Oracle® Communications Network Integrity

SNMP Discovery and UIM Integration Cartridge Guide

Release 7.5
G13608-01
December 2024

ORACLE

Oracle Communications Network Integrity SNMP Discovery and UIM Integration Cartridge Guide, Release 7.5

G13608-01

Contents

4 Using the Cartridge

5 About Cartridge Modeling

6 About Model Correction

Preface

This guide describes the functionality and design of the Oracle Communications Network Integrity generic SNMP cartridge.

Audience

This guide is intended for Network Integrity administrators who want to understand the design and evaluate the functionality of this cartridge, and for Network Integrity developers who want either to build or to extend similar cartridges.

Developers should have a good working knowledge of SNMP and SNMP operations, specifications, Network Integrity, UIM, and the use of Oracle Communications Design Studio for Network Integrity.

You should be familiar with the following documents included with this release:

- *Oracle Communications Network Integrity Concepts*
- *Oracle Communications Network Integrity Developer's Guide*
- *Oracle Communications Network Integrity MIB-II SNMP Cartridge Guide*

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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1
Overview

This chapter provides an overview of the Oracle Communications Network Integrity Generic SNMP cartridge.

This chapter contains the following sections:

About the Generic SNMP Cartridge

The Generic SNMP cartridge provides functionality for the following:

- Generic management information base (MIB) MIB-II logical discovery and modeling
- Generic device logical discovery and modeling of asynchronous transfer mode (ATM) media configurations
- Generic device physical discovery and modeling

This cartridge produces both logical and physical device hierarchies that represent a discovered device. The logical hierarchy includes a logical device, child interfaces, sub-interfaces (collectively called interfaces), and device interface configurations. The physical hierarchy includes physical device, equipment, equipment holders, and physical ports. In addition, this cartridge creates associations between the physical and logical hierarchies. The first association is at the device level, between the physical device and the logical device, and the second association is at the interface level between physical ports and Interfaces.

This cartridge discovers the following network entities:

- Physical device
- Logical Device
- Chassis
- Shelf
- Card
- Port
- Interface / Sub-Interface

Discovery scans require that devices support the following MIBs to acquire complete and meaningful results and to properly model both the physical and logical sides.

- For MIB-II logical discovery, the following MIBs are required:
  - RFC1213-MIB
  - IF-MIB (RFC 2863)

  - IANAifType-MIB
  - IP-MIB (RFC 4293)
- For Generic device physical discovery, the ENTITY-MIB (RFC 2737) is required.
- For Generic devices, the ATM-MIB (RFC 2515) is required.

The discovery of a specific device fails if the target device does not support MIB-II or any of the MIBs being used. This does not necessarily mean the scan in its entirety fails. For example, if you have a scan with a broad scope (for example, multiple IP addresses), only the devices that do not meet the required criteria fail to be rendered; other devices within the scan may succeed. A user can inspect the failed device to determine the cause of the failure.

This cartridge is designed to be used on a standalone basis displaying the model hierarchy in Network Integrity. The cartridge provides no integration with other products but may be extended.

CPU Utilization-enabled Discovery

The Generic SNMP cartridge enables you to discover devices based on their CPU utilization by setting a threshold value (in percentage) in the Discover Generic SNMP scan. If the CPU utilization value of a device exceeds the user-specified threshold value, the scan for that device is skipped. The discovery scans are run only for those devices whose CPU utilization value is less than the user-specified threshold value.

In the Discover Generic SNMP scan, you can specify the CPU utilization threshold value between 1% and 99%. If a device's CPU utilization is greater than the specified threshold value, the scan for that device stops, the status of the scan changes to Failed, and the following error message is displayed in the scan results:

```
The device's CPU Utilization of device_cpu_value was greater than the threshold value
set by the user.
```

Where:

device_cpu_value is the CPU utilization value of the device.

For example, if you specified 50% as the CPU utilization value and if the device's CPU utilization is 60%, the Network Integrity application skips scanning that device because its CPU utilization (60%) is greater than the specified threshold value (50%). In this case, the following error message is displayed in the scan results:

```
The device's CPU Utilization of 60 was greater than the threshold value set by the user.
```

To support the discovery of devices based on CPU utilization, a new scan parameter group, **CPU Utilization Parameters**, has been added in the NetworkIntegritySDK cartridge. This scan parameter group adds the **CPU Utilization %** field to the Network Integrity UI Scan Configuration screen. The **CPU Utilization %** field accepts a value between 1 to 99. See "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide* for more information.

When creating a discovery scan based on CPU utilization, ensure that you do the following:

- On the **General** tab, from the **Scan Action** list, select **Discover Generic SNMP**.
- In the **Scan Action Parameters** section, from the **Select Parameter Group** list, select **CPU Utilization Parameters**.
- In the **CPU Utilization %** field, specify a value between 1 to 99.

See "Using Network Integrity" in *Network Integrity Help* for information about creating a scan.

> **Note:**
>
> The CPU utilization feature is disabled by default. To enable it, Network Administrators must follow these steps:
>
> - Create a collector processor with the appropriate Device MIBs to gather CPU utilization values from the device.
> - Place the created collector processor before the Generic Device CPU Set Processor.
> - Set the collected CPU utilization value in the 'device_set_cpu' variable inside Generic Device CPU Set Processor to activate this feature.

About Cartridge Dependencies

This section provides information on dependencies that the Generic SNMP cartridge has on other entities.

Run-Time Dependencies

This cartridge requires that the Address_Handlers cartridge be deployed to Network Integrity.

