoftwareVersionIterator_I versionIt)
raises(globaldefs::ProcessingFailureException);
```

Description
The NMS uses this operation to retrieve a list of potential ME versions (or software versions) that can be managed by the EMS.
Parameters

Name	Type	Input/ Output	Description
emsName	gloaldefs::NamingAttributes_T	In	Name of the EMS for which the supported ME software versions are retrieved.
how_many	unsigned long	In	Maximum number of software versions to report in the first batch.
versionList	SupportedMESoftwareVersionList_T	Out	First batch of supported software versions. (See the definition below.)
versionIt	SupportedMESoftwareVersionIterator_I	Out	Iterator to retrieve the remaining software versions. (See the definition below.)

```
struct SupportedMESoftwareVersion_T
  {
    string meModel;
    string swVersion;
   globaldefs:NVSList_T additionalInfo;
    };
```

where:
- String MEmodel—Model of the managed element (such as Cisco ONS 15454).
- String swVersion—Version of the ME software that the NMS can manage. This string is identical to the version attribute in the ManagedElement_T structure.

- globaldefs::NVSList_T additionalInfo—The EMS communicates to the NMS additional information that is not explicitly modeled.

To support retrieval in batches, a SupportedMESoftwareVersionIterator_I can be defined.

```
 typedef sequence<SupportedMESoftwareVersion_T>
SupportedMESoftwareVersionList_T;

 interface SupportedMESoftwareVersionIterator_I

   {

      boolean next_n (in  unsigned long    how_many,

                      out SupportedMESoftwareVersionList_T neVersionList)

        raises (globaldefs::ProcessingFailureException);


      unsigned long getLength()

        raises (globaldefs::ProcessingFailureException);


      void destroy()

        raises (globaldefs::ProcessingFailureException);

   };
```

Throws

globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators reaches 128.

Compliance

Prime Optical-defined.

3.20.11 downloadMESoftware

Synopsis

```
void downloadMESoftware(

in globaldefs::NamingAttributes_T managedElementName,

in SWDownloadData_T swDownloadData)

 raises(globaldefs::ProcessingFailureException);
```

Description

This interface schedules a software download on a managed element.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T	In	Name of the ME on which to schedule a software download operation.
swDownloadData	SWDownloadData_T	In	Data needed to schedule a software download. (See the following definition.)

```
struct SWDownloadData_T

              {
```

```
    globaldefs::Time_T startTime;
    string userComment;
    FileSource source;
    string downLoadFileName;
    string remoteServerAddress;
    string ftpUserName;
    string ftpPassword;
  };
```

The following table defines each parameter.

Parameters

Name	Type	Input/ Output	Description
startTime	globaldefs::Time_T	In	Start time of the scheduled software download.
userComment	string	In	User comments.
source	FileSource	In	Image source (FTP, TFTP, server, and so on). (See the following definition.)
downLoadFileName	string	In	Name of the image to download.
remoteServerAddress	string	In	Address of the remote server.
ftpUserName	string	In	FTP username.
ftpPassword	string	In	FTP password.

The definition of the file source is:

```
enum FileSource{FROM_LOCAL, FROM_TFTP, FROM_FTP};
```

where:

- FROM_LOCAL—The file source is the same server (that is, the Prime Optical server that you are currently logged into). The package file must be placed in the directory <Prime Optical Server Installation Dir>/image/Cisco_ONS_<NE model Name>.
- FROM _TFTP—The file source is the remote server.
- FROM_FTP—The file source is the remote server.

Currently, GateWay/CORBA does not support software download from the local machine.

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised if the managed element does not exist.
EXCPT_INVALID_INPUT - Raised when invalid input is provided.
EXCPT_NE_COMM_LOSS - Raised when communication to the managed element is lost.
```

Compliance

Prime Optical-defined.

3.20.12 addSupportedMESoftwareVersion

Synopsis

```
void addSupportedMESoftwareVersion(
        in globaldefs::NamingAttributes_T emsName,
        in SupportedMESoftwareVersion_T versionData)
```

```
raises (globaldefs::ProcessingFailureException)
```

Description
This interface allows an NMS to add a new software version to the list of supported ME software versions.

Parameters

Name	Type	Input/ Output	Description
emsName	globaldefs::NamingAttributes_T	In	Name of the EMS on which to add a new supported ME software version.
versionData	SupportedMESoftwareVersion_T	In	Input details for the addition of a new software version.

```
struct SupportedMESoftwareVersion_T
  {
    string meModel;
    string swVersion;
   globaldefs:NVSList_T additionalInfo;
    };
```

The versionData contains all of the necessary fields specific to the EMS with the required information for the addition of a new supported version. All of the fields are supported and have the following descriptions:

- string meModel—The model string of the ME model type for which to add the software version. (For a complete set of strings for all MEs, see 3.9.44 addManagedElement.)
- string swVersion—The new software version to add.
- globaldefs:NVSList_T additionalInfo—The additionalInfo field consists of various parameters that must be specified to the EMS to add the software version. The structure of additionalInfo is as follows:

Name	Value	Description	Mandatory or Optional
ModelName	Any value (used for display purposes).	A version name used for display purposes. This value usually indicates the model type and release.	Optional.
OldSWVersion	A valid, existing software version.	The old software version of the ME model type for which to add the new software version.	Optional.

Throws
globaldefs::ProcessingFailureException

EXCPT_INVALID_INPUT - Raised in case of invalid input. This could be specific to the EMS.

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

Compliance
Prime Optical-defined.

3.20.13 getSWDownloadStatus

Synopsis
```
void getSWDownloadStatus (
```

```
in globaldefs::NamingAttributes_T managedElementName,
out BackupStatus_T  backupStatus)
raises(globaldefs::ProcessingFailureException)
```

Description

The NMS uses this method to obtain the current software download status of an individual ME.

Parameters

Name	Type	Input/ Output	Description
managedElementName	globaldefs::NamingAttributes_T managedElementName	In	Name of the ME for which a software download status is required.
backupStatus	BackupStatus_T	Out	Status of the download.

```
struct BackupStatus_T
{
Current_OperationStatus_T opStatus;
string failureReason;
}
```

BackupStatus_T is defined as:

- Current_OperationStatus_T opStatus—Indicates the current operational status of the software download.
- string failureReason—A free-form text string that is provided if the opStatus value is COS_Aborted to explain the reason for the abort. (For example, “Comms loss with NE.”)

This data structure identifies the status of the software download operation for a managed element. The failure reason is present if the operation status indicates a failure (in abort state).

Throws

```
globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_ENTITY_NOT_FOUND - Raised when the ME does not exist.
```

Compliance

Prime Optical-defined.

4 Using GateWay/CORBA Connectionless Interfaces

This chapter describes each connectionless interface that Cisco Prime Optical supports. This chapter includes the following sections:

4.1 Understanding Key Concepts

This section describes the following key concepts referenced in this chapter:

4.1.1 Ethernet Virtual Circuit

An Ethernet Virtual Circuit (EVC) represents a Carrier Ethernet service and is an entity that provides an end-to-end connection between two or more customer endpoints. The instance of a specific EVC service on the physical interface of each network device through which the EVC passes is called an Ethernet Flow Point (EFP).

The key attributes of the flow point (FP) are:

- FP ID—Generated when the FP is created.
- EVC ID—Associated EVC that the EFP a is part of.
- Encapsulation string—Defines the classification criteria for an incoming packet.
- Forwarding operation—Defines the operation to be applied on frames that belong to the EFP.
- Ingress rewrite operation—Defines the rewrites to be performed on the frames that belong to the EFP before proceeding with the forwarding operation.
- Egress rewrite operation—Defines the rewrites to be performed on the frames being transmitted out of the EFP. For multipoint services, only symmetrical egress operations are supported on the PTF_10GE_4, PT_10GE_4, and PTSA_GE cards. For point-to-point

services, the same result is achieved using various combinations of ingress rewrite operations on User Network Interface (UNI) and Network Node Interface (NNI) ports.

When a packet arrives on an FP, it is processed only if the packet's VLAN tag matches the VLAN tag mentioned in the encapsulation string of this FP. The ingress rewrite operation is applied on the matching Ethernet frames and is forwarded to other FPs participating in this EVC based on forwarding operations.

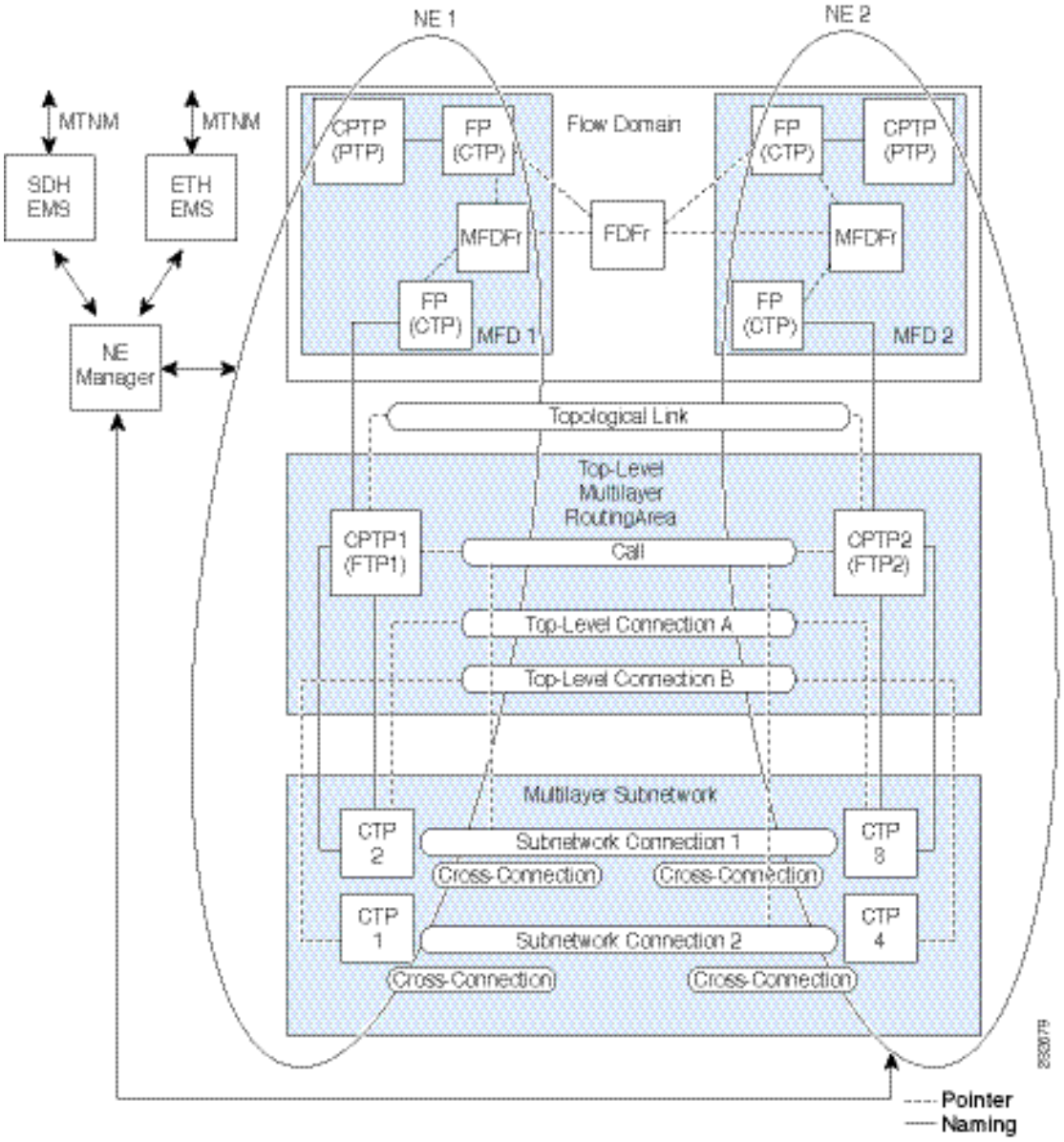
All the FPs within a node, for a particular EVC, are tied together using a bridge domain construct. A bridge domain is an Ethernet broadcast domain internal to a device. The bridge domain enables you to decouple a VLAN from a broadcast domain. The bridge domain number is local to the node and need not be unique across the network for the entire EVC. Different EVC nodes can have the same or a different bridge domain number. However, the bridge domain number is unique for an EVC within a node.

The APIs related to the following entities are described in this chapter:

- Connectionless PTPs (CPTPs)
- Matrix Flow Domains (MFDs)
- Flow Domains (FDs)
- Flow Domain Fragments (FDFrs)

Figure 4-1 displays the relationships among the above-mentioned entities.

Figure 4-1: Relationships Among Entities



All administrative information related to CPTPs, MFDs, and FDs is stored in the Prime Optical server internal database and is available only through this Prime Optical server. The administrative information is not visible in other Prime Optical servers. If any service (Carrier Ethernet EVC or pseudowire) is created outside this scope, the service collides with the administrative organization.

For example, a CTC operator can create a pseudowire (PW) or an EVC that has EFPs. The EFPs are not based on CPTPs associated to the same FD. In this case, services are discovered and associated to the given FD, but a corresponding notification is generated to inform the Prime Optical operator.

For descriptions of EVC provisioning and inventory interfaces, see the following sections:

- 4.5 EVC Provisioning Interfaces
- 4.9 EVC Inventory Interfaces

4.1.2 Connectionless Port Termination Point

A Connectionless Port Termination Point (CPTP) is a potential port capability for connectionless technologies. A CPTP is not a new object, but it is a logical entity of a piece of equipment and supports a connectionless client layer. A CPTP is a grouping of potential FPs on the server layer. The FPs are the clients of a CPTP connected using MFD.

A CPTP is created as a Physical Termination Point (PTP) if the port is an external port and if it does not support encapsulation and link aggregation. CPTP is created as a Floating Termination Point (FTP) if the port is an internal encapsulation port and if it supports encapsulation and link aggregation. A CPTP on an MFD can be either a PTP or an FTP.

ConnectionlessPort is a boolean layered parameter that identifies termination points (TPs) as CPTPs at connectionless layers; for example, Ethernet. For an Ethernet client layer, a CPTP corresponds to an IEEE bridge port, which can either be a UNI or an NNI.

The role played by CPTP for its connectionless client layer is stored in the “PortTPRoleState” layered parameter, which is associated to the PTP/FTP object. The “PortTPRoleState” parameter can assume the following values:

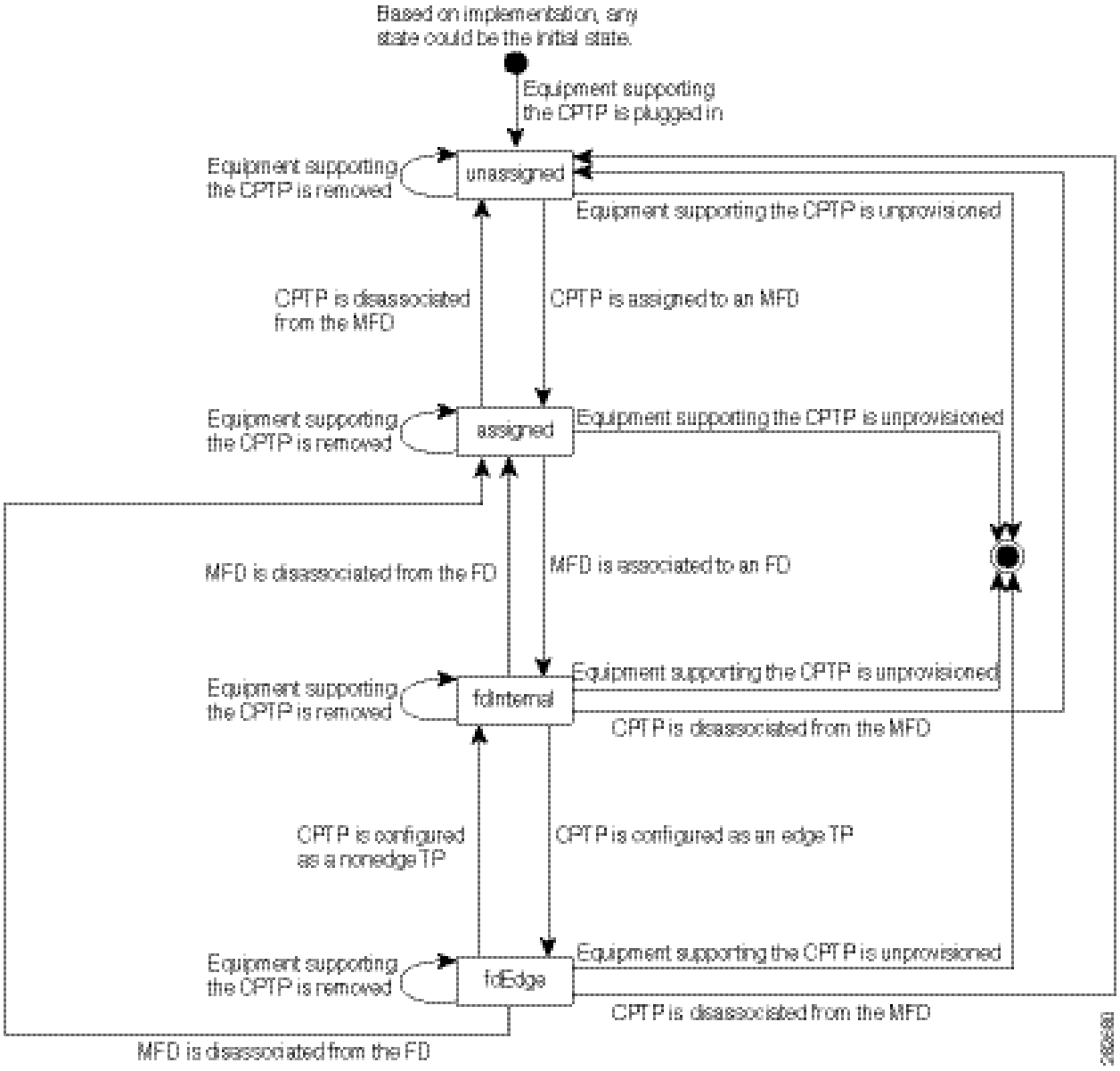
- unassigned CPTP—The initial role of a CPTP is unassigned. If CPTPs are automatically created, they are created as unassigned CPTPs when the equipment that supports the port is plugged into the NE. In this role, the CPTP cannot carry any traffic.
- assigned CPTP—An unassigned CPTP becomes an assigned CPTP when it is associated to an MFD through a management operation. In this role, the CPTP cannot carry any traffic because the MFD is not associated to an FD.
- fdInternal CPTP—An assigned CPTP becomes an fdInternal CPTP when the MFD is associated to an FD. An unassigned CPTP becomes an fdInternal CPTP when it is assigned to an MFD that is already associated to an FD. In this role, the potential client FPs of the CPTP can be used as internal points of the route of a Flow Domain Fragment (FDFr) and can carry traffic.
- fdEdge CPTP—An fdEdge CPTP is the same as an fdInternal CPTP. In this role, the CPTP acts as an edge in the FD.

For descriptions of CPTP provisioning and inventory interfaces, see the following sections:

- 4.5.1 CPTP Provisioning Interfaces
- 4.9.1 CPTP Inventory Interfaces

Figure 4-2 displays the different CPTP states.

Figure 4-2: CPTP States



4.1.3 Matrix Flow Domain

A Matrix Flow Domain (MFD) is a logical entity that is contained within an ME. The ME can contain many MFDs. An MFD consists of a grouping of assigned CPTPs (MFD ports). An MFD is related to only one FD. An MFD is an FD at the lowest level of decomposition that represents the actual minimum decomposition of the hardware.

For descriptions of MFD provisioning and inventory interfaces, see the following sections:

- 4.5.2 MFD Provisioning Interfaces
- 4.9.2 MFD Inventory Interfaces

4.1.4 Flow Domain

The Flow Domain (FD) associates more MFDs (one for each NE) and the server layer TPs of FPs assigned to it. An FD indicates the potential for flow of traffic between a set of points and contains an administrative partitioning of the connectionless network domain.

Connection-oriented subnetworks constitute the widespread transport layer (DWDM, SONET/SDH). Connection-oriented subnetworks are shared by many network applications, but connectionless subnetworks, such as Metro Ethernet, are deployed as smaller islands dedicated to a single network application; for example, a corporate customer site.

FD provisioning capability allows a Network Management System (NMS) to instantiate and to change an FD so it can meet the infrastructure requirement (CPTPs, MFDs) needed to fulfill requests (FDFr setup, tear-down, and modification) received from a service order system. An ME can participate in more than one FD at the same layer rate, but in only one subnetwork.

For descriptions of Flow Domain provisioning and inventory interfaces, see the following sections:

- 4.5.3 Flow Domain Provisioning Interfaces
- 4.9.3 Flow Domain Inventory Interfaces

4.1.5 EVC Flow Domain Fragment

An EVC Flow Domain Fragment (FDFr) is a logical entity that contains a transparent end-to-end connectivity between two or more FPs (at the same connectionless layer) within an FD. The FDFr represents a Virtual Private Network (VPN) for a single customer in the provider network and enables the flow of traffic between FPs.

The server-layer CPTPs of the FPs that are connected through an FDFr must be assigned to MFDs that are associated to the FD that contains the FDFr. If traffic arrives at a point that is a member of an FDFr, it emerges at one or more of the other edge FPs that are members of the same FDFr.

The edge FPs that act as endpoints of the FDFr can be associated with CPTPs connected to customer domains or to other provider domains (of the same or different providers). The VLAN IDs of the FPs of the same FDFr must be equal and in particular must be the VLAN ID of the outermost frame. An FDFr may also support untagged frames or may be unaware of frame tags.

An FDFr is used to model the EVC and has the following attributes:

- Directionality—Either bidirectional or unidirectional. For Ethernet, directionality is always bidirectional.
- Layered transmission parameters—Technology-specific parameters associated with the layer that the FDFr is connecting; for example, Ethernet.
- aEnd TPs—A list of FPs that delimit the FDFr and characterize the edges (entry or exit points). aEnd TPs are clients of the fdEdge CPTPs. For a bidirectional FDFr, this attribute may be combined with zEnd TPs to obtain all the FPs that are associated to the FDFr. For a bidirectional Point-to-Point (PPP) FDFr, it is recommended that you specify one TP in aEnd and the other TP in zEnd. For a multipoint FDFr or a PPP FDFr that may be expanded to multipoint, it is recommended that you specify all the TPs in aEnd.
- zEnd TPs—Represents a list of FPs that delimit the FDFr and characterize the edges (entry or exit points). zEnd TPs are clients of the fdEdge CPTPs. For a bidirectional FDFr, this attribute may be combined with aEnd TPs to obtain all the FPs that are associated to the FDFr.
- Flexible—Indicates whether the FDFr is fixed or flexible. If the FDFr is fixed, the NMS cannot modify or delete it and you cannot add or remove FPs.
- Administrative State—Indicates whether the FDFr is locked or unlocked. If the FDFr is locked, traffic units cannot flow through the FDFr. If the FDFr is unlocked, traffic units are allowed to flow through the FDFr.

- FDFr state—Indicates one of the following values:
  - Active—All MFDFrs and all edge FPs and internal FPs for the FDFr are active in the network.
  - Partial—All parts (MFDFrs or FPs) of the FDFr either were not created during the creation operation or were not deleted during the deletion operation.
- FDFr type—Represents the type of the FDFr:
  - Point-to-point
  - Point-to-multipoint (E-Tree)
  - Multipoint

An FP is a point in a connectionless layer, which represents an association between a CPTP and an FDFr. An FP is modeled as a connection termination point (CTP) and it is either an FDFr endpoint where traffic enters or exits an FDFr or an FDFr internal point used to define the route of an FDFr.
FPs are created as CTP objects when the associated FDFr is created and are deleted when the associated FDFr is deleted. FPs do not exist without an associated FDFr. As a result, only in-use FPs are represented as CTP objects at the interface, and therefore only in-use FPs can be inventoried.
For Ethernet, FPs are always bidirectional. Operations on frames, which either enter or exit an FDFr are defined on the CTP object. The connectionless layered parameters are specified in the layered transmission parameters attribute inherited from the TP object. This attribute represents the technology-specific parameters associated with the different connectionless layers that are supported by the FP. If the NMS does not provide a name for the FP, the Element Management System (EMS) uses the FDFr VLAN ID.
For descriptions of EVC FDFr provisioning and inventory interfaces, see the following sections:

- 4.5.4 EVC FDFr Provisioning Interfaces
- 4.9.4 EVC FDFr Inventory Interfaces

4.1.6 Link Aggregation

Link Aggregation (LAG) is supported using the Link Aggregation Control Protocol (LACP). LACP guarantees the compatibility of both sides of the aggregated link. LACP, which is specified in IEEE 802.3ad, has many attributes and configuration parameters that are handled at the EMS level. LAG support in Multi-Technology Network Management (MTNM) version 3.5 does not deal with these attributes and configuration parameters at the NMS level.
A LAG is represented by an FTP, and the new layer rate defined for it is LR_LAG_Fragment(305). The LAG FTP is the Edge CPTP. A LAG FTP may either be created by the EMS and discovered by the NMS or it can be created by the NMS using the createFTP() operation.
A LAN port, which usually can be an edge CPTP, cannot be an edge CPTP if it is a member of a LAG. Whether created by an EMS or NMS, LAG FTPs act like other fragment TPs. You can configure the maximum number of allowed members using the AllocationMaximum attribute. You can configure a specific number of members using the AllocatedNumber attribute.
If LAGs are created by the EMS, the EMS creates all the potential LAG FTPs that it can handle, each with FragmentServerLayer set to a layer rate at which LAG can be supported and with AllocationMaximum set to the maximum number of members that can be supported for that layer rate.
As with Cisco equipment, an Ethernet LAG port can aggregate client Ethernet ports of different ME cards (for example, PTSA_GE). The corresponding FTP is logically positioned on the unique shelf of the ME itself. As a result, the FTP name does not contain any reference to slots:

```
name="EMS";value="CompanyName/EMSname"

name="ManagedElement";value="ManagedElementName"

name="FTP";value="/rack=1/shelf=1/port=nn"
```

For descriptions of LAG provisioning and inventory interfaces, see the following sections:

- 4.5.5 Link Aggregation Provisioning Interfaces

- 4.9.5 Link Aggregation Inventory Interfaces

4.1.7 MPLS and MPLS-TP

Multiprotocol Label Switching (MPLS) allows the forwarding of packets based on labels. In a normal IP network, the packets are switched based on the destination IP address. In an MPLS network, the packets are switched based on labels.

In an MPLS network, the labels can be distributed using three different protocols:

- Label Distribution Protocol (LDP)
- Resource Reservation Protocol-Traffic Engineering (RSVP-TE)
- Border Gateway Protocol (BGP)

BGP is a routing protocol used in big IP networks where the labeling layer is mostly used to implement Layer 3 VPNs. After the labels are distributed within a network, the packets are labeled and forwarded based on labels. The advantages of labeled networks are:

- Protocol agnostic—Can transport any kind of protocol; for example, IP, ATM, Any Transport over MPLS (AToM).
- High scalability.
- Traffic Engineering—Load balancing and automatic adaption to link changes.

Multiprotocol Label Switching-Transport Profile (MPLS-TP) is a carrier-grade packet transport technology that enables the move from SONET and SDH time-division multiplexing (TDM) to packet switching. MPLS-TP enables MPLS to be deployed in a transport network and to operate similarly to existing transport technologies. MPLS-TP enables MPLS to support packet transport services with a degree of predictability that is similar to that of the existing transport networks.

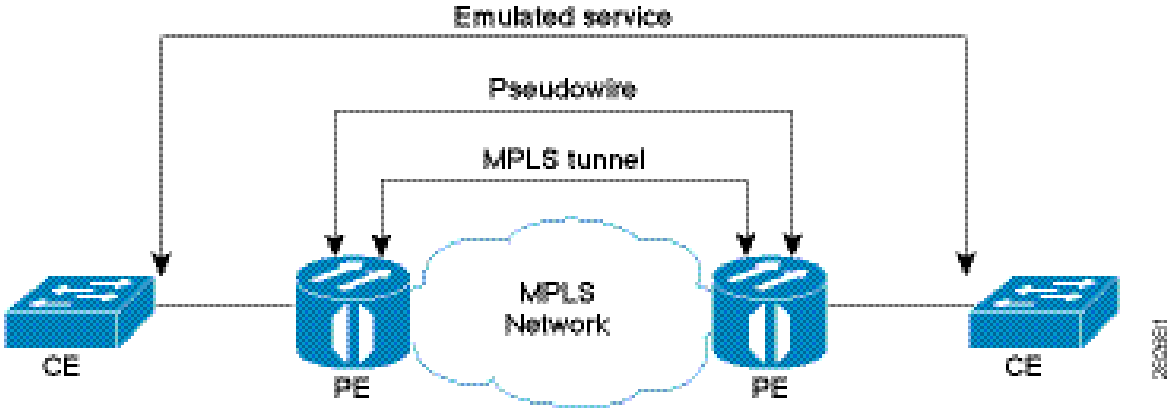
The goal of MPLS-TP is to provide connection-oriented transport for packet and TDM services over optical networks leveraging the widely deployed MPLS technology. Operations, administration, and maintenance (OAM) and resiliency features are defined and implemented in MPLS-TP to ensure:

- Scalable operations
- High availability
- Performance monitoring
- Multidomain support
- Carrier-grade packet transport networks

MPLS-TP can be carried over the existing transport network infrastructure. MPLS-TP defines an MPLS profile targeted at transport applications and networks. This profile specifies the MPLS characteristics and extensions required to meet the transport requirements.

PTF_10GE_4, TP_10GE_4, and PTSA_GE equipment is mainly dedicated to forwarding Ethernet frames from customer networks, thus focusing only on AToM. The method used to transport such a frame is called pseudowire. Pseudowire is the emulation of a native service over the MPLS network. For managing PTF_10GE_4, TP_10GE_4, and PTSA_GE equipment, pseudowire and emulated (EVC-based) service are in a one-to-one relationship. Pseudowire does not provide bundling of additional EVCs. As shown in Figure 4-3, pseudowires and emulated services are represented by only one service or FDFr.

Figure 4-3: MPLS-TP Network



For descriptions of MPLS-TP provisioning and inventory interfaces, see the following sections:

- 4.6 MPLS-TP Provisioning Interfaces
- 4.10 MPLS-TP Inventory Interfaces

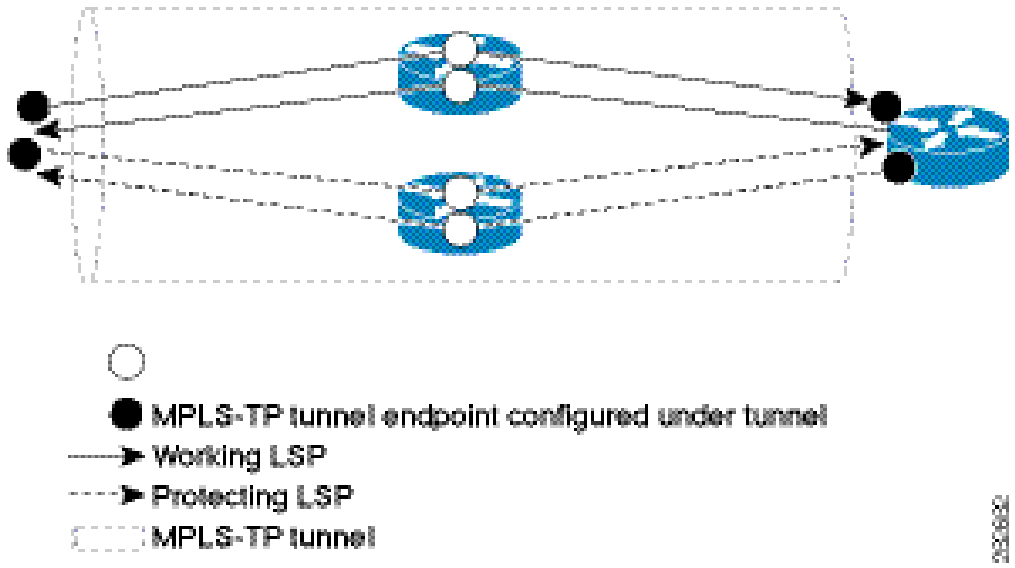
4.1.8 MPLS-TP Tunnel

The MPLS-TP entity is characterized as follows:

- Has two endpoints.
- Has one or two bidirectional Label Switched Path (LSP) pairs between the endpoints. If there are two pairs, then one is the working path and the second is the protected path.
- The bidirectional LSPs in a pair are congruent. Bidirectional LSPs follow the same path in both directions.
- The tunnel can be configured with bandwidth. However, unlike MPLS-TE, there is no bandwidth reservation or preemption on the NE. Use the show command to view the configured bandwidth. Any bandwidth reservation must be handled within the NMS.
- Bidirectional Forwarding Detection (BFD) can be configured to monitor any of the bidirectional LSP pairs.
- MPLS-TP tunnel is used to carry pseudowires. Pseudowires or PW segments are explicitly configured to use a particular MPLS-TP tunnel using a given PW-CLASS configuration.

Figure 4-4 shows a typical MPLS-TP Tunnel.

Figure 4-4: MPLS-TP Tunnel



For descriptions of MPLS-TP Tunnel provisioning and inventory interfaces, see the following sections:

- 4.6.1 MPLS-TP Tunnel Provisioning Interfaces
- 4.10.1 MPLS-TP Tunnel Inventory Interfaces

4.1.9 MPLS-TP LSP

The LSP circuit provides the TP tunnel circuit with the path used for routing traffic. It also provides the distribution of labels along the defined route. Both TP tunnels and LSP subnetwork connections (SNCs) have the following attributes in common:

- Layer rate—LR_MPLS (165)
- Name
- Service ID

The information that discriminates the LSP from the TP tunnel is the Unique ID number, which serves also to distinguish one LSP from another in the case of a protected TP tunnel. Before you create the second LSP in a protected TP tunnel, you must wait until the TP tunnel and the first LSP are discovered. This is necessary in order to exclude the links used by the first LSP.

For descriptions of MPLS-TP LSP provisioning and inventory interfaces, see the following sections:

- 4.6.2 MPLS-TP LSP Provisioning Interfaces
- 4.10.2 MPLS-TP LSP Inventory Interfaces

4.1.10 Pseudowire

A pseudowire (PW) is an emulation of a Layer 2 point-to-point, connection-oriented service over a packet-switching network (PSN). Pseudowire is the technique used to transport these types of frames. It is the emulation of a native service over the MPLS network.

LDP, MPLS-TE, and MPLS-TP are the basement for the pseudowire technology used to transport any kind of payload over a structured network (AToM). Currently, MPLS-TP has only one client network layer, which is pseudowire. The only way to route traffic into an MPLS-TP tunnel is to configure it as the preferred path of a pseudowire.

The kinds of pseudowire are:

- Interface based
- EVC based

- Mixed configuration

Each pseudowire can carry only one EVC service and is represented and exported directly by one FDFr. From the Network Circuit Provisioning (NCP) level, the pseudowire is represented by only one service. For descriptions of pseudowire provisioning and inventory interfaces, see the following sections:

- 4.7 Pseudowire Provisioning Interfaces
- 4.11 Pseudowire Inventory Interfaces

4.1.11 Pseudowire Flow Domain Fragment

An EVC-based pseudowire with an attachment circuit (AC) type of VLAN-based or port-based is modeled as an FDFr. In this model, an EVC containing the pseudowire and an FDFr on top the EVC containing the EVC do not exist; only an FDFr exists.
Ethernet FPs are represented by CTPs and the naming rule depends on the properties of the MFD that connects it.

The following table lists the MFD types and the corresponding value.

MFD Type	Value	Comments
VLAN unaware	/eth=1	Only one FP is present.
VLAN aware with single VLAN tag	/ethvid=n	Can be used either for Customer VLAN ID (C-VID) or Service VLAN ID (S-VID) in MFDs that only process a single VLAN tag.
VLAN aware with one or two VLAN tags	[/ethsvid=n][/ethcvid=m]	Either S-VID or C-VID or both maybe present. The square brackets (“[ ]”) indicate options.
Proprietary	/eth=P<string>	—

For descriptions of PW FDFr provisioning and inventory interfaces, see the following sections:

- 4.7.1 Pseudowire FDFr Provisioning Interfaces
- 4.11.1 Pseudowire FDFr Inventory Interfaces

4.2 Interface Managers

The following interface managers support APIs:

- FlowDomain
- TrafficConditioningProfile

The following table lists the supported APIs and the corresponding interface manager.

Table 4-1: Interface Managers and APIs

Interface Manager	API	New API	TMF Extension
TrafficConditioningProfile	createTCProfile	Yes	No
	deleteTCProfile	Yes	No
	modifyTCProfile	Yes	No
	getAllTCProfiles	Yes	No
	getTCProfile	Yes	No
FlowDomain	createMFD	Yes	No
	assignCPTPsToMFD	Yes	No
	unassignCPTPsFromMFD	Yes	No
	deleteMFD	Yes	No
	modifyMFD	Yes	No
	createFlowDomain	Yes	No
	deleteFlowDomain	Yes	No

Interface Manager	API	New API	TMF Extension
	modifyFlowDomain	Yes	No
	associateMFDsWithFlowDomain	Yes	No
	deAssociateMFDsFromFlowDomain	Yes	No
	createAndActivateFDFr	Yes	No
	deactivateAndDeleteFDFr	Yes	No
	modifyFDFr	Yes	No
	createFTP	Yes	No
	deleteFTP	Yes	No
	getAllCPTPs	Yes	No
	getAllAssignedCPTPs	Yes	No
	getAllAssociatedMFDs	Yes	No
	getAllSupportedMFDs	Yes	No
	getMFD	Yes	No
	getAssigningMFD	Yes	No
	getAllFlowDomains	Yes	No
	getFlowDomainsByUserLabel	Yes	No
	getFlowDomain	Yes	No
	getAllTopologicalLinksOfFD	Yes	No
	getAssociatingFD	Yes	No
	getAllFDFrs	Yes	No
	getFDFrsWithTP	Yes	No
	getFDFrsByUserLabel	Yes	No
	getFDFrNamesWithTP	Yes	Yes
	getAllFDFrNames	Yes	Yes
	getFDFr	Yes	No
	getFDFrRoute	Yes	No
	getTransmissionParams	Yes	No
	ConfigConnLessInfos	Yes	Yes
	DeleteAllConnLessInfos	Yes	Yes
Common	setAdditionalInfo	No	No
ManagedElement	provisionEquipment	No	No
	unprovisionEquipment	No	No
	getAllPTPs	No	No
	getAllFTPs	No	No
	getAllFTPNames	No	No
	getTP	No	No
	getAllCrossConnections	No	No
	getManagedElement	No	No
MultilayerSubnetwork	getTopologicalLink	No	No
	getAllTopologicalLinks	No	No
	getAllTopologicalLinkNames	No	No
	getAllTopLevelTopologicalLinks	No	No
	getAllTopLevelTopologicalLinkNames	No	No
	getTopLevelTopologicalLink	No	No
	createAndActivateSNC	No	No
	modifySNC	No	No
	getAllSNCsWithTP	No	No
	getAllSubnetworkConnectionsWithTP	No	No

Interface Manager	API	New API	TMF Extension
	getAllSubnetworkConnections	No	No
	getAllSubnetworkConnectionNames	No	No
	getSNC	No	No
	getSNCbyUserLabel	No	No

4.3 Entity Characteristics

This section describes the following characteristics for the entities managed by the GateWay/CORBA interface:

- 4.3.1 Entity Naming
- 4.3.2 Entity Iterators
- 4.3.3 Entity Notifications

4.3.1 Entity Naming

The following table lists the naming hierarchy for the entities managed by the GateWay/CORBA interface.

Table 4-2: Entity and Naming Hierarchy

Entity	Naming Hierarchy
EMS	name="EMS";value="CompanyName/EMSname"
Subnetwork	name="EMS";value="CompanyName/EMSname" name="MultiLayerSubnetwork";value="SubnetworkName"
SubnetworkConnection	name="EMS";value="CompanyName/EMSname" name="MultiLayerSubnetwork";value="SubnetworkName" name="SubnetworkConnection";value="SubnetworkConnectionName"
ManagedElement	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName"
TopologicalLink	name="EMS";value="CompanyName/EMSname" name="TopologicalLink";value="TopologicalLinkName"
PTP	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName" name="PTP";value="PTPName"
FTP	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName" name="FTP";value="FTPName"

Entity	Naming Hierarchy
CTP	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName" name="PTP";value="PTPName" name="CTP";value="CTPName"
EquipmentHolder	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName" name="EquipmentHolder";value="EquipmentHolderName"
Equipment	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName" name="EquipmentHolder";value="EquipmentHolderName" name="Equipment";value="EquipmentName"
FlowDomain	name="EMS";value="CompanyName/EMSname" name="FlowDomain";value=" FlowDomainName"
MatrixFlowDomain	name="EMS";value="CompanyName/EMSname" name="ManagedElement";value="ManagedElementName"name="MatrixFlowDomain";value=MatrixFlowDomainName”
FlowDomainFragment	name="EMS";value="CompanyName/EMSname" name="FlowDomain";value="FlowDomainName" name="FlowDomainFragment";value="FlowDomainFragmentName"
TrafficConditioningProfile	name="EMS";value="CompanyName/EMSname" name="tcprofile";value="TCProfileName"

4.3.2 Entity Iterators

The NMS uses iterators to retrieve large amounts of data. Iterators provide a mechanism for retrieving data in batches. The NMS specifies the ideal size of a batch. When the NMS requests a list of objects, it specifies the maximum number of objects it can handle in the first reply. The number sent by the EMS can be less than this number.

The EMS gets a snapshot of the current objects to be returned. The response contains a list with the specified number of objects and a reference to an iterator. The NMS uses this iterator to access other objects in the snapshot.

Synopsis

```
getAllObjectName(
in unsigned long how_many,
out objectList list,
out iterator iteratorReference)
```

```
raises(globaldefs::ProcessingFailureException);
```

Description

The how_many parameter determines the maximum number of response entries in the list output parameter.

The iteratorReference parameter provides access to the remaining objects, if any.

- If the list contains the complete set of ObjectName objects, then the iteratorReference is a reference to a CORBA::Object::_nil object.
- If there are more objects, the list output parameter contains the first batch of objects known to the EMS and the iteratorReference parameter provides access to the other objects.
- If you specify 0 in the how_many parameter, no objects are returned in the list and all objects must be retrieved from the iteratorReference parameter.

The element management layer (EML) can return fewer objects than specified in the how_many value under either of the following conditions:

- Fewer objects than the number specified in the how_many value actually exist.
- The EML determines that the how_many value exceeds the server's stated performance restrictions.

For example, the EML can have 100 objects to return, the network management layer (NML) may request 50 through the how_many parameter and the EML may return 20 along with an iteratorReference.

The NMS uses the iteratorReference parameter to request further batches. This reference can be used to query how many objects can be returned by the getLength method. The NMS can subsequently start retrieving or removing the iterator. If all objects are specified in the list, it is a reference to a CORBA::Object::_nil object.

Each entry is returned only once, so you must use the next_n method to retrieve the additional entries in the selected set.

The EML may not get the total number of elements returned by the iterator in the following instances:

- The EML has to fetch all of the data that has been requested and then count the number of elements. This may not produce the desired results.
- The EML may be able to get a count of the number of elements, but due to concurrent modifications, the actual number may vary over the time of the iteration.

The interface client can determine when all data has been collected by repeatedly fetching chunks of data until the iterator indicates that no further data remains. If the EMS is unable to determine the length, the getLength method returns the EXCPT_CAPACITY_EXCEEDED exception.

The EXCPT_TOO_MANY_OPEN_ITERATORS exception is returned if the EMS has reached an implementation limitation. An EMS supports a minimum of 10 iterators.

In this interface, the iterators have the following operations:

- getLength—Returns the total number of elements in the iterator. For example, if the EML has 100 objects and 10 are returned in the initial list, the getLength method should return 90, regardless of how many objects are or are not retrieved using the next_n method. The EXCPT_CAPACITY_EXCEEDED exception is returned if the EMS cannot efficiently provide a value for the number of elements on this occasion. The next_n method may still be performed.
- next_n—Returns no more than the “n” entries, but it may return fewer. Returns true if there are more entries to be returned. If there are fewer entries, it returns false, removes the iterator, and releases the memory being used. The EML may choose to limit the number of objects returned in a single request to prevent performance issues.
- destroy—If you decide not to access the remaining objects, you can invoke the destroy operation to delete the iterator object.

4.3.3 Entity Notifications

For all entities introduced up to and including MTNM version 2.0, the ObjectType (filterable) field within the notification structure was used to identify the object types. As backward compatibility has to be supported, you cannot extend the MTNM version 2.0 enumeration type ObjectType_T to include new object types.

In MTNM version 3.0 and onward, the objectType OT_AID (alarm identifier) is used as an escape value to represent new objects being managed, in conjunction with the new filterable field objectTypeQualifier that contains the real object type:

- OT_FLOW_DOMAIN—Flow Domain.
- OT_FLOW_DOMAIN_FRAGMENT—Flow Domain Fragment.
- OT_MATRIX_FLOW_DOMAIN—Matrix Flow Domain.
- OT_TRANSMISSION_DESCRIPTOR—Transmission Descriptor.
- OT_TRAFFIC_CONDITIONING_PROFILE—Traffic Conditioning Profile.
- OT_FLOATING_TERMINATION_POINT—LAG FTP.
- OT_PW_CLASS—Pseudowire class. Any attribute related to a pseudowire class.
- OT_BFD_TEMPLATE—BFD template. Any attribute related to a BFD template.
- (empty string) indicates a proper OT_AID alarm identifier (used to represent the EMS object types that are not modeled, but can emit alarms).

MFD Examples

```
*** Thu Dec 02 12:47:57 CET 2010 *** Received
notification.**************************


Fixed Header        Domain          = tmf_mtnm
                              Type                = NT_OBJECT_CREATION
Variable Header   Event Reliability  = 0
                              Priority            = 0
                             Timeout             = 864000000000
Filterable Data
       [0] notificationId     = OCE-1
                     [1] objectName         = [0] EMS=Cisco
Systems/PRIMEOPTICAL; [1] ManagedElement=15454-ANSI-68-7; [2]
MatrixFlowDomain=GIANCARLO_MFD
                     [2] objectType         = OT_AID
                     [2] objectTypeQualifier = MatrixFlowDomain
                     [3] emsTime            = 20101202124109.0Z
                     [4] neTime             =
                     [5] edgePoint          = false
                     [6] isPropagated       = false


**************************************************** **************************
*** Thu Dec 09 15:12:19 CET 2010 *** Received
notification.**************************


Fixed Header       Domain              = tmf_mtnm
```

```
                              Type                  = NT_OBJECT_DELETION
Variable Header   Event Reliability  = 0
                        Priority           = 0
                              Timeout            = 864000000000


Filterable Data
       [0] notificationId     = ODE-1
                      [1] objectName         = [0] EMS=Cisco
Systems/PRIMEOPTICAL; [1] ManagedElement=15454-ANSI-68-7; [2]
MatrixFlowDomain=GIANCARLO_MFD
                      [2] objectType         = OT_AID
                      [2] objectTypeQualifier = MatrixFlowDomain
                      [3] emsTime            = 20101209150546.0Z
                      [4] neTime             =
                      [5] edgePoint          = false
                      [6] isPropagated       = false
```

FTP Examples

```
*********************************************************


Fixed Header              Domain             = tmf_mtnm
                          Type               = NT_OBJECT_DELETION
Variable Header           Event Reliability  = 0
                          Priority           = 0
                          Timeout            = 864000000000


Filterable Data      [0] notificationId     = ODE-1
                      [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
FTP=/rack=1/shelf=1/port=1
                      [2] objectType         = OT_AID
                      [2] objectTypeQualifier = OT_FLOATING_TERMINATION_POINT
                      [3] emsTime            = 20110608152520.0Z
                      [4] neTime             =
                      [5] edgePoint          = false
                      [6] isPropagated       = false
Remainder of body:
null


*********************************************************
Fixed Header              Domain             = tmf_mtnm
```

```
                          Type                = NT_OBJECT_CREATION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = OCE-1
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
FTP=/rack=1/shelf=1/port=1
                      [2] objectType          = OT_AID
                      [2] objectTypeQualifier = OT_FLOATING_TERMINATION_POINT
                      [3] emsTime             = 20110608152436.0Z
                      [4] neTime              =
                      [5] edgePoint           = false
                      [6] isPropagated        = false
*********************************************************

Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = AVC-4
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
FTP=/rack=1/shelf=1/port=1
                      [2] objectType          = OT_AID
                      [3] objectTypeQualifier = OT_FLOATING_TERMINATION_POINT
                      [3] emsTime             = 20110804133021.0Z
                      [4] neTime              =
                      [5] edgePoint           = false
                      [6] attributeList       =
                       [5] isPropagated        = false

*********************************************************
```

PW Class Examples

```
***************************

Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_CREATION
```

```
Variable Header          Event Reliability  = 0
                         Priority           = 0
                         Timeout            = 864000000000

Filterable Data      [0] notificationId     = OCE-1
                     [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] objectType         = OT_AID
                     [2] objectTypeQualifier = OT_PW_CLASS
                     [3] emsTime            = 20110627155440.0Z
                     [4] neTime             =
                     [5] edgePoint          = false
                     [6] isPropagated       = false
Remainder of body:
null

*********************************************************

Fixed Header             Domain             = tmf_mtnm
                         Type               = NT_OBJECT_DELETION
Variable Header          Event Reliability  = 0
                         Priority           = 0
                         Timeout            = 864000000000

Filterable Data      [0] notificationId     = ODE-1
                     [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] objectType         = OT_AID
                     [2] objectTypeQualifier = OT_PW_CLASS
                     [3] emsTime            = 20110627155455.0Z
                     [4] neTime             =
                     [5] edgePoint          = false
                     [6] isPropagated       = false
Remainder of body:
null

*********************************************************
```

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = AVC-3
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                      [2] objectType          = OT_AID
                      [3] objectTypeQualifier = OT_PW_CLASS
                      [3] emsTime             = 20110804132925.0Z
                      [4] neTime              =
                      [5] edgePoint           = false
                      [6] attributeList       =
                       [5] isPropagated        = false

*********************************************************
```

BFD Template Examples

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_CREATION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = OCE-2
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                      [2] objectType          = OT_AID
                      [2] objectTypeQualifier = OT_BFD_TEMPLATE
                      [3] emsTime             = 20110627155508.0Z
                      [4] neTime              =
                      [5] edgePoint           = false
                      [6] isPropagated        = false
Remainder of body:
null

*********************************************************
```

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_DELETION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data      [0] notificationId       = ODE-2
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] objectType           = OT_AID
                     [2] objectTypeQualifier = OT_BFD_TEMPLATE
                     [3] emsTime              = 20110627155516.0Z
                     [4] neTime               =
                     [5] edgePoint            = false
                     [6] isPropagated         = false
Remainder of body:
null

*********************************************************

Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data      [0] notificationId       = AVC-2
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] objectType           = OT_AID
                     [3] objectTypeQualifier = OT_BFD_TEMPLATE
                     [3] emsTime              = 20110804132846.0Z
                     [4] neTime               =
                     [5] edgePoint            = false
                     [6] attributeList        =
                      [5] isPropagated         = false
```

```
*********************************************************
```

Traffic Conditioning Profile Examples

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_CREATION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = OCE-3
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=class-map classmap01
                      [2] objectType          = OT_AID
                      [2] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                      [3] emsTime             = 20110802124005.0Z
                      [4] neTime              =
                      [5] edgePoint           = false
                      [6] isPropagated        = false
Remainder of body:
null

*********************************************************


*** Tue Aug 02 14:42:28 CEST 2011 *** Received
notification.***************************

Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId       = AVC-5
                      [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=policy-map policymap01
                      [2] objectType           = OT_AID
                      [3] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                      [4] emsTime              = 20110802124228.0Z
                      [5] neTime               =
                      [6] edgePoint            = false
```

```
                    [7] attributeList       = [0]userLabel=policymap01mod
                     [8] isPropagated        = false
Remainder of body:
null

*******************************************************

Fixed Header            Domain              = tmf_mtnm
                        Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header         Event Reliability   = 0
                        Priority            = 0
                        Timeout             = 864000000000

Filterable Data     [0] notificationId      = AVC-8
                    [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=policy-map policymap01mod
                    [2] objectType          = OT_AID
                    [3] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                    [4] emsTime             = 20110802124632.0Z
                    [5] neTime              =
                    [6] edgePoint           = false
                    [7] attributeList       = [0]userLabel=policymap01
[1]additionalInfo=additionaInfo changed
                     [8] isPropagated        = false
Remainder of body:
null

*******************************************************
Fixed Header            Domain              = tmf_mtnm
                        Type                = NT_OBJECT_DELETION
Variable Header         Event Reliability   = 0
                        Priority            = 0
                        Timeout             = 864000000000

Filterable Data     [0] notificationId      = ODE-8
                    [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=policy-map policymap01
                    [2] objectType          = OT_AID
                    [2] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                    [3] emsTime             = 20110802130447.0Z
```

```
                        [4] neTime              =
                        [5] edgePoint           = false
                        [6] isPropagated        = false
Remainder of body:
null

*******************************************************
```

4.3.3.1 Attribute Value Change for PT System Events

An attribute value change (AVC) notification is sent to all GateWay/CORBA clients when an attribute referring to the PT System entity object is modified.

The following attributes are managed by the PT System:

- OperationMode
- nodeId
- routerIdIpAddr
- routerIdMask
- staticMinLabel
- staticMaxLabel
- dynamicMinLabel
- dynamicMaxLabel
- PtSystemState

When an AVC notification is sent, the notification refers to all the attributes managed by GateWay/CORBA. The notification is sent even if the modified attribute is not currently managed by GateWay/CORBA.

Example

```
*** Thu Aug 25 17:21:42 CEST 2011 *** Received
notification.***************************


Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000


Filterable Data       [0] notificationId       = AVC-6
                      [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117
                      [2] objectType           = OT_MANAGED_ELEMENT
                      [3] objectTypeQualifier =
                      [4] emsTime              = 20110825152142.0Z
                      [5] neTime               =
                      [6] edgePoint            = false
                      [7] attributeList        = [0]OperationMode=CTC Mode
[1]nodeId=1.1.1.10 [2]routerIdIpAddr=1.1.1.1 [3]routerIdMask=255.255.255.255
```

```
[4]staticMinLabel=20 [5]staticMaxLabel=4000 [6]dynamicMinLabel=4001
[7]dynamicMaxLabel=8000 [8]PtSystemState=Ready
                        [8] isPropagated        = false
Remainder of body:
null
*******************************************************
```

4.4 Supported Provisioning Interfaces

The following table lists the provisioning interfaces supported in Prime Optical 10.7.

