EMC® Smarts®
Optical Transport Manager

Version 9.4

API Object Factory Reference Guide

P/N 302-001-512
REV 01

EMC²

Copyright ©2004 - 2015 EMC Corporation. All rights reserved. Published in the USA.

Published January 2015

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC^2, EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com).

CONTENTS

Chapter 1 Introduction

Introduction 6
Overview of the API 6
Relationships between OTM components 8

Chapter 2 SONET/SDH Object Factory API

SONET/SDH components and API summary 12
SONET/SDH physical components 13
SONET/SDH Logical Components 24

Chapter 3 WDM Object Factory API

WDM components and API summary 44
WDM physical components 47
WDM logical components 73

Chapter 4 Low Order SONET/SDH Object Factory API

Low Order SONET/SDH components and API summary 86
Low Order SONET/SDH physical components 87
Low Order SONET/SDH Logical Components 98

CHAPTER 1
Introduction

This chapter presents an overview of the object factory API as well as some background on the OTM domains, relationships between objects, and syntax for the API calls.

This chapter contains the following sections:

- Introduction 6
- Overview of the API 6
- Relationships between OTM components 8

Introduction

This document describes the object factory Application Programmer Interface (API) for the three EMC® Smarts® Optical Transport Manager (OTM) domains: SONET/SDH, Low Order SONET/SDH, and WDM. The object factory API is used to directly interact with the OTM Domain Managers for these three OTM domains.

This document only covers the existing API capabilities. As part of an effort to improve and enhance the performance and capabilities of its product lines, the API and this document will be updated to reflect any changes or enhancements made to the API. Therefore, some functions described in this document may not be supported by all versions of OTM or hardware currently in use.

Overview of the API

The OTM object factory API is the direct interface to the OTM Domain Managers for the three OTM domains:

- SONET/SDH
- Low Order SONET/SDH
- WDM

The goal of the API is to facilitate topology creation and modification within the OTM Domain Managers repositories, and setting and clearing attributes as part of alarm and event processing within different OTM Domain Managers.

The API can be used by:

- Topology adapters (for example, file topology adapters, TMF814 adapters, and TL1 adapters)
- Event adapters (for example, TMF814 adapters, TMF854 adapters, and TL1 adapters)
- Post-processing scripts

An API can be called from any language that implements supports for API functionality such as C, C++, Java, Perl, or the EMC® Smarts® Adapter Scripting Language.

OTM Domain Managers

It is important to understand the architectural layout of the OTM Domain Managers and their interactions with other entities in the EMC Smarts OTM solution. These are shown in Figure 1 on page 7.

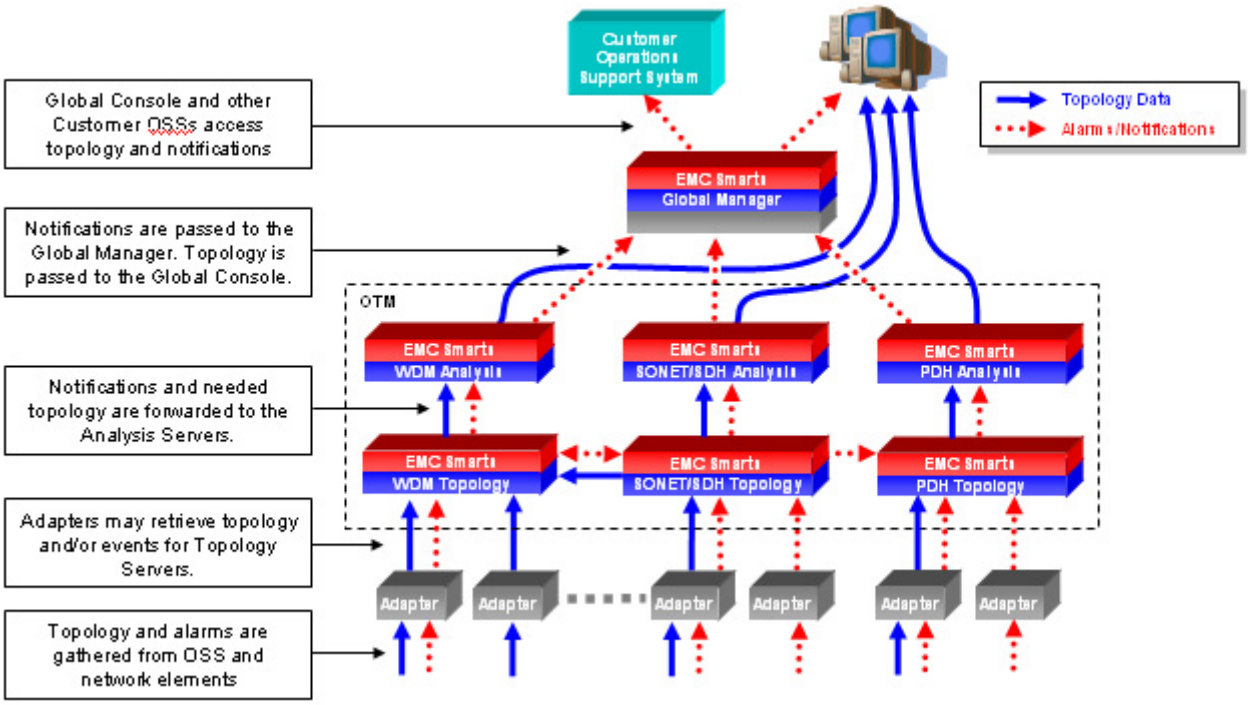

Figure 1 Data flows in EMC Smarts Optical Transport Manager

The functions of each OTM domain— SONET/SDH, Low Order SONET/SDH and WDM—are split into Topology/Monitoring server (or simply Topology server) and Analysis server. Currently, each Domain Manager supports a single Topology server and a single Analysis server. All external software entities, such as the adapters shown as gray boxes at the bottom of Figure 1 on page 7, interact only with the Topology servers.

Topology server

The Topology server consists of the underlying managed objects, physical and logical network connectivity, and all attributes related to monitored status and events for all components of the underlying network. All realtime event and alarm updates from various events and alarms monitoring systems will be fed to this server by one or more adapters.

Analysis server

The Analysis server consists of all the information needed for performing root cause analysis. Topology is imported automatically from the Topology server using EMC Smarts SDXA. Events are imported automatically from the Topology server using Remote Accessory Instrumentation. It contains information like objects, their attributes and relationships, events, symptoms, and problems needed for analysis.

Relationships between OTM components

This is a brief overview of OTM relationships. A more detailed discussion of the relationships and the OTM model definitions is described in the Optical Transport Manager User Guide.

- There are different types of relationships between the various physical and logical components that are needed for event correlation and root cause analysis.
- Once a bidirectional relationship is created in one direction, the reverse relationship is automatically created.
- Some relationships are known beforehand and are created by the API at the time of creating the components. These are indicated in relationship tables given for each physical and logical component.
- Some relationships are not known at the time the component is created and need to be created later.
- Physical and logical component relationships are detailed in the relationship tables indicating the Objects/Classes involved in the relationship, class to which it is related, reverse relation, cardinality of relation, cardinality of the reverse relation, and whether the relation is optional or mandatory.
- The component name is the same as the class name and is also the type of object instantiated by the API make functions and returned by the find function for the component.
- A relationship is optional when the objects involved in the relationship are not always present for each other. However, the relationship must be created when the related objects are available and present.

Creating relationship and accessing related objects

A relationship is created or accessed using the arrow operator, “->”.

The following examples illustrate how relationships can be created using EMC Smarts ASL language when the relationships are not automatically created or available from the APIs.

```
Hocct -> LayeredOver = Hotrail1;
```

Creates a “LayeredOver” relationship between object “Hocct” (of type HighOrder_Circuit) and “Hotrail1” (of type HighOrder_Trail).

```
Hocct -> LayeredOver += Hotrail2;
```

Creates a “LayeredOver” relationship between object “Hocct” (of type HighOrder_Circuit) and “Hotrail2” (of type HighOrder_Trail) in addition to existing “LayeredOver” relationships.

```
Hotrail1 -> LayeredOver = tlg1;
```

Creates a “LayeredOver” relationship between object “Hotrail1” (of type HighOrder_Trail) and “tlg1” (of type TopologicalLinkGroup).

```
Hotrail1 -> LayeredOver += tl1;
```

Creates a “LayeredOver” relationship between object “Hotrail1” (of type HighOrder_Trail) and “tl1” (of type TopologicalLink) in addition to existing “LayeredOver” relationships.

The following example illustrates how the objects involved in a relationship can be obtained after the relationship is created using EMC Smarts ASL language.

```
hoTrailList = list(); hoTrailList = Hocct -> LayeredOver;
```

This populates the list “hoTrailList” with a list of objects “Underlying” the object “Hocct.” “Hocct” is of type HighOrder_Circuit and its “Underlying” objects are of type HighOrder_Trail. “Underlying” is the reverse relationship of “LayeredOver.”

CHAPTER 2
SONET/SDH Object Factory API

These APIs are intended for use within the SONET/SDH Topology server. They facilitate topology creation and modification for physical and logical network components in the OTM SONET/SDH Domain Manager. The API set is categorized into two parts: Physical components and Logical components.

This chapter contains the following information:

- SONET/SDH components and API summary 12
- SONET/SDH physical components..... 13
- SONET/SDH Logical Components 24

SONET/SDH components and API summary

Table 1 on page 12 lists the SONET/SDH OTM components along with the APIs used to make and find that component. Also indicated is whether the component is physical hardware or a logical construction.

Note: API names for creating components start with "make" and API names for finding components already created and existing within OTM repository start with "find." Some API names use the convention "makeN*xxx*" or "findN*xxx.*" The "N" emphasizes the fact that the native name (also known as the native AID) is used in forming the final component name without undergoing any change or normalization.

Table 1 SONET/SDH components and API summary (page 1 of 2)

Component Name/Class Instantiated	API Name	Entity Type	Details
EMS	makeEMS findEms	Physical	page 13
OpticalNetworkElement	makeNOpticalNetworkElement findOpticalNetworkElement	Physical	page 15
Equipment/Card	makeNEquipment, findNEquipment makeEquipment, findEquipment makeNCard, findNCard	Physical	page 16
PTP	makeNPTP, findNPTP **Note:** makePTP and findPTP are deprecated. Use makeNPTP and findNPTP instead.	Physical	page 19
Rack	makeRack findRack	Physical	page 20
Shelf	makeShelf, findShelf makeNShelf, findNShelf	Physical	page 21
TopologicalLink	makeTopologicalLink findTopologicalLink	Physical	page 23
DropSideTopologicalLink	makeDropSideTopologicalLink findDropSideTopologicalLink	Physical	page 24
CTP	makeNCTP, findNCTP makeCTP, findCTP	Logical	page 25
High Order CTP	makeHOCTP, findHOCTP	Logical	page 27
HighOrder_Circuit	makeHOCircuit findHOCircuit	Logical	page 30
HighOrder_Trail	makeHOTrail findHOTrail	Logical	page 31
SubnetworkConnection	makeSubnetworkConnection findSubnetworkConnection	Logical	page 32

Table 1 SONET/SDH components and API summary (page 2 of 2)

TopologicalLinkGroup	makeTLGroup make2FBLSR	Logical	page 34
BlackBoxTopologicalLink	makeBlackBoxTL findBlackBoxTL	Logical	page 36
BlackBoxDropSideTopologicalLink	makeBlackBoxDSTL findBlackBoxDSTL	Logical	page 37
RingProtectionGroup	makeRPGroup	Logical	page 38
EquipmentProtectionGroup	makeEqPGroup	Logical	page 40
LogicalConnectionTPGroup	makeLCTPGroup	Logical	page 41

Data layout

Under each component name, the following information is provided:

The object name to be created or found is a heading in the left column.

API Usage

Under the heading is the commands showing the usage syntax including the order of parameter.

Mandatory Parameters

These parameters are required. Parameters are described in three columns:

Parameter name, Data type, Description

Optional Parameters

These parameters are not required. Parameters are described in three columns:

Parameter name, Data type, Description

Return Values

Lists the object types returned by the APIs. Return values are listed in two columns:

Type of object reference returned, Description

Dependency

List of dependencies of the API on other objects.

Relationships

Lists of the relationships of the object with all other objects.

SONET/SDH physical components

This section describes the APIs that enable creation and or alteration of managed objects that form the physical components of the underlying network.

EMS

makeEMS (emsName, displayName = "*value*", className = "*value*")

findEMS (emsName)

Mandatory Parameters

emsName	String	Name of the EMS to which the SONET/SDH Domain Manager will be connecting. Must be a unique name.

Optional Parameters

displayName	String	User-friendly name of the EMS object.
className	String	Class name of the object to be created or found. Currently “EMS” is the only Class name for this API.

Return Values

EMS	Reference to the EMS object if the EMS object with the given emsName is successfully created or found.
null	If makeEMS is invoked with the NULL emsName, or if the EMS is not found.

Dependency

None

Relationships

Relationship	Related Class	Reverse Relationship	Card-inality	Mandatory (M) Optional (O)	API Relation
Manages	OpticalNetworkElement	ManagedBy	1:N	M	Y
ManagesSNC	SubnetworkConnection	ManagedBy	1:N	O	Y
ManagesTL	TopologicalLink	ManagedBy	1:N	M	Y

OpticalNetworkElement

makeNOpticalNetworkElement (tid, emsName, displayName = "*value*", className = "*value*")

findOpticalNetworkElement (tid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
emsName	String	Name of the EMS to which OpticalNetworkElement belongs.

Optional Parameters

displayName	String	user-friendly name of the object displayed in the console.
className	String	Class name of the object to be created or found. Currently OpticalNetworkElement is the only Class name for this API.

