Oracle® Communications Network Integrity

Multi-Domain Circuit Topology Stitching Cartridge Guide

Release 8.0
G34941-01
October 2025

Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Cartridge Guide, Release 8.0

G34941-01

Contents

2 Multi-Domain Circuit Topology Stitching Reconciliation Cartridge

About This Content

This guide explains the functionality and design of the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery and Multi-Domain Circuit Topology Stitching Reconciliation cartridges.

Audience

This guide is intended for Network Integrity administrators, developers, and integrators.

This guide assumes that you are familiar with the following documents:

- *Oracle Communications Network Integrity Developer's Guide*
- *Oracle Communications Network Integrity Installation Guide*
- *Oracle Communications Network Integrity CORBA Cartridge Guide*
- *Oracle Communications Network Integrity MIB-II SNMP Cartridge Guide*
- *Oracle Communications Network Integrity SNMP Discovery and UIM Integration Cartridge Guide*

This guide assumes that you are familiar with the following concepts:

- TMF814 standards and terminology
- Common Object Request Broker Architecture (CORBA) standards and terminology
- Oracle Communications Design Studio
- Oracle Communications Information Model
- SNMP MIBs

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Multi-Domain Circuit Topology Stitching Discovery Cartridge

Learn about the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge.

The multi-domain circuit topology stitching features are implemented using two cartridges. Network reconciliation is performed by the Multi-Domain Circuit Topology Stitching Reconciliation cartridge. For more information, see Multi-Domain Circuit Topology Stitching Reconciliation Cartridge.

Network discovery is performed by the Multi-Domain Circuit Topology Stitching Discovery cartridge. For more information, see the following topics:

Overview

The Multi-Domain Circuit Topology Stitching Discovery cartridge is used to discover and map networks by identifying devices in the network and the topological links between them. This cartridge allows users to perform network stitching within or across domains. It can be used to discover networks in the network domains:

- Synchronous digital hierarchy (SDH)
- Dense wavelength-division multiplexing (DWDM)
- Internet Protocol/Multiprotocol Label Switching (IP/MPLS)

This cartridge can discover physical devices and topological links in the networks of the specified domains. The discovered devices and links are modeled and reconciled to the inventory system.

This cartridge can also discover any Unmanaged Network Elements (UMEs) when performing circuit stitching across domains. UMEs are endpoint devices of a link present in two different domains. When this cartridge is used to perform circuit stitching for links with UMEs, the UME device information will be compared with the physical device discovery scan results stored in Network Integrity. Once the device is identified, its details are retrieved and updated. The modeled device information then appears in the results of the discovery scan.

When identifying networks in IP/MPLS domains, this cartridge uses the Oracle Communications Network Integrity IP Assimilation cartridge to discover topological links. The physical device and topological link details of the IP/MPLS devices are retrieved from the network management systems (NMSs) or element management systems (EMSs). The cartridge then uses these details to discover and create links between the devices; this is

performed by the **Assimilate IP Links** action in the IP Assimilation cartridge. The discovered IP links are then reconciled to the inventory to provide the complete mapping of the IP network.

For more information on using the IP Assimilation cartridge for performing circuit stitching in IP/MPLS networks, see the chapters "About the Cartridge Components" and "Using the Cartridge" in the *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*.

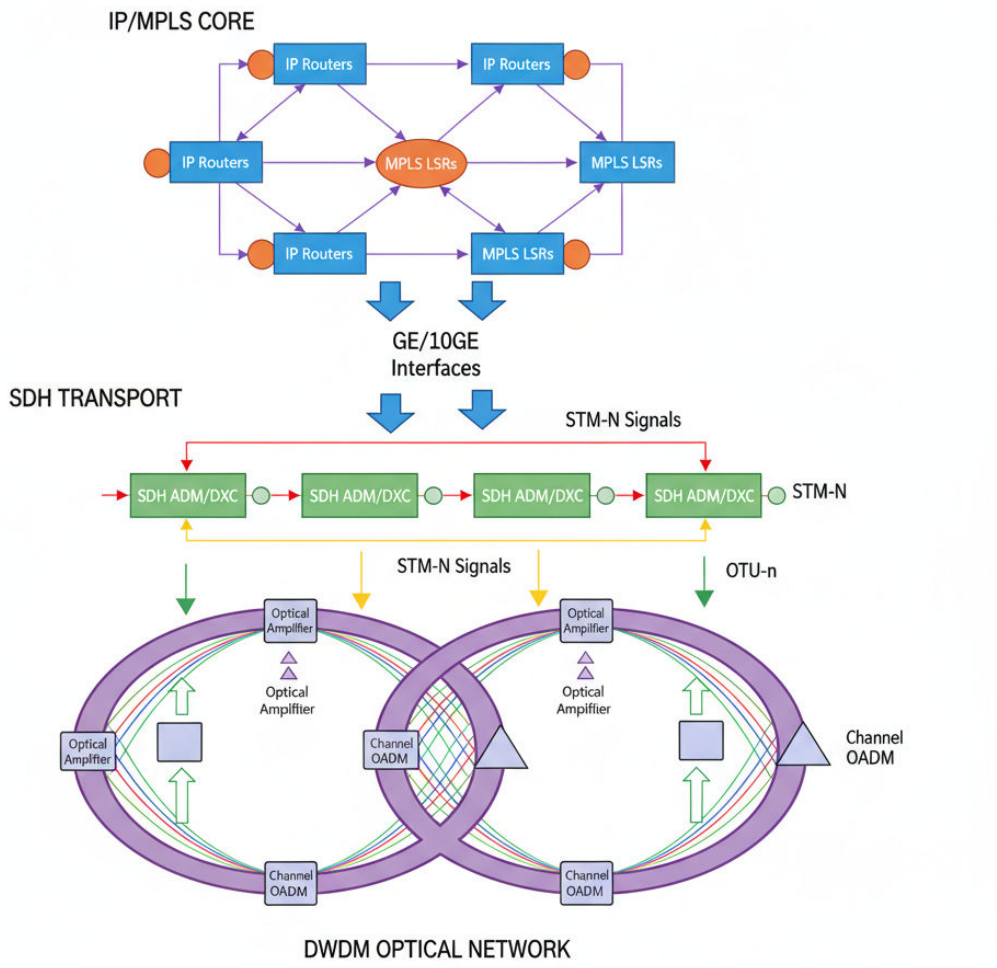
When identifying networks in the SDH and DWDM domains, the Multi-Domain Circuit Topology Stitching Discovery cartridge uses the TMF814 common object request broker architecture (CORBA) interface as a discovery protocol to connect and retrieve details from NMSs or EMSs. The cartridge provides discovery actions capable of discovering topological links.

Using this cartridge, you can configure Network Integrity to capture and retrieve data about the network system topological links for system vendors that have adopted the TMF814 standard.