Table 5-3: Provisioning Entities and Interfaces

Entity	Interface
EVC	
4.5.1 CPTP Provisioning Interfaces	4.5.1.1 provisionEquipment
	4.5.1.2 unprovisionEquipment
	4.5.1.3 setAdditionalInfo
	4.5.1.4 ConfigConnLessInfos
	4.5.1.5 DeleteAllConnLessInfos
4.5.2 MFD Provisioning Interfaces	4.5.2.1 createMFD
	4.5.2.2 assignCPTPsToMFD
	4.5.2.3 unassignCPTPsFromMFD
	4.5.2.4 deleteMFD
	4.5.2.5 modifyMFD
4.5.3 Flow Domain Provisioning Interfaces	4.5.3.1 createFlowDomain
	4.5.3.2 deleteFlowDomain
	4.5.3.3 modifyFlowDomain
	4.5.3.4 associateMFDsWithFlowDomain
	4.5.3.4 deAssociateMFDsWithFlowDomain
4.5.4 EVC FDFr Provisioning Interfaces	4.5.4.1 createAndActivateFDFr
	4.5.4.2 deactivateAndDeleteFDFr
	4.5.4.3 modifyFDFr
4.5.5 Link Aggregation Provisioning Interfaces	4.5.5.1 createFTP
	4.5.5.2 deleteFTP

Entity	Interface
4.5.6 Traffic Conditioning Profile Provisioning Interfaces	4.5.6.1 createTCProfile
	4.5.6.2 deleteTCProfile
	4.5.6.3 modifyTCProfile
MPLS-TP	
4.6.1 MPLS-TP Tunnel Provisioning Interfaces	4.6.1.1 createAndActivateSNC
	4.6.1.2 deactivateAndDeleteSNC
	4.6.1.3 modifySNC
4.6.2 MPLS-TP LSP Provisioning Interfaces	4.6.2.1 createAndActivateSNC
	4.6.2.2 deactivateAndDeleteSNC
	4.6.2.3 modifySNC
4.6.3 MPLS-TP Protection (Provisioning)	—
4.6.4 BFD Template, Node ID, and Label Range Provisioning Interface	4.6.4.1 setAdditionalInfo
Pseudowire	
4.7.1 Pseudowire FDFr Provisioning Interfaces	4.7.1.1 createAndActivateFDFr
	4.7.1.2 deactivateAndDeleteFDFr
	4.7.1.3 modifyFDFr
4.7.2 Pseudowire Mixed Configuration (Provisioning)	—
4.7.3 Pseudowire Class and Loopback Provisioning Interface	4.7.3.1 setAdditionalInfo
4.7.4 Pseudowire QoS Provisioning Interface	4.7.4.1 TCProfileMgr::setAdditionalInfo

4.5 EVC Provisioning Interfaces

This section describes the following EVC provisioning interfaces:

- 4.5.1 CPTP Provisioning Interfaces
- 4.5.2 MFD Provisioning Interfaces
- 4.5.3 Flow Domain Provisioning Interfaces
- 4.5.4 EVC FDFr Provisioning Interfaces
- 4.5.5 Link Aggregation Provisioning Interfaces
- 4.5.6 Traffic Conditioning Profile Provisioning Interfaces

4.5.1 CPTP Provisioning Interfaces

This section describes the following interfaces:

- 4.5.1.1 provisionEquipment
- 4.5.1.2 unprovisionEquipment

4.5.1.1 provisionEquipment

Synopsis

```
public void provisionEquipment(EQTCreateData_T equipmentCreateData,
                               Equipment_THolder createdEquipment)
      throws ProcessingFailureException
(on EquipmentInventoryMgrImpl)
```

Description

This interface allows you to provision CPTPs. You can provision PTPs by specifying the ConnectionlessPort additional parameter. If a PTP has been recently characterized as connectionless, you cannot set the PortTpRoleState parameter with this interface because the CPTP state machine does not allow it.

4.5.1.2 unprovisionEquipment

Synopsis

```
public void unprovisionEquipment(NameAndStringValue_T[] equipmentName)
      throws ProcessingFailureException
(on EquipmentInventory)
```

Description

This interface allows you to unprovision a PTP. If the admin state is inactive and all other conditions are met, unprovisioning is successful and the corresponding CPTP entity is deleted.

4.5.1.3 setAdditionalInfo

Synopsis

```
public void setAdditionalInfo(NameAndStringValue_T[] objectName,
                              NVSList_THolder additionalInfo)
      throws ProcessingFailureException
```

Description

This interface allows you to specify the following additional parameters:

- ConnectionlessPort
- PortTpRoleState

The PortTpRoleState parameter can be modified only if the CPTP state machine allows it.
The following table describes the parameters.

TP Parameter Name	TP Type	Layers	Valid Values	AVC	Description
ConnectionlessPort	• PTP • FTP	All layers	• true • false	Yes	Identifies the TP as a port at a connectionless matrix, capable of supporting a connectionless client layer; for example, an external port or internal encapsulation port. The clients of a ConnectionlessPort (FPs) are

TP Parameter Name	TP Type	Layers	Valid Values	AVC	Description
					connected through the matrix.
PortTPRoleState	• PTP • FTP	All connectionless layers	• unassigned • assigned • fdInternal • fdEdge	Yes	Identifies the kind of role that a CPTP is playing for its connectionless client layer: • unassigned—The initial role of a CPTP is unassigned. In this role, the CPTP cannot carry any traffic. • assigned—An unassigned CPTP becomes an assigned CPTP when it is associated to an MFD through a management operation. In this role, the CPTP cannot carry any traffic because the MFD is not associated to an FD. • fdInternal—An assigned CPTP becomes an fdInternal CPTP when the MFD is associated to an FD. An unassigned CPTP becomes an fdInternal CPTP when it is assigned to an MFD that is already associated to an FD. In this role, the potential client FPs of the CPTP can be used as internal points of the route of an FDFr and can carry traffic. • fdEdge—An assigned CPTP becomes an fdEdge CPTP through a management operation. In this role, the potential client FPs of the port can be used as entry or exit points of an FDFr and can carry traffic. ✎ The MFD must already be associated to the FD.

4.5.1.4 ConfigConnLessInfos

Synopsis

```
void ConfigConnLessInfos(NmsSession_I session)
        throws ProcessingFailureException
```

Description

This interface allows you to automatically populate all FDs, MFDs, and CPTPs based on the current EMS inventory.

The EMS does the following:

1. Creates an FD for each network partition.
2. Within each FD, creates an MFD for each ME already assigned to the FD.
3. Within each MFD, creates a CPTP for each Ethernet port and for each port channel (link aggregation). Each CPTP (PTP and FTP) is automatically assigned to the corresponding MFD and assumes the fdEdge CPTP role state. Port channel FTPs are also populated.

During discovery or provisioning, this interface allows you to manage the FDFr that represents pseudowires and EVCs. If this method is called, only the new MEs that were previously unavailable in the EMS are analyzed and contribute to the connectionless EMS information. In an administrative organization, all the APIs that allow you to create, delete, modify, assign, and unassign CPTPs, MFDs, and FDs are available.

For more information about the completion of the process started by this API, see 4.5.1.6 NMS Session Callback.

4.5.1.5 DeleteAllConnLessInfos

Synopsis

```
void DeleteAllConnLessInfos(NmsSession_I session)
      throws ProcessingFailureException
```

Description

This interface allows you to delete all the FDs, MFDs, and CPTPs currently stored in the EMS. You may use this interface at any time to reorganize the network from scratch.

For more information about the completion of the process started by this API, see 4.5.1.6 NMS Session Callback.

4.5.1.6 NMS Session Callback

Synopsis

```
public interface NmsSession_I
{
      void eventLossOccurred(java.lang.String startTime, java.lang.String
notificationId);
      void eventLossCleared(java.lang.String endTime);
      void historyPMDataCompleted(java.lang.String fileName);
      void historyPMDataFailed(java.lang.String errorReason);
      void operationInfo(java.lang.String opname,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalInfo);
      void alarmLossOccurred(java.lang.String startTime, java.lang.String
notificationId);
}
```

Description

The operationInfo method allows you to have a general callback mechanism that can be used for the configConnLessInfos and deleteAllConnLessInfos APIs. A background process is started at the EMS level and the termination of this process is reported through callback.

This method has two parameters:

- String opname—The string name of the invoked method. For example, configConnLessInfos or deleteAllConnLessInfos.

- NameAndStringValue_T[] additionalInfo—Contains a detailed description of the result as a name-value pair. For configConnLessInfos and deleteAllConnLessInfos APIs, it returns <name="status", value="completed">.

Example 1

```
opname="deleteAllConnLessInfos"
additionalInfo[0]= NameAndStringValue_T("status", "completed")
```

Example 2

```
opname="deleteAllConnLessInfos"
additionalInfo[0]= NameAndStringValue_T("status", "failed")
```

4.5.2 MFD Provisioning Interfaces

This section describes the following interfaces:

- 4.5.2.1 createMFD
- 4.5.2.2 assignCPTPsToMFD
- 4.5.2.3 unassignCPTPsFromMFD
- 4.5.2.4 deleteMFD
- 4.5.2.5 modifyMFD

4.5.2.1 createMFD

Synopsis

```
public void createMFD(MFDCreateData_T createData,
                      TPDataList_THolder tpsToModify,
                      MatrixFlowDomain_THolder theMFD,
                      StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to create an MFD with the parameters specified in the method. The NMS must specify the CPTPs to be associated with the MFD that will be created.

Parameters

- MFDCreateData_T createData—Describes the structure of the MFD to be created.
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- MatrixFlowDomain_T theMFD—The newly created MFD. The EMS is responsible for guaranteeing MFD name uniqueness. The NMS can specify the name using createData.
- string errorReason—Specifies the creation errors, if any.

Throws

```
Raises globaldefs::ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the createData field is invalid.
EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request
because at least one of the parameters, although valid, cannot be set or if an
unassigned CPTP could not be assigned.
EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is
not met.
```

EXCPT_OBJECT_IN_USE - Raised if a specified CPTP is already associated to an existing MFD or if the MFD name specified by the NMS is already assigned to an existing object.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to the ME that contains the MFD.

EXCPT_ENTITY_NOT_FOUND - Raised when one of the provided TPs does not exist.

Relevant Data Structures

MFDCreateData_T:

```
NameAndStringValue_T[]  name
String                  userLabel
boolean                 forceUniqueness
String                  owner
String                  networkAccessDomain
NameAndStringValue_T[][]  unassignedCPTPs
LayeredParameters_T[]   transmissionParams
NameAndStringValue_T[]  additionalCreationInfo
```

The name of each MFD must be in the following format:

name="EMS";value="CompanyName/EMSname"
name="ManagedElement";value="ManagedElementName"
name="MatrixFlowDomain"; value= "MatrixFlowDomainName"

In Prime Optical, each ME name is unique irrespective of which network partition or network access domain it belongs to. In this data structure, networkAccessDomain separates an ME from another that has the same name but belongs to a different network access domain.

Use Case Description

The following describes how the system requests to create an MFD:

1. The NMS sends the request to the EMS to create an MFD with the provided parameters.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an MFD object with the same user label exists, a User Label In Use exception is raised.
   c. If one of the specified TPs is unknown to the EMS, an Entity Not Found exception is raised.
   d. If at least one of the MFD parameters could not be set, an Unable To Comply exception is raised.
   e. If any of the specified TPs is already in use by another MFD, an Object In Use exception is raised.
3. If the request is valid, the EMS:
   a. Creates the MFD.
   b. Assigns the requested CPTPs to the MFD.
   c. Replies with a success indication.
   d. Sends object creation notifications to the notification service.

Limitations

The tpsToModify parameter is not supported.

4.5.2.2 assignCPTPsToMFD

Synopsis

```
public void assignCPTPsToMFD(NameAndStringValue_T[] mfdName,
```

```
                                    NameAndStringValue_T[][] tpNames,
                                    TPDataList_THolder tpsToModify,
                                    StringHolder errorReason)
        throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to assign one or more CPTPs to an MFD.

Parameters

- globaldefs::NamingAttributes_T mfdName—The MFD name to be modified.
- globaldefs::NamingAttributesList_T tpNames—The CPTP names to be assigned to the MFD. If the list is empty or if all the CPTPs are already assigned to the MFD, no operation is performed on the EMS and the method returns "success."
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—Specifies the reason for the fault, if any.

Throws

Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the mfdName parameter does not refer to an MFD object.

EXCPT_OBJECT_IN_USE - Raised when a CPTP is already assigned to another MFD.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName or one of the tpNames references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when a CPTP could not be assigned.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to the ME containing the MFD.

Use Case Description

The following describes how the system requests to assign CPTPs to an MFD:

1. The NMS sends a request to the EMS to assign a list of CPTPs to an existing MFD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified MFD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If the specified MFD is fixed, an Unable To Comply exception is raised.
   d. If one or more of the specified CPTPs is unknown to the EMS, an Entity Not Found exception is raised.
   e. If one or more of the specified CPTPs is not a potential CPTP for this MFD (is not in the unassigned CPTP PortTPRoleState or is not in the same equipment or on the same rack with backplane connectivity), an Unable To Comply exception is raised.
3. If the request is valid, the EMS:
   a. Assigns the specified CPTPs to the MFD. The CPTP PortTPRoleState attribute is set to assigned.
   b. Replies with a success indication.
   c. Sends an attribute value change notification to the notification service (using a notification of the MFD assignedCPTPs attribute). The notification includes the complete list of CPTP names that are assigned to the MFD.

Limitations

- The tpsToModify parameter is not supported.
- Notification changes are not sent.

4.5.2.3 unassignCPTPsFromMFD

Synopsis

```
void unassignCPTPsFromMFD(
NameAndStringValue_T[]  mfdName,
                                        NameAndStringValue_T[][] tpNames,
                                        TPDataList_THolder tpsToModify,
                                        StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to unassign one or more CPTPs from an MFD.

Parameters

- globaldefs::NamingAttributes_T mfdName—The MFD name to be modified.
- globaldefs::NamingAttributesList_T tpNames—The CPTP names to be unassigned from the MFD. If no CPTP has been specified, the method returns "success."
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—Specifies the reason for the fault, if any.

Throws

```
Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the mfdName parameter does not refer to an MFD object.

EXCPT_OBJECT_IN_USE - Raised when a CPTP to be unassigned is carrying traffic
or is connected to a call.

EXCPT_ENTITY_NOT_FOUND - Raised when the mfdName or tpNames parameter
references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when a CPTP could not be unassigned. For
example, the CPTP is not assigned to the MFD.

EXCPT_NOT_IN_VALID_STATE - Raised when at least one of the specified CPTPs is
in the unassigned PortTPRoleState.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to the ME that contains
the MFD.
```

Use Case Description

The following describes how the system requests to unassign CPTPs from an MFD:

1. The NMS sends a request to the EMS to unassign a list of CPTPs from an existing MFD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified MFD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If the specified MFD is fixed, an Unable To Comply exception is raised.
   d. If one or more of the specified CPTPs is unknown to the EMS, an Entity Not Found exception is raised.
   e. If one or more of the specified CPTPs is in the unassigned PortTPRoleState, a Not In Valid State exception is raised.
   f. If one or more of the specified CPTPs is used by an FDFr, an Object In Use exception is raised.
   g. If one or more of the specified CPTPs is not assigned to the specified MFD, an Unable To Comply exception is raised.

3. If the request is valid, the EMS:
   a. Unassigns the specified CPTPs from the MFD. The PortTPRoleState attribute of the CPTPs is set to unassigned. If any of the CPTPs could not be unassigned, no CPTP is unassigned and an Unable To Comply exception is raised.
   b. Replies with a success indication.
   c. Sends the appropriate notifications to the notification service.

Limitations

- The tpsToModify parameter is not supported.
- Notification changes are not sent.

4.5.2.4 deleteMFD

Synopsis

```
public void deleteMFD(NameAndStringValue_T[] mfdName,
                      TPDataList_THolder tpsToModify,
                      StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request the deletion of an MFD from the EMS.

Parameters

- NamingAttributes_T mfdName—The MFD name to be deleted.
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—If a best-effort parameter could not be set, the EMS provides the fault reason.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal  failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the mfdName parameter does not refer to an MFD object.

EXCPT_OBJECT_IN_USE - Raised when the MFD is associated with an FD.

EXCPT_ENTITY_NOT_FOUND - Raised when the mfdName attribute references an
object that does not exist.

EXCPT_ACCESS_DENIED - Raised if the MFD is fixed.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to the ME that contains
the MFD.
```

Use Case Description

The following describes how the system requests to delete an MFD:

1. The NMS sends the request to the EMS to delete an MFD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified MFD object is unknown to the EMS, an Entity Not Found exception is raised.
   c. The MFD to be deleted must not be associated with an FD. If the MFD is still associated, an Object In Use exception is raised.
   d. The MFD to be deleted must not be fixed. If the MFD is fixed, an Unable To Comply exception is raised.
3. If the request is valid, the EMS:

a. Deletes the MFD.
b. Releases all assigned TPs from the MFD.
c. Replies with a success indication.
d. Sends an object deletion notification to the notification service.

Limitations

- The tpsToModify parameter is not supported.

4.5.2.5 modifyMFD

Synopsis

```
public void modifyMFD(NameAndStringValue_T[] mfdName,
                      MFDModifyData_T mfdModifyData,
                      MatrixFlowDomain_THolder newMFD,
                      StringHolder failedAttributes,
                      StringHolder errorReason)
    throws org.tmforum.mtnm.globaldefs.ProcessingFailureException
```

Description

This interface enables the NMS to request a modification of an MFD.

Parameters

- globaldefs::NamingAttributes_T mfdName—The MFD name to be modified.
- MFDModifyData_T mfdModifyData—Structure describing how the MFD will be modified. Best effort is not supported. If modifying one parameter fails, an exception is raised.
- string failedAttributes—The list of attributes that could not be modified.
- MatrixFlowDomain_T newMFD—The modified MFD.
- string errorReason—If a best-effort parameter could not be set, the EMS provides the fault reason.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the mfdModifyData field is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when the mfdName attribute references an
object that does not exist.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is
not met.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

Relevant Data Structures

```
MFDModifyData_T

    String                      userLabel
    boolean                 forceUniqueness
    String                      owner
    String                      networkAccessDomain
    LayeredParameters_T[]   transmissionParams
    NameAndStringValue_T[]  additionalModificationInfo
```

The read and write attributes required to modify an MFD on the EMS are bundled in the MFDModifyData structure, which the NMS passes to the EMS.
The following describes the MFDModifyData structure attributes:

- userLabel—Can be specified by the NMS or it can be empty.
- forceUniqueness—Specifies whether userLabel uniqueness is required for EMS MFDs. If the userLabel is unique and is already in use, the operation fails.
- owner—Can be specified by the NMS or it can be empty.
- networkAccessDomain—The network access domain to which the FD is assigned.
- transmissionParameters::LayeredParameterList_T transmissionParams—A list of modified connectionless parameters. As an input only, the list of parameters to be changed, removed, or added is provided. If an entry must be removed, a hyphen (-) is specified as the value. When the method is returned, this attribute contains the list of parameters that could not be applied.
- globaldefs::NVSList_T additionalModificationInfo—Additional modification information can be specified by the NMS. When the method is returned, this attribute contains the parameters that could not be applied.

Use Case Description

The following describes how the system requests to modify an MFD:

1. The NMS sends a request to the EMS to modify an MFD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified MFD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an MFD object with the same user label exists, a User Label In Use exception is raised.
   d. If the EMS cannot satisfy any attribute that must be modified, an Unable To Comply exception is raised.
   e. If the MFD already has the required information, the EMS replies with a success indication, but no notification is generated.
3. If the request is valid:
   a. If the MFD has all the required information, the EMS does not send any notifications to the notification service.
   b. If the MFD does not have the required information, the EMS modifies the MFD attributes as requested.
   c. The EMS replies with a success indication.
   d. If the EMS makes a change, the appropriate notification is sent to the notification service.

Limitations

- The failedAttributes parameter is not supported.
- Notification changes are not sent.

4.5.3 Flow Domain Provisioning Interfaces

This section describes the following interfaces:

4.5.3.1 createFlowDomain

Synopsis

```
public void createFlowDomain(FDCreateData_T createData,
                             NamingAttributesList_THolder assignedCPTPs,
                             TPDataList_THolder tpsToModify,
                             FlowDomain_THolder theFD,
                             StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to create an FD with the parameters specified in the method. The NMS can specify MFDs or fdEdge CPTPs to be associated with the created FD.

Parameters

- FDCreateData_T createData—Describes the FD structure to be created.
- globaldefs::NamingAttributesList_T assignedCPTPs—Identifies the list of assigned CPTPs to be associated to the FD as fdEdge CPTPs. This list can be empty. Associating CPTPs to the FD is done on a best-effort basis. When the method is returned, the list contains the names of the CPTPs that could not be associated with the FD.
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- FlowDomain_T theFD—The newly created FD. The EMS is responsible for guaranteeing the uniqueness of the FD name. The NMS specifies the name in the createData parameter.
- string errorReason—The EMS provides the fault reason if a best-effort parameter could not be set or if a CPTP could not be associated with the FD.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the createData field is invalid.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request because at least one of the parameters, although valid, cannot be set and that parameter is identified as not "best effort" or when a CPTP is not assigned to one of the MFDs specified in the createData parameter or when one of the specified MFDs could not be associated.

EXCPT_ENTITY_NOT_FOUND - Raised when at least one of the specified CPTPs or MFDs does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met.

EXCPT_OBJECT_IN_USE - Raised when a specified MFD is already associated to an existing FD or if the FD name specified by the NMS is already assigned to an existing object.

Relevant Data Structures

FDCreateData_T

```
      NameAndStringValue_T[]  name;
```

```
String                          userLabel;
Boolean                   forceUniqueness;
String                          owner;
String                          networkAccessDomain;
NameAndStringValue_T[][]        mfds;
LayeredParameters_T[]     transmissionParams;
NameAndStringValue_T[]    additionalCreationInfo;
```

The name of each FD must be in the following format:

name="EMS";value="CompanyName/EMSname"

name="FlowDomain";value=" FlowDomainName"

In Prime Optical, each ME name is unique irrespective of which network partition or network access domain it belongs to. In this data structure, networkAccessDomain separates an ME from another that has the same name but belongs to a different network access domain.

Use Case Description

The following describes how the system requests to create an FD:

1. The NMS sends the request to the EMS to create an FD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If an FD object with the name specified already exists, an Object In Use exception is raised.
   c. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an FD object with the same user label already exists, a User Label In Use exception is raised.
   d. If one of the specified resources (MFDs or CPTPs) does not exist, an Entity Not Found exception is raised.
   e. If any of the MFDs to be associated is already associated to another FD, an Object In Use exception is raised.
   f. If any of the MFDs to be associated could not be associated to the FD, no MFD is associated and an Unable To Comply exception is raised.
   g. If a CPTP is not already assigned to one of the provided MFDs, an Unable To Comply exception is raised.
3. If the request is valid, the EMS:
   a. Creates the FD as requested.
   b. Associates the requested CPTPs to the new FD (the PortTPRoleState attribute of the CPTPs is set to fdEdge). All names of the CPTPs that could not be associated are returned in the reply.
   c. Replies with a success indication.
   d. Sends an FD object creation notification to the notification service.

✎ TPData (tpsToModify) structure contains TP data that can be set by the NMS; for example, the transmission parameters that must be applied to the TP. Only a subset of the parameters is specified in the list and only these parameters should be applied in the NE. If the list is empty, nothing will be done. To remove a parameter from the list, use a hyphen (-) in the value part of the structure.

Limitations

- The tpsToModify parameter is not supported.
- The assignedCPTPs parameter is not supported. All CPTPs that are currently assigned to the MFDs specified in the createData attribute are automatically moved to fdEdge role state after the MFD is assigned to the FD. To explicitly assign CPTPs to an FD, use the assignCPTPsToMFD interface.

4.5.3.2 deleteFlowDomain

Synopsis

```
public void deleteFlowDomain(NameAndStringValue_T[] fdName,
                                TPDataList_THolder tpsToModify,
                                StringHolder errorReason)
      throws ProcessingFailureException
```

Description
This interface enables the NMS to request the EMS to delete an FD.
Parameters
- NamingAttributes_T fdName—The name of the FD to be deleted.
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—Specifies the reason for the fault, if any.

Throws
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the fdName attribute does not refer to an FD object.

EXCPT_OBJECT_IN_USE - Raised when the FD contains at least one FDFr.

EXCPT_ENTITY_NOT_FOUND - Raised when the fdName attribute references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when an associated fdEdge CPTP or MFD could not be disassociated.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

Use Case Description
This interface allows an NMS to delete an existing FD. The EMS verifies that no FDFr exists within the FD. The operation disassociates the fdEdge CPTPs and MFDs, and deletes the FD.
The following describes how the system requests to delete an FD:
1. The NMS sends the request to the EMS to delete an FD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the FD object is unknown to the EMS, an Entity Not Found exception is raised.
   c. If any FDFr is used in the FD, an Object In Use exception is raised.
   d. If any of the MFDs cannot be disassociated from the specified FD, no MFD is disassociated and an Unable To Comply exception is raised.
   e. If any fdEdge CPTPs cannot be disassociated from the FD (the PortTPRoleState attribute of the CPTPs was set to assigned), no fdEdge CPTPs are disassociated and an Unable To Comply exception is raised.
3. If the request is valid, the EMS:
   a. Deletes the FD.
   b. Replies with a success indication.
   c. Sends an FD object deletion notification to the notification service.
4. If the request is successful:
   a. The fdEdge CPTPs associated to the FD are disassociated (the PortTPRoleState attribute of the CPTPs is set to assigned).
   b. The MFDs associated to the FD are disassociated.
   c. The FD is deleted.

4.5.3.3 modifyFlowDomain

Synopsis

```
public void modifyFlowDomain(NameAndStringValue_T[] fdName,
                             FDModifyData_T fdModifyData,
                             FlowDomain_THolder modifiedFD,
                             StringHolder failedAttributes,
                             StringHolder errorReason)
throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to modify an existing FD, as specified by the parameters in the method. The NMS can modify the user label, owner, network access domain, connectionless layered parameters, or additional information on an existing FD.

Parameters

- NamingAttributes_T fdName—The FD name to be modified.
- FDModifyData_T fdModifyData—Describes how the FD should be modified. If the FD contains the required information, nothing is done on the EMS and the method returns "success."
- string failedAttributes—The list of attributes that could not be modified.
- FlowDomain_T modifiedFD—The modified FD.
- string errorReason—Specifies the reason for the fault, if any.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the fdModifyData field is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when the fdName attribute references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request because at least one of the parameters, although valid, cannot be set.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation, which prevents the creation of an MFD or CPTP.

Relevant Data Structures

FDModifyData_T

```
String                    userLabel
boolean                   forceUniqueness
String                    owner
String                    networkAccessDomain
LayeredParameters_T[]     transmissionParams
NameAndStringValue_T[]    additionalModificationInfo
```

Use Case Description

The following describes how the system requests to modify an FD:

1. The NMS sends the request to the EMS to modify an FD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified FD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an FD object with the same user label exists, a User Label In Use exception is raised.
   d. If the EMS cannot satisfy any attribute that must be modified, an Unable To Comply exception is raised.
3. If the request is valid:
   a. If the FD already has all the required information, the EMS does not send any notifications to the notification service.
   b. If the FD does not have the required information, the EMS modifies the FD attributes as requested.
   c. The EMS replies with a success indication.
4. If the EMS makes a change, the appropriate notification is sent to the notification service.

Limitations

- The failedAttributes parameter is not supported.
- Notification changes are not sent.

4.5.3.4 associateMFDsWithFlowDomain

Synopsis

```
public void associateMFDsWithFlowDomain(NameAndStringValue_T[] fdName,
                                        NameAndStringValue_T[][] mfdNames,
                                        TPDataList_THolder tpsToModify,
                                        StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request the EMS to associate one or more MFDs with an FD.

Parameters

- NamingAttributes_T fdName—The FD name to which the MFDs will be associated.
- NamingAttributesList_T mfdNames—The names of the MFDs to be associated with the FD. If the list is empty, nothing is done on the EMS and the method returns "success."
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—Specifies the reason for the fault, if any.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the fdName attribute does not refer to an FD object.

EXCPT_OBJECT_IN_USE - Raised when an MFD is already associated with another
FD.

EXCPT_ENTITY_NOT_FOUND - Raised when the fdName or mfdNames attribute
references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when an MFD could not be associated.
```

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

Use Case Description

The following describes how the system requests to associate additional MFDs to an existing FD:

1. The NMS sends the request to the EMS to associate additional MFDs to an existing FD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified FD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If one or more of the specified MFDs is unknown to the EMS, an Entity Not Found exception is raised.
   d. If one or more of the specified MFDs are already associated to another FD, an Object In Use exception is raised.
   e. If any of the MFDs could not be associated, no MFD is associated and an Unable To Comply exception is raised.
3. If the request is valid, the EMS:
   a. Associates the specified MFDs to the FD.
   b. Verifies the server layer connectivity between the associated MFDs. The FD Connectivity State attribute is modified accordingly (fully connected, not fully connected, or unknown).
   c. Replies with a success indication.
   d. Sends an attribute value change notification to the notification service using the MatrixFlowDomain attribute of the FD. The notification includes the complete list of MFD names that are associated to the FD.
4. If the FD Connectivity State attribute is modified, the EMS sends a state change notification to the notification service.

Limitations

- Notification changes are not sent.

4.5.3.5 deAssociateMFDsFromFlowDomain

Synopsis

```
public void associateMFDsWithFlowDomain(NameAndStringValue_T[] fdName,
                                        NameAndStringValue_T[][] mfdNames,
                                        TPDataList_THolder tpsToModify,
                                        StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to disassociate one or more MFDs from an existing FD. This operation also disassociates the fdEdge CPTPs that are associated to the MFDs to be disassociated.

Parameters

- NamingAttributes_T fdName—The FD name to be modified.
- NamingAttributesList_T mfdNames—The names of the MFDs to be disassociated from the FD. If the list is empty, nothing is done on the EMS and the method returns "success."
- TPDataList_T tpsToModify:
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—Specifies the reason for the fault, if any.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the fdName attribute does not refer to an FD object.

EXCPT_OBJECT_IN_USE - Raised when an MFD to be disassociated carries traffic.

EXCPT_ENTITY_NOT_FOUND - Raised when the fdName or mfdNames attribute references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when an MFD could not be disassociated. For example, the MFD is not associated with the FD or if an fdEdge CPTP specified by one of the MFDs could not be disassociated.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

Use Case Description

The EMS must validate the data provided by the NMS and disassociate the requested MFDs from the specified FD. If the EMS cannot disassociate the specified MFDs, an appropriate exception is raised. Best effort is not supported. The operation also disassociates the fdEdge CPTPs that are associated to the MFDs to be disassociated. After the disassociation, the EMS verifies the server layer connectivity between the associated MFDs using the FD Connectivity State attribute.

The following describes how the system requests to disassociate MFDs from an existing FD:

1. The NMS sends the request to the EMS to disassociate MFDs from an existing FD.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the specified FD is unknown to the EMS, an Entity Not Found exception is raised.
   c. If one or more of the specified MFDs were not previously associated to the FD, an Unable To Comply exception is raised.
   d. If one or more of the specified MFDs is carrying traffic, an Object In Use exception is raised.
   e. If any fdEdge CPTPs cannot be disassociated from the MFDs that are being disassociated (the PortTPRoleState attribute of the CPTPs is set to assigned), no fdEdge CPTPs are disassociated and an Unable To Comply exception is raised.
   f. If any of the MFDs could not be disassociated, no MFD is disassociated and an Unable To Comply exception is raised.
3. If the request is valid, the EMS:
   a. Disassociates the specified MFDs from the FD.
   b. Replies with a success indication.
   c. Sends an attribute value change notification to the notification service using the MatrixFlowDomain attribute of the FD. The notification includes the complete list of MFD names that are associated to the FD.
   d. Verifies the server layer connectivity between the associated MFDs. The FD Connectivity State attribute is modified accordingly (fully connected, not fully connected, or unknown).
4. If the FD Connectivity State attribute is modified, the EMS sends a state change notification to the notification service.
5. If the request is successful:
   a. The requested MFDs are disassociated from the FD.
   b. The corresponding fdEdge CPTPs are also disassociated from the FD.
   c. If the request fails, there is no change in the system or the MFD association to the FD.

Limitations

Notification changes are not sent.

4.5.4 EVC FDFr Provisioning Interfaces

This section describes the following interfaces:

4.5.4.1 createAndActivateFDFr

Synopsis

```
public void createAndActivateFDFr(
FDFrCreateData_T              createData,
                                    ConnectivityRequirement_T
      connectivityRequirement,
                                    NamingAttributesList_THolder  aEnd,
                                    NamingAttributesList_THolder  zEnd,
                                    NamingAttributesList_THolder  internalTPs,
                                    MatrixFlowDomainFragmentList_THolder
mfdfrs,
                                    TPDataList_THolder
      tpsToModify,
                                    FlowDomainFragment_THolder    theFDFr,
                                    NamingAttributesList_THolder
      notConnectableCPTPList,
                                    NamingAttributesList_THolder
      parameterProblemsTPList,
                                    StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface requests the EMS to create and activate an FDFr representing an EVC using the parameters specified in the method. This operation allows you to create and activate the following EVCs at the LR_EVC (168) layer rate:

- Ethernet Private Line
- Ethernet Virtual Private Line
- Ethernet Private LAN
- Ethernet Virtual Private LAN

Figure 4-5 shows the point-to-point EFP.

Figure 4-5: Point-to-Point EFP

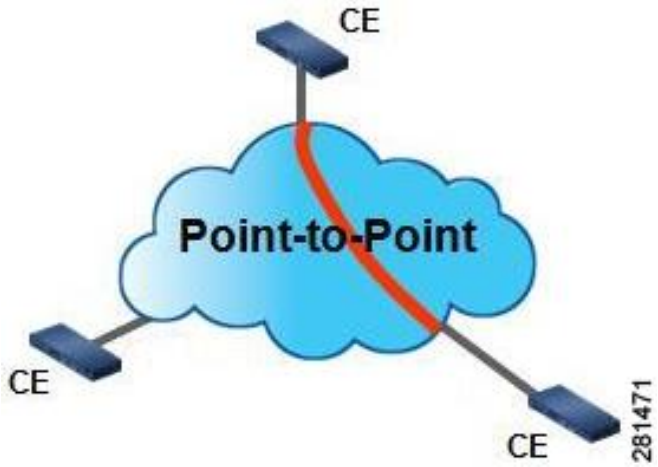
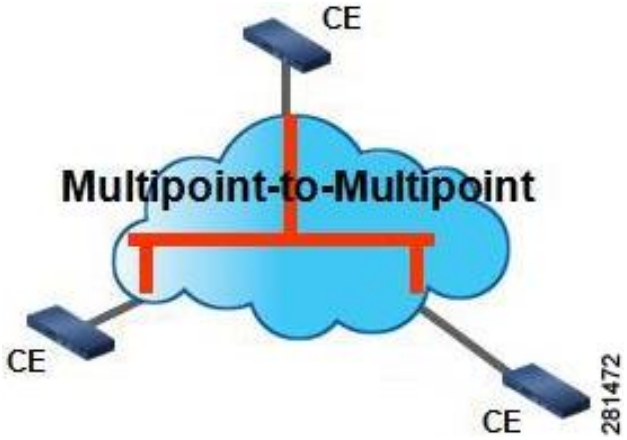


Figure 4-6 shows the multipoint-to-multipoint EFP.

Figure 4-6: Multipoint-to-Multipoint EFP



For example, if the network can automatically route, no internal CPTPs have to be specified. To configure EFP, you must consider the following:

- If aEnd and zEnd are in the same ME—Specify the following EFP layered parameters:
  - aEnd[0] (external source) EFP
  - zEnd[0] (external destination) EFP
- If aEnd and zEnd are in different MEs—Specify the following EFP layered parameters:
  - aEnd[0] (external source) EFP
  - aEnd[1] (internal source) EFP
  - zEnd[0] (external destination) EFP

If aEnd and zEnd are in different MEs, the EMS performs an Apply-All EFP configuration of the external and internal source EFPs and then overrides the external destination EFP with the one you provide.

The network creates the internal intermediate EFPs. The aEnd and zEnd parameters contain the associated CTPs; for example:

```
CPT600-65-126
/rack=1/shelf=1/slot=7/ppm_holder=3/port=1
/ethvid=720
```

or

```
CPT600-65-126
/rack=1/shelf=1/slot=7/ppm_holder=3/port=1
```

```
/eth=default
```

The EFP parameters must be specified using the tpsToModify parameter. The LR_EVC (168) and the layered parameters are reported below:

```
CPT600-65-126
/rack=1/shelf=1/slot=7/ppm_holder=3/port=1


168
outerVlanTpId
DOT1Q
innerVlanTpId
DOT1Q
isExact
true
rewriteOper
REWRITE_NONE
rewriteOuterVlanTpId
TP_NONE
rewriteOuterVlanTag
0
rewriteInnerVlanTpId
TP_NONE
rewriteInnerVlanTag
0
rewriteIsSymmetric
false
enableStatistics
false
```

For Ethernet Private Line EVCs, the number of drops must always be two. For Ethernet Private LAN EVCs, the number of drops must be two at creation, but more drops can be added using the modifyFDFr( ) API. For Private EVCs (Ethernet Private Line or Ethernet Private LAN), at least one of the drops must have the "/eth=default" tagging. To include MEs in EVC routing, you must specify the MEs in the tpsToModify parameter.

Use Case Description

The following describes how the system requests to create and activate an FDFr:

1. The NMS sends a request to the EMS to create and activate an FDFr.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If one of the referenced CPTP objects is unknown to the EMS, an Entity Not Found exception is raised.
   c. If the specified CPTPs are not associated with the referenced FD, a TP Invalid Endpoint exception is raised.
   d. If an FDFr with the same properties as specified in the NMS request already exists, the EMS reuses that FDFr.

   e. If any of the specified Edge CPTPs do not have the fdEdge role, a Not In Valid State exception is raised.
   f. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an FDFr object with the same user label exists already, a User Label In Use exception is raised.
   g. If the FDFr being created will have fewer than two edge FPs, an Unable To Comply exception is raised.
3. The transmission parameters for the involved CPTPs and FPs are sent to the network by the EMS, which activated the MFDFrs as appropriate, based on the automatic route taken by the network. If any entity or parameter cannot be provisioned, a corresponding exception is raised (failed TP list is not managed).
4. The EMS initiates the activation of the FDFr, which involves establishing the MFDFrs at the MEs.
5. If all of the MFDFrs comprising the FDFr have been established, the EMS sets the FDFr state to active.
6. If one or more of the MFDFrs comprising the FDFr are not established, the EMS sets the FDFr state to partial.
7. The EMS replies with a success indication.

For descriptions of the input parameters, see 4.5.4.1 createAndActivateFDFr.
Manual routing is not required for EVCs. Nodes or links can be included or excluded to drive the automatic routing performed by the network.
For pseudowires, the following layered parameters associated to the new LR_EVC layer rate change.

Name	Type	Description
evcType	String	Valid values are: • Ethernet Private Line • Ethernet Virtual Private Line • Ethernet Private LAN • Ethernet Virtual Private LAN
bw	Long	EVC bandwidth
bwUnits	String	Valid values are: • Kbps • Mbps • Gbps

The following layered parameters that refer to the EFP configuration are specified within the tpsToModify parameter that is associated to the LR_EVC layer rate.

✐ Pseudowire class, VC ID, static labels, and protection parameters do not apply to EVC drops.

Name	Type	Description

Name	Type	Description
outerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200
innerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 In Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
isExactTag	Boolean	If the VLAN tag is exact, set the value to true.
rewriteOper	String	Valid values are: • REWRITE_NONE • PUSH_1 • PUSH_2 • POP_1 • POP_2 • TRANSLATE_1_TO_1 • TRANSLATE_1_TO_2 • TRANSLATE_2_TO_1 • TRANSLATE_2_TO_2
rewriteOuterVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200
rewriteOuterVlanTag	Integer	—

Name	Type	Description
rewriteInnerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 In Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
rewriteInnerVlanTag	Integer	—
rewriteIsSymmetric	Boolean	Valid values are: • true • false
enableStatistics	Boolean	Enables statistics counter. Valid values are: • true • false
statisticsCategory	String	Valid values are: • INGRESS • EGRESS • BOTH This is mandatory only if enableStatistics is set to true. In Prime Optical 10.7, by default, INGRESS is set to true.

The following table lists the mandatory parameters based on tagging type.

Name	Single Tagged	Double Tagged	Untagged	Default
outerVlanTpId	Yes	Yes	No	No
innerVlanTpId	No	Yes	No	No
isExactTag	No	No	No	No
rewriteOper	Yes	Yes	No	No
rewriteOuterVlanTpId	No	No	No	No
rewriteOuterVlanTag	No	No	No	No
rewriteInnerVlanTpId	No	No	No	No
rewriteInnerVlanTag	No	No	No	No
rewriteIsSymmetric	No	No	No	No

Name	Single Tagged	Double Tagged	Untagged	Default
enableStatistics	Yes	Yes	Yes	Yes
statisticsCategory	No	No	No	No
qosTableMap	No	No	No	No
qosIngPolicy	No	No	No	No
qosEgrPolicy	No	No	No	No

The network automatically assigns the SID to the FDFr at the time of creation. This SID is returned in the name. For pseudowires, see the following example:

```
Flow Domain Fragment Name   =       [0] EMS=Cisco Systems/PrimeOptical;
[1] FlowDomain=CTM_NP_1;
[2] FlowDomainFragment=GIAN_EVC_LAN_1:SID=8:LR=EVC
Native EMS Name  =            GIAN_EVC_LAN_1
```

When the creation API is executed, the service is not discovered for all parameters. The API returns a dummy object based on the input parameters. The actual objects are retrieved through the standard inventory APIs.

Limitations

Object create notifications are not generated.

4.5.4.2 deactivateAndDeleteFDFr

Synopsis

```
public void deactivateAndDeleteFDFr(
  NameAndStringValue_T[]      fdfrName,
                                      TPDataList_THolder    tpsToModify,
                                      StringHolder                errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to deactivate and delete an FDFr that represents an EVC from an FD. For descriptions of the input parameters, see 4.5.4.2 deactivateAndDeleteFDFr.

The EVC name is specified as <EVC native name>:SID=nn:LR=EVC.

In this example, LR=EVC stands for Layer Rate = EVC. This is introduced because the API does not expect an LR to be specified.

Parameters

- NamingAttributes_T fdfrName—Enables to deactivate and delete the EVC name.
- string errorReason—Specifies the reason for the fault, if any.

Parameter “errorReason.value” refers to an empty string if the operation is performed successfully.

Limitations

Object delete notifications are not generated.

Throws

```
Raises ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, fdfrName does not refer to an FDFr object.
```

EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an object that does not exist.

EXCPT_ACCESS_DENIED - Raised if the EMS does not allow the NMS to delete the FDFr. For example, the FDFr state is fixed.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.5.4.3 modifyFDFr

Synopsis

```
modifyFDFr(
 NameAndStringValue_T[]              fdfrName,
                         FDFrModifyData_T                    fdfrModifyData,
                         ConnectivityRequirement_T
      connectivityRequirement,
                         TPDataList_THolder                  tpsToModify,
                         NamingAttributesList_THolder        failedTPList,
                         NamingAttributesList_THolder
      parameterProblemsTPList,
                         FlowDomainFragment_THolder   newFDFr,
                         StringHolder                        errorReason)
      throws ProcessingFailureException
```

Description

This interface enables the EMS to modify an existing FDFr representing an EVC as specified by the parameters in the method.

The EVC name is specified as <EVC native name>:SID=nn:LR=EVC.

In this example, LR=EVC stands for Layer Rate = EVC. This is introduced because the API does not expect an LR to be specified. For descriptions of the input parameters, see 4.5.4.3 modifyFDFr.

The difference between pseudowires and EVCs is that for Ethernet Private LANs, you can add or remove drops after you create the FDFr. You must add or delete only one drop at a time. To achieve this, you must use the following structure:

```
FDFrModifyData_T
      String                        userLabel (description)
      boolean                 forceUniqueness
      String                        owner
      String                        networkAccessDomain
      AdministrativeState_T   administrativeState
      LayeredParameters_T     transmissionParams
      NameAndStringValue_T[][]      tpNamesToRemove
      NameAndStringValue_T[][]      aEndTPNames
```

```
NameAndStringValue_T[][]      zEndTPNames
NameAndStringValue_T[][]      internalTPNames
NameAndStringValue_T[]  additionalModificationInfo
```

To add or remove a new drop, you must specify the tpNamesToRemove or zEndTPNames parameters.

✐ In multipoint services, all endpoints are considered as aEnd point drops.

Change Admin State

The administrative state change applies to the entire service only if the tpsToModify or zEndTPNames parameters are not specified. If they are specified, the administrative state change applies only to those services specified in the tpsToModify parameter and to the new drop specified.

To change the administrative state, you must set the changeAdminState parameter to true in the additionalModificationInfo parameter.

The additionalModificationInfo parameter contains the following additional parameter.

Name	Type	Description
changeAdminState	Boolean	Indicates whether or not the admin state is changed based on the new value specified in the administrativeState field. Valid values are: • true • false

Add New Drop

To add a new drop to a multipoint EVC, you must specify the new CTPs in the zEndTPNames parameter and the EFP parameters in the tpsToModify parameter.

For creating and modifying an EVC, consider the following:

- The new drop (zEnd[0]) is in an ME which is already part of the EVC. In this case, you must not specify the EFP configuration for the internal-intermediate drops as being already present. You need to only specify the new drop in zEnd[0] and its associated EFP parameters in the tpsToModify parameter.
- The new drop (zEnd[0]) is in an ME which currently does not belong to the EVC. In this case, you must specify the EFP configuration for both the external drop to be added (zEnd[0]) and the internal-intermediate drop (zEnd[1]) in the same ME, which is used for all the intermediate drops in the route taken by the network.

If the new drop is in an ME which currently does not belong to the EVC, the SID associated to the EVC can change. If the current SID value associated to the EVC is not available in the ME of the new drop, the network automatically picks up the first available SID shared by the current ME belonging to the EVC and the new drop ME. The new SID value is returned by the API in the result.

✐ The source and new endpoint EFP configuration must be the same.

The addition or deletion of drops and the modification of the administrative state are mutually exclusive. For example, if you want to add a drop and remove another one, you must perform the modifyFDFr operation twice with the corresponding parameters.

QoS Change

Associate the QoS parameters to a given EVC by specifying the following layered parameters (associated to the LR_EVC layer rate) in the transmission parameters:

Name	Type	Description
qosTableMap	String	QoS table map name (TC profile)
qosIngPolicy	String	QoS ingress policy map name (TC profile)
qosEgrPolicy	String	QoS egress policy name (TC profile)

splitHorizontal Change

Select the splitHorizontal parameter to a given EFP by specifying the following layered parameter (associated to the LR_EVC layer rate) in the transmission parameters:

Name	Type	Description
splitHorizontal	Boolean	Indicates whether or not the splitHorizontal EFP parameters of a TP are enabled. Valid values are: • true • false

Limitations
Object modification notifications are not generated.

4.5.5 Link Aggregation Provisioning Interfaces

This section describes the following interfaces:

- 4.5.5.1 createFTP
- 4.5.5.2 deleteFTP
- 4.5.5.3 setTPData

4.5.5.1 createFTP

Synopsis

```
public void createFTP(
FTPCreateData_T          createData,
                         TPDataList_THolder            tpsToModify,
                         TerminationPoint_THolder      theFTP,
                         StringHolder                  errorReason)
      throws ProcessingFailureException
```

Description
This interface enables the NMS to create an FTP. A LAN port, which usually can be an edge CPTP, cannot be an edge CPTP if it is a member of a LAG. The LAG FTP is the edge CPTP.

Parameters

- FTPCreateData_T createData—Describes the FTP structure to be created.
- TPDataList_T tpsToModify—List of TPs and parameters to be applied and updated to provide the resulting parameters. The transmission parameters are applied on a best-effort basis and the resulting values of the transmission parameters are provided in the updated tpsToModify parameter. If the parameter is not best effort, the entire operation is rejected and the appropriate exception is returned. The tpsToModify parameter can be used by the NMS to determine the number of server layer CTPs to create for the specified FTP. An empty list signifies that the number of server layer CTPs is determined by the EMS.
- TerminationPoint_T theFTP—The new FTP. The EMS is responsible for guaranteeing FTP name uniqueness. The name is specified by the NMS in the createData parameter.
- string errorReason—Specifies the errors during creation, if any.

Throws

Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the createData field is invalid.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS (or NE) does not support explicit creation of CPTPs.

EXCPT_ENTITY_NOT_FOUND - Raised when the specified equipment does not exist.

EXCPT_CAPACITY_EXCEEDED - Raised when the equipment does not have the resources. For example, less backplane bandwidth or less usable time slots.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME that contains the FTP.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met.

EXCPT_OBJECT_IN_USE - Raised when the object identified in the request is currently in use.

Relevant Data Structures

FTPCreateData_T

```
     NameAndStringValue_T[]   equipmentName
                        The FTP name built as:
name="EMS";value="CompanyName/EMSname"
name="ManagedElement";value="ManagedElementName"
name="FTP";value="/rack=1/shelf=1/port=nn"

     string                   userLabel
                        free text description for the FTP LAG. Optional. The
maximum length is 254 characters.

     boolean                  forceUniqueness;
     string                   owner;
     string                   networkAccessDomain;
     NameAndStringValue_T[]   ingressTransmissionDescriptorName;
     NameAndStringValue_T[]   egressTransmissionDescriptorName;
     TerminationMode_T  tpMappingMode;
     Directionality_T         direction;
     LayeredParameters_T[]    transmissionParams;
                        Specify the new layer rate LR_LAG_Freagment (305)
     NameAndStringValue_T[]   additionalCreationInfo;

LayeredParameters_T
      short                         layer
      NameAndStringValue_T[]  transmissionParams
```

The following table describes the mandatory LACP protocol attributes at the FTP Level.

Name	Value	Description
MTU	Integer	MTU number. Valid values are from 64 to 9600.
SoakTime	Integer	Optional. The default value is 10 seconds. SoakTime is the duration for which the failed interface remains inoperative even after the failure is rectified. SoakTime can be configured if Remote Ethernet Port Shutdown is disabled, but it is effective only when Remote Ethernet Port Shutdown is enabled. Valid values are from 0 to 300 seconds. SoakTime applies only to individual ports, and not to the ports of a channel group if the channel group is configured as an attachment circuit. This attribute is available in Prime Optical 10.7.
RemoteLinkFailureNotify	Boolean	Optional. If you omit, the attribute value is set to true. If you specify, the attribute value must be set to true when creating a channel group. This attribute is available in Prime Optical 10.7.
FastSwitchOver	Boolean	Enables 1:1 LACP. The valid value is true.
L2ActionCDP	String	CDP is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
L2ActionDOT1X	String	DOT1X is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
L2ActionDTP	String	DTP is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
L2ActionLACP	String	LACP is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
L2ActionPAGP	String	PAGP is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
L2ActionSTP	String	STP is a Layer 2 protocol. Valid actions are: • Peer • Forward

Name	Value	Description
		• Drop
L2ActionVTP	String	VTP is a Layer 2 protocol. Valid actions are: • Peer • Forward • Drop
IngressPolicyMap	String	Ingress policy map name. Valid value is an empty string (""). In Prime Optical 10.7, this attribute is not supported. You must specify this attribute and set it to an empty string. For more information about policy maps, see 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.
IngressTableMap	String	Ingress table map name. Valid value is an empty string (""). In Prime Optical 10.7, this attribute is not supported. You must specify this attribute and set it to an empty string. For more information about table maps, see 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.
IngressTableMapConfig	String	Ingress table map config name. Valid value is an empty string (""). In Prime Optical 10.7, this attribute is not supported. You must specify this attribute and set it to an empty string.
EgressPolicyMap	String	Egress policy map name. Valid value is an empty string (""). In Prime Optical 10.7, this attribute is not supported. You must specify this attribute and set it to an empty string. For more information about policy maps, see 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.
enableLACP	Boolean	Enables LACP if the value is true. The value is not applied if the command does not include aggregated ports.
MinBundle	Integer	Minimum bundle number. If portLACP is false, this attribute must be set to 1. Valid values are from 1 to 8.
MaxBundle	Integer	Maximum bundle number. If portLACP is false, this attribute must be set to 8. Valid values are from 1 to 8.

Relevant Data Structures

```
TPDataList_THolder
TPData_T[]              value
TPData_T
      NameAndStringValue_T[]  tpName
This is the PTP TMF name of the Ethernet built as:
```

```
name="EMS";value="CompanyName/EMSname"
name="ManagedElement";value="ManagedElementName"
name="PTP";value:"/rack=1/shelf=1/slot=2/ppm_holder=1/port to be
aggregated to the FTP LAG=1"

      TerminationMode_T              tpMappingMode (not handled NA)
      LayeredParameters_T[]    transmissionParams
      NameAndStringValue_T[]   ingressTrafficDescriptorName (empty)
      NameAndStringValue_T[]   egressTrafficDescriptorName (empty)

LayeredParameters_T
      short                          layer
                               in our case must be specified the new LR LR_LACP
(169)
      NameAndStringValue_T[]   transmissionParams
```

If LACP is not enabled at the FTP level, the transmissionParams attribute is left empty. If LACP is enabled at the FTP level, the transmissionParams attribute is used to communicate with LACP parameters:

Name	Value	Valid Values
LacpConfig	String	Valid values are: • Active • Passive
LacpPriority	Integer	Valid values are from 0 to 65535.

Use Case Description

The following describes how the system requests to create an FTP.:

- The EMS creates the requested FTP consistently with its implementation and the NMS specifications.
- If the EMS cannot create the FTP as specified, an appropriate exception is raised.
- The NMS specifies the FTP by choosing a location within the NE.
- After the EMS validates the input data, and if it can satisfy the input constraints, the EMS proceeds with implementing the FTP and returns the name of the new FTP and all of the attributes to the NMS.
- If the EMS fails to validate the input, it raises an appropriate exception.
- If transmission parameters are specified for the contained CTPs that are created, the EMS applies the transmission parameters after the FTP creation.
- Alarm reporting on the FTPs and contained CTPs might be enabled by the EMS using the alarm reporting transmission parameter.
- An object creation notification is sent to notify the NMS about the existence of the new FTP.

✐ If virtual concatenation applies, no notification is sent to the contained CTPs.

4.5.5.2 deleteFTP

Synopsis

```
public void deleteFTP(
NameAndStringValue_T[]  ftpName,
```

```
                          TPDataList_THolder            tpsToModify,
                          StringHolder                  errorReason)
throws ProcessingFailureException
```

Description
This interface enables the NMS to request the deletion of an FTP from the EMS. The service may be used to delete CPTP FTPs.
Parameters

- NamingAttributes_T ftpName—The FTP LAG name to be deleted. Because an Ethernet LAG port can aggregate client Ethernet ports of different ME cards, the corresponding FTP is logically positioned on the unique shelf of the ME itself:
  name="EMS";value="CompanyName/EMSname"
  name="ManagedElement";value="ManagedElementName"
  name="FTP";value="/rack=1/shelf=1/port=nn"
- TPDataList_T tpsToModify
  - in—The list of TPs with associated parameters to be applied.
  - out—The list of TPs with associated applied parameters.
- string errorReason—If a best-effort parameter could not be set, the EMS provides the fault reason.

Throws

```
Raises ProcessingFailureException:

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, the tpName does not refer to a CPTP TP object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not
exist.

EXCPT_OBJECT_IN_USE - Raised when the CPTP is an ELL endpoint.

EXCPT_UNABLE_TO_COMPLY - Raised if the TP cannot be deleted.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

4.5.5.3 setTPData

Synopsis

```
public void setTPData(TPData_T tpInfo,
                                        TerminationPoint_THolder modifiedTP)
        throws ProcessingFailureException
```

Description
This interface allows the NMS to set parameters on a specified TP.