Return Values

OpticalNetworkElement	Reference to the OpticalNetworkElement object if the object is successfully created or found.
null	If API fails to create or find the OpticalNetworkElement object.

Dependency

An EMS object must exist for OpticalNetworkElement object to be created.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory/Optional (M/O)	API Relation
ManagedBy	EMS	Manages	1:1	M	Y
ComposedOfOpticalComponent	Equipment, Card, PTP	PartOf	1:N	M	Y
ComposedOfPTPs	PTP	None	1:N	M	Y
ComposedOfEquipment	Equipment, Card	PartOf	1:N	M	Y
ComposedOfCTPs	CTP	None	1:N	M	Y
SystemPackagedIn	Rack	PackagesSystems	1:1	O	Y

Equipment/Card

makeEquipment (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", cardType, displayName = "*value*", className = "*value*")

makeNCard (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", cardType, displayName = "*value*", className = "*value*")

Note: makeCard is deprecated. Use makeNCard instead.

---

makeNEquipment (neName, shelfAid = "*value*", cardAid, cardType, displayName = "*value*", className = "*value*")

findEquipment (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*")

findNCard (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*")

Note: findCard is deprecated. Use findNCard instead.

---

findNEquipment (neName, cardAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Equipment or Card resides.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
cardAid	String	Access Identifier for the Equipment or Card within the network element.
cardType	String	Type of card. Used to uniquely identify between multiple cards from the same vendor. This plays a role while constructing EquipmentProtectionGroup and CrossConnect Equipments or the Card equivalents.

Optional Parameters

shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element. **Note:** For findEquipment, must be “0”.
displayName	String	user-friendly name to label the Equipment in the console.
className	String	Class name of the object to be created or found. Currently Equipment is the only Class name for this API.

Return Values

Equipment	Reference to the Equipment object if Equipment is successfully created or found.
null	If API is not able to create or find Equipment.

Dependency

- Object of type OpticalNetworkElement with the name "neName".
- Object of type Shelf to be created if "PartOfShelf" relationship is to be created later.
- An Equipment/Card containing another Equipment/Card (mother card - daughter card) PartOfEquipment relationship needs to be explicitly created after creating both the Equipments/Cards.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
Part Of	OpticalNetworkElement	ComposedOfOpticalComponent	1:1	M	Y
PartOfShelf	Shelf	ComposedOfEquipment	1:1	O	N
PartOfEquipment	Equipment	ComposedOf	1:1	O	N
Realizes	PTP	RealizedBy	1:N	M	Y
PartOfEquipmentProtectionGroup	EquipmentProtectionGroup	ComposedOf	1:1	O	Y
CrossConnects	PTP	CrossConnectedByEqpt	1:N	O	N

PTP (Physical Termination Point)

makeNPTP (neName, cardAid, ptpAid, displayName = "*value*", className = "*value*", relateEquipment = "*value*")

Note: makePTP is deprecated. Use makeNPTP instead.

findNPTP (neName, ptpAid)

Note: findPTP is deprecated. Use findNPTP instead.

Mandatory Parameters

neName	String	Name of the Network Element in which the PTP resides.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
portAid portNumber	String Integer	Access Identifier of the port on the Equipment/Card in the specified slot or subslot.
cardAid	String	Access Identifier for the Equipment/Card within the network element to which the PTP belongs. Uniquely identifies the Equipment/Card object within the OTM repository.
ptpAid	String	Access Identifier for the PTP within the network element. Uniquely identifies the PTP within the OTM repository

Optional Parameters

displayName	String	user-friendly name to represent the PTP in the console.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element.
className	String	Class name of the object to be created or found. Currently PTP is the only Class name for this API.
relateEquipment	Boolean	By default this is set to "TRUE." It is recommended not to change the default. This facilitates creating the relationship between the PTP and the Equipment/Card containing the PTP.

Return Values

PTP	Reference to the PTP object if created or found.
null	If API is not able to create or find PTP.

Dependency

- Object of type OpticalNetworkElement with the name "neName"
- Object of type Equipment/Card with the name "cardAid"

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
RealizedBy	Equipment	Realizes	N:1	M	Y
Underlying	CTP	LayeredOver	1:N	M	Y
ConnectedVia	TopologicalLink	ConnectedTo	1:1	M	Y
ConnectedVia	DropSideTopologicalLink	ConnectedTo	1:1	M	Y
WestSendsTo	TopologicalLink	None	1:1	M	Y
WestSendsTo	DropSideTopologicalLink	None	1:1	O	Y
EastSendsTo	TopologicalLink	None	1:1	M	Y
EastSendsTo	DropSideTopologicalLink	None	1:1	O	Y
CrossConnectedBy	EquipmentProtectionGroup	CrossConnects	1:1	O	Y
CrossConnectedByEqpt	Equipment	CrossConnects	N:N	O	N

Rack

makeRack (neName, rackAid, displayName = "*value*", className = "*value*")

findRack (neName, rackAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Rack resides.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository.

Optional Parameters

displayName	String	User-friendly name to represent the Rack in the console.
className	String	Class name of the object to be created or found. Currently Rack is the only Class name for this API.

Return Values

Rack	Reference to the Rack object if successfully created or found.
null	If API is not able to create or find the Rack.

Dependency

Object of type OpticalNetworkElement (with name/tid "neName") containing the Rack.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOf	Shelf	PartOf	1:N	O	Y
PackagesSystems	OpticalNetworkElement	SystemPackagedIn	1:1	O	Y

Shelf

makeShelf (neneName, rackAid, shelfAid, displayName = "*value*", className = "*value*")

makeNShelf (neneName, shelfAid, displayName = "*value*", className = "*value*")

findShelf (neneName, rackAid, shelfAid)

findNShelf (neneName, shelfAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Shelf resides.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository

Optional Parameters

displayName	String	User-friendly name to represent the Shelf in the console.
className	String	Class name of the object to be created or found. Currently Shelf is the only Class name for this API.

Return Values

Shelf	Reference to the Shelf object if successfully created or found.
null	If API is not able to create or find the Shelf.

Dependency

- Object of type OpticalNetworkElement (with name/tid "neName") containing the Shelf.
- Object of type Rack in which the Shelf resides when using makeShelf.
- makeShelf uses neName, rackAid and shelfAid to form the unique shelf name in the repository. findShelf uses neName, rackAid and shelfAid to form the unique shelf name to do a lookup.
- makeNShelf uses neName and shelfAid to form the unique shelf name in the repository. findNShelf uses neName and shelfAid to form the unique shelf name to do a lookup.
- If Shelf is created using makeShelf, findShelf should be used to find it and not findNShelf.
- If Shelf is created using makeNShelf, findNShelf should be used to find it and not findShelf.

- If makeNShelf is used, the relationship between Rack and Shelf has to be explicitly created.
- If Shelf is created using makeShelf, shelfAid should not be used in makeNEquipment, a blank shelfAid should be used, since makeNEquipment uses findNShelf (with no rackAid) and creates the relationship between the Equipment and the Shelf.
- If Shelf is created using makeShelf and an Equipment object in that Shelf is created using makeNEquipment, the relationship (PartOfShelf) between the Equipment and Shelf has to be explicitly created.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
PartOf	Rack	ComposedOf	1:1	O	Y
ComposedOfEquipment	Equipment	PartOfShelf	1:N	O	N

TopologicalLink

makeTopologicalLink (emsName = "*value*", topologicalLinkName, AendPTP, ZendPTP, displayName = "*value*", className = "*value*")

findTopologicalLink (emsName, topologicalLinkName)

Mandatory Parameters

topologicalLinkName	String	Name of the TopologicalLink used to uniquely identify it within the OTM repository.
AendPTP	Reference to PTP	PTP object at one end of the TopologicalLink
ZendPTP	Reference to PTP	PTP object at the other end of the TopologicalLink

Optional Parameters

emsName	String	Name of the EMS to which SONET/SDH Domain Manager will be connecting. It must be a unique name.
displayName	String	User-friendly name to represent the TopologicalLink in the console.
className	String	Class name of the object to be created or found. Currently TopologicalLink is the only Class name for this API.

Return Values

TopologicalLink	Reference to the TopologicalLink if successfully created or found.
null	If API is not able to create or find a TopologicalLink object.

Dependency

- Object of type EMS with the name "emsName" if emsName provided.
- Object of type PTP - AendPTP created with its unique name/aid
- Object of type PTP - ZendPTP created with its unique name/aid

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedTo	PTP	ConnectedVia	1:N	M	Y
EastSendsTo	PTP	None	1:1	M	Y
WestSendsTo	PTP	None	1:1	M	Y
ManagedBy	EMS	ManagesTL	1:1	M	Y
PartOf	TopologicalLinkGroup	ComposedOf	1:1	O	Y
RouteProtectedFor	RingProtectionGroup	RouteProtectedBy	N:N	O	Y

DropSideTopologicalLink

makeDropSideTopologicalLink (dropSidePTP, isWestPTP, displayName = "*value*", className = "*value*", emsName = "*value*")

findDropSideTopologicalLink (dropSidePTP)

Mandatory Parameters

dropSidePTP	Reference to PTP	PTP object at the drop side of the Circuit that forms the DropSideTopologicalLink.
isWestPTP	Boolean	Relative indication of whether the drop sidePTP is on the East or West end of the Circuit.

Optional Parameters

displayName	String	User-friendly name to represent the DropSideTopologicalLink in the console.
className	String	Class name of the object to be created or found. Currently DropSideTopologicalLink is the only Class name for this API.
emsName	String	Name of the EMS to which SONET/SDH Domain Manager will be connecting. It must be a unique name.

Return Values

Dependency

DropSideTopologicalLink	Reference to the DropSideTopologicalLink, if successfully created or found.
null	If API is not able to create or find DropSideTopologicalLink object.

- Object of type EMS with the name "emsName" if emsName provided.
- Object of type PTP - dropSidePTP created with its unique name/aid

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedTo	PTP	ConnectedVia	1:1	M	Y
EastSendsTo	PTP	None	1:1	O	Y
WestSendsTo	PTP	None	1:1	O	Y
Underlying	HighOrder_Trail, HighOrder_Circuit	LayeredOver	1:N	M	N
PartOf	TopologicalLinkGroup	ComposedOf	1:1	O	Y

SONET/SDH Logical Components

This section describes the APIs that enable creation and or alteration of managed objects that form the logical components of the underlying network.

CTP (Connection Termination Point)

makeCTP (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", portNumber, timeSlot, ctpType, displayName = "*value*", className = "*value*", createContainment = "*value*")

makeNCTP (neName, cardAid, ptpAid, ctpAid, ctpType, displayName = "*value*", className = "*value*", createContainment = "*value*")

findCTP (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", portNumber, timeSlot)

findNCTP (neName, ctpAid)

Mandatory Parameters

neName	String	Name of the Network Element (neName/tid) containing the PTP to which the CTP belongs.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
portNumber	Integer	Access Identifier of the port on the Equipment/Card in the specified slot or subslot.
timeSlot	Integer	
cardAid	String	Access Identifier for the Card within the network element to which the PTP belongs. Uniquely identifies the Card within the OTM repository.
ptpAid	String	Access Identifier for the PTP to which the CTP is "LayeredOver". Uniquely identifies the PTP within the OTM repository.

ctpAid	String	Access Identifier for the CTP within the network element. Uniquely identifies the CTP within the OTM repository.
ctpType	Enum	Enum of type Optical_GlobalDefs::ctp_type_e. Category of the CTP from the following types:

CTP_UNKNOWN	CTP_VC4_8c
CTP_DS3	CTP_VC4_16
CTP_STS1	CTP_VC4_64c
CTP_STS3	CTP_ETS1
CTP_STS3C	CTP_ETS3
CTP_STS12	CTP_ETS6
CTP_STS12C	CTP_ETS9
CTP_STS24C	CTP_ETS12
CTP_STS48	CTP_ETS24
CTP_STS48C	CTP_E3
CTP_STS192C	CTP_34MB
CTP_AU3	CTP_45MB
CTP_AU4	CTP_50MB
CTP_VC4_4	CTP_100MB
CTP_VC4_4c	

Optional Parameters

displayName	String	User-friendly name to represent the CTP.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element.
className	String	Class name of the object to be created or found. Currently CTP is the only Class name for this API.
createContainment	Boolean	This is by default set to TRUE. It is recommended not to change the default. This facilitates the creation of the relationship between the CTP and underlying PTP.

Return Values

CTP	Reference to the CTP if successfully created or found.
null	If API is not able to create or find CTP object.

Dependency

- Object of type OpticalNetworkElement (with name “neName”) where the PTP containing this CTP resides.
- Object of type Equipment (with name/aid cardAid) containing the PTP (with name/aid ptpAid).

- Object of type PTP (with name/aid “ptpAid”) containing this CTP.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedVia	SubnetworkConnection	ConnectedTo	1:1	O	Y
LayeredOver	PTP	Underlying	1:1	M	Y
RoutedBy	HighOrder_Circuit, HighOrder_Trail	CTPsInRoute	1:1	M	Y

High Order CTP (Connection Termination Point)

makeHOCTP (neName, rackAid, shelfAid = “*value*“, slotNumber, subSlotNumber = “*value*“, portNumber, timeSlot, channel, ctpType, displayName = “*value*“, className = “*value*“, createContainment = “*value*“)

findHOCTP (neName, rackAid, shelfAid = “*value*“, slotNumber, subSlotNumber = “*value*“, portNumber, timeSlot, channel)

Mandatory Parameters

neName	String	Name of the Network Element (neName/tid) containing the PTP to which the High Order CTP belongs.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
portNumber	Integer	Access Identifier of the port on the Equipment/Card in the specified slot or subslot.
timeSlot	Integer	
channel	Integer	

ctpType	Enum	Enum of type Optical_GlobalDefs::ctp_type_e. Category of the High Order CTP from the following types:

CTP_UNKNOWN	CTP_VC4_8c
CTP_DS3	CTP_VC4_16
CTP_STS1	CTP_VC4_64c
CTP_STS3	CTP_ETS1
CTP_STS3C	CTP_ETS3
CTP_STS12	CTP_ETS6
CTP_STS12C	CTP_ETS9
CTP_STS24C	CTP_ETS12
CTP_STS48	CTP_ETS24
CTP_STS48C	CTP_E3
CTP_STS192C	CTP_34MB
CTP_AU3	CTP_45MB
CTP_AU4	CTP_50MB
CTP_VC4_4	CTP_100MB
CTP_VC4_4c	

Optional Parameters

displayName	String	User-friendly name to represent the High Order CTP.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element.
className	String	Class name of the object to be created or found. Currently HOCTP is the only Class name for this API.
createContainment	Boolean	This is by default set to TRUE. It is recommended not to change the default. This facilitates the creation of the relationship between the CTP and underlying PTP.