It translates Multi Technology Network Management (MTNM) objects obtained during discovery into the Oracle Communications Information Model and then writes the objects to the Network Integrity database. To ensure scalability, this cartridge processes topological links individually. The duration of the discovery actions is proportional to the number and size of topological links to be discovered.

The circuit stitching process across domains is illustrated in Figure 1-1.

Figure 1-1 Circuit Stitching Across Domains



About Cartridge Dependencies

The Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge depends on the IP Assimilation cartridge to perform circuit stitching for devices in IP/MPLS domains.

For more information on the dependencies for the IP Assimilation cartridge, see *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*

The Multi-Domain Circuit Topology Stitching Discovery cartridge also has other dependencies as described further.

Run-Time Dependencies

There are no run-time dependencies for this cartridge.

Design Studio Dependencies

To load the Multi-Domain Circuit Topology Stitching Discovery cartridge (for SDH and DWDM domains) into Design Studio, the following cartridge must be installed:

- Network Integrity Cartridge for CORBA, including all of its dependencies

Opening the Cartridge Files in Design Studio

To review and extend the Multi-Domain Circuit Topology Stitching Discovery cartridge, you must download the **Network_Integrity-TMF814Discovery_Cartridge-8.0.0.0.0-generic.zip** file for Network Integrity, which contains the Design Studio cartridge files, from the Oracle software delivery website:

https://edelivery.oracle.com/

The **Network_Integrity-TMF814Discovery_Cartridge-8.0.0.0.0-generic.zip** cartridge file has the following structure:

- UIM_Cartridge_Projects\ora_ni_uim_ocim
- UIM_Cartridge_Projects\TMF814_Model
- UIM_Cartridge_Projects\ora_ni_uim_device_ports_interfaces_connectors
- UIM_Cartridge_Projects\ora_uim_network_device
- UIM_Cartridge_Projects\ora_ni_uim_sdh_optical
- UIM_Cartridge_Projects\ora_ni_uim_device_dwdm_optical
- UIM_Cartridge_Projects\ora_uim_basetechnologies
- UIM_Cartridge_Projects\ora_uim_basemeasurements
- Network_Integrity_Cartridge_Projects\TMF814Discovery_Cartridge
- Network_Integrity_Cartridge_Projects\Abstract_CORBA_Cartridge

The **TMF814Discovery_Cartridge** project contains extendable Design Studio files. You must open the files in Design Studio before you can review and extend the cartridge.

For more information on the guidelines and best practices for extending cartridges, see *Network Integrity Concepts*. For more information about opening files in Design Studio, see *Network Integrity Developer's Guide* and *Design Studio Modeling Network Integrity*.

For more information on the cartridge files for the IP Assimilation cartridge, see "Opening the Cartridge Files in Design Studio" in *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*

Building and Deploying the Cartridge

See *Design Studio Modeling Network Integrity* for information about building and deploying cartridges.

About the Cartridge Components

This section provides information about the components that make up the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge.

This cartridge contains the following actions:

- Abstract Discover Links Circuit Stitching
- Discover TMF814 Links Circuit Stitching

For more information about how actions are built, see About Design Studio Construction.

For more information about the actions used to perform circuit stitching in IP/MPLS domains, see *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*.

Abstract Discover Links Circuit Stitching

This is an abstract action that can be extended in Design Studio to discover specified network objects.

This action contains scan parameter groups; it can be extended to add new scan parameters, but the original scan parameters must remain.

The Abstract Discover Links Circuit Stitching processors are run in the following order:

1. ProcessStitchingDiscoveryScanInput
2. PrepareConnectionInputParams
3. ConnectCorbaCollectorService
4. CollectTopologicalLinks
5. ModelTopologicalLinks
6. DisconnectCollectorService

ProcessStitchingDiscoveryScanInput

This processor initializes the values for the CircuitStitchingScanParams parameters by gathering inputs and storing them in the ScanInputParams class. Table 1-1 contains the parameters for this processor.

Table 1-1 Description of the CircuitStitchingScanParams Parameters

Parameter	Description	TMF814 Transmission Scan Parameter
circleName	NMS Circle identifier	Nms Circle Name Nms Notification Circle
vendorName	NMS Vendor identifier	Nms Vendor Name Nms Notification Vendor
isFullNetworkScan	Set to true to perform a full network scan	Full Network Scan

PrepareConnectionInputParams

This processor validates the EmsConnection parameters and stores them into the class EmsConnectionConfigInput. Table 1-2 contains the parameters for this processor.

Table 1-2 Description of the EmsConnectionConfigInput Parameters

Parameter	Description	EmsConnection Parameter
emsType	Vendor-specific type used to connect with EMS.	Ems Type
emsInstanceName	Instance name used to connect with EMS.	Ems Instance Name
emsUserName	User name used to connect with EMS.	Ems UserName

Table 1-2 (Cont.) Description of the EmsConnectionConfigInput Parameters

Parameter	Description	EmsConnection Parameter
emsPassword	Password of the user name used to connect with EMS.	Ems Password
emsClass	EMS class used for naming service.	Ems Class
emsVersion	EMS version used for naming service.	Ems Version
emsSessionFactory	EMS session factory identifier used for naming service.	Ems Session Factory
ior/corbla loc	Unique identifier of port and IP used to connect to EMS.	scope address

This processor also initializes the parameters of the Incremental Parameters Scan, which are described in Table 1-3.

Table 1-3 Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label Description
IncrementalScan	Check box	UIncremental Scan set this box to indicate whether to enable incremental scans

Table 1-3 (Cont.) Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label description
		can. By default, this box is checked in the UI.

Table 1-3 (Cont.) Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label description
nmsNotificationCircle	String	UNms Notification Circle set this field to provide NMS circle/OSS name.

Table 1-3 (Cont.) Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label description
nmsNotificationVendor	String	UNms Notification Circle set this field to provide NMS vendor name.

Table 1-3 (Cont.) Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label escription
nmsNotificationCount	int	UNms Notification Vendor set this field to provide how many NMS notifications are re

Table 1-3 (Cont.) Description of Incremental Parameters Scan

Characteristic Name	Parameter Type	DUI Label description
		to be fetched.

ConnectCorbaCollectorService

This processor accepts data within the CorbaConnectionConfigInput and ScanInputParams classes as input and establishes a connection with TMF814 Collector Service. It provides the CorbaCollectorService object, which can be used to invoke the TMF814 server calls to connect network elements.

CollectTopologicalLinks

This processor accepts data from the CommonDataHolderVO and CorbaCollectorService as input and retrieves topological links from the NMS. It stores the details of the processing and common data in the commonDataHolderVO class.

It also provides the list of the collected topological links along with the DiscoveryTaskResponse<*TopologicalLinkT*> class object as output.

ModelTopologicalLinks

This processor creates topological links for network objects discovered in the scan. It accepts data from the commonDataHolderVO, DiscoveryTaskResponse<*TopologicalLinkT*> and ScanInputParams objects as inputs.