```
TPData_T

      NameAndStringValue_T[]  tpName

The FTP name built as:

name="EMS";value="CompanyName/EMSname"

name="ManagedElement";value="ManagedElementName"

name="FTP";value="/rack=1/shelf=1/port=nn"

      TerminationMode_T             tpMappingMode (not handled NA)
```

```
LayeredParameters_T[]    transmissionParams
NameAndStringValue_T[]   ingressTrafficDescriptorName (empty)
NameAndStringValue_T[]   egressTrafficDescriptorName (empty)

LayeredParameters_T
      short                          layer
                                 i Specify the new layer rate LR_LAG_Freagment
(305)
      NameAndStringValue_T[]   transmissionParams
```

The following table lists the transmission parameters.

Name	Value	Description
IngressPolicyMap	String	Ingress policy map name. An empty string ("") is an invalid value. The None value removes the policy reference from the FTP. For more information about policy maps, see 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.
IngressTableMap	String	Ingress table map name. An empty string ("") is an invalid value. The None value removes the policy reference from the FTP. For more information about table maps, 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.
IngressTableMapConfig	String	Ingress table map configuration name. Valid values are: • None • Cos • MPLS The None value resets the current setting. An empty string ("") is an invalid value.
EgressPolicyMap	String	Egress policy map name. An empty string ("") is an invalid value. The None value removes the policy reference from the FTP. For more information about policy maps, see 4.5.6 Traffic Conditioning Profile Provisioning Interfaces.

Throws

globaldefs::ProcessingFailureException:

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_ENTITY_NOT_FOUND - Raised if the TP referred to in tpInfo does not exist.

EXCPT_INVALID_INPUT - Raised when tpInfo does not reference a valid CTP object.

```
EXCPT_NE_COMM_LOSS - Raised when communication to an ME is lost.
```

Compliance
TMF-defined.

4.5.6 Traffic Conditioning Profile Provisioning Interfaces

The Traffic Conditioning (TC) function is a transport processing function. It does the following:

- Accepts the characteristic information about the layer network at input.
- Classifies the traffic units according to configured rules.
- Meters each traffic unit to determine eligibility.
- Polices nonconforming traffic units.
- Presents the remaining traffic units at output.

The TC function is modeled by a TC profile that contains the bandwidth parameters that police the traffic at the ingress and egress of a connectionless layer network.
The API that handles Traffic Conditioning (TC) Profiles uses the TCProfile_T and TCProfileCreateData_T data structures.

TCProfile_T

```
struct TCProfile_T {
     globaldefs::NamingAttributes_T name;
     string userLabel;
     string nativeEMSName;
     string owner;
     boolean defaultProfile;
     transmissionParameters::LayeredParameterList_T transmissionParams;
     globaldefs::NVSList_T additionalInfo;
   };

TCProfileCreateData_T
struct TCProfileCreateData_T {
    string userLabel;
    boolean forceUniqueness;
    string owner;
    transmissionParameters::LayeredParameterList_T transmissionParams;
    globaldefs::NVSList_T additionalCreationInfo;
   };
```

Prime Optical 10.7 models and groups CTP QoS parameters into four different classes of objects:

- Class maps
- Actions
- Policy maps
- Table maps

The QoS parameters are stored in the Prime Optical database, which you can access from the Prime Optical user interface and the NMS through the GateWay/CORBA API. The EMS models each of the objects with a TC profile. The NMS can use theGateWay/CORBA API to create, fetch, modify, and delete TC profiles in the repository.

The EMS enforces and guarantees the uniqueness of user labels within any given QoS class of objects (class maps), but not across different QoS classes of objects.
For any newly created TC profile, the EMS automatically assigns a name with the following space-separated concatenation:
<QoS object class>" "<user label>

Example

```
class-map cMap123
class-map map143
policy-map pol23
policy-map polMap23
action act88
action a32
table-map tab34
table-map tab31
```

As per TMF standards, the forceUniqueness parameter can be used to guarantee user label uniqueness across all types of TC profiles. In Prime Optical 10.7, the forceUniqueness parameter is not implemented.
The following table describes the usage of the TCProfile_T and TCProfileCreateData_T fields.
The maximum attribute length is 254 characters.

Field	Valid Values	Description
additionalCreationInfo	globaldefs::NVSList_T	—
additionalInfo	globaldefs::NVSList_T	Same as additionalCreationInfo, but omits the first tuple TCProfile.
name	globaldefs::NamingAttributes_T	A name for the object that is unique in the EMS.
nativeEMSName	String	Same as userLabel.
owner	String	Not used. The EMS always sets the value to an empty string. If the NMS passes a different value to the EMS, the EMS returns an exception.
transmissionParams	transmissionParameters:: LayeredParameterList_T	Not used. The EMS always sets the value to an empty string. If the NMS passes a different value to the EMS, the EMS returns an exception.
userLabel	String	The human-readable object name, as displayed in Prime Optical user interface.

The additionalInfo and additionalCreationInfo fields contain a list of tuples with values specified as per the tables below. Each tuple consists a name-value pair. Tuples in the list can be in any order, unless otherwise specified. Additional parameter names are separated by a single space.
The following table describes the additionalInfo and additionalCreationInfo parameters for the class map.

Name	Type	Description

tcprofile	class-map	Mandatory in additionalCreationInfo and must be omitted from additionalInfo. The maximum length is 39 characters.
match-all	String	Valid values are: • True—If all parameters match the QoS criteria. • False—If at least one parameter matches the QoS criteria.
cos	Integer	Optional. Multiple values must be separated by a single space.
ip dscp	Integer	Optional. Multiple values must be separated by a single space. Must be omitted if an ip precedence tuple exists.
ip precedence	Integer	Optional. Multiple values must be separated by a single space. Must be omitted an ip dscp tuple exists.
mpls experimental topmost	Integer	Optional. Multiple values must be separated by a single space.
qos-group	Integer	Optional. Multiple values must be separated by a single space.
vlan	String	List of allowed VLAN identifiers. The list can be: • Numbers separated by a comma. • Ranges—Numbers in a range must be separated by a hyphen (-).

The following table describes the additionalInfo and additionalCreationInfo parameters for action.

Name	**Type**	**Description**
tcprofile	action	Mandatory in additionalCreationInfo and must be omitted from additionalInfo.
cos	Integer	Optional.
ip dscp	Integer	Optional.
ip precedence	Integer	Optional.
qos-group	Integer	Optional.
discard-class	Integer	Optional.
rate	single-rate dual color cir| single-rate dual color pir| single-rate three color|dual-rate three color	Optional. If you do not provide, you must not provide the following tuples: • cir • pir • bc • be
cir	bps|kbps|mbps|gbps|%	If the rate is not single-rate dual color pir, cir is mandatory; otherwise, you must omit it.

Name	Type	Description
pir	bps|kbps|mbps|gbps|%	If the rate is single-rate dual color pir or dual-rate three color, pir is mandatory; otherwise, you must omit it.
bc	bytes|kbytes|mbytes|gbytes|ms|us	If the rate is single-rate dual color cir or single-rate three color, bc is mandatory; otherwise, you must omit it.
be	bytes|kbytes|mbytes|gbytes|ms|us	If the rate is single-rate dual color pir or single-rate three color, be is mandatory; otherwise, you must omit it.
conform-action drop	Empty string	Optional.
conform-action transmit	Empty string	Optional.
conform-action set-prec-transmit	Integer	Optional.
conform-action set-dscp-transmit	Integer	Optional.
conform-action set-qos-transmit	Integer	Optional.
conform-action set-cos-transmit	Integer	Optional.
conform-action set-discard-class-transmit	Integer	Optional.
exceed-action drop	Empty string	Optional.
exceed-action transmit	Empty string	Optional.
exceed -action set-prec-transmit	Integer	Optional.
exceed -action set-dscp-transmit	Integer	Optional.
exceed -action set-qos-transmit	Integer	Optional.
exceed -action set-cos-transmit	Integer	Optional.
exceed -action set-discard-class-transmit	Integer	Optional.
violate -action drop	Empty string	Optional.
violate -action transmit	Empty string	Optional.
violate -action set-prec-transmit	Integer	Optional.
violate -action set-dscp-transmit	Integer	Optional.

Name	Type	Description
violate -action set-qos-transmit	Integer	Optional.
violate -action set-cos-transmit	Integer	Optional.
priority	blank|{Integer %|kbps|level}	Priority or shape average; you cannot provide both.
shape average	bps|kbps|mbps|gbps|%	Shape average or priority; you cannot provide both.
bandwidth	kbps|%	If you provide shape average, bandwidth or bandwidth remaining must be provided. You cannot provide both. If you do not provide shape average, omit bandwidth.
bandwidth remaining	ratio|%	If you provide shape average, bandwidth or bandwidth remaining must be provided. You cannot provide both. If you do not provide shape average, omit bandwidth.

The following table describes the additionalInfo and additionalCreationInfo parameters for policy maps.

Name	Type	Description
tcprofile	policy-map	tcprofile is mandatory in additionalCreationInfo and must be omitted from additionalInfo. The maximum length is 39 characters.
action	String	User label of the action.
class-map	String	User label of the class map.
policy-map	String	User label of the policy map. It can be an empty string ("") and requires no child policy map.

"action," "class-map," and "policy-map" are optional. If one is provided, you can also repeat it. The respective values are the user labels of the action, class map, and policy map, and the values must be indicated.

The following table describes the additionalInfo and additionalCreationInfo parameters for table maps.

Name	Type	Description
tcprofile	table-map	tcprofile is mandatory in additionalCreationInfo and must be omitted from additionalInfo. The maximum length is 39 characters.
default	copy|Integer	—
qos-group	Integer	—
discard-class	Integer	—
to	Integer	—

"qos-group," "discard-class," and "to" are optional. If one is provided, you can also repeat it.

This section describes the following interfaces:

- 4.5.6.1 createTCProfile
- 4.5.6.2 deleteTCProfile
- 4.5.6.3 modifyTCProfile

4.5.6.1 createTCProfile

Synopsis

```
public void createTCProfile(
TCProfileCreateData_T   newTCProfileCreateData,
                                    TCProfile_THolder
      newTCProfile)
      throws ProcessingFailureException
```

Description

This interface is used to create a new TC profile on the server. TCProfileCreateData is passed as input, and the resulting TC profile is returned as a result or an exception is returned.

Use Case Description

The following describes how the system requests to create a new TC profile:

1. The NMS sends a request to the EMS to create a TC profile. The NMS provides the TC profile name and a list of bandwidth parameters.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.
   b. If the maximum number of TC profiles in the EMS has already been reached and the TC profile cannot be created, a Capacity Exceeded exception is raised.
3. If the request is valid, the EMS creates the TC profile.
4. The EMS sends out an object creation event notification.
5. The EMS replies with a success indication.

If the request is successful, the NMS receives an indication of the success and an object creation event notification.

If the request is unsuccessful, the NMS receives an exception as an indication of the failure.

Object Creation Notification Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_CREATION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data      [0] notificationId       = OCE-4
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=class-map classmap02
                     [2] objectType           = OT_AID
                     [2] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                     [3] emsTime              = 20110802124039.0Z
                     [4] neTime               =
                     [5] edgePoint            = false
                     [6] isPropagated         = false
Remainder of body:
null
```

The above notification shows:

- objectType is set to OT_AID
- objectTypeQualifier is set to OT_TRAFFIC_CONDITIONING_PROFILE

- Newly created TC profile name in objectName

Relevant Data Structures

```
TCProfileCreateData_T
      String                        userLabel
      boolean                 forceUniqueness
      String                        owner
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  additionalCreationInfo

TCProfile_THolder
TCProfile_T value

TCProfile_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
      boolean                 defaultProfile
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  additionalInfo
```

4.5.6.2 deleteTCProfile

Synopsis

```
public void deleteTCProfile( NameAndStringValue_T[] tcProfileName)
      throws ProcessingFailureException
```

Description

This interface deletes a TC profile on the server. This operation is idempotent. If the service is called with the name of a nonexistent TC profile, the operation succeeds. If a TC profile is deleted successfully, the EMS sends out an object deletion notification.

Object Deletion Notification Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_OBJECT_DELETION
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data       [0] notificationId      = ODE-8
                      [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=policy-map policymap01
                      [2] objectType          = OT_AID
                      [2] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                      [3] emsTime             = 20110802130447.0Z
```

```
                        [4] neeTime              =
                        [5] edgePoint            = false
                        [6] isPropagated         = false
Remainder of body:
null
```

The above notification shows:

- objectType is set to OT_AID
- objectTypeQualifier is set to OT_TRAFFIC_CONDITIONING_PROFILE
- Deleted TC profile name in objectName

4.5.6.3 modifyTCProfile

Synopsis

```
public void modifyTCProfile(NameAndStringValue_T[] tcProfileName,
                                TCProfileCreateData_T tcProfileModifyData,
                                TPDataList_THolder tpsToModify,
                                TCProfile_THolder modifiedTCProfile,
                                StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface modifies an existing TC profile as specified by the parameters in the method. A TC profile with the data to be changed is passed as input. The resulting TC profile is returned as a result. If the TC profile modification completes successfully, the EMS sends out an AVC event notification.

Object Modification Notification Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000


Filterable Data       [0] notificationId       = AVC-21
                      [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] tcprofile=table-map tablemap01mod
                      [2] objectType           = OT_AID
                      [3] objectTypeQualifier = OT_TRAFFIC_CONDITIONING_PROFILE
                      [4] emsTime              = 20110802130158.0Z
                      [5] neeTime              =
                      [6] edgePoint            = false
                      [7] attributeList        = [0]userLabel=tablemap01
[1]additionalInfo=additionaInfo changed
                       [8] isPropagated         = false
Remainder of body:
null
```

The above notification shows:

- objectType is set to OT_AID
- objectTypeQualifier is set to OT_TRAFFIC_CONDITIONING_PROFILE
- objectName contains the TC profile name before the object was changed
- atributeList can contain up to two tuples:
  - userLabel—If the object name has changed, userLabel provides the new user label.
  - additionalInfo—Contains the indication "additionalInfo changed" if the object additionalInfo has changed.

4.6 MPLS-TP Provisioning Interfaces

This section describes the following MPLS-TP provisioning interfaces:

4.6.1 MPLS-TP Tunnel Provisioning Interfaces

This section describes the following interfaces:

4.6.1.1 createAndActivateSNC

Synopsis

```
public void createAndActivateSNC(
      SNCCreateData_T                 createData,
      GradesOfImpact_T                tolerableImpact,
      EMSFreedomLevel_T               emsFreedomLevel,
      TPDataList_THolder              tpsToModify,
      SubnetworkConnection_THolder    theSNC,
      StringHolder                    errorReason)
      throws ProcessingFailureException
```

Description

This interface allows you to create and activate a subnetwork connection for the TP tunnel using a single command. According to TMF standards, an SNC name is assigned by the EMS when it creates an SNC. However, if the NMS must control the SNC name, it can provide the name using the additionalInfo parameter sncName.

The layer rate that applies to TP tunnel circuits is LR_MPLS (165).

When you create an MPLS-TP tunnel, the routing is linked to the associated LSPs whenever the MPLS-TP tunnel is created with layer rate LR_MPLS_PATH.

Parameters

- SNCCreateData createData—Describes the SNC structure to be created and activated.
- GradesOfImpact_T tolerableImpact—The maximum tolerable impact allowed.
- EMSFreedomLevel_T emsFreedomLevel—The maximum level of freedom allowed to the EMS to perform the creation and activation.

- TPDataList_T tpsToModify—A list of TPs and parameters that are applied and updated to provide the resulting parameters.
- SubnetworkConnection theSNC—The resulting SNC with the sncState and name set. The EMS selects the SNC names so that they are not reused for different SNCs.
- string errorReason—Specifies the creation and activation errors, if any.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the createData field is invalid.

EXCPT_OBJECT_IN_USE - Raised if the SNC cannot be created and activated because the given tunnel number is already in use for the given endpoint ME.

EXCPT_ENTITY_NOT_FOUND - Raised when createData or tpsToModify fields reference objects that do not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request because at least one of the parameters, although valid, cannot be set and when the EMS cannot determine the reason it could not comply.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost and this prevents SNC creation.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met.

EXCPT_NOT_IN_VALID_STATE - Raised where the SNC would involve a CTP that is not reachable due to the server TP state.

Relevant Data Structures

SNCCreateData_T:

string userLabel:

The value entered in this field will be the name of the newly created SNC. This is the TP tunnel name.

boolean forceUniqueness:

Prime Optical guarantees the uniqueness if all SNCs are created by Prime Optical. If any other tool is used to create an SNC, Prime Optical cannot guarantee uniqueness.

string owner;

Not supported.

globaldefs::ConnectionDirection_T direction:

For MPLS-TP tunnels, the value is always CD_BI (bidirectional).

StaticProtectionLevel_T staticProtectionLevel:

For MPLS-TP tunnels, it can be UNPROTECTED or FULLY_PROTECTED.

ProtectionEffort_T protectionEffort:

Not considered for MPLS-TP tunnels.

Reroute_T rerouteAllowed:

Not supported.

NetworkRouted_T networkRouted:

Not supported.

SNCType_T sncType:

Not supported.

transmissionParameters::LayerRate_T layerRate:

The layer rate for MPLS-TP tunnels is LR_MPLS (165).

CrossConnectList_T ccInclusions:

Not supported.

ResourceList_T neTpInclusions:

When creating the MPLS-TP tunnels, routing is taken care by specifying either a list of ME or topological link names to be included in the LSP SNC route.

boolean fullRoute;

Not supported.

ResourceList_T neTpSncExclusions;

When creating the MPLS-TP tunnels, routing is taken care by specifying either a list of ME or topological link names to be included in the LSP SNC route.

globaldefs::NamingAttributesList_T aEnd;

The CTP name reports the tunnel number associated to the given aEnd point in the following form:

"tunnelNum=ddd"

“ddd” represents the integer number provided.

MPLS-TP tunnel endpoints at low level (Cisco IOS interfaces) is not located on physical ports, but on Virtual Tunnel Interfaces (VTIs) located in a given card. Even though you specify a PTP for the tunnel endpoints, the relevant information of it will be the slot in the shelf where the card is equipped.For example:

name="EMS";value="Cisco Systems/PRIMEOPTICAL"

name="ManagedElement";value="M2-65-122"

name="PTP";value="/rack=1/shelf=1/slot=2/ppm_holder=1/port=1"

name="CTP";value="tunnelNum=44"

globaldefs::NamingAttributesList_T zEnd;

Same as aEnd field.

globaldefs::NVSList_T additionalCreationInfo;

The following table describes the additional parameters that you can specify.

Name	Value	Description
srcBFD	String	BFD template name of the source endpoint of the MPLS-TP tunnel. If the provided BFD name does not appear among the available names associated to the source ME, a corresponding exception is generated, reporting the actual list of available names. If the source BFD is not provided, the EMS picks up the first available name in the list associated to the given source ME.
destBFD	String	BFD template name of the destination endpoint of the MPLS-TP tunnel. If the provided BFD name does not appear among the available names associated to the destination ME, a corresponding exception is generated, reporting the actual list of available names. If the destination BFD is not provided, the EMS picks up the first

Name	Value	Description
		available name in the list associated to the given destination ME.
txBW	Long	Transmission bandwidth associated to the MPLS-TP tunnel.
rxBW	Long	Receiving bandwidth associated to the MPLS-TP tunnel.
txBWunits	String	Valid values are: • Kbps • Mbps • Gbps
txBWunits	String	Valid values are: • Kbps • Mbps • Gbps
ServiceID	Integer	Optional.
		It is recommended that you not specify the ServiceID.
		The EMS automatically searches the network for the first Service ID (SID) number available between the MEs at the TP tunnel endpoints. To successfully provision the TP tunnel, the same SID must be available on both MEs. SID values on a given ME are shared between TP tunnels, pseudowires, and EVCs passing through it. The SID value automatically chosen by the EMS is returned as additional parameter in the SNC structure for the TP tunnel. If you provide the ServiceID for the given TP tunnel, the EMS only checks if it is available on both the source and destination MEs. If the ServiceID is not available, an exception is generated.
sncName	String	User-defined SNC name. If no sncName is defined, the EMS automatically assigns the following: tpTunnel:<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID> The node IDs are automatically retrieved by the EMS from the TP tunnel endpoint MEs, and the tunnel numbers are provided through the endpoint CTP names; for example, tpTunnel:3.3.3.0:55:5.5.5.0:66.

Limitations

Object create notifications are not generated.

4.6.1.2 deactivateAndDeleteSNC

Synopsis

```
public void deactivateAndDeleteSNC(
NameAndStringValue_T[]  sncName,
                                        GradesOfImpact_T
      tolerableImpact,
                                        EMSFreedomLevel_T      emsFreedomLevel,
                                        TPDataList_THolder           tpsToModify,
                                        SubnetworkConnection_THolder theSNC,
                                        StringHolder errorReason)
      throws ProcessingFailureException
```

Description

This interface allows you to deactivate and then delete the SNC, which represents a given MPLS-TP tunnel in one operation. This operation implies the automatic deletion of the associated LSP SNCs. Prime Optical removes resources allocated to this SNC from each node. This is an asynchronous operation. Successful completion of this operation guarantees only the delivery of deletion request to Prime Optical. The NMS must wait for an OBJECT_DELETION event for this SNC. See also 3.9.20 deactivateAndDeleteSNC
Because TP tunnel names are not guaranteed to be unique in the network, you must provide the TP tunnel name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66.
The TKEY represents the TP tunnel key and is returned as an additional parameter whenever the corresponding SNC is retrieved. This key is composed in the following format:
<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID>
The TKEY represents an invariant component for every TP tunnel. A prevention check is done in case any pseudowire is configured upon the given MPLS-TP tunnel. If any pseudowire is configured, an exception is generated and the pseudowire is removed in advance.

Limitations
Object delete notifications are not generated.

4.6.1.3 modifySNC

Synopsis

```
public void modifySNC(
NameAndStringValue_T[]  sncName,
                        String                          routeId,
                        SNCModifyData_T         SNCModifyData,
                        GradesOfImpact_T        tolerableImpact,
                        ProtectionEffort_T              tolerableImpactEffort,
                        EMSFreedomLevel_T       emsFreedomLevel,
                        TPDataList_THolder              tpsToModify,
                        SubnetworkConnection_THolder newSNC,
                        StringHolder                    errorReason)
      throws ProcessingFailureException
```

Description
This interface allows the modification of the circuit user label for the given TP tunnel SNC. Because TP tunnel names are not guaranteed to be unique in the network, you must provide the TP tunnel name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66.
The TKEY represents the TP tunnel key and is returned as an additional parameter whenever the corresponding SNC is retrieved. This key is composed in the following format:
<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID>
The TKEY represents an invariant component for every TP tunnel.
Relevant Data Structures
SNCModifyData_T

```
String                          userLabel
boolean                         forceUniqueness
String                          owner
ConnectionDirection_T           direction
String                          modifyType
boolean                         retainOldSNC
boolean                         modifyServers_allowed
StaticProtectionLevel_T         staticProtectionLevel
ProtectionEffort_T              protectionEffort
Reroute_T                       rerouteAllowed
NetworkRouted_T                 networkRouted
SNCType_T                       sncType
short                           layerRate
RouteDescriptor_T[]             addedOrNewRoute
RouteDescriptor_T[]             removedRoute
NameAndStringValue_T[][]        neTpInclusions
boolean                         fullRoute
NameAndStringValue_T[][]        neTpSncExclusions
NameAndStringValue_T[][]        aEnd
NameAndStringValue_T[][]        zEnd
NameAndStringValue_T[]          additionalCreationInfo
```

additionalCreationInfo allows TP Tunnel bandwidth modification.
The following table lists the parameters.

Name	Value	Comment/Example
BW	String	String that represents a numeric value. It corresponds to the TP Tunnel bandwidth value. and is measured by the BWunits parameter. If the parameter is omitted, the default unit of measure is Kbps.
BW Units	String	Bandwidth unit of measure. Valid values are: • Kbps • Mbps • Gbps

Limitations
Object modify notifications are not generated.

4.6.2 MPLS-TP LSP Provisioning Interfaces

This section describes the following interfaces:

- 4.6.2.1 createAndActivateSNC
- 4.6.2.2 deactivateAndDeleteSNC

- 4.6.2.3 modifySNC

4.6.2.1 createAndActivateSNC

Synopsis

```
public void createAndActivateSNC(
SNCCreateData_T                 createData,
                                      GradesOfImpact_T
      tolerableImpact,
                                      EMSFreedomLevel_T
      emsFreedomLevel,
                                      TPDataList_THolder
      tpsToModify,
                                      SubnetworkConnection_THolder  theSNC,
                                      StringHolder
      errorReason)
      throws ProcessingFailureException
```

Description

This interface allows you to create and activate a subnetwork connection (an LSP) in a single command. As per TMF standards, an SNC name is assigned by the EMS when it creates an SNC. However, if the NMS must control the SNC name, you can use the additionalInfo parameter sncName.

The LSP name must match the TP tunnel circuit name. As TP tunnel names are not guaranteed to be unique in the network, you must provide the TP tunnel name in the following format:

<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>

For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66.

The TKEY represents the TP tunnel key and is returned as an additional parameter whenever retrieving the corresponding SNC. The layer rate that applies to the LSP circuits is LR_MPLS_PATH (166), which provides MPLS path layering to the TP tunnel entity.

✐ Local and out labels in the endpoints and midpoints are automatically assigned by the network. To modify them, use the modifySNC API.

The EMS must guarantee that the same SID is shared among all the MEs traversed by all TP tunnel LSPs. If the route proposed by the EMS for the new LSP crosses MEs that are different from the ones currently belonging to the TP tunnel, the EMS must verify if the same SID is also available on the new MEs. If the same SID is not available, the EMS must pick up a new SID and reassign to the same MEs.

Relevant Data Structures

```
SNCCreateData_T:

string userLabel:

Optional. The user label that is provided. If it is empty, EMS automatically
assigns the default label LSP.

boolean forceUniqueness:

Prime Optical guarantees uniqueness if all SNCs are created by Prime Optical.
If any other tool is used to create an SNC, Prime Optical cannot guarantee
uniqueness.

string owner;

Not supported.

globaldefs::ConnectionDirection_T direction:

For LSP circuits, the value is always CD_BI (bidirectional).

StaticProtectionLevel_T staticProtectionLevel:
```

For LSP circuits, it is UNPROTECTED.

ProtectionEffort_T protectionEffort:

Not considered for LSP circuits.

Reroute_T rerouteAllowed:

Not supported.

NetworkRouted_T networkRouted:

Not supported.

SNCType_T sncType:

Not supported.

transmissionParameters::LayerRate_T layerRate:

The layer rate for MPLS-TP tunnels is LR_MPLS_PATH (166).

CrossConnectList_T ccInclusions:

Not supported.

ResourceList_T neTpInclusions:

You can specify a list of ME or topological link names to be included in the LSP SNC route. For complete manual routing, you must pass the list of all MEs or topological links through which the circuit is desired to be routed to.

boolean fullRoute;

Not supported.

ResourceList_T neTpSncExclusions;

Specify either a list of MEs or topological link names to be excluded in the LSP SNC route.

globaldefs::NamingAttributesList_T aEnd;

The CTP name reports the LSP number associated to the given aEnd point in the following format:

"lspNum=ddd"

"ddd" represents the integer number provided.

Example:

name="EMS";value="Cisco Systems/PRIMEOPTICAL"

name="ManagedElement";value="M2-65-122"

name="PTP";value="/rack=1/shelf=1/slot=2/ppm_holder=1/port=1"

name="CTP";value="lspNum=0"

globaldefs::NamingAttributesList_T zEnd;

Same as the aEnd field. LSP numbers at the endpoints always match one another.

globaldefs::NVSList_T additionalCreationInfo;

You can specify the following additional parameters:

Name	Value	Description
isProtLsp	Boolean	• True, if the LSP created is protected. • False, if the LSP created is working.

Name	Value	Description
sncName	String	The defined SNC name, which must match the TP tunnel SNC name that this LSP will be associated to. The name must be in the following format: <TP-Tunnel native name>:<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID> For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66.

Limitations

Object create notifications are not generated.

4.6.2.2 deactivateAndDeleteSNC

Synopsis

```
public void deactivateAndDeleteSNC(
NameAndStringValue_T[]  sncName,
                                        GradesOfImpact_T
      tolerableImpact,
                                        EMSFreedomLevel_T      emsFreedomLevel,
                                        TPDataList_THolder           tpsToModify,
 NameAndStringValue_T[]       additionalInfo,
                                        SubnetworkConnection_THolder theSNC,
                                        StringHolder errorReason)
      throws ProcessingFailureException
```

Description

See also 3.9.20 deactivateAndDeleteSNC

This interface enables you to deactivate and then delete the LSP SNC in one operation. Prime Optical removes resources allocated to this SNC from each node. This is an asynchronous operation. Successful completion of this operation guarantees only the delivery of the deletion request to Prime Optical. The NMS must wait for an OBJECT_DELETION event for this SNC.

As TP tunnel names are not guaranteed to be unique in the network, you must provide the LSP name in the following format:

<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>:LSP=<LSP number>

For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66:LSP=0.

The TKEY represents the TP tunnel key and is returned as an additional parameter whenever retrieving the corresponding SNC. This key is composed as follows:

<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID>

The TKEY represents an invariant component for every TP tunnel. To differentiate among the working and protected LSPs of the TP tunnel, the LSP number and the last token of the composite name is provided.

Limitations

Object delete notifications are not generated.

4.6.2.3 modifySNC

Synopsis

```
public void modifySNC(
NameAndStringValue_T[]  sncName,
                        String                          routeId,
```

```
                        SNCModifyData_T         SNCModifyData,
                        GradesOfImpact_T        tolerableImpact,
                        ProtectionEffort_T            tolerableImpactEffort,
                        EMSFreedomLevel_T       emsFreedomLevel,
                        TPDataList_THolder      tpsToModify,
                        SubnetworkConnection_THolder newSNC,
                        StringHolder                  errorReason)
      throws ProcessingFailureException
```

Description

This interface allows the modification of the following information for the given TP tunnel SNC:

- Local label and out label in the endpoints and midpoints.
- Lockout in the endpoints.

This information is provided as layered parameters associated to the LR_MPLS in transmissionParams of the tpsToModify input parameter. As TP tunnel names are not guaranteed to be unique in the network, you must provide the LSP name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>:LSP=<LSP number>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66:LSP=0.

Relevant Data Structures

```
SNCModifyData_T
      String                        userLabel
      boolean                 forceUniqueness
      String                        owner
      ConnectionDirection_T   direction
      String                        modifyType
      boolean                 retainOldSNC
      boolean                 modifyServers_allowed
      StaticProtectionLevel_T       staticProtectionLevel
      ProtectionEffort_T            protectionEffort
      Reroute_T               rerouteAllowed
      NetworkRouted_T         networkRouted
      SNCType_T               sncType
      short                         layerRate
      RouteDescriptor_T[]           addedOrNewRoute
      RouteDescriptor_T[]           removedRoute
      NameAndStringValue_T[][]      neTpInclusions
      boolean                 fullRoute
      NameAndStringValue_T[][]      neTpSncExclusions
      NameAndStringValue_T[][]      aEnd
      NameAndStringValue_T[][]      zEnd
      NameAndStringValue_T[]  additionalCreationInfo
TPDataList_THolder
```

```
TPData_T[]                value
TPData_T
      NameAndStringValue_T[]  tpName
      TerminationMode_T             tpMappingMode
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  ingressTrafficDescriptorName
      NameAndStringValue_T[]  egressTrafficDescriptorName
LayeredParameters_T
      short                         layer
      NameAndStringValue_T[]  transmissionParams
```

The following table describes the layered parameters that can be modified on TPs. The TPs can be either midpoints or endpoints of the given LSP circuit.

Name	Value	Description
fwLocalLabel	Integer	Local label value associated to the forward path of the LSP.
fwOutLabel	Integer	Out label value associated to the forward path of the LSP.
rvtLocalLabel	Integer	Local label value associated to the reverse path of the LSP.
rvtOutLabel	Integer	Out label value associated to the reverse path of the LSP.

The following table describes the layered parameter that can be modified on endpoints of the given LSP circuit.

Name	Value	Description/Valid Value
switchState	String	New administrative switch state: LOCKOUT or CLEAR.

Limitations

Object modify notifications are not generated.

4.6.3 MPLS-TP Protection (Provisioning)

In the PTF_10GE_4, PT_10GE_4, and PTSA_GE equipment, the TP tunnel circuit names are not unique. The following pair represents their primary key:
<sncName, SID>
The component LSP circuits are uniquely identified by:
<sncName, SID, uniqueID>
sncName and SID are the same as the TP tunnel key.
When creating an LSP circuit, if you provide the circuit name, SID, and Unique ID through additional parameters, the association with the corresponding TP tunnel is automatically established.

4.6.4 BFD Template, Node ID, and Label Range Provisioning Interface

When creating MPLS-TP tunnels, you can leave some parameters with the default value or you can specify it. If you specify it, you should retrieve and specify the valid range of values at the node level.
For the BFD template and the static label range, you should change the values that are currently specified for the involved ME. To achieve this, because all these parameters are seen as additional parameters at the ME level, you must call the standard setAdditionalInfo API.
This section describes the following interface:

- 4.6.4.1 setAdditionalInfo

4.6.4.1 setAdditionalInfo

Synopsis

```
public void setAdditionalInfo(NameAndStringValue_T[] objectName,
```

```
                                    NVSList_THolder additionalInfo)
        throws ProcessingFailureException
```

Description
This interface allows you to set additional information associated to an ME. Provide the name as the input parameter.
The following table is an example of the BFD template report.

BFD Name	Single Hop	Unit	minTxInterval	minRxInterval	Multiplier
Bfd1	True	Millisec	300	400	3
Bfd2	True	Millisec	500	600	6
Bfd3	True	Microsec	700000	800000	9

GateWay/CORBA NBI Result

```
bfdTable.Count =                    3
bfdTable.Name=                      "Bfd1, BFD2, BFD3"
bfdTable.SingleHop=                 "true, true, true"
bfdTable.Unit=                      "Millisec, Millisec, Microsec"
bfdTable.MinTxInterval=             "300, 500, 700000"
bfdTable.MinRxInterval=             "400, 600, 800000"
bfdTable.Multiplier=                "3, 6, 9"
```

You can make multiple changes in the same table, because the overall table values are provided in provisioning. To change any field in the table, the entire table must be regenerated again. The EMS analyzes the table content and provisions the necessary changes in the NE.
You can do the following operations:

- Delete a row—Do not report the row with the given BFD name in the table.
- Add a row—Report a new row with a new BFD name in the table
- Modify a row—For a row corresponding to a given BFD, change any column value. The BFD name represents the primary key.
- Delete the whole table—Set bfdtable.Count=0 and all column parameters to an empty string ("").

The routeIdName parameter is read-only. The routeID parameter can only be modified; it cannot be created or deleted.

4.7 Pseudowire Provisioning Interfaces

This section describes the following pseudowire provisioning interfaces:

- 4.7.1 Pseudowire FDFr Provisioning Interfaces
- 4.7.2 Pseudowire Mixed Configuration (Provisioning)
- 4.7.3 Pseudowire Class and Loopback Provisioning Interface
- 4.7.4 Pseudowire QoS Provisioning Interface

4.7.1 Pseudowire FDFr Provisioning Interfaces

This section describes the following interfaces:

- 4.7.1.1 createAndActivateFDFr
- 4.7.1.2 deactivateAndDeleteFDFr
- 4.7.1.3 modifyFDFr

4.7.1.1 createAndActivateFDFr

Synopsis

```
public void createAndActivateFDFr(
FDFrCreateData_T              createData,
                                    ConnectivityRequirement_T
      connectivityRequirement,
                                    NamingAttributesList_THolder  aEnd,
                                    NamingAttributesList_THolder  zEnd,
                                    NamingAttributesList_THolder  internalTPs,
                                    MatrixFlowDomainFragmentList_THolder
mfdfrs,
                                    TPDataList_THolder
      tpsToModify,
                                    FlowDomainFragment_THolder    theFDFr,
                                    NamingAttributesList_THolder
      notConnectableCPTPList,
                                    NamingAttributesList_THolder
      parameterProblemsTPList,
                                    StringHolder errorReason)
      throws ProcessingFailureException
```

Description

The NMS invokes the createAndActivateFDFr service to request the EMS to create and activate a pseudowire with the parameters specified in the method.

Parameters

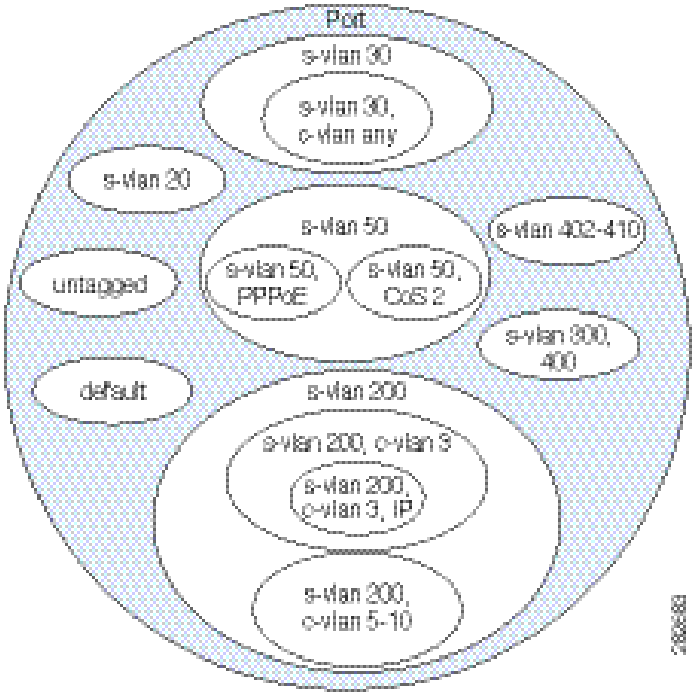
- FDFrCreateData_T createData—Describes the structure of the FDFr to be created. Unless specified otherwise, connectionless parameter settings are on a best-effort basis.
- ConnectivityRequirement_T connectivityRequirement—For a connectivity-aware EMS, this parameter identifies the requested operation mode if not all FPs have potential connectivity to one another. If the EMS is not connectivity-aware, ignore this parameter.
- NamingAttributesList_T aEnd—The source CTP/EFP based on a CPTP. The CTP name implicitly provides the AC type and VLAN configuration. Pseudowires always have only one aEnd point. The name of each aEnd must be in the following format:
  name="EMS";value="CompanyName/EMSname"
  name="ManagedElement";value="name"
  name="PTP";value="/rack/shelf/slot/ppm_holder=m/port=n"
  name="CTP";value="CTP Name"

For PTSA_GE endpoints, the PTP name must be in the following format:
name="PTP";value="/remote_unit=1/shelf=1/slot=36/ppm_holder=9/port=1"
Slot 36 is mapped to the PTSA_GE remote equipment.
The following table describes the possible use cases with the associated CTP names.

VLAN Type	CTP Native Name	Valid Tag Values
Single tagged	/ethvid=n	VLAN tag is a numeric value or a range from 1 to 4094; for example: • /ethvid=5 • /ethvid=5,10-13
Double tagged	/ethsvid=n/ethcvid=m	First (outer or S-VID) VLAN tag must be unique. Second (inner or C-VID) VLAN tag is any unique list or range; for example:

VLAN Type	CTP Native Name	Valid Tag Values
		• /ethsvid=10/ethcvid=45 • /ethsvid=12/ethcvid=45-50,61
Untagged	/eth=untagged	Match untagged frames.
Default	/eth=default	• VLAN unaware • Port-based drop
Untagged and single tagged	/eth=untagged/ethvid=n	Match untagged and single tagged frames. A valid value can be a numeric value or a range from 1 to 4094; for example: • /eth=untagged/ethvid=5 • /eth=untagged/ethvid=5,10-13

Figure 4-7: Use Cases with Associated CTP Names



For channel group (Ethernet Link Aggregation), the CTP refers to the FTP representing the logical interface for the LAG rather than a physical PTP. The CTP name should be in the following format:

name="EMS";value="CompanyName/EMSname"
name="ManagedElement";value="name"
name="FTP";value="/rack/shelf/port=n"
name="CTP";value="CTP Name"

Parameters

- globaldefs::NamingAttributesList_T zEnd—Same as aEnd. For dual-homed protected pseudowires, there are two zEnd TPs.
- NamingAttributesList_T internalTPs—Not used. A list of internal CPTP names that must be included in the route of the FDFr. As a result of creating the FDFr, FPs are created as clients of internal CPTPs.
- MatrixFlowDomainFragmentList_T mfdfrs—Not used. An optional list of MFDFrs that represents the route of the FDFr. You can omit this attribute if the FDFr is routed by the network. As a result of creating the FDFr, MFDFrs are created in the various MFDs.

- TPDataList_T tpsToModify—A list of TPs and parameters to be applied. When the method is returned, the list is updated to provide the resulting parameters. The list may refer to FPs that are being created during the createAndActivateFDFr request or to the containing CPTPs.
- flowDomainFragment::FlowDomainFragment_T theFDFr—The newly created and activated FDFr. The EMS is responsible for guaranteeing uniqueness of the name of the FDFr.
- NamingAttributesList_T notConnectableCPTPList—The list of CPTPs that could not be connected; for example, CPTPs that are not part of the same FD.
- NamingAttributesList_T parameterProblemsTPList—The list of CPTPs and FPs for which only best-effort transmission parameters could not be set.
- string errorReason—If a best-effort parameter could not be set or an FP could not be connected, the EMS indicates the fault reason.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the createData field is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when one of the provided CPTPs does not exist.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met. The specified user label is currently in use.

EXCPT_NE_COMM_LOSS - Raised in case of communication failure between the EMS and the MEs.

EXCPT_NOT_IN_VALID_STATE - Raised when the state of the specified object is such that the EMS cannot perform the request.

EXCPT_TP_INVALID_ENDPOINT - Raised when the specified TP does not exist or cannot be created.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request because of one of the following conditions:

1. The name specified by the NMS already exists in the EMS.

2. Unrecognized mapping criteria.

3. Frames map to more than one FDFr.

4. The FDFr has less than two FPs.

Relevant Data Structures

FDFrCreateData_T

```
      NameAndStringValue_T[]  name
      String                        userLabel
      boolean                 forceUniqueness;
      String                        owner
      String                        networkAccessDomain
      ConnectionDirection_T   direction
                              For Ethernet, the directionality is always
bidirectional.
      AdministrativeState_T   administrativeState
      LayeredParameters_T     transmissionParams
```

The layer rate is LR_PW (167). The associated layered parameters are described in the table below.

boolean fullRoute

Always considered "false" by the EMS because the routing is done automatically by the network.

String fdfrType

The possible values are "point-to-point," "point-to-multipoint," "multipoint." For pseudowire, only point-to-point is allowed.

NameAndStringValue_T[] additionalCreationInfo

The following table describes the transmission parameters associated to the new LR_PW.

Name	Type	Description	Mandatory
pwType	String	Valid values are: • Ethernet • VLAN	Yes
bw	Long	Receiving bandwidth associated to the MPLS-TP tunnel.	Yes
bwUnits	String	Valid values are: • Kbps • Mbps • Gbps	Yes
isRedundant	Boolean	Valid values are: • True • False If the pseudowire is protected, the value must be true. If true, specify values for the aEnd and zEnd of both the pwClass and bkpPwClass corresponding to the two TP tunnels that provide pseudowire redundancy. The two TP tunnels must be routed across different paths to reach the same endpoints.	Yes
isDualHomedPeer	Boolean	Mandatory if isRedundant is set to true. If true, specify the aEnd of both pwClass and bkpPwClass corresponding to the two TP tunnels that provide dual-homed pseudowire redundancy. For the two zEnd points, only pwClass is required. The two TP tunnels must have the same source node, but distinct destination nodes. Valid values are: • True • False	No
enableDelayValue	Integer	Enable delay value. Mandatory if isRedundant is set to true.	No
isDisableDelay	Boolean	If disable delay is set, the valid value is true. Mandatory if isRedundant is set to true. Valid values are: • True • False	No
disableDelayValue	Integer	Disable delay value. Mandatory if isRedundant is set to true.	No
isNeverDisDelay	Boolean	Mandatory if isRedundant is set to true. Valid values are: • True	No

Name	Type	Description	Mandatory
		• False	

The tpsToModify parameter is defined as:

```
TPDataList_THolder
TPData_T[]              value


TPData_T
      NameAndStringValue_T[]  tpName


A PTP or FTP which corresponds to a given managed endpoint of the pseudowire.
A pseudowire can have only one EFP from a given CPTP.
1. name="EMS";value="CompanyName/EMSname"
2. name="ManagedElement";value="name"
3. name="PTP";value="/rack/shelf/slot/ppm_holder=m/port=n"


or


1. name="EMS";value="CompanyName/EMSname"
2. name="ManagedElement";value="name"
3. name="FTP";value="/rack/shelf/port=n"

TerminationMode_T             tpMappingMode
                              not handled (NA)
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  ingressTrafficDescriptorName
      NameAndStringValue_T[]  egressTrafficDescriptorName


LayeredParameters_T
      short                         layer
                              a new layer rate LR_PW (167)
      NameAndStringValue_T[]  transmissionParams
```

The following table describes the layered parameters for the layer rate LR_PW that is specified on TP endpoints of the pseudowire.

Name	Type	Description
outerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD

Name	Type	Description
		• TP_9100 • TP_9200
innerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 In Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
isExactTag	Boolean	If the VLAN tag is exact, set it to true.
rewriteOper	String	Valid values are: • REWRITE_NONE • PUSH_1 • PUSH_2 • POP_1 • POP_2 • TRANSLATE_1_TO_1 • TRANSLATE_1_TO_2 • TRANSLATE_2_TO_1 • TRANSLATE_2_TO_2
rewriteOuterVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200
rewriteOuterVlanTag	Integer	—
rewriteInnerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 In Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
rewriteInnerVlanTag	Integer	—
rewriteIsSymmetric	Boolean	Valid values are: • True • False
enableStatistics	Boolean	Enables statistics counter. Valid values are: • True • False
statisticsCategory	String	Valid values are:

Name	Type	Description
		• INGRESS • EGRESS • BOTH Mandatory only if enableStatistics is true. In Prime Optical 10.7, by default, INGRESS is always set to true.
manLoadBal	—	Not supported.
qosTableMap	String	QoS table map name (TC profile).
qosIngPolicy	String	QoS ingress policy map name (TC profile).
qosEgrPolicy	String	QoS egress policy name (TC profile).
pwClass	String	Pseudowire class name. This is accessible through the getManagedElement API additionalInfo. It allows association with the TP tunnel.
vcID	Integer	VC ID.
staticLabel	Integer	(Optional) Specify a static value for the local label. If it is not provided, the first available label is automatically picked up from the node.
bkpPwClass	String	Backup (protected) pseudowire class.
bkpVcID	Integer	Backup (protected) VC ID.
bkpStaticLabel	Integer	(Optional) Specify a static value for the protected local label. If it is not provided, the first available label is automatically picked up from the node.

The following table lists the mandatory information based on tagging type.

Name	Single Tagged	Double Tagged	Untagged	Default
outerVlanTpId	Yes	Yes	No	No
innerVlanTpId	No	Yes	No	No
isExactTag	No	No	No	No
rewriteOper	Yes	Yes	No	No
rewriteOuterVlanTpId	No	No	No	No
rewriteOuterVlanTag	No	No	No	No
rewriteInnerVlanTpId	No	No	No	No
rewriteInnerVlanTag	No	No	No	No
rewriteIsSymmetric	No	No	No	No
enableStatistics	Yes	Yes	Yes	Yes
statisticsCategory	No	No	No	No
qosTableMap	No	No	No	No
qosIngPolicy	No	No	No	No
qosEgrPolicy	No	No	No	No
pwClass	Yes	Yes	Yes	Yes
vcID	Yes	Yes	Yes	Yes
staticLabel	No	No	No	No
bkpPwClass	No	No	No	No
bkpVcID	No	No	No	No
bkpStaticLabel	No	No	No	No

The network automatically assigns the SID to the FDFr at the time of creation. This SID is returned in the name. For pseudowires, see the example below:

```
Flow Domain Fragment Name   =       [0] EMS=Cisco Systems/PrimeOptical;
[1] FlowDomain=CTM_NP_1;
[2] FlowDomainFragment=GIAN_PW_1:SID=25
Native EMS Name  =            GIAN_PW_1
```

When the creation API is executed, the service is not discovered for all parameters. The API returns a dummy object based on the input parameters. The actual objects are retrieved through the standard inventory APIs.

Limitations

Object create notifications are not generated.

4.7.1.2 deactivateAndDeleteFDFr

Synopsis

```
public void deactivateAndDeleteFDFr(
  NameAndStringValue_T[]      fdfrName,
                                      TPDataList_THolder    tpsToModify,
                                      StringHolder                errorReason)
      throws ProcessingFailureException
```

Description

This interface allows the NMS to request the deactivation and deletion of a pseudowire FDFr from an FD.

Parameters

- NamingAttributes_T fdfrName—The pseudowire name to be deactivated and deleted. The name must be in the following format: <PW native name>:SID=nn.
- TPDataList_T tpsToModify—A list of TPs and parameters to be applied. When the method is returned, the list is updated to provide the resulting parameters. The list may refer only to TPs that take part in the FDFr deletion process.
- string errorReason—Specifies the reason for the fault, if any.
- errorReason—Refers to an empty string if the operation is performed successful.

Throws

```
Raises ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically
incorrect. For example, fdfrName does not refer to an FDFr object.
EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an object that does
not exist.
EXCPT_ACCESS_DENIED - Raised if the EMS does not allow the NMS to delete the
FDFr. For example, the FDFr state is fixed.
EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

Use Case Description

The following describes how the system requests to deactivate and delete an FDFr:

1. The NMS sends the request to the EMS to deactivate and delete an FDFr.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.

   b. If the specified FDFr object is unknown to the EMS, an Entity Not Found exception is raised.
3. If the request is valid:
   a. The EMS initiates the deactivation of the FDFr. It attempts to remove, from the applicable MEs, all the MFDFrs which comprise this FDFr.
   b. If the EMS succeeds in deactivating the FDFr, it then deletes the FDFr object and all associated FPs.
   c. If some MFDFrs were removed and some were not, the FDFr is not deleted from the EMS, and it is marked as Failed.
4. The EMS provides a success indication to the NMS.
5. The EMS sends FP and FDFr object deletion notifications to the notification service.

If the request is successful:

1. The NMS receives a success indication.
2. An Object Deletion notification is sent.
3. For all MFDFrs which are removed, the associated FPs are deleted.
4. If all MFDFrs were removed, the FDFr has been deleted.

If the FDFr was not deleted, it is marked as Failed.

If the request is not successful, the NMS receives an exception as an indication of the failure of the request.

Limitations

Object delete notifications are not generated.

4.7.1.3 modifyFDFr

Synopsis

```
modifyFDFr(
 NameAndStringValue_T[]              fdfrName,
                         FDFrModifyData_T                    fdfrModifyData,
                         ConnectivityRequirement_T
      connectivityRequirement,
                         TPDataList_THolder                  tpsToModify,
                         NamingAttributesList_THolder        failedTPList,
                         NamingAttributesList_THolder
      parameterProblemsTPList,
                         FlowDomainFragment_THolder   newFDFr,
                         StringHolder                        errorReason)
      throws ProcessingFailureException
```

Description

This interface allows you to request the EMS to modify an existing FDFr as specified by the parameters in the method.

Parameters

- NameAndStringValue_T[] fdfrName—The FDFr name to be modified. To uniquely identify the FDFr, use the following format: <FDFr native name>:SID=34.
- FDFrModifyData_T fdfrModifyData—Describes the structure of the FDFr to be modified. Modification of connectionless parameters is done on a best-effort basis. For pseudowires, only UserLabel and administrative state can be changed.
- ConnectivityRequirement_T connectivityRequirement—Not supported for pseudowires.

- subnetworkConnection::TPDataList_T tpsToModify—A list of TPs and parameters to be applied. When the method is returned, the list is updated to provide the resulting parameters. The list may refer to the FPs or to the containing CPTPs.
- globaldefs::NamingAttributesList_T failedTPList—Not supported for pseudowires.
- globaldefs::NamingAttributesList_T parameterProblemsTPList—The list of CPTPs and FPs for which only best-effort transmission parameters could not be set.
- flowDomainFragment::FlowDomainFragment_T newFDFr—The modified FDFr.
- string errorReason—If a best-effort parameter could not be set, the EMS provides the fault reason.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when any input parameter is syntactically incorrect. For example, the fdfr ModifyData field is invalid.

EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an object that does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the EMS is unable to execute the request because at least one of the parameters, although valid, cannot be set.

EXCPT_USERLABEL_IN_USE - Raised when the userLabel uniqueness constraint is not met.

EXCPT_ACCESS_DENIED - Raised if the EMS does not allow the NMS to modify the FDFr. For example, the FDFr state is fixed.

EXCPT_NOT_IN_VALID_STATE - Raised when at least one of the specified CPTPs is in the unassigned PortTPRoleState.

Relevant Data Structures

FDFrModifyData_T

```
String                      userLabel
boolean                 forceUniqueness
String                      owner
String                      networkAccessDomain
AdministrativeState_T   administrativeState
LayeredParameters_T     transmissionParams
NameAndStringValue_T[][]    tpNamesToRemove
NameAndStringValue_T[][]    aEndTPNames
NameAndStringValue_T[][]    zEndTPNames
NameAndStringValue_T[][]    internalTPNames
NameAndStringValue_T[]  additionalModificationInfo
```

Use Case Description

The NMS can modify the user label, administrative state, and QoS information associated to the FDFR FPs.

The following describes how the system requests to modify an FDFr:

1. The NMS sends a request to the EMS to modify an FDFr.
2. The EMS validates the request:
   a. If the syntax is incorrect, an Invalid Input exception is raised.

b. If the specified FDFr object is unknown to the EMS, an Entity Not Found exception is raised.
c. If user label uniqueness is required, the EMS checks the user label for uniqueness. If an FDFr object with the same user label exists, a User Label In Use exception is raised.
d. If any of the TP objects specified for modification is unknown to the EMS, an Entity Not Found exception is raised.
e. If any of the TP objects specified in the modification is not associated with the specified FDFr, an Entity Not Found exception is raised.
f. If an unrecognized name exists in the QoS policy names provided for specified FPs, an Unable To Comply exception is raised.

3. If the request is valid, the EMS modifies the FDFr attributes and parameters, as requested.
4. The EMS returns a success indication.

Change Admin State

The administrative state change applies to the entire service only if tpsToModify or zEndTPNames is specified. If neither of these parameters is specified, the change applies only to those tpsToModify that have been specified and to the new drop specified. To change the administrative state, set the changeAdminState parameter to true in the additionalModificationInfo.

The following table describes the additionalModificationInfo parameter.

Name	Type	Description
changeAdminState	Boolean	Indicates whether the admin state can be changed based on the new value specified in the administrativeState field. Valid values are: • True • False

Limitations

Object modify notifications are not generated.

4.7.2 Pseudowire Mixed Configuration (Provisioning)

Prime Optical supports pseudowire mixed configuration. You can configure a pseudowire with an interface-based drop and a VLAN-based drop. The difference between the two is in the types of parameters that are required to characterize the two types of CTPs. For more information, see 4.5.4.2 deactivateAndDeleteFDFr and 4.5.4.3 modifyFDFr.