Return Values

HOCTP	Reference to the High Order CTP if successfully created or found.
null	If API is not able to create or find High Order CTP object.

Dependency

- Object of type OpticalNetworkElement (with name “neName”) where the PTP containing this CTP resides.
- Object of type Equipment (with name/aid cardAid) containing the PTP (with name/aid ptpAid).
- Object of type PTP (with name/aid “ptpAid”) containing this CTP

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedVia	SubnetworkConnection	ConnectedTo	1:1	O	Y
LayeredOver	PTP	Underlying	1:1	M	Y
RoutedBy	HighOrder_Circuit, HighOrder_Trail	CTPsInRoute	1:1	M	Y

HighOrder_Circuit

makeHOCircuit (circuitName, circuitType, AendCTP, ZendCTP, ctpsInRoute, className = "*value*")

findHOCircuit (circuitName)

Mandatory Parameters

circuitName	String	Name of the Circuit used to uniquely identify it within OTM repository.
circuitType	String	Indicates the layer rate of the circuit (like STS3C, STS1, OC3, etc).
AendCTP	Object type of CTP	CTP object at one end of the HighOrder_Circuit.
ZendCTP	Object type of CTP	CTP object at the other end of the HighOrder_Circuit.
ctpsInRoute	List of objects of type CTP	CTPs through which HighOrder_Circuit is routed (excluding the CTPs at either end).

Optional Parameters

className	String	Class name of the object. Currently "HighOrder_Circuit" is the only Class name for this API.

Return Values

HighOrder_Circuit	Reference to the HighOrder_Circuit object if object is successfully created or found.
null	If API fails to create or find the HighOrder_Circuit object.

Dependency

- Object of type CTP - AendCTP created with its unique name/aid.
- Object of type CTP - ZendCTP created with its unique name/aid.
- Objects of type CTP - for all the CTPs in the route of the Circuit.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
aEnd	CTP	None	1:1	M	Y
zEnd	CTP	None	1:1	M	Y
CTPsInRoute	CTP	RoutedBy	1:N	M	Y
LayeredOver	HighOrder_Trail	Underlying	N:N	O	N
LayeredOver	DropSideTopologicalLink	Underlying	N:N	M	N
LayeredOver	TopologicalLinkGroup	Underlying	N:N	O	N
LayeredOver	RingProtectionGroup	Underlying	N:N	O	N
LayeredOver	SubnetworkConnection	Underlying	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N

HighOrder_Trail

makeHOTrail (trailName, trailType, ctpsInRoute, className = "*value*")

findHOTrail (trailName)

Mandatory Parameters

trailName	String	Name of the HighOrder_Trail used to uniquely identify it within the OTM repository.
trailType	String	Indicates the layer rate of the trail (like STS3C, OC3, etc.).
ctpsInRoute	List of objects of type CTP	CTPs through which HighOrder_Trail is routed.

Optional Parameters

className	String	Class name of the object to be created or found. Currently, HighOrder_Trail is the only Class name for this API.

Return Values

HighOrder_Trail	Reference to the HighOrder_Trail object, if the object is successfully created or found.
null	If API fails to create or find the HighOrder_Trail object.

Dependency

- Objects of type CTP - for all the CTPs in the route of the Trail
- Objects of type CTP for the two ends of the Circuit must be in the list of CTPs in route of the Trail.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
CTPsInRoute	CTP	RoutedBy	1:N	M	Y
Underlying	HighOrder_Circuit	LayeredOver	1:N	M	N
LayeredOver	TopologicalLink	Underlying	N:N	O	N
LayeredOver	TopologicalLinkGroup	Underlying	N:N	O	N
LayeredOver	RingProtectionGroup	Underlying	N:N	O	N
LayeredOver	SubnetworkConnection	Underlying	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N

SubnetworkConnection

makeSubnetworkConnection (emsName, sncName, AendCTP, ZendCTP, listOfTls, displayName = "*value*")

findSubnetworkConnection (emsName, sncName)

Mandatory Parameters

emsName	String	Name of the ems used to manage the SNC (Sub Network Connection).
sncName	String	Name of the SNC used to uniquely identify it within the OTM repository.
AendCTP	Object type of CTP	CTP object at one end of the SubnetworkConnection.
ZendCTP	Object type of CTP	CTP object at the other end of the SubnetworkConnection.
listOfTls	List of objects of type TopologicalLink	Lists all the TopologicalLink objects in the SubnetworkConnection.

Optional Parameters

displayName	String	User-friendly name to represent the SubnetworkConnection in the console.

Return Values

SubnetworkConnection	Reference to the SubnetworkConnection object if the object is successfully created or found.
null	If API fails to create or find the SubnetworkConnection object.

Dependency

- Object of type EMS (with the name "emsName") managing the SubnetworkConnection.
- Object of type CTP - AendCTP (with its unique name/aid).
- Object of type CTP - ZendCTP (with its unique name/aid) forming the two ends of the SubnetworkConnection.
- Objects of type TopologicalLink for all the TopologicalLinks in the SubnetworkConnection.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ManagedBy	EMS	ManagesSNC	N:1	O	Y
aEnd	CTP	None	1:1	M	Y
zEnd	CTP	None	1:1	M	Y
ConnectedTo	CTP	ConnectedVia	1:N	M	Y
LayeredOver	TopologicalLink	Underlying	N:N	O	Y
LayeredOver	TopologicalLink Group	Underlying	N:N	O	N
LayeredOver	RingProtectionGroup	Underlying	N:N	O	N
Underlying	HighOrder_Circuit	LayeredOver	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N

TopologicalLinkGroup

makeTLGroup (workingTl, protectedTl, pgType, bidirectional = "*value*", displayName = "*value*", className = "*value*")

make2FBLSR (aEndNode, zEndNode, workingTls, protectedTls, displayName, className)

makeUpsrTlg (aEndPTP, zEndPTP, workingTls, protectedTls, displayName, className)

findUpsrTlg (aEndPTP, zEndPTP)

Mandatory Parameters

Name	Type	Description
workingTl	Object of type TopologicalLink	Working TopologicalLink in the group of TopologicalLinks.
protectedTl	Object of type TopologicalLink	Protected TopologicalLink in the group of TopologicalLinks.
pgType	Enum	Enum of Type Optical_GlobalDefs::pg_type_e Category of protection group from the following types: PGT_EQP_1_FOR_N PGT_UPSR PGT_MSP_1_PLUS_1 PGT_NA PGT_MSP_1_FOR_N PGT_SNCP PGT_2_FIBER_BLSR PGT_UNKNOWN PGT_4_FIBER_BLSR **Note:** The pgType is automatically set to PGT_2_FIBER_BLSR when make2FBLSR is used.
aEndNode	String	Name/tid of OpticalNetworkElement at one end where the circuit enters/leaves the physical ring.
zEndNode	String	Name/tid of OpticalNetworkElement at other end where the circuit enters/leaves the physical ring.
workingTls	List of objects of type TopologicalLink	TopologicalLinks forming the working route in 2F-BLSR ring.
protectedTls	List of objects of type TopologicalLink	TopologicalLinks forming the alternate route in 2F-BLSR ring.
AendPTP	Object Type of PTP	PTP object at Aend of the UPSR ring.
ZendPTP	Object of Type PTP	PTP object at Aend of the UPSR ring.

Optional Parameters

bidirectional	Boolean	Indicates directionality and by default set to TRUE.
displayName	String	User-friendly name to represent the TopologicalLinkGroup in the console.
className	String	Class name of the object to be created or found. Currently, TopologicalLinkGroup is the only Class name for this API.

Return Values

TopologicalLinkGroup	Reference to the TopologicalLinkGroup object if the object is successfully created or found.
null	If API fails to create or find the TopologicalLinkGroup object.

Dependency

- Object of type TopologicalLink - workingTl (with its unique name/aid).
- Object of type TopologicalLink - protectedTl (with its unique name/aid) forming the TopologicalLinkGroup.
- Object of type PTP - aEndPTP (with its unique name/aid).
- Object of type PTP - zEndPTP (with its unique name/aid).

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOf	TopologicalLink	PartOf*	1:N	M	Y
Underlying	HighOrder_Circuit, HighOrder_Trail, SubnetworkConnection	LayeredOver	N:N	O	N
PartOfRPG	RingProtectionGroup	ComposedOfTLG	N:N	O	Y
Route	TopologicalLink	None	1:N	M	Y
AlternateRoute	TopologicalLink	None	1:N	M	Y

BlackBoxTopologicalLink

This API is used to make a black box topological link object within the OTM repository.

makeBlackBoxTL (topologicalLinkName, AendPTP, ZendPTP, displayName, className, emsName, subnetworkName, BBName, clfi)

findBlackBoxTL (topologicalLinkName, emsName)

Mandatory Parameters

topologicalLinkName	String	Name of the Black Box TopologicalLink used to uniquely identify it within OTM repository.
AendPTP	Reference to PTP Object	PTP Object at one end of the Black Box TopologicalLink.
ZendPTP	Reference to PTP Object	PTP Object at other end of the Black Box TopologicalLink

Optional Parameters

displayName	String	User-friendly name of the Object
className	String	Class name of the object. Currently "BBTopologicalLink" is the only Class name for this API.
emsName	String	Name of the EMS to which SONET/SDH Domain Manager will be connecting to. It has to be a unique name.
subnetworkName Name	String	Name of the subnetwork to which the black box belongs.
BBName	String	Name of the black box.
clfi	String	Name of the Facility to which the black box belongs.

Return Values

BBTopologicalLink	Reference to "BBTopologicalLink" object if created or Found
NULL	If API is not able to create or find "BBTopologicalLink" object

Dependency

Following objects should be created before creating the makeBlackBoxTL object.

- Object of type EMS with the name "emsName" if emsName provided.
- Object of type PTP - AendPTP created with its unique name/aid
- Object of type PTP - ZendPTP created with its unique name/aid

Relationships

Refer to “TopologicalLink” on page 23.

BlackBoxDropSideTopologicalLink

makeBlackBoxDSTL (dropSidePTP, isWestPTP, displayName, className, emsName, BBName, clfi)

findBlackBoxDSTL (dropSidePTP)

Mandatory Parameter

dropSidePTP	Object Type of PTP	PTP Object at the drop side of the Circuit (forming the Black Box DropSideTopologicalLink).
isWestPTP	Boolean	Relative indication whether the (drop side) PTP is on the East or West of the Circuit.

Optional Parameters

displayName	String	User-friendly name of the object
className	String	Class name of the object. Currently, "BBDropSideTopologicalLink" is the only Class name for this API.
emsName	String	Name of the EMS to which SONET/SDH Domain Manager will be connecting to. It has to be a unique name.
BBName	String	Name of the black box.
clfi	String	Name of the Facility to which the black box belongs.

Return Values

BBDropSideTopologicalLink	Reference to "BBDropSideTopologicalLink" object if created or found.
NULL	If API is not able to create or find "BBDropSideTopologicalLink" object

Dependency

Following objects should be created before creating the BBDropSideTopologicalLink object.

- "Object of type EMS with the name "emsName" if emsName provided.
- "Object of type PTP - dropSidePTP created with its unique name/aid

Relationships

Refer to “DropSideTopologicalLink” on page 24.

RingProtectionGroup (RPG)

makeRPGroup (AendPTP, ZendPTP, tlgs, tls, pgType = "*value*", displayName = "*value*", className = "*value*")

Mandatory Parameters

AendPTP	Object type of PTP	PTP object at one end of the RingProtectionGroup.
ZendPTP	Object type of PTP	PTP object at the other end of the RingProtectionGroup.
Tlgs	List of objects of type TopologicalLinkGroup	TopologicalLinkGroups that make up the RingProtectionGroup.
tls	List of objects of type TopologicalLink	List of protecting TopologicalLinks that back up the working TopologicalLinks in the tlgs described above.

Optional Parameters

pgType	Enum	Enum of type Optical_GlobalDefs::pg_type_e Category of protection group from the following types: PGT_2_FIBER_BLSR PGT_4_FIBER_BLSR PGT_NA PGT_2_FIBER_MS_SPRING PGT_4_FIBER_MS_SPRING PGT_SNCP PGT_UNKNOWN **Note:** The pgType default is PGT_4_FIBER_BLSR.
displayName	String	User-friendly name to represent the RingProtectionGroup in the console.
className	String	Class name of the object. Currently RingProtectionGroup is the only Class name for this API.

Return Values

RingProtectionGroup	Reference to the RingProtectionGroup object if the object is successfully created or found.
null	If API fails to create or find the RingProtectionGroup object.

Dependency

- Object of type PTP - AendPTP (with its unique name/aid).
- Object of type PTP - ZendPTP (with its unique name/aid) forming the two ends of RingProtectionGroup.
- Objects of type TopologicalLinkGroup forming the protected Ring route.
- Objects of type TopologicalLink that are protecting the TopologicalLinks forming the TopologicalLinkGroups (tlgs).