Design-Time Dependencies

The Generic SNMP cartridge has the following dependencies:

- Address_Handlers
- Generic_SNMP_Model
- MIB_II_SNMP cartridge
- NetworkIntegritySDK
- ora_ni_uim_device
- UIM_Integration_Cartridge

Opening the Cartridge Files in Design Studio

To review and extend the Generic SNMP cartridge, you must first download the Oracle Communications Generic SNMP Cartridge software from the Oracle software delivery web site:

`https://edelivery.oracle.com`

The software contains the Generic SNMP cartridge ZIP file, which has the following structure:

- **\UIM_Cartridge_Projects**
- **\Network_Integrity_Cartridge_Projects**
- **\SNMP_MIBs**
- **Address_Handlers-R7.iar**

- **Generic_SNMP_Cartridge-R7.iar**

See "Getting Started with Design Studio for Network Integrity (1)" *Design Studio Online Help* and "Using Design Studio to Extend Network Integrity" in *Oracle Communications Network Integrity Developer's Guide* for information about opening files in Design Studio.

Building and Deploying the Cartridge

See Getting Started with Design Studio for Network Integrity (1) in *Design Studio Online Help* for information about building and deploying cartridges.

2

About the Cartridge Components

This chapter provides information about the components of the Oracle Communications Network Integrity Generic SNMP cartridge.

The Generic SNMP cartridge contains the following action:

Discover Generic SNMP Action

Discover Generic SNMP Action

The Discover Generic SNMP action scans generic devices and provides a physical and logical hierarchical model of what is discovered. This action also models the associations between the physical and logical hierarchies.

This discovery action extends the Discover MIB-II SNMP action (from the MIB-II SNMP cartridge) and inherits all its processors. For information about the inherited processors, see "Overview" in *Network Integrity MIB-II SNMP Cartridge Guide*.

This action also extends the Abstract CPU Utilization Discovery action (from the NetworkIntegritySDK cartridge) to provide CPU utilization enabled discovery. For information about the processors inherited from the Abstract CPU Utilization Discovery action, see "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide*.

The Discover Generic SNMP action contains the following processors that run in the order listed as follows:

1. CPU Property Initializer (inherited)
2. Generic Device CPU Set Processor
3. CPU Utilization Compare Processor (inherited)
4. MIB II Properties Initializer (inherited)
5. DI Name remodel Initializer (inherited)
6. MIB II SNMP Collector (inherited)
7. MIB II SNMP Modeler (inherited)
8. Generic SNMP Logical Collector
9. Generic SNMP Logical Modeler
10. DI Name Remodeler (inherited)
11. Generic SNMP Physical Collector
12. Generic SNMP Physical Modeler

Generic Device CPU Set Processor

This processor is used to set the device's CPU utilization threshold value, which is collected by the CPU Collector custom processor created by Network Administration when enabling the CPU Utilization feature.

To enable CPU Utilization feature, see "CPU Utilization-enabled Discovery".

Generic SNMP Logical Collector

This processor collects asynchronous transfer mode (ATM) data from the device. For more information, see "About Poll Lists".

Generic SNMP Logical Modeler

This processor models the data collected from the Generic SNMP Logical Collector.

Generic SNMP Physical Collector

This processor collects the physical aspects (such as chassis, container, module, port) of the device. For more information, see "About Poll Lists".

Generic SNMP Physical Modeler

This processor models the data collected from the Generic SNMP Physical Collector.

Detect Generic Device UIM Discrepancies

The Detect Generic Device UIM Discrepancies action detects discrepancies between discovery scan results of the action Discover Generic SNMP Device and data imported from UIM.

This discrepancy detection action extends the Abstract Detect UIM Discrepancies action (from the UIM Integration cartridge) and inherits all its processors. For information about the inherited processors, see "Overview" in *Network Integrity UIM Integration Cartridge Guide*.

The Detect Generic Device UIM Discrepancies action contains the following processors that run in the following order:

1. UIM Discrepancies Filter Initializer (inherited)
2. Discrepancy Detector (inherited)

Resolve Generic Device in UIM

The Resolve Generic Device in UIM action resolves discrepancies on logical and physical hierarchies and associations between the logical and physical entities in UIM.

The discrepancy resolution action extends the Abstract Resolve in UIM action (from the UIM Integration cartridge) and inherits all its processors. For information about these inherited processors, see "Overview" in *Network Integrity UIM Integration Cartridge Guide*.

The Resolve Generic Device in UIM action contains the following processors that run in the following order:

1. UIM Resolution Framework Initializer (inherited)
2. UIM Resolution Initializer (inherited)
3. UIM Resolution Framework Dispatcher (inherited)

3

About Poll Lists

This chapter provides poll lists for processors in the Oracle Communications Network Integrity Generic SNMP cartridge.

MIB-II SNMP Collector Poll List

The following list shows the poll lists for the MIB-II SNMP Collector:

- RFC1213-MIB

  - .mgmt.mib-2.system.sysObjectID
  - .mgmt.mib-2.system.sysDescr
  - .mgmt.mib-2.system.sysName
  - .mgmt.mib-2.system.sysLocation
  - .mgmt.mib-2.interfaces.ifNumber
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifIndex
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifSpeed
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifLastChange
  - .mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr
  - .mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex
  - .mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask
  - .mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr

- IF-MIB

  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName
  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHighSpeed
  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifPromiscuousMode
  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifConnectorPresent
  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifAlias
  - .mgmt.mib-2.ifMIB.ifMIBObjects.ifStackTable.ifStackEntry.ifStackStatus
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifType
  - .mgmt.mib-2.interfaces.ifTable.ifEntry.ifOperStatus

- IP-MIB

  - .mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressPrefix
  - .mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressType
  - .mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressIfIndex

Generic SNMP Logical Collector Poll List

The following list shows the poll lists for the Generic SNMP Collector:

- ATM-MIB

```
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVclAdminStatu
s
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVclOperStatus
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVclLastChange
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVccAalType
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVccAal5Encaps
Type
- .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccCrcErro
rs
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceMaxVpcs
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceMaxVccs
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceMaxActiveVpiBits
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceMaxActiveVciBits
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceIlmiVpi
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceIlmiVci
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceAddressType
- .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEn
try.atmInterfaceAdminAddress
- .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccSarTime
Outs
- .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccOverSiz
edSDUs
```