4.7.3 Pseudowire Class and Loopback Provisioning Interface

When you create a pseudowire, a given set of parameters can be either left as default values or specified. If these parameters are specified, you must retrieve and specify the valid range of values at the ME level. The pseudowire class is a set of attributes which associates a given pseudowire with a given TP tunnel during creation. To do this, use the setAdditionalInfo API. For more information about loopback interface, see 4.11.2.1 getManagedElement.

This section describes the following interface:

- 4.7.3.1 setAdditionalInfo

4.7.3.1 setAdditionalInfo

Synopsis

```
public void setAdditionalInfo(NameAndStringValue_T[] objectName,
                              NVSList_THolder additionalInfo)
```

```
throws ProcessingFailureException
```

Description

This interface allows you to create and associate a new pseudowire class to the ME and make it available while creating the pseudowire. You can set additional information associated to an ME by providing the name as the input parameter.

Name	Type	Valid Values	Description
pwClassTable.Count	Integer	—	Number of rows in the pseudowire class table.
pwClassTable.OperType	String	• add • rem • mod	To manage add, remove, and modify operations on huge tables, the pwClass table has this new column which specifies the operation to be performed on the single row.
pwClassTable.Name	String	Up to 31 characters in length	Name of the pseudowire class.
pwClassTable.Encaps	String	MPLS	Encapsulation type. The default is MPLS.
pwClassTable.Protocol	String	• NONE • LDP	Protocol type.
pwClassTable.Interwork	String	• VLAN • ETHERNET • NONE	Interworking type.
pwClassTable.CtrlWord	String	• True • False	Control word.
pwClassTable.MasterRed	String	• True • False	Master redundancy.
pwClassTable.PrefPathEnabled	String	• True • False	Preferred path enabled.
pwClassTable.TunnType	String	• TP • TE	Tunnel type. Considered only if PrefPathEnabled is true.
pwClassTable.TunnNum	Integer	—	Tunnel number. Considered only if PrefPathEnabled is true.
pwClassTable.DisFallback	String	• True • False	Disable fallback. Considered only if PrefPathEnabled is

Name	Type	Valid Values	Description
			true.
pwClassTable.SeqEnabled	String	• True • False	Sequencing enabled.
pwClassTable.SeqOper		• Transmit • Receive • Both	Sequencing operation. Considered only if SeqEnabled is true.
pwClassTable.ResyncTimer	Integer	—	Resync Timer. Considered only if SeqEnabled is true.
pwClassTable.StaticOAMEnabled	String	• True • False	Static OAM enabled.
pwClassTable. StaticOamClass	String	Up to 31 characters in length	Static OAM Class. Considered only if StatOAMEnabled is true.
pwClassTable.BFDoVCCVEnabled	String	• True • False	BFD over CCV (BFDoCCV) enabled.
pwClassTable.BfdTemplate	String	Up to 31 characters in length	BFD template.
pwClassTable. AcStat	String	• True • False	AC status signaling. Considered only if BFDoVCCVEnabled is true.
qoSPolicies	String	Comma-separated policy names	Applies to the TCProfile names; for example: • class-map xxxx • policy-map yyy • action zzz • table-map kkk
routerIdName	String	Up to 200 characters in length	MPLS-TP router ID name; for example, loopback0.
routerIdIpAddr	String	IP Address (Ipv4)	Loopback IP address.
routerIdMask	String	IP Address Mask (Ipv4)	Loopback net mask.

The pwClassTable attributes are all required if at least one of the pwClassTable attribute is specified. The attributes values are not passed in the creation request sent to the node if:

- pwClassTable.PrefPathEnabled is false, the pwClassTable.TunnType, pwClassTable.TunnNum, pwClassTable.DisFallback attributes are required, but the content is not passed to the node in the creation command.
- pwClassTable.SeqEnabled is true and pwClassTable.Protocol attribute is NONE, the pwClassTable.ResyncTimer attribute is required, but the content is not passed to the node in the creation command.

- pwClassTable.SeqEnabled is false, the pwClassTable.SeqOper and pwClassTable.ResyncTimer attributes are required, but the content is not passed to the node in the creation command.
- pwClassTable.StaticOAMenabled is false, the pwClassTable.StaticOamClass attribute is required, but the content is not passed to the node in the creation command.
- pwClassTable.BFDoVCCVEnabled is false, the pwClassTable.BfdTemplate and pwClassTable.AcStat attributes are required, but the content is not passed to the node in the creation command.

If pwClass creation or modification fails, an exception is returned if:

- The attribute values do not match the attribute description listed in the table.
- pwClassTable.Name is empty.
- pwClassTable.Interwork is not NONE and pwClassTable.SeqEnabled is true.
- pwClassTable.StaticOAMEnabled is true and pwClassTable.StaticOamClass is an empty string ("").
- pwClassTable.BFDoVCCVEnabled is true and pwClassTable.BfdTemplate is an empty string ("").
- pwClassTable.BFDoVCCVEnabled is true and pwClassTable.CtrlWord is false.
- pwClassTable.Protocol is NONE; and pwClassTable.BFDoVCCVEnabled, pwClassTable.StaticOAMEnabled, and pwClassTable.AcStat are true.
- pwClassTable.Protocol is LDP and pwClassTable.StaticOAMEnabled is true.
- pwClassTable.Protocol is LDP and pwClassTable.StaticOAMEnabled is false and pwClassTable.AcStat is true.
- pwClassTable.PrefPathEnabled is true, pwClassTable.TunnType is TP, and pwClassTable.Protocol is LDP.

If pwClass deletion fails, an exception is returned if:

- The attribute values do not match the attribute description listed in the table.
- pwClassTable.Name is empty.

The following table shows an example of adding a pseudowire class.

PW Class	**Operation Type**	**Encaps**	**Protocol**	**Interwork**	**Ctrl Word**	**Preferred Path**	**Tunnel Type**	**Tunnel Number**	**Enable Sequencing**	**Sequencing Mode**
Pw-300	add	MPLS	NONE	ETHERNET	true	true	TP	300	true	TRANSMIT

The following table shows an example of deleting a pseudowire class.

PW Class	**Operation Type**	**Encaps**	**Protocol**	**Interwork**	**Ctrl Word**	**Preferred Path**	**Tunnel Type**	**Tunnel Number**	**Enable Sequencing**	**Sequencing Mode**
Pw-301	rem	MPLS	LDP	ETHERNET	false	false	TP	301	true	TRANSMIT

The following table shows an example of modifying a pseudowire class.

PW Class	Operation Type	Encaps	Protocol	Interwork	Ctrl Word	Preferred Path	Tunnel Type	Tunnel Number	Enable Sequencing	Sequencing Mode
Pw-302	mod	MPLS	LDP	VLAN	true	true	TP	302	true	TRANSMIT

GateWay/CORBA NBI Result

```
pwClassTable.Count =          1
pwClassTable.OperType =   "add"
pwClassTable.Name=            "Pw-300"
pwClassTable.Encaps=          "MPLS"
pwClassTable.Protocol=        "NONE"
pwClassTable.Interwork=       "ETHERNET"
pwClassTable.CtrlWord=        "true"
pwClassTable.PrefPath=      "true"
pwClassTable.TunnType=    "TP"
pwClassTable.TunnNum=     "300"
pwClassTable.EnableSeq=    "true"
pwClassTable.SeqMode=       "Transmit"
...........
```

You can add, modify, or delete only one row in the table.

4.7.4 Pseudowire QoS Provisioning Interface

When you create a pseudowire, specify the QoS policies to be associated to the pseudowire drops. If you specify policy maps, they must be available in the given ME the Ethernet drop belongs to.

Use Case Description

- Create the TC profiles for policy maps, class maps, actions, and table maps using the createTCProfile interface. The new policies are stored in the EMS database, not in the ME.
- Modify and delete QoS policies through the modifyTCProfile and deleteTCProfile interfaces.
- Associate the given policies that are already present in the EMS database to an ME using the setAdditionalInfo interface, which downloads the given policies to the NE.
- Use the getManagedElement interface to get the QoS policies present in an ME. The QoS policy names are returned with the additional parameters.
- When you create a pseudowire, specify the desired QoS policy map names (at ingress and egress) as additional parameters at pseudowire endpoints within the createAndActivateFDFr interface.

This section describes the following interface:

- 4.7.4.1 TCProfileMgr::setAdditionalInfo

4.7.4.1 TCProfileMgr::setAdditionalInfo

Synopsis

```
public void setAdditionalInfo(NameAndStringValue_T[] objectName,
                              NVSList_THolder additionalInfo)
      throws ProcessingFailureException
```

Description

This interface allows you to associate a list of QoS policies to an ME. The NMS provides the ME name in the objectName parameter and the list of QoS policies in the additionalInfo parameter. The EMS returns the list of policies whose provisioning has succeeded on the given ME in the additionalInfo parameter. If provisioning failed for all the given QoS policies, the additionalInfo output contains an empty array.

The following table describes the parameters for additionalInfo in TCProfileMgr::setAdditionalInfo.

Name	Type	Description
class-map	String	Optional. The user label of the class map. The maximum length is 39 characters.
policy-map	String	Optional. The user label of the policy map. The maximum length is 39 characters.
table-map	String	Optional. The user label of the table map. The maximum length is 39 characters.

Each tuple is optional, but can be repeated as many times as needed to direct the EMS to provision multiple class maps, policy maps, and table maps on an ME. The order is irrelevant.

✐ You cannot modify or delete TC profiles that are provisioned on the node with the current API implementation.

4.8 Supported Inventory Interfaces

The following table lists the inventory interfaces supported in Prime Optical 10.7.

Entity	Interface
EVC	
4.9.1 CPTP Inventory Interfaces	4.9.1.1 getAllCPTPs
	4.9.1.2 getAllAssignedCPTPs
	4.9.1.3 getAllAssignableCPTPs
	4.9.1.4 getAllPTPs
	4.9.1.5 getTP
5.9.2 MFD Inventory Interfaces	4.9.2.1 getAllAssociatedMFDs
	4.9.2.2 getAllSupportedMFDs
	4.9.2.3 getMFD
	4.9.2.4 getAssigningMFD
4.9.3 Flow Domain Inventory Interfaces	4.9.3.1 getAllFlowDomains
	4.9.3.2 getFlowDomainsByUserLabel
	4.9.3.3 getFlowDomain
	4.9.3.4 getAllTopologicalLinksOfFD

Entity	Interface
	4.9.3.5 getAssociatingFD
4.9.4 EVC FDFr Inventory Interfaces	4.9.4.1 getFDFr
	4.9.4.2 getAllFDFrs
	4.9.4.3 getFDFrsWithTP
	4.9.4.4 getFDFrsByUserLabel
	4.9.4.5 getAllFDFrNames
	4.9.4.6 getFDFrNamesWithTP
	4.9.4.7 getFDFrRoute
4.9.5 Link Aggregation Inventory Interfaces	4.9.5.1 getAllFTPs
	4.9.5.2 getAllFTPNames
	4.9.5.3 getTP
4.9.6 Common Inventory Interface	4.9.6.1 getTransmissionParams
4.9.7 Traffic Conditioning Profile Inventory Interfaces	4.9.7.1 getAllTCProfiles
	4.9.7.2 getTCProfile
MPLS-TP	
4.10.1 MPLS-TP Tunnel Inventory Interfaces	4.10.1.1 getSNC
	4.10.1.2 getAllSubnetworkConnectionsWithTP
	4.10.1.3 getAllSubnetworkConnectionsNamesWithTP
	4.10.1.4 getAllSubnetworkConnections
	4.10.1.5 getAllSubnetworkConnectionNames
	4.10.1.6 getSNCsByUserLabel
4.10.2 MPLS-TP LSP Inventory Interfaces	4.10.2.1 getSNC
	4.10.2.2 getAllSubnetworkConnectionsWithTP
	4.10.2.3 getAllSubnetworkConnectionsNamesWithTP
4.10.3 MPLS-TP Tunnel Link Inventory Interfaces	4.10.3.1 getTopologicalLink
	4.10.3.2 getAllTopologicalLinks
	4.10.3.3 getAllTopologicalLinkNames

Entity	Interface
	4.10.3.4 getAllTopLevelTopologicalLinks
	4.10.3.5 getAllTopLevelTopologicalLinkNames
	4.10.3.6 getTopLevelTopologicalLink
4.10.4 MPLS-TP LSP Route Inventory Interfaces	4.10.4.1 getRoute
	4.10.4.2 getRouteAndTopologicalLinks
	4.10.4.3 getAllCrossConnections
4.10.5 MPLS-TP Protection (Inventory)	—
4.10.6 BFD Templates, Node ID, Label Range, and ME OperationMode Inventory Interface	4.10.6.1 getManagedElement
Pseudowire	
4.11.1 Pseudowire FDFr Inventory Interfaces	4.11.1.1 getFDFr
	4.11.1.2 getAllFDFrs
	4.11.1.3 getFDFrsWithTP
	4.11.1.4 getAllFDFrNames
	4.11.1.5 getFDFrNamesWithTP
	4.11.1.6 getFDFrsByUserLabel
	4.11.1.7 getFDFrRoute
4.11.2 Pseudowire Class, Loopback, and QoS Policy Inventory Interface	4.11.2.1 getManagedElement

4.9 EVC Inventory Interfaces

This section describes the following EVC inventory interfaces:

- 4.9.1 CPTP Inventory Interfaces
- 4.9.2 MFD Inventory Interfaces
- 4.9.3 Flow Domain Inventory Interfaces
- 4.9.4 EVC FDFr Inventory Interfaces
- 4.9.5 Link Aggregation Inventory Interfaces
- 4.9.6 Common Inventory Interface
- 4.9.7 Traffic Conditioning Profile Inventory Interfaces

4.9.1 CPTP Inventory Interfaces

This section describes the following interfaces:

- 4.9.1.1 getAllCPTPs
- 4.9.1.2 getAllAssignedCPTPs
- 4.9.1.3 getAllAssignableCPTPs
- 4.9.1.4 getAllPTPs

- 4.9.1.5 getTP

4.9.1.1 getAllCPTPs

Synopsis

```
public void getAllCPTPs(
  NameAndStringValue_T[]            fdName,
                          CPTP_Role_T                 cptpRole,
                          int                               how_many,
                          TerminationPointList_THolder      cptpList,
                          TerminationPointIterator_IHolder  cptpIt)
      throws ProcessingFailureException
```

Description

This interface returns the structure of all CPTPs, fdEdge CPTPs, or fdInternal CPTPs associated with an FD.

Parameters

- NamingAttributes_T fdName—FD name. The associated CPTPs of this FD are returned.
- CPTP_Role_T cptpRole—CPTPs to be returned (fdEdge CPTP, fdInternal CPTP, or all CPTPs).
- unsigned long how_many—Maximum number of CPTPs to be returned in the first batch.
- TerminationPointList_T cptpList—First batch of CPTPs.
- TerminationPointIterator_I cptpIt—Iterator to retrieve the remaining CPTPs.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when fdName does not reference an FD object.

EXCPT_ENTITY_NOT_FOUND - Raised when fdName references an object that does not
exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that
the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

4.9.1.2 getAllAssignedCPTPs

Synopsis

```
void getAllAssignedCPTPs(
NameAndStringValue_T[]        mfdName,
                                    int                           how_many,
                                    TerminationPointList_THolder  cptpList,
                                    TerminationPointIterator_IHolder
      cptpIt)
      throws ProcessingFailureException
```

Description

This interface returns the structure of all CPTPs assigned to an MFD.

Parameters

- NamingAttributes_T mfdName—MFD name. The assigned CPTPs of this MFD are returned.

- unsigned long how_many—Maximum number of CPTPs to be returned in the first batch.
- TerminationPointList_T cptpList—First batch of CPTPs.
- TerminationPointIterator_I cptpIt—Iterator to retrieve the remaining CPTPs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mfdName does not reference an MFD object.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.1.3 getAllAssignableCPTPs

Synopsis

```
public void getAllAssignableCPTPs(
NameAndStringValue_T[]        mfdName,
                                    int                            how_many,
                                    TerminationPointList_THolder   cptpList,
                                    TerminationPointIterator_IHolder
      cptpIt)
      throws ProcessingFailureException
```

Description

This interface returns the structure of all CPTPs that can be assigned to an MFD. These CPTPs must be on the same equipment or same rack with backplane connectivity.

Parameters

- NamingAttributes_T mfdName—MFD name. The potential CPTPs of this MFD are returned.
- unsigned long how_many—Maximum number of CPTPs to be returned in the first batch.
- TerminationPointList_T cptpList—First batch of CPTPs.
- TerminationPointIterator_I cptpIt—Iterator to retrieve the remaining CPTPs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mfdName does not reference an MFD object.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.1.4 getAllPTPs

Synopsis

```
public void getAllPTPs(
NameAndStringValue_T[]  managedElementName,
```

```
                        short[]                 tpLayerRateList,
                        short[]                 connectionLayerRateList,
                        int                     how_many,
                        TerminationPointList_THolder tpList,
                        TerminationPointIterator_IHolder tpIt)
      throws ProcessingFailureException
```

Description
For CPTPs, use this interface to determine the parameters that characterize a CPTP.
Example
ConnectionlessPort=true and PortTpRoleState reporting to the current role state can be unassigned, assigned, fdInternal, or fdEdge.

4.9.1.5 getTP

Synopsis

```
public void getTP(
NameAndStringValue_T[] tpName,
                        TerminationPoint_THolder tp)
      throws ProcessingFailureException
```

Description
This interface returns the TP structure for the given TP name (CTP, FTP, or PTP). The TP structure consists of transmission parameters. The transmission parameters returned are the parameters in place on the actual TP on the NE. If there are no transmission parameters or the TP does not actually exist on the NE, transmissionParams will be empty.
Parameters
- NamingAttributes tpName—TP name. The structure of this TP is retrieved.
- TerminationPoint_T tp—The retrieved TP.

Throws
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a TP object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a TP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

For CPTPs, use this interface to determine the parameters that characterize a CPTP.
Example
ConnectionlessPort=true and PortTpRoleState reporting to the current role state can be unassigned, assigned, fdInternal, or fdEdge.

The following table describes the ConnectionlessPort and PortTPRoleState parameters.

TP Parameter Name	TP Type	Layers	Valid Values	AVC	Description
ConnectionlessPort	• PTP • FTP	All layers	• true • false	Yes	See 4.5.1.3 setAdditionalInfo.
PortTPRoleState	• PTP • FTP	All connectionless	• unassigned • assigned	Yes	See 4.5.1.3 setAdditionalInfo. The mismatch operational state

TP Parameter Name	TP Type	Layers	Valid Values	AVC	Description
		layers	• fdInternal • fdEdge • mismatch		exists for the CPTP if there is a mismatch in the port type. This is applicable only on pluggable PPM ports when you change a previously defined CPTP with a port that no longer supports Ethernet.

4.9.2 MFD Inventory Interfaces

This section describes the following interfaces:

- 4.9.2.1 getAllAssociatedMFDs
- 4.9.2.2 getAllSupportedMFDs
- 4.9.2.3 getMFD
- 4.9.2.4 getAssigningMFD

4.9.2.1 getAllAssociatedMFDs

Synopsis

```
public void getAllAssociatedMFDs(NameAndStringValue_T[] tmdOrFdName,
                                 int how_many,
                                 MFDList_THolder mfds,
                                 MFDIterator_IHolder mfdIt)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request a list of MFDs that are associated with a specified transmission descriptor (TMD) or FD. This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- NamingAttributes_T tmdOrFdName—Name of the TMD or FD.
- unsigned long how_many—Maximum number of MFDs to be returned in the first batch.
- MFDList_T mfds—First batch of MFDs.
- MFDIterator_I mfdIt—Iterator to retrieve the remaining MFDs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tmdOrFdName does not reference a TMD or FD object.

EXCPT_ENTITY_NOT_FOUND - Raised when tmdOrFdName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators that the EMS can support has been reached.

Limitations

In Prime Optical 10.7, this interface is supported only if the tmdOrFdName parameter contains an FD name.

4.9.2.2 getAllSupportedMFDs

Synopsis

```
public void getAllSupportedMFDs(NameAndStringValue_T[] holderName,
                                int how_many,
                                MFDList_THolder mfds,
                                MFDIterator_IHolder mfdIt)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request a list of MFDs that are contained by a specified ME or supported with specified equipment (holderName). This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- NamingAttributes_T holderName—Name of the ME or equipment.
- unsigned long how_many—Maximum number of MFDs to be returned in the first batch.
- MFDList_T mfds—First batch of MFDs.
- MFDIterator_I mfdIt—Iterator to retrieve the remaining MFDs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when holderName does not reference an ME or equipment object.

EXCPT_ENTITY_NOT_FOUND - Raised when holderName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when the maximum number of iterators that the EMS can support has been reached.

Limitations

This interface supports the request only for MEs, slots, and TPs.

4.9.2.3 getMFD

Synopsis

```
public void getMFD(NameAndStringValue_T[] mfdName,
                   MatrixFlowDomain_THolder mfd)
      throws ProcessingFailureException
```

Description

This interface returns the structure of an MFD.

Parameters

- globaldefs::NamingAttributes_T mfdName—MFD name that must be retrieved. The name of each MFD must be in the following format:
  name="EMS";value="CompanyName/EMSname"
  name="ManagedElement";value="ManagedElementName"
  name="MatrixFlowDomain"; value= "MatrixFlowDomainName"
- MatrixFlowDomain_T mfd—MFD structure that is returned.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mfdName does not reference an MFD object.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName references an MFD object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.2.4 getAssigningMFD

Synopsis

```
public void getAssigningMFD(NameAndStringValue_T[] cptpName,
                            MatrixFlowDomain_THolder mfd)
      throws ProcessingFailureException
```

Description

This interface returns the MFD to which a CPTP is assigned.

Parameters

- NamingAttributes_T cptpName—Name of the CPTP to which the MFD is assigned.
- MatrixFlowDomain_T mfd—The MFD structure that is returned.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when cptpName does not reference a CPTP TP.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName references a TP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.3 Flow Domain Inventory Interfaces

This section describes the following interface:

- 4.9.3.1 getAllFlowDomains

4.9.3.1 getAllFlowDomains

Synopsis

```
public void getAllFlowDomains(int how_many,
                              FDList_THolder flowDomains,
                              FDIterator_IHolder fdIt)
      throws ProcessingFailureException
```

Description

This interface enables the NMS to request a list of FDs that exist in the EMS. This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- unsigned long how_many—Maximum number of FDs to be returned in the first batch.
- FDList_T flowDomains—First batch of FDs.
- FDIterator_I fdIt—Iterator to retrieve the remaining FDs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.3.2 getFlowDomainsByUserLabel

Synopsis

```
public void getFlowDomainsByUserLabel(String userLabel,
                                      FDList_THolder flowDomains)
      throws ProcessingFailureException
```

Description

This interface returns the FD structure for the FDs. You must use userLabel as the parameter. The operation does not use an iterator because the number of FDs returned is 1.

Parameters

- string userLabel—The user label of the FD that must be retrieved.
- FDList_T flowDomains—The list of identified FDs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.3.3 getFlowDomain

Synopsis

```
public void getFlowDomain(NameAndStringValue_T[] fdName,
                          FlowDomain_THolder flowDomain)
      throws ProcessingFailureException
```

Description

This interface returns the FD structure.

Parameters

- NamingAttributes_T fdName—Name of the FD that must be retrieved. The name of each FD must be in the following format:
  name="EMS";value="CompanyName/EMSname"
  name="FlowDomain";value=" FlowDomainName"
- FlowDomain_T flowDomain—The FD structure that is returned.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when fdName does not reference an FD object.

EXCPT_ENTITY_NOT_FOUND - Raised when fdName references an FD object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.3.4 getAllTopologicalLinksOfFD

Synopsis

```
public void getAllTopologicalLinksOfFD(NameAndStringValue_T[] flowDomainName,
```

```
                                                int how_many,
                                                TopologicalLinkList_THolder topoList,
                                                TopologicalLinkIterator_IHolder
topoIt)
      throws ProcessingFailureException
```

Description

This interface returns a list of topological links at the encapsulation layer. These links are terminated at the FD. The topological link name must be passed as a parameter. This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- NamingAttributes_T flowDomainName—FD name.
- unsigned long how_many—Maximum number of topological links to be returned in the first batch.
- TopologicalLinkList_T topoList—First batch of topological links.
- TopologicalLinkIterator_I topoIt—Iterator to retrieve the remaining topological links.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when flowDomainName does not reference an FD object.

EXCPT_ENTITY_NOT_FOUND - Raised when flowDomainName references an object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

4.9.3.5 getAssociatingFD

Synopsis

```
public void getAssociatingFD(NameAndStringValue_T[] mfdName,
                                 FlowDomain_THolder flowDomain)
      throws ProcessingFailureException
```

Description

This interface returns the FD structure that the MFD is associated to.

Parameters

- NamingAttributes_T mfdName—The name of MFD.
- FlowDomain_T flowDomain—The FD that the MFD is associated to.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when mfdName does not reference an MFD object.

EXCPT_ENTITY_NOT_FOUND - Raised when mfdName references an MFD object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.9.4 EVC FDFr Inventory Interfaces

This section describes the following interfaces:

4.9.4.1 getFDFr

Synopsis

```
public void getFDFr(NameAndStringValue_T[] fdfrName,
                    FlowDomainFragment_THolder fdfrHolder)
      throws ProcessingFailureException
```

Description

This interface returns the FDFr representing the EVC. This EVC name must be provided as a parameter.

Parameters

- NamingAttributes_T fdfrName—The EVC name to be retrieved. The EVC name is provided as:
  <EVC native name>:SID=nn:LR=EVC.
- FlowDomainFragment_T fdfr—The FDFr structure that is retrieved.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when fdfrName does not reference an FDFr object.

EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an FDFr object that
does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

Relevant Data Structures

```
FlowDomainFragment_T

      NameAndStringValue_T[]  name

      String                        userLabel

      String                        nativeEMSName

      String                        owner

      ConnectionDirection_T   direction

      LayeredParameters_T     transmissionParams

      NameAndStringValue_T[][]      aEnd

      NameAndStringValue_T[][]      zEnd

      TPData_T[]              aEnd

In EVC Line, only one EFP exists. In EVC LAN, there is no distinction between
source and destination drops. All the EVC EFPs is contained here. Endpoints
are either Ethernet ports or channel groups (PTPs or FTPs).
```

```
      TPData_T[]              zEnd
In EVC Line, only one EFP exists. In EVC LAN, the list is empty because all
the EVC EFPs is contained in the aEnd list. Endpoints are either Ethernet
ports or channel groups (PTPs or FTPs).
      String                        networkAccessDomain
      boolean                 flexible
      AdministrativeState_T   administrativeState
In Prime Optical 10.7, the administrative state associated to the service is
not supported.
      SNCState_T              fdfrState
      String                        fdfrType
      NameAndStringValue_T[]  additionalInfo
```

The following table describes the additionalInfo parameter.

Name	Type	Description
evcType	String	Valid values are: • Ethernet Private Line • Ethernet Virtual Private Line • Ethernet Private LAN • Ethernet Virtual Private LAN
bw	Long	EVC allocated bandwidth.
bwUnits	String	Valid values are: • Kbps • Mbps • Gbps

The EFP configuration parameters that characterize the EVC drops are similar to pseudowire drops. For more information, see 4.11.1.1 getFDFr.

4.9.4.2 getAllFDFrs

Synopsis

```
public void getAllFDFrs(NameAndStringValue_T[] fdName,
                        int how_many,
                        short[] connectivityRateList,
                        FDFrList_THolder fdfrList,
                        FDFrIterator_IHolder fdfrIt)
      throws ProcessingFailureException
```

Description

This interface requests a list of FDFrs for the specified FD at the defined connectivity rates. This operation uses an iterator to allow the NMS to deal with a large number of objects. For more information, see 4.11.1.1 getFDFr.

Limitations

For EVC FDFRs, all the get APIs reflect as a sequence of requests issued directly to the nodes in the network. Database data cannot be used for EVCs.

4.9.4.3 getFDFrsWithTP

Synopsis

```
public void getFDFrsWithTP(NameAndStringValue_T[] cptpName,
                           int how_many,
                           FDFrList_THolder fdfrList,
                           FDFrIterator_IHolder fdfrIt)
      throws ProcessingFailureException
```

Description
This interface enables the NMS to request the EMS for the FDFrs that are connected to a specified FP or CPTP. If you use an FP as an input parameter, not more than one FDFr is returned. For more information, see 4.11.1.3 getFDFrsWithTP.

4.9.4.4 getFDFrsByUserLabel

Synopsis

```
public void getFDFrsByUserLabel(
String                  userLabel,
                                FDFrList_THolder  fdfrs)
      throws ProcessingFailureException
```

Description
This interface returns the FDFr by userLabel. Provide the userLabel as the parameter. The operation does not use an iterator because the number of FDFrs returned is expected to be 1. For more information, see 4.11.1.6 getFDFrsByUserLabel.

4.9.4.5 getAllFDFrNames

Synopsis

```
void getAllFDFrNames(
NameAndStringValue_T[]        fdName,
                        int                           how_many,
            short[]                           connectivityRateList,
                  NamingAttributesList_THolder  nameList,
                        NamingAttributesIterator_IHolder    nameIt)
      throws ProcessingFailureException
```

Description
This interface returns the list of all the pseudowire names configured in an FD. For more information, see 4.11.1.4 getAllFDFrNames.

4.9.4.6 getFDFrNamesWithTP

Synopsis

```
void getFDFrNamesWithTP(
NameAndStringValue_T[]        cptpName,
                        int                           how_many,
                        NamingAttributesList_THolder  nameList,
                        NamingAttributesIterator_IHolder    nameIt)
      throws ProcessingFailureException
```

Description
This interface returns the list of all the pseudowire names; the list has at least one EFP in the CPTP provided as the input parameter. For more information, see 4.11.1.3 getFDFrsWithTP.

Limitations
For EVC FDFRs, all the get APIs reflect as a sequence of requests issued directly to the nodes in the network. Database data cannot be used for EVCs.

4.9.4.7 getFDFrRoute

Synopsis

```
public void getFDFrRoute(
NameAndStringValue_T[]  fdfrName,
                                FDFrRoute_THolder              route)
      throws ProcessingFailureException
```

Description
This interface returns the FDFr route representing the EVC. The EVC name is specified as a parameter.

Parameters

- NamingAttributes_T fdfrName—The EVC name to be retrieved. The EVC name is provided as:
  <EVC native name>:SID=nn:LR=EVC
- LR=EVC represents Layer Rate = EVC. The API does not expect an LR to be specified.
- FDFrRoute_T route—The route of the FDFr.

Throws

```
Raises ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when fdfrName does not reference an FDFr object.
EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an FDFr object that
does not exist.
EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

Relevant Data Structures

```
FDFrRoute_THolder
            MatrixFlowDomainFragment_T[]  value


An MFDFr object represents the portion of an FDFr within an MFD inside an
NE.An MFDFr is primarily used in the specification of an FDFr route where the
route must be specified by the NMS.


MatrixFlowDomainFragment_T
      ConnectionDirection_T         direction
                                    (Bidirectional for Ethernet)


      LayeredParameters_T           transmissionParams
                                    (LR_EVC and the related connectionless
parameters)


      NameAndStringValue_T[][]      aEnd
```

EVC Edge EFPs. EVC Ethernet Virtual Private Line is always one. EVC Ethernet Virtual Private LAN can contain more EFPs (all the Edge EFPs on that node). The EFP/CTP native name reports information about the Ethernet VLAN tagging.

NameAndStringValue_T[][] zEnd

EVC internal EFPs. These are EFPs with their own EFP configuration, which are selected by the network in order to provide the routing path to the EVC service.

boolean flexible

FDFr is fixed or flexible.

boolean active

MFDFr has been successfully activated on the ME.

String mfdfrType

Can be one of the following:

VIRTUAL PRIVATE LAN (MULTIPOINT)

VIRTUAL PRIVATE LINE (POINT-TO-POINT)

PRIVATE LAN (MULTIPOINT)

PRIVATE LINE (POINT-TO-POINT)

NameAndStringValue_T[] additionalInfo

4.9.5 Link Aggregation Inventory Interfaces

This section describes the following interfaces:

- 4.9.5.1 getAllFTPs
- 4.9.5.2 getAllFTPNames
- 4.9.5.3 getTP

4.9.5.1 getAllFTPs

Synopsis

```
void getAllFTPs(
    NameAndStringValue_T[]   managedElementName,
                             short[]                        tpLayerRateList,
                             short[]
    connectionLayerRateList,
                             int                            how_many,
                             NamingAttributesList_THolder   nameList,
                             NamingAttributesIterator_IHolder nameIt)
      raises(globaldefs::ProcessingFailureException)
```

Description

This interface returns the FTPs.

Parameters

- NamingAttributes_T managedElementName—ME name.
- LayerRateList_T tpLayerRateList—List of FTP layer rates for which the FTPs are to be fetched. An FTP must contain at least one of the layer rates specified. If the list is empty, FTPs of all rates are returned.
- LayerRateList_T connectionLayerRateList—List of connection layer rates for which the FTPs are to be fetched. An FTP must support connections for at least one of the layer rates specified. If the list is empty, FTPs for all connection rates are returned.
- unsigned long how_many—Maximum number of FTPs to be returned in the first batch.
- TerminationPointList_T tpList—First batch of FTPs.
- TerminationPointIterator_I tpIt—Iterator to retrieve remaining FTPs.

Throws

Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the ME name does not reference an ME object, or tpLayerRateList or connectionLayerRateList contain undefined rates.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME name references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communications to the ME is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

4.9.5.2 getAllFTPNames

Synopsis

```
void getAllFTPNames(
     NameAndStringValue_T[]   managedElementName,
                              short[]                          tpLayerRateList,
                              short[]
      connectionLayerRateList,
                              int                              how_many,
                              NamingAttributesList_THolder     nameList,
                              NamingAttributesIterator_IHolder nameIt)
      throws ProcessingFailureException
```

Description

This interface returns the FTP names.

Parameters

- NamingAttributes_T managedElementName—ME name.
- LayerRateList_T tpLayerRateList—List of FTP layer rates for which the FTPs are to be fetched. An FTP must contain at least one of the layer rates specified. If the list is empty, then FTPs of all rates are returned.
- LayerRateList_T connectionLayerRateList—List of connection layer rates for which the FTPs are to be fetched. An FTP must support connections for at least one of the layer rates specified. If the list is empty, FTPs for all connection rates are returned.
- unsigned long how_many—Maximum number of FTPs to be returned in the first batch.
- NamingAttributesList_THolder nameList—First batch of FTP names.

- NamingAttributesIterator_IHolder nameIt—Iterator to retrieve remaining FTP names.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when the ME name does not reference an ME object, or tpLayerRateList or connectionLayerRateList contain undefined rates.

EXCPT_ENTITY_NOT_FOUND - Raised when the ME name references an ME object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

4.9.5.3 getTP

Synopsis

```
public void getTP(
NameAndStringValue_T[]  tpName,
                        TerminationPoint_THolder      tp)
      throws ProcessingFailureException
```

Description

This interface returns the TP structure for the given TP names (CTP, FTP, or PTP). The TP name must be explicit. The transmission parameters returned are the parameters in place on the actual TP on the NE. If there are no transmission parameters or the TP does not actually exist on the NE, the transmissionParams are empty.

Parameters

- NamingAttributes tpName—Name of the FTP to be retrieved. Because an Ethernet LAG port can aggregate client Ethernet ports of different ME cards, the corresponding FTP is logically positioned on the unique shelf of the ME itself. The TP name must be in the following format:
  name="EMS";value="CompanyName/EMSname"
  name="ManagedElement";value="ManagedElementName"
  name="FTP";value="/rack=1/shelf=1/port=nn"
- TerminationPoint_T tp—The retrieved TP.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference a TP object.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references a TP object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.

Relevant Data Structures

TerminationPoint_THolder

TerminationPoint_T value

TerminationPoint_T

```
      NameAndStringValue_T[]  name;
```

```
name="EMS";value="CompanyName/EMSname"
name="ManagedElement";value="ManagedElementName"
name="FTP";value="/rack=1/shelf=1/port=nn"

      String                        userLabel
      String                        nativeEMSName
                              "/rack=1/shelf=1/port=nn"
      String                        owner
      NameAndStringValue_T[]  ingressTrafficDescriptorName;
      NameAndStringValue_T[]  egressTrafficDescriptorName;
      TPType_T                type;
      TPConnectionState_T     connectionState;
      TerminationMode_T             tpMappingMode;
      Directionality_T        direction;
      LayeredParameters_T[]   transmissionParams;
                              Shall be LR_LAG_Fragment.
      TPProtectionAssociation_T     tpProtectionAssociation;
      boolean                 edgePoint;
      NameAndStringValue_T[]  additionalInfo;

LayeredParameters_T
      short                         layer
      NameAndStringValue_T[]  transmissionParams
```

The additionalInfo parameter specifies all the following attributes at FTP Level.

Name	Value	Description
MTU	Integer	MTU number. Valid values are from 64 to 9600.
FastSwitchOver	Boolean	To enable LACP 1:1, set it to true.
L2ActionCDP	String	CDP is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
L2ActionDOT1X	String	DOT1X is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
L2ActionDTP	String	DTP is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
L2ActionLACP	String	LACP is a Layer 2 protocol. Valid action names are:

Name	Value	Description
		• Peer • Forward • Drop
L2ActionPAGP	String	PAGP is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
L2ActionSTP	String	STP is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
L2ActionVTP	String	VTP is a Layer 2 protocol. Valid action names are: • Peer • Forward • Drop
IngressPolicyMap	String	Ingress policy map name. Valid value is an empty string ("").
IngressTableMap	String	Ingress table map name. Valid value is an empty string ("").
IngressTableMapConfig	String	Ingress table map config name. Valid value is an empty string ("").
EgressPolicyMap	String	Egress policy map name. Valid value is an empty string ("").
enableLACP	Boolean	To enable LACP, set it to true.
MinBundle	Integer	Minimum bundle number. If portLACP is false, this attribute must be set to 1. Valid values are from 1 to 8.
MaxBundle	Integer	Maximum bundle number. If portLACP is false, this attribute must be set to 8. Valid values are from 1 to 8.
nativeNameAggrPorts	String	Comma-separated concatenation of the native names of the Ethernet PTPs currently aggregated to this FTP LAG. Example "/rack=1/shelf=1/slot=3/ppm_holder=2/port=1, /rack=1/shelf=1/slot=4/ppm_holder=1/port=1, /rack=1/shelf=1/slot=1/ppm_holder=3/port=1"

4.9.6 Common Inventory Interface

This section describes the following interface:

- 4.9.6.1 getTransmissionParams

4.9.6.1 getTransmissionParams

Synopsis

```
public void getTransmissionParams(NameAndStringValue_T[] name,
                                  String[] filter,
```

```
                                        LayeredParameterList_THolder
transmissionParams)
      throws ProcessingFailureException
```

Description
This interface returns the transmission parameters of an FD and MFD. Specify the set of transmission parameters to be returned.
Limitations
Filter is not supported.

4.9.7 Traffic Conditioning Profile Inventory Interfaces

This section describes the following interfaces:

- 4.9.7.1 getAllTCProfiles
- 4.9.7.2 getTCProfile

4.9.7.1 getAllTCProfiles

Synopsis

```
public void getAllTCProfiles(int how_many,
                              TCProfileList_THolder tcProfileList,
                              TCProfileIterator_IHolder tcProfileIt)
      throws ProcessingFailureException
```

Description
This interface allows an NMS to request all of the TC profiles that are under the control of the TCProfileMgr. To allow the NMS to deal with a large number of objects, this operation uses an iterator.

4.9.7.2 getTCProfile

Synopsis

```
getTCProfile(NameAndStringValue_T[] tcProfileName,
                          TCProfile_THolder tcProfile)
      throws ProcessingFailureException
```

Description
This interface returns the TC profile structure for the specified tcProfileName. The TC profile structure contains an NVSList of traffic parameters. If there are no traffic parameters, the NVSList is empty.
Relevant Data Structures

```
TCProfileList_THolder
TCProfile_T[] value;
TCProfile_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
      boolean                 defaultProfile
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  additionalInfo
```

4.10 MPLS-TP Inventory Interfaces

This section describes the following MPLS-TP inventory interfaces:

4.10.1 MPLS-TP Tunnel Inventory Interfaces

This section describes the following interfaces:

4.10.1.1 getSNC

Synopsis

```
void getSNC(
                    in NamingAttributes_T          sncName,
                    out SubnetworkConnection_T     snc)
   raises (ProcessingFailureException);
```

Description

This interface returns the SubnetworkConnection structure for the TP tunnel SNC identified by the sncName. Because TP tunnel names are not guaranteed to be unique in the network, you must provide the TP tunnel name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66.
The BFD template order follows the order of the aEnd and zEnd endpoints of the TP tunnel. Because the order is not stored in the ME, it depends on the order of discovery. The TKEY is composed as follows:
<source Node ID>:<source Tunnel ID>:<dest Node ID>:<dest Tunnel ID>
The TKEY remains invariant with the order of discovery.

Relevant Data Structures

```
SubnetworkConnection_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
      SNCState_T                    sncState
      ConnectionDirection_T         direction
      short                         rate
      StaticProtectionLevel_T       staticProtectionLevel
```

```
SNCType_T                    sncType
TPData_T[]                   aEnd
TPData_T[]                   zEnd
Reroute_T                    rerouteAllowed
NetworkRouted_T              networkRouted
NameAndStringValue_T[]       additionalInfo
```

The following table describes the additionalInfo parameters.

Parameter Name	Valid Values	Description
workLspName	String	Name of the working LSP circuit.
workLspSid	Integer	SID of the working LSP circuit.
workLspUniqueID	Integer	Unique ID of the working LSP circuit.
workLspNum	Integer	Working LSP number.
protLspName	String	Name of the protected LSP circuit.
protLspSid	Integer	SID of the protected LSP circuit.
protLspUniqueID	Integer	Unique ID of the protected LSP circuit.
protkLspNum	Integer	Protected LSP number.
tunnelKey	String	The tunnel key is composed in the following format: sourceNodeId:SourceTunNum:DestNodeId:DestTunNum
ServiceID	Integer	The service ID associated by the network to the service.

aEnd and zEnd CTPs report as Prime Optical native name. The tunnel number must be in the following format: TunnelNum=ddd.
The following are possible values for the SNC state:

- Pending—The SNC has been created by an NMS and has not been activated by any NMS, or the SNC has been successfully deactivated by an NMS. This state has no relationship with the network state of the SNC cross-connections.
- Partial—The SNC is not in pending state. A route has not been assigned to the SNC, or not all of the SNC cross-connections are active in the network. This may or may not include activated SNCs for which there are currently no active cross-connections in the network. A TP tunnel without any LSPs configured is in partial state.
- Active—The SNC is not in pending state. A route has been assigned to the SNC, and all SNC cross-connections are active in the network.

4.10.1.2 getAllSubnetworkConnectionsWithTP

Synopsis

```
public void getAllSubnetworkConnectionsWithTP(
NameAndStringValue_T[]        tpName,
                                                    short[]
      connectionRateList,
```

```
                                                    int
      how_many,

SubnetworkConnectionList_THolder sncList,
                                                    SNCIterator_IHolder
      sncIt)
      throws ProcessingFailureException
```

Description
This interface allows the NMS to request a list of MPLS-TP tunnel SNCs using the specified TP at the specified connection rates. The TP tunnel layer rate is LR_MPLS (165). A TP can be a PTP. Because an MPLS-TP tunnel is terminated in the equipment and not in the PTP, all the MPLS-TP tunnels that end in the device and are suspended by the PTP are returned.

4.10.1.3 getAllSubnetworkConnectionsNamesWithTP

Synopsis

```
void getAllSubnetworkConnectionNamesWithTP(
    NameAndStringValue_T[]          tpName,
                                                        short[]
      connectionRateList,
                                                        int
      how_many,

NamingAttributesList_THolder  nameList,

NamingAttributesIterator_IHolder nameIt)
```

Description
This interface allows the NMS to request a list of MPLS-TP tunnel SNC names using the specified TP at the specified connection rates. The TP tunnel layer rate is LR_MPLS (165). A TP can be a PTP. Because an MPLS-TP tunnel is terminated in the equipment and not in the PTP, all the MPLS-TP tunnels that end in the device and are suspended by the PTP are returned.

4.10.1.4 getAllSubnetworkConnections

Synopsis

```
public void getAllSubnetworkConnections(NameAndStringValue_T[] subnetName,
                                        short[] connectionRateList,
                                        int how_many,
                                        SubnetworkConnectionList_THolder
sncList,
                                        SNCIterator_IHolder sncIt)
      throws ProcessingFailureException
```

Description
This interface allows the NMS to request a list of SNCs for the specified subnetwork at the specified connection rates.

4.10.1.5 getAllSubnetworkConnectionNames

Synopsis

```
public void getAllSubnetworkConnectionNames(NameAndStringValue_T[] subnetName,
                                            short[] connectionRateList,
                                            int how_many,
                                            NamingAttributesList_THolder
nameList,
                                            NamingAttributesIterator_IHolder
nameIt)
      throws ProcessingFailureException
```

Description

The NML-EML interface allows the NMS to retrieve the names of all the SNCs contained within an NMS-specified Multilayer Subnetwork (MLSN).

4.10.1.6 getSNCsByUserLabel

Synopsis

```
void getSNCsByUserLabel(
                         in string userLabel,
                        out subnetworkConnection::SubnetworkConnectionList_T
sncList)
   raises (ProcessingFailureException)
```

Description

This interface returns the SNC structures by userLabel. Specify the userLabel as a parameter. This operation does not use an iterator because the number of SNCs returned is expected to be 1.

Parameters

- string userLabel—The user label of the SNCs to be retrieved.
- SubnetworkConnection_T sncList—The SNCs retrieved.

Throws

```
raises ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_NE_COMM_LOSS - Raised when communication to the ME is lost.
```

4.10.2 MPLS-TP LSP Inventory Interfaces

This section describes the following interfaces:

- 4.10.2.1 getSNC
- 4.10.2.2 getAllSubnetworkConnectionsWithTP
- 4.10.2.3 getAllSubnetworkConnectionsNamesWithTP

4.10.2.1 getSNC

Synopsis

```
void getSNC(
                  in NamingAttributes_T         sncName,
                  out SubnetworkConnection_T    snc)
   raises (ProcessingFailureException)
```

Description

This interface returns the SNC structure for the LSP SNC identified by sncName, SID, and UniqueID. For PTF_10GE_4, PT_10GE_4, and PTSA_GE MPLS implementation, the SNC name is not guaranteed to be unique within the EMS. Each LSP is specified by a given sncName, Service ID, and Unique ID, which allows for discrimination between the two LSPs (if it is a protected MPLS-TP tunnel). This operation returns the SNC structure for the TP tunnel SNC identified by sncName.
Because TP tunnel names are not guaranteed to be unique in the network, you must provide the LSP name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>:LSP=<LSP number>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66:LSP=0.

Relevant Data Structures

```
SubnetworkConnection_T
        NameAndStringValue_T[]        name
        String                        userLabel
        String                        nativeEMSName
        String                        owner
        SNCState_T                    sncState
        ConnectionDirection_T         direction
        short                         rate
        StaticProtectionLevel_T       staticProtectionLevel
        SNCType_T                     sncType
        TPData_T[]                    aEnd
        TPData_T[]                    zEnd
        Reroute_T                     rerouteAllowed
        NetworkRouted_T               networkRouted
        NameAndStringValue_T[]        additionalInfo
```

The following table describes the additionalInfo parameter.

Name	Value	Description
tunnelKey	String	The tunnel key is composed in the following format: sourceNodeId:SourceTunNum:DestNodeId:DestTunNum
isProtLsp	Boolean	True, if the LSP being created is a protected LSP. False, if the LSP being created is a working LSP.

```
TPData_T
        NameAndStringValue_T[]  tpName
        TerminationMode_T             tpMappingMode
        LayeredParameters_T[]   transmissionParams
        NameAndStringValue_T[]  ingressTrafficDescriptorName
        NameAndStringValue_T[]  egressTrafficDescriptorName
LayeredParameters_T
        short                         layer
        NameAndStringValue_T[]  transmissionParams
```

The transmissionParams associated to the layer rate LR_MPLS_PATH (166) reports the two attributes described in the following table.

Name	Value	Description
switchState	String	Represents the administrative switch state: LOCKOUT or CLEAR.
operState	String	Represents the current operational state of each cross-connection: ACTIVE or STDBY.

The aEnd and zEnd CTPs report the LSP number as LspNum=ddd.

4.10.2.2 getAllSubnetworkConnectionsWithTP

Synopsis

```
public void getAllSubnetworkConnectionsWithTP(
NameAndStringValue_T[]                          tpName,
                                                short[]
      connectionRateList,
                                                int
      how_many,

SubnetworkConnectionList_THolder sncList,
                                                SNCIterator_IHolder
      sncIt)
      throws ProcessingFailureException
```

Description

This interface allows the NMS to request a list of LSP SNCs using the specified TP at the specified connection rates. The LSP layer rate is LR_MPLS_PATH (166). A TP can be a PTP. Because an MPLS-TP tunnel is terminated in the equipment and not in the PTP, all the MPLS-TP tunnels that end in the device and suspended by the PTP are returned.

4.10.2.3 getAllSubnetworkConnectionsNamesWithTP

Synopsis

```
void getAllSubnetworkConnectionNamesWithTP(
    NameAndStringValue_T[]          tpName,
                                                    short[]
      connectionRateList,
                                                    int
      how_many,

NamingAttributesList_THolder  nameList,

NamingAttributesIterator_IHolder nameIt)
```

Description

This interface allows the NMS to request a list of LSP SNC names using the specified TP at the specified connection rates. The LSP layer rate is LR_MPLS_PATH (166). A TP can be a PTP. Because an MPLS-TP tunnel is terminated in the equipment and not in the PTP, all the MPLS-TP tunnels that end in the device and suspended by the PTP are returned.

4.10.3 MPLS-TP Tunnel Link Inventory Interfaces

This section describes the following interfaces:

4.10.3.1 getTopologicalLink

Synopsis

```
public void getTopologicalLink(NameAndStringValue_T[] topoLinkName,
                               TopologicalLink_THolder topoLink)
      throws ProcessingFailureException
```

4.10.3.2 getAllTopologicalLinks

Synopsis

```
public void getAllTopologicalLinks(NameAndStringValue_T[] subnetName,
                                   int how_many,
                                   TopologicalLinkList_THolder topoList,
                                   TopologicalLinkIterator_IHolder topoIt)
      throws ProcessingFailureException
```

4.10.3.3 getAllTopologicalLinkNames

Synopsis

```
public void getAllTopologicalLinkNames(NameAndStringValue_T[] subnetName,
                                       int how_many,
                                       NamingAttributesList_THolder
nameList,
                                       NamingAttributesIterator_IHolder
nameIt)
      throws ProcessingFailureException
```

4.10.3.4 getAllTopLevelTopologicalLinks

Synopsis

```
public void getAllTopLevelTopologicalLinks(int how_many,
                                           TopologicalLinkList_THolder
topoList,
                                           TopologicalLinkIterator_IHolder
topoIt)
      throws ProcessingFailureException
(on EMS Mgr Impl)
```

4.10.3.5 getAllTopLevelTopologicalLinkNames

Synopsis

```
public void getAllTopLevelTopologicalLinkNames(int how_many,
                                               NamingAttributesList_THolder
nameList,

NamingAttributesIterator_IHolder nameIt)
```

```
        throws ProcessingFailureException
```

4.10.3.6 getTopLevelTopologicalLink

Synopsis

```
public void getTopLevelTopologicalLink(NameAndStringValue_T[] topoLinkName,
                                       TopologicalLink_THolder topoLink)
        throws ProcessingFailureException
```

4.10.4 MPLS-TP LSP Route Inventory Interfaces

The Route object represents the route of an SNC. An SNC route is represented as a partially ordered series of cross-connections through which the SNC travels. Only SNC layer rate cross-connections are part of the route. The cross-connections that are part of the route are listed from the NE on which the SNC starts to the NE on which it ends. LSP aEnd and zEnd are not available in cross-connections.
This section describes the following interfaces:

- 4.10.4.1 getRoute
- 4.10.4.2 getRouteAndTopologicalLinks
- 4.10.4.3 getAllCrossConnections

4.10.4.1 getRoute

Synopsis

```
public void getRoute(
NameAndStringValue_T[]  sncName,
                        boolean                 includeHigherOrderCCs,
                        Route_THolder           route)
        throws ProcessingFailureException
```

Description

The NML-EML interface allows the NMS to retrieve the attributes of a route using the NMS-specified SNC name. The route information, in the form of ordered cross-connections, must identify the resources allocated to the SNC at the time of the request and the way in which these resources are used.
The NMS specifies if it will retrieve resources in the SNC layer only. Because TP tunnel names are not guaranteed to be unique in the network, you must provide the LSP name in the following format:
<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>:LSP=<LSP number>
For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66:LSP=0.

Parameters

- NamingAttributes sncName—SNC name.
- boolean includeHigherOrderCCs—Specifies whether the higher order cross-connections of other SNCs used to carry the queried SNC must be included in addition to the cross-connections of the queried SNC.
- Route_T route—The route of the SNC.

Throws

```
Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when sncName does not reference an SNC object.

EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that
does not exist.

EXCPT_UNABLE_TO_COMPLY - Raised when the provided value for
includeHigherOrderCCs is True and the EMS is not able to provide higher order
cross-connections in the reply.
```

Relevant Data Structures

```
Route_THolder
            CrossConnect_T[]        value
CrossConnect_T
      boolean                   active;
      ConnectionDirection_T     direction;
      SNCType_T                 ccType;
      NameAndStringValue_T[][]        aEndNameList;
      NameAndStringValue_T[][]        zEndNameList;
      NameAndStringValue_T[]    additionalInfo;
```

The following table describes the relevant parameters returned in the additionalInfo field.

Name	Value	Description
fwLinkNum	Integer	Link number associated to the forward path of the LSP.
fwLocalLabel	Integer	Local label value associated to the forward path of the LSP.
fwOutLabel	Integer	Out label value associated to the forward path of the LSP.
rvtLinkNum	Integer	Link number associated to the reverse path of the LSP.
rvtLocalLabel	Integer	Local label value associated to the reverse path of the LSP.
rvtOutLabel	Integer	Out label value associated to the reverse path of the LSP.

4.10.4.2 getRouteAndTopologicalLinks

Synopsis

```
void getRouteAndTopologicalLinks(
         in NamingAttributes_T            sncName,
         out Route_T                 route,
         out TopologicalLinkList_T  topologicalLinkList)
        raises(ProcessingFailureException);
```

Description

This interface returns the route, in terms of cross-connections, for the LSP SNC. The LSP SNC name is specified as a parameter. This interface also returns the list of topological links for that SNC that are used in the route. Because TP tunnel names are not guaranteed to be unique in the network, you must provide the LSP name in the following format:

<Native TP-Tunnel name>:TKEY=<TP-Tunnel key>:LSP=<LSP number>

For example, tunnel192_193:TKEY=3.3.3.0:55:5.5.5.0:66:LSP=0.

Parameters

- NamingAttributes sncName—SNC name.
- Route_T route—The route of the SNC.
- TopologicalLinkList_T topologicalLinkList—The list of the topological links.

Throws

```
Raises ProcessingFailureException:
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when sncName does not reference an SNC object.
EXCPT_ENTITY_NOT_FOUND - Raised when sncName references an SNC object that
does not exist.
```

4.10.4.3 getAllCrossConnections

Synopsis

```
void getAllCrossConnections(
NameAndStringValue_T[]  managedElementName,
                                    short[]
      connectionRateList,
                                    int                     how_many,
                                    CrossConnectList_THolder     ccList,
                                    CCIterator_IHolder           ccIt)
      throws ProcessingFailureException
```

Description

This interface allows the NMS to request a list of cross-connections for the specified ME at the specified layer rate (LR_MPLS_PATH). This operation returns cross-connections between CTPs and FTPs. This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- NamingAttributes_T managedElementName—ME name for which cross-connections must be retrieved.
- LayerRateList_T connectionRateList—List of rates for which cross-connections must be retrieved. If the list is empty, an INVALID_INPUT exception is raised.
- unsigned long how_many—Maximum number of cross-connections to report in the first batch.
- CrossConnectList_T ccList—First batch of cross-connections.
- CCIterator_I ccIt—Iterator to retrieve the remaining cross-connections.

Throws

```
Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised if connectionRateList is empty or contains
invalid rates, or if managedElementName does not reference an ME.