For each topological link created, a separate thread is invoked to collect and process the topological link hierarchy and the thread count is controlled by the value of the NI work manager. The topological links are modeled and persisted based on the input parameters accepted by the processor.

DisconnectCollectorService

This processor closes the connection with the EMS using the TMF814 Collector Service.

Discover TMF814 Links Circuit Stitching

The Discover TMF814 Links Circuit Stitching action extends the *Abstract Discover Links Circuit Stitching* abstract action and is a complete and deployable action. It can be configured using scan parameters and provides control over what topological link can and cannot be discovered.

This action can be extended to add new scan parameters, but the original scan parameters must be retained. It can also be extended to discover additional types of connectivity objects.

The Discover TMF814 Links Circuit Stitching action inherits all the processors from the *Abstract Discover Links Circuit Stitching*, which run in the following order:

1. ProcessStitchingDiscoveryScanInput
2. PrepareConnectionInputParams
3. ConnectCorbaCollectorService
4. CollectTopologicalLinks
5. ModelTopologicalLinks
6. DisconnectCollectorService

Using the Cartridge

This section provides instructions for using the Multi-Domain Circuit Topology Stitching Discovery cartridge in Network Integrity to create and run scans for identifying and creating topological links.

For more information about the discovery and reconciliation of topological link device physical hierarchies, see *Network Integrity Optical TMF814 CORBA Cartridge Guide* and *Network Integrity Optical UIM Integration Cartridge Guide*.

For more information on using the IP Assimilation Cartridge Guide for performing circuit stitching for IP/MPLS networks, see *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*

> **Note**
>
> Property Locations and Network Entity Codes must be loaded into UIM before the reconciliation of physical layer data.

Creating a Discover TMF814 Links Circuit Stitching Scan

To create a Discover TMF814 Links Circuit Stitching scan:

1. Create a new scan.

   For more information, see *Network Integrity Online Help*.
2. On the **General** tab, do the following:
   a. From the Scan Action list, select **Discover TMF814 Links Circuit Stitching**.

      The Scan Type field displays **Discovery**.

b. Select **Detect Discrepancies**.

c. Select **Enabled** to enable discrepancy detection.

3. Enter values for the following EMS Connection scan parameters:

a. In the **Ems User Name** field, enter the username for the target EMS.

b. In the **Ems Password** field, enter the password for the target EMS.

c. In the **Ems Type** field, use ECI or HUAWEI2000 based on EMS vendor.

d. In the **Ems Instance Name** field, enter the name of the EMS instance connection.

e. In the **Ems Vendor** field, enter the name of the EMS vendor. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

f. In the **Ems Class** field, enter the name of the EMS class. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

g. In the **Ems Version** field, enter the EMS version value. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

h. In the **Ems Session Factory** field, enter the EMS session factory value. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

4. Do the following to add values for the circuit stitching scan parameters:

a. Select the **Full Network Scan** check box to perform a full network discovery.

b. In the **Nms Vendor Name** field, enter the NMS Vendor identifier value.

c. In the **Nms Circle Name** field, enter the NMS Circle identifier value.

5. Perform any other required configurations.

6. Save and run the scan.

Creating a Discover TMF814 Links Circuit Stitching Incremental Scan

To create a TMF814 Links Circuit Stitching incremental scan:

1. Create a new scan.

   For more information, see *Network Integrity Online Help*.

2. On the **General** tab, do the following:

a. From the Scan Action list, select **Discover TMF814 Links Circuit Stitching**.

   The Scan Type field displays *Discovery*.

b. Select **Detect Discrepancies**.

c. Select **Enabled** to enable discrepancy detection.

3. Enter values for the following EMS Connection scan parameters:

- In the **Ems User Name** field, enter the username for the target EMS.
- In the **Ems Password** field, enter the password for the target EMS.
- In the **Ems Type** field, use ECI or HUAWEI2000 based on the EMS vendor.
- In the **Ems Instance Name** field, enter the name of the EMS instance connection.
- In the **Ems Vendor** field, enter the name of the EMS vendor. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

- In the **Ems Class** field, enter the name of the EMS class. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.
- In the **Ems Version** field, enter the EMS version value. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.
- In the **Ems Session Factory** field, enter the EMS session factory value. This is not mandatory if the EMS uses a naming service to provide vendor names for name resolution.

4. Do the following to add values for the incremental scan parameters:

> **Note**
>
> Ensure that you run a successful incremental import scan before running an incremental discovery scan.

   a. Select the **Incremental Scan** check box to perform incremental discovery.
   b. In **Nms Notification Vendor** field, enter the NMS Vendor identifier value.
   c. In **Nms Notification Circle** field, enter the NMS Circle identifier value.
   d. In **Nms Notification Count** field, enter the count of notifications to be processed.

5. On the **Scope** tab, do any one of the following:
   - Enter the EMS CORBA Location URL.
   - Import the Interoperable Object Reference (IOR) file.
   - Enter the contents of the IOR file.

> **Note**
>
> All entries on the **Scope** tab must be unique. All entries are validated against the CorbaURLAddressHandler address handler.

6. Perform any other required configurations.
7. Save and run the scan.

Filtering Data Based on Vendor and Circle Parameters

You can filter data during discovery and import scans by defining the **Nms Notification Vendor** and **Nms Notification Circle** parameters. This ensures that only data matching these criteria will be available in the scan results.

When reconciling this data with UIM, the topological link will include both vendor and circle as characteristics.

To reconcile data based on vendor and circle parameters:

1. Create a Circuit Stitching import scan with the requisite **Nms Vendor** and **Nms Circle** parameter values and run the scan.
2. Create a Circuit Stitching discovery scan with the requisite **Nms Vendor** and **Nms Circle** parameter values and run the scan with discrepancy detection enabled.

Ensure that the **Nms Vendor** and **Nms Circle** parameter values entered in both import and discovery scans are the same.

3. Perform the reconciliation of discovered data.
4. Verify the reconciled data by using the same **Nms Vendor** and **Nms Circle** parameter values.
5. Run the import scan.
6. Run the discovery scan.

About Collected Data

This section explains how the Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge treats collected data.

The Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge uses a standard TMF814 CORBA interface, which models network elements using the MTNM standard.

Table 1-4 lists the MTNM objects and corresponding TMF814 IDL API class definitions.

Table 1-4 MTNM IDL Class Definitions

MTNM Object Name	TMF814 IDL API Class Definition
Topological Link	TopologicalLink_T

Multi Technology Network Management Hierarchy

Table 1-5 describes the properties of each MTNM object collected by the Multi-Domain Circuit Topology Stitching Discovery cartridge.