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOfTLG	TopologicalLinkGroup	PartOfRPG	N:N	M	Y
RouteProtectedBy	TopologicalLink	RouteProtectedFor	N:N	M	Y

EquipmentProtectionGroup

makeEqPGroup (neName, eqpts, ptps, displayName = "*value*", className = "*value*")

Mandatory Parameters

neName	String	Name/tid of OpticalNetworkElement in which the Equipment Protection is configured.
Eqpts	List of objects of type Equipment	Equipment or Card objects forming the Equipment Protection Group.
Ptps	List of objects of type PTP	List of PTP objects that are protected by the Equipment/Card objects.

Optional Parameters

displayName	String	User-friendly name to represent the EquipmentProtectionGroup in the console.
className	String	Class name of the object to be created or found. Currently, EquipmentProtectionGroup is the only Class name for this API.

Return Values

EquipmentProtectionGroup	Reference to the EquipmentProtectionGroup object if the object is successfully created or found.
null	If API fails to create or find the EquipmentProtectionGroup object.

Dependency

- Object of type OpticalNetworkElement where the EquipmentProtectionGroup is configured.
- Objects of type Equipment forming the EquipmentProtectionGroup.
- Objects of type PTP that are involved in the EquipmentProtectionGroup.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOf	Equipment	PartOfEquipmentProtectionGroup	1:N	M	Y
CrossConnects	PTP	CrossConnectedBy	1:N	M	Y
UnderlyingLogical	HighOrder_Circuit, HighOrder_Trail, SubnetworkConnection	LayeredOverPhysical	N:N	O	N

LogicalConnectionTPGroup

makeLCTPGroup (AendPTP, ZendPTP, workingCtps, protectedCtps, displayName = "*value*", className = "*value*")

Mandatory Parameters

AendPTP	Object of type PTP	PTP object at one end of the LogicalConnectionTPGroup.
ZendPTP	Object of type PTP	PTP object at the other end of the LogicalConnectionTPGroup.
workingCtps	List of objects of type CTP	List of working CTP objects in the route of the LogicalConnectionTPGroup.
protectedCtps	List of objects of type CTP	List of protecting CTP objects in the alternate route that protects the working CTP objects in the LogicalConnectionTPGroup.

Optional Parameters

displayName	String	User-friendly name to represent the LogicalConnectionTPGroup in the console.
className	String	Class name of the object to be created or found. Currently LogicalConnectionTPGroup is the only Class name for this API.

Return Values

LogicalConnectionTPGroup	Reference to the LogicalConnectionTPGroup object if object is successfully created or found.
null	If API fails to create or find the LogicalConnectionTPGroup object.

Dependency

- Object of type PTP - AendPTP (with its unique name/aid).
- Object of type PTP - ZendPTP (with its unique name/aid) forming the two ends of LogicalConnectionTPGroup.
- Objects of type CTP forming the LogicalConnectionTPGroup (in the route).
- Objects of type CTP that protect the LogicalConnectionTPGroup (in the alternate route).

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
Route	CTP	None	1:N	M	Y
AlternateRoute	CTP	None	1:N	M	Y
Underlying	HighOrder_Circuit, HighOrder_Trail, SubnetworkConnection	LayeredOver	N:N	O	N

CHAPTER 3
WDM Object Factory API

These APIs are intended for use within the WDM Topology server. They facilitate topology creation and modification for physical and logical network components in the OTM WDM Domain Manager. The API set is categorized into two parts: Physical components and Logical components.

This chapter contains the following information:

- WDM components and API summary 44
- WDM physical components 47
- WDM logical components 73

WDM components and API summary

This chapter describes all the APIs needed to create the object instances in the WDM Domain Manager's repository. Figure 2 on page 44 is a basic WDM network segment configuration. It has two end terminals with an in line amplifier (ILA). This figure is for reference and shows various physical and logical classes that can be instantiated in the repository of WDM Domain Manager for building the topology into the WDM Domain Manager using the APIs.

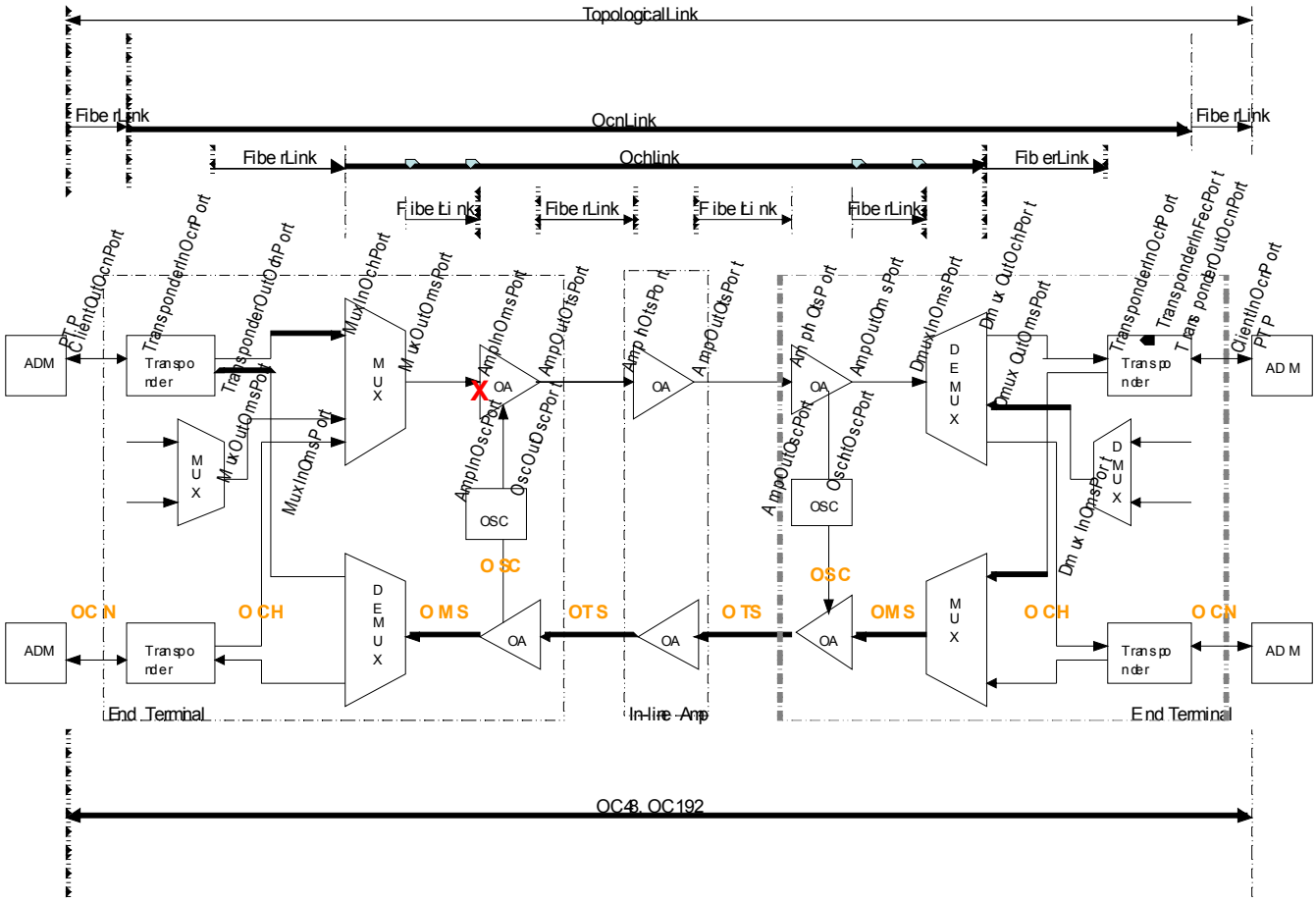

Figure 2 WDM network segment configuration

Table 2 on page 45 lists the WDM OTM components along with the APIs used to make and find that component. Also indicated is whether the component is physical hardware or a logical construction.

Note: API names for creating components start with “make” and API names for finding components already created and existing within OTM repository start with “find.” Some API names use the convention “makeN*xxx*” or “findN*xxx*.” The “N” emphasizes the fact that the native name (also known as the native AID) is used in forming the final component.name without undergoing any change or normalization.

Table 2 WDM components and API summary (page 1 of 2)

Component Name	API Name	Details	Entity Type	Class Used/Returned
EMS	makeEMSs	page 47	Physical	EMS
	findEms	page 49		
OpticalNetworkElement	makeOpticalNetworkElement	page 50	Physical	OpticalNetworkElement
	findOpticalNetworkElement	page 51		
Card	makeNCard	page 52	Physical	Card Amplifier Transponder OscCard
	makeNSubcard	page 53		
	findNCard	page 54		
Input Port	makeNInputPort **Note:** makeInputPort is deprecated. Use makeNInputPort instead.	page 55	Physical	TransponderInOcnPort TransponderInOchPort MuxInOchPort MuxInOmsPort DemuxInOmsPort AmpInOmsPort AmpInOtsPort AmpInOscPort OscInOscPort
	findNInputPort **Note:** findInputPort is deprecated. Use findNInputPort instead.	page 58		
Output Port	makeNOutputPort **Note:** makeOutputPort is deprecated. Use makeNOutputPort instead.	page 59	Physical	TransponderOutOcnPort TransponderOutOchPort DemuxOutOchPort MuxOutOmsPort DemuxOutOmsPort AmpOutOmsPort AmpOutOtsPort AmpOutOscPort OscOutOscPort
	findNOutputPort **Note:** findOutputPort is deprecated. Use findNOutputPort instead.	page 62		
FiberLink	makeFiberLink	page 63	Physical	FiberLink
	findFiberLink	page 65		
PassiveFiberLink	makePassiveFiberLink	page 66	Physical	FiberLink

Table 2 WDM components and API summary (page 2 of 2)

BlackBoxFiberLink	makeBlackFL	page 68	Physical	FiberLink
	findBlackFL	page 69		
PTP	makeNPTP, findNPTP **Note:** makePTP and findPTP are deprecated. Use makeNPTP and findNPTP instead.	page 70	Physical	PTP
TopologicalLink	makeTopologicalLink	page 72	Physical	TopologicalLink
LinkGroup	makeLinkGroup	page 73	Logical	LinkGroup
findLinkGroup	findLinkGroup	page 76	Logical	LinkGroup
OchLink	makeLogicalLink	page 77	Logical	OchLink
	findLogicalLink	page 78		
OcnLink	makeLogicalLink	page 77	Logical	OcnLink
	findLogicalLink	page 78		
ClientTrail	makeClientTrail	page 79	Logical	ClientTrail
	findClientTrail	page 81		
ClientCircuit	makeClientCircuit	page 82	Logical	ClientCircuit
	findClientCircuit	page 83		

Data layout

Under each component name, the following information is provided:

The object name to be created or found is a heading in the left column.

Under the heading is a description and the command showing the usage syntax including the order of parameter.

Mandatory Parameters

These parameters are required. Parameters are described in three columns:

Parameter name, Data type, Description

Optional Parameters

These parameters are not required. Parameters are described in three columns:

Parameter name, Data type, Description

Return Values

Lists the object types returned by the APIs. Return values are listed in two columns: Type of object reference returned, Description

Dependency

List of dependencies of the API on other objects.

Relationships

Lists of the relationships of this component to other components.

In the API Relation column the values are listed as follows:

Y* represents that the relationship is created as part of the API functionality of the *related class*, not the API of the class being discussed. For example, in makeEMS, the Manages OpticalNetworkElement relationship is created by the makeOpticalNetworkElement API, not by the makeEMS API.

Y represents that the relationship is created as part of the API functionality of the class in discussion itself. For example, in makeNCard, the PartOf OpticalNetworkElement is created within the makeNCard API.

N represents that the relationship must be created outside the API functionality by directly accessing the instance of that class. For example, after making an Amplifer Card using makeNCard, to create the ReverseAmp relationship to another Amplifier Card use the following ASL code:

```
AmpCard1Ref -> ReverseAmp = AmpCard2Ref
```

In the above example AmpCard1Ref and AmpCard2Ref are the references to the respective Amplifier Cards returned from the findNCard API.

WDM physical components

This section describes the APIs that enable creation and or alteration of managed objects that form the logical components of the underlying network.

makeEMS

This API is used to create an object representing the Element Management System (EMS) to which WDM Domain Manager will be connecting to receive all the events pertaining to various network entities.

makeEMS (emsName, displayName = "*value*", className = "*value*")

Mandatory Parameters

emsName	String	Name of the EMS to which the WDM Domain Manager will be connecting. Must be a unique name.

Optional Parameters

displayName	String	User-friendly name to represent the EMS in the console.
className	String	Class name of the object to be created. Currently EMS is the only Class name for this API.

Return Values

EMS	Reference to the EMS object if the object is successfully created.
null	If API fails to create the EMS object.

Dependency

None

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Manages	OpticalNetworkElement	ManagedBy	1:N	M	Y*
ManagesLink	FiberLink OchLink OcnLink	ManagedBy	1:N	M	Y*

findEMS

This API is used to find the EMS object in the repository of WDM Domain Manager.

findEMS (emsName)

Mandatory Parameters

emsName	String	Name of the EMS that needs to be found.

Return Values

EMS	Reference to the EMS object if the object is found.
null	If API fails to find the EMS object.

Dependency

None

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Manages	OpticalNetworkElement	ManagedBy	1:N	M	Y*
ManagesLink	FiberLink OchLink OcnLink	ManagedBy	1:N	M	Y*

makeOpticalNetworkElement

This API is used to create an object representing the WDM Network Element.

makeOpticalNetworkElement (tid, emsName, displayName = "*value*", className = "*value*")

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
emsName	String	Name of the EMS to which OpticalNetworkElement belongs.