EXCPT_ENTITY_NOT_FOUND - Raised when managedElementName references an ME
object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication to an ME is lost.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that
the EMS can support has been reached.
```

4.10.5 MPLS-TP Protection (Inventory)

The relationship between an MPLS-TP tunnel circuit and its LSP circuits is provided by inventorying the MPLS-TP tunnel circuit. Additional parameters identify the working and protected LSPs. These parameters are taken into account whenever the LSP circuits are inventoried for further details.

For the LSPs, the current network implementation provides the same sncName and SID as for the TP tunnel they are part of. LSPs are addressed individually by specifying the unique ID.

The following table describes the names of the additional parameters returned by the TP tunnel circuit.

Parameter Name	Valid Values	Description
workLspName	String	Name of the working LSP circuit.
workLspSid	Integer	SID of the working LSP circuit.
workLspUniqueID	Integer	Unique ID of the working LSP circuit.
workLspNum	Integer	Working LSP number.

Parameter Name	Valid Values	Description
protLspName	String	Name of the protected LSP circuit.
protLspSid	Integer	SID of the protected LSP circuit.
protLspUniqueID	Integer	Unique ID of the protected LSP circuit.
protkLspNum	Integer	Protected LSP number.

4.10.6 BFD Templates, Node ID, Label Range, and ME OperationMode Inventory Interface

This section describes the following interface:

- 4.10.6.1 getManagedElement

4.10.6.1 getManagedElement

Synopsis

```
void getManagedElement(
NameAndStringValue_T[] managedElementName,
                                    ManagedElement_THolder me)
      throws ProcessingFailureException
```

Description

This interface returns the ME with all the relevant information about BFD template, static and dynamic label range, node ID, and router ID reported in the additionalInfo parameter.

Relevant Data Structures

```
ManagedElement_THolder
      ManagedElement_T              value
ManagedElement_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
      String                        location
      String                        version
      String                        productName
      CommunicationState_T    communicationState
      boolean                 emsInSyncState
      short[]                       supportedRates
      NameAndStringValue_T[]  additionalInfo
```

The following table describes the additionalInfo parameters involved.

Name	Type	Valid Values	Description
bfdTable.Count	Integer	—	Number of rows in the BFD template table.
bfdTable.Name	String	Up to 31 characters in length	Name of the BFD template.
bfdTable.Unit	String	Sequence of comma-separated: • Millisec • Microsec	Unit for received and transmitted interval capability.

Name	Type	Valid Values	Description
bfdTable.MinTxInterval	String	millisec—33 to 999 microsec—33000 to 999000	Minimum transmission interval capability.
bfdTable.MinRxInterval	String	millisec—33 to 999 microsec—33000 to 999000	Minimum receive interval capability.
bfdTable.Multiplier	String	3 to 50	Multiplier.
bfdTable.SingleHop	String	• True • False	Single-hop information. In Prime Optical 10.7, true is the only allowed value.
staticMinLabel	Integer	16 to 8000	—
staticMaxLabel	Integer	16 to 8000	—
dynamicMinLabel	Integer	16 to 8000	—
dynamicMaxLabel	Integer	16 to 8000	—
nodeId	String	IP address (IPv4)	—
routerIdName	String	Up to 200 characters	MPLS-TP router ID name; for example, loopback0.
routerIdIpAddr	String	IP address (IPv4)	—
routerIdMask	String	IP address mask (IPv4)	—
OperationMode	String	Current ME operation mode	Read-only attribute. The value can be CTC Mode or IOS Mode.
PtSystemState	String	notReady, Ready	PT System Status Configuration

Each BFD template is characterized by a sequence of attributes. The information of all the BFD templates on an ME is modeled as a BFD table. The TMF standard anticipates to model tables in the following way:

- An attribute ("bfdTable.Count") reporting the number of rows of the table.
- For every column in the table, an attribute that consists of a string of comma-separated values. The number of comma-separated values in each parameter must be equal to the value of the bfdtableCount attribute.

The following table is an example of the BFD table.

BFD Name	Single Hop	Unit	minTxInterval	minRxInterval	Multiplier
Bfd1	True	Millisec	300	400	3
Bfd2	True	Millisec	500	600	6
Bfd3	True	Microsec	700000	800000	9

GateWay/CORBA NBI Result

```
bfdTable.Count =                    3
bfdTable.Name=                      "Bfd1, BFD2, BFD3"
bfdTable.SingleHop=                 "true, true, true"
bfdTable.Unit=                      " Millisec, Millisec, Microsec"
bfdTable.MinTxInterval=             "300, 500, 700000"
bfdTable.MinRxInterval=             "400, 600, 800000"
```

```
bfdTable.Multiplier=                    "3, 6, 9"
```

4.11 Pseudowire Inventory Interfaces

This section describes the following pseudowire inventory interfaces:

- 4.11.1 Pseudowire FDFr Inventory Interfaces
- 4.11.2 Pseudowire Class, Loopback, and QoS Policy Inventory Interface

4.11.1 Pseudowire FDFr Inventory Interfaces

This section describes the following interfaces:

- 4.11.1.1 getFDFr
- 4.11.1.2 getAllFDFrs
- 4.11.1.3 getFDFrsWithTP
- 4.11.1.4 getAllFDFrNames
- 4.11.1.5 getFDFrNamesWithTP
- 4.11.1.6 getFDFrsByUserLabel
- 4.11.1.7 getFDFrRoute

4.11.1.1 getFDFr

Synopsis

```
public void getFDFr(
NameAndStringValue_T[]        fdfrName,
                        FlowDomainFragment_THolder    fdfrHolder)
      throws ProcessingFailureException
```

Description

This interface returns the FDFr representing the pseudowire. The pseudowire name is provided as a parameter.

Parameters

- NamingAttributes_T fdfrName—The name of the FDFr to be retrieved. The pseudowire name is provided as:
  <PW native name>:SID=nn.
- FlowDomainFragment_T fdfr—The FDFr structure retrieved.

Throws

```
Raises ProcessingFailureException
EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.
EXCPT_INVALID_INPUT - Raised when fdfrName does not reference an FDFr object.
EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an FDFr object that
does not exist.
EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in
this operation.
```

Relevant Data Structures

```
FlowDomainFragment_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
```

```
ConnectionDirection_T   direction
LayeredParameters_T     transmissionParams
NameAndStringValue_T[][]      aEnd
NameAndStringValue_T[][]      zEnd
TPData_T[]              aEnd
TPData_T[]              zEnd
String                        networkAccessDomain
boolean                 flexible
AdministrativeState_T   administrativeState
SNCState_T              fdfrState
String                        fdfrType
NameAndStringValue_T[]  additionalInfo
```

The transmission parameters report the following information.

Name	Type	Description
pwType	String	Valid values are: • Ethernet • VLAN
bw	Long	Receiving bandwidth to be associated to the MPLS-TP tunnel.
bwUnits	String	Valid values are: • Kbps • Mbps • Gbps
isRedundant	Boolean	True, if the pseudowire is redundant.
isDualHomedPeer	Boolean	True, if dual-home peer is set.
enableDelayValue	Integer	Enable delay value.
disableDelayValue	Integer	Disable delay value.
isNeverDisDelay	Boolean	True, if the disable delay is never set.
isDisableDelay	Boolean	Valid values are: • True (if disable delay is set) • False

The additionalInfo parameter contains the following information.

Name	Type	Description
ServiceID	Integer	Service ID associated by the network to the service.

For FDFr, TMF introduced the AdministrativeState parameter. It assumes the following values:

```
AdministrativeState_T
int                     value
            int                     _AS_Locked = 0;
int                     _AS_Unlocked = 1;
```

```
MTNM version 3.5 has defined the aEnd and zEnd FDFr data types as an array of
NameAndStringValue_T rather than an array of TPData_T as for the SNC
structure. Prime Optical has substituted the NameAndStringValue_T with
TPData_T in the Interface Design Languages (IDLs).
TPData_T
      NameAndStringValue_T[]  tpName
      TerminationMode_T             tpMappingMode
      LayeredParameters_T[]   transmissionParams
      NameAndStringValue_T[]  ingressTrafficDescriptorName
      NameAndStringValue_T[]  egressTrafficDescriptorName
LayeredParameters_T
      short                         layer
      NameAndStringValue_T[]  transmissionParams
```

The following table describes the layered parameters that characterize the pseudowire endpoints.

Name	Type	Description
outerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200
innerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
isExactTag	Boolean	True, if the VLAN tag is exact.
pwClass	String	Pseudowire class name.
vcID	Integer	VC ID.
staticLabel	Integer	Local label.
bkpPwClass	String	Backup (protected) pseudowire class.
bkpVcID	Integer	Backup (protected) VC ID.
bkpStaticLabel	Integer	(Optional) Specify a static value for the backup local label.
manLoadBal	—	Not supported.
qosTableMap	String	QoS table map name (TC profile).

Name	Type	Description
qosIngPolicy	String	QoS ingress policy map name (TC profile).
qosEgrPolicy	String	QoS egress policy name (TC profile).
rewriteOper	String	Valid values are: • REWRITE_NONE • PUSH_1 • PUSH_2 • POP_1 • POP_2 • TRANSLATE_1_TO_1 • TRANSLATE_1_TO_2 • TRANSLATE_2_TO_1 • TRANSLATE_2_TO_2
rewriteOuterVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200
rewriteOuterVlanTag	Integer	—
rewriteInnerVlanTpId	String	Valid values are: • TP_NONE • DOT1Q • DOT1AD • TP_9100 • TP_9200 In Prime Optical 10.7, for double tagging, the only allowed value is DOT1Q.
rewriteInnerVlanTag	Integer	—
rewriteIsSymmetric	Boolean	Valid values are: • True • False
enableStatistics	Boolean	Enables statistics counter. Valid values are: • True • False
statisticsCategory	String	Valid values are: • INGRESS • EGRESS • BOTH

4.11.1.2 getAllFDFrs

Synopsis

```
public void getAllFDFrs(
  NameAndStringValue_T[]      fdName,
```

```
                                int                          how_many,
                                short[]                 connectivityRateList,
                                FDFrList_THolder             fdfrList,
                                FDFrIterator_IHolder    fdfrIt)
      throws ProcessingFailureException
```

Description

This interface allows retrieval of a list of FDFrs for the specified FD at the specified connectivity rates. This operation uses an iterator to allow the NMS to deal with a large number of objects.

Parameters

- NamingAttributes_T fdName—FD name.
- LayerRateList_T connectivityRateList—List of FDFr rates to be reported. If an empty list is specified, all FDFrs of all rates are to be reported.
- unsigned long how_many—Maximum number of FDFrs to be reported in the first batch.
- FDFrList_T fdfrList—First batch of FDFrs.
- FDFrIterator_I fdfrIt—Iterator to retrieve the remaining FDFrs.

Throws

Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when fdName does not reference an FD or connectivityRateList contains undefined values.

EXCPT_ENTITY_NOT_FOUND - Raised when fdName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.11.1.3 getFDFrsWithTP

Synopsis

```
public void getFDFrsWithTP(
     NameAndStringValue_T[]   cptpName,
                              int                          how_many,
                              FDFrList_THolder        fdfrList,
                              FDFrIterator_IHolder         fdfrIt)
      throws ProcessingFailureException
```

Description

This interface allows the NMS to request the EMS for the FDFrs that are connected to a specified FP or CPTP. If the FP is an input parameter, not more than one FDFr is returned.

Parameters

- NamingAttributes_T tpName—The name of the FP or CPTP for which the connected FDFrs are returned.
- unsigned long how_many—Maximum number of FDFrs to be reported in the first batch.
- FDFrList_T fdfrList—First batch of FDFrs.
- FDFrIterator_I fdfrIt—Iterator to retrieve the remaining FDFrs.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when tpName does not reference an FP or CPTP.

EXCPT_ENTITY_NOT_FOUND - Raised when tpName references an object that does not exist.

EXCPT_TOO_MANY_OPEN_ITERATORS - Raised when maximum number of iterators that the EMS can support has been reached.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

4.11.1.4 getAllFDFrNames

Synopsis

```
void getAllFDFrNames(
NameAndStringValue_T[]        fdName,
                        int                           how_many,
            short[]                           connectivityRateList,
                  NamingAttributesList_THolder  nameList,
                        NamingAttributesIterator_IHolder    nameIt)
      throws ProcessingFailureException
```

Description

This interface returns the list of all the pseudowire names configured in an FD.

4.11.1.5 getFDFrNamesWithTP

Synopsis

```
void getFDFrNamesWithTP(
NameAndStringValue_T[]        cptpName,
                        int                           how_many,
                        NamingAttributesList_THolder  nameList,
                        NamingAttributesIterator_IHolder    nameIt)
      throws ProcessingFailureException
```

Description

This interface returns the list of all the pseudowire names that have at least one EFP in the CPTP provided as an input parameter.

4.11.1.6 getFDFrsByUserLabel

Synopsis

```
public void getFDFrsByUserLabel(
String                  userLabel,
                                     FDFrList_THolder  fdfrs)
      throws ProcessingFailureException
```

Description

This interface returns the FDFr by userLabel. The userLabel is provided as a parameter. The operation does not use an iterator because the number of FDFrs returned is expected to be 1.

Parameters

- string userLabel—The user label of the FDFrs to be retrieved.
- FDFrList_T fdfrs—The list of identified FDFrs.

Throws

Raises globaldefs::ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation

4.11.1.7 getFDFrRoute

Synopsis

```
public void getFDFrRoute(
NameAndStringValue_T[]  fdfrName,
                                FDFrRoute_THolder             route)
      throws ProcessingFailureException
```

Description

This interface returns the FDFr route. The FDFr name is specified as a parameter. Use this interface only if the route is provisioned by the NMS.

Parameters

- NamingAttributes_T fdfrName—The FDFr name. The pseudowire name is provided as: <PW native name>:SID=nn.
- FDFrRoute_T route—The FDFr route.

Throws

Raises ProcessingFailureException

EXCPT_INTERNAL_ERROR - Raised in case of nonspecific EMS internal failure.

EXCPT_INVALID_INPUT - Raised when fdfrName does not reference an FDFr object.

EXCPT_ENTITY_NOT_FOUND - Raised when fdfrName references an FDFr object that does not exist.

EXCPT_NE_COMM_LOSS - Raised when communication is lost to an ME involved in this operation.

Relevant Data Structures

```
FDFrRoute_THolder
            MatrixFlowDomainFragment_T[]  value
```

An MFDFr object represents the portion of an FDFr within an MFD and inside an NE.An MFDFr is primarily used in the specification of an FDFr route where the route is specified by the NMS.

```
MatrixFlowDomainFragment_T
      ConnectionDirection_T         direction
                                    (Bidirectional for Ethernet)
      LayeredParameters_T           transmissionParams
                                    (LR_PW and related connectionless
parameters)
      NameAndStringValue_T[][]      aEnd
Pseudowire endpoint is always one (pseudowire EFP).
      NameAndStringValue_T[][]      zEnd
```

Pseudowire path points (PP). These are CTP endpoints of TP tunnels. It can be one or two, depending on the protection schema.

boolean flexible

FDFr can be fixed or flexible.

boolean active

MFDFr has been successfully activated on the ME.

String mfdfrType

Can be point-to-point, point-to-multipoint (E-tree), multipoint

NameAndStringValue_T[] additionalInfo

The additionalInfo parameter allows the communication of additional information (not explicitly modeled) from the EMS to the NMS. The following table describes the path1Link and path2Link parameters.

Name	Type	Description
path1Link	String	MPLS-TP link name that ends in the first path point of the cross-connection.
path2Link	String	MPLS-TP link name that ends in the second path point of the cross-connection (if supported by the protection schema).

Figure 4-8 shows the unprotected pseudowire.

Figure 4-8: Unprotected Pseudowire

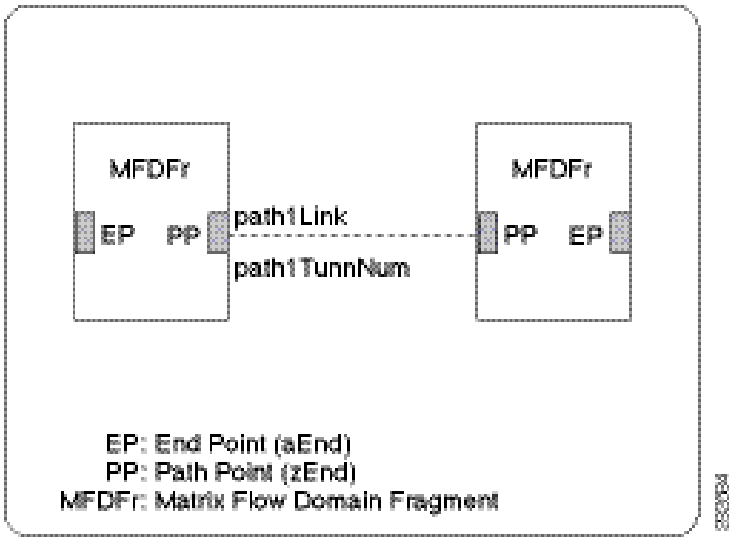


Figure 4-9 shows the protected pseudowire.

Figure 4-9: Protected Pseudowire

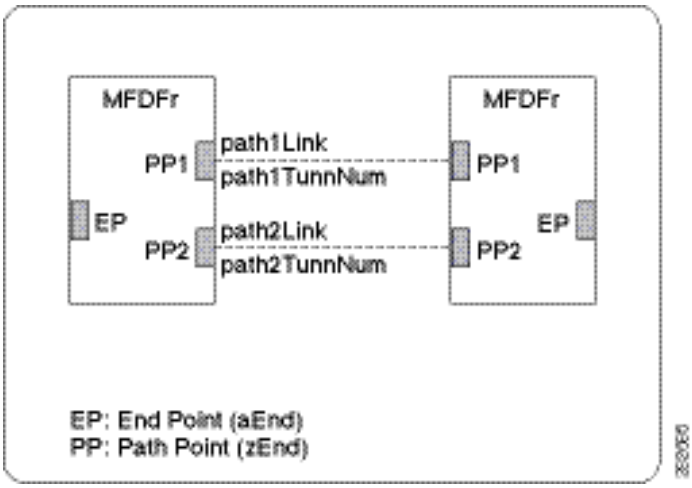
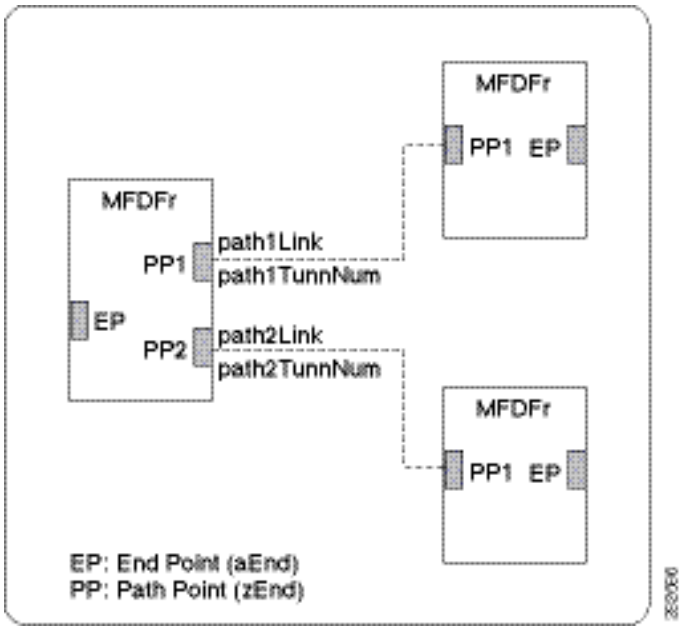


Figure 4-10 shows the dual-homed protected pseudowire.

Figure 4-10: Protected Pseudowire-Dual Homed



4.11.2 Pseudowire Class, Loopback, and QoS Policy Inventory Interface

This section describes the following interface:

- 4.11.2.1 getManagedElement

4.11.2.1 getManagedElement

Synopsis

```
void getManagedElement(
NameAndStringValue_T[] managedElementName,
                                        ManagedElement_THolder me)
      throws ProcessingFailureException
```

Description

This interface returns the ME with all the relevant information about BFD template, static and dynamic label range, node ID, and router ID reported in the additionalInfo parameter.

Relevant Data Structures

```
ManagedElement_THolder
      ManagedElement_T              value
ManagedElement_T
      NameAndStringValue_T[]  name
      String                        userLabel
      String                        nativeEMSName
      String                        owner
      String                        location
      String                        version
      String                        productName
      CommunicationState_T    communicationState
```

```
boolean                 emsInSyncState
short[]                 supportedRates
NameAndStringValue_T[]  additionalInfo
```

The following table describes the additionalInfo parameters involved. All fields contain a sequence of comma-separated values.

Name	Type	Valid Values	Description
pwClassTable.Count	Integer	—	Number of rows in the pseudowire class table.
pwClassTable.Name	String	Up to 200 characters	Pseudowire class name.
pwClassTable.Encaps	String	MPLS	Encapsulation type. The default is MPLS.
pwClassTable.Protocol	String	• NONE • LDP	Protocol type.
pwClassTable.Interwork	String	• VLAN • ETHERNET • NONE	Minimum receive interval capability.
pwClassTable.CtrlWord	String	• True • False	Control word.
pwClassTable.MasterRed	String	• True • False	Master redundancy.
pwClassTable.PrefPathEnabled	String	• True • False	Preferred Path enabled.
pwClassTable.TunnType	String	• TP • TE	Tunnel type. Considered only if PrefPathEnabled is true.
pwClassTable.TunnNum	Integer	—	Tunnel number. Considered only if PrefPathEnabled is true.
pwClassTable.DisFallback	String	• True • False	Disable fallback. Considered only if PrefPathEnabled is true.
pwClassTable.SeqEnabled	String	• True • False	Sequencing enabled.
pwClassTable.SeqOper		• Transmit • Receive • Both	Sequencing operation. Considered only if SeqEnabled is true.
pwClassTable.ResyncTimer	Integer	—	Resync timer. Considered only if SeqEnabled is true.

Name	Type	Valid Values	Description
pwClassTable.StaticOAMEnabled	String	• True • False	Static OAM enabled.
pwClassTable. StaticOamClass	String	—	Static OAM class. The maximum length is 31 characters. Considered only if StatOAMEnabled is true.
pwClassTable.BFDoVCCVEnabled	String	• True • False	BFD over CCV.
pwClassTable.BfdTemplate	String	—	BFD template. Considered only if BFDoVCCVEnabled is true. The maximum length is 31 characters.
pwClassTable. AcStat	String	• True • False	AC status signaling. Considered only if BFDoVCCVEnabled is true.
qoSPolicies	String	Comma-separated policy names	Applies to the following TCProfile names: • class-map xxxx • policy-map yyy • action zzz • table-map kkk
routerIdName	String	Up to 200 characters	MPLS-TP router ID name; for example, loopback0.
routerIdIpAddr	String	IP address (IPv4)	Loopback IP address.
routerIdMask	String	IP address mask (IPv4)	Loopback net mask.

Each pseudowire class is characterized by a sequence of attributes. The information about all pseudowire classes on a given ME is modeled as a pseudowire table. The TMF standard anticipates to model tables in the following way:

- An attribute ("pwClassTable.Count") reporting the number of rows of the table.
- For every column of the table, an attribute that consists of a string of comma-separated values. The number of comma-separated values in each parameter must be equal to the value of the pwClassTable.Count attribute.

The following is an example of the pseudowire class table.

PW Class Name	Encapsulation	Protocol	Interwork	Ctrl Word	Preferred Path	Tunnel Type	Tunnel Number	Enable Sequencing	Sequencing Mode
Pw-300	MPLS	NONE	ETHERNET	true	true	TP	300	true	TRANSMIT
Pw-301	MPLS	LDP	ETHERNET	false	false	TP	301	true	TRANSMIT
Pw-302	MPLS	LDP	VLAN	true	true	TP	302	true	TRANSMIT

GateWay/CORBA NBI Result

```
pwClassTable.Count =          3
pwClassTable.Name=            "Pw-300, Pw-301, Pw-302"
pwClassTable.Encaps=          "MPLS, MPLS, MPLS"
pwClassTable.Protocol=        "NONE, LDP, LDP"
pwClassTable.Interwork=       "ETHERNET, ETHERNET, VLAN"
pwClassTable.CtrlWord=        "true, false, true"
pwClassTable.PrefPath=        "true, false, true"
pwClassTable.TunnType=        "TP, TP, TP"
pwClassTable.TunnNum=         "300, 301, 302"
pwClassTable.EnableSeq=       "true, true, true"
pwClassTable.SeqMode=         "Transmit, true, true"
...........
```

4.12 Alarm Summaries

Synopsis

```
void getAllActiveAlarms(NameAndStringValue_T[] meName,
                                String[]
      excludeProbCauseList,
                                PerceivedSeverity_T[]
      excludeSeverityList,
                                int                        how_many,
                                EventList_THolder          eventList,
                                EventIterator_IHolder            eventIt)
      throws ProcessingFailureException
```

Description

The network can generate Layer 2 service-related alarm summaries on Ethernet ports, channel group ports, and shelves. You can use the getAllActiveAlarms method to get information about the alarms. Provide the meName parameter as shown in the following table.

Parameter Name	Value Type	Example
meName	AID	meName[0] = new NameAndStringValue_T("EMS", "Cisco Systems/Prime Optical")

Parameter Name	Value Type	Example
		meName [1] = new NameAndStringValue_T("ManagedElement", "CTM-454-141"); meName [2] = new NameAndStringValue_T("AID", "notificationId=10");

4.12.1 Alarm Summary Example—Channel Group Port

```
Filterable Data      [0] notificationId     = 105
                     [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-124; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] nativeEMSName      = shelf1
                     [3] nativeProbableCause = Wokring Pseudowire Local Ac Rx
Port Fault
                     [4] objectType         = OT_EQUIPMENT_HOLDER
                     [5] emsTime            = 20110613133455.0Z
                     [6] neTime             = 20110613124547.0Z
                     [7] isClearable        = true
                     [8] layerRate          = 0 (Not used)
                     [9] probableCause      = UNIDENTIFIED
                    [10] probableCauseQualifier = Wokring Pseudowire Local Ac
Rx Port Fault
                    [11] perceivedSeverity  = PS_MAJOR
                    [12] serviceAffecting   = SA_SERVICE_AFFECTING
                    [12] affectedTPList     = 0 Affected PTs
                    [14] additionalText     = CHNGRP-1-1
```

Table 4-4: Alarm Entity and Probable Cause

Alarm Entity Category	Probable Cause
Pseudowire	• PRT-LOC-PW-NOT-FRWD • WKG-REM-PW-NOT-FRWD • PRT-REM-PW-NOT-FRWD • WKG-PW-CP-DOWN PRT-PW-CP-DOWN • WKG-PW-CC-DOWN PRT-PW-CC-DOWN • PW-WKSPWR • WKG-PW-LOC-AC-TX-FLT PRT-PW-LOC-AC-TX-FLT • WKG-PW-LOC-AC-RX-FLT PRT-PW-LOC-AC-RX-FLT • WKG-PW-REM-AC-TX-FLT • PRT-PW-REM-AC-TX-FLT • WKG-PW-REM-AC-RX-FLT

Alarm Entity Category	Probable Cause
	• PRT-PW-REM-AC-RX-FLT • PW-CLEAR-ALL
MPLS-TP Tunnel/LSP	• WKG-LSP-DOWN • PRT-LSP-DOWN • WKG-LSP-AIS • PRT-LSP-AIS • WKG-LSP-RDI • PRT-LSP-RDI • WKG-LSP-LKR • PRT-LSP-LKR • WKG-LSP-LDI • PRT-LSP-LDI • LSP-LDI • BFD-DOWN • TP-WKSWPR • TP-TUNNEL-DOWN • WKG-TP-LOCKOUT • PRT-TP-LOCKOUT • TP-LSP-FAULT-CLEAR
EVC	EFP-FAIL

The returned alarms have affectedObjList (an array of nameAndVaue type_T), which reports the list of affected Layer 2 services.

4.12.2 Pseudowire Alarm Event Example

```
Filterable Data      [0] notificationId     = 92
                     [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
EquipmentHolder=/rack=1/shelf=1
                     [2] nativeEMSName      = shelf1
                     [3] nativeProbableCause = Working Remote Pseudowire Not
Forwarding
                     [4] objectType         = OT_EQUIPMENT_HOLDER
                     [5] emsTime            = 20110613133447.0Z
                     [6] neTime             = 20110613123657.0Z
                     [7] isClearable        = true
                     [8] layerRate          = 0 (Not used)
                     [9] probableCause      = UNIDENTIFIED
                    [10] probableCauseQualifier = Working Remote Pseudowire
Not Forwarding
                    [11] perceivedSeverity  = PS_WARNING
                    [12] serviceAffecting   = SA_NON_SERVICE_AFFECTING
                    [12] affectedTPList     = 0 Affected PTs
                    [14] additionalText     = PTSYS-1
```

```
                    [15] additionalInfo     = [0]
MultiLayerSubnetwork=Subnetwork-00000003; [1] IPAddress=10.58.65.117; [2]
VendorName=Cisco Systems
                    [16] isPropagated       = false
                    [17] aresFDN            =
                    [18] affectedObjList    = 2 Affected OBJs
                                        [0  ] = [0] EMS=Cisco
Systems/PrimeOptical; [1] FlowDomain=NP_1; [2] FlowDomainFragment=pw_33:SID=4
                                        [1  ] = [0] EMS=Cisco
Systems/PrimeOptical; [1] FlowDomain=NP_1; [2] FlowDomainFragment=pw_34:SID=5
```

4.12.3 MPLS-TP Tunnel Alarm Event Example

```
Filterable Data      [0] notificationId     = 91
                     [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-117; [2]
PTP=/rack=1/shelf=1/slot=4/ppm_holder=3/port=1
                     [2] nativeEMSName      = M6-65-
117//rack=1/shelf=1/slot=4/ppm_holder=3/port=1
                     [3] nativeProbableCause = Working Label Switched Path
Link Down Indication
                     [4] objectType         = OT_PHYSICAL_TERMINATION_POINT
                     [5] emsTime            = 20110613133447.0Z
                     [6] neTime             = 20110613090323.0Z
                     [7] isClearable        = true
                     [8] layerRate          = 106 (LR_Ten_Gigabit_Ethernet)
                     [9] probableCause      = UNIDENTIFIED
                    [10] probableCauseQualifier = Working Label Switched Path
Link Down Indication
                    [11] perceivedSeverity  = PS_MINOR
                    [12] serviceAffecting   = SA_NON_SERVICE_AFFECTING
                    [12] affectedTPList     = 0 Affected PTs
                    [14] additionalText     = ETH-4-3-1
                    [15] additionalInfo     = [0]
MultiLayerSubnetwork=Subnetwork-00000003; [1] IPAddress=10.58.65.117; [2]
VendorName=Cisco Systems
                    [16] isPropagated       = false
                    [17] aresFDN            =
                    [18] affectedObjList    = 1 Affected OBJs
                                        [0  ] = [0] EMS=Cisco
Systems/PrimeOptical; [1] SubnetworkConnection=TP-
2:TKEY=1.1.1.10:200:3.3.3.10:200
```

4.12.4 Channel Group Alarm Event Example

```
Filterable Data      [0] notificationId     = 106
```

```
                    [1] objectName         = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=M6-65-124; [2]
EquipmentHolder=/rack=1/shelf=1
                    [2] nativeEMSName      = shelf1
                    [3] nativeProbableCause = Wokring Pseudowire Local Ac Rx
Port Fault
                    [4] objectType         = OT_EQUIPMENT_HOLDER
                    [5] emsTime            = 20110617130307.0Z
                    [6] neTime             = 20110613124547.0Z
                    [7] isClearable        = true
                    [8] layerRate          = 0 (Not used)
                    [9] probableCause      = UNIDENTIFIED
                   [10] probableCauseQualifier = Wokring Pseudowire Local Ac
Rx Port Fault
                   [11] perceivedSeverity  = PS_MAJOR
                   [12] serviceAffecting   = SA_SERVICE_AFFECTING
                   [12] affectedTPList     = 0 Affected PTs
                   [14] additionalText     = CHNGRP-1-1
                   [15] additionalInfo     = [0]
MultiLayerSubnetwork=Subnetwork-3; [1] IPAddress=10.58.65.124; [2]
VendorName=Cisco Systems
                   [16] isPropagated       = false
                   [17] aresFDN            =
                   [18] affectedObjList    = 1 Affected OBJs
                                        [0  ] = [0] EMS=Cisco
Systems/PrimeOptical; [1] FlowDomain=NP_1; [2] FlowDomainFragment=PW_45:SID=9
```

4.13 Error Messages

This section lists the error messages generated by the GateWay/CORBA connectionless interfaces. It includes the following sections:

- 4.13.1 MPLS-TP Tunnel Error Messages
- 4.13.2 LSP Error Messages
- 4.13.3 Pseudowire Error Messages
- 4.13.4 EVC Error Messages
- 4.13.5 FD, MFD, and CPTP Error Messages

4.13.1 MPLS-TP Tunnel Error Messages

This section lists the MPLS-TP tunnel error messages. The error messages appear if you enter invalid parameters or values. For valid parameters and values, see:

- 4.6 MPLS-TP Provisioning Interfaces
- 4.10 MPLS-TP Inventory Interfaces

The following are the error messages:

- The MPLS-TP tunnel name is invalid for managing MPLS.
- The MPLS-TP tunnel was not found: nativeName.
- Only one endpoint was discovered for the TP tunnel.

- The TP tunnel does not exist.
- bwUnits must be Kbps, Mbps, or Gbps.
- aEnd and zEnd points must be in the same multilayer subnetwork.
- Source aEnd point is not available for TP tunnel creation.
- Destination zEnd point is not available for TP tunnel creation.
- You cannot create the TP tunnel SNC name based on node ID and tunnel number.
- The SID SID-Number is already in use on ME-Name.
- The BFD that you provided is invalid.
- The BFD userBfd is not in the following list of available BFDs: listOfAvailableBFDs.
- The CTP name is invalid.
- The MPLS-TP tunnel CTP name ctpNativeName is invalid. Use the format tunnelNum=nnn.
- The TP tunnel endpoint was discovered incorrectly.
- An internal error occurred while reading rows from the Tbl_MPLS_TP_EP table.
- An internal error occurred while reading rows from the MPLS_TP_TABLE table.
- An internal error occurred while reading rows from the MPLS_SERVICE_TABLE table.

4.13.2 LSP Error Messages

This section lists the LSP tunnel error messages. The error messages appear if you enter invalid parameters or values. For valid parameters and values, see:

- 4.6.2 MPLS-TP LSP Provisioning Interfaces
- 4.10.2 MPLS-TP LSP Inventory Interfaces

The following are the error messages:

- The LSP name is invalid for managing MPLS.
- The MPLS-TP tunnel for the LSP lspNativeName was not found.
- The LSP lspNativeName was not found.
- An LSP with the unique ID uniqueId was not found.
- You must provide an sncName and TKEY.
- The aEnd and zEnd LSP numbers must be the same.
- The aEnd and zEnd points must be in the same multilayer subnetwork.
- The LSP aEnd does not match the TP tunnel aEnd or zEnd.
- The TP tunnel already has working and protected LSPs.
- Source aEnd point is not available for TP tunnel creation.
- Destination zEnd point is not available for TP tunnel creation.
- The protected LSP number protLspNum is already being used by the working LSP.
- The working LSP number workLspNum is already being used by the protected LSP.
- The protected LSP already exists for the TP tunnel.
- The working LSP already exists for the TP tunnel.
- TopologicalLink input is not supported.
- neTPinclusions or neTpSncExclusions is invalid.
- neTpInclusions is invalid. The type is unsupported.
- neTpSncExclusions is invalid. The type is unsupported.
- An internal error occurred while reading rows from the MPLS_LSP_CTP table.

4.13.3 Pseudowire Error Messages

This section lists the pseudowire error messages. The error messages appear if you enter invalid parameters or values. For valid parameters and values, see:

- 4.7 Pseudowire Provisioning Interfaces
- 4.11 Pseudowire Inventory Interfaces

The following are the error messages:

- Category is not specified for the enableStatistics parameter.
- You must specify pwClass for managed endpoints.
- You must specify the pseudowire vcID for managed endpoints.
- The pseudowire layer rate must be LR_PW.
- The FD fdName does not exist in the EMS.
- Pseudowire FDFrs must have only one aEnd point.
- You have not entered a value for aEnd.
- Pseudowire FDFrs must have one or two zEnd points for dual-homed protection.
- The specified pseudowire endpoint ptpName is unavailable.
- The egress policy map egrPolMapName is not contained in the ME ME-Name.
- The ingress policy map ingrPolMapName is not contained in the ME ME-Name.
- The table map TabMapName is not contained in the ME ME-Name.
- Available endpoints on ME ME-Name could not be retrieved.
- You must specify pseudowire transmission parameters.
- Redundancy information is missing.
- The necessary free labels within the NE NE-Name could not be retrieved.
- The FDFr name format is invalid.
- checkPWClassCompliancy for node ME-Name failed because pwClass is not specified.
- A connection could not be created on the node ME-Name because the VC ID is in use.
- The operation failed because the VLAN IDs vlanIds are already configured under other service instances.

4.13.4 EVC Error Messages

This section lists the EVC error messages. The error messages appear if you enter invalid parameters or values. For valid parameters and values, see:

- 4.5 EVC Provisioning Interfaces
- 4.9 EVC Inventory Interfaces

The following are the error messages:

- The tpsToModify parameter must include the associated EFP parameters for VLAN-Based EVCs.
- The EVC layer rate must be LR_EVC.
- The drop to be added is on a new ME, so EFP for the internal drop is required.
- Unique ID could not be set on the EMS server.
- EVC constraints could not be set.
- Only ME inclusions are foreseen in EVC routing.
- The service has been created, but its SID is unavailable. Retrieve the SID using inventory APIs.
- The specified EVC endpoint ptp is unavailable.
- EVC endpoints on ME-Name could not be set.
- Category is not specified for the enableStatistics parameter.
- The FD fdName does not exist in the EMS.
- You have not entered a value for aEnd.
- Available endpoints on the ME ME-Name could not be retrieved.

- The FDFr name format is invalid.
- The operation failed because the VLAN IDs vlanIds are already configured under other service instances.

4.13.5 FD, MFD, and CPTP Error Messages

This section lists the FD, MFD, and CPTP error messages. The error messages appear if you enter invalid parameters or values. For valid parameters and values, see:

- 4.5.1 CPTP Provisioning Interfaces
- 4.5.2 MFD Provisioning Interfaces
- 4.5.3 Flow Domain Provisioning Interfaces
- 4.9.1 CPTP Inventory Interfaces
- 4.9.2 MFD Inventory Interfaces
- 4.9.3 Flow Domain Inventory Interfaces

The following are the error messages:

- An internal error occurred while reading rows from the Tbl_MATRIX_FLOW_DOMAIN_TABLE table.
- An internal error occurred while reading rows from the Tbl_FLOW_DOMAIN_TABLE table.
- An internal error occurred while reading rows from the Tbl_CONNLESS_TP_TABLE table.
- The FD fdName does not exist in the EMS.
- An unexpected error occurred while trying to retrieve the FD with user label userLabel from the EMS.
- The FD does not exist in the EMS.
- The ME ME-Name does not exist in the EMS.
- More than one MFD exists for the same NE/CPTP.
- The MFD mfdName does not exist in the EMS.
- The MFD is associated to an FD and thus cannot be deleted.
- The MFD is fixed and thus cannot be deleted.
- The MFD mfdName is not associated to the selected ME ME-Name.
- Some TPs are already associated to the existing MFDs.
- At least one TP is not a CPTP.
- Some TPs are in the unassigned PortTPRoleState.
- Some TPs are still being used by an FDFr.
- At least one TP is not assigned to the specified MFD.
- The CPTP name you specified is invalid.
- An FD with the name fdName was not found.

5 Notifications, Fault Management, and Probable Cause Mapping

This chapter includes information on Cisco Prime Optical Gateway/CORBA's notifications, fault management, and probable cause mapping. This chapter includes the following sections:

5.1 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.

The Prime Optical server and the Operations Support System (OSS) host must have Domain Name System (DNS) enabled in order to receive notifications. OSS clients running behind a firewall must configure an IIOP listening port to enable the TCP connection with the notification server. Prime Optical must configure the notification service listening port number to allow OSS clients to communicate with the notification service.

Prime Optical creates the notification channel with the QoS parameters shown in the following table.

Table 5-1: QoS Parameters for the Notification Channel

Parameter	Value
ConnectionReliability	BestEffort
StartTimeSupported	false
OrderPolicy	FifoOrder
DiscardPolicy	FifoOrder
MaxEventsPerConsumer	10000

Prime Optical sets the RejectNewEvents administration property of the channel to true, as defined by the TMF.

Every notification that Prime Optical generates has the QoS parameters listed in the following table. These parameter values are defined by the TMF and cannot be changed.

Table 5-2: 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. GateWay/CORBA forwards Prime Optical-specific alarms to the NMS.

For NE-related alarms, Prime Optical inserts the original probable cause in the alarm's nativeProbableCause field. The TMF-mapped probable cause is available in the probableCause field.

A.2Resolving the Notification Service

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

1. 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.
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.
3. Resolve 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 file://localhost/CTM_baseDir/cfg/NotificationService.ior. You can specify a URL that points to a remote web server.

5.2 Tips to Improve Notification Throughput

While 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.

Because all CORBA calls are blocking, the notification server thread blocks 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();
        }
}
```

As shown in the example, you must catch all exceptions, or you will stop receiving notifications.

5.3 Supported Events

The following sections list the attributes of each event.

5.3.1 Object Creation

The following table lists the object creation types and attributes.

Table 5-3: Object Creation

Attribute Name	Type	Supported by Prime Optical?
notificatId	string	Yes. The uniqueness and sequence of the notification ID 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 the following table.

Table 5-4:Object Creation Notifications

Object Type	Type Sent in remainder_of_body for Object Creation Event
OT_MANAGED_ELEMENT	managedElement::ManagedElement_T
OT_MULTILAYER_SUBNETWORK	multiLayerSubnetwork::MultiLayerSubnetwork_T
OT_TOPOLOGICAL_LINK	topologicalLink::TopologicalLink_T
OT_SUBNETWORK_CONNECTION	subnetworkConnection::SubnetworkConnection_T
OT_EQUIPMENT	equipment::Equipment_T
OT_PROTECTION_GROUP	protection::ProtectionGroup_T
OT_L2TOPOLOGY	MultiLayerSubnetwork::L2Topology_T
OT_MLVLAN	MultiLayerSubnetwork::MLVLAN_T
OT_ROLL	MultiLayerSubnetwork::Roll_T
OT_VCAT	SubnetworkConnection::VCAT_T

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

5.3.2 Object Deletion

The following table lists the object deletion types and attributes.

Table 5-5: Object Deletion

Attribute Name	Type	Supported by Prime Optical?
notificationId	string	Yes. The uniqueness and sequence of the notification ID 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 the following table.

Table 5-6: Object Deletion Notifications

Object Type	Type Sent in remainder_of_body for Object Deletion Event
OT_MANAGED_ELEMENT	managedElement::ManagedElement_T
OT_MULTILAYER_SUBNETWORK	multiLayerSubnetwork::MultiLayerSubnetwork_T

Object Type	Type Sent in remainder_of_body for Object Deletion Event
OT_TOPOLOGICAL_LINK	topologicalLink::TopologicalLink_T
OT_SUBNETWORK_CONNECTION	subnetworkConnection::SubnetworkConnection_T
OT_EQUIPMENT	equipment::Equipment_T
OT_PROTECTION_GROUP	protection::ProtectionGroup_T
OT_L2TOPOLOGY	MultiLayerSubnetwork::L2Topology_T
OT_MLVLAN	MultiLayerSubnetwork::MLVLAN_T
OT_ROLL	MultiLayerSubnetwork::Roll_T
OT_VCAT	SubnetworkConnection::VCAT_T

- The number of equipment holders for all NEs supported by Prime Optical is constant. To reduce traffic between Prime Optical and the NMS, Prime Optical does not generate object deletion events for equipment holders. The number of PTPs on equipment is also constant. Therefore, Prime Optical generates only object deletion events for equipment.
- If there is a card object deletion, GW/CORBA sends an object deletion event for the card and the termination points. In some cases, it is not possible to identify the termination point of the card because the object deletion event for the card arrives before the object deletion event for the termination point. In this case, the name of the termination point will report only the reference to the slot.

5.3.3 Attribute Value Change

The following table lists the attribute value change (AVC) types and attributes.

Table 5-7: Attribute Value Change

Attribute Name	Type	Supported by Prime Optical?
notificationId	string	Yes. The uniqueness and sequence of the notification ID 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.
attributeList	notifications::NVList_T	Yes.

Attribute value change notifications are sent to the NMS for the objects and attributes listed in the following table.

Table 5-8: Attribute Value Change Notifications

Object Type	Attributes
OT_EMS	nativeEMSName
OT_MANAGED_ELEMENT	nativeEMSName—This field reflects the new managed element name if it is changed. • location • version • productName
OT_MULTILAYER_SUBNETWORK	NativeEMSName
OT_TOPOLOGICAL_LINK	None
OT_SUBNETWORK_CONNECTION	None
OT_L2TOPOLOGY	NativeEMSName

Object Type	Attributes
OT_MLVLAN	None This event notifies the NMS that the VLAN was updated.
OT_ROLL	None This event notifies the NMS that the roll was updated.
OT_VCAT	None This event notifies the NMS that the VCAT changed.
OT_PHYSICAL_TERMINATION_POINT	Operation applied to PPM port rate.

5.3.3.1 Alarm Acknowledge/Unacknowledge—AttributeValueChange

The alarm acknowledge/unacknowledge API generates an attribute value change that reports the fields changed for a given alarm. The following table displays the attribute list reported.

Attribute Name	Type	Description
Ack	Yes/No	—
Ack Time Stamp (GMT)	—	yyyymmddhhmmss.0Z
Ack Username	String	OSS user

Example

```
Fixed Header            Domain              = tmf_mtnm
                        Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header         Event Reliability   = 0
                        Priority            = 0
                        Timeout             = 864000000000


Filterable Data      [0] notificationId      = AVC-31
                     [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=15454-ANSI-68-7; [2]
AID=notificationId=10
                     [2] objectType          = OT_AID
                     [3] objectTypeQualifier =
                     [4] emsTime             = 20111201143219.0Z
                     [5] neTime              = 20030714115221.0Z
                     [6] edgePoint           = false
                     [7] attributeList       = [0]Ack=Yes [1]Ack Time Stamp
(GMT)=20111201143219.0Z [2]Ack Username=SuperUser
                      [8] isPropagated        = false
Remainder of body:
```

5.3.3.2 Alarm Note Change—AttributeValueChange

The setAdditional attribute for the alarm note API generates an attribute value change that reports the fields changed for a given alarm. The following table lists the attributes reported.

Attribute Name	Type	Description

Note	• Append • Delete • Replace	The type of operation.
Value	String	The note.

Example:

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data      [0] notificationId       = AVC-26
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] ManagedElement=15454-ANSI-68-7; [2]
AID=notificationId=3
                     [2] objectType           = OT_AID
                     [3] objectTypeQualifier =
                     [4] emsTime              = NA
                     [5] neTime               = 20030714115207.0Z
                     [6] edgePoint            = false
                     [7] attributeList        = [0]Note=append OSS
(User Name: CorbaUser   Timestamp: 12/13/11 11:27:24 AM GMT)

                      [8] isPropagated         = false
Remainder of body:
```

5.3.3.3 SNCs—AttributeValueChange

Updates or modifications to the subnetwork connections (**SNC**) trigger an Attribute Value Change (AVC) event. When the circuit name is modified, the notification reports two attributes to notify the old and the new circuit names.

The following table lists the attributes that are reported only when the circuit (SNC) name is modified:

Attribute Name	Type	Description
New SubnetworkConnection Name	String	New circuit name.
Previous SubnetworkConnection Name	String	Previous circuit name.

Example:

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
```

```
                        Timeout            = 30000


Filterable Data      [0] notificationId      = AVC-70
                     [1] objectName          = [0] EMS=Cisco
Systems/PrimeOptical; [1] SubnetworkConnection=buzzo
                     [2] objectType          = OT_SUBNETWORK_CONNECTION
                     [3] objectTypeQualifier =
                     [4] emsTime             = 19700116235926.0Z
                     [5] neTime              =
                     [6] edgePoint           = false
                     [7] attributeList       = [0]New SubnetworkConnection
Name=buzzo [1]Previous SubnetworkConnection Name=buzzo1
                      [8] isPropagated        = false
Remainder of body:
```

5.3.3.4 OCHCC GMPLS SNCs Operations—AttributeValueChange

The validation of a manual or automatic OCHCC GMPLS SNC triggers an Attribute Value Change (AVC) event. This event is represented by the following parameters:

- gmplsAcptThreshold
- gmplsOptValid

Some operations in GMPLS circuits are asynchronous. The response to the command indicates only that the command has been received and not that the command has been executed. An AVC event is sent as a notification that command execution is complete.

Operations for which AVCs are sent are:

- CircuitAdded
- Reroute
- Promote
- Validate
- ManualRevert
- UpgradeRestored
- UpgradeRestoreInfo

The type of AVC is indicated in the attribute list present in the notification data.

The following table lists the attributes that are reported.

Attribute Name	Type	Description
gmplsAcptThreshold	String	This threshold acceptance value is checked against the actual optical value. It can assume one of the following values: • Green • Yellow • Orange • Red • NotApplicable

gmplsOptValid	String	Current optical valid value. It can assume one of the following values: • Green • Yellow • Orange • Red • NotApplicable
EventCause	String	The event that has caused the AVC. Valid values are: • CircuitAdded • Reroute • Promote • Validate • ManualRevert • UpgradeRestored • UpgradeRestoreInfo
EventMessage	String	Message sent at success or failure of the operation.

- The number of attributes listed is related to the type of operation executed.
- AVC events for changed parameter values (for example, gmplsAcptThreshold) are not supported for GateWay/CORBA.
- AVC events are not sent for deletion, as object deletion notifications are sent after deletion.
- An AVC event is sent for circuit creation to indicate that the circuit being created is an MPLS circuit.

Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 30000


Filterable Data      [0] notificationId       = AVC-3
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] SubnetworkConnection=WSON
                     [2] objectType           = OT_SUBNETWORK_CONNECTION
                     [3] objectTypeQualifier =
                     [4] emsTime              = 20121017140705.0Z
                     [5] neTime               =
                     [6] edgePoint            = false
                     [7] attributeList        = [0]gmplsAcptThreshold=Orange
[1]gmplsOptValid=Green [2]EventCause=Validate [3]EventMessage=Command forward
to the network. Operation done
                     [8] isPropagated         = false
Remainder of body:
```

5.3.3.5 OCHCC GMPLS SNCs Creation Failure—AttributeValueChange

Creating an OCHCC GMPLS SNC requires a background request validation. The API performs a first parsing level of passed information. A successful response does not mean that the SNC has been created. The SNC is created only after a successful network validation.

Notification of OCHCC GMPLS SNC creation occurs through an attribute value change event. If the API executes successfully but the network validation fails for any reason, a new notification of an attribute value change event is sent. The notification contains the SNC name and the reason for the failure.

For OCHCC, OCHNC, and OCH trail circuits, you can provide additional GMPLS information about the associated SNCs.

The following table lists the reported attributes.

Attribute Name	Type	Description
Creation Result	String	Creation result. For example, circuit creation failed.
Error Reason	String	Describes the reason for the failure.

Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 864000000000

Filterable Data      [0] notificationId       = AVC-21
                     [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] SubnetworkConnection=ochnc-unprot-gmpls
                     [2] objectType           = OT_SUBNETWORK_CONNECTION
                     [3] objectTypeQualifier =
                     [4] emsTime              = 20120208223853.0Z
                     [5] neTime               =
                     [6] edgePoint            = false
                     [7] attributeList        = [0]Creation Result=Circuit
Creation Failed [1]Error Reason=Operation Failed node: 10.58.65.104 No Trunk
port matching the request
                      [8] isPropagated         = false
Remainder of body:
```

5.3.3.6 SubnetworkConnection Name Change - AttributeValueChange

When the subnetwork connection name is modified from the CORBA client, a new AtrributeValueChange event is generated.

The following table lists the reported attributes.

Attribute Name	Type	Description
Previous SubnetworkConnection Name	String	Previous name of the subnetwork connection
New SubnetworkConnection Name	String	New name of the subnetwork connection

Example

```
Fixed Header              Domain              = tmf_mtnm
                          Type                = NT_ATTRIBUTE_VALUE_CHANGE
Variable Header           Event Reliability   = 0
                          Priority            = 0
                          Timeout             = 30000
Filterable Data       [0] notificationId       = AVC-1
                      [1] objectName           = [0] EMS=Cisco
Systems/PrimeOptical; [1] SubnetworkConnection=new24
                      [2] objectType           = OT_SUBNETWORK_CONNECTION
                      [3] objectTypeQualifier =
                      [4] emsTime              = 20121120094200.0Z
                      [5] neTime               =
                      [6] edgePoint            = false
                      [7] attributeList = [0]New value=new24 [1]Old value=new23
                      [8] isPropagated         = false
Remainder of body:
```

5.3.4 State Change Event

The following table lists the state change event types and attributes.

Table 5-9: State Change Event

Attribute Name	Type	Supported by Prime Optical?
notificationId	string	Yes. The uniqueness and sequence of the notification ID 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.
attributeList	notifications::NVList_T	Yes.

State change event notifications are sent to the NMS for the objects and attributes in the following table.