Table 1-5 Topological Link Properties

Property Name	Description
nativeEMSName	Indicates how the topological link is referred to on EMS displays.
direction	Indicates the direction of the topological link. A topological link can be unidirectional even if both its ends are bidirectional Termation Points. Possible values are CD_UNI (unidirectional) and CD_BI (bidirectional).
rate	Indicates the layer rate (bandwidth) of the topological link.
aEndTP	Indicates the name of the aEnd for the Physical Termination Point (PTP), Connection Termination Point (CTP), or Floating Termination Point (FTP).
name	Indicates the name of the Topological Link, assigned by the EMS upon creation.
zEndTP	Indicates the name of the zEnd for the PTP, CTP, or FTP.
userLabel	Indicates the topological link user label (end-to-end trail data) in the NMS data.
owner	Provided by the NMS.

Table 1-5 (Cont.) Topological Link Properties

Property Name	Description
additionalInfo	Represents a list of name/value pairs that allow EMSs or NMSs to give additional information that is not explicitly modeled at the MTNM interface, but some parameter names and values may be predefined. Some predefined parameter names may include: AlarmReporting, AllocatedNumber, ASAPpointer, FragmentServerLayer, NetworkAccessDomain.

TMF814 APIs

Table 1-6 lists the APIs used by the Multi-Domain Circuit Topology Stitching cartridge to collect data.

Table 1-6 CORBA APIs used by the Multi-Domain Circuit Topology Stitching Cartridge

API	Operations Used
org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_I	• getEmsSession(): Used to obtain the EmsSession objects.
org.tmforum.mtnm.emsSession.EmsSession_I	• getManager(): Used to obtain managers. • endSession(): Used to close the EMS session.
org.tmforum.mtnm.multiLayerSubnetwork.MultiLayerSubnetworkMgr_I	• public void getAllTopologicalLinks(NameAndStringValue_T[] names, int count, TopologicalLinkList_THolder holder, TopologicalLinkIterator_IHolder iterator): Obtain all topological link mapped to each subnetwork.
org.tmforum.mtnm.emsMgr.EMSMgr_I	• public void getAllTopLevelSubnetworks(int count, SubnetworkList_THolder holder, SubnetworkIterator_IHolder sncIterator): Collects all top level subnetworks.
org.tmforum.mtnm.nmsSession.NmsSession_I	• EmsSessionFactory_I.getEmsSession: Required nmsSesion while getting an Ems session.

About Cartridge Modeling

This chapter explains how the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge models collected data.

The Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge models collected data according to the Oracle Communications Information Model. Collected data is modeled into the following entities:

- Pipe
- PipeTerminationPoint
- PipePipeTerminationPointRel

See About the Oracle Communications Information Model for more information about the Information Model.

For more information on modeling device information for IP/MPLS network devices, see "About Cartridge Modeling" in *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*.

About the Oracle Communications Information Model

This section details how the Multi Technology Network Management (MTNM) model is mapped to the Information Model.

> **Note**
>
> Multi-Domain Circuit Stitching Discovery relies on the Optical TMF814 CORBA cartridge to model the collected topological link data.

Cartridge Modeling for Topological Link Data

This section explains how the Optical TMF814 CORBA cartridge models collected topological link data.

Topological links are modeled as Information Model pipe entities and Topological Link endpoints (aEndTP and zEndTP) are modeled as pipe termination point entities.

Some vendors represent bidirectional topological links as two unidirectional topological links (two links sharing the same aEnd and zEnd ports). Such links are merged and modeled as one bidirectional topological link.

Table 1-7 lists the specification mapping.

Table 1-7 Specification Mapping for the Pipe and PTP Object

MTNM Object	Information Model Entity	Specification
Topological Link	Pipe	topologicalLink
aEndTP, zEndTP (of a topological link object)	Pipe Termination Point	portTerminationPoint

Table 1-8 and Table 1-9 describe the information model mapping for MTNM objects.

Table 1-8 Model Mapping for Pipe Objects

Information Model Attribute	Information Model Support	Type
description	Static	Text
gapPipe	Static	Boolean, always set to True.
id	Static	Text The value is derived. Possible values are PRIMARY, BACKUP.
medium	Static	Text
name	Static	Text
transmissionSignalType	Static	Text
versioned	Static	Boolean
aDeviceIPAddress	Chars	Text
aEndEMS	Chars	Text
aEndPoint	Chars	Text
associatedTrail	Chars	Text

Table 1-8 (Cont.) Model Mapping for Pipe Objects

Information Model Attribute	Information Model Support	Type
circleName	Chars	Text
connectivityType	Chars	Text
consistentState	Chars	Text
customer	Chars	Text
directionType	Chars	Text
diverseRouteSearchEffort	Chars	Text
domain	Chars	Text
endPointA	Chars	Text
endPointACircle	Chars	Text
endPointANativeEmsName	Chars	Text
endPointZ	Chars	Text
endPointZCircle	Chars	Text
endPointZNativeEmsName	Chars	Text
hangingType	Chars	Text
inconsistencyReason	Chars	Text
layerRate	Chars	Text
linkCost	Chars	Text
linkCreationDate	Chars	Text
linkDistance	Chars	Text
linkName	Chars	Text
linkQuality	Chars	Text
linkType	Chars	Text
numberOfDiverseRoutes	Chars	Text
protection	Chars	Text
ringName	Chars	Text
sourceOfTL	Chars	Text
srlgDetails1	Chars	Text
srlgDetails2	Chars	Text
srlgDetails3	Chars	Text
status	Chars	Text
technology	Chars	Text
tlName	Chars	Text
trailId	Chars	Text
trailName	Chars	Text
VCATSize	Chars	Text
Vendor	Chars	Text
zDeviceIPAddress	Chars	Text
zEndEMS	Chars	Text
zEndPoint	Chars	Text

Table 1-9 Model Mapping for the PipeTerminationPoint Object

Information Model Attribute	Information Model Support	Type
name	Static	Text The name of the PTP (port) cross-connect endpoint.
id	Static	Text
description	Static	Text
device	Dynamic	Text
directionality	Dynamic	Text
rate	Dynamic	Text This value is derived from the line layer rate for the endPort represented by the PortTerminationPoint.
channel	Dynamic	Text This attribute is not used.

About Design Studio Construction

This section explains how the Network Integrity Multi-Domain Circuit Topology Stitching Discovery cartridge is built from the Design Studio perspective.

For more information about the IP Assimilation cartridge model collection, see "About Design Studio Construction" in *Network Integrity IP Network Links Assimilation and Reconciliation Cartridge Guide*.

Model Collections

Table 1-10 shows the Design Studio construction of the Multi-Domain Circuit Topology Stitching cartridge collection.

Table 1-10 Multi-Domain Circuit Topology Stitching Discovery Cartridge Model Collection

Specification Name	Dynamic Entity Type
topologicalLink	Pipe Specification
portTerminationPoint	Pipe Termination Point Specification

Actions

Table 1-11 outlines the Design Studio construction of the Multi-Domain Circuit Topology Stitching cartridges actions and associated components.