Optional Parameters

displayName	String	User-friendly name to represent the OpticalNetworkElement in the console.
className	String	Class name of the object to be created. Currently OpticalNetworkElement is the only Class name for this API.

Return Values

OpticalNetworkElement	Reference to the OpticalNetworkElement object if object is successfully created.
null	If API fails to create the OpticalNetworkElement object.

Dependency

EMS object should be there for the OpticalNetworkElement object to be created.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
ComposedOf	Card	Part of	1:N	M	Y*
ManagedBy	EMS	Manages	N:1	M	Y

findOpticalNetworkElement

This API is used to find the Optical Network Element object with the given Target Identifier (tid).

findOpticalNetworkElement (tid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.

Return Values

OpticalNetworkElement	Reference to the OpticalNetworkElement object if the object is found.
null	If the API fails to find the OpticalNetworkElement object with the given target identifier (tid).

Dependency

None

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
ComposedOf	Card	Part of	1:N	M	Y*
ManagedBy	EMS	Manages	N:1	M	Y

makeNCard

This API is used to create an object representing the Card/Circuit Pack/Equipment installed in the WDM Network Element. This API can be used to create objects of various classes representing a variety of cards:

- Card
- Amplifier
- OscCard
- ClientCard
- Transponder

makeNCard (tid, cardAid, displayName = "*value*", className = "*value*")

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Card object in a given network.
cardAid	String	Name of the Card object within the given network element.

Optional Parameters

displayName	String	User-friendly name to represent the Card object in the console.
className	String	Class name of the Card object to be created. See the above list.

Return Values

Card	Reference to the Card object if the object is successfully created. See the above list.
null	If API fails to create the Card object.

Dependency

OpticalNetworkElement object should exist for the Card object to be created.

Relationships

Card, Transponder, OscCard, and ClientCard

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
PartOf	OpticalNetworkElement	ComposedOf	N:1	M	Y
Realizes	All Port Classes	RealizedBy	1:N	M	Y*

Amplifier

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
ReverseAmp	Amplifier	ReverseAmp	1:1	O	N
OscReportedBy	Amplifier	ReportsOscFor	1:1	O	N
ReportsOscFor	Amplifier	OscReportedBy	1:N	O	N

makeNSubcard

This API is used to create a daughter card for the mother card created using the makeNCard API. After the daughter card is created, the API will automatically set the daughter-mother card relationship. To be able to create this relationship correctly, the cardAid representing the mother card in the makeNSubcard API call must be identical to the cardAid in the makeNCard API call.

makeNSubcard (tid, cardAid, subcardAid, displayName = "*value*", className = "*value*")

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Card object in a given network.
cardAid	String	Name of the mother card object within the given network element.
subcardAid	String	Name of the daughter card object within the given network element.

Optional Parameters

displayName	String	User-friendly name to represent the Subcard object in the console.
className	String	Class name of the Subcard object to be created. See the above list.

Return Values

Subcard	Reference to the Subcard object if the object is successfully created. See the above list.
null	If API fails to create the Subcard object.

Dependency

Card object should exist for the Subcard object to be created.

Relationships

Card, Transponder, OscCard, and ClientCard

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
PartOfCard	Card	ComposedOf	N:1	M	Y
Realizes	All Port Classes	RealizedBy	1:N	M	Y

Amplifier

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
ReverseAmp	Amplifier	ReverseAmp	1:1	O	N
OscReportedBy	Amplifier	ReportsOscFor	1:1	O	N
ReportsOscFor	Amplifier	OscReportedBy	1:N	O	N

findNCard

This API is used to find the Card object with the given tid and AID. Card objects can be any of the various classes representing a variety of cards:

- Card
- Amplifier
- OscCard
- ClientCard
- Transponder

findNCard (tid, cardAid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
cardAid	String	Uniquely identifies the Card in a given network element.

Return Values

Card	Reference to the Card object if the object is found. See the above list.
null	If API fails to find the Card object.

Dependency

None

Relationships

Card, Transponder, OscCard, and ClientCard

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
PartOf	OpticalNetworkElement	ComposedOf	N:1	M	Y
Realizes	All Port Classes	RealizedBy	1:N	M	Y*

Amplifier

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
ReverseAmp	Amplifier	ReverseAmp	1:1	O	N
OscReportedBy	Amplifier	ReportsOscFor	1:1	O	N
ReportsOscFor	Amplifier	OscReportedBy	1:N	O	N

makeNInputPort

Note: makeInputPort is deprecated. Use makeNInputPort instead.

This API is used to create an object representing the physical input port in the WDM Network Element. There is no such class as InputPort. The term is used symbolically to represent any of the following Classes that makeNInputPort can create:

- TransponderInOcnPort
- MuxInOchPort
- MuxInOmsPort
- AmpInOmsPort
- AmpInOtsPort
- DemuxImOmsPort
- AmpInOscPort
- TransponderInOchPort
- OscInOscPort

makeNInputPort (tid, cardAid, portAid, className, displayName = "*value*")

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
cardAid	String	Uniquely identifies the Card in a given network element.
portAid	String	Uniquely identifies the Input Port in the Card network element.
className	String	Class name of the object to be created. See the above list.

Optional Parameters

displayName	String	User-friendly name to represent the Input Port object in the console.

Return Values

Input Port	Reference to the Input Port object if the object is successfully created. See the above list.
null	If API fails to create the Input Port object.

Dependency

Corresponding Card object should be created before creating the Input Port object.

Relationships

Port, MuxInOchPort, MuxInOmsPort, AmpInOmsPort, AmpInOscPort, OscInOscPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Feeds	OchLink OcnLink TopologicalLink FiberLink	FedBy	1:1	M	Y*
FedBy	OchLink OcnLink TopologicalLink FiberLink	Feeds	1:1	M	Y*
NextPort	All Port Classes	PrevPort	1:1	M	makeFiberLink makeLogicalLink
PrevPort	All Port Classes	NextPort	1:1	M	makeFiberLink makeLogicalLink
RealizedBy	Card	Realizes	N:1	M	Y
OwnedBy	ClientTrail	AendInport ZednInport AendOutport ZendOutport	1:1	O	Y*

TransponderInOcnPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
OutOchPorts	TransponderOutOchPort	InOcnPort	1:N	M	N

TransponderInOchPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
OutOcnPort	TransponderOutOcnPort	InOchPorts	N:1	M	N
ReversePort	TransponderOutOchPort	ReversePort	1:1	M	N

DeMuxInOmsPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Passes	OchLink	PassesThrough	1:N	M	N

AmpInOmsPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
OutOtsPort	AmpOutOtsPort	InOmsPort	1:1	M	N
Passes	OchLink	PassesThrough	1:N	M	N

AmpInOtsPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
OutOmsPort	AmpOutOmsPort	InOtsPort	1:1	M	N

findNInputPort

This API is used to find an object representing the physical input port in the WDM Network Element. There is no such class as InputPort. The term is used symbolically to represent any of the following Classes to which findNInputPort can return a reference:

- TransponderInOcnPort
- MuxInOchPort
- MuxInOmsPort
- AmpInOmsPort
- AmpInOtsPort
- DemuxImOmsPort
- AmpInOscPort
- TransponderInOchPort
- OscInOscPort

findNInputPort (tid, portAid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
portAid	String	Uniquely identifies the Input Port in the Network Element.

Return Values

Input Port	Reference to the Input Port object if the object is successfully created. See the above list.
null	If API fails to find the Input Port object.

Dependency

None

Relationships

See the Relationships section of "makeNInputPort" on page 55.

makeNOutputPort

This API is used to create an object representing the physical output port in the WDM Network Element. There is no such class as OutputPort. The term is used symbolically to represent any of the following Classes that makeNOutputPort can create:

- TransponderOutOchPort
- MuxOutOmsPort
- AmpOutOtsPort
- AmpOutOmsPort
- AmpOutOscPort
- OscOutOscPort
- DemuxOutOmsPort
- TransponderOutOcnPort
- DemuxOutOchPort

makeNOutputPort (tid, portAid, className, displayName = "*value*")

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
portAid	String	Uniquely identifies the Output Port in the Network Element.
className	String	Class name of the Output Port object to be created.

Optional Parameters

displayName	String	User-friendly name to represent the Output Port in the console.

Return Values

Output Port	Reference to the Output Port object if the object is successfully created. See the above list.
null	If API fails to create the Output Port object.

Dependency

The corresponding "Card" Object should be created before creating the Output Port object.

Relationships

Port, DemuxOutOmsPort, DemuxOutOshPort, AmpOutOscPort, OscOutOscPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Feeds	OchLink OcnLink TopologicalLink FiberLink	FedBy	1:1	M	Y*
FedBy	OchLink OcnLink TopologicalLink FiberLink	Feeds	1:1	M	Y*
NextPort	All Port Classes	PrevPort	1:1	M	Y* makeFiberLink makeLogicalLink
PrevPort	All Port Classes	NextPort	1:1	M	Y* makeFiberLink makeLogicalLink
RealizedBy	Card	Realizes	N:1	M	Y
OwnedBy	ClientTrail	AendInport ZednInport AendOutport ZendOutport	1:1	O	Y*

TransponderOutOcnPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
InOchPorts	TransponderInOchPort	OutOcnPort	1:N	M	N

TransponderOutOchPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
InOcnPort	TransponderInOcnPort	OutOchPorts	N:1	M	N
ReversePort	TransponderInOchPort	ReversePort	1:1	M	N

MuxOutOmsPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Passes	OchLink	PassesThrough	1:N	M	N

AmpOutOmsPort

Relationship	Related Class	Reverse Relationship	Cardinalit y	Mandatory or optional	API Relation
InOtsPort	AmpInOtsPort	InOmsPort	1:1	M	N
Passes	OchLink	PassesThrough	1:N	M	N

AmpOutOtsPort

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
InOmsPort	AmpInOmsPort	OutOtsPort	1:1	M	N

findNOutputPort

Note: findOutputPort is deprecated. Use findNOutputPort instead.

This API is used to find an object representing the physical output port in the WDM Network Element. There is no such class as OutputPort. The term is used symbolically to represent any of the following Classes to which findNOutputPort can return a reference:

- TransponderOutOchPort
- MuxOutOmsPort
- AmpOutOtsPort
- AmpOutOmsPort
- AmpOutOscPort
- OscOutOscPort
- DemuxOutOmsPort
- TransponderOutOcnPort
- DemuxOutOchPort

findNOutputPort (tid, portAid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
portAid	String	Uniquely identifies the Output Port in the Network Element.

Return Values

Output Port	Reference to the Output Port object if the object is found. See the above list.
null	If API fails to find the Output Port object.

Dependency

None

Relationships

See the Relationships Created by Other Components section of "makeNInputPort" on page 55.

makeFiberLink

This API is used to create the fiber-optic link between the given Output Port and Input Port.

makeFiberLink (emsName, aEndRef, zEndRef, displayName = "*value*", className = "*value*")

Mandatory Parameters

emsName	String	Uniquely identifies the EMS to which this Fiber Link belongs.
aEndRef	Object reference	A reference to the aEnd of the FiberLink. aEnd must be one of the Output Port types.
zEndRef	Object reference	A reference to the zEnd of the FiberLink. zEnd must be one of the Input Port types.

Optional Parameters

displayName	String	User-friendly name to represent the FiberLink object in the console.
className	String	Class name of the FiberLink object to be created Currently FiberLink is the only class name used for this API.

Return Values

FiberLink	Reference to the FiberLink object if object is successfully created.
null	If API fails to create the FiberLink object.

Dependency

Following objects should be created before creating the FiberLink object.

- EMS
- aEnd Port
- zEnd Port

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API relation
Underlying [1]	OchLink or OcnLink or TopologicalLink	LayeredOver	1:N	O	N
DownstreamLink	FiberLink	UpstreamLink	1:N	O	N
UpstreamLink	FiberLink	DownstreamLink	1:N	O	N
ManagedBy	EMS	ManagesLink	N:1	M	Y
Feeds	All Input Port Classes	FedBy	1:1	M	Y
FedBy	All Output Port Classes	Feeds	1:1	M	Y

1. FiberLink must have a LayeredOver/Underlying relationship with *only* those Links (OchLink, OcnLink or Topological Link) that are *directly layered over* the FiberLink. For example, a FiberLink that has a LayeredOver/Underlying relationship with an OchLink must not have any Underlying or LayeredOver relationship with the OcnLink that has a LayeredOver/Underlying relationship with the OchLink.

findFiberLink

This API is used to find the FiberLink between the given Output Port and Input Port.

findFiberLink (aEndRef, zEndRef)

Mandatory Parameters

aEndRef	Object reference	A reference to the aEnd of the FiberLink. aEnd must be one of the Output Port types.
zEndRef	Object reference	A reference to the zEnd of the FiberLink. zEnd must be one of the Input Port types.

Return Values

FiberLink	Reference to the FiberLink object if the object is found.
null	If the API fails to find the FiberLink object.

Dependency

None

Relationships

See the Relationships Created by Other Components section of "makeFiberLink" on page 63.

makePassiveFiberLink

This API is used to create a PassiveFiberLink object.

```
MR_Object makePassiveFiberLink (in string emsName,
                in OutputPort aEndRef,
                  in InputPort zEndRef,
                  in string displayName = "",
                  in string className = "")
    "Construct and return a PassiveFiberLink object given the aEnd and zEnd. "
    "If the object already exists, it is returned with a possibly "
    "modified DisplayName. If the displayName parameter is not "
    "specified then the value of the PassiveFiberLinkName "
    "parameter is used as the DisplayName. ";
```

Mandatory Parameter

emsName	String	It is a string which uniquely identifies the EMS to which this Fiber Link belongs.
aEndRef	Object Reference	A reference to the aEnd of the Fiber Link. aEnd must be one of the Output Port types.
zEndRef	Object Reference	A reference to the zEnd of the Fiber Link. zEnd must be one of the Input Port types.