Table 5-10: State Change Event Notifications

Object Type	Attributes
OT_MANAGED_ELEMENT	communicationState—Prime Optical supports the following values: • CS_AVAILABLE • CS_UNAVAILABLE emsInSyncState—Prime Optical supports the following values: • True • False

Object Type	Attributes
OT_EQUIPMENT_HOLDER	holderState—State change. Prime Optical supports the following values: • EMPTY • INSTALLED_AND_EXPECTED • EXPECTED_AND_NOT_INSTALLED • MISMATCH_OF_INSTALLED_AND_EXPECTED
OT_SUBNETWORK_CONNECTION	sncState—Prime Optical supports the following values: • SNCS_DELETING • SNCS_PARTIAL • SNCS_ACTIVE
OT_L2TOPOLOGY	L2TopoState—Prime Optical supports the following values: • L2TS_COMPLETE • L2TS_INCOMPLETE topoInSyncState—Prime Optical supports the following values: • true • false
OT_MLVLAN	vlanState—Prime Optical supports the following values: • VLS_COMPLETE • VLS_INCOMPLETE
OT_ROLL	rollState—Prime Optical supports the following values: • ROLL_PENDING • ROLL_COMPLETED • ROLL_CANCELLED • ROLL_INCOMPLETE
OT_VCAT	vcatState—Prime Optical supports the following values: • VCATS_DELETING • VCATS_PARTIAL • VCAT_ACTIVE

5.3.5 Protection Switch

The following table lists the protection switch types and attributes.

Table 5-11: Protection Switch

Attribute Name	Type	Supported by Prime Optical?
notificationId	String	Yes. The uniqueness and the sequence of the notification ID are not guaranteed.
emsTime	globaldefs::Time_T	Yes.
neTime	globaldefs::Time_T	No.
ProtectionType	protection::ProtectionType_T	Yes.
switchReason	protection::switchReason_T	Yes. Only SR_AUTOMATIC_SWITCH, SR_MANUAL_SWITCH e SR_RESTORED are handled because NO edvent correlation is supported by NBI

Attribute Name	Type	Supported by Prime Optical?
layerRate	transmissionParameters::LayerRate_T	Yes.
groupName	globaldefs::NamingAttributes_T	No.
protectedTP	globaldefs::NamingAttributes_T	Yes.
switchAwayFromTP	globaldefs::NamingAttributes_T	Yes
switchToTP	globaldefs::NamingAttributes_T	Yes.
affectedTP This is not defined by the TMF.	globaldefs::NamingAttributes_T	No.
Description This is not defined by the TMF.	String	Yes. The value of this field is the native NE event value.

5.3.6 Threshold Crossing Alert

The following table lists the TCA types and attributes.

Table 5-12: Threshold Crossing Alert

Attribute Name	Type	Supported by Prime Optical?
notificationId	String	Yes. The uniqueness and the sequence of the notification ID are not guaranteed.
objectName	globaldefs::NamingAttributes_T	Yes.
nativeEMSName	String	Yes.
objectType	notifications::ObjectType_T	Yes.
emsTime	globaldefs::Time_T	Yes.
neTime	globaldefs::Time_T	Yes.
isClearable	boolean	Yes. Always set to false.
perceivedSeverity	notifications::PerceivedSeverity_T	Yes.Always set to PerceivedSeverity_T.PS_INDETERMINATE.
layerRate	transmissionParameters::LayerRate_T	Yes.
granularity	Granularity_T	Yes.
pmParameterName	PMParameterName_T	Yes. If the NE does not report the location, this field is empty.
pmLocation	PMLocation_T	Yes. If the NE does not report the location, this field is empty.
thresholdType	PMThresholdType_T	No.
value	Float	Yes.
Unit	String	No.

5.3.7 Alarm

The following table lists the most common alarm messages.

If there is an alarm on an STS/VC4 CTP utilizing all ports of a DS1/E1 card, the alarm is reported with the containing PTP as port 127. If an STS/VC4 SNC exists with one of the drop CTPs of the SNC using

all ports on a DS1/E1 card, and if there is an alarm on the STS CTP, Prime Optical reports the port number of the containing PTP as 127.

✐ For ONS 15454 and ONS NCS2K SONET and SDH, the virtual port details are displayed in additionalText for AR_XPE sync loss alarms.

Table 5-13: Alarm

Attribute Name	Type	Supported by Prime Optical?
notificationId	String	Yes. The uniqueness and sequence of the notification ID are not guaranteed.
objectName	globaldefs::NamingAttributes_T	Yes.
nativeEMSName	String	Yes.
nativeProbableCause	String	Yes. ✐ In case of external condition alarms, the string value of the condition type is appended to this field.
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	Yes.
probableCause	String	Yes.
probableCauseQualifier	String	Yes. Prime Optical provides the same value as in the nativeProbableCause field.
perceivedSeverity	notifications::PerceivedSeverity_T	Yes.
serviceAffecting	notifications::ServiceAffecting_T	Yes.
affectedTPList	globaldefs::NamingAttributesList_T	Yes.
additionalText	String	Yes.
additionalInfo	globaldefs::NVSList_T	Yes. Prime Optical provides subnetwork name when available. If the alarmed object is OC-n PTP, Prime Optical also includes SDH_SONET_SS_BITS in this list. Prime Optical provides the IP address for ME-related alarms. Prime Optical provides the ME name when the object type is OT_EMS and the alarm is ME-related. Prime Optical provides Card Name when available. Prime Optical provides Module Type when available. Prime Optical provides Affected Object when available.
isPropagated	Boolean	Yes, but this field is always set to False.
aresFDN	String	Yes.

5.3.8 Heartbeat Event

Prime Optical 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 the Prime Optical Control Panel. If the interval value is set to 0, the heartbeat event is disabled. By monitoring this event, the northbound interface 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 the following table.

Table 5-14: Heartbeat Event

Attribute Name	Type	Supported by Prime Optical?
notificationId	string	Yes. The uniqueness and sequence of the notification ID are not guaranteed.

5.3.9 Backup Status Event

The backup status event contains details about changes to a managed element's backup status.

Table 5-15: Backup Status Event

Attribute Name	Type	Supported by Prime Optical?
notificationId	String	Yes. The uniqueness and sequence of the notification ID are not guaranteed.
emsTime	globaldefs::Time_T	Yes. The time when the EMS reported the event.
neTime	globaldefs::Time_T	Yes. The time provided by the NE. If the NE does not report time, this field is optional or reports an empty string.
backupStatus	softwareManager::BackupStatus_T	Yes. The backup status of the managed element.
meName	globaldefs::NamingAttributes_T	Yes. The name of the managed element from which the backup was taken.

5.4 Unsupported Events

The Events with INDETERMINATE severity that are reported in the Alarm Log Browser of the Cisco Prime Optical are ***not*** pushed to the external OSS clients through CORBA Gateway.

Note: *The severity of the Performance Monitoring Threshold Crossing Alerts(TCA's) are always set to INDETERMINATE and they are pushed to the external OSS Client.*Please refer to section 5.4.6

5.5 Probable Cause Mapping

5.5.1 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
```

5.5.2 GateWay/CORBA Mapping—EMS

The following table lists the GateWay/CORBA mapping for the EMS.

Table 5-16: GateWay/CORBA Mapping for the EMS

TMF Mapping	NE Native Probable Cause
EMS	Loss of communication
EMS	Memory automatic or manual backup failure
SECURITY_VIOLATION	Maximum login attempts exceeded
EMS	Partition 0...6 free disk space low
EMS	Alarm resync unsuccessful
EMS	Server monitor threshold crossed

5.5.3 GateWay/CORBA Mapping—ONS 15216

The following table lists the alarms for the ONS 15216 OADM R2.2.

Table 5-17: GateWay/CORBA Alarms for the ONS 15216 OADM R2.2

TMF Mapping	NE Native Probable Cause
LOS	Add channel signal loss
LOS	Drop channel signal loss
EQPT	Add and drop channels are out of tolerance
EQPT	Add channel signal is too weak and VOA cannot adjust it to match the drop signal
EQPT	Power unit A failed
EQPT	Power unit B failed

5.5.3.1 ONS 15216 EDFA2

The following table lists the alarms for the ONS 15216 EDFA2 R2.3 and 2.4.

Table 5-18: GateWay/CORBA Alarms for the ONS 15216 EDFA2 R2.3, R2.4

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Unknown alarm
EQPT	Excessive pump 1 current
EQPT	Excessive pump 2 current
EQPT	Excessive pump 1 laser temperature
EQPT	Excessive pump 2 laser temperature
EQPT	Loss of output power
EQPT	Loss of input power
EQPT	Gain out of range
EQPT	Case temperature out of range
EQPT	Power unit A failed
EQPT	Power unit B failed
EQPT	Database backup
EQPT	Database restore
EQPT	Cutover reset

TMF Mapping	NE Native Probable Cause
EQPT	Software load
EQPT	Software reset

5.5.3.2 ONS 15216 EDFA3

The following table lists the alarms for the ONS 15216 EDFA3 R1.0.

Table 5-19: GateWay/CORBA Alarms for the ONS 15216 EDFA3 R1.1

TMF Mapping	NE Native Probable Cause	Prime Optical Internal Index (Not Reported in the Notification)
UNIDENTIFIED	Unknown alarm	8701
EQPT	Gain degrade high	8702
EQPT	Gain degrade low	8703
EQPT	Laser 1 bias degrade	8704
EQPT	Laser 2 bias degrade	8705
EQPT	Laser 1 bias fail	8706
EQPT	Laser 2 bias fail	8707
EQPT	Power degrade high LINE1TX port	8708
EQPT	Power degrade low LINE1TX port	8709
EQPT	Power fail low LINE1RX port	8710
EQPT	Power fail low LINE1TX port	8711
EQPT	Power fail low LINE2RX port	8712
EQPT	VOA degrade high	8713
EQPT	VOA degrade low	8714
EQPT	VOA fail high	8715
EQPT	Case temperature out of range	8716
EQPT	Fiber temperature out of range	8717
EQPT	Excessive pump 1 temperature	8718
EQPT	Excessive pump 2 temperature	8719
EQPT	Backup/restore in progress	8720
EQPT	Data integrity fault	8721
EQPT	Equipment failure	8722
EQPT	Exceeding memory capacity	8723
EQPT	Exceeding flash file system capacity	8724
EQPT	Module communication failure	8725
EQPT	Power bus A alarm	8726
EQPT	Power bus B alarm	8727

5.5.4 GateWay/CORBA Mapping—ONS 15305 and ONS 15305 CTC

The following table lists the GateWay/CORBA mapping for the ONS 15305 and ONS 15305 CTC.

Table 5-20: GateWay/CORBA Mapping for the ONS 15305, ONS 15305 CTC

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	ONS 15305 condition unknown
AIS	Alarm indication signal
AIS	Alarm indication signal network side
LOF	Alarm indication signal customer side
LOF	Loss of frame alignment
LOF	Loss of frame alignment network side
LOF	Loss of frame alignment customer side

TMF Mapping	NE Native Probable Cause
LOP	Loss of pointer
LOS	Loss of signal
BER_SD	Degraded signal defect
LOM	Loss of multiframe alignment
RAI	Remote defect indication
TIM	Trace identifier mismatch
PLM	Payload mismatch
UNEQ	Unequipped
BER_SF	Excessive error defect
UNIDENTIFIED	Communication subsystem failure
SSF	Server signal failure
UNIDENTIFIED	Alarm condition detected on an alarm input port
EQPT	Device main unit failure
EQPT	High temperature alarm
EQPT	Fan failure
EQPT	Power input failure
EQPT	Power output failure
EQPT	Module mismatch
EQPT	Module removed
EQPT	Module failure
UNIDENTIFIED	MSP signaling problem
UNIDENTIFIED	Power module failure
UNIDENTIFIED	LAN port is not working
UNIDENTIFIED	WAN port is not working
UNIDENTIFIED	Too large delay on WAN channel
UNIDENTIFIED	Sequence number fail on WAN channel
UNIDENTIFIED	No synchronization source available
UNIDENTIFIED	Defecting hardware impacting internal T0 clock
UNIDENTIFIED	No T4 synchronization source available with QL >= QL min
UNIDENTIFIED	Inventory failure
UNIDENTIFIED	Diagnostic failure
UNIDENTIFIED	DXC inlet failure
UNIDENTIFIED	DXC inlet bit error
UNIDENTIFIED	Card isolated
UNIDENTIFIED	Card anomaly
UNIDENTIFIED	Hot swap failure
UNIDENTIFIED	Power input failure
UNIDENTIFIED	Transmit degrade on laser
UNIDENTIFIED	Transmit fail on laser
UNIDENTIFIED	Port activated without mapping to an available VC-12
UNIDENTIFIED	Differential VC12 delay for the WAN port is greater than +/- 2 ms
UNIDENTIFIED	Rx buffer overflow in LAN or link interface
UNIDENTIFIED	Interport queue overflow in LAN or link interface
UNIDENTIFIED	Reset of the router/bridge required
UNIDENTIFIED	Device aborted FTP session
UNIDENTIFIED	Automatic switchover to backup link, main link fault
UNIDENTIFIED	Overflow layer 2 forward table
UNIDENTIFIED	Connection failed: frame relay switch and WAN gate

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Connection established: frame relay switch and WAN gate
UNIDENTIFIED	Error occurred during initialization
UNIDENTIFIED	Overflow in the client table
UNIDENTIFIED	Server not responding to dispatcher polling
UNIDENTIFIED	SNMP SET request was rejected
UNIDENTIFIED	Ping sequence completed
UNIDENTIFIED	Backup taken over for main server or server up after failure
UNIDENTIFIED	DHCP failed to allocate IP address to requesting host
UNIDENTIFIED	Overflow IGMP table
UNIDENTIFIED	Overflow PIM table
UNIDENTIFIED	Overflow condition in routing table
UNIDENTIFIED	Open gate IPX RIP table overflow
UNIDENTIFIED	Open gate IPX SAP table overflow
UNIDENTIFIED	FACS state NE with operation blockAndReport
UNIDENTIFIED	Zero hop routing connections table overflow
UNIDENTIFIED	No available IP virtual address
UNIDENTIFIED	Virtual IP address appeared as a source IP
UNIDENTIFIED	Source IP address sent an ARP specifying a virtual IP
UNIDENTIFIED	PPP link got an unrecognized secret
UNIDENTIFIED	FR DLCI status change
UNIDENTIFIED	CHAP failed communication
UNIDENTIFIED	IP SFFT overflow
UNIDENTIFIED	IP NFFT overflow
UNIDENTIFIED	IPX SFFT overflow
UNIDENTIFIED	IPX NFFT overflow
UNIDENTIFIED	IPM FFT overflow
UNIDENTIFIED	PAP failed communication
UNIDENTIFIED	Automatic switchover to backup link, main link fault
UNIDENTIFIED	Informational event
UNIDENTIFIED	T0 sync switch
UNIDENTIFIED	T0QI failed
UNIDENTIFIED	T0QIDnu
UNIDENTIFIED	Switch to protection
UNIDENTIFIED	Switch to working
UNIDENTIFIED	MSP command timeout
UNIDENTIFIED	MSP command overruled
UNIDENTIFIED	Loop closed
UNIDENTIFIED	Loop opened
UNIDENTIFIED	Link up
UNIDENTIFIED	Link down
UNIDENTIFIED	The communication link has come up
UNIDENTIFIED	Failure in communication link
UNIDENTIFIED	Addressee of a protocol message not properly authenticated
UNIDENTIFIED	Reinitializing; configuration or the protocol entity implementation may be altered
UNIDENTIFIED	Reinitializing; neither configuration nor protocol entity implementation altered
UNIDENTIFIED	RMON alarm has crossed the rising threshold
UNIDENTIFIED	RMON alarm has crossed the falling threshold
UNIDENTIFIED	Loss of an adjacency with a PIM neighbor

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Device finished TFTP transaction
UNIDENTIFIED	Device initiated TFTP transaction
UNIDENTIFIED	Auto configuration completed successfully
UNIDENTIFIED	VLAN port dynamically added
UNIDENTIFIED	VLAN port dynamically changed
UNIDENTIFIED	Physical description device changed
UNIDENTIFIED	Port transition from learning to forwarding
UNIDENTIFIED	Port transition from forwarding to learning
UNIDENTIFIED	Packet drop due to the QoS policy
UNIDENTIFIED	Packet forwarded based on the QoS policy
UNIDENTIFIED	Protection link activated
UNIDENTIFIED	Working link activated
UNIDENTIFIED	MSP command timeout
UNIDENTIFIED	MSP command overruled
UNIDENTIFIED	Loss of external synchronization
UNIDENTIFIED	ONS 15305 condition unknown

5.5.5 GateWay/CORBA Mapping—ONS 15310 CA, ONS 15310 MA, ONS 15310 MA SDH, ONS 15327 SONET, ONS 15454 SONET, ONS 15454 SDH, ONS NCS2k SONET, ONS NCS2k SDH, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH

The following table lists the GateWay/CORBA mapping for ONS 15310 CA, ONS 15310 MA, ONS 15310 MA SDH, ONS 15327 SONET, ONS 15454 SONET, ONS 15454 SDH, , ONS NCS2K SONET, ONS NCS2K SDH, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH NEs.

Table 5-21: GateWay/CORBA Mapping for the ONS 15310 CA, ONS 15310 MA, ONS 15310 MA SDH, ONS 15327 SONET, 0NS 15454 SONET, ONS 15454 SDH, 0NS NCS2K SONET, ONS NCS2K SDH, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Normal condition
ENV	Failure detected external to the NE
ENV	External error
UNIDENTIFIED	Excessive switching
UNIDENTIFIED	Incoming failure condition
AIS	Alarm indication signal
AIS	Alarm indication signal—line
AIS	Alarm indication signal—path
FOP_APS	APS channel failure
FOP_APS	Byte failure
FOP_APS	Protection switching channel match failure
FOP_APS	Automatic protection switch mode mismatch
FOP_APS	Far-end protection line failure
UNIDENTIFIED	Bipolar violation
UNIDENTIFIED	Carrier loss on the LAN
UNIDENTIFIED	STS concatenation error
UNIDENTIFIED	Excess collisions on the LAN
UNIDENTIFIED	Facility failure
UNIDENTIFIED	Far-end block error
LOF	Loss of frame

TMF Mapping	NE Native Probable Cause
LOP	Loss of pointer
LOP	Loss of pointer—path
LOS	Loss of signal
UNIDENTIFIED	Out of frame
UNIDENTIFIED	Path selector failure
RAI	Remote alarm indication
RAI	Remote failure indication
RAI	Remote failure indication—line
RAI	Remote failure indication—path
BER_SD	Signal degrade
UNIDENTIFIED	Severely errored frame
UNIDENTIFIED	Invalid alarm
BER_SF	Signal failure
UNIDENTIFIED	Signal label mismatch failures
UNIDENTIFIED	Payload defect indication
UNIDENTIFIED	Payload defect indication—path
PLM	Payload label mismatch—path
UNEQ	Unequipped—path
TIMING_SYNCH	Loss of synchronization
TIMING_SYNCH	Out of synchronization
TIMING_SYNCH	Primary synchronization reference failure
TIMING_SYNCH	Secondary synchronization reference failure
TIMING_SYNCH	Third synchronization reference failure
TIMING_SYNCH	Fourth synchronization reference failure
TIMING_SYNCH	Fifth synchronization reference failure
TIMING_SYNCH	Sixth synchronization reference failure
UNIDENTIFIED	Outgoing failure condition
RAI	Remote defect indication—line
RAI	Remote defect indication—path
TIMING_SYNCH	Free running synchronization mode
TIMING_SYNCH	Holdover synchronization mode
EQPT	Internal fault
UNIDENTIFIED	Internal error
UNIDENTIFIED	Internal message error
EQPT	Mismatch of equipment and attributes
UNIDENTIFIED	Watchdog timer timeout
ENV	Software fault or failure
ENV	Software fault—data integrity fault
UNIDENTIFIED	Program failure
EQPT	Control equipment failure
EQPT	Control processor failure
EQPT	Working memory failure
EQPT	Interconnection equipment failure
EQPT	Time slot interchange equipment failure
EQPT	Equipment failure
EQPT	High temperature
EQPT	Facility termination equipment failure
EQPT	Automatic laser shutdown

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Failure to release from protection
EQPT	Receiver failure
EQPT	Transmit failure
UNIDENTIFIED	Failure to switch to protection
EQPT	Equipment unit plug-in
ENV	Internal power failure
ENV	Fuse alarm
TIMING_SYNCH	Synchronization unit failure
EQPT	Synchronization switching equipment failure
EQPT	Equipment unit unplugged
EQPT	Manually caused abnormal condition
UNIDENTIFIED	Manual alarm cutoff
UNIDENTIFIED	Procedural error
EQPT	Improper removal
UNIDENTIFIED	Protection unit not available
UNIDENTIFIED	Protection switch
UNIDENTIFIED	Recovery or service protection action has been initiated
UNIDENTIFIED	Automatic system reset
UNIDENTIFIED	Cold restart
UNIDENTIFIED	Forced switch back to working
UNIDENTIFIED	Forced switch to protection
UNIDENTIFIED	Initialization initiated
UNIDENTIFIED	Lockout of protection
UNIDENTIFIED	Lockout of working
UNIDENTIFIED	Manual system reset
UNIDENTIFIED	Manual switch to internal clock
UNIDENTIFIED	Manual switch to primary reference
UNIDENTIFIED	Manual switch to secondary reference
UNIDENTIFIED	Manual switch to third reference
UNIDENTIFIED	Manual switch to fourth reference
UNIDENTIFIED	Manual switch to fifth reference
UNIDENTIFIED	Manual switch to sixth reference
UNIDENTIFIED	Manual switch back to working
UNIDENTIFIED	Manual switch to protection
UNIDENTIFIED	Power fail restart
EQPT	Software download in progress
TIMING_SYNCH	Switch to internal clock
TIMING_SYNCH	Switch to primary reference
TIMING_SYNCH	Switch to secondary reference
TIMING_SYNCH	Switch to third reference
TIMING_SYNCH	Switch to fourth reference
TIMING_SYNCH	Switch to fifth reference
TIMING_SYNCH	Switch to sixth reference
UNIDENTIFIED	Switched back to working
UNIDENTIFIED	Switched to protection
UNIDENTIFIED	Warm restart
UNIDENTIFIED	Ring is in wait-to-restore state

TMF Mapping	NE Native Probable Cause
EQPT	Primary nonvolatile backup memory failure
EQPT	Secondary nonvolatile backup memory failure
EQPT	Control bus failure
EQPT	Control communications equipment failure
UNIDENTIFIED	Loopback
UNIDENTIFIED	Loopback, network
UNIDENTIFIED	Loopback, terminal
ENV	Fan failure
DCC_FAILURE	SDCC termination failure
UNIDENTIFIED	Loopback facility
EQPT	Payload bus failure to I/O slot 1 to 17 (XCON slot 8)
EQPT	Control bus 1 to 2 failure
EQPT	Invalid MAC address
EQPT	Board failure
EQPT	Diagnostics failure
EQPT	Medium access control failure
UNIDENTIFIED	Duplicate node ID
UNIDENTIFIED	Failure to switch to protection—ring
UNIDENTIFIED	Failure to switch to protection—span
UNIDENTIFIED	Manual switch back to working—ring
UNIDENTIFIED	Manual switch back to working—span
UNIDENTIFIED	Manual switch to protection—ring
UNIDENTIFIED	Manual switch to protection—span
UNIDENTIFIED	Forced switch back to working—ring
UNIDENTIFIED	Forced switch back to working—span
UNIDENTIFIED	Forced switch to protection—ring
UNIDENTIFIED	Forced switch to protection—span
UNIDENTIFIED	Lockout of protection—ring
UNIDENTIFIED	Lockout of protection—span
UNIDENTIFIED	Lockout of working—ring
UNIDENTIFIED	Lockout of working—span
UNIDENTIFIED	Ring is squelching traffic
FOP_APS	Inconsistent APS code
FOP_APS	Node ID mismatch
FOP_APS	Default K byte
FOP_APS	Connection loss
EQPT	TCC [A...B] to shelf slot 1...17 [DROP 1...8 | TRUNK 1...2 | TCC A...B | XCON A...B] communication failure
TIMING_SYNCH	Fast start synchronization mode
FOP_APS	Improper APS code
UNIDENTIFIED	BLSR out of sync
AIS	Alarm indication signal—VT
RAI	Remote failure indication—VT
PLM	Signal label mismatch failure—payload label mismatch—VT
UNEQ	Signal label mismatch failure—unequipped—VT

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Peer state mismatch
EQPT	Facility termination equipment—receiver missing
EQPT	Facility termination equipment—transmitter missing
LOP	Loss of pointer—VT
LINK DOWN	Embedded operations channel failure—link down
TIMING_SYNCH	Failed to receive synchronization status message
TIMING_SYNCH	Synchronization status messages are disabled on this interface
TIMING_SYNCH	Primary reference source—stratum 1 traceable
TIMING_SYNCH	Synchronized—traceability unknown
TIMING_SYNCH	Stratum 2 traceable
TIMING_SYNCH	Transit node clock traceable
TIMING_SYNCH	Stratum 3E traceable
TIMING_SYNCH	Stratum 3 traceable
TIMING_SYNCH	SONET minimum clock traceable
TIMING_SYNCH	Stratum 4 traceable
TIMING_SYNCH	Do not use for synchronization
TIMING_SYNCH	Reserved for network synchronization use
UNIDENTIFIED	Failure to switch to protection—path
UNIDENTIFIED	Manual switch of working facility—equipment to protection—path
UNIDENTIFIED	Working facility—equipment forced to switch to protection—path
UNIDENTIFIED	Lockout of protection—path
UNIDENTIFIED	Automatic UPSR switch caused by [AIS | LOP | UNEQ | PDI | SFBER | SDBER]
UNIDENTIFIED	DS3 idle condition
UNIDENTIFIED	DS1 loopback due to FEAC command
UNIDENTIFIED	DS1 loopback command sent to far end
UNIDENTIFIED	DS3 loopback due to FEAC command
UNIDENTIFIED	DS3 loopback command sent to far end
UNIDENTIFIED	DS2 loopback due to far-end command
UNIDENTIFIED	DS2 loopback command sent to far end
ENV	Far-end AIS
LOS	Far-end multiple DS1 LOS detected on DS3
ENV	Far-end DS1 equipment failure—non-service affecting
ENV	Far-end DS1 equipment failure—service affecting
LOS	Far-end single DS1 LOS
ENV	Far-end DS3 equipment failure—non-service affecting
ENV	Far-end DS3 equipment failure—service affecting
ENV	Far-end common equipment failure—non-service affecting
UNIDENTIFIED	Far end idle
LOS	Far end LOS
LOF	Far end LOF

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Performance monitor threshold crossing alert
UNIDENTIFIED	Ethernet bridge is new root of spanning tree
UNIDENTIFIED	Ethernet bridge topology change
UNIDENTIFIED	BLSR tables resynchronized
ENV	System reboot
UNIDENTIFIED	Manual switch request on facility—equipment
UNIDENTIFIED	Forced switch request on facility or equipment
UNIDENTIFIED	Lockout switch request on facility—equipment
UNIDENTIFIED	RMON histories and alarms reset reboot
UNIDENTIFIED	RMON alarm
LOS	Video interface card loss of video IF signal
LOS	VIC loss of video baseband signal
LOS	VIC loss of audio FM signal
LOS	VIC loss of audio baseband channel [1...4] signal
TIM	STS path trace identifier mismatch
ENV	NE power failure at connector A
ENV	NE power failure at connector B
EQPT	Free memory on card very low
EQPT	Free memory on card near zero
UNIDENTIFIED	Exercise request on ring
UNIDENTIFIED	Exercise request on span
UNIDENTIFIED	Squelching path
UNIDENTIFIED	Extra traffic preempted
UNIDENTIFIED	Far-end lockout of working—ring
UNIDENTIFIED	Far-end lockout of working—span
UNIDENTIFIED	Far-end lockout of protection—ring
UNIDENTIFIED	Far-end lockout of protection—all spans
UNIDENTIFIED	Far-end working facility forced to switch to protection—ring
UNIDENTIFIED	Far-end working facility forced to switch to protection—span
UNIDENTIFIED	Far-end manual switch of working facility to protection—ring
UNIDENTIFIED	Far-end manual switch of working facility to protection—span
UNIDENTIFIED	Far-end exercising ring
UNIDENTIFIED	Far-end exercising span
ENV	Far-end BER threshold passed for signal failure—ring
ENV	Far-end BER threshold passed for signal failure—span
ENV	Far-end BER threshold passed for signal degrade—ring
ENV	Far-end BER threshold passed for signal degrade—span
ENV	APS channel—far-end protection line signal degrade
UNIDENTIFIED	Ring switch is active on the east side

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Ring switch is active on the west side
UNIDENTIFIED	Span switch is active on the east side
UNIDENTIFIED	Span switch is active on the west side
UNIDENTIFIED	Unidirectional full pass-through is active
UNIDENTIFIED	Bidirectional full pass-through is active
UNIDENTIFIED	K bytes pass-through is active
UNIDENTIFIED	Ring is segmented
UNIDENTIFIED	Ring topology is under construction
UNIDENTIFIED	Lockout of protection—all spans
UNIDENTIFIED	Far-end of fiber is provisioned with different ring ID
UNIDENTIFIED	Both ends of fiber provisioned as east or both as west
SECURITY_VIOLATION	Security—invalid login—see audit trail
UNIDENTIFIED	Autonomous messages inhibited
UNIDENTIFIED	Traffic storm on LAN; LAN temporarily disabled
UNIDENTIFIED	REPT^DBCHG messages inhibited
SECURITY_VIOLATION	Security—user ID has expired
EQPT	Partial fan failure
UNIDENTIFIED	Force switch request on ring
UNIDENTIFIED	Force switch request on span
UNIDENTIFIED	Lockout switch request on ring
UNIDENTIFIED	Lockout switch request on span
UNIDENTIFIED	Manual switch request on ring
UNIDENTIFIED	Manual switch request on span
EQPT	Peer-to-peer slot communication failure
EQPT	Peer-to-peer slot communication failure
EQPT	TCC A to shelf slot communication failure
EQPT	TCC B to shelf slot communication failure
EQPT	Interconnection equipment failure—working XC payload bus
EQPT	Interconnection equipment failure—protect XC payload bus
EQPT	Inhibit switch to protect request on equipment
EQPT	Inhibit switch to working request on equipment
BER_SD	BER threshold exceeded for signal degrade—line
BER_SD	BER threshold exceeded for signal degrade—path
BER_SF	BER threshold exceeded for signal failure—line
BER_SF	BER threshold exceeded for signal failure—path
UNIDENTIFIED	Exercising ring successfully
UNIDENTIFIED	Exercising span successfully
UNIDENTIFIED	Span is in wait-to-restore state
EQPT	Peer card not responding
UNIDENTIFIED	Exercise request on ring failed
UNIDENTIFIED	Exercise request on span failed
UNIDENTIFIED	Far-end lockout of protection—span
EQPT	Manufacturing data memory (EEPROM) failure
EQPT	Replaceable equipment/unit is missing

TMF Mapping	NE Native Probable Cause
ENV	Software download failure
TIM	TIM section—trace identifier mismatch failure
AIS	AIS—multiplex section—alarm indication signal
RAI	RFI—multiplex section—remote failure/alarm indication
TIM	TIM—high order—trace identifier mismatch failure
AU_AIS	AIS—administration unit—alarm indication signal
LOP	LOP—administration unit—loss of pointer
UNEQ	SLMF—unequipped high order—path unequipped
PLM	SLMF—PLM high order—path label mismatch
RAI	RFI—high order—remote failure/alarm indication
LOP	LOP—tributary unit—loss of pointer
TU-AIS	AIS—tributary unit—alarm indication signal
UNEQ	SLMF—unequipped low order—path unequipped
PLM	SLMF—PLM low order—path label mismatch
TIM	TIM low order—trace identifier mismatch failure
RAI	RFI—low order—remote failure/alarm indication
TIMING_SYNCH	G811—primary reference clock traceable
TIMING_SYNCH	G812—transit node clock traceable
TIMING_SYNCH	G812—local node clock traceable
TIMING_SYNCH	G813—synchronous equipment timing source traceable
UNIDENTIFIED	E1 loopback due to FEAC command
UNIDENTIFIED	E1 loopback command sent to far end
UNIDENTIFIED	E3 loopback due to FEAC command
ENV	Far-end multiple E1 LOS detected on E3
ENV	Far-end E1 equipment failure non-service affecting
ENV	Far-end E1 equipment failure service affecting
ENV	Far-end single E1 LOS
ENV	Far-end E3 equipment failure service affecting
UNIDENTIFIED	E3 loopback command sent to far end
ENV	Far-end E3 equipment failure non-service affecting
ENV	Low voltage—battery A
ENV	High voltage—battery A
ENV	Low voltage—battery B
ENV	High voltage—battery B
UNIDENTIFIED	Procedural error—MS-SPRing out of sync
UNIDENTIFIED	MS-SPRing tables resynchronized
UNIDENTIFIED	Automatic SNCP switch caused by [AIS | LOP | UNEQ | PDI | SFBER | SDBER]
UNIDENTIFIED	STM concatenation error
UNIDENTIFIED	E3 idle condition
UNIDENTIFIED	IOS config copy failed
UNIDENTIFIED	IOS config copy in progress
UNIDENTIFIED	Alarms suppressed by user command
UNIDENTIFIED	Alarms suppressed for maintenance
UNIDENTIFIED	—

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Synchronization reference frequency out of bounds
UNIDENTIFIED	Ether Tx excess flow control
UNIDENTIFIED	Ether Tx oversubscribed
UNIDENTIFIED	Transport layer failure
UNIDENTIFIED	Ether Rx excess flow control
UNIDENTIFIED	Ether Rx oversubscribed
UNIDENTIFIED	Ether Tx underrun
UNIDENTIFIED	SNTP host failure
UNIDENTIFIED	DS3 frame format mismatch
UNIDENTIFIED	Alarms/events suppressed for this object
BER_SD	BER threshold exceeded for signal degrade—high order
BER_SF	BER threshold exceeded for signal failure—high order
BER_SD	BER threshold exceeded for signal degrade—low order
BER_SF	BER threshold exceeded for signal failure—low order
UNIDENTIFIED	Failure to switch to protection—high-order path
UNIDENTIFIED	Failure to switch to protection—low-order path
UNIDENTIFIED	Wait to restore
ENV	Extreme high voltage—battery A
ENV	Extreme low voltage—battery A
ENV	Extreme high voltage—battery B
ENV	Extreme low voltage—battery B
UNIDENTIFIED	Switching matrix module failure
UNIDENTIFIED	Signaling unable to set up circuit
UNIDENTIFIED	RSVP hello FSM to neighbor down
UNIDENTIFIED	LMP hello FSM to control channel down
UNIDENTIFIED	LMP neighbor discovery has failed
UNIDENTIFIED	Unauthorized incoming signaling request
UNIDENTIFIED	Signaled circuit going down
UNIDENTIFIED	Autonomous PM report message inhibited
EQPT	I/O slot to XCON communication failure
UNIDENTIFIED	Forced switch to primary reference
UNIDENTIFIED	Forced switch to secondary reference
UNIDENTIFIED	Forced switch to third reference
UNIDENTIFIED	Forced switch to internal clock
LOM	LOF—administration unit—loss of multiframe
UNIDENTIFIED	Admin logout of user
UNIDENTIFIED	User locked out
UNIDENTIFIED	Admin lockout of user
UNIDENTIFIED	Admin lockout clear
SECURITY_VIOLATION	Security—invalid login—username—see audit log
SECURITY_VIOLATION	Security—invalid login—password—see audit log
SECURITY_VIOLATION	Security—invalid login—locked out—see audit log
SECURITY_VIOLATION	Security—invalid login—already logged on—see audit log

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Login of user
UNIDENTIFIED	Automatic logout of idle user
UNIDENTIFIED	Logout of user
UNIDENTIFIED	Cross-connect loopback
UNIDENTIFIED	Error in startup config
UNIDENTIFIED	No startup config
EQPT	Laser approaching end of life
UNIDENTIFIED	Database backup failed
UNIDENTIFIED	Database restore failed
UNIDENTIFIED	8B10B out of sync
AIS	ODUk—alarm indication signal
AIS	OTUk—alarm indication signal
UNIDENTIFIED	OTUk—backward defect indicator
UNIDENTIFIED	ODUk—backward defect indicator
UNIDENTIFIED	FEC uncorrected word
UNIDENTIFIED	GCC embedded operation channel failure
UNIDENTIFIED	OTUk—incoming alignment error
UNIDENTIFIED	ODUk—locked defect—PM
LOM	Loss of multiframe
UNIDENTIFIED	ODUk—open connection indication
PLM	Payload type identifier mismatch
TIM	ODUk—trail trace identifier mismatch
TIM	OTUk—trail trace identifier mismatch
EQPT	Equipment high laser bias
EQPT	Equipment high laser temp
EQPT	Equipment high laser Peltier
EQPT	Equipment high Rx power
EQPT	Equipment high Tx power
EQPT	Equipment high Rx temperature
EQPT	Equipment low Tx temperature
EQPT	Equipment high transceiver voltage
EQPT	Equipment low laser bias
EQPT	Equipment low laser temp
EQPT	Equipment low laser Peltier
EQPT	Equipment low Rx power
EQPT	Equipment low Tx power
EQPT	Equipment low transceiver voltage
EQPT	Equipment Rx locked
EQPT	Equipment squelched
EQPT	Equipment Tx locked
BER_SF	OTUk—signal failure
BER_SF	ODUk—signal failure
BER_SD	OTUk—signal degrade
BER_SD	ODUk—signal degrade
UNIDENTIFIED	Pluggable port missing
UNIDENTIFIED	Pluggable port rate mismatch
SECURITY_VIOLATION	Pluggable port security code mismatch
UNIDENTIFIED	TCI not selected

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	TCI 1 clock failure
UNIDENTIFIED	ODUk—backward defect indicator
UNIDENTIFIED	ODUk—open connection indication
UNIDENTIFIED	ODUk—locked defect—[TCM1 | TCM2]
AIS	ODUk—alarm indication signal
TIM	ODUk—trail trace identifier mismatch
BER_SD	ODUk—signal degrade—[TCM1 | TCM2]
BER_SF	ODUk—signal failure—[TCM1 | TCM2]
UNIDENTIFIED	ODUk—open connection indication
UNIDENTIFIED	ODUk—locked defect—TCM1
AIS	ODUk—alarm indication signal
TIM	ODUk—trail trace identifier mismatch
BER_SD	ODUk—signal degrade—TCM1
BER_SF	ODUk—signal failure—TCM1
LOF	OTUk—loss of frame
UNIDENTIFIED	TCI 2 clock failure
UNIDENTIFIED	Audit log 80 percent full
UNIDENTIFIED	Module communication failure
UNIDENTIFIED	Need to save running config
UNIDENTIFIED	Audit log 100 percent full; oldest records will be lost
UNIDENTIFIED	Standby database out of sync
EQPT	Redundant power capability lost
EQPT	Equipment wavelength mismatch
UNIDENTIFIED	DSP communication failure
UNIDENTIFIED	DSP failure
UNIDENTIFIED	Loss of channel
UNIDENTIFIED	FEC mismatch
UNIDENTIFIED	Optical power degrade low
UNIDENTIFIED	Optical power degrade high
SECURITY_VIOLATION	Security intrusion attempt detected—see audit log
SECURITY_VIOLATION	Security intrusion attempt detected—see audit log
UNIDENTIFIED	Variable optical attenuator degrade low
UNIDENTIFIED	Variable optical attenuator degrade high
UNIDENTIFIED	Variable optical attenuator failure low
UNIDENTIFIED	Variable optical attenuator failure high
UNIDENTIFIED	Laser bias degrade
UNIDENTIFIED	Laser bias failure
UNIDENTIFIED	Laser temperature degrade
UNIDENTIFIED	Optical amplifier gain degrade low
UNIDENTIFIED	Optical amplifier gain degrade high
UNIDENTIFIED	Optical amplifier gain failure low
UNIDENTIFIED	Optical amplifier gain failure high
UNIDENTIFIED	Laser auto power reduction
UNIDENTIFIED	Case temperature degrade
UNIDENTIFIED	Fiber temperature degrade
UNIDENTIFIED	Shutter open
UNIDENTIFIED	AWG temperature degrade

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	AWG temperature failure
UNIDENTIFIED	AWG over temperature
UNIDENTIFIED	Shutter insertion loss variation degrade low
UNIDENTIFIED	Shutter insertion loss variation degrade high
UNIDENTIFIED	AWG warm up
UNIDENTIFIED	Plug-in module communication failure
UNIDENTIFIED	Optical amplifier initialization
UNIDENTIFIED	Optical channel connection failure
UNIDENTIFIED	Optical channel activation failure
UNIDENTIFIED	Optical channel deactivation failure
UNIDENTIFIED	Network topology incomplete
UNIDENTIFIED	Optical network type mismatch
UNIDENTIFIED	Automatic power control failure
UNIDENTIFIED	Automatic power control disabled
UNIDENTIFIED	Ring ID mismatch
TIM	TIM section monitor—trace identifier mismatch failure
UNIDENTIFIED	Optical channel incomplete
UNIDENTIFIED	Enhanced remote failure indication—path—server
UNIDENTIFIED	Enhanced remote failure indication—path—connectivity
UNIDENTIFIED	Enhanced remote failure indication—path—payload
UNIDENTIFIED	Automatic laser shutdown
UNIDENTIFIED	Firewall has been disabled
UNIDENTIFIED	Optical power failure low
UNIDENTIFIED	Optical power failure high
EQPT	Battery failure
EQPT	Extreme high voltage
EQPT	Extreme low voltage
UNIDENTIFIED	High voltage
UNIDENTIFIED	Low voltage
UNIDENTIFIED	Connection equipment mismatch
UNIDENTIFIED	Disable inactive user
UNIDENTIFIED	Disable inactive user
UNIDENTIFIED	Suspend user
UNIDENTIFIED	Suspend user clear
UNIDENTIFIED	Line DCC termination failure
UNIDENTIFIED	Multiplex section DCC termination failure
UNIDENTIFIED	Gigabit Ethernet out of sync
AIS	Alarm indication signal in Tx
AIS	Remote alarm indication in Tx
LOF	Loss of frame in Tx
LOS	Incoming signal loss on fibre channel interface
UNIDENTIFIED	Incoming synchronization loss on fibre channel interface
UNIDENTIFIED	Out of frame detected by GFP receiver
UNIDENTIFIED	Client signal loss frames detected by GFP receiver

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Client synchronization loss frames detected by GFP receiver
UNIDENTIFIED	Sequence mismatch
UNIDENTIFIED	Loss of alignment
UNIDENTIFIED	Out of use—administrative command
UNIDENTIFIED	Out of use—transport failure
UNIDENTIFIED	VCAT group down
UNIDENTIFIED	VCAT group degraded
UNIDENTIFIED	VCAT group incomplete
UNIDENTIFIED	K byte channel failure
UNIDENTIFIED	Industrial high temperature
FOP_APS	APS invalid mode
UNIDENTIFIED	IP address already in use within the same DCC area
UNIDENTIFIED	Node name already in use within the same DCC area
UNIDENTIFIED	Rear panel Ethernet link removed
UNIDENTIFIED	Manual switch to protect resulted in no traffic switch
UNIDENTIFIED	Manual switch back to working resulted in no traffic switch
UNIDENTIFIED	Forced switch to protection resulted in no traffic switch
UNIDENTIFIED	Forced switch back to working resulted in no traffic switch
LOS	Incoming payload signal absent
UNIDENTIFIED	Incoming overhead signal absent
UNIDENTIFIED	Optical safety remote interlock on
UNIDENTIFIED	OSPF hello fail
UNIDENTIFIED	Automatic power control correction skipped
UNIDENTIFIED	Plug-in module range settings mismatch
UNIDENTIFIED	APC cannot set value due to range limits
UNIDENTIFIED	Automatic WDM ANS finished
UNIDENTIFIED	Port failure
UNIDENTIFIED	Unreachable port target power
UNIDENTIFIED	Port add power degrade low
UNIDENTIFIED	Port add power degrade high
UNIDENTIFIED	Port add power fail low
UNIDENTIFIED	Port add power fail high
UNIDENTIFIED	Equipment power failure at connector A
UNIDENTIFIED	Equipment power failure at connector B
UNIDENTIFIED	Equipment power failure at return connector A
UNIDENTIFIED	Far-end manual switch back to working—span
UNIDENTIFIED	Far-end forced switch back to working—span
UNIDENTIFIED	Universal transponder module hardware failure
UNIDENTIFIED	Universal transponder module communication failure
UNIDENTIFIED	Automatic power control terminated on manual request

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Fibre channel distance extension credit starvation
PLM	GFP user payload mismatch
UNIDENTIFIED	GFP fibre channel distance extension mismatch
UNIDENTIFIED	GFP fibre channel distance extension buffer starvation
BER_SF	GFP client signal fail detected
LOF	GFP loss of frame delineation
UNIDENTIFIED	GFP extension header mismatch
UNIDENTIFIED	Signal loss on data interface
UNIDENTIFIED	Encapsulation mismatch—path
UNIDENTIFIED	Encapsulation mismatch—VT
UNIDENTIFIED	Encapsulation mismatch high order—path
UNIDENTIFIED	Encapsulation mismatch low order—path
UNIDENTIFIED	Synchronization loss on data interface
AIS	ODUk-1...4—alarm indication signal
UNIDENTIFIED	LCAS control word CRC check failure
UNIDENTIFIED	Duplicate serial number detected on a pluggable entity
UNIDENTIFIED	LCAS VCG member Tx side in DNU state
UNIDENTIFIED	LCAS VCG member Tx side in ADD state
UNIDENTIFIED	LCAS VCG member Rx side in FAIL state
UNIDENTIFIED	Optimized 1+1 APS primary facility
UNIDENTIFIED	Optimized 1+1 APS primary section mismatch
UNIDENTIFIED	Optimized 1+1 APS invalid primary section
UNIDENTIFIED	Composite clock high line voltage
BER_SD	BER threshold exceeded for signal degrade—VT
BER_SF	BER threshold exceeded for signal failure—VT
UNIDENTIFIED	Alarms suppressed on out-of-group VCAT member
UNIDENTIFIED	Span length out of range
UNIDENTIFIED	Temperature reading mismatch between SC cards
UNIDENTIFIED	Voltage reading mismatch between SC cards
TIM	VT path trace identifier mismatch
UNIDENTIFIED	BLSR software version mismatch
UNIDENTIFIED	Bridge and roll occurred
UNIDENTIFIED	Bridge and roll is pending a valid signal
TIM	Open I/O slot(s)
UNIDENTIFIED	ISIS adjacency failure
SECURITY_VIOLATION	Session time limit expired
SECURITY_VIOLATION	User password change required
SECURITY_VIOLATION	Remote authentication fail
TIM	Section trace identifier mismatch
TIM	Regenerator section trace identifier mismatch
UNIDENTIFIED	Switching matrix module failure—working
UNIDENTIFIED	Switching matrix module failure—protect
UNIDENTIFIED	Provisioning mismatch
UNIDENTIFIED	Slot communication disabled
UNIDENTIFIED	MSSP software version mismatch
UNIDENTIFIED	Archival of audit log failed

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	RPR wrapped
UNIDENTIFIED	Software mismatch
UNIDENTIFIED	Link layer keepalive failure
UNIDENTIFIED	Bad packet count exceeds threshold
UNIDENTIFIED	Autonegotiation remote failure indication
UNIDENTIFIED	Shelf communication failure
UNIDENTIFIED	Duplicated shelf identifier
UNIDENTIFIED	Ring is squelching STS traffic
UNIDENTIFIED	Ring is squelching VT traffic
UNIDENTIFIED	Clock bus failure—shelf controller A
UNIDENTIFIED	Clock bus failure—shelf controller B
UNIDENTIFIED	Loss of clock from mate shelf controller
UNIDENTIFIED	Payload missing indication
UNIDENTIFIED	Forward defect indication
UNIDENTIFIED	Optical termination incomplete
UNIDENTIFIED	Span loss not checked
UNIDENTIFIED	Ring is squelching higher order traffic
UNIDENTIFIED	Ring is squelching lower order traffic
UNIDENTIFIED	Trail signal fail
UNIDENTIFIED	DS1 loopback command sent to far end
UNIDENTIFIED	Multiplex section—signal degraded
UNIDENTIFIED	Multiplex section—excessive errors
UNIDENTIFIED	High-order path—signal degraded
UNIDENTIFIED	High-order path—excessive errors
UNIDENTIFIED	Low-order path—signal degraded
UNIDENTIFIED	Low-order path—excessive errors
UNIDENTIFIED	Regenerator section—DCC termination failure
UNIDENTIFIED	Network memory pool low
UNIDENTIFIED	OSPF routing table overflow
UNIDENTIFIED	Auto laser shutdown disabled
UNIDENTIFIED	RPR protection is active
UNIDENTIFIED	Max RPR station number exceeded
UNIDENTIFIED	RPR protection configuration mismatched
UNIDENTIFIED	Reserved bandwidth link rate exceeded on ringlet [0|1]
UNIDENTIFIED	RPR interface in pass-through mode
UNIDENTIFIED	RPR peer node is missing
UNIDENTIFIED	RPR RI failure
UNIDENTIFIED	RPR signal failure
UNIDENTIFIED	RPR signal degrade
UNIDENTIFIED	RPR span mismatch
UNIDENTIFIED	LMP failure
UNIDENTIFIED	LMP signal degrade
UNIDENTIFIED	LMP signal failure
UNIDENTIFIED	LMP unallocated data link
UNIDENTIFIED	APC wrong gain setpoint
EQPT	Non-Cisco PPM inserted
EQPT	Unqualified PPM inserted

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Fast automatic protection switching
UNIDENTIFIED	Fast automatic protection switching config mismatch
UNIDENTIFIED	ADM peer cards interlink failure
EQPT	Fan tray mismatch
UNIDENTIFIED	Fibre channel distance extension function not established
UNIDENTIFIED	LCAS sink group error
UNIDENTIFIED	LCAS VCG member Rx side in DNU state
UNIDENTIFIED	Card/ports unable to provide protection
UNIDENTIFIED	TCC front port link loss
UNIDENTIFIED	Shutter insertion loss variation degrade low
UNIDENTIFIED	Shutter insertion loss variation degrade high
UNIDENTIFIED	Idle signal condition
UNIDENTIFIED	Idle signal condition in TX
UNIDENTIFIED	Automatic power control correction skipped
UNIDENTIFIED	Port add power fail low
UNIDENTIFIED	Port add power fail high
UNIDENTIFIED	Span length out of range
UNIDENTIFIED	Port add power degrade low
UNIDENTIFIED	Port add power degrade high
UNIDENTIFIED	DCU loss failure

UNIDENTIFIED	OCHNC maintenance
UNIDENTIFIED	K-byte channel failure
UNIDENTIFIED	PRP protection configuration mismatched
UNIDENTIFIED	Reserved bandwidth link rate exceeded on ring Let0
UNIDENTIFIED	Reserved bandwidth link rate exceeded on ring Let1
UNIDENTIFIED	Security invalid login—locked out
UNIDENTIFIED	Security invalid login—already logged on
UNIDENTIFIED	Syslog messages
UNIDENTIFIED	Configuration events
UNIDENTIFIED	Link up/link down trap
UNIDENTIFIED	Cold/warm start trap
UNIDENTIFIED	Authentication failure notification
EQPT	Unqualified PPM inserted
UNIDENTIFIED	Fast automatic protection switching
UNIDENTIFIED	Fast automatic protection switching config mismatch
UNIDENTIFIED	ADM peer cards interlink failure
EQPT	Fan tray mismatch
UNIDENTIFIED	Fibre channel distance extension function not established
UNIDENTIFIED	LCAS sink group error
UNIDENTIFIED	LCAS VCG member Rx side in DNU state
UNIDENTIFIED	Card/ports unable to provide protection

UNIDENTIFIED	TCC front port link loss
UNIDENTIFIED	Shutter insertion loss variation degrade low
UNIDENTIFIED	Shutter insertion loss variation degrade high
UNIDENTIFIED	Idle signal condition
UNIDENTIFIED	Idle signal condition in TX
UNIDENTIFIED	Automatic power control correction skipped
UNIDENTIFIED	Port add power fail low
UNIDENTIFIED	Port add power fail high
UNIDENTIFIED	DCU loss failure
UNIDENTIFIED	OCHNC maintenance
UNIDENTIFIED	K-byte channel failure
UNIDENTIFIED	PRP protection configuration mismatched
UNIDENTIFIED	Reserved bandwidth link rate exceeded on ring Let0
UNIDENTIFIED	Reserved bandwidth link rate exceeded on ring Let1
UNIDENTIFIED	Security invalid login—locked out
UNIDENTIFIED	Security invalid login—already logged on
UNIDENTIFIED	Syslog messages
UNIDENTIFIED	Configuration events
UNIDENTIFIED	Link up/link down trap
UNIDENTIFIED	Cold/warm start trap
UNIDENTIFIED	Authentication failure notification
UNIDENTIFIED	Alarm indication signal in TX—customer installation
UNIDENTIFIED	Remote alarm indication in TX—customer installation

5.5.6 GateWay/CORBA Mapping—ONS 15454-M2, ONS 15454-M6, ONS NCS2K-M2, ONS NCS2K -M6, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH

The following table lists the GateWay/CORBA mapping for the ONS 15454-M2, ONS 15454-M6, ONS NCS2K-M2, ONS NCS2K -M6, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH NEs.

Table 5-22: GateWay/CORBA Mapping for the ONS 15454-M2, ONS 15454-M6, ONS NCS2K-M2, ONS NCS2K -M6, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Ethernet OSC termination failure
UNIDENTIFIED	Laser shutdown due to insertion of non-Cisco PPM
UNIDENTIFIED	Software signature verification failure
UNIDENTIFIED	Active volume software signature verification failure
UNIDENTIFIED	Protect volume software signature verification failure
UNIDENTIFIED	Channel shutdown due to wavelength drift
UNIDENTIFIED	Software activation failure
UNIDENTIFIED	USB write failure
UNIDENTIFIED	USB sync in progress

5.5.7 GateWay/CORBA Mapping—PTS

The following table lists the GateWay/CORBA mapping for PTS.

Table 5-23: GateWay/CORBA Mapping for PTS

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Satellite panel discovery failure
UNIDENTIFIED	Satellite panel active link failure
UNIDENTIFIED	Satellite panel communication failure
UNIDENTIFIED	Satellite panel improper configuration
UNIDENTIFIED	Satellite panel fan mismatch of equipment and attributes
UNIDENTIFIED	Satellite panel fan failure
UNIDENTIFIED	Satellite panel partial fan failure
UNIDENTIFIED	Satellite panel fan manufacturing data memory (EEPROM) failure
UNIDENTIFIED	Satellite panel fan unit is missing
UNIDENTIFIED	Satellite panel industrial high temperature
UNIDENTIFIED	Satellite panel high temperature
UNIDENTIFIED	Satellite panel battery failure A
UNIDENTIFIED	Packet transport service failed
UNIDENTIFIED	PMD degrade
UNIDENTIFIED	License will expire within 24 hours
UNIDENTIFIED	License will expire at some time (after 1 day, but before 14 days)
UNIDENTIFIED	License expired
UNIDENTIFIED	License count violation
UNIDENTIFIED	Temporary license is in use
UNIDENTIFIED	Evaluation license is in use
UNIDENTIFIED	License is missing
UNIDENTIFIED	Planned switchover
UNIDENTIFIED	Protection card configuration mismatch
UNIDENTIFIED	Running low on resources
UNIDENTIFIED	No more resources available
UNIDENTIFIED	Route processor switchover occurred
UNIDENTIFIED	Standby TCC-NE clock is internal clock
UNIDENTIFIED	Pseudowire down
UNIDENTIFIED	Working pseudowire control plane down
UNIDENTIFIED	Protect pseudowire control plane down
UNIDENTIFIED	Working pseudowire connectivity check down
UNIDENTIFIED	Protect pseudowire connectivity check down
UNIDENTIFIED	Pseudowire traffic switched protection
UNIDENTIFIED	Working pseudowire local AC Tx port fault
UNIDENTIFIED	Protect pseudowire local AC Tx port fault
UNIDENTIFIED	Working pseudowire local AC Rx port fault
UNIDENTIFIED	Protect pseudowire local AC Rx port fault
UNIDENTIFIED	Working pseudowire remote AC Tx port fault
UNIDENTIFIED	Protect pseudowire remote AC Tx port fault
UNIDENTIFIED	Working pseudowire remote AC Rx port fault
UNIDENTIFIED	Protect pseudowire remote AC Rx port fault

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Working local pseudowire not forwarding
UNIDENTIFIED	Protect local pseudowire not forwarding
UNIDENTIFIED	Working remote pseudowire not forwarding
UNIDENTIFIED	Protected remote pseudowire not forwarding
UNIDENTIFIED	TP tunnel down
UNIDENTIFIED	Working LSP down
UNIDENTIFIED	Protect LSP down
UNIDENTIFIED	Working LSP alarm indication signal
UNIDENTIFIED	Protect LSP alarm indication signal
UNIDENTIFIED	Working LSP remote defect indication
UNIDENTIFIED	Protect LSP remote defect indication
UNIDENTIFIED	BFD down
UNIDENTIFIED	TP traffic switched from working to protection
UNIDENTIFIED	Working TP lockout
UNIDENTIFIED	Protect TP lockout
UNIDENTIFIED	EFP failed
UNIDENTIFIED	TE tunnel down
UNIDENTIFIED	MAC system limit reached
UNIDENTIFIED	MAC bridge domain limit reached
UNIDENTIFIED	Working LSP link down indication
UNIDENTIFIED	Protect LSP link down indication
UNIDENTIFIED	Chromatic dispersion
UNIDENTIFIED	OTUk-SM backward incoming alignment error
UNIDENTIFIED	Resource allocation failed
UNIDENTIFIED	Working LSP lock report
UNIDENTIFIED	Protect LSP lock report
UNIDENTIFIED	Satellite panel battery failure B

5.5.8 GateWay/CORBA Mapping—ONS 15600 SONET

The following table lists the GateWay/CORBA mapping for the ONS 15600 SONET.

Table 5-24: GateWay/CORBA Mapping for the ONS 15600 SONET

TMF Mapping	NE Native Probable Cause
EQPT	Alarms/events suppressed for this object
UNIDENTIFIED	Audit log 100 percent full
UNIDENTIFIED	Audit log 80 percent full
UNIDENTIFIED	BLSR out of sync
UNIDENTIFIED	BLSR multinode table update completed
UNIDENTIFIED	TSC switched to alternate Ethernet port
ENV	Failure detected external to the NE
DCC_FAILURE	DCC channel loss
DCC_FAILURE	SDCC termination failure
AIS	Alarm indication signal
AIS	Alarm indication signal—line
AIS	Alarm indication signal—path
FOP_APS	Byte failure
FOP_APS	Protection switching channel match failure
FOP_APS	Automatic protection switch mode mismatch

TMF Mapping	NE Native Probable Cause
FOP_APS	Connection loss
FOP_APS	Default K byte
FOP_APS	Far-end protection line failure
FOP_APS	APS channel—far-end protection line signal degrade
FOP_APS	Improper APS code
FOP_APS	Inconsistent APS code
FOP_APS	Node ID mismatch
UNIDENTIFIED	STS concatenation error
LOF	Loss of frame
LOP	Loss of pointer—path
LOS	Loss of signal
RAI	Remote failure indication—line
RAI	Remote failure indication—path
BER_SD	BER threshold exceeded for signal degrade—line
BER_SD	BER threshold exceeded for signal degrade—path
BER_SD	Far-end BER threshold passed for signal degrade—ring
BER_SD	Far-end BER threshold passed for signal degrade—span
BER_SF	BER threshold exceeded for signal failure—line
BER_SF	BER threshold exceeded for signal failure—path
BER_SF	Far-end BER threshold passed for signal failure—ring
BER_SF	Far-end BER threshold passed for signal failure—span
UNIDENTIFIED	Payload defect indication—path
UNEQ	Unequipped—path
UNIDENTIFIED	Don’t use for synchronization
UNIDENTIFIED	Failed to receive synchronization status message
UNIDENTIFIED	Synchronization status messages are disabled on this interface
UNIDENTIFIED	Primary reference source—stratum 1 traceable
UNIDENTIFIED	Reserved for network synchronization use
UNIDENTIFIED	SONET minimum clock traceable
UNIDENTIFIED	Stratum 2 traceable
UNIDENTIFIED	Stratum 3 traceable
UNIDENTIFIED	Stratum 3E traceable
UNIDENTIFIED	Stratum 4 traceable
UNIDENTIFIED	Synchronized—traceability unknown
UNIDENTIFIED	Transit node clock traceable
UNIDENTIFIED	Fifth synchronization reference failure
UNIDENTIFIED	Fourth synchronization reference failure
UNIDENTIFIED	Synchronization reference frequency out of bounds
UNIDENTIFIED	Primary synchronization reference failure
UNIDENTIFIED	Secondary synchronization reference failure
UNIDENTIFIED	Sixth synchronization reference failure
UNIDENTIFIED	Third synchronization reference failure
TIM	STS path trace identifier mismatch
ENV	NE power failure at connector A
ENV	NE power failure at connector B
EQPT	Power fuse failure
UNIDENTIFIED	Free running synchronization mode
UNIDENTIFIED	Stratum 3E fast start synchronization mode

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Holdover synchronization mode
UNIDENTIFIED	Autonomous messages inhibited
UNIDENTIFIED	Autonomous PM report message inhibited
UNIDENTIFIED	Mismatch of equipment and attributes
UNIDENTIFIED	Ring topology is under construction
ENV	Software fault—data integrity fault
ENV	Airflow failure
EQPT	Equipment fails to boot
UNIDENTIFIED	Connection ID mismatch on CXC 0
UNIDENTIFIED	Connection ID mismatch on CXC 1
EQPT	Clock module failure
EQPT	Control bus failure—I/O—TSC A
EQPT	Control bus failure—I/O—TSC B
EQPT	Clock bus failure—TSC A
EQPT	Clock bus failure—TSC B
EQPT	Control communication equipment failure
EQPT	Primary nonvolatile backup memory failure
EQPT	Manufacturing data memory (EEPROM) failure
EQPT	Payload bus failure—matrix A
EQPT	Payload bus failure—matrix B
UNIDENTIFIED	CXC operations suspended
EQPT	Equipment failure
EQPT	Diagnostics failure
EQPT	Equipment failure—high temperature
EQPT	Invalid MAC address
UNIDENTIFIED	Failure to switch to protection
UNIDENTIFIED	Failure to switch to protection—path
UNIDENTIFIED	Failure to switch to protection—ring
UNIDENTIFIED	Failure to switch to protection—span
EQPT	Fan failure
EQPT	Partial fan failure—speed degradation
EQPT	Partial fan failure—parts failure
EQPT	Clock module frequency mismatch
UNIDENTIFIED	OSPF hello fail
EQPT	Laser bias current high
EQPT	High laser temperature
UNIDENTIFIED	Software download failed
UNIDENTIFIED	Different software version
EQPT	Synchronization equipment unavailable
EQPT	Unprotected synchronization equipment
EQPT	Unprotected matrix equipment
UNIDENTIFIED	System upgrade in progress
EQPT	Matrix equipment unavailable
UNIDENTIFIED	Loopback—cross connect
UNIDENTIFIED	Loopback facility
UNIDENTIFIED	Loopback payload
UNIDENTIFIED	Loopback
SECURITY_VIOLATION	Security—invalid login (see audit trail)

TMF Mapping	NE Native Probable Cause
SECURITY_VIOLATION	Invalid login attempt threshold exceeded
UNIDENTIFIED	Normal condition
UNIDENTIFIED	Performance monitor threshold crossing alert
UNIDENTIFIED	Duplicate node ID
EQPT	Improper removal
UNIDENTIFIED	Both ends of fiber provisioned as east
UNIDENTIFIED	Far end of fiber is provisioned with different ring ID
UNIDENTIFIED	Protection switch
EQPT	Equipment power failure
EQPT	Equipment power failure at connector A
EQPT	Equipment power failure at connector B
EQPT	Equipment power failure at return connector A
EQPT	Equipment power failure at return connector B
UNIDENTIFIED	Automatic reset
UNIDENTIFIED	Automatic path-protection switch caused by AIS
UNIDENTIFIED	Automatic path-protection switch caused by LOP
UNIDENTIFIED	Automatic path-protection switch caused by PDI
UNIDENTIFIED	Automatic path-protection switch caused by SDBER
UNIDENTIFIED	Automatic path-protection switch caused by SFBER
UNIDENTIFIED	Automatic path-protection switch caused by UNEQ
UNIDENTIFIED	Cold restart
UNIDENTIFIED	Exercise ring
UNIDENTIFIED	Exercising ring successfully
UNIDENTIFIED	Far-end exercise ring
UNIDENTIFIED	Exercise request on ring failed
UNIDENTIFIED	Exercise span
UNIDENTIFIED	Exercising span successfully
UNIDENTIFIED	Far-end exercise span
UNIDENTIFIED	Exercise request on span failed
UNIDENTIFIED	Force switch request on facility/equipment
UNIDENTIFIED	Force switch request on ring
UNIDENTIFIED	Force switch request on span
UNIDENTIFIED	Far-end working facility forced to switch to protection unit—ring
UNIDENTIFIED	Far-end working facility forced to switch to protection unit—span
UNIDENTIFIED	Bidirectional full pass-through is active
UNIDENTIFIED	Unidirectional full pass-through is active
EQPT	Inhibit switch to protect request on equipment
EQPT	Inhibit switch to working request on equipment
UNIDENTIFIED	K bytes pass-through is active
UNIDENTIFIED	Far-end lockout of protection—all spans
UNIDENTIFIED	Far-end lockout of protection—ring
UNIDENTIFIED	Far-end lockout of working—ring
UNIDENTIFIED	Far-end lockout of working—span
UNIDENTIFIED	Far-end lockout of protection—span
UNIDENTIFIED	Lockout of protection
UNIDENTIFIED	Lockout of protection—ring
UNIDENTIFIED	Lockout of protection—span
UNIDENTIFIED	Lockout of working—ring

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Lockout of working—span
UNIDENTIFIED	Lockout switch request on facility/equipment
UNIDENTIFIED	Lockout switch request on ring
UNIDENTIFIED	Manual reset
UNIDENTIFIED	Far-end manual switch of working facility to protection unit—ring
UNIDENTIFIED	Far-end manual switch of working facility to protection unit—span
UNIDENTIFIED	Manual switch to fifth reference
UNIDENTIFIED	Manual switch to fourth reference
UNIDENTIFIED	Manual switch to internal clock
UNIDENTIFIED	Manual switch to primary reference
UNIDENTIFIED	Manual switch to second reference
UNIDENTIFIED	Manual switch to sixth reference
UNIDENTIFIED	Manual switch to third reference
UNIDENTIFIED	Manual switch request on facility/equipment
UNIDENTIFIED	Manual switch request on ring
UNIDENTIFIED	Manual switch request on span
UNIDENTIFIED	Powerfail restart
UNIDENTIFIED	Ring is segmented
UNIDENTIFIED	Ring switch is active on the east side
UNIDENTIFIED	Ring switch is active on the west side
UNIDENTIFIED	Span switch is active on the east side
UNIDENTIFIED	Span switch is active on the west side
UNIDENTIFIED	Ring is squelching traffic
UNIDENTIFIED	Squelching path
UNIDENTIFIED	Software download in progress
UNIDENTIFIED	Switch to fifth reference
UNIDENTIFIED	Switch to fourth reference
UNIDENTIFIED	Switch to primary reference
UNIDENTIFIED	Switch to second reference
UNIDENTIFIED	Switch to sixth reference
UNIDENTIFIED	Switch to third reference
ENV	System reboot
UNIDENTIFIED	Extra traffic preempted
UNIDENTIFIED	Switched back to working unit
UNIDENTIFIED	Switched to protection unit
UNIDENTIFIED	Warm restart
UNIDENTIFIED	Wait to restore
UNIDENTIFIED	Ring is in wait-to-restore state
UNIDENTIFIED	Span is in wait-to-restore state
UNIDENTIFIED	Bridge and roll has occurred
UNIDENTIFIED	Bridge and roll is pending a valid signal
UNIDENTIFIED	Admin logout of user
UNIDENTIFIED	Admin lockout of user
UNIDENTIFIED	Admin lockout clear
UNIDENTIFIED	Automatic logout of idle user
UNIDENTIFIED	Login of user
SECURITY_VIOLATION	Invalid login—locked out
SECURITY_VIOLATION	Invalid login—already logged on

TMF Mapping	NE Native Probable Cause
SECURITY_VIOLATION	Invalid login—password
SECURITY_VIOLATION	Invalid login—username
UNIDENTIFIED	Logout of user
UNIDENTIFIED	User locked out
UNIDENTIFIED	Open I/O slot(s)
UNIDENTIFIED	Improper matrix equipment inserted
EQPT	Equipment fails to boot
UNIDENTIFIED	No description
UNIDENTIFIED	Firewall has been disabled
UNIDENTIFIED	Loss of clock from mate TSC
UNIDENTIFIED	Alarms suppressed for maintenance
UNIDENTIFIED	SNTP host failure
UNIDENTIFIED	Disable inactive user
UNIDENTIFIED	Disable inactive clear
UNIDENTIFIED	Suspend user
UNIDENTIFIED	Duplicate serial number detected on a pluggable entity
EQPT	Equipment problem on carrier or PIM
EQPT	Equipment problem on PIM or PPM
UNIDENTIFIED	Suspend user clear
UNIDENTIFIED	Forced switch to primary reference
UNIDENTIFIED	Forced switch to secondary reference
UNIDENTIFIED	Forced switch to third reference
UNIDENTIFIED	Forced switch to internal clock
PLM	GFP user payload mismatch
UNIDENTIFIED	GFP fibre channel distance extension mismatch
UNIDENTIFIED	GFP fibre channel distance extension buffer starvation
BER_SF	GFP client signal fail detected
LOF	GFP loss of frame delineation
UNIDENTIFIED	GFP extension header mismatch
UNIDENTIFIED	Carrier loss on the LAN
UNIDENTIFIED	Encapsulation type mismatch
UNIDENTIFIED	Transport layer failure
PLM	Payload label mismatch—path
DCC_FAILURE	Line DCC termination failure
UNIDENTIFIED	K byte channel failure
UNIDENTIFIED	BLSR software version mismatch
SECURITY_VIOLATION	Security intrusion attempt detected—see audit log
UNIDENTIFIED	IP address already in use within the same DCC area
UNIDENTIFIED	Node name already in use within the same DCC area
UNIDENTIFIED	Free memory on card near zero
UNIDENTIFIED	Free memory on card very low
ENV	NE power failure at connector
UNIDENTIFIED	Standby database out of sync
UNIDENTIFIED	Database backup failed
UNIDENTIFIED	Database restore failed
UNIDENTIFIED	ISIS adjacency failure
EQPT	Equipment high laser bias
EQPT	Equipment high Rx power

TMF Mapping	NE Native Probable Cause
EQPT	Equipment high Tx power
EQPT	Equipment low laser bias
EQPT	Equipment low Rx power
EQPT	Equipment low Tx power
UNIDENTIFIED	Provisioning mismatch
UNIDENTIFIED	Illegal route addition to the network
UNIDENTIFIED	Session time limit expired
UNIDENTIFIED	User password change required
SECURITY_VIOLATION	User authentication rejected
UNIDENTIFIED	APS invalid mode
UNIDENTIFIED	Far-end manual switch back to working—span
UNIDENTIFIED	Far-end forced switch back to working—span
UNIDENTIFIED	Alarms suppressed by user command
UNIDENTIFIED	Ring is squelching STS traffic
UNIDENTIFIED	REPT^DBCHG messages inhibited
TIM	Regenerator section trace identifier mismatch
MS-AIS	AIS—multiplex section—alarm indication signal
RAI	RFI—multiplex section—remote failure/alarm indication
BER_SF	Multiplex section—excessive BER
BER_SD	Multiplex section—signal degrade
DCC_FAILURE	Multiplex section DCC termination failure
UNIDENTIFIED	G811—primary reference clock traceable
UNIDENTIFIED	G812T—transit node clock traceable
UNIDENTIFIED	G812L—local node clock traceable
UNIDENTIFIED	G813—synchronous equipment timing source traceable
UNIDENTIFIED	MS-SPRing out of sync
AU-AIS	AIS—administration unit—alarm indication signal
LOP	LOP—administration unit—loss of pointer
BER_SF	High-order path—excessive BER
BER_SD	High-order path—signal degrade
PLM	SLMF—PLM high order—path label mismatch
TIM	TIM high order—trace identifier mismatch failure
RAI	RFI—high order—remote failure/alarm indication
UNEQ	SLMF—unequipped high order—path unequipped
UNIDENTIFIED	Automatic SNCP switch caused by AIS
UNIDENTIFIED	Automatic SNCP switch caused by LOP
UNIDENTIFIED	Automatic SNCP switch caused by UNEQ
UNIDENTIFIED	Automatic SNCP switch caused by SFBER
UNIDENTIFIED	Automatic SNCP switch caused by SDBER
UNIDENTIFIED	Failure to switch to protection—high-order path
UNIDENTIFIED	MSSP multinode table update completed
UNIDENTIFIED	Bipolar violation
UNIDENTIFIED	High-order path—payload defect indication
UNIDENTIFIED	MSSP software version mismatch
DCC_FAILURE	Regenerator section—DCC termination failure
EQPT	Battery failure
UNIDENTIFIED	Extreme high voltage
UNIDENTIFIED	Extreme low voltage

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	High voltage
UNIDENTIFIED	Low voltage
UNIDENTIFIED	Voltage reading mismatch between SC cards
EQPT	Wavelength out of lock
EQPT	Automatic laser shutdown
UNIDENTIFIED	Ring is squelching high-order traffic
EQPT	Non-Cisco PPM inserted
EQPT	Unqualified PPM inserted

5.5.9 GateWay/CORBA Mapping—ONS 15600 SDH

The following table lists the GateWay/CORBA mapping for the ONS 15600 SDH.

Table 5-24: GateWay/CORBA Mapping for the ONS 15600 SDH

TMF Mapping	NE Native Probable Cause
EQPT	Alarms/events suppressed for this object
UNIDENTIFIED	Audit log 100 percent full
UNIDENTIFIED	Audit log 80 percent full
UNIDENTIFIED	BLSR out of sync
UNIDENTIFIED	BLSR multinode table update completed
UNIDENTIFIED	TSC switched to alternate Ethernet port
ENV	Failure detected external to the NE
DCC_FAILURE	DCC channel loss
DCC_FAILURE	SDCC termination failure
AIS	Alarm indication signal
AIS	Alarm indication signal—line
AIS	Alarm indication signal—path
FOP_APS	Byte failure
FOP_APS	Protection switching channel match failure
FOP_APS	Automatic protection switch mode mismatch
FOP_APS	Connection loss
FOP_APS	Default K byte
FOP_APS	Far-end protection line failure
FOP_APS	APS channel—far-end protection line signal degrade
FOP_APS	Improper APS code
FOP_APS	Inconsistent APS code
FOP_APS	Node ID mismatch
UNIDENTIFIED	STS concatenation error
LOF	Loss of frame
LOP	Loss of pointer—path
LOS	Loss of signal
RAI	Remote failure indication—line
RAI	Remote failure indication—path
BER_SD	BER threshold exceeded for signal degrade—line
BER_SD	BER threshold exceeded for signal degrade—path
BER_SD	Far-end BER threshold passed for signal degrade—ring
BER_SD	Far-end BER threshold passed for signal degrade—span
BER_SF	BER threshold exceeded for signal failure—line

TMF Mapping	NE Native Probable Cause
BER_SF	BER threshold exceeded for signal failure—path
BER_SF	Far-end BER threshold passed for signal failure—ring
BER_SF	Far-end BER threshold passed for signal failure—span
UNIDENTIFIED	Payload defect indication—path
UNEQ	Unequipped—path
UNIDENTIFIED	Don’t use for synchronization
UNIDENTIFIED	Failed to receive synchronization status message
UNIDENTIFIED	Synchronization status messages are disabled on this interface
UNIDENTIFIED	Primary reference source—stratum 1 traceable
UNIDENTIFIED	Reserved for network synchronization use
UNIDENTIFIED	SONET minimum clock traceable
UNIDENTIFIED	Stratum 2 traceable
UNIDENTIFIED	Stratum 3 traceable
UNIDENTIFIED	Stratum 3E traceable
UNIDENTIFIED	Stratum 4 traceable
UNIDENTIFIED	Synchronized—traceability unknown
UNIDENTIFIED	Transit node clock traceable
UNIDENTIFIED	Fifth synchronization reference failure
UNIDENTIFIED	Fourth synchronization reference failure
UNIDENTIFIED	Synchronization reference frequency out of bounds
UNIDENTIFIED	Primary synchronization reference failure
UNIDENTIFIED	Secondary synchronization reference failure
UNIDENTIFIED	Sixth synchronization reference failure
UNIDENTIFIED	Third synchronization reference failure
TIM	STS path trace identifier mismatch
ENV	NE power failure at connector A
ENV	NE power failure at connector B
EQPT	Power fuse failure
UNIDENTIFIED	Free running synchronization mode
UNIDENTIFIED	Stratum 3E fast start synchronization mode
UNIDENTIFIED	Holdover synchronization mode
UNIDENTIFIED	Autonomous messages inhibited
UNIDENTIFIED	Autonomous PM report message inhibited
UNIDENTIFIED	Mismatch of equipment and attributes
UNIDENTIFIED	Ring topology is under construction
ENV	Software fault—data integrity fault
ENV	Airflow failure
EQPT	Equipment fails to boot
UNIDENTIFIED	Connection ID mismatch on CXC 0
UNIDENTIFIED	Connection ID mismatch on CXC 1
EQPT	Clock module failure
EQPT	Control bus failure—I/O—TSC A
EQPT	Control bus failure—I/O—TSC B
EQPT	Clock bus failure—TSC A
EQPT	Clock bus failure—TSC B
EQPT	Control communication equipment failure
EQPT	Primary nonvolatile backup memory failure
EQPT	Manufacturing data memory (EEPROM) failure

TMF Mapping	NE Native Probable Cause
EQPT	Payload bus failure—matrix A
EQPT	Payload bus failure—matrix B
UNIDENTIFIED	CXC operations suspended
EQPT	Equipment failure
EQPT	Diagnostics failure
EQPT	Equipment failure—high temperature
EQPT	Invalid MAC address
UNIDENTIFIED	Failure to switch to protection
UNIDENTIFIED	Failure to switch to protection—path
UNIDENTIFIED	Failure to switch to protection—ring
UNIDENTIFIED	Failure to switch to protection—span
EQPT	Fan failure
EQPT	Partial fan failure—speed degradation
EQPT	Partial fan failure—parts failure
EQPT	Clock module frequency mismatch
UNIDENTIFIED	OSPF hello fail
EQPT	Laser bias current high
EQPT	High laser temperature
UNIDENTIFIED	Software download failed
UNIDENTIFIED	Different software version
EQPT	Synchronization equipment unavailable
EQPT	Unprotected synchronization equipment
EQPT	Unprotected matrix equipment
UNIDENTIFIED	System upgrade in progress
EQPT	Matrix equipment unavailable
UNIDENTIFIED	Loopback—cross connect
UNIDENTIFIED	Loopback facility
UNIDENTIFIED	Loopback payload
UNIDENTIFIED	Loopback
SECURITY_VIOLATION	Security—invalid login (see audit trail)
SECURITY_VIOLATION	Invalid login attempt threshold exceeded
UNIDENTIFIED	Normal condition
UNIDENTIFIED	Performance monitor threshold crossing alert
UNIDENTIFIED	Duplicate node ID
EQPT	Improper removal
UNIDENTIFIED	Both ends of fiber provisioned as east
UNIDENTIFIED	Far end of fiber is provisioned with different ring ID
UNIDENTIFIED	Protection switch
EQPT	Equipment power failure
EQPT	Equipment power failure at connector A
EQPT	Equipment power failure at connector B
EQPT	Equipment power failure at return connector A
EQPT	Equipment power failure at return connector B
UNIDENTIFIED	Automatic reset
UNIDENTIFIED	Automatic path-protection switch caused by AIS
UNIDENTIFIED	Automatic path-protection switch caused by LOP
UNIDENTIFIED	Automatic path-protection switch caused by PDI
UNIDENTIFIED	Automatic path-protection switch caused by SDBER

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Automatic path-protection switch caused by SFBER
UNIDENTIFIED	Automatic path-protection switch caused by UNEQ
UNIDENTIFIED	Cold restart
UNIDENTIFIED	Exercise ring
UNIDENTIFIED	Exercising ring successfully
UNIDENTIFIED	Far-end exercise ring
UNIDENTIFIED	Exercise request on ring failed
UNIDENTIFIED	Exercise span
UNIDENTIFIED	Exercising span successfully
UNIDENTIFIED	Far-end exercise span
UNIDENTIFIED	Exercise request on span failed
UNIDENTIFIED	Force switch request on facility/equipment
UNIDENTIFIED	Force switch request on ring
UNIDENTIFIED	Force switch request on span
UNIDENTIFIED	Far-end working facility forced to switch to protection unit—ring
UNIDENTIFIED	Far-end working facility forced to switch to protection unit—span
UNIDENTIFIED	Bidirectional full pass-through is active
UNIDENTIFIED	Unidirectional full pass-through is active
EQPT	Inhibit switch to protect request on equipment
EQPT	Inhibit switch to working request on equipment
UNIDENTIFIED	K bytes pass-through is active
UNIDENTIFIED	Far-end lockout of protection—all spans
UNIDENTIFIED	Far-end lockout of protection—ring
UNIDENTIFIED	Far-end lockout of working—ring
UNIDENTIFIED	Far-end lockout of working—span
UNIDENTIFIED	Far-end lockout of protection—span
UNIDENTIFIED	Lockout of protection
UNIDENTIFIED	Lockout of protection—ring
UNIDENTIFIED	Lockout of protection—span
UNIDENTIFIED	Lockout of working—ring
UNIDENTIFIED	Lockout of working—span
UNIDENTIFIED	Lockout switch request on facility/equipment
UNIDENTIFIED	Lockout switch request on ring
UNIDENTIFIED	Manual reset
UNIDENTIFIED	Far-end manual switch of working facility to protection unit—ring
UNIDENTIFIED	Far-end manual switch of working facility to protection unit—span
UNIDENTIFIED	Manual switch to fifth reference
UNIDENTIFIED	Manual switch to fourth reference
UNIDENTIFIED	Manual switch to internal clock
UNIDENTIFIED	Manual switch to primary reference
UNIDENTIFIED	Manual switch to second reference
UNIDENTIFIED	Manual switch to sixth reference
UNIDENTIFIED	Manual switch to third reference
UNIDENTIFIED	Manual switch request on facility/equipment
UNIDENTIFIED	Manual switch request on ring
UNIDENTIFIED	Manual switch request on span
UNIDENTIFIED	Powerfail restart
UNIDENTIFIED	Ring is segmented

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Ring switch is active on the east side
UNIDENTIFIED	Ring switch is active on the west side
UNIDENTIFIED	Span switch is active on the east side
UNIDENTIFIED	Span switch is active on the west side
UNIDENTIFIED	Ring is squelching traffic
UNIDENTIFIED	Squelching path
UNIDENTIFIED	Software download in progress
UNIDENTIFIED	Switch to fifth reference
UNIDENTIFIED	Switch to fourth reference
UNIDENTIFIED	Switch to primary reference
UNIDENTIFIED	Switch to second reference
UNIDENTIFIED	Switch to sixth reference
UNIDENTIFIED	Switch to third reference
ENV	System reboot
UNIDENTIFIED	Extra traffic preempted
UNIDENTIFIED	Switched back to working unit
UNIDENTIFIED	Switched to protection unit
UNIDENTIFIED	Warm restart
UNIDENTIFIED	Wait to restore
UNIDENTIFIED	Ring is in wait-to-restore state
UNIDENTIFIED	Span is in wait-to-restore state
UNIDENTIFIED	Bridge and roll has occurred
UNIDENTIFIED	Bridge and roll is pending a valid signal
UNIDENTIFIED	Admin logout of user
UNIDENTIFIED	Admin lockout of user
UNIDENTIFIED	Admin lockout clear
UNIDENTIFIED	Automatic logout of idle user
UNIDENTIFIED	Login of user
SECURITY_VIOLATION	Invalid login—locked out
SECURITY_VIOLATION	Invalid login—already logged on
SECURITY_VIOLATION	Invalid login—password
SECURITY_VIOLATION	Invalid login—username
UNIDENTIFIED	Logout of user
UNIDENTIFIED	User locked out
UNIDENTIFIED	Open I/O slot(s)
UNIDENTIFIED	Improper matrix equipment inserted
EQPT	Equipment fails to boot
UNIDENTIFIED	No description
UNIDENTIFIED	Firewall has been disabled
UNIDENTIFIED	Loss of clock from mate TSC
UNIDENTIFIED	Alarms suppressed for maintenance
UNIDENTIFIED	SNTP host failure
UNIDENTIFIED	Disable inactive user
UNIDENTIFIED	Disable inactive clear
UNIDENTIFIED	Suspend user
UNIDENTIFIED	Duplicate serial number detected on a pluggable entity
EQPT	EQPT problem on carrier or PIM
EQPT	EQPT problem on PIM or PPM

TMF Mapping	NE Native Probable Cause
UNIDENTIFIED	Suspend user clear
UNIDENTIFIED	Forced switch to primary reference
UNIDENTIFIED	Forced switch to secondary reference
UNIDENTIFIED	Forced switch to third reference
UNIDENTIFIED	Forced switch to internal clock
PLM	GFP user payload mismatch
UNIDENTIFIED	GFP fibre channel distance extension mismatch
UNIDENTIFIED	GFP fibre channel distance extension buffer starvation
BER_SF	GFP client signal fail detected
LOF	GFP loss of frame delineation
UNIDENTIFIED	GFP extension header mismatch
UNIDENTIFIED	Carrier loss on the LAN
UNIDENTIFIED	Encapsulation type mismatch
UNIDENTIFIED	Transport layer failure
PLM	Payload label mismatch—path
DCC_FAILURE	Line DCC termination failure
UNIDENTIFIED	K byte channel failure
UNIDENTIFIED	BLSR software version mismatch
SECURITY_VIOLATION	Security intrusion attempt detected—see audit log
UNIDENTIFIED	IP address already in use within the same DCC area
UNIDENTIFIED	Node name already in use within the same DCC area
UNIDENTIFIED	Free memory on card near zero
UNIDENTIFIED	Free memory on card very low
ENV	NE power failure at connector
UNIDENTIFIED	Standby database out of sync
UNIDENTIFIED	Database backup failed
UNIDENTIFIED	Database restore failed
UNIDENTIFIED	ISIS adjacency failure
EQPT	Equipment high laser bias
EQPT	Equipment high Rx power
EQPT	Equipment high Tx power
EQPT	Equipment low laser bias
EQPT	Equipment low Rx power
EQPT	Equipment low Tx power
UNIDENTIFIED	Provisioning mismatch
UNIDENTIFIED	Illegal route addition to the network
UNIDENTIFIED	Session time limit expired
UNIDENTIFIED	User password change required
SECURITY_VIOLATION	User authentication rejected
UNIDENTIFIED	APS invalid mode
UNIDENTIFIED	Far-end manual switch back to working—span
UNIDENTIFIED	Far-end forced switch back to working—span
UNIDENTIFIED	Alarms suppressed by user command
UNIDENTIFIED	Ring is squelching STS traffic
UNIDENTIFIED	REPT^DBCHG messages inhibited
TIM	Regenerator section trace identifier mismatch
MS-AIS	AIS—multiplex section—alarm indication signal
RAI	RFI—multiplex section—remote failure/alarm indication

TMF Mapping	NE Native Probable Cause
BER_SF	Multiplex section—excessive BER
BER_SD	Multiplex section—signal degrade
DCC_FAILURE	Multiplex section DCC termination failure
UNIDENTIFIED	G811—primary reference clock traceable
UNIDENTIFIED	G812T—transit node clock traceable
UNIDENTIFIED	G812L—local node clock traceable
UNIDENTIFIED	G813—synchronous equipment timing source traceable
UNIDENTIFIED	MS-SPRing out of sync
AU-AIS	AIS—administration unit—alarm indication signal
LOP	LOP—administration unit—loss of pointer
BER_SF	High-order path—excessive BER
BER_SD	High-order path—signal degrade
PLM	SLMF—PLM high order—path label mismatch
TIM	TIM high order—trace identifier mismatch failure
RAI	RFI—high order—remote failure/alarm indication
UNEQ	SLMF—unequipped high order—path unequipped
UNIDENTIFIED	Automatic SNCP switch caused by AIS
UNIDENTIFIED	Automatic SNCP switch caused by LOP
UNIDENTIFIED	Automatic SNCP switch caused by UNEQ
UNIDENTIFIED	Automatic SNCP switch caused by SFBER
UNIDENTIFIED	Automatic SNCP switch caused by SDBER
UNIDENTIFIED	Failure to switch to protection—high-order path
UNIDENTIFIED	MSSP multinode table update completed
UNIDENTIFIED	Bipolar violation
UNIDENTIFIED	High-order path—payload defect indication
UNIDENTIFIED	MSSP software version mismatch
DCC_FAILURE	Regenerator section—DCC termination failure
EQPT	Battery failure
UNIDENTIFIED	Extreme high voltage
UNIDENTIFIED	Extreme low voltage
UNIDENTIFIED	High voltage
UNIDENTIFIED	Low voltage
UNIDENTIFIED	Voltage reading mismatch between SC cards
EQPT	Wavelength out of lock
EQPT	Automatic laser shutdown
UNIDENTIFIED	Ring is squelching high-order traffic
EQPT	Non-Cisco PPM inserted
EQPT	Unqualified PPM inserted

6 Server Administration and Configuration

This chapter includes the following information on the administration and configuration of Cisco Prime Optical GateWay/CORBA:

6.1 Creating an OSS Client Profile for GateWay/CORBA

The CORBA gateway authenticates the OSS against a previously created user profile before allowing access to Prime Optical. You can create OSS client profiles for GateWay/CORBA sessions. Each OSS profile defines GateWay/CORBA parameters, such as the OSS profile name, password, and IP address. OSS client profiles are stored in the GateWay/CORBA Users table.

Step 1 Log into the Cisco Prime Optical client with administrator privileges.

Step 2 In the Domain Explorer window, choose **Administration** > **GW/CORBA Users**.

Step 3 Choose **Edit** > **Add** (or click the **Create a New User** tool).

Step 4 In the **Add GW/CORBA User** window, enter the following OSS client information:

- OSS Profile Name—Name of the OSS profile.
- Password—Password that the OSS client uses to log into the Prime Optical server. The password must contain at least one special character, at least two alphabetic characters (A-Z, a-z), and at least one numeric character (0-9). Apostrophes (‘) are not accepted.
- Confirm Password—Re-enter the password to confirm it.

Step 5 Click **OK** to confirm the information. Changes take effect immediately. The GW/CORBA Users table receives a refresh event. If automatic refresh is enabled, the new OSS client profile appears as a new row in the table. If automatic refresh is disabled, click the Refresh Data tool to see the new OSS client profile in the table.

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

6.2 Deleting an OSS Client Profile for GateWay/CORBA

Step 1 Log into the Prime Optical client with administrator privileges.

Step 2 In the Domain Explorer window, choose **Administration** > **GW/CORBA Users**.
Step 3 The GW/CORBA Users table displays a list of available OSS users. Select the OSS user to delete.
Step 4 Choose **Edit** > **Delete** (or click the **Delete a User** tool) to delete the OSS profile from the Prime Optical database.
Step 5 Click **OK** to confirm the deletion. The OSS client profile name is deleted from the GW/CORBA Users table.

✐ If the OSS is connected to Prime Optical when the profile is being deleted, Prime Optical does not terminate the OSS session.

6.3 Viewing Currently Logged In GateWay/CORBA OSS Users

Step 1 Log into the Prime Optical client with administrator privileges.
Step 2 In the Domain Explorer window, choose **Administration** > **GW/CORBA Users**.
Step 3 The GW/CORBA Users table displays a list of available OSS users. Click the **Show Logged in GW CORBA Users** tool.
Step 4 In the **Active GW/CORBA Users** table, a list of currently logged-in users is displayed, including the OSS profile name, IP address to which the user is logged in, and the login time.

6.4 Logging Out GateWay/CORBA OSS Users

Step 1 Log into the Prime Optical client with administrator privileges.
Step 2 In the Domain Explorer window, choose **Administration** > **GW/CORBA Users**.
Step 3 The **GW/CORBA Users** table displays a list of available OSS users. Click the **Show Logged in GW CORBA Users** tool.
Step 4 In the **Active GW/CORBA Users** table, a list of currently logged-in users is displayed, including the OSS profile name, IP address to which the user is logged in, and the login time. Select the user to log off.
Step 5 Click the **Log Out GW CORBA User** tool.

- The user session is cleared from the Active GW/CORBA Users table. You will notice the loss of session during the next ping cycle or when you try to perform an operation on another manager. Some examples of user operations are:
- The user is connected and performs a query operation on the EMS. The OSS user starts to query the EMS by getting a fresh object reference from the manager through an emsSession query. Because the session has been cleared by the GateWay/CORBA service, the OSS user receives an exception and notices the loss of session.
- The Prime Optical client forcefully logs out the user. The OSS user does not immediately notice the loss of session when the Prime Optical client forces a logout. To immediately log out the user, the GateWay/CORBA service makes a call to the NMS session interface, which forces the OSS client applications to modify their shutdown application. This is not the preferred method.
- The GateWay/CORBA service clears the user session information from its internal memory and database.

6.5 Using Encryption Between the OSS Client and GateWay/CORBA

Prime Optical uses improved encryption of usernames and passwords for network security. You can set the Control Panel to send encrypted usernames and passwords to GateWay/CORBA:

Step 1 Log into the Prime Optical client with administrator privileges.
Step 2 In the Domain Explorer window, choose **Administration > Control Panel**.
Step 3 Click the GateWay/CORBA Service tab for the GateWay/CORBA Service property sheet.
Step 4 Click the **Global** tab and check the **Enable Encryption for Username and Password** check box.
Step 5 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 Prime Optical, use the emsSessionFactory::EmsSessionFactory_I:: getEmsPublicKey API. See 4.5.2 getEmsPublicKey.
- To obtain the RSA public key pair from Prime Optical, use the emsSessionFactory::EmsSessionFactory_I:: getEmsPublicKeyPair API. See 4.5.3 getEmsPublicKeyPair.

Prime Optical uses a 512-bit (64-byte) key size and 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 Prime Optical for login, OSS clients should encode the encrypted username and password by using Base64 encoders to obtain the string equivalent of the encrypted data.

✎ 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. Prime Optical does not provide these cryptographic libraries.

6.6 Using Multiple Naming Servers

Prime Optical can register with multiple naming servers. You must add the following parameters to the Prime Optical-server-installation-directory/cfg/corbagw.properties file:

- corbagw.namingservice.ServerList=ctmc4-u80,ctm7-u60
- corbagw.namingservice.RootIORLoc=/namingroot.ior

Complete the following steps to allow Prime Optical to use multiple naming servers:

Step 1 In the Domain Explorer window, choose **Administration > Control Panel**.
Step 2 Click GateWay/CORBA Service to open the GateWay/CORBA Service pane.
Step 3 In the **Global** tab > **GateWay/CORBA Configuration** area, specify the following parameters:

- Name Service Server List—Lists all the hosts on which the naming service is running. The hosts should be reachable from the Prime Optical server host, and

the HTTP server must be running on all naming service hosts. Enter **ctmc4-u80, ctm7-u60**.

✐ In addition to these naming service hosts, Prime Optical registers itself with the local naming service. The local naming service port is 14005 and is bundled with Prime Optical.

- Name Service Root IOR—Defines the location and name of the file that contains the naming service root Interoperable Object Reference (IOR). The IOR file must be accessible through the following HTTP call: *http://name-server-IP-address:80/namingroot.ior*. Enter **/namingroot.ior**.

Step 4 Restart the GateWay/CORBA service.

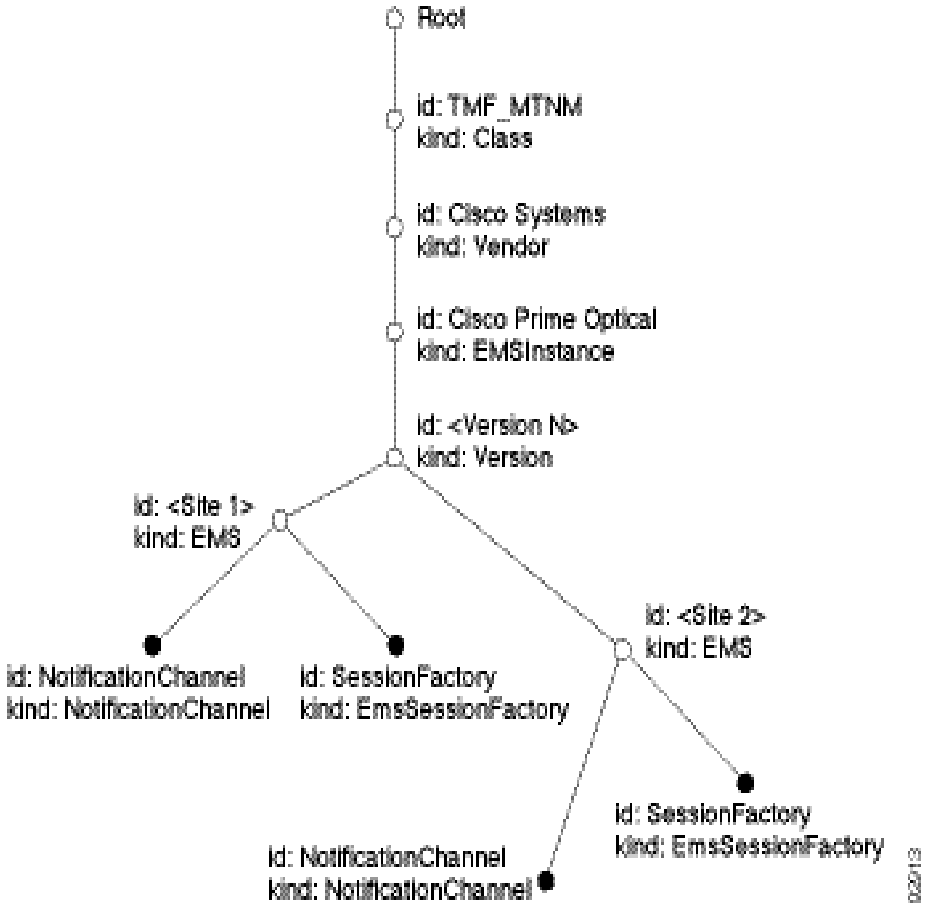
6.7 Naming Conventions for Published GateWay/CORBA Objects

GateWay/CORBA publishes two top-level objects: EMSSessionFactory and NotificationChannel. Prime Optical creates these objects and registers them with the CORBA name server.
GateWay/CORBA creates naming contexts under the root as shown in Figure 6-1. The last context in the tree must have a different name. To change this value in the Prime Optical client GUI:

Step 1 Log into the Prime Optical client with the appropriate Prime Optical user access profile.
Step 2 In the Domain Explorer window, click the **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 *CTM*. By using different names, you can install multiple instances of Prime Optical and use a centralized naming server and repository.

The following figure shows the naming scheme for GateWay/CORBA objects.

Figure 11: Naming Scheme for GateWay/CORBA Objects



6.8 Location of the Naming Service IOR File

The naming service IOR is located at:
/opt/CiscoTransportManagerServer/openfusion/domains/OpenFusion/localhost/NameService/NameSingleton/NameSingleton.ior

6.9 Useful Debugging Utilities for Resolving Naming Service-Related Issues

The following are samples of Prime Optical commands (bundled utility programs) for debugging naming service connectivity issues.

Obtain the list of registered objects in the OpenFusion naming service:

```
setenv PATH /opt/CiscoTransportManagerServer/openfusion/bin:$PATH

setenv NS_IOR_LOCATION
file:///opt/CiscoTransportManagerServer/openfusion/domains/OpenFusion/localhost/NameService/NameSingleton/NameSingleton.ior

nsMgrTool -l
```

Decode an IOR file:

```
setenv PATH /opt/CiscoTransportManagerServer/openfusion/bin:$PATH

dior -f <IOR file name>, or

dior -i <IOR string>
```

Check if the naming service is running:

```
setenv PATH /opt/CiscoTransportManagerServer/openfusion/bin:$PATH
server -status NameService
```

✎ The /opt/CiscoTransportManagerServer/openfusion/bin directory contains nsMgrTool, dior, and server utility tools.

6.10 Configuring GateWay/CORBA

You can configure the following GateWay/CORBA properties in the Prime Optical client Control Panel.

Step 1 From the Domain Explorer, choose Administration > Control Panel.

Step 2 In the Control Panel, click GateWay/CORBA Service. Configure the following properties:

✎ If GateWay/CORBA is running, changes to the config file do not take effect dynamically. You must restart GateWay/CORBA for the changes to take effect.

- Enable Encryption for username and password:
- This property defines whether to encrypt the username and password used for the GateWay/CORBA client.
- Heartbeat for Notification Channel (min): 0
- This property is the rate at which the notification service is checked. A zero entry means not to check the notification service.
- Enter the maximum number of simultaneous sessions: 4
- This property is the number of GateWay/CORBA sessions that can be active at the same time. The range is from 4 to 25 sessions.
- Enter the maximum events per consumer: 10000
- 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.

  💣 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 Prime Optical notification server will run out of memory. The current value is set to handle alarm bursts of 10,000 events per minute.

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

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

- Enter the notification service naming context: services/NotifyChannelFactory

NamingContext defines the naming context of NotificationService. This property is used when resolve_initial_reference fails to resolve NotificationService. GateWay/CORBA contacts the naming service to resolve the name context defined in this property. The value of this property must match the value published by your notification server.

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

- Enter the notification service factory IOR filename: file:/opt/CiscoTransportManagerServer/openfusion/domains/OpenFusion/localhost/NotificationService/NotificationSingleton/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. 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.

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

- Enter the notification service listening port number: 0
  This property is used to set the port that the notification service uses to listen for incoming requests. The port number is set in the IOR for the notification service. The use IOR and use IOR endpoint properties are set correctly. The default port number is zero, which signifies the port number allocated by the operating system.
- Enter the session port number: 0
  This property configures the EMSSessionFactory port. If this property is set to zero, the operating system allocates the session port number.
- Enter the name service server list:
  This property defines where the name servers are running. This property accepts a comma-separated list of hostnames.
- Enter the name service root IOR:
  This 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-ServerListRootIORLoc*
- Error level: Minor
  This property defines the error level of messages to log.
  You can configure this GateWay/CORBA property by modifying a configuration file in *Prime Optical-server-installation-directory/cfg/corbagw.properties*.
- corbagw.CTP.getLayeredParameters=false
  By default, this property is not enabled. If the NMS requires CTP-related transmission parameters to be included as part of an object reporting TerminationPoint_T structure, this property must be set to true. However, the ManagedElementMgr_I.getTP interface always returns transmission parameters as part of the TerminationPoint_T structure and is independent of this property setting.

6.11 Using the CLI to Start and Stop GateWay/CORBA

Prime Optical can manage the GateWay/CORBA service from the command line:

- To start a GateWay/CORBA service, run the /opt/CiscoTransportManagerServer/bin/gwcorba-start script from the command line.

- To stop a GateWay/CORBA service, run the /opt/CiscoTransportManagerServer/bin/gwcorba-stop script from the command line.

Only Prime Optical users with administrative privileges can run these scripts. If the GateWay/CORBA service is already running and you attempt to run the gw-start script, the script exits with the message "GWCORBA already running." If the GateWay/CORBA service is stopped and you attempt to run the gw-stop script, the script exits with the message "GWCORBA not running."
You must have a Prime Optical username and password with a SysAdmin or SuperUser profile to start or stop the scripts.

6.12 Configuring Secure Socket Layer for GateWay/CORBA

To ensure network security, CORBA calls can be made over Secure Socket Layer (SSL).
The current JacORB implementation is precompiled with JacORB security libraries. To configure SSL for GateWay/CORBA, you must set up a keystore and configure the properties in the client-side jacorb.properties file.
The client must enforce SSL by modifying the jacorb.properties file. The server-side keystore is generated using the JSSE keystore. Prime Optical bundles a default keystore and a certificate for the GateWay/CORBA service.
As explained in the following sections, you must generate the server-side certificate and add it to the client-side keystore; then generate and add the client-side certificate to the server-side keystore.

6.12.1 Generating the Server-Side Certificate

Step 1 Enter the keytool command to generate a keystore and a key:

```
keytool -genkey -alias gwcorba_service -validity 25000 -keystore
gwcorba_service_ks -storepass gwcorba_service_ks_pass -keypass
gwcorba_service_ks_pass
What is your first and last name?
  [Unknown]:  gateway corba server
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:  cisco
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is <CN=gateway corba server, OU=Unknown, O=cisco, L=Unknown,
ST=Unknown, C=Unknown> correct?
  [no]:  y
```

Step 2 Verify that the generated keystore and key have the following attributes:

```
Keystore name: gwcorba_service_ks
Alias: gwcorba_service
```

```
Keystore password: gwcorba_service_ks_pass
Key password: gwcorba_service_ks_pass
Validity: 25000 days
```

Step 3 Enter the following command to generate a server-side certificate that will be issued to the client:

```
keytool -export -keystore gwcorba_service_ks -alias gwcorba_service
-storepass gwcorba_service_ks_pass -file gwcorba_service_cert
Certificate stored in file <gwcorba_service_cert>
```

Step 4 Verify that the certificate is stored in the gwcorba_service_cert file. The server-side certificate and keystore are present in the /opt/CiscoTransportManagerServer/cfg directory.

6.12.2 Generating the Client-Side Certificate

Note the following conventions:

- ascii_client_ks—Denotes a client-side keystore.
- ascii_client_cert—Denotes a client-side certificate.
- gwcorba_service_ks—Denotes a server-side keystore.
- gwcorba_service_cert—Denotes a server-side certificate.

Step 1 Enter the keytool command to generate a keystore:

```
keytool -genkey -alias ascii_client -validity 25000 -keystore
ascii_client_ks -storepass ascii_client_ks_pass -keypass
ascii_client_ks_pass
What is your first and last name?
  [Unknown]:  ascii client
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:  cisco
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=ascii client, OU=Unknown, O=cisco, L=Unknown, ST=Unknown,
C=Unknown correct?
[no]:  y
```

Step 2 Verify that the generated keystore and key have the following attributes:

```
Keystore name: ascii_client_ks
Alias: ascii_client
```

```
Keystore password: ascii_client_ks_pass
Key password: ascii_client_ks_pass
Validity: 25000 days
```

Step 3 Enter the following command to generate a client-side certificate that will be issued to the server:

```
keytool -export -keystore ascii_client_ks -alias ascii_client -
storepass ascii_client_ks_pass -file ascii_client_cert
Certificate stored in file <ascii_client_cert>
```

Step 4 Verify that the certificate is stored in the ascii_client_cert file.

6.12.3 Adding the Client-Side Certificate to the Server-Side Keystore

Step 1 Enter the following command to add the client-side certificate to the server-side keystore. (Use FTP or a similar tool to deliver the ascii_client_cert file to the server. The server-side keystore is located in the /opt/CiscoTransportManagerServer/cfg directory on the server.)

```
keytool -import -keystore gwcorba_service_ks -alias ascii_client -
storepass gwcorba_service_ks_pass -file ascii_client_cert
```

The command output resembles the following example:

```
Owner: CN=ascii client, OU=Unknown, O=cisco, L=Unknown, ST=Unknown,
C=Unknown
Issuer: CN=ascii client, OU=Unknown, O=cisco, L=Unknown, ST=Unknown,
C=Unknown
Serial number: serial-number
Valid from: Sat Mar 18 17:18:44 GMT+05:30 2008 until: Fri Jul 23
10:50:28 GMT+05:30 2038
Certificate fingerprints:
MD5: 42:53:98:7C:BA:CB:28:39:50:50:9F:E4:56:F2:43:FF
SHA1: F5:1D:B9:BB:1D:66:C2:A1:32:BE:47:0C:85:47:17:16:A2:69:17:4C
Trust this certificate? [no]: y
Certificate was added to keystore
```

Step 2 Verify that the certificate issued by the client was added to the server keystore.

6.12.4 Adding the Server-Side Certificate to the Client-Side Keystore

Step 1 Enter the following command to add the server-side certificate to the client-side keystore. (Use FTP or a similar tool to deliver the previously generated gwcorba_service_cert file from the server. The server-side certificate is located in the /opt/CiscoTransportManagerServer/cfg directory on the server.)

```
keytool -import -keystore ascii_client_ks -alias gwcorba_service -
storepass ascii_client_ks_pass -file gwcorba_service_cert
```

The command output resembles the following example:

```
Owner: CN=gateway corba server, OU=Unknown, O=cisco, L=Unknown,
ST=Unknown, C=Unknown

Issuer: CN=gateway corba server, OU=Unknown, O=cisco, L=Unknown,
ST=Unknown, C=Unknown

Serial number: serial-number

Valid from: Sat Mar 18 17:21:24 GMT+05:30 2008 until: Fri Jul 23
10:53:08 GMT+05:30 2038

Certificate fingerprints:

MD5: 5C:41:39:AD:D0:F8:63:5D:81:8D:47:A0:33:02:8E:7D

SHA1: 38:CD:C8:57:F7:15:22:DC:1A:6E:99:CD:13:A1:9A:67:90:2C:65:C2

Trust this certificate? [no]: y

Certificate was added to keystore
```

Step 2 Verify that the certificate issued by the server was added to the client keystore.

6.12.5 Configuring the Client-Side Properties

To enforce SSL, complete the following configuration on the client-side jacorb.properties file.

- To enforce SSL, the supported and required options must be set to 60.
- No changes are required to the server-side jacorb.properties file, because it has already been changed.