> **Note**
>
> Parameter values are case-sensitive and must be entered in capital letters when commands are run from a command line interface.

Table 1-11 Actions for Design Studio Construction

Action Name	Result Category	Address Handler	Scan Parameter Group	Processors
Discover TMF814 Links Circuit Stitching	TopologicalLink	CorbaURLAddress Handler	EMS Connection Circuit Stitching Scan Params IncrementalScanParameter	ProcessStitchingDiscoveryScanInput PrepareConnectionInputParams ConnectCorbaCollectorService CollectTopologicalLinks ModelTopologicalLinks DisconnectCollectorService

Scan Parameters

Table 1-12 and Table 1-13 contain information about the scan paramters in Design Studio for the Multi-Domain Circuit Topology Stitching Discovery cartridge.

> **Note**
>
> Parameter values are case-sensitive and must be entered in capital letters when commands are run from a command line interface.

Table 1-12 Circuit Stitching Scan Parameters for Design Studio Construction

Parameter Name	Parameter Type	Description
Full Network Scan	Check box	Select this check box for full network discovery.
Nms Vendor Name	Text box	Name of NMS/EMS vendor, generally used to filter data while importing.
Nms Circle Name	Text box	Name of NMS/EMS circle, generally used to filter data while importing.

Table 1-13 Discovery Processors for Design Studio Construction

Discovery Processors	Variable
ProcessStitchingDiscoveryScanInput	Input: Not Applicable Output: • commonDataHolderVO(oracle.communications.integrity.tmf814discovery.circuitstitching.vo.CommonDataHolderVO)

Table 1-13 (Cont.) Discovery Processors for Design Studio Construction

Discovery Processors	Variable
PrepareConnectionInputParams	Input: Not Applicable Output: • corbaConnectionConfigInput(com.oracle.communications.network.integrity.tmfcollectorservice.connection.CorbaConnectionConfigInput)
ConnectCorbaCollectorService	Input: • corbaConnectionConfigInput(com.oracle.communications.network.integrity.tmfcollectorservice.connection.CorbaConnectionConfigInput) Output: • corbaCollectorService(oracle.communications.integrity.abstractcorbacartridge.common.collector.CorbaCollectorService)
CollectTopologicalLinks	Input: • commonDataHolderVO(oracle.communications.integrity.tmf814discovery.circuitstitching.vo.CommonDataHolderVO) • corbaConnectionConfigInput(com.oracle.communications.network.integrity.tmfcollectorservice.connection.CorbaConnectionConfigInput) Output: • discoveryTaskResponses(java.util.List<oracle.communications.integrity.abstractcorbacartridge.common.task.DiscoveryTaskResponse<TopologicalLinkT>>)
ModelTopologicalLinks	Input: • commonDataHolderVO(oracle.communications.integrity.tmf814discovery.circuitstitching.vo.CommonDataHolderVO) • corbaConnectionConfigInput(com.oracle.communications.network.integrity.tmfcollectorservice.connection.CorbaConnectionConfigInput) • discoveryTaskResponses(java.util.List<oracle.communications.integrity.abstractcorbacartridge.common.task.DiscoveryTaskResponse<TopologicalLinkT>>)
DisconnectCollectorService	Input: • corbaCollectorService(oracle.communications.integrity.abstractcorbacartridge.common.collector.CorbaCollectorService) Output: Not Applicable

2

Multi-Domain Circuit Topology Stitching Reconciliation Cartridge

Learn about the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Reconciliation cartridge.

Network Integrity's multi-domain circuit topology stitching features are implemented using two cartridges. Network discovery is performed by the Multi-Domain Circuit Topology Stitching Discovery cartridge. For more information, see Multi-Domain Circuit Topology Stitching Discovery Cartridge. Network reconciliation is performed by the Multi-Domain Circuit Topology Stitching Reconciliation cartridge. For more information, see the following topics:

Overview

The Multi-Domain Circuit Topology Stitching Reconciliation cartridge demonstrates the end-to-end integration of TMF814 discovery and assimilation with Oracle Communications Unified Inventory Management (UIM).

The reference implementation demonstrates full integration of NI with UIM for import, discrepancy detection, and discrepancy resolution with all discoverable entities.

For more information, see *Network Integrity Developer's Guide*.

About the Cartridge Dependencies

This section provides information about dependencies the Multi-Domain Circuit Topology Stitching Reconciliation cartridge has on other entities.

Run-Time Dependencies

For this cartridge to work at run time, ensure that UIM is installed and available and that the following cartridges are deployed in Network Integrity:

- Address_Handlers
- TMF814Discovery_Cartridge
- Optical UIM Integration Cartridge
- UIM Integration Cartridge

The following components must be deployed to UIM:

- UIM Integration web service

- ora_uim_network_device
- ora_ni_uim_sdh_optical
- ora_ni_uim_ocim
- OracleComms_UIM_Device_PortsInterfacesConnector
- ora_ni_uim_device_ports_interfaces_connectors
- ora_ni_uim_device_dwdm_optical

Design-Time Dependencies

This cartridge has the following dependencies:

- Abstract_CORBA_Cartridge
- Address_Handlers
- NetworkIntegritySDK
- OpticalAssimilation_Cartridge
- Optical_Model
- OpticalAssimilation_Model
- ora_ni_uim_device_dwdm_optical
- ora_ni_uim_device_ports_interfaces_connectors
- ora_ni_uim_ocim
- ora_ni_uim_sdh_optical
- ora_uim_model
- ora_uim_network_device
- TMF814_Model
- TMF814Discovery_Cartridge
- UIM_Integration_Cartridge
- OracleComms_UIM_Device_PortsInterfacesConnectors

Opening the Cartridge Files in Design Studio

To review and extend the Multi-Domain Circuit Topology Stitching Reconciliation cartridge, download the Oracle Communications Network Integrity Optical UIM Integration cartridge software from the Oracle software delivery website:

https://edelivery.oracle.com/

The software contains the Optical UIM Integration cartridge ZIP file, which has the following structure:

- \UIM_Cartridge_Projects\ora_ni_uim_ocim
- \UIM_Cartridge_Projects\OracleComms_UIM_Device_PortsInterfacesConnectors
- \UIM_Cartridge_Projects\ora_ni_uim_device_ports_interfaces_connectors
- \UIM_Cartridge_Projects\ora_uim_network_device
- \UIM_Cartridge_Projects\ora_ni_uim_sdh_optical
- \UIM_Cartridge_Projects\ora_ni_uim_device_dwdm_optical

- \UIM_Cartridge_Projects\SDH_Service_Model
- \UIM_Cartridge_Projects\ora_ni_uim_webservice
- \Network_Integrity_Cartridge_Projects\SDH_UIM_Cartridge
- \Network_Integrity_Cartridge_Projects\Generic_SNMP_Model
- \Network_Integrity_Cartridge_Projects\MIB_II_UIM_Cartridge
- \Network_Integrity_Cartridge_Projects\Optical_Model
- \Network_Integrity_Cartridge_Projects\Optical_UIM_Cartridge
- \Network_Integrity_Cartridge_Projects\UIM_Integration_Cartridge

The Network_Integrity_Cartridge_Projects\Optical_UIM_Cartridge\ project contains the extendable files for Oracle Communications Service Catalog and Design - Design Studio.