Optional Parameters

displayName	String	User-friendly name to represent the PassiveFiberLink object in the console.
className	String	Class name of the PassiveFiberLink object to be created. PassiveFiberLink is the only class name used for this API.

Return Values

FiberLink	Reference to the PassiveFiberLink object if API is successfully created.
NULL	If API fails to create the PassiveFiberLink object.

Dependency

Following objects should be created before creating the BBFiberLink object.

- EMS
- aEnd Port
- zEnd Port

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Underlying	OchLink, OcnLink, FiberLink, TopologicalLink, PassiveFiberLink	LayeredOver	1:N	O	N
DownstreamLink	FiberLink PassiveFiberLink	UpstreamLink	1:N	O	N
UpstreamLink	FiberLink PassiveFiberLink	DownstreamLink	1:N	M	Y
ManagedBy	EMS	ManagesLink	N:1	M	Y
Feeds	InputPort	FedBy	1:1	M	
FedBy	OutputPort	Feeds	1:1	M	Y

makeBlackFL

This API is used to create the Black Box fiber Link between the given Output port and Input Port.

makeBlackFL (emsName, aEndRef, zEndRef, displayName, className, BBName, clfi)

Mandatory Parameter

emsName	String	It is a string which uniquely identifies the EMS to which this Black Box Fiber Link belongs.
aEndRef	Object Reference	It is a reference to the zEnd of the Black Box Fiber Link. zEnd shall be one of the Input Ports.
zEndRef Object	Object Reference	It is a reference to the zEnd of the Black Box Fiber Link. zEnd shall be one of the Input Ports.

Optional Parameters

displayName	String	User-friendly name of the Object.
className	String	Class name of the object. "BBFiberLink" is the only Class name used for this API.
BBName	String	Name of the black box.
clfi	String	Name of the Facility to which the black box belongs

Return Values

FiberLink	Reference to the BBFiberLink object if the API is successful in creating the BBFiberLink object.
NULL	If the API fails to create the BBFiberLink Object.

Dependency

The following objects should be created before creating the BBFiberLink object.

- EMS
- aEnd Port
- zEnd Port

findBlackFL

This API is used to find the black box fiber link between the given Output port and Input Port.

findBlackFL (aEndRef, zEndRef)

Mandatory Parameter

aEndRef	Object Reference	It is a reference to the aEnd of the BBFiberLink. aEnd shall be one of the Output Ports.
zEndRef	Object Reference	It is a reference to the zEnd of the BBFiberLink. zEnd shall be one of the Input Ports.

Return Values

BBFiberLink	Reference to the BBFiberLink object if the API is successful in finding the BBFiberLink object.
NULL	If the API fails to find the BBFiberLink Object.

Dependency

None

Relationships

None

makeNPTP

The Topology Adapter needs to invoke the makeNPTP API (along with the makeTopologicalLink API) to complete the topology. Essentially, makeNPTP will replicate the PTP object existing in the SONET/SDH server.

The makeNPTP API needs to be invoked to instantiate the SONET/SDH PTP in WDM to create an association between WDM and SONET/SDH servers and provide the correlation between failures occurring in the WDM topology causing failures in the SONET/SDH topology.

During the invocation of makeNPTP two other objects are automatically constructed in the WDM topology:

- ClientInOcnPort
- ClientOutOcnPort

Note: Relationship between these objects and corresponding PTPs are automatically created in the makeNPTP API, but a relationship needs to be created between these ports and the corresponding FiberLinks connecting them to Transponder Ports.

makeNPTP (neneName, ptpAid, cardAid, displayName = "*value*", className = "*value*", relateEquipment = "*value*")

findNPTP (neneName, ptpAid)

Note: findPTP is deprecated. Use findNPTP instead.

Mandatory Parameters

neName	String	Name of the Network Element in which the PTP resides.
cardAid	String	Access Identifier for the Equipment/Card within the network element to which the PTP belongs. Uniquely identifies the Equipment/Card object within the OTM repository.
ptpAid	String	Access Identifier for the PTP within the network element. Uniquely identifies the PTP within the OTM repository

Optional Parameters

displayName	String	User-friendly name to represent the PTP in the console.
className	String	Class name of the object to be created or found. Currently, PTP is the only Class name for this API.
relateEquipment	Boolean	By default, this is set to "TRUE." It is recommended not to change the default. This facilitates creating the relationship between the PTP and the Equipment/Card containing the PTP.

Return Values

PTP	Reference to the PTP object if created or found.
null	If the API is not able to create or find the PTP.

Dependency

- Object of type OpticalNetworkElement with the name "neName"

- Object of type Equipment/Card with the name "cardAid"

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
aEnd	TopologicalLink	aEnd	1:1	M	Y*
zEnd	TopologicalLink	zEnd	1:1	M	Y*
InPort	ClientInOcnPort	PartOfBDPort	1:1	M	Y
OutPort	ClientInOcnPort	PartOfBDPort	1:1	M	Y
OcnInPort	ClientInOcnPort	PartOfBDPort	1:1	M	Y
OcnOutPort	ClientInOcnPort	PartOfBDPort	1:1	M	Y

ClientInOcnPort

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
FedBy	FiberLink	Feeds	1:1	M	Y*
PrevPort	TransponderOutOcnPort	NextPort	1:1	M	Y* makeFiber Link
PartOfBDPort	PTP	InPort	1:1	M	Y*

ClientOutOcnPort

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
Feeds	FiberLink	FedBy	1:1	M	Y*
NextPort	TransponderInOcnPort	PrevPort	1:1	M	Y* makeFiber Link
PartOfBDPort	PTP	OutPort	1:1	M	Y*

TopologicalLink

The Topology Adapter needs to invoke the makeTopologicalLink API (along with the makeNPTP API) to complete the topology. Essentially, makeTopologicalLink will replicate the TopologicalLink object existing in the SONET/SDH server.

The makeTopologicalLink API needs to be invoked to instantiate the SONET/SDH Topological Link in WDM to create an association between WDM and SONET/SDH servers and provide the correlation between failures occurring in the WDM topology causing failures in the SONET/SDH topology.

makeTopologicalLink (emsName = "*value*", topologicalLinkName, AendPTP, ZendPTP, displayName = "*value*", className = "*value*")

findTopologicalLink (emsName, topologicalLinkName)

Mandatory Parameters

topologicalLinkName	String	Name of the TopologicalLink used to uniquely identify it within OTM repository.
emsName	String	Name of the EMS to which SONET/SDH Domain Manager will be connecting. It must be a unique name.
AendPTP	Reference to PTP	The PTP object at one end of the TopologicalLink
ZendPTP	Reference to PTP	The PTP object at the other end of the TopologicalLink

Optional Parameters

displayName	String	User-friendly name to represent the TopologicalLink in the console.
className	String	Class name of the object to be created or found. Currently TopologicalLink is the only Class name for this API.

Return Values

EMS	Reference to the TopologicalLink if successfully created or found.
null	If the API is not able to create or find the TopologicalLink object.

Dependency

- Object of type EMS with the name "emsName" if emsName provided
- Object of type PTP - AendPTP created with its unique name/aid
- Object of type PTP - ZendPTP created with its unique name/aid

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
LayeredOver	FiberLink OcnLink ClientTrail	Underlying	1:N	O	N
ManagedBy	EMS	ManagesLink	N:1	M	Y
aEnd	PTP	ToplogicalLink	1:1	M	Y
zEnd	PTP	TopologicalLink	1:1	M	Y

WDM logical components

makeLinkGroup

This API is used to create the Link Group containing either OcnLinks or OchLinks. A Link Group must not mix OcnLinks with OchLinks.

makeLinkGroup (emsName, workingLinks, protectingLinks, displayName = "*value*", className = "*value*")

Mandatory Parameters

emsName	String	Uniquely identifies the EMS to which this Link Group belongs.
workingLinks	List	A list of links that will form the working part of the Link Group.
protectingLinks	List	A list of links that would form the protecting part of the Link Group.

Optional Parameters

displayName	String	User-friendly name to represent the Link Group object in the console.
className	String	Class name of the Link Group object to be created. Currently LinkGroup is the only class name to be used for this API.

Return Values

LinkGroup	Reference to the LinkGroup object if the object is successfully created.
null	If the API fails to create the LinkGroup object.

Dependency

Following objects should be created before creating LinkGroup Object.

- EMS
- OcnLinks
- OchLinks

Relationships Created within API

ComposedOf/PartOf: This relationship is created within the API and is established between the OcnLink or OchLink objects and the LinkGroup object created using this API.

Route: This relationship is created within makeLinkGroup and contains working OcnLinks or OchLinks.

AlternateRoute: This relationship is created within makeLinkGroup and contains protecting OcnLinks or OchLinks.

Relationships

LinkGroup

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Route	OcnLink OchLink		1:N	M	Y
AlternateRoute	OcnLink OchLink		1:N	M	Y
ComposedOf	OchLink OcnLink		1:N	M	Y

OchLink

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Underlying	OcnLink	LayeredOver	1:N	O	N
LayeredOver	FiberLink	Underlying	1:N	O	N
ManagedBy	EMS	ManagesLink	N:1	M	Y
PassesThrough	MuxOutOmsPort DemuxInOmsPort AmpInOmsPort AmpOutOmsPort	Passes	N:1	O	N
Feeds	DemuxOutOchPort	FedBy	1:1	M	Y
FedBy	MuxInOchPort	Feeds	1:1	M	Y

OcnLink

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Underlying	TopologicalLink	LayeredOver	1:N	O	N
LayeredOver	FiberLink OchLink	Underlying	1:N	O	N
ManagedBy	EMS	ManagesLink	N:1	M	Y
Feeds	TransponderOutOcnPort	FedBy	1:1	M	Y
FedBy	TransponderInOcnPort	Feeds	1:1	M	Y

findLinkGroup (linkGrpName)

This API is used to find the LinkGroup object with the given linkGrpName.

findLinkGroup (linkGrpName)

Mandatory Parameters

linkGrpName	String	Name of the LinkGroup used to uniquely identify it within WDM repository.

Return Values

linkGroup	Reference to the LinkGroup object if successfully found.
null	If API fails to find the LinkGroup object.

Dependency

None

Relationships

See the Relationships section of "makeLinkGroup" on page 73.

makeLogicalLink

This API is used to create the Logical Link between the given Output Port and Input Port. Logical Links can be one of the following types:

- OchLink
- OcnLink

makeLogicalLink (emsName, aEndRef, zEndRef, displayName = "*value*", className = "*value*")

Mandatory Parameters

emsName	String	Uniquely identifies the EMS to which this Logical Link belongs.
aEndRef	Object reference	A reference to the aEnd of the Logical Link. aEnd must be one of the Input Port types.
zEndRef	Object reference	A reference to the zEnd of the Logical Link. zEnd must be one of the Ouput Port types.

Optional Parameters

displayName	String	User-friendly name to represent the Logical Link object in the console.
className	String	Class name of the Logical Link object to be created. Currently either OchLink or OcnLink are the class names used for this API.

Return Values

OcnLink or OchLink	Reference to the Logical Link object if the object is successfully created.
null	If API fails to create the Logical Link object.

Dependency

Following objects should be created before creating the Logical Link object.

- EMS
- aEnd Port
- zEnd Port

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
Feeds	InputPort	FedBy	1:1	M	Y
FedBy	OutputPort	Feeds	1:1	M	Y*
ManagesLink	EMS	ManagedBy	N:1	M	Y
NextPort	aEndPort	PrevPort	1:1	M	Y
PrevPort	zEndPort	NextPort	1:1	M	Y*

NextPort/PrevPort: This relationship is created within this API and is established between the InputPort and OutputPort. aEndPort has the relationship NextPort to zEndPort, and zEndPort has the relationship PrevPort to aEndPort.

findLogicalLink

This API is used to find the Logical Link between the given Output Port and Input Port. Logical Links can be one of the following types:

- OchLink
- OcnLink

findLogicalLink (aEndRef, zEndRef)

Mandatory Parameters

aEndRef	Object reference	A reference to the aEnd of the Logical Link. aEnd must be one of the Input Port types.
zEndRef	Object reference	A reference to the zEnd of the Logical Link. zEnd must be one of the Output Port types.

Return Values

OcnLink or OchLink	Reference to the Logical Link object if the object is found. See the above list.
null	If the API fails to find the Logical Link object.

Dependency

None

Relationships

None

makeClientTrail

This API is used to create the Client Trail for a given WDM network segment.

makeClientTrail (emsName, trailName, aEndInputPort, zEndOutputPort, zEndInputPort, aEndOutputPort, className = "*value*")

Mandatory Parameters

emsName	String	Uniquely identifies the EMS to which this Client Trail belongs.
trailName	String	Uniquely identifies the Client Trail within the EMS.
aEndInputPort	Object reference	A reference to the TransponderInOcnPort at the aEnd of the Client Trail.
zEndOutputPort	Object reference	A reference to the TransponderOutOcnPort at the zEnd of the Client Trail.
zEndInputPort	Object reference	A reference to the TransponderInOcnPort at the zEnd of the Client Trail.
aEndOutputPort	Object reference	A reference to the TransponderOutOcnPort at the aEnd of the Client Trail.

Optional Parameters

className	String	Class name of the Client Trail object to be created. Currently Client Trail is the only class name to be used for this API.

Return Values

Client Trail	Reference to the Client Trail object if the object is successfully created.
null	If the API fails to create the Client Trail object.