```
jacorb.security.support_ssl=on
# IIOP/SSL parameters (numbers are decimal values):
# EstablishTrustInClient = 40
# EstablishTrustInTarget = 20
# mutual authentication = 60
jacorb.security.ssl.client.supported_options=60
jacorb.security.ssl.client.required_options=60

jacorb.ssl.socket_factory=org.jacorb.security.ssl.sun_jsse.SSLSocketFactory

jacorb.security.keystore_password= ascii_client_ks_pass
jacorb.security.keystore= ascii_client_ks

# Read trusted certificates from the keystore
jacorb.security.jsse.trustees_from_ks=on
```

6.13 Installation Program

The Prime Optical installation program installs the GateWay/CORBA component, which includes OpenFusion 4.2.3 (JacORB, Notification Service, and Name Service) from Prism Technologies. Interface Definition Language (IDL) files are installed under the /opt/CiscoTransportManagerServer/idl directory. See the Cisco Prime Optical 10.7 Installation Guide for more information.

6.14 Cisco Prime Optical 9.8-to-Cisco Prime Optical 10.7 Migration

All procedures related to the migration from Cisco Prime Optical 9.6.3 to Cisco Prime Optical 10.7 are performed during the installation phase; no additional operations are required.

7 OSS Use Cases and Client Development

This chapter lists interceptors and use cases to resynchronize the NMS from the EMS. This chapter includes the following sections:

7.1 Use Cases to Resynchronize the NMS from the EMS

Table 7-1, Table 7-2, and Table 7-3 list use cases to resynchronize the NMS from the EMS.

Table 7-1: Use Cases to Resynchronize the NMS from the EMS

Information	Detail
Name	The NMS retrieves Cisco Prime Optical information.
Summary	The NMS retrieves all EMS and ME alarms and information for existing multilayer subnetworks, MEs, PTPs, CTPs, SNCs, and protection groups from Cisco Prime Optical GateWay/CORBA interfaces.
Actor(s)	NMS.
Preconditions	The NMS locates the EmsSessionFactory object and obtains references to EMSMgr_I, MultiLayerSubnetworkMgr_I, ManagedElementMgr_I, EquipmentInventoryMgr_I, MaintenanceMgr_I, ProtectionMgr_I, and PerformanceManagementMgr_I interfaces.
Begins When	The NMS sends a request to retrieve all of the Prime Optical information.
Description	The NMS does the following to request information from the GateWay/CORBA interface: 1. Uses the emsMgr::EMSMgr_I:: getAllEMSSystemActiveAlarms interface to request all EMS alarms. 2. Uses the emsMgr::EMSMgr_I:: getAllTopLevelSubnetworks interface to request all multilayer subnetworks in Prime Optical. • Prime Optical returns a list of MultiLayerSubnetwork_T objects with detailed information about a multilayer subnetwork. 3. Uses the emsMgr::EMSMgr_I:: getAllTopLevelTopologicalLinks interface to request all topological links across multiple multilayer subnetworks. • Prime Optical returns a list of TopologicalLink_T objects with detailed information about a topological link. 4. For each MultiLayerSubnetwork_T object, Prime Optical uses the multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllManagedElements interface to request all MEs that belong to that multilayer subnetwork.

Information	Detail
	• Prime Optical returns a list of ManagedElement_T objects with detailed information about an ME. • For each ManagedElement object, executes the use case "NMS Resynchronizes Information Specific to a Managed Element." 5. Uses the multiLayerSubnetwork::MultiLayerSubnetworkMgr_I:: getAllVLANs interface to request all VLANs associated with the multilayer subnetwork.
Ends When	The NMS retrieves all Prime Optical information.
Exceptions	Refer to the exceptions thrown by the individual interface method.
Post Conditions	• The NMS synchronizes with Prime Optical. • The NMS registers with Prime Optical to retrieve notifications related to changes in the managed object on Prime Optical, new alarms, and TCAs.
Traceability	—

Table 7-2: NMS Resynchronizes Information Specific to a Managed Element

Information	Detail
Name	The NMS retrieves Prime Optical information specific to the ME.
Summary	The NMS retrieves all PTP, CTP, SNC, and protection group information for the specific ME.
Actor(s)	NMS.
Preconditions	The NMS obtains references to ManagedElementMgr_I, EquipmentInventoryMgr_I, MaintenanceMgr_I, ProtectionMgr_I, and PerformanceManagementMgr_I interfaces. The NMS also obtains the ME 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 to retrieve information.
Description	The NMS does the following to request information from the GateWay/CORBA interface: 1. Uses the managedElementManager::ManagedElementManager_I:: getAllActiveAlarms interface to request all current alarms on the ME. 2. Uses the equipment::EquipmentInventoryMgr_I:: getAllEquipment interface to request all existing equipment on the ME. • Prime Optical returns a list of EquipmentHolder_T objects with detailed information about equipment holders and equipment on the ME. 3. For the equipment information retrieved in

Information	Detail
	EquipmentorHolder_T object, uses the equipment::EquipmentInventoryMgr_I:: getAllSupportedPTPs interface to request all supported PTPs on the equipment. • Prime Optical returns a list of TerminationPoint_T objects with detailed PTP information, such as SDH or SONET port, admin state, service state, line code, frame format, and so on. 4. For each TerminationPoint_T object, executes the "NMS Resynchronizes Information Specific to TerminationPoint" use case. 5. Uses the managedElementManager::ManagedElementManager_I:: getAllSNCs interface to request all SNCs originating, terminating, or passing through the ME. • Prime Optical returns a list of SubnetworkConnection_T objects with detailed SNC information, such as layer rate, SNC state, SNC protection state, a-end CTP, z-end CTP, and so on. 6. Uses the protection::ProtectionMgr_I:: getAllProtectionGroups interface to request all protection groups created on the ME. • Prime Optical returns a list of ProtectionGroup_T objects with detailed protection group information, such as 1:1, 1+1, or 2F-BLSR; revertive or nonrevertive; and so on.
Ends When	The NMS retrieves all ME-specific information.
Exceptions	Refer to the exceptions thrown by the individual interface method.
Post Conditions	The NMS synchronizes information about an ME with Prime Optical.
Traceability	—

Table 7-3: NMS Resynchronizes Information Specific to a Termination Point (continued)

Information	Detail
Name	The NMS retrieves Prime Optical information specific to the termination point.
Summary	The NMS retrieves all loopback and threshold information for PTPs and CTPs used by the SNC.
Actor(s)	NMS.
Preconditions	The NMS obtains references to ManagedElementMgr_I, MaintenanceMgr_I, and PerformanceManagementMgr_I interfaces. The NMS also obtains a list of PTPs.
Begins When	The NMS sends a request to retrieve all threshold and loopback information on a PTP and on any in-use CTPs. The NMS identifies the name and type (PTP/CTP) of the

Information	Detail
	TP for which to retrieve information.

Description	The NMS does the following to request information from the GateWay/CORBA interface:
	• Uses the maintenanceOps::MaintenanceMgr_I:: getActiveMaintenanceOperations interface to request all loopback information on the TP.
	  • Prime Optical returns a list of CurrentMaintenanceOperation_T objects with detailed information about the type of loopback set on the PTP, if any. Prime Optical supports two types of loopback, FACILITY_LOOPBACK and TERMINAL_LOOPBACK. If there are no loopbacks set on the TP, the returned list is empty.
	• Uses the performance::PerformanceManagementMgr_I:: getTCATPParameter interface to request the threshold values set for all PM thresholds on the given TP object.
	  • Prime Optical returns the TCAParameter_T object with detailed information about all PM threshold name/value pairs for the PTP.
	• For each PTP that the NMS specifies as the TP, uses the managedElementManager::ManagedElementManager_I:: getContainedInUseTPs interface to request all CTPs contained in the PTP participating in an SNC.
	  • Prime Optical returns a list of TerminationPoint_T objects with detailed information about the CTPs participating in an SNC and the associated provisioning details, such as IPPM monitor status, J1 path trace status, and so on.
	• For each CTP object, uses the performance::PerformanceManagementMgr_I:: getTCATPParameter interface to request threshold information.
	  • Prime Optical returns a list of TCAParameter_T objects with detailed information about the PM threshold name/value pairs for the CTP.

Ends When	The NMS retrieves loopback and threshold information for all PTPs and CTPs.
Exceptions	Refer to the exceptions thrown by the individual interface method.
Post Conditions	The NMS synchronizes information about a TP object

Information	Detail
	with Prime Optical.
Traceability	—

7.2 Developing a GateWay/CORBA Client

GateWay/CORBA implements 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.

7.2.1 Recommended Tools

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

- Oracle Java 1.7 (available at http://www.oracle.com/technetwork/java/index.html)
- jacORB 2.x

7.2.2 Sample Code in Java

A typical CORBA client application involves the following steps:

Step 1 Initialize the client connection to the object request broker (ORB).
Step 2 Obtain a reference to the naming service.
Step 3 Resolve the reference to EMSSessionFactory. See 7.2.2.3 Get Reference to EMSSessionFactory.
Step 4 Implement NmsSession_IOperations. See 7.2.2.4 Implement NmsSession_IOperations.
Step 5 Retrieve EmsSession by supplying the username and password. See 7.2.2.5 Log In and Retrieve EmsSession.
Step 6 Query EmsSession and obtain a list of managers available for operations. See 7.2.2.6 Retrieve List of Managers.
Step 7 Invoke the desired method on that manager. See 7.2.2.7 getEMS Operation on EMS Manager.
If you want your client to receive notifications from GateWay/CORBA, the following additional steps are required:
Step 8 Obtain a reference to an EventChannel object in the notification server in Step 2.
Step 9 Obtain a ConsumerAdmin object reference.
Step 10 Invoke obtain_notification_push_consumer() on the SupplierAdmin object, specifying CosNotifyChannelAdmin::STRUCTURED_EVENT as a parameter. This operation returns a reference to the StructuredProxyPushSupplier.
Step 11 Implement an instance of StructuredPushConsumer (defined by the OMG).
Step 12 Invoke the connect_structured_push_consumer() operation on the StructuredProxyPushSupplier object to connect the StructuredPushConsumer implementation object to the notification channel.
Step 13 Monitor incoming notifications.

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

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

7.2.2.1 Initialize the Client Connection

```
// Import OMG packages
```

```
import org.omg.CORBA.IntHolder;
import org.omg.CORBA.ORB;
import org.omg.CORBA.Policy;

// Import naming context packages.
import org.omg.CosNaming.NameComponent;
import org.omg.CosNaming.NamingContextExt;
import org.omg.CosNaming.NamingContextExtHelper;

// Import notify channel and event service packages.
import org.omg.CosEventChannelAdmin.AlreadyConnected;
import org.omg.CosEventChannelAdmin.TypeError;
import org.omg.CosNotifyChannelAdmin.AdminLimitExceeded;
import org.omg.CosNotifyChannelAdmin.AdminNotFound;
import org.omg.CosNotifyChannelAdmin.ClientType;
import org.omg.CosNotifyChannelAdmin.ConsumerAdmin;
import org.omg.CosNotifyChannelAdmin.EventChannel;
import org.omg.CosNotifyChannelAdmin.EventChannelHolder;
import org.omg.CosNotifyChannelAdmin.ProxySupplier;
import org.omg.CosNotifyChannelAdmin.StructuredProxyPushSupplier;
import org.omg.CosNotifyChannelAdmin.StructuredProxyPushSupplierHelper;
import org.omg.CosNotifyComm.StructuredPushConsumer;
import org.omg.CosNotifyComm.StructuredPushConsumerPOATie;

// Import POA packages
import org.omg.PortableServer.POA;
import org.omg.PortableServer.POAHelper;
import org.omg.PortableServer.POAPackage.ServantAlreadyActive;
import org.omg.PortableServer.POAPackage.WrongPolicy;

// Import TMF packages
import org.tmforum.mtnm.emsSession.EmsSession_I;
import org.tmforum.mtnm.emsSession.EmsSession_IHolder;
import org.tmforum.mtnm.emsSession.EmsSession_IPackage.managerNames_THolder;
import org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_I;
import org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_IHelper;
import org.tmforum.mtnm.nmsSession.NmsSession_I;
import org.tmforum.mtnm.nmsSession.NmsSession_IPOATie;
public static void main(String[] args)
```

```
{

try {
// Optional: set up ORB properties: you can set them here or in the launch
file for the client
// Properties sys_properties = System.getProperties();

// For JacORB
// sys_properties.put("org.omg.CORBA.ORBClass",
"org.jacorb.orb.ORB"
// sys_properties.put("org.omg.CORBA.ORBSingletonClass",
"org.jacorb.orb.ORBSingleton"

// For Orbix
// sys_properties.put("org.omg.CORBA.ORBClass",
"com.iona.corba.art.artimpl.ORBImpl");
// sys_properties.put("org.omg.CORBA.ORBSingletonClass",
"com.iona.corba.art.artimpl.ORBSingleton");

// For Visibroker
// sys_properties.put("org.omg.CORBA.ORBClass",
"com.inprise.vbroker.orb.ORB"
// sys_properties.put("org.omg.CORBA.ORBSingletonClass",
"com.inprise.vbroker.orb.ORBSingleton"

// For other ORBs check the related ORB programming guide

// Step 1: Initialize the ORB and obtain ROOT POA reference
// Note: ORB_init will process any -ORB arguments
// be called before any other argument processing.
//
global_orb = ORB.init(args, null);
org.omg.CORBA.Object root_poa =
global_orb.resolve_initial_references("RootPOA");
POA rpoa = POAHelper.narrow(root_poa);
POA poa = rpoa.create_POA("myPolicy",null, new Policy[0]);
poa.the_POAManager().activate();
}
catch (SystemException ex) {
// Exception handling
```

```
}
}
```

7.2.2.2 Get Reference to the Naming Service

```
// Get Nameservice reference
NamingContext nsRootContext = null;
try {
// Step 2: Get reference to the name service
// Option 1: Resolve initial reference (RIR)
// org.omg.CORBA.Object obj =
global_orb.resolve_initial_references("NameService");
// or

 // Option 2: corbaloc URL
 String objRef = "corbaloc:iiop:gatewayserver.cisco.com:14005/NameServiceGWC";
org.omg.CORBA.Object obj = global_orb.string_to_object(objRef);
/*  NOTE: Please replace "gatewayserver.cisco.com" with the name of the server
on which CORBA Naming service is running. */
// Narrow to root naming context
NamingContextExt root_context = NamingContextExtHelper.narrow(obj);

}
catch (org.omg.CORBA.ORBPackage.InvalidName inEx) {
// Exception handling
}
```

7.2.2.3 Get Reference to EMSSessionFactory

Follow the example in 7.2.2.2 Get Reference to the Naming Service to obtain a reference to the naming service.

```
NameComponent name = new NameComponent[6];
name[0] = new NameComponent("TMF_MTNM", "Class");
name[1] = new NameComponent("Cisco Systems", "Vendor");
name[2] = new NameComponent("Cisco Prime Optical", "EMSInstance");
name[3] = new NameComponent(version, "Version"); //where version = "9_5" for
Prime Optical 9.5
name[4] = new NameComponent(ctm_sys_id, "EMS"); // ctm_sys_id = "PrimeOptical"
name[5] = new NameComponent("SessionFactory", "EmsSessionFactory");
try {
org.omg.CORBA.Object emsSessionI = root_context.resolve(name);
}
catch (InvalidName inEx) {
// Exception handling
}
```

```
catch (NotFound nfEx) {
// Exception handling
}
```

7.2.2.4 Implement NmsSession_IOperations

```
import org.tmforum.mtnm.session.*;
import org.tmforum.mtnm.nmsSession.*;

public class SessionImpl implements NmsSession_IOperations {


      Session_I myAssociatedSession = null;

      public SessionImpl() {
            super();
            // TODO Auto-generated constructor stub
      }

      public void setAssociatedSession(Session_I emsSession) {
        myAssociatedSession = emsSession;
    }
      public Session_I getAssoicatedSession (){ return myAssociatedSession;}

      public void eventLossOccurred(String startTime, String notificationId) {
            // TODO Auto-generated method stub

      }
      public void eventLossCleared(String endTime) {
            // TODO Auto-generated method stub

      }

      public void historyPMDataCompleted(String fileName) {
            // TODO Auto-generated method stub

      }

      public void historyPMDataFailed(String errorReason) {
            // TODO Auto-generated method stub
```

```
        }
        public Session_I associatedSession() {
                // TODO Auto-generated method stub
                return null;
        }


        public void ping() {
                // TODO Auto-generated method stub

        }

        public void endSession() {
                // TODO Auto-generated method stub

        }
}
}
```

7.2.2.5 Log In and Retrieve EmsSession

To perform operations from GateWay/CORBA, your client must log in using a username and password created on the Prime Optical client. See 6.1 Creating an OSS Client Profile for GateWay/CORBA.

```
EmsSession_I m_emsSession = null;
SessionImpl mySessionImpl = new SessionImpl();
try {
EmsSessionFactory_I ems_ref = EmsSessionFactory_IHelper.narrow(emsSessionI);
EmsSession_IHolder emsSessionHldr = new EmsSession_IHolder();
NmsSession_IPOATie tieobj = new NmsSession_IPOATie(mySessionImpl, poa);
poa.activate_object(tieobj);
NmsSession_I nmsSession_ref = tieobj._this();
if (ems_ref != null) {
ems_ref.getEmsSession(user, password, nmsSession_ref, emsSessionHldr);
m_emsSession = emsSessionHldr.value;
}

} catch (Exception ex) {
// System.out.println("Could not narrow");
ex.printStackTrace();
}
```

7.2.2.6 Retrieve List of Managers

```
managerNames_THolder names = new managerNames_THolder();
m_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]);
}
```

7.2.2.7 getEMS Operation on EMS Manager

```
EMS_T m_ems;
EMS_THolder m_emsHolder = new EMS_THolder();
try {
Common_IHolder mgrHolder = new Common_IHolder();
m_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);
```

7.2.2.8 Get Reference to EventChannel

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

7.2.2.9 Obtain ConsumerAdmin Reference

```
//retrieve default consumer admin
try {
```

```
ConsumerAdmin cadmin = notifChannel.get_consumeradmin(0);
}
catch (AdminNotFound anfSe) {
// Exception handling
}
```

7.2.2.10 Obtain ProxyPushSupplier

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

7.2.2.11 Implement StructuredPushConsumer

```
class StructuredPushConsumerImpl extends _StructuredPushConsumerPOA
{
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.");
}
}
```

7.2.2.12 Connect StructuredPushConsumerImpl

```
try {
```

```
StructuredPushConsumerImpl structProxyPushConsumer = new
StructuredPushConsumerImpl();
StructuredPushConsumerPOATie structuredPushConsumerTieObj = new
StructuredPushConsumerPOATie (structProxyPushConsumer, poa);
     poa.activate_object(structuredPushConsumerTieObj);
     StructuredPushConsumer pushCon = structuredPushConsumerTieObj._this();
     structuredProxyPushSupplier.connect_structured_push_consumer(pushCon);

global_orb.run();
}
catch (ServantAlreadyActive sae) {
// Exception handling
}
catch (WrongPolicy wrongPolicyEx) {
// Exception handling
}
```

7.2.3 Running the Client

If the initial naming context is not resolved using the Option 2 method shown in 7.2.2.2 Get Reference to the Naming Service, the following JVM flags must be used for CORBA clients written for Orbix 6.2, Visibroker, or jacORB:

```
Orbix 6.2 or later:
-ORBInitRef NameService=corbaloc:iiop:1.2@<host name>:<port>/NameServiceGWC
Visibroker:
-DORBInitRef NameService=corbaloc::<host name>:<port>/NameServiceGWC
-DORBInitRef NameService=corbaname::<host name>:<port>/NameServiceGWC
jacORB:
-DORBInitRef.NameService=corbaname::<host name>:<port>/NameServiceGWC
```

In addition, if you do not set the system properties org.omg.CORBA.ORBClass and org.omg.CORBA.ORBSingletonClass in your client code before Step 1 in 7.2.2.1 Initialize the Client Connection, you must set them in your client launch file:

```
Orbix 6.2 or later:
-Dorg.omg.CORBA.ORBClass=com.iona.corba.art.artimpl.ORBImpl
-Dorg.omg.CORBA.ORBSingletonClass=com.iona.corba.art.artimpl.ORBSingleton
Visibroker:
-Dorg.omg.CORBA.ORBClass=com.inprise.vbroker.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=com.inprise.vbroker.orb.ORBSingleton
jacORB:
-Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton
```

For other ORBs, check the related ORB programming guide.

7.2.4 Known Interoperability Issues When Using Orbix

If you use Orbix to set the following property in the jacorb.properties file:
Jacorb.compactTypecodes=0

Pay attention to the compatibility of the domain you use with Orbix. If you use an insecure domain, set the following property in the jacorb.properties file:
Jacorb.security.support_ssl = off

8 Equipment List for NEs

This chapter lists the equipment that Cisco Prime Optical GateWay/CORBA reports for each NE type.
This chapter contains the following sections:

8.1 Equipment List for ONS 15216

The equipment is the managed element itself for the following ONS 15216 NEs: ONS 15216 terminal filters, ONS 15216 OADMs, ONS 15216 EDFA, and ONS 15216 OSC.
The following table lists the ONS 15216 equipment that Prime Optical 10.7 supports.

Table 8-1: Prime Optical-Supported ONS 15216 Equipment

Card Type	Equipment
DCU	• DCU-100 (100 ps/nm) • DCU-1150 (1150 ps/nm) • DCU-350 (350 ps/nm) • DCU-450 (450 ps/nm) • DCU-550 (550 ps/nm) • DCU-750 (750 ps/nm) • DCU-950 (950 ps/nm) • DCU-DS-L-100 (100 ps/nm) • DCU-DS-L-200 (200 ps/nm) • DCU-DS-L-300 (300 ps/nm) • DCU-E-200 (200 ps/nm, ELEAF type) • DCU-E-350 (350 ps/nm, ELEAF type) • DCU-L-1000 (1000 ps/nm) • DCU-L-1100 (1100 ps/nm)

Card Type	Equipment
	• DCU-L-300 (300 ps/nm) • DCU-L-600 (600 ps/nm) • DCU-L-700 (700 ps/nm) • DCU-L-800 (800 ps/nm)
FLA	FLA-8-channel-ID, where channel-ID is 36.6, 44.5, 52.5, or 60.6
FLB	FLB-2-channel-ID, where channel-ID is 31.1, 32.6, 35.0, 36.6, 38.9, 40.5, 42.9, 44.5, 46.9, 48.5, 50.9, 52.5, 54.9, 56.5, 58.9, or 60.6 • CS-2 • CS-3 • CS-4 • CS-MM-Y • CS-SM-Y
SC	SC-4B
VOA	VOA-4

8.2 Equipment List for ONS 15305

The following table lists the ONS 15305 equipment that Prime Optical 10.7 supports.

Table 8-2: Prime Optical-Supported ONS 15305 Equipment

Card Type	Equipment
E	• E100-8 • E3T3-6 • E1-8 • E1-21 • E1-63
GE	• GE-1-LC • GE-2-LC
S	• S1.1-8-LC • S4.1-2-LC • S4.1-4-LC • S16.1-1-LC • S1.1-2-LC

8.3 Equipment List for ONS 15305 CTC

The following table lists the ONS 15305 CTC equipment that Prime Optical 10.7 supports.

Table 8-3: Prime Optical-Supported ONS 15305 CTC Equipment

Card Type	Equipment
E	• E1-8
GE	GigE-WAN-2
S	• S1.1-2-LC • E3T3-6 • E100-8 • L1.2-2-LC

Card Type	Equipment
	• S1.1-8-LC • S16.1-1-LC • S4.1-2-LC • L16.1-1-LC • S4.1-2-LC

8.4 Equipment List for ONS 15310 CL

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15310 CL user documentation for actual equipment names used by the NE.

Table 8-4: Prime Optical-Supported ONS 15310 CL Equipment

Card Type	Equipment
ML	• ML2_L2L3 • ML2_MAPPER
PPM	PPM_1_PORT
XC	CTX

8.5 Equipment List for ONS 15310 MA SDH

The following table lists the ONS 15310 MA SDH equipment that Prime Optical 10.7 supports.

Table 8-5: Prime Optical-Supported ONS 15310 MA SDH Equipment

Card Type	Equipment
DS/E	• E1_21_E3_DS3_3 • E1_63_E3_DS3_3
Data Cards	• ML-100T-8 • CE-100T-8 • CE-MR-6
Panel Cards	FILLER

8.6 Equipment List for ONS 15310 MA SONET

The following table lists the ONS 15310 MA SONET equipment that Prime Optical 10.7 supports.

Table 8-6: Prime Optical-Supported ONS 15310 MA SONET Equipment

Card Type	Equipment
Controller	CTX-2500
Data Cards	ML2-100T-8
DS Cards	• DS1-28/DS3-EC1-3 • DS1-84/DS3-EC1-3
Fillers	• CTX-FILLER • EXP-FILLER
PPM	PPM_1_PORT

8.7 Equipment List for ONS 15327

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15327 user documentation for actual equipment names used by the NE.

Table 8-7: Prime Optical-Supported ONS 15327 Equipment

Card Type	Equipment
ETH	ETH100
G	G1000_2
OC	• OC12 • OC3 • OC48
MIC	XTC
XT	MIC

8.8 Equipment List for ONS 15454 SDH

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15454 SDH user documentation for actual equipment names used by the NE.

Table 8-8: Prime Optical-Supported ONS 15454 SDH Equipment

Card Type	Equipment
AD	AD_1B AD_1C AD_2C AD_4B AD_4C
ADM	ADM_10G
Aux	AIC AICI AEP AIE MMU PSM ALM_PWR CRFT_TMG
CE	CE-100T-8 CE-1000-4 CE-MR-10
Controller	TCC TCC2 TCC2P TCC3
DS	DS3I E1_42 E3
ETH	ETH100 ETH1000
FMEC	FMEC_SMZ_E1

Card Type	Equipment
G	G1000_4
ML	ML100T ML1000 ML-MR-10 ML100X-8
MR	MRC-12 MRC25G_4 FCMR
MSC	MS-ISC-100T
MXP	MXP_2.5G_10G MXP_2.5G_10E 40G_MXP_C 40E_MXP_C 10_DME
MUX/DMX	40 DMX 40 MUX 32DMX 32DMX_L 32MUX_O 32_DMX_O
OPT	OPT_BST OPT_PRE OPT_BST_E OPT_BST_L OPT_AML_L OPT_AMP_17_C OPT_AMP_C OPT_RAMP_C OPT_RAMP_CE OPT_EDFA_17 OPT_EDFA_24
OSC	OSC_CSM OSCM
SMR	40-SMR1-C 40-SMR2-C
STM	STM1E_12 STM1 STM1_8 STM4 STM4_4 STM16 STM64 STM64_XFP
TDCU	TDCU-C TDCU-F
TXP	• TXP_MR_10G • TXP_MR_2.5G • TXPP_MR_2.5G

Card Type	Equipment
	• TXP_MR_10E • 2.5G_DM 2.5G_DMP
WSS	32WSS
WXC	40WXC
XC	• XC • XCVT • XC10G • XCVXL_10G • XCVXC • XCVXC25G
XP	• OTU2_XP • 10GE_XP • GE_XP

8.9 Equipment List for ONS 15454 SONET

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15454 SONET user documentation for actual equipment names used by the NE.

Table 8-9: Prime Optical-Supported ONS 15454 SONET Equipment

Card Type	Equipment
AD	• AD_1B • AD_1C • AD_2C • AD_4B • AD_4C
ADM	ADM_10G
Aux	• AIC • AICI • AEP • AIE • MMU • PSM • ALM_PWR • CRFT_TMG
CE	• CE-100T-8 • CE-1000-4 • CE-MR-10
Controller	• TCC • TCC2 • TCC2P • TCC3
DS	• DS1 • DS1N • DS1-63

Card Type	Equipment
	• DS3 • DS3N • DS3E • DS3NE • DS3XM • DS3XM_12 • DS3I • DS3IN • DS3_EC1_48 • DS1_E1_56 • EC1
ETH	• ETH100 • ETH1000 • FMEC FMEC_SMZ_E1 • FMEC_SMZ_E3 • FMEC_DB • FMEC_SMZ_DS1I
G	G1000_4
ML	• ML100T • ML1000 • ML-MR-10 • ML100X-8
MR	• MRC-12 • MRC25G_4 • FCMR
MSC	MS-ISC-100T
MXP	• MXP_2.5G_10G • MXP_2.5G_10E • 40G_MXP_C • 40E_MXP_C • 10_DME
MUX/DMX	40 DMX
OC	• OC3 • OC3_8 • OC12 • OC12_4 • OC48 • OC192 • OC192_XFP
OPT	• OPT_BST • OPT_PRE • OPT_BST_E • OPT_BST_L • OPT_AML_L • OPT_AMP_17_C

Card Type	Equipment
	• OPT_AMP_C • OPT_RAMP_C • OPT_RAMP_CE
OSC	• OSC_CSM • OSCM
SMR	40-SMR1-C
TXP	• TXP_MR_10G • 40-SMR2-C • TDCU • TDCU-C • TDCU-F • TXP_MR_2.5G • TXPP_MR_2.5G • TXP_MR_10E • 2.5G_DM 2.5G_DMP • 40E_TXP_C • AR_MXP
WSS	• 32WSS • 32WSS_L • 40WSS
WXC	40WXC
XC	• XC • XCVT • XC10G • XCVXL_10G • XCVXC • XCVXC_10G • XCVXC25G
XP	• OTU2_XP • 10GE_XP • GE_XP • AR_MXP

The following table lists the supported ports for ONS 15454 SONET.

Table 8-10: Prime Optical-Supported ONS 15454 Ports

Port Category	Port Equipment Type
PPM	PPM_1_PORT
OC	• OC3_PORT • OC12_PORT • OC48_PORT • OC192_PORT • OC768_PORT
FC	• FC1G_PORT • FC2G_PORT • FC10G_PORT • FC4G_PORT

Port Category	Port Equipment Type
	• FC8G_PORT
FICON	• FICON1G_PORT • FICON2G_PORT • FICON4G_PORT
GE	• ONE_GE_PORT • TEN_GE_PORT • FORTY_GE_PORT • ONEHUNDRED_GE_PORT
Data/Storage	• DV6000_PORT • ESCON_PORT • SDI_D1_VIDEO_PORT • HDTV_PORT IB_5G_PORT
ISC	• ISC_PORT • ISC3_PORT • ISC3_1G_PORT • ISC3_2G_PORT • ISC3_STP_1G_PORT • ISC3_STP_2G_PORT
GFP	GFP_PORT
OTU	• OTU1_PORT • OTU2_PORT • OTU3_PORT • OTU4_PORT
INTERLINK	INTERLINK_PORT
Other	• PASS_THRU_PORT • ETR_CLO_PORT • SDI_3G_VIDEO_PORT • SD_SDI_PORT • HD_SDI_PORT • AUTO_PORT

8.10 Equipment List for ONS 15454-M2

The ONS 15454-M2 supports all equipment that is supported by the ONS 15454 MSTP system.
The following additional equipment are supported.

Table 8-11: Prime Optical- Supported ONS 15454-M2

Card Type	Equipment
TXP	• M-CFP-LC • 100G-LC-C • 100G-CK-C • 10x10G_LC • WSE

✐ ONS 15454-M2 does not support OSCM, AICs, EOL units, or ancillary units.

8.11 Equipment List for ONS 15454-M6

The ONS 15454-M6 supports all equipment that is supported by the ONS 15454 MSTP system.
The following additional equipment are supported.

Table 8-12 Prime Optical-Supported ONS 15454-M6

Card Type	Equipment
TXP	• M-CFP-LC • 100G-LC-C • 100G-CK-C • 10x10G_LC • WSE

✐ ONS 15454-M6 does not support OSCM, AIC, EOL units, or ancillary units.

8.12 Equipment List for CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH

The following table lists the supported cards for CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH.

Table 8-13: Prime Optical-Supported CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH Cards

Card Category	Card Equipment Type
Controller	• TNC • TSC
PTS	• PT_10GE_4 • PTF_10GE_4 • PTSA_GE
OUT	OTU2_XP
OPT	• OPT_AMP_17_C • OPT_AMP_C
ADM	ADM_10G
SMR	• 40-SMR1-C • 40-SMR2-C
TXP	AR_MXP
XP	• AR_XP • AR_XPE

The following table lists the supported ports for CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH.

Table 8-14: Prime Optical-Supported CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH Ports

Port Category	Port Equipment Type
PPM	PPM_1_PORT
OC	• OC3_PORT(SONET) • STM1_PORT(SDH) • OC12_PORT • STM4_PORT(SDH) • OC48_PORT • STM16_PORT(SDH) • OC192_PORT • STM64_PORT(SDH)
GE	• ONE_GE_PORT • TEN_GE_PORT • FE_PORT

OUT	OTU2
INTERLINK	INTERLINK_PORT
FC	FC10G_PORT
Data/Storage	IB_5G_PORT

8.13 Equipment List for ONS 15600 SDH

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15600 SDH user documentation for actual equipment names used by the NE.

Table 8-15: Prime Optical-Supported ONS 15600 SDH Equipment

Card Category	Equipment
ASAP	ASAP_4
Controller	TSC/TSCE
PPMs	• PPM_1_PORT • PIM_4_PPM • PIM_1_PPM
STM cards	• STM64_4 • STM16_16 • STM48_4 • STM64_4_DWDM
STM ports	• STM16_PORT • STM64_PORT • STM1_PORT • STM4_PORT
XC	• SSXC • CXC

8.14 Equipment List for ONS 15600 SONET

The equipment name is reported by the actual managed element. The following table lists the supported equipment names. For the most current list, refer to the Cisco ONS 15600 SONET user documentation for actual equipment names used by the NE.

Table 8-16: Prime Optical-Supported ONS 15600 SONET Equipment

Card Type	Equipment
ASAP	ASAP_4
Controller	TSC/TSCE
OC cards	• OC48_16 • OC48_32 • OC192_4 • OC192_4_DWDM
OC ports	• OC3_PORT • OC12_PORT • OC48_PORT • OC192_PORT
PPMs	• PIM_4_PPM • PPM_1_PORT • PIM_1_PPM

Card Type	Equipment
XC	• SSXC • CXC

8.15 Equipment List for Unmanaged NEs

Unmanaged NEs do not report any equipment.

8.16 CRS1 and CRS3 NEs

CRS1 and CRS3 NEs are managed as unmanaged NEs.

8.17 Equipment List for ONS NCS2K SONET and ONS NCS2K SDH

For the list of cards supported by an NCS2K SONET managed element equipped with an ONS 15454 M6 shelf, refer to the ONS15454 SONET list of supported cards and the other tables related to single- and double-slot managed cards by the NCS2K Managed Element only.

For the list of cards supported by an NCS2K SONET managed element equipped with an ONS 15454 M2 shelf, refer to the ONS15454 SONET list of supported cards and the other tables related to single- and double-slot managed cards by the NCS2K Managed Element only.

For the list of cards supported by an NCS2K SONET managed element equipped with an M6 shelf, refer to the ONS15454 SONET list of supported cards and the other tables related to single- and double-slot managed cards by the NCS2K Managed Element only.

For the list of cards supported by an NCS2K SDH managed element equipped with an M6 shelf, refer to the ONS15454 SONET list of supported cards and the other tables related to single- and double-slot managed cards by the NCS2K Managed Element only.

For the list of cards supported by an NCS2K SDH managed element equipped with an M2 shelf, refer to the ONS15454 SONET list of supported cards and the other tables related to single- and double-slot managed cards by the NCS2K Managed Element only.

The equipment name is reported by the actual managed element. The following tables lists the supported equipment names. For the most current list, refer to the Cisco ONS NCS2K SONET/SDH user documentation for actual equipment names used by the NE.

The Table 8-11: list the cards (single or double slot managed by NCS2K SONET/SDH only.

Table 8-17: Prime Optical-Supported ONS NCS2K SONET/ONS NCS2K SDH single slot Equipment

Card Type	Equipment
AD	
ADM	
Aux	
CE	
Controller	-
DS	-
ETH	-
FMEC	-
G	-

Card Type	Equipment
ML	-
MR	-
MSC	-
MXP	-
MUX/DMX	-
OPT	-
OSC	-
SMR	-
STM	-
TDCU	-
TXP	-
WSS	-
WXC	-
XC	-
XP	-
MESH	• 17-SMR9-FS • 24-SMR9-FS • 34-SMR9-FS • 20-SMR-FS
MD	• 12-AD-FS • 16-AD-FS

Table 8-18: Prime Optical-Supported ONS NCS2K SONET/SDH double slot Equipment

Card Type	Equipment
AD	-
ADM	-
Aux	-
CE	-
Controller	-
DS	-
ETH	-
G	-
ML	-
MR	-
XP	-
WXC	16-WXC-FS
MSC	-
MXP	-
MUX/DMX	-
OC	-
OPT	EDRA1-26C EDRA1-35C EDRA2-26C EDRA2-35C
OSC	-
SMR	-

Card Type	Equipment
TXP	-
WSS	-
WXC	-
XC	-
XP	-

ONS NCS2K SONET,ONS NCS 2K SDH, ONS 15454 M2, and ONS 15454 M6 NEs do not support OSCM, AIC, EOL or ancillary units.

The following table lists the supported ports for ONS NCS2K SONET and ONS NCS2K SDH.

Table 8-19: Prime Optical-Supported ONS NCS2K Ports

Port Category	Port Equipment Type
PPM	PPM_1_PORT
OC	OC3_PORT OC12_PORT OC48_PORT OC192_PORT OC768_PORT
FC	FC1G_PORT FC2G_PORT FC10G_PORT FC4G_PORT FC8G_PORT
FICON	FICON1G_PORT FICON2G_PORT FICON4G_PORT
GE	ONE_GE_PORT TEN_GE_PORT FORTY_GE_PORT ONEHUNDRED_GE_PORT
Data/Storage	DV6000_PORT ESCON_PORT SDI_D1_VIDEO_PORT HDTV_PORT IB_5G_PORT
ISC	ISC_PORT ISC3_PORT ISC3_1G_PORT ISC3_2G_PORT ISC3_STP_1G_PORT ISC3_STP_2G_PORT
GFP	GFP_PORT
OTU	OTU1_PORT OTU2_PORT OTU3_PORT OTU4_PORT
INTERLINK	INTERLINK_PORT

Port Category	Port Equipment Type
Other	PASS_THRU_PORT ETR_CLO_PORT SDI_3G_VIDEO_PORT SD_SDI_PORT HD_SDI_PORT AUTO_PORT

9 Layered Parameters and Layer Rates

This chapter lists the layered parameters associated with each layer rate for Prime Optical 10.7 NEs. Table 9-1 and Table 9-2 list the layered parameters associated with each layer rate for all cards of ONS 14454, ONS 15454 SDH, ONS NCS2K, ONS NCS2K SDH, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH. For all other ME types, all parameters are associated with all layers, as in previous releases.

The ALSMode attribute is not returned for amplifiers, and no layered parameters are returned for WXC legacy cards.

✐ If a card supports a subset of defined layered parameters for a specific layer, only the supported layered parameters are returned.

Table 9-1: Layered Parameters and Layer Rates Matrix (Layer Rates 2 - 40)

Parameter Name	Layer Rate																		
	2	**3**	**4**	**5**	**6**	**7**	**10**	**13**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**40**
AlarmReporting																			X
AdminState																			X
OpticsType																			X
OperationalState																			X
PortMediaType																			X
Autoadjust GFP Buffer Threshold																			
GFP Buffers Available																			
Ingress Idle Filtering																			
MaxFrameSize																			
Enable Distance Extension																			
Auto Detect Credits																			
Number Credits																			
ALSMode																			X
PulseWidth																			X
RecoveryInterval																			X
HasLaserBeenShutdown																			X
RestartLaser																			X
Reach																			X
Wavelength																			X

Parameter Name	Layer Rate																		
	2	3	4	5	6	7	10	13	19	20	21	22	23	24	25	26	27	28	40
FrameFormat	X	X	X	X	X	X													
LineCode			X	X															
SDH_SONET_SS_BITS								X						X	X	X	X	X	
EnableSyncMsg														X	X	X	X	X	
DetectedLineType			X																
FDLMode							X	X						X					
IPPMMonitor														X	X	X	X	X	
XCLoopback														X	X	X	X	X	
TrailTraceMonitor	X		X	X										X	X	X	X	X	
TrailTraceExpectedRx	X		X	X										X	X	X	X	X	
TrailTraceActualTx			X	X															

Table 9-2: Layered Parameters and Layer Rates Matrix (Layer Rates 46 - 124)

Parameter Name	Layer Rate													
	46	47	61	68	79	88	89	90	91	112	113	116	117	124
AlarmReporting	X	X	X	X										
AdminState	X	X	X	X										
OpticsType		X												
OperationalState	X	X	X											
PortMediaType		X												
Autoadjust GFP Buffer Threshold														X
GFP Buffers Available														X
Ingress Idle Filtering										X	X			
MaxFrameSize										X	X			
Enable Distance Extension														X
Auto Detect Credits														X
Number Credits														X
ALSMode		X												

Parameter Name	Layer Rate													
	46	47	61	68	79	88	89	90	91	112	113	116	117	124
PulseWidth		X												
RecoveryInterval		X												
HasLaserBeenShutdown		X												
RestartLaser		X												
Reach		X												
Wavelength		X												
FrameFormat														
LineCode					X									
SDH_SONET_SS_BITS							X		X					
EnableSyncMsg							X		X				X	
DetectedLineType														
FDLMode														
IPPMMonitor							X		X					
XCLoopback							X		X					
TrailTraceMonitor							X		X					
TrailTraceExpectedRx							X		X					
TrailTraceActualTx														

Table 10-3 and Table 10-4 list additional parameters for AR_XP, AR_MXP, AR_XPE, TNC, 100G_LC, 10x10G_LC, WSE and M-CFP-LC cards for ONS 14454, ONS 15454 SDH, ONS NCS2K, ONS NCS2K SDH, CPT 200, CPT 200 SDH, CPT 600, and CPT 600 SDH.

✎ If a card supports a subset of defined layered parameters for a specific layer, only the supported layered parameters are returned.

Table 9-3: Layered Parameters and Layer Rates Matrix for ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH (Layer Rates 2 - 28)

Parameter Name	Layer Rate														
	2	4	5	10	13	19	20	21	22	23	24	25	26	27	28
G7090TN															
FEC															X
SFBer															

Parameter Name	Layer Rate														
	2	4	5	10	13	19	20	21	22	23	24	25	26	27	28
SDBer															
AsyncSyncMapping															
OTUMapping															
ProvideSync															
EnableSyncMsgIn															
AdminSSMin															
SendDoNotUse															
ODUTransparency															
AINSSoak														X	
TerminationMode														X	
VideoType														X	
CurrentTransmitString						X	X	X	X	X	X	X			X
CurrentExpectedString						X	X	X	X	X	X	X			X
CurrentReceivedString						X	X	X	X	X	X	X			X
TraceLevel						X	X	X	X	X	X	X			X
DisableFDIonTTIM						X	X	X	X	X	X	X			X
AINSSquelch														X	
RequestRestart															
TraceMode						X	X	X	X	X	X	X			X
Overclock														X	
MTU															
Incomin MAC Address															
Mapping															
SyncMsgIn															
Type															
DisableAISRDIonTIMS															
TransmitLength															

Parameter Name	Layer Rate														
	2	4	5	10	13	19	20	21	22	23	24	25	26	27	28
AIS_Squelch_Configuration															

Table 9-4: Layered Parameters and Layer Rates Matrix for ONS 15454 SONET, ONS 15454 SDH, ONS NCS2K SONET, and ONS NCS2K SDH (Layer Rates 40 - 10019)

Parameter Name	Layer Rate															
	40	47	79	88	89	90	91	106	112	113	116	117	118	124	162	10019
G7090TN	X															
FEC													X			
SFBer	X				X											
SDBer	X				X											
AsyncSyncMapping	X															
OTUMapping	X															
ProvideSync	X				X											
EnableSyncMsgIn	X															
AdminSSMin	X				X											
SendDoNotUse	X				X											
ODUTransparency	X															
AINSSoak		X														
TerminationMode		X												X	X	X
VideoType		X														
CurrentTransmitString	X		X	X	X	X	X	X	X							
CurrentExpectedString	X		X	X	X	X	X	X	X							
CurrentReceivedString	X		X	X	X	X	X	X	X							
TraceLevel	X		X	X	X	X	X	X	X							
DisableFDIonTTIM	X		X	X	X	X	X	X	X							
AINSSquelch		X														
RequestRestart		X														
TraceMode	X		X	X	X											

Parameter Name	Layer Rate															
	40	**47**	**79**	**88**	**89**	**90**	**91**	**106**	**112**	**113**	**116**	**117**	**118**	**124**	**162**	**10019**
Overclock		X														
MTU														X	X	X
Incomin MAC Address														X	X	X
Mapping														X	X	X
SyncMsgIn					X											
Type					X											
DisableAISRDIonTIMS					X											
TransmitLength					X											
AIS_Squelch_Configuration														X		

10 Payloads and Layer Rates Matrix

Table 10-1 lists the associated layer rates for each port payload for the following ME types and cards:
ONS15454 and ONS15454 SDH: AR-XP, AR-MXP, AR-XPE, TNC, 100G_LC, 10x10G_LC, WSE, M-CFP-LC, ADM_10G, OPT_AMP_17_C, OPT_AMP_C, OPT_EDFA_17, OPT_EDFA_24, OTU2_XP, RAMAN_COP, RAMAN_CPT, SMR1_40_C, SMR2_40_C, TXP_MR_10E, 10GE_XP, XTP_FAB_10GE, XTP_PTSA_GE, MXP_2.5G_10E, 40G_MXP_C, 40G_TXP_C, 40E_MXP_C, 10_DME, TXP_MR_10E, 10GE_XP.

NCS2K and NCS2K SDH: AR-XP, AR-MXP, AR-XPE, TNC, 100G_LC, 10x10G_LC, WSE, M-CFP-LC, ADM_10G, OPT_AMP_17_C, OPT_AMP_C, OPT_EDFA_17, OPT_EDFA_24, OTU2_XP, RAMAN_COP, RAMAN_CPT, SMR1_40_C, SMR2_40_C, TXP_MR_10E, 10GE_XP, XTP_FAB_10GE, XTP_PTSA_GE, MXP_2.5G_10E, 40G_MXP_C, 40G_TXP_C, 40E_MXP_C, 10_DME, TXP_MR_10E, 10GE_XP.

CPT200, CPT 200 SDH, CPT600, and CPT 600 SDH: AR_XP, AR_MXP, AR_XPE, TNC, ADM_10G, OPT_AMP_17_C, OPT_AMP_C, OTU2, SMR1_40_C, SMR2_40_C, XTP_10GE, XTP_FAB_10GE, XTP_PTSA_GE

Table 100-1: Payloads and Layer Rates Matrix

Payload	Card	Layer Rate Name	Layer Rate Number
AUTO_PORT	Any	LR_AUTO	10008
DV6000_PORT	Any	LR_DV_6000	148
		LR_Optical_Channel	40
DVBASI_PORT	Any	LR_DVBASI	156
		LR_Optical_Channel	40
ESCON_PORT	Any	LR_ESCON	59
		LR_Optical_Channel	40
ETR_CLO_PORT	Any	LR_ETR	60
		LR_Optical_Channel	40
FC1G_PORT	Any	LR_FC_100_1063M	65
		LR_Optical_Channel	40
FC2G_PORT	Any	LR_FC_200_2125M	111
		LR_Optical_Channel	40
FC4G_PORT	Any	LR_4_GFC	134
		LR_Optical_Channel	40
FC8G_PORT	Any	LR_8_GFC	160
		LR_Optical_Channel	40
FC10G_PORT	Any	LR_10_GFC	133
		LR_Optical_Channel	40
FE_PORT	TNC	LR_Fast_Ethernet	61
		LR_PHYSICAL_OPTICAL	47

Payload	Card	Layer Rate Name	Layer Rate Number
	Any	LR_Fast_Ethernet	61
		LR_Optical_Channel	40
FORTY_GE_PORT	CFP, Any	LR_Forty_Gigabit_Ethernet	162
		LR_PHYSICAL_OPTICAL	47
HD_SDI_PORT	Any	LR_Optical_Channel	40
		LR_HD_SDI	10004
ISC1_PORT	Any	LR_ISC1	155
		LR_Optical_Channel	40
ISC3_PORT	Any	LR_ISC3_PEER_1G	143
		LR_ISC3_PEER_2G	144
		LR_Optical_Channel	40
ISC3_1G_PORT	Any	LR_ISC3_PEER_1G	143
		LR_Optical_Channel	40
ISC3_2G_PORT	Any	LR_ISC3_PEER_2G	144
		LR_Optical_Channel	40
OC3_PORT	TNC	LR_Section_OC3_STS3_and_RS_STM1	20
		LR_Line_OC3_STS3_and_MS_STM1	25
		LR_PHYSICAL_OPTICAL	47
		LR_DSR_OC3_STM1	73
		LR_OPTICAL_SECTION	49
		LR_Optical_Channel	40
		LR_Optical_Transmission_Section	42
		LR_Optical_Multiplex_Section	41
	Any	LR_Section_OC3_STS3_and_RS_STM1	20
		LR_Line_OC3_STS3_and_MS_STM1	25
		LR_DSR_OC3_STM1	73
		LR_Optical_Channel	40
OC12_PORT	Any	LR_Section_OC12_STS12_and_RS_STM4	21
		LR_Line_OC12_STS12_and_MS_STM4	26
		LR_DSR_OC12_STM4	74
		LR_Optical_Channel	40
OC48_PORT	Any	LR_Section_OC48_STS48_and_RS_STM16	22
		LR_Line_OC48_STS48_and_MS_STM16	27
		LR_DSR_OC48_STM16	76
		LR_Optical_Channel	40

Payload	Card	Layer Rate Name	Layer Rate Number
OC192_PORT	Any	LR_Section_OC192_STS192_and_RS_STM64	23
		LR_Line_OC192_STS192_and_MS_STM64	28
		LR_DSR_OC192_STM64	77
		LR_Optical_Channel	40
ONE_GE_PORT	TNC	LR_Gigabit_Ethernet	68
		LR_PHYSICAL_OPTICAL	47
	PTSA	LR_Gigabit_Ethernet	68
		LR_Optical_Channel	40
		LR_E1_2M	5
		LR_E3_34M	7
		LR_T1_and_DS1_1_5M	2
		LR_T3_and_DS3_45M	4
	Any	LR_Gigabit_Ethernet	68
		LR_Optical_Channel	40
ONEHUNDRED_GE_PORT	100G, CFP, Any	LR_OneHundred_Gigabit_Ethernet	10019
		LR_PHYSICAL_OPTICAL	47
OPTICS_AMP_LINE_PORT	Any	LR_PHYSICAL_OPTICAL	47
		LR_Optical_Transmission_Section	42
OPTICS_LINE_PORT	Any	LR_PHYSICAL_OPTICAL	47
		LR_Optical_Transmission_Section	42
OTL_PORT	Any	LR_Optical_Channel	40
OTU1_PORT	Any	LR_Optical_Transport_Section	117
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_FEC	118
		LR_Physical_2_5_Gigabit_ITU	108
		LR_OCH_OTU1	10010
		LR_Optical_Channel	40
		LR_CPO_DSR_OTU1	10009
OTU2_PORT	PTF_10GE	LR_Ten_Gigabit_Ethernet	106
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_Section	117
		LR_Optical_Transport_FEC	118
		LR_Optical_Channel	40
	TXP_MR_10E	LR_Section_OC192_STS192_and_RS_STM64	23

Payload	Card	Layer Rate Name	Layer Rate Number
		LR_Line_OC192_STS192_and_MS_STM64	28
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_Section	117
		LR_Optical_Transport_FEC	118
		LR_Optical_Channel	40
	Any	LR_Optical_Transport_Section	117
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_FEC	118
		LR_Physical_10_Gigabit_ITU	107
		LR_OCH_OTU2	10011
		LR_Optical_Channel	40
OTU3_PORT	CFP, Any	LR_Optical_Transport_Section	117
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_FEC	118
		LR_Physical_40_Gigabit_ITU	152
		LR_OCH_OTU3	10012
		LR_Optical_Channel	40
OTU4_PORT	400G-XP-LC,MR-MXP, 100G, CFP, Any	LR_Optical_Transport_Section	117
		LR_Optical_Transport_Path	116
		LR_Optical_Transport_FEC	118
		LR_Physical_100_Gigabit_ITU	10018
		LR_OCH_OTU4	10013
		LR_Optical_Channel	40
RAMAN_OPTICS_LINE_PORT	Any	LR_PHYSICAL_OPTICAL	47
		LR_Optical_Transmission_Section	42
SD_SDI_PORT	Any	LR_Optical_Channel	40
		LR_SD_SDI	10003
SDI_3G_VIDEO_PORT	Any	LR_Optical_Channel	40
		LR_SDI_3G_VIDEO	10005
SDI_D1_VIDEO_PORT	Any	LR_D1_Video	58
		LR_Optical_Channel	40
TEN_GE_PORT	PTF_10GE, PT_10GE	LR_Ten_Gigabit_Ethernet	106
		LR_L2_10_Gigabit_ITU	153
	10X10G-LC WSE	LR_Ten_Gigabit_Ethernet	106
		LR_PHYSICAL_OPTICAL	47

Payload	Card	Layer Rate Name	Layer Rate Number
	Any	LR_Ten_Gigabit_Ethernet	106
		LR_Optical_Channel	40

11 PTP dataRate attribute

The dataRate attribute is available in the additionalInfo section for each returned PTP (see 3.8.15 getTP, 3.8.5 getAllPTPs and 3.6.3 getAllSupportedPTPs APIs).

The attribute is provided for all cards supporting the get PTP payload service for the follow list of NE types:

- ONS15454
- ONS15454 SDH
- CPT200
- CPT200 SDH
- CPT600
- CPT600 SDH
- NCS2K
- NCS2K SDH
- ONS15310 MA
- ONS15310 MA SDH
- ONS15310 CL
- ONS15600
- ONS15600 SDH

Valid Data Rate values are:
E1, E3, DS3, DS1, OTU1, OTU2, OTU3, OTU4, OC3_STM1, OC12_STM4, OC192_STM64, OC48_STM16, OC768_STM256, ETH, FE, ONE_GE, TEN_GE, FORTY_GE, ONEHUNDRED_GE, OCH, OTS, OMS, GFP, FC1G, FC2G, FC4G, FC8G, FC10G, FICON1G, FICON2G, CLIENT, TRUNK, POS, EC1, ISC3_STP_2G, OTLx_y, ISC, ISC3, ISC3_1G, ISC3_2G, ISC3_STP_1G, AUTO, SDI_3G, SDI_3G_VIDEO, SDI_D1_VIDEO, HD_SDI, SD_SDI, ESCON, DTV, ETR_CLO, PASS_THRU, IB_5G, HDTV.

Example:

```
Name                                = [0] EMS=Cisco Systems/PrimeOptical; [1]
ManagedElement=15454-ANSI-68-8; [2]
PTP=/rack=1/shelf=1/slot=4/ppm_holder=19/port=1
User Label                          =
Native EMS Name                     = 15454-ANSI-68-
8//rack=1/shelf=1/slot=4/ppm_holder=19/port=1
Owner                               =
Ingress Traffic Descriptor Name     =
Egress Traffic Descriptor Name      =
Type                                = TPT_PTP
Connection State                    = TPCS_NA
Mapping Mode                        = TM_NA
Direction                           = D_BIDIRECTIONAL
Transmission Params                 = layer=[23]
LR_Section_OC192_STS192_and_RS_STM64 -->
                                      layer=[28]
LR_Line_OC192_STS192_and_MS_STM64 -->
                                      layer=[77] LR_DSR_OC192_STM64 -->
                                      layer=[109] LR_OCH_10_Gigabit_ITU -->
                                      layer=[40] LR_Optical_Channel --> [0]
AdminState=OUT_OF_SERVICE; [1] OperationalState=OOS-MA, DSBLD; [2]
ALSMode=DISABLED; [3] AINSSoak=8h:0min; [4] PulseWidth=100.0; [5]
RecoveryInterval=300; [6] HasLaserBeenShutdown=notAvailable; [7]
```

```
AlarmReporting=on
Protection Association              = TPPA_NA
Edge Point                          = true
Additional Info                     = [0] dataRate=OC192_STM64
```