See *Network Integrity Concepts* for guidelines and best practices for extending cartridges.

See *Design Studio Modeling Network Integrity* for more information about opening projects in Design Studio.

Building and Deploying the Cartridge

See *Design Studio Modeling Network Integrity* for information about building and deploying cartridges.

About the Cartridge Components

This section provides information about the components of the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Reconciliation cartridge.

This cartridge contains the following actions:

- Import TMF814 Links Circuit Stitching from UIM
- Detect Discrepancy of Links Circuit Stitching
- Resolve TMF814 Topological Links Circuit Stitching in UIM

Import TMF814 Links Circuit Stitching from UIM

The Import TMF814 Links Circuit Stitching action imports the data reconciled into UIM.

The processors of the Import TMF814 Links Circuit Stitching action run in the following order:

- ProcessScanFilterUIM: This processor implements the discrepancy detection filter.
- Collect Links: Retrieves all the topological links, applying filters if defined.
- Model Links: Iterates over each topological link ID to:
  - Retrieve and model the topological link.
  - Retrieve and model the pipe termination point.
  - Persist the topological link.

Detect Discrepancy of Links Circuit Stitching

The Detect Discrepancy of Links Circuit Stitching action detects discrepancies between the discovery scan results obtained from the Discover TMF814 Links Circuit Stitching action and the data imported from UIM.

This action inherits the Detect Discrepancies processor (from NetworkIntegritySDK). The processors of the Detect TMF814 Optical Transmission Devices Discrepancies action run in the following order:

- Circuit Stitching TL DD Filters (this processor implements discrepancy detection filter)
- Discrepancy Detector (inherited)

Resolve TMF814 Topological Links Circuit Stitching in UIM

The Resolve TMF814 Topological Links Circuit Stitching in UIM action extends the Abstract Resolve in UIM action and inherits all its processors. For more information about the Abstract Resolve in UIM action, see *Network Integrity UIM Integration Cartridge Guide*.

The Resolve TMF814 Topological Links Circuit Stitching in UIM action contains the following processors run in the following order:

- UIM Resolution Framework Initializer (inherited)
- UIM Resolution Initializer (inherited)
- TMF814 Circuit Stitching Resolution Initializer (this processor defines the Pipe handler to initialize the creation and deletion and update operation over topological link)
- UIM Resolution Framework Dispatcher (inherited)

Rehome Support for Circuit Stitching

When you rehome a connectivity, you alter one of its endpoints. This action may be necessary for load balancing purposes or due to the removal or replacement of devices and interfaces. For more information, see UIM Concepts. The groom feature can also be used for:

- Moving customer connections to different network nodes to improve service quality.
- Physically relocating network devices to different location.
- Switching network services from one provider to another.
- Shifting network resources, such as IP addresses or bandwidth, to different parts of the network.

Rehoming a facility requires changes to the termination of the facility itself and to any channels it provides. Channels are re-terminated on sub-device interfaces provided by the new device interface on which the facility is terminated.

UIM exposes a Rehome API over REST protocol. Network Integrity detects such port change discrepancies for each connectivity facility and generates a Rehome request and invokes Rehome endpoint of UIM. Upon receiving the rehome request, UIM immediately acknowledges with a 202 response and processes it in the background. Network Integrity then regularly checks UIM for a Rehome response and addresses the discrepancies accordingly.

For more information on how to invoke Rehome REST APIs, see *REST API for Unified Inventory Management*.

A default API is introduced within the DiscrepancyHandler interface of the UIM_Integration_Cartridge, which takes a collection of discrepancies (such as missing or extra pipe or PTP entities) as input, as shown below. The design path can be modified by making device changes or termination point changes on one side.

The input to the API is passed from the oracle.communications.integrity.uim.resolutionprocessors.base.BaseResolutionElement class within the UIM_Integration_Cartridge. This class contains the capability to group discrepancies respective to the missing or extra entities for each result group.

The below out of the box reconciliation handler has the logic to rehome DWDM layers.

Action → Resolve TMF814 Topological Links Circuit Stitching in UIM

Processor → TMF814 Circuit Stitching Resolution Initializer

Class → oracle.communications.integrity.uim.resolutionprocessors.uimresolutioninitializer.entityHandlers.circuitstitching.CSPipeTerminationPointHandler .

Method → public void handleDiscrepancyRehome(Discrepancy missingEntity, Discrepancy extraEntity)

> **Note**
>
> For rehome scenarios, Entity- and Entity+ discrepancies need to be submitted together for processing.

Using the Cartridge

To use the Oracle Communications Network Integrity Multi-Domain Circuit Topology Stitching Reconciliation cartridge, see the following topics:

- Creating an Import TMF814 Links Circuit Stitching from UIM Scan
- Creating an Import TMF814 Links Circuit Stitching from UIM Incremental Scan
- Detect Discrepancy of Links Circuit Stitching
- Resolve TMF814 Topological Links Circuit Stitching in UIM

For the discovery and reconciliation of Topological link device physical hierarchies, see *Network Integrity Optical TMF814 CORBA Cartridge Guide* and *Network Integrity Optical UIM Integration Cartridge Guide*.

> **Note**
>
> Property Locations and Network Entity Codes must be loaded into UIM before reconciling physical layer data.

Creating an Import TMF814 Links Circuit Stitching from UIM Scan

To create an Import TMF814 Links Circuit Stitching scan:

1. Create a new scan.
   For more information, see *Network Integrity Online Help*.

2. On the **General** tab, do the following:
   a. From the Scan Action list, select **Import TMF814 Links Circuit Stitching**. The Scan Type field displays **Import**.
   b. Select **Enabled**.
3. Enter the following circuit stitching scan parameters:
   a. In the **Nms Vendor Name** field, enter the name of the NMS vendor.
   b. In the **Nms Circle Name** field, enter the identifier of the NMS circle.
4. Perform any other required configurations.
5. Save and run the scan.

Creating an Import TMF814 Links Circuit Stitching from UIM Incremental Scan

To create an Import TMF814 Links Circuit Stitching Incremental Scan:

> **Note**
>
> Before creating a scan, check if the NMS Listener is configured for the EMS system and is able to populate notifications in Network Integrity notification related tables.