Dependency

Following objects should be created before creating Client Trail object:

- EMS
- TransponderInOcnPort at A and Z ends
- TransponderOutOcnPort at A and Z ends

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
RiddenBy	ClientCircuit	RidesOver	N:1	O	Y*
AendInPort	TransponderInOcnPort	OwnedBy	1:1	M	Y
ZendOutPort	TransponderOutOcnPort	OwnedBy	1:1	M	Y
ZendInPort	TransponderInOcnPort	OwnedBy	1:1	M	Y
AendOutPort	TransponderOutOcnPort	OwnedBy	1:1	M	Y
Owns	TransponderIn(Out)OcnPort	OwnedBy	1:N	M	Y

Owns/OwnedBy: This relationship is created within the API. Owns is created by the ClientTrail and OwnedBy is created by the TransponderInOcnPort and TransponderOutOcnPort objects. The values of the relationship contain references to the Port and Trail objects.

findClientTrail

This API is used to find the Client Trail object with the given trailName.

findClientTrail (trailName)

Mandatory Parameters

trailName	String	Uniquely identifies the Client Trail.

Return Values

Client Trail	Reference to the Client Trail object if the object is found.
null	If the API fails to find the Client Trail object.

Dependency

None

Relationships

See the Relationships section of "makeClientTrail" on page 79.

makeClientCircuit

This API is used to create the Client Circuit representing non-SONET service transported directly over the WDM network.

makeClientCircuit (emsName, circuitName, underlyingTrails, className = "*value*")

Mandatory Parameters

emsName	String	Uniquely identifies the EMS to which the Client Circuit belongs.
circuitName	String	Uniquely identifies the Client Circuit.
underlyingTrails	List	A list of references to the underlying Trails that transport the ClientCircuit.

Optional Parameters

className	String	Class name of the Client Circuit object to be created. Currently ClientCircuit is the only class name that can be used with this API.

Return Values

ClientCircuit	Reference to the Client Circuit object if the object is successfully created.
null	If the API fails to create the Client Circuit object.

Dependency

Following objects should be created before creating the Client Circuit object.

- EMS
- Client Trails

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory or optional	API Relation
RidesOver	ClientTail	RiddenBy	1:N	M	Y*
ManagedBy	EMS	ManagesLink	N:1	M	Y

findClientCircuit

findClientCircuit (circuitName)

This API is used to find the Client Circuit object with the given circuitName.

Mandatory Parameters

circuitName	String	Uniquely identifies the Client Circuit.

Return Values

Client Circuit	Reference to the Client Circuit object if the object is found.
null	If the API fails to find the Client Circuit object.

Dependency

None

Relationships

See the Relationships section of "makeClientCircuit" on page 82.

CHAPTER 4
Low Order SONET/SDH Object Factory API

These APIs are intended for use within the Low Order SONET/SDH Topology server. They facilitate topology creation and modification for physical and logical network components in the OTM Low Order SONET/SDH Domain Manager. The API set is categorized into two parts: Physical components and Logical components.

This chapter contains the following information:

- Low Order SONET/SDH components and API summary 86
- Low Order SONET/SDH physical components 87
- Low Order SONET/SDH Logical Components 98

Low Order SONET/SDH components and API summary

Table 3 on page 86 lists the Low Order SONET/SDH OTM components along with the APIs used to make and find that component. Also indicated is whether the component is physical hardware or a logical construction.

Note: API names for creating components start with "make" and API names for finding components already created and existing within the OTM repository start with "find." Some API names use the convention "makeN*xxx*" or "findN*xxx.*" The "N" emphasizes the fact that the native name (also known as the native AID) is used in forming the final component name without undergoing any change or normalization.

Table 3 Low Order SONET/SDH components and API summary (page 1 of 2)

Component Name/Class Instantiated	API Name	Entity Type	Details
EMS	makeEMS findEms	Physical	page 88
OpticalNetworkElement	makeNOpticalNetworkElement findOpticalNetworkElement	Physical	page 89
Equipment/Card	makeNEquipment, findNEquipment makeEquipment, findEquipment makeNCard, findNCard **Note:** makeCard and findCard deprecated. Use makeNCard and findNCard instead.	Physical	page 90
PTP	makeNPTP, findNPTP **Note:** makePTP and findPTP are deprecated. Use makeNPTP and findNPTP instead.	Physical	page 93
Rack	makeRack findRack	Physical	page 94
Shelf	makeShelf, findShelf makeNShelf, findNShelf	Physical	page 95
TopologicalLink	makeTopologicalLink findTopologicalLink	Physical	page 97
DropSideTopologicalLink	makeDropSideTopologicalLink findDropSideTopologicalLink	Physical	page 98
CTP	makeNCTP findNCTP	Logical	page 99
LowOrder CTP	makeLOCTP, findLOCTP	Logical	
LowOrder_Circuit	makeLOCircuit findLOCircuit	Logical	page 103

Table 3 Low Order SONET/SDH components and API summary (page 2 of 2)

HighOrder_Trail	makeHOTrail findHOTrail	Logical	page 105
EquipmentProtectionGroup	makeEqPGroup	Logical	page 106
LowOrder_Trail	makeLOTrail findLOTrail	Logical	page 105

Data layout

For each component, there is a section describing the following: the component name, API names used to make and find the component, object type returned by the APIs, parameters to the API, dependency of the API on other objects and finally the relationships of the component with all other components.

Under each component name, the following information is provided:

The object name to be created or found is a heading in the left column.

API Usage

Under the heading are the commands showing the usage syntax including the order of parameter.

Mandatory Parameters

These parameters are required. Parameters are described in three columns:

Parameter name, Data type, Description

Optional Parameters

These parameters are not required. Parameters are described in three columns:

Parameter name, Data type, Description

Return Values

Lists the object types returned by the APIs. Return values are listed in two columns:

Type of object reference returned, Description

Dependency

List of dependencies of the API on other objects.

Relationships

Lists of the relationships of the object with all other objects.

Low Order SONET/SDH physical components

This section describes the APIs that enable creation and or alteration of managed objects that form the physical components of the underlying network.

displayName: This is not used to uniquely identify an object in the OTM repository. It is used only for display purposes. The actual component name is used to uniquely identify the object and is also used to do a lookup to find whether the object exists or not.

Optional parameter: When specifying one or more optional parameters, you must specify preceding optional parameters with respective correct data types.

Enum type parameter: All enum type parameters must be passed as Strings (of valid enum value) for Java and ASL implementation. For C and C++ implementation, the exact enum type value should be passed.

EMS

makeEMS (emsName, displayName = "*value*", className = "*value*")

findEMS (emsName)

Mandatory Parameters

emsName	String	Name of the EMS to which Low Order SONET/SDH Domain Manager will be connecting. Must be a unique name.

Optional Parameters

displayName	String	User-friendly name of the EMS object.
className	String	Class name of the object to be created or found. Currently "EMS" is the only Class name for this API.

Return Values

EMS	Reference to the EMS object if the EMS object with the given emsName is successfully created or found.
null	If makeEMS is invoked with NULL emsName, or if EMS is not found.

Dependency

None

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	Mandatory (M) Optional (O)	API Relation
Manages	OpticalNetworkElement	ManagedBy	1:N	M	Y
ManagesTL	TopologicalLink	ManagedBy	1:N	M	Y

OpticalNetworkElement

makeNOpticalNetworkElement (tid, emsName, displayName = "*value*", className = "*value*")

findOpticalNetworkElement (tid)

Mandatory Parameters

tid	String	Target Identifier - Uniquely identifies the Network Element in a given network.
emsName	String	Name of the EMS to which OpticalNetworkElement belongs.

Optional Parameters

displayName	String	User-friendly name of the object displayed in the console.
className	String	Class name of the object to be created or found. Currently OpticalNetworkElement is the only Class name for this API.

Return Values

OpticalNetworkElement	Reference to the OpticalNetworkElement object if the object is successfully created or found.
null	If the API fails to create or find the OpticalNetworkElement object.

Dependency

EMS object must exist for the OpticalNetworkElement object to be created.

Relationships

Relationship	Related Classes	Reverse Relationship	Cardinality	Mandatory/ Optional (M/O)	API Relation
ManagedBy	EMS	Manages	1:1	M	Y
ComposedOfOpticalComponent	Equipment, Card, PTP	PartOf	1:N	M	Y
ComposedOfPTPs	PTP	None	1:N	M	Y
ComposedOfEquipment	Equipment, Card	PartOf	1:N	M	Y
ComposedOfCTPs	CTP	None	1:N	M	Y
SystemPackagedIn	Rack	PackagesSystems	1:1	O	Y

Equipment

makeEquipment (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", cardType, displayName = "*value*", className = "*value*")

makeNCard (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", cardType, displayName = "*value*", className = "*value*")

Note: makeCard is deprecated. Use makeNCard instead.

---

makeNEquipment (neName, shelfAid = "*value*", cardAid, cardType, displayName = "*value*", className = "*value*")

findEquipment (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*")

findNCard (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*")

Note: findCard is deprecated. Use findNCard instead.

---

findNEquipment (neName, cardAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Equipment or Card resides.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
cardAid	String	Access Identifier for the Equipment or Card within the network element.
cardType	String	Type of card. Used to uniquely identify between multiple cards from the same vendor. This plays a role while constructing EquipmentProtectionGroup and CrossConnect Equipments or the Card equivalents.

Optional Parameters

shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element. **Note:** For findEquipment, must be “0”.
displayName	String	User-friendly name to label the Equipment in the console.
className	String	Class name of the object to be created or found. Currently Equipment is the only Class name for this API.

Return Values

Equipment	Reference to the Equipment object if Equipment is successfully created or found.
null	If the API is not able to create or find Equipment.

Dependency

- Object of type OpticalNetworkElement with the name "neName".
- Object of type Shelf to be created if the "PartOfShelf" relationship is to be created later.
- For an Equipment/Card containing another Equipment/Card, (mother card - daughter card) PartOfEquipment relationship needs to be explicitly created after creating both the Equipments/Cards.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
Part Of	OpticalNetworkElement	ComposedOfOpticalComponent	1:1	M	Y
PartOfShelf	Shelf	ComposedOfEquipment	1:1	O	N
PartOfEquipment	Equipment	ComposedOf	1:1	O	N
Realizes	PTP	RealizedBy	1:N	M	Y
PartOfEquipmentProtectionGroup	EquipmentProtectionGroup	ComposedOf	1:1	O	Y
CrossConnects	PTP	CrossConnectedByEqpt	1:N	O	N

PTP (Physical Termination Point)

makeNLOPTP (neName, cardAid, ptpAid, displayName = "*value*", className = "*value*", relateEquipment = "*value*")

findNLOPTP (neName, ptpAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the PTP resides.
cardAid	String	Access Identifier for the Equipment/Card within the network element to which the PTP belongs. Uniquely identifies the Equipment/Card object within OTM repository.
ptpAid	String	Access Identifier for the PTP within the network element. Uniquely identifies the PTP within the OTM repository

Optional Parameters

displayName	String	User-friendly name to represent the PTP in the console.
className	String	Class name of the object to be created or found. Currently PTP is the only Class name for this API.
relateEquipment	Boolean	By default this is set to "TRUE." It is recommended not to change the default. This facilitates creating the relationship between the PTP and the Equipment/Card containing the PTP.

Return Values

PTP	Reference to the PTP object if created or found.
null	If the API is not able to create or find PTP.

Dependency

- Object of type OpticalNetworkElement with the name "neName"
- Object of type Equipment/Card with the name "cardAid"

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
RealizedBy	Equipment	Realizes	N:1	M	Y
Underlying	CTP	LayeredOver	1:N	M	Y
ConnectedVia	TopologicalLink	ConnectedTo	1:1	M	Y
ConnectedVia	DropSideTopologicalLink	ConnectedTo	1:1	M	Y
WestSendsTo	TopologicalLink	None	1:1	M	Y
WestSendsTo	DropSideTopologicalLink	None	1:1	O	Y
EastSendsTo	TopologicalLink	None	1:1	M	Y
EastSendsTo	DropSideTopologicalLink	None	1:1	O	Y
CrossConnectedBy	EquipmentProtectionGroup	CrossConnects	1:1	O	Y
CrossConnectedByEqpt	Equipment	CrossConnects	N:N	O	N

Rack

makeRack (neName, rackAid, displayName = "*value*", className = "*value*")

findRack (neName, rackAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Rack resides.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository.

Optional Parameters

displayName	String	User-friendly name to represent the Rack in the console.
className	String	Class name of the object to be created or found. Currently Rack is the only Class name for this API.

Return Values

Rack	Reference to the Rack object if successfully created or found.
null	If the API is not able to create or find the Rack.

Dependency

Object of type OpticalNetworkElement (with name/tid "neName") containing the Rack.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOf	Shelf	PartOf	1:N	O	Y
PackagesSystems	OpticalNetworkElement	SystemPackagedIn	1:1	O	Y

Shelf

makeShelf (neName, rackAid, shelfAid, displayName = "*value*", className = "*value*")

makeNShelf (neName, shelfAid, displayName = "*value*", className = "*value*")

findShelf (neName, rackAid, shelfAid)

findNShelf (neName, shelfAid)

Mandatory Parameters

neName	String	Name of the Network Element in which the Shelf resides.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within the OTM repository

Optional Parameters

displayName	String	User-friendly name to represent the Shelf in the console.
className	String	Class name of the object to be created or found. Currently Shelf is the only Class name for this API.

Return Values

Shelf	Reference to the Shelf object if successfully created or found.
null	If the API is not able to create or find the Shelf.