Generic SNMP Physical Collector Poll List

The following list shows the poll lists for the Generic SNMP Physical Collector:

- Entity-MIB

```
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalDescr
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalVendorType
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalContainedIn
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalClass
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalParentRelPos
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalFirmwareRev
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalSoftwareRev
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalHardwareRev
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
```

```
tPhysicalEntry.entPhysicalSerialNum
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalMfgName
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalModelName
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalAlias
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalAssetID
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.en
tPhysicalEntry.entPhysicalIsFRU
- .mgmt.mib-2.entityMIB.entityMIBObjects.entityMapping.entAliasMappingTable
.entAliasMappingEntry.entAliasMappingIdentifier
```

4

Using the Cartridge

This chapter provides instructions for using the Oracle Communications Network Integrity Generic SNMP cartridge in Network Integrity.

Creating a Discover Generic SNMP Device Scan

Discover Generic SNMP Device scan is used to discover and model data of various device types supported by different vendors.

The Discover Generic SNMP Device discovery scan discovers network information and creates network entities Physical Device, Logical Device, chassis, shelf, card, port, interface/ sub-interface and ATM Information.

To create a Discover Generic SNMP Device discovery scan:

1. Create a new scan.

   For more information, see "Using Network Integrity" in *Network Integrity Online Help*.
2. On the General tab, do the following:
   - From the Scan Action list, select **Discover Generic SNMP Device**.

     The Scan Type field displays *Discovery*.
   - In the **Scan Action Parameters** section, configure the scan with appropriate credentials and version details.
3. On the **Scope** tab, specify addresses.
4. Save and run the scan.

   The scan discovers and models logical and physical devices.

The scan creates a device model for each logical and physical device. The physical device structure uses GenericPhysicalDeviceSpecification and various references defined in the Discover Generic SNMP action. For more information, see "Model Collections".

Populating UIM with Discovered Data for Generic SNMP

This procedure describes steps to populate UIM with network data discovered by the Discover Generic SNMP Device discovery action.

To populate UIM with discovered network data:

1. Create a new scan.

   For more information, see "Using Network Integrity" in *Network Integrity Online Help*.
2. On the General tab of the Create Scan page, do the following:
   - From the Scan Action list, select **Discover Generic SNMP Device**.

     The Scan Type field displays *Discovery*.
   - Select **Detect Discrepancies**.

- In the **Scan Action Parameters** area, make any necessary configurations.

3. Save the scan.
4. Run the discovery scan.

   The scan generates Entity+ discrepancies for each discovered device.
5. Right-click on the discrepancies you want to populate into UIM and select **Correct in UIM**.
6. Click Submit.
7. Verify that UIM is populated with the discovered data.

> **Note:**
>
> The import action "Import from UIM" available in UIM_Integration_Cartridge can be used to import data from UIM.

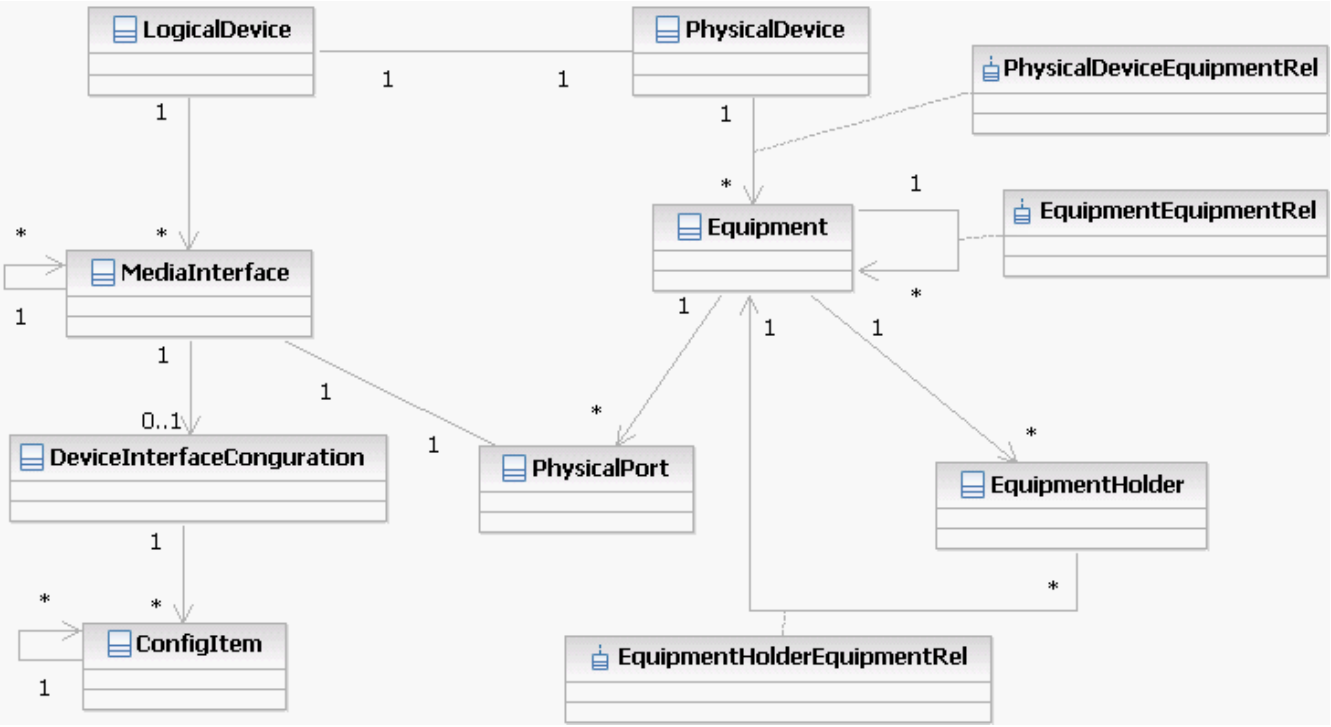
5
About Cartridge Modeling

This chapter provides information on modeling the Network Integrity Generic SNMP cartridge.