1. Create a new scan.
   For more information, see *Network Integrity Online Help*
2. On the **General** tab, do the following:
   a. From the Scan Action list, select **Import TMF814 Links Circuit Stitching from UIM**. The Scan Type field displays **Import**.
   b. Select **Enabled**.
3. Enter the following incremental scan parameters: (Only required for incremental discovery)
   - Select the **Incremental Scan** checkbox to perform incremental discovery.
   - In the **Nms Notification Vendor** field, enter the NMS vendor name.
   - In the **Nms Notification Circle** field, enter the identifier of the NMS circle.
   - In the **Nms Notification Count** field, enter the count of notifications to be processed.
4. Perform any other required configurations.
5. Save and run the scan.

Filtering DWDM Data

You can filter data during discovery and import scans by defining the **Nms Notification Vendor** and **Nms Notification Circle** parameters. This ensures that only data matching these criteria will be available in the scan results. When reconciling this data with UIM, the topological link will include both vendor and circle as characteristics.

To reconcile data based on vendor and circle parameters:

1. Create a Circuit Stitching import scan with the requisite **Nms Notification Vendor** and **Nms Notification Circle** parameter values and run the scan.

2. Create a Circuit Stitching discovery scan with the requisite **Nms Notification Vendor** and **Nms Notification Circle** parameter values and run the scan with discrepancy detection enabled.

> **Note**
>
> Ensure that the Nms Vendor and Nms Circle parameter values entered in both import and discovery scans are the same.

3. Perform reconciliation of discovered data.
4. Verify the reconciled data by using the same **Nms Notification Vendor** and **Nms Notification Circle** parameter values.
5. Run the Import Scan.
6. Run the Discovery Scan.

Detect Discrepancy of Links Circuit Stitching

The Multi-Domain Circuit Topology Stitching Reconciliation cartridge allows you to detect and resolve discrepancies between your discovered data and your imported UIM data circuit discrepancies. When you resolve a discrepancy, the resolution is submitted to UIM by Network Integrity. For more information about working with discrepancies, see *Network Integrity Online Help*.

Discrepancies on optical entities must be resolved in a specific order:

1. Begin by detecting and resolving discrepancies on link entities.
2. When the Optical UIM Integration cartridge is deployed to your run-time application, you can use Network Integrity for:
   - Detect Discrepancy of Links Circuit Stitching

The following procedure describes the steps to populate UIM with the network data discovered by the Discover TMF814 Links Circuit Stitching discovery action.

To populate UIM with the discovered network data:

1. Create a new scan.

   For more information, see *Network Integrity Online Help*.
2. On the **General** tab, do the following:
   a. From the Scan Action list, select **Discover TMF814 Links Circuit Stitching**.

      The Scan Type field displays **Discovery**.
   b. Select the **Detect Discrepancies** checkbox.
   c. In the **Scan Action Parameters** area, perform any necessary additional configurations.
3. Save and run the discovery scan.

   The scan generates **Entity+** discrepancies for each discovered device.

Resolve TMF814 Topological Links Circuit Stitching in UIM

For reconciliation of discovered devices, the network location and network entity codes for the devices must be present in UIM beforehand.

All devices are linked with the network location and the network entity code that can be verified from discovery result. If both the characteristics are not present in UIM, the devices discovered will not be reconciled.

For more information on creating bulk network location and network entity codes in UIM, see *UIM System Administrator's Guide*.

1. Review the scan results for a scan that was run with **Detect Discrepancies** enabled.
2. On the Scan Details page, click **Review Discrepancies**.
3. Right-click on the discrepancy to be resolved and select **Resolve TMF814 Topological Links Circuit Stitching in UIM**.
4. Click **Submit**.

   If the status is successful, then the topological link is created in UIM and linked to the network location and network entity code.

About Cartridge Modeling

This section describes how the Network Integrity Multi-Domain Circuit Topology Stitching Reconciliation cartridge is modeled in Design Studio.

About the Oracle Communications Information Model

All entities in the Optical UIM Integration cartridge comply with the Oracle Communications Information Model 8.0 for static fields. The dynamic fields (sometimes referred to as characteristics) are application-specific.

This involves first modeling the inventory (UIM) specifications in an inventory cartridge using Design Studio so that the Discovery model is converted into a UIM model. Next, you must define the cartridge dependency so that the Network Integrity cartridge is dependent on the inventory cartridge. Finally, the inventory cartridge specifications can be used in the Network Integrity cartridge model.

Reconciling Scans into UIM

You need to convert the discovery scan results (in the Discovery model) into the reconciliation model (UIM model) to be reconciled into the inventory system.

During reconciliation, all discovered topological links are created as Connectivities in UIM. The specification for these newly created UIM Connectivities is set as **Channelized Connectivity**. Furthermore, the end points of the topological links are mapped to the Pipe Termination Points of the linked device interfaces (STM16, STM64, STM1, OTU4, ODU4, etc.) in UIM.

Table 2-1 describes the mapping between the discovery models and the reconciliation models.

Table 2-1 Discovery Model to UIM Reconciliation Model

Discovery Model	UIM Model
TopologicalLink is modeled as pipe of spec topologicalLink	topologicalLink is created as Channelized Connectivity
TopologicalLink aEnd and Zend is modeled as Pipe Termination Point of spec portTerminationPoint	portTerminationPoint is present as Device Interface of Device (STM16, STM64, STM1, OTU4, ODU4, etc)

Table 2-2 describes the attribute mapping for the Channelized Connectivity in the reconciliation model.

Table 2-2 Attribute Mapping for the Channelized Connectivity Object

Information Model Attribute in Discovery Model	Information Model Support in UIM	Type
description	Static	Text
gapPipe	Static	Boolean, always set to True.
id	Static	Text The value is derived. Possible values are PRIMARY, BACKUP.
medium	Static	Text
name	Static	Text
transmissionSignalType	Static	Text
versioned	Static	Boolean
aDeviceIPAddress	Chars	Text
aEndEMS	Chars	Text
aEndPoint	Chars	Text
associatedTrail	Chars	Text
circleName	Chars	Text
connectivityType	Chars	Text
consistentState	Chars	Text
customer	Chars	Text
directionType	Chars	Text
diverseRouteSearchEffort	Chars	Text
domain	Chars	Text
endPointA	Chars	Text
endPointACircle	Chars	Text
endPointANativeEmsName	Chars	Text
endPointZ	Chars	Text
endPointZCircle	Chars	Text
endPointZNativeEmsName	Chars	Text
hangingType	Chars	Text
inconsistencyReason	Chars	Text
layerRate	Chars	Text
linkCost	Chars	Text
linkCreationDate	Chars	Text
linkDistance	Chars	Text
linkName	Chars	Text
linkQuality	Chars	Text
linkType	Chars	Text
numberOfDiverseRoutes	Chars	Text
protection	Chars	Text
ringName	Chars	Text
sourceOfTL	Chars	Text
srlgDetails1	Chars	Text
srlgDetails2	Chars	Text

Table 2-2 (Cont.) Attribute Mapping for the Channelized Connectivity Object

Information Model Attribute in Discovery Model	Information Model Support in UIM	Type
srlgDetails3	Chars	Text
status	Chars	Text
technology	Chars	Text
tlName	Chars	Text
trailId	Chars	Text
trailName	Chars	Text
VCATSize	Chars	Text
Vendor	Chars	Text
zDeviceIPAddress	Chars	Text
zEndEMS	Chars	Text
zEndPoint	Chars	Text

Discovery Model to UIM Reconciliation Model

Table 2-3 Discovery Model to UIM Reconciliation Model

Discovery Model	UIM Model
Channelized Connectivity	Channelized Connectivity is mapped to pipe of spec topologicalLink.
Device Interface of Device. any one of them (STM16, STM64, STM1, OTU4, ODU4 etc)	Device Interface of Device is mapped as Pipe Termination Point of spec portTerminationPoint.