Dependency

- Object of type OpticalNetworkElement (with name/tid "neName") containing the Shelf.
- Object of type Rack in which the Shelf resides.
- makeShelf uses neName, rackAid and shelfAid to form the unique shelf name in the repository. findShelf uses neName, rackAid and shelfAid to form the unique shelf name to do a lookup.
- makeNShelf uses neName and shelfAid to form the unique shelf name in repository. findNShelf uses neName and shelfAid to form the unique shelf name to do a lookup.
- If Shelf is created using makeShelf, findShelf should be used to find it and not findNShelf.
- If Shelf is created using makeNShelf, findNShelf should be used to find it and not findShelf.
- If makeNShelf is used, the relationship between Rack and Shelf has to be explicitly created.
- If Shelf is created using makeShelf, shelfAid should not be used in makeNEquipment, a blank shelfAid should be used, since makeNEquipment uses findNShelf (with no rackAid) and creates the relationship between the Equipment and the Shelf.
- If Shelf is created using makeShelf and an Equipment in that Shelf is created using makeNEquipment, the relationship (PartOfShelf) between the Equipment and Shelf has to be explicitly created.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
Part Of	Rack	ComposedOf	1:1	O	Y
ComposedOfEquipment	Equipment	PartOfShelf	1:N	O	N

TopologicalLink

makeTopologicalLink (emsName = "*value*", topologicalLinkName, AendPTP, ZendPTP, displayName = "*value*", className = "*value*")

findTopologicalLink (emsName, topologicalLinkName)

Mandatory Parameters

topologicalLinkName	String	Name of the TopologicalLink used to uniquely identify it within the OTM repository.
AendPTP	Reference to PTP	PTP object at one end of the TopologicalLink.
ZendPTP	Reference to PTP	PTP object at the other end of the TopologicalLink.

Optional Parameters

emsName	String	Name of the EMS to which Low Order SONET/SDH Domain Manager will be connecting. It must be a unique name.
displayName	String	User-friendly name to represent the TopologicalLink in the console.
className	String	Class name of the object to be created or found. Currently TopologicalLink is the only Class name for this API.

Return Values

TopologicalLink	Reference to the TopologicalLink if successfully created or found.
null	If the API is not able to create or find TopologicalLink object.

Dependency

- Object of type EMS with the name "emsName" if emsName provided
- Object of type PTP - AendPTP created with its unique name/aid
- Object of type PTP - ZendPTP created with its unique name/aid

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedTo	PTP	ConnectedVia	1:N	M	Y
EastSendsTo	PTP	None	1:1	M	Y
WestSendsTo	PTP	None	1:1	M	Y
ManagedBy	EMS	ManagesTL	1:1	M	Y
PartOf	TopologicalLinkGroup	ComposedOf	1:1	O	Y

DropSideTopologicalLink

makeDropSideTopologicalLink (dropSidePTP, isWestPTP, displayName = "*value*", className = "*value*", emsName = "*value*")

findDropSideTopologicalLink (dropSidePTP)

Mandatory Parameters

dropSidePTP	Reference to PTP	PTP object at the drop side of the Circuit that forms the DropSideTopologicalLink.
isWestPTP	Boolean	Relative indication of whether the drop sidePTP is on the East or West end of the Circuit.

Optional Parameters

displayName	String	User-friendly name to represent the DropSideTopologicalLink in the console.
className	String	Class name of the object to be created or found. Currently DropSideTopologicalLink is the only Class name for this API.
emsName	String	Name of the EMS to which Low Order SONET/SDH Domain Manager will be connecting. It must be a unique name.

Return Values

DropSideTopologicalLink	Reference to the DropSideTopologicalLink if successfully created or found.
null	If the API is not able to create or find DropSideTopologicalLink object.

Dependency

- Object of type EMS with the name "emsName" if emsName provided
- Object of type PTP - dropSidePTP created with its unique name/aid

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedTo	PTP	ConnectedVia	1:1	M	Y
EastSendsTo	PTP	None	1:1	O	Y
WestSendsTo	PTP	None	1:1	O	Y
Underlying	HighOrder_Trail, LowOrder_Trail, LowOrder_Circuit	LayeredOver	1:N	M	N
PartOf	TopologicalLinkGroup	ComposedOf	1:1	O	Y

Low Order SONET/SDH Logical Components

This section describes the APIs that enable creation and or alteration of managed objects that form the logical components of the underlying network.

CTP (Connection Termination Point)

makeNLOCTP (neName, cardAid, ptpAid, ctpAid, ctpType, displayName = "*value*", className = "*value*", createContainment = "*value*")

findNLOCTP (neName, ctpAid)

Mandatory Parameters

neName	String	Name of the Network Element (neName/tid) containing the PTP to which the CTP belongs.
cardAid	String	Access Identifier for the Card within the network element to which the PTP belongs. Uniquely identifies the Card within the OTM repository.
ptpAid	String	Access Identifier for the PTP to which the CTP is "LayeredOver". Uniquely identifies the PTP within the OTM repository.
ctpAid	String	Access Identifier for the CTP within the network element. Uniquely identifies the CTP within the OTM repository.
ctpType	Enum	Enum of type Optical_GlobalDefs::ctp_type_e. Category of the CTP from the following types:

CTP_UNKNOWN	CTP_DS1
CTP_2MB	CTP_4MB
CTP_6MB	CTP_8MB
CTP_10MB	CTP_128K
CTP_256K	CTP_512K
CTP_1024MB	CTP_1984K

Optional Parameters

displayName	String	User-friendly name to represent the CTP.
className	String	Class name of the object to be created or found. Currently CTP is the only Class name for this API.
createContainment	Boolean	This is by default set to TRUE. It is recommended not to change the default. This facilitates the creation of the relationship between the CTP and underlying PTP.

Return Values

CTP	Reference to the CTP if successfully created or found.
null	If the API is not able to create or find CTP object.

Dependency

- Object of type OpticalNetworkElement (with name "neName") where the PTP containing this CTP resides
- Object of type Equipment (with name/aid cardAid) containing the PTP (with name/aid ptpAid)

- Object of type PTP (with name/aid “ptpAid”) containing this CTP

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
LayeredOver	PTP	Underlying	1:1	M	Y
RoutedBy	LowOrder_Circuit, LowOrder_Trail, HighOrder_Trail	CTPsInRoute	1:1	M	Y

Low Order CTP (Connection Termination Point)

makeLOCTP (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", portNumber, timeSlot, channel, ctpType, displayName = "*value*", className = "*value*", createContainment = "*value*")

findLOCTP (neName, rackAid, shelfAid = "*value*", slotNumber, subSlotNumber = "*value*", portNumber, timeSlot, channel)

Mandatory Parameters

neName	String	Name of the Network Element (neName/tid) containing the PTP to which the Low Order CTP belongs.
rackAid	String	Access Identifier for the Rack within the network element. Uniquely identifies the Rack within OTM repository.
slotNumber	String	Access Identifier of the slot within the network element.
portNumber	integer	Access Identifier of the port on the Equipment/Card in the specified slot or subslot.
timeSlot	integer	
channel	integer	
ctpType	Enum	Enum of type Optical_GlobalDefs::ctp_type_e. Category of the Low Order CTP from the following types:

CTP_UNKNOWN	CTP_VC4_8c
CTP_DS3	CTP_VC4_16
CTP_STS1	CTP_VC4_64c
CTP_STS3	CTP_ETS1
CTP_STS3C	CTP_ETS3
CTP_STS12	CTP_ETS6
CTP_STS12C	CTP_ETS9
CTP_STS24C	CTP_ETS12
CTP_STS48	CTP_ETS24
CTP_STS48C	CTP_E3
CTP_STS192C	CTP_34MB
CTP_AU3	CTP_45MB
CTP_AU4	CTP_50MB
CTP_VC4_4	CTP_100MB
CTP_VC4_4c	

Optional Parameters

displayName	String	User-friendly name to represent the Low Order CTP.
shelfAid	String	Access Identifier for the Shelf within the network element. Uniquely identifies the Shelf within the OTM repository.
subSlotNumber	String	Access Identifier of the subslot within the network element.
className	String	Class name of the object to be created or found. Currently LOCTP is the only Class name for this API.
createContainment	Boolean	This is by default set to TRUE. It is recommended not to change the default. This facilitates the creation of the relationship between the CTP and underlying PTP.

Return Values

LOCTP	Reference to the Low Order CTP if successfully created or found.
null	If API is not able to create or find Low Order CTP object.

Dependency

- Object of type OpticalNetworkElement (with name “neName”) where the PTP containing this CTP resides.
- Object of type Equipment (with name/aid cardAid) containing the PTP (with name/aid ptpAid).
- Object of type PTP (with name/aid “ptpAid”) containing this CTP

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ConnectedVia	SubnetworkConnection	ConnectedTo	1:1	O	Y
LayeredOver	PTP	Underlying	1:1	M	Y
RoutedBy	LowOrder_Circuit, LowOrder_Trail	CTPsInRoute	1:1	M	Y

LowOrder_Circuit

makeLOCircuit (circuitName, circuitType, AendCTP, ZendCTP, ctpsInRoute, className = "*value*")

makeLOCircuitWithoutEndCTP (circuitName, circuitType, ctpsInRoute, className = "*value*")

findLOCircuit (circuitName)

Mandatory Parameters

circuitName	String	Name of the Circuit used to uniquely identify it within OTM repository.
circuitType	String	Indicates the layer rate of the circuit (like 2MB, DS1, 10MB, etc.).
AendCTP	Reference to CTP	CTP object at one end of the LowOrder_Circuit.
ZendCTP	Reference to CTP	CTP object at the other end of the LowOrder_Circuit.
ctpsInRoute	List of objects of type CTP	CTPs through which LowOrder_Circuit is routed (excluding the CTPs at either end).

Optional Parameters

className	String	Class name of the object. Currently "LowOrder_Circuit" is the only Class name for this API.

Return Values

LowOrder_Circuit	Reference to the LowOrder_Circuit object if other object is successfully created or found.
null	If the API fails to create or find the LowOrder_Circuit object.

Dependency

- Object of type CTP - AendCTP created with its unique name/aid
- Object of type CTP - ZendCTP created with its unique name/aid
- Objects of type CTP - for all the CTPs in the route of the Circuit

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
aEnd	CTP	None	1:1	M	Y
zEnd	CTP	None	1:1	M	Y
CTPsInRoute	CTP	RoutedBy	1:N	M	Y
LayeredOver	HighOrder_Trail, LowOrder_Trail	Underlying	N:N	O	N
LayeredOver	DropSideTopologicalLink	Underlying	N:N	M	N
LayeredOver	TopologicalLinkGroup	Underlying	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N

HighOrder_Trail

makeHOTrail (trailName, trailType, className = "*value*")

findHOTrail (trailName)

Mandatory Parameters

trailName	String	Name of the HighOrder_Trail used to uniquely identify it within the OTM repository.
trailType	String	Indicates the layer rate of the trail (like STS3C, OC3 etc.).
ctpsInRoute	List of objects of type CTP	CTPs through which HighOrder_Trail is routed.

Optional Parameters

className	String	Class name of the object to be created or found. Currently HighOrder_Trail is the only Class name for this API.

Return Values

HighOrder_Trail	Reference to the HighOrder_Trail object if the object is successfully created or found.
null	If the API fails to create or find the HighOrder_Trail object.

Dependency

- Objects of type CTP - for all the CTPs in the route of the Trail
- Objects of type CTP for the two ends of the Circuit must be in the list of CTPs in route of the Trail.

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
CTPsInRoute	CTP	RoutedBy	1:N	M	Y
LayeredOver	TopologicalLink	Underlying	N:N	O	N
LayeredOver	TopologicalLinkGroup	Underlying	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N

EquipmentProtectionGroup

makeEqPGroup (neName, eqpts, ptps, displayName = "*value*", className = "*value*")

Mandatory Parameters

neName	String	Name/tid of OpticalNetworkElement in which the Equipment Protection is configured.
Eqpts	List of objects of type Equipment	Equipment or Card objects forming the Equipment Protection Group.
Ptps	List of objects of type PTP	List of PTP objects that are protected by the Equipment/Card objects.

Optional Parameters

displayName	String	User-friendly name to represent the EquipmentProtectionGroup in the console.
className	String	Class name of the object to be created or found. Currently EquipmentProtectionGroup is the only Class name for this API.

Return Values

EquipmentProtectionGroup	Reference to the EquipmentProtectionGroup object if the object is successfully created or found.
null	If the API fails to create or find the EquipmentProtectionGroup object.

Dependency

- Object of type OpticalNetworkElement where the EquipmentProtectionGroup is configured
- Objects of type Equipment forming the EquipmentProtectionGroup
- Objects of type PTP that are involved in the EquipmentProtectionGroup

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
ComposedOf	Equipment	PartOfEquipmentProtectionGroup	1:N	M	Y
CrossConnects	PTP	CrossConnectedBy	1:N	M	Y
UnderlyingLogical	LowOrder_Circuit, LowOrder_Trail, HighOrder_Trail	LayeredOverPhysical	N:N	O	N

LowOrder_Trail

makeLOTrail (trailName, trailType, ctpsInRoute, className = "*value*")

findLOTrail (trailName)

Mandatory Parameters

trailName	String	
trailType	String	
ctpsInRoute	List of objects of type CTP	List of CTP objects through which the LowOrder_Trail is routed.

Optional Parameters

className	String	Class name of the object to be created or found. Currently, LowOrder_Trail is the only Class name for this API.

Return Values

LowOrder_Trail	Reference to the LowOrder_Trail object if the object is successfully created or found.
null	If the API fails to create or find the LowOrder_Trail object.

Dependency

- All objects of type CTP forming the route of the LowOrder_Trail
- Objects of type CTP for the two ends of the Circuit must be in list of CTPs forming the route of the LowOrder_Trail

Relationships

Relationship	Related Class	Reverse Relationship	Cardinality	(M/O)	API Relation
CTPsInRoute	CTP	RoutedBy	1:N	M	Y
Underlying	LowOrder_Circuit	LayeredOver	1:N	M	N
LayeredOver	TopologicalLink	Underlying	N:N	O	N
LayeredOver	TopologicalLinkGroup	Underlying	N:N	O	N
LayeredOverPhysical	EquipmentProtectionGroup	UnderlyingLogical	N:N	O	N