Generic SNMP Cartridge UML Representation

Figure 5-1 displays a unified modeling language (UML) diagram depicting the object relationship being rendered.

Figure 5-1 Generic SNMP Cartridge UML Representation



Hierarchy Mapping

The data sourced from RFC1213-MIB.mgmt.mib-2.system tables establishes and seeds the logical device object.

The media interface encapsulates the common information about an interface as a device is discovered. The device interface configuration captures the media type information that decorates the interface with media-specific parameters. These media-specific parameters define the behavior of the interface (Generic, ATM).

The media interfaces are established and seeded with data sourced from the following:

- RFC1213-MIB.mgmt.mib-2.system

- RFC1213-MIB.mgmt.mib-2.interfaces.ifTable
- IF-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry
- IP-MIB.mgmt.mib-2.ip.ipAddressTable.ipAddressEntry
- IF-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifStackTable.ifStackEntry.ifStackStatus

IF-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifStackTable.ifStackEntry.ifStackStatus establishes the interface hierarchy.

The generic media device interface configuration is established and seeded with data sourced from the following:

- RFC1213-MIB.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry
- IP-MIB.mgmt.mib-2.ip.ipAddressTable.ipAddressEntry

The ATM media device interface configuration is established and seeded with data sourced from the ATM-MIB. See "About Poll Lists".

Oracle Communications Information Model Information

All entities shown in Figure 5-1 (for example, physical device, logical device, media interface, and so on) are Oracle Communications Information Model 1.0-compliant for static fields. The dynamic fields (sometimes referred to as characteristics) are application-specific. You can customize application specific data with the device interface configuration mechanism.

The Generic SNMP cartridge supports the following configurations:

- Generic Media
- AtmMedia

For a listing of the Information Model fields, see "Logical Mapping" and "Physical Mapping".

Specifications

This section lists the specifications included in the ora_ni_uim_device cartridge for modeling Generic vendor devices.

You must first model inventory (UIM) specifications in an inventory cartridge using Design Studio, define the cartridge dependency such that the Network Integrity cartridge is dependent on the inventory cartridge, and then use the inventory cartridge specifications in the Network Integrity cartridge model.

Specifications shared with Oracle Communications Unified Inventory Management (UIM) are defined in the ora_ni_uim_device cartridge. These cartridges are used to directly deploy specifications to UIM.

PhysicalDevice

This specification models a physical device entity.

Table 5-1 shows the specifications for physical device.

Table 5-1 Physical Device Specifications

Specification	Cartridge	Intended Usage
GenericPhysicalDeviceSpecification	ora_ni_uim_device	Used to model all types of generic devices.

GenericPhysicalDeviceSpecification is available in ora_ni_uim_device and used to model all type of generic devices.

Table 5-2 shows the characteristics applied to physical device specifications.

Table 5-2 Physical Device Characteristics

Characteristics	Field Type	Field Content
mgmtIpAddress	String	Text
modelName	String	Text
discoveredModelNumber	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
serialNumber	String	Text
physicalLocation	String	Text
hardwareRev	String	Text
softwareRev	String	Text
nativeEmsName	String	Text

Equipment

Table 5-3 Equipment Specifications

Specification	Cartridge	Intended Usage
GenericEquipmentShelfSpecification	ora_ni_uim_device	Used to model shelf for all generic devices.
GenericEquipmentSpecification	ora_ni_uim_device	Used to model card for all generic devices.

GenericEquipmentShelfSpecification and GenericEquipmentSpecification are available in ora_ni_uim_device and used to model generic devices.

Table 5-4 GenericEquipmentShelfSpecification Characteristics

Characteristics	Field Type	Field Content
discoveredModelNumber	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
hardwareRev	String	Text
softwareRev	String	Text

Table 5-4 (Cont.) GenericEquipmentShelfSpecification Characteristics

Characteristics	Field Type	Field Content
modelName	String	Text
nativeEmsName	String	Text
owner	String	Text
userLabel	String	Text

Table 5-5 GenericEquipmentSpecification Characteristics

Characteristics	Field Type	Field Content
discoveredModelNumber	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
hardwareRev	String	Text
softwareRev	String	Text
modelName	String	Text
nativeEmsName	String	Text
owner	String	Text
userLabel	String	Text

Equipment Holder

Table 5-7 shows the characteristics applied to the EquipmentHolder specifications.

Table 5-6 Equipment Holder Specifications

Specification	Cartridge	Intended Usage
GenericEquipmentHolderSpecification	ora_ni_uim_device	Used to model all types of Generic Vendor devices.

GenericEquipmentHolderSpecification is available in ora_ni_uim_device and used to model Generic Vendor devices.

Table 5-7 EquipmentHolder Characteristics

Characteristics	Field Type	Field Content
modelName	String	Text
nativeEmsName	String	Text
discoveredModelNumber	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
hardwareRev	String	Text
softwareRev	String	Text
owner	String	Text
userLabel	String	Text

Physical Port

Table 5-8 shows the characteristics applied to physical port specifications.

Table 5-8 Physical Port Characteristics

Characteristics	Field Type	Field Content
modelName	String	Text
discoveredModelNumber	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
serialNumber	String	Text
physicalLocation	String	Text
hardwareRev	String	Text
softwareRev	String	Text
nativeEmsName	String	Text
physicalAddress	String	Text

Table 5-9 Physical Port Characteristics

Characteristics	Field Type	Field Content
direction	String	Text
discoveredVendorName	String	Text
discoveredPartNumber	String	Text
discoveredVendorName	String	Text
hardwareRev	String	Text
softwareRev	String	Text
edgePoint	String	Text
nativeEmsName	String	Text
tpProtectionAssociation	String	Text
portLevel	String	Text
portType	String	Text

Logical Device

For logical device and device interface specification details, see "Overview" in *Network Integrity MIB-II SNMP Cartridge Guide*.

Specification Cardinality

The cardinality of all specification parent-child relationships is included in the software code so that min= 0 and max= n. This approach allows Network Integrity to programmatically instantiate all objects on demand as they are discovered using the web service.

Field Mapping

The Generic SNMP cartridge supports the following field mappings:

- Text: Implies Text [255]
- static: Information Model 1.0 defines this field to be static on the entity specification. The specification provides getters and setters for this field.
- dynamic: This is a dynamic field where the entity specification treats the field as a name and value pair. The specification does not provide getters and setters but generically has get and set characteristics method holding a HashSet of entries.

Logical Mapping

The Generic SNMP cartridge supports the following logical mappings:

- LogicalDevice
- MediaInterface
- DeviceInterfaceConfigurationItem Mapping (IPv4)
- DeviceInterfaceConfigurationItem Mapping (IPv6)
- DeviceInterfaceConfigurationItem Mapping (ATM Media)
- Mapping Table

LogicalDevice

Table 5-10 shows characteristics for the LogicalDevice specification.

Table 5-10 LogicalDevice Characteristics

Characteristics (LogicalDevice)	Information Model Support	MIB Object	Field Type	Intended Usage
Id	static	N/A	Text	Programmatically generated as MgmtIPAddress::sysName::" LogicalDevice"
Name	static	sysName	Text	N/A
Description	static	sysDescr	Text	N/A
Specification	static	N/A	N/A	Programmatically applies deviceGeneric specification
nativeEmsAdminServiceState	static	N/A	Enum with the following values: • UNKNOWN • IN_SERVICE • OUT_OF_SERVICE • TESTING • IN_MAINTENANCE	Nothing available to source the field
nativeEmsName	static	sysName	Text	N/A

Table 5-10 (Cont.) LogicalDevice Characteristics

Characteristics (LogicalDevice)	Information Model Support	MIB Object	Field Type	Intended Usage
nativeEmsServiceState	static	N/A	Enum with the following values: • UNKNOWN • IN_SERVICE • OUT_OF_SERVICE • TESTING • IN_MAINTENANCE	Nothing available to source the field
mgmtIpAddress	dynamic	N/A	Text	discoveryAddress
sysObjectId	dynamic	sysObjectId	Text	Support legacy systems

MediaInterface

Table 5-11 shows characteristics for the MediaInterface specification.

Table 5-11 MediaInterface Characteristics

Characteristics (MediaInterface)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Id	static	N/A	Text	Programmatically generated as MgmtIPAddress::sysName::ifDescr::"MediaInterface"
Name	static	ifDescr	Text	N/A
Description	static	ifDescr	Text	N/A
Specification	static	N/A	N/A	Programmatically applies interfaceGeneric specification
interfaceNumber	static	N/A	Text	Nothing available to source the field.
customerInterfaceNumber	static	N/A	Text	Nothing available to source the field.
vendorInterfaceNumber	static	ifName	Text	N/A
nativeEmsName	static	ifDescr	Text	The field must be unique. ifDescr guarantees the uniqueness of a device.
nativeEmsAdminServiceState	static	ifAdminStatus	Enum with the following values: • UNKNOWN • IN_SERVICE • OUT_OF_SERVICE • TESTING • IN_MAINTENANCE	Mapped, see Table 5-17.

Table 5-11 (Cont.) MediaInterface Characteristics

Characteristics (MediaInterface)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
nativeEmsServiceState	static	ifOperStatus	Enum with the following values: • UNKNOWN • IN_SERVICE • OUT_OF_SERVICE • TESTING • IN_MAINTENANCE	Mapped, see Table 5-17.
ifType	dynamic	ifType	Text	Mapped from IANA MIB using the snmpIfTypeMap properties file. Look up returning null results in a value of "n".
mtuSupported	static	N/A	Float	Nothing available to source the field. Defaults to 0.0
mtuCurrent	static	ifMtu	Float	N/A
physicalAddress	static	ifPhysAddress	Text	N/A
physicalLocation	static	sysLocation	Text	N/A
minSpeed	static	N/A	Float	Nothing available to source the field. Defaults to 0.0
maxSpeed	static	N/A	Float	Nothing available to source the field. Default to 0.0
nominalSpeed	static	ifSpeed | ifHighSpeed	Float	ifHighSpeed overrides ifSpeed when ifHighSpeed is available
ifAlias	dynamic	ifAlias	Text	N/A
ifName	dynamic	ifName	Text	Support legacy systems

DeviceInterfaceConfigurationItem Mapping (IPv4)

Table 5-12 shows characteristics for the DeviceInterfaceConfigurationItem (IPv4) specification.

Table 5-12 DeviceInterfaceConfigurationItem (IPv4) Characteristics

Characteristics (Generic Media)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
ipAddress	dynamic	ipAdEntAddr	Text	N/A
prefix	dynamic	ipAdEntNetMask	Text	N/A
ipVersion	dynamic	ipAddressType	Enum • IPV4 • IPV6	Programmatically set to IPV4.
Specification	static	N/A	N/A	Programmatically applies GenericMedia specification

DeviceInterfaceConfigurationItem Mapping (IPv6)

Table 5-13 shows characteristics for the DeviceInterfaceConfigurationItem (IPv6) specification.

Table 5-13 Characteristics for the DeviceInterfaceConfigurationItem (IPv6) Specification

Characteristics (Generic Media)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
ipAddress	dynamic	ipAddressIfIndex	Text	For IPv6 addresses the actual IP is derived from the index since ipAddressAddr is not acceptable.
prefix	dynamic	ipAddressPrefix	Text	N/A
ipVersion	dynamic	ipAddressType	Enum • IPV4 • IPV6	Programmatically set to IPV6.
Specification	static	N/A	N/A	Programmatically applies GenericMedia specification

DeviceInterfaceConfigurationItem Mapping (ATM Media)

Table 5-14 shows characteristics for the DeviceInterfaceConfigurationItem mapping (ATM media) specification.

Table 5-14 Characteristics for the DeviceInterfaceConfigurationItem (ATM media) Specification

Characteristics (ATM Media)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
VPI	dynamic	Table index	Text	Parsed from the table index, which is in the format: <ifIndex>.<VPI>.<VCI>
VCI	dynamic	Table index	Text	Parsed from the table index, which is in the format: <ifIndex>.<VPI>.<VCI>

Table 5-15 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (VirtualChannelLinkData) specification.

Table 5-15 Characteristics for the DeviceInterfaceConfigurationItem (VirtualChannelLinkData) Specification

Characteristics (VirtualChannelLinkData)	Information Model Support	MIB Object	Field Type
atmVccAal5EncapsType	dynamic	atmVccAal5EncapsType	Text
atmVccAalType	dynamic	atmVccAalType	Text
atmVclLastChange	dynamic	atmVclLastChange	Text
atmVclAdminStatus	dynamic	atmVclAdminStatus	Text
atmVclOperStatus	dynamic	atmVclOperStatus	Text

Table 5-16 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (AtmInterfaceConfigurationData) specification.

Table 5-16 Characteristics for the DeviceInterfaceConfigurationItem (AtmInterfaceConfigurationData) Specification

Characteristics (AtmInterfaceConfigurationData)	Information Model Support	MIB Object	Field Type
atmInterfaceMaxVpcs	dynamic	atmInterfaceMaxVpcs	Text
atmInterfaceMaxVccs	dynamic	atmInterfaceMaxVccs	Text
atmInterfaceMaxActiveVpiBits	dynamic	atmInterfaceMaxActiveVpiBits	Text
atmInterfaceMaxActiveVciBits	dynamic	atmInterfaceMaxActiveVciBits	Text
atmInterfaceIlmiVpi	dynamic	atmInterfaceIlmiVpi	Text
atmInterfaceIlmiVci	dynamic	atmInterfaceIlmiVci	Text
atmInterfaceAddressType	dynamic	atmInterfaceAddressType	Text
atmInterfaceAdminAddress	dynamic	atmInterfaceAdminAddress	Text

Mapping Table

Table 5-17 shows a mapping table.

Table 5-17 Mapping Table

ifOperStatus	ifAdminStatus	nativeEmsServiceState nativeEmsAdminState
4: unknown, 6: notPresent	N/A	UNKNOWN
1: up	1: up	IN_SERVICE
2: down, 5: dormant, 7: lowerLayerDown	2: down	Programmatically set to OUT_OF_SERVICE.
3: testing	3: testing	TESTING
N/A	N/A	IN_MAINTENANCE

Physical Mapping

This cartridge support the following physical mappings:

- Information Model Nomenclature Mapping
- PhysicalDevice
- Equipment
- EquipmentHolder
- PhysicalPort

Information Model Nomenclature Mapping

Information Model mapping helps in classification of physical components. For example, if an equipment is classified as "Power Supply," this cartridge maps the SNMP data into the Information Model entity type Equipment. Table 5-18 lists the various physical classes and the Information Model entity type they are mapped to.

Table 5-18 Information Model Nomenclature Mapping

entPhysicalClass	Information Model
root	Physical Device
Other (1)	Unsupported
Unknown (2)	Unsupported
Chassis (3)	Equipment-Shelf
Backplane (4)	Equipment-Shelf
Container (5)	Equipment-Holder
Power supply (6)	Equipment
Fan (7)	Equipment
Sensor (8)	Equipment
Module (9)	Equipment-Card
Port (10)	Physical Port
Stack (11)	Unsupported

PhysicalDevice

Table 5-19 shows the characteristics for the PhysicalDevice specification.

Table 5-19 PhysicalDevice Characteristics

Characteristics (Physical Device)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Id	static	N/A	Text	Programmatically generated as MgmtIPAddress::sysName::0::"PhysicalDevice".
Name	static	sysName	Text	N/A
Description	static	sysDescr	Text	N/A
Specification	static	N/A	Text	Programmatically applies GenericPhysicalDeviceSpecification specification.
discoveredModelNumber	dynamic	sysObjectId	Text	N/A
discoveredVendorName	dynamic	N/A	Text	Set to "Network Vendor".
serialNumber	static	N/A	Text	N/A
physicalLocation	N/A	location	Text	N/A
discoveredPartNumber	dynamic	N/A	Text	N/A
hardwareRev	dynamic	N/A	Text	N/A
softwareRev	dynamic	N/A	Text	N/A
ModelName	dynamic	N/A	Text	Set to "Network Equipment Type".
mgmtIpAddress	dynamic	N/A	Text	discoveryAddress
nativeEmsName	static	sysName	Text	N/A

Equipment

Table 5-20 shows characteristics for the Equipment specification.

Table 5-20 Equipment Characteristics

Characteristics (Equipment)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Id	static	N/A	Text	Programmatically set to MgmtIPAddress::sysName::[absRelativePosition]::"Equipment.Rack|Shelf|Card". See "About the absRelativePosition Value".
Name	static	entPhysicalDescr	Text	N/A
Description	static	entPhysicalDescr	Text	N/A
Specification	static	N/A	Text	Programmatically applies GenericEquipmentShelfSpecification or GenericEquipmentSpecification based on device type (shelf/card).
discoveredModelNumber	dynamic	entPhysicalVendorType	Text	N/A
discoveredVendorName	dynamic	entPhysicalMfgName	Text	N/A
serialNumber	static	entPhysicalSerialNum	Text	N/A
physicalLocation	static	location	Text	N/A
discoveredPartNumber	dynamic	entPhysicalModelName	Text	N/A
hardwareRev	dynamic	entPhysicalHardwareRev	Text	N/A
softwareRev	dynamic	entPhysicalSoftwareRev	Text	N/A
ModelName	dynamic	N/A	Text	Programmatically applies based on entPhysicalDescr or entPhysicalModelName.
nativeEmsName	static	entPhysicalDescr and absRelativePosition	Text	N/A

EquipmentHolder

Table 5-21 shows characteristics for the EquipmentHolder specification.

Table 5-21 EquipmentHolder Characteristics

Characteristics (EquipmentHolder)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Id	static	N/A	Text	Programmatically generated as MgmtIPAddress::sysName::[absRelativePosition]::"EquipmentHolder". See "About the absRelativePosition Value".

Table 5-21 (Cont.) EquipmentHolder Characteristics

Characteristics (EquipmentHolder)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Name	static	entPhysicalDescr and entPhysicalParentRelPos	Text	Programmatically generated as entPhysicalDescr::entPhysicalParentRelPos.
Description	static	entPhysicalDescr	Text	N/A
Specification	static	N/A	Text	Programmatically applies GenericEquipmentHolderSpecification.
discoveredModelNumber	dynamic	entPhysicalVendorType	Text	N/A
discoveredVendorName	dynamic	entPhysicalMfgName	Text	N/A
serialNumber	static	entPhysicalSerialNum	Text	N/A
physicalLocation	static	location	Text	N/A
discoveredPartNumber	dynamic	entPhysicalModelName	Text	N/A
hardwareRev	dynamic	entPhysicalHardwareRev	Text	N/A
softwareRev	dynamic	entPhysicalSoftwareRev	Text	N/A
ModelName	dynamic	entPhysicalDescr or entPhysicalModelName	Text	Programmatically applies based on entPhysicalDescr or entPhysicalModelName
nativeEmsName	static	entPhysicalParentRelPos	Text	N/A

PhysicalPort

Table 5-22 shows characteristics for the PhysicalPort specification.

Table 5-22 PhysicalPort Characteristics

Characteristics (PhysicalPort)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
Id	static	N/A	Text	Programmatically generated as MgmtIPAddress::sysName::[absRelativePosition]::"PhysicalPort". See "About the absRelativePosition Value".
Name	static	entPhysicalDescr and entPhysicalParentRelPos	Text	Programmatically generated as entPhysicalDescr::entPhysicalParentRelPos.
Description	static	entPhysicalName	Text	N/A
Specification	static	N/A	Text	Programmatically applies GenericPhysicalPortSpecification.
discoveredModelNumber	dynamic	entPhysicalVendorType	Text	N/A

Table 5-22 (Cont.) PhysicalPort Characteristics

Characteristics (PhysicalPort)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
portNumber	static	entPhysicalParentRelPos	Int	N/A
customerPortName	static	entPhysicalParentRelPos	Text	N/A
vendorPortName	static	N/A	Text	N/A
discoveredVendorName	dynamic	entPhysicalMfgName	Text	N/A
serialNumber	static	entPhysicalSerialNum	Text	N/A
physicalLocation	static	location	Text	N/A
discoveredPartNumber	dynamic	entPhysicalModelName	Text	N/A
hardwareRev	dynamic	entPhysicalHardwareRev	Text	N/A
softwareRev	dynamic	entPhysicalSoftwareRev	Text	N/A
ModelName	dynamic	entPhysicalDescr or entPhysicalModelName	Text	Programmatically applies based on entPhysicalDescr or entPhysicalModelName
nativeEmsName	static	entPhysicalParentRelPos	Text	N/A

6

About Model Correction

This chapter provides model correction information for the Oracle Communications Network Integrity Generic SNMP cartridge.

About Model Correction

Model correction occurs when the SNMP information received through discovery from a generic device does not conform to the Information Model and therefore cannot be persisted as is within Network Integrity.

The Generic SNMP cartridge applies the following model corrections:

- Multiple Equipment Occupying the Same Slot
- Equipment Under Physical Port
- Physical Port Under EquipmentHolder
- EquipmentHolder Under EquipmentHolder
- EquipmentHolder Containing Multiple Equipment
- EquipmentHolder Under Physical Port
- Multiple Equipment Under Physical Port
- Naming Pattern for Artificial Entities

Multiple Equipment Occupying the Same Slot

In this scenario, multiple equipment report the same relative position within the context of the same parent. The Generic SNMP cartridge applies the algorithm detailed in "About the absRelativePosition Value". No hierarchical change is required.

Equipment Under Physical Port

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
           EquipmentHolder
                    Equipment
                         Physical Port
                              Equipment-Card
```

The Generic SNMP cartridge converts the PhysicalPort to an EquipmentHolder as follows:

```
Physical Devices
      Equipment-Shelf
           EquipmentHolder
                    Equipment
                        EquipmentHolder
                                Equipment-Card
```

Physical Port Under EquipmentHolder

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
            EquipmentHolder
                  Physical Port
```

The Generic SNMP cartridge inserts an artificial Equipment entity between the EquipmentHolder and the PhysicalPort as follows:

```
Physical Device
      Equipment-Shelf
            EquipmentHolder
                  Equipment
                        Physical Port
```

EquipmentHolder Under EquipmentHolder

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
             EquipmentHolder
                  EquipmentHolder
```

The Generic SNMP cartridge inserts an artificial Equipment entity between the two EquipmentHolders as follows:

```
Physical Device
      Equipment-Shelf
            EquipmentHolder
                  Equipment
                        EquipmentHolder
```

EquipmentHolder Containing Multiple Equipment

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
            EquipmentHolder
                  Equipment
                  Equipment
```

The Generic SNMP cartridge inserts an artificial Equipment entity as the parent to the two equipments and each equipment will be put under a Equipment Holder as follows:

```
Physical Device
      Equipment-Shelf
      EquipmentHolder
            Equipment
                  EquipmentHolder
                        Equipment
                  EquipmentHolder
                        Equipment
```

EquipmentHolder Under Physical Port

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
           EquipmentHolder
                    Equipment
                        Physical Port
                            EquipmentHolder
```

The Generic SNMP cartridge converts the Physical Port to an Equipment as follows:

```
Physical Devices
      Equipment-Shelf
           EquipmentHolder
                    Equipment
                        Equipment
                            EquipmentHolder
```

Multiple Equipment Under Physical Port

For a discovered generic device hierarchy, such as the following:

```
Physical Device
      Equipment-Shelf
           EquipmentHolder
                    Equipment-Card
                        Physical Port
                            Equipment
                            Equipment
```

The Generic SNMP cartridge first converts the Physical Port to an EquipmentHolder, then inserts an artificial Equipment between the EquipmentHolder and the Equipment, The Equipment and each sub-equipment will then be put under a Equipment Holder as follows:

```
Physical Device
   Equipment-Shelf
       EquipmentHolder
           Equipment-Card
               EquipmentHolder
                   Equipment
                        EquipmentHolder
                           Equipment
                        EquipmentHolder
                           Equipment
```

Naming Pattern for Artificial Entities

Naming pattern for artificial entities are introduced through model corrections.

The following example shows the EquipmentHolder naming pattern.

```
EquipmentHolder-Artificial::entPhysicalParentRelPos
```

The following example shows the Equipment naming pattern.

```
Equipment-Artificial::entPhysicalParentRelPos
```

7

About Design Studio Construction

This chapter provides information on the composition of the Oracle Communications Network Integrity Generic SNMP cartridge from the Oracle Communications Design Studio perspective.

Model Collections

Table 7-1 shows the MIB-II model collection used in the Generic SNMP cartridge.

Table 7-1 MIB-II Model Collection

Specifications	Information Model Entity Type	Intended Usage/ Notes
deviceGeneric	LogicalDevice	Represents any root object discovered on the network.
interfaceGeneric	MediaInterface	Represents any interface discovered under deviceGeneric.
GenericMedia	DeviceInterfaceConfigurationItem	Represents IpAddresses that are applied to an interfaceGeneric.
IPAddresses	DeviceInterfaceConfigurationItem	Ip Addresses container.
IPAddress	DeviceInterfaceConfigurationItem	Ip Address details container.

Table 7-2 shows the Generic SNMP model collection.

Table 7-2 Generic SNMP Model Collection

Specifications	Information Model Entity Type	Intended Usage/ Notes
GenericPhysicalDeviceSpecification	PhysicalDevice	Represents any generic physical device discovered on the network.
GenericEquipmentHolderSpecification	EquipmentHolder	Represents any equipment container.
GenericEquipmentShelfSpecification	Equipment	Represents equipment shelf.
GenericEquipmentSpecification	Equipment	Represents equipment card.
GenericPhysicalPortSpecification	PhysicalPort	Represents any physical port.
AtmMedia	DeviceInterfaceConfigurationItem	Represents an ATM media configuration that can be applied to an interface or sub-interface.
VirtualChannelLink	DeviceInterfaceConfigurationItem	Represents a single Virtual Channel Link within an ATM Media configuration.
VPI	DeviceInterfaceConfigurationItem	Holds the VPI value and is contained within the Virtual Channel Link.
VCI	DeviceInterfaceConfigurationItem	Holds the VCI value and is contained within the Virtual Channel Link.
VirtualChannelLinkData	DeviceInterfaceConfigurationItem	Contains the Virtual Channel Link characteristics and is contained with the Virtual Channel Link.

Table 7-2 (Cont.) Generic SNMP Model Collection

Specifications	Information Model Entity Type	Intended Usage/ Notes
AtmInterfaceConfigurationData	DeviceInterfaceConfigurationItem	Contains the ATM configuration applied to an interface.

Logical Specification Lineage

Example 7-1 shows a logical specification lineage. This lineage shows the intended relationship between specifications.

Example 7-1 Logical Specification Lineage

```
deviceGeneric
    [0..*] interfaceGeneric
        [0..1] GenericMedia
            [0..1] IP Addresses
                [0..*] IpAddress
                    IpAddress (characteristic)
                    Prefix (characteristic)
                    IpVersion (characteristic)
        [0..1] Atm Media
            [0..1] IP Addresses
                [0..*] IpAddress
                    IpAddress (characteristic)
                    Prefix (characteristic)
                    IpVersion (characteristic)
            [0..1] AtmInterfaceConfigurationData
            [0..*] VirtualChannelLink
                [0..1] VPI
                [0..1] VCI
                [0..1] VirtualChannelLinkData
```

Physical Specification Lineage

This section provides an example of a physical specification lineage.

Example 7-2 Physical Specification Lineage

```
GenericPhysicalDeviceSpecification
    [0..*]GenericEquipmentShelfSpecification
        [0..*]GenericEquipmentHolderSpecification
            [1]GenericEquipmentSpecification
                [0..*]GenericPhysicalPortSpecification
                [0..*]GenericEquipmentHolderSpecification
                    [1]GenericEquipmentSpecification
                        [0..*]GenericEquipmentHolderSpecification
                        [0..*]GenericPhysicalPortSpecification
```

Discovery Action

The Generic SNMP Cartridge supports the Discover Generic SNMP Device action. Table 7-3 shows the discovery action in the Generic SNMP cartridge.

Table 7-3 Discover Generic SNMP Action

Result Category	AddressHandler	Scan Parameters	Model	Processors
Device	IPAddressHandler	• version • port • snmpReadCommunity • snmpTimeout • snmpRetries • username • contextName • authProtocol • authPassword • privacyProtocol • privacyPassword **Note:** These scan parameters must