Table 2-4 Attribute Mapping for the topologicalLink Object

Information Model Attribute	Information Model Support	Typre
description	Static	Text
gapPipe	Static	Boolean, always set to True.
id	Static	Text The value is derived. Possible values are PRIMARY, BACKUP.
medium	Static	Text
name	Static	Text
transmissionSignalType	Static	Text
versioned	Static	Boolean
aDeviceIPAddress	Chars	Text
aEndEMS	Chars	Text
aEndPoint	Chars	Text
associatedTrail	Chars	Text
circleName	Chars	Text
connectivityType	Chars	Text
consistentState	Chars	Text
customer	Chars	Text

Table 2-4 (Cont.) Attribute Mapping for the topologicalLink Object

Information Model Attribute	Information Model Support	Typre
directionType	Chars	Text
diverseRouteSearchEffort	Chars	Text
domain	Chars	Text
endPointA	Chars	Text
endPointACircle	Chars	Text
endPointANativeEmsName	Chars	Text
endPointZ	Chars	Text
endPointZCircle	Chars	Text
endPointZNativeEmsName	Chars	Text
hangingType	Chars	Text
inconsistencyReason	Chars	Text
layerRate	Chars	Text
linkCost	Chars	Text
linkCreationDate	Chars	Text
linkDistance	Chars	Text
linkName	Chars	Text
linkQuality	Chars	Text
linkType	Chars	Text
numberOfDiverseRoutes	Chars	Text
protection	Chars	Text
ringName	Chars	Text
sourceOfTL	Chars	Text
srlgDetails1	Chars	Text
srlgDetails2	Chars	Text
srlgDetails3	Chars	Text
status	Chars	Text
technology	Chars	Text
tlName	Chars	Text
trailId	Chars	Text
trailName	Chars	Text
VCATSize	Chars	Text
Vendor	Chars	Text
zDeviceIPAddress	Chars	Text
zEndEMS	Chars	Text
zEndPoint	Chars	Text

Table 2-5 Model Mapping for the PipeTerminationPoint Object

Information Model Attribute	Information Model Support	Typre
name	Static	Text The name of the PTP (port) cross-connect endpoint.
id	Static	Text

Table 2-5 (Cont.) Model Mapping for the PipeTerminationPoint Object

Information Model Attribute	Information Model Support	Typre
description	Static	Text
device	Dynamic	Text
directionality	Dynamic	Text
rate	Dynamic	Text This value is derived from the line layer rate for the endPort represented by the PortTerminationPoint.
channel	Dynamic	Text This attribute is not used.

About Design Studio Construction

This section describes how the Network Integrity Multi-Domain Circuit Topology Stitching Reconciliation cartridge is built from the Design Studio perspective.

Model Collections

Model Collections shows the Connectivity specifications.

Table 2-6 Connectivity Specifications

Specification Name	UIM Entity Type
Channelized Connectivity	Connectivity
Topological Link	Pipe
Port	PipeTerminationPoint

Actions

Table 2-7 outlines the Design Studio construction of the Multi-Domain Circuit Topology Stitching Reconciliation cartridge actions.

Table 2-7 Actions of Design Studio Construction

Action	Result Category	Address Handler	Scan Parameter Groups	Processors
Detect Discrepancy of Links Circuit Stitching	TopologicalLink	N/A	N/A	Circuit Stitching TL DD Filters Discrepancy Detector
Import TMF814 Links Circuit Stitching from UIM	TopologicalLink	N/A	FilterLinks IncrementalScanParameter	ProcessScanFilterUIM Collect Links Model Links

Table 2-7 (Cont.) Actions of Design Studio Construction

Action	Result Category	Address Handler	Scan Parameter Groups	Processors
Resolve TMF814 Topological Links Circuit Stitching in UIM	TopologicalLink	N/A	N/A	UIM Resolution Framework Initializer UIM Resolution Initializer TMF814 Circuit Stitching Resolution Initializer UIM Resolution Framework Dispatcher

Scan Parameter Groups

Scan Parameter Groups contains information about the scan parameters in Design Studio for the Multi-Domain Circuit Topology Stitching Reconciliation cartridge.

Table 2-8 FilterLinks Scan Parameter Group

Parameter	Description	Filter Links
circleName	NMS Circle identifier	Nms Circle Name
vendorName	NMS Vendor identifier	Nms Vendor Name

Network Integrity System Configuration for Circuit Stitching

You can customize the circuit stitching functionality by using the following properties included in the **system-config.properties** file located in the *domain_home***/ni/config** directory.

Table 2-9 Network Integrity System Configuration Properties for Circuit Stitching

Property Name	Default Value	Expected Value
circuit.stitching.batch.size	500	This numeric count (in minutes) defines how many CORBA element will be fetched from CORBA at one go. The maximum value for this property is 5000.
circuit.stitching.discovery.task.timeout	60	This numeric value (in minutes) defines the timeout of collector task from CORBA. The maximum value for this property is 180.
circuit.stitching.single.modelling.task.timeout	5	This numeric value (in minutes) defines the timeout of one topological link modeling task. The maximum value for this property is 30.
circuit.stitching.complete.modelling.task.timeout	120	This numeric value (in minutes) defines the timeout to complete topological link modeling task. The maximum value for this property is 720.

Table 2-9 (Cont.) Network Integrity System Configuration Properties for Circuit Stitching

Property Name	Default Value	Expected Value
circuit.stitching.unknown.device.identifier	UME	UME Identification : LSN:NMS/UME_BARODA_DWDM14-1236718543956 Here, the identifier is **UME**.
circuit.stitching.unknown.device.matching.operation	afterumeidentifierIndex	The expected value for this property is **afterumeidentifierIndex**, **beforeumeidentifierIndex**, or **removeumeidentifier**.

Note

- Unknown device identification is a critical feature of circuit stitching. You can customize the functionality provided.
- If you customize the functionality, you must do it in the *oracle.communications.integrity.abstractcorbacartridge.network.ume.UMEManager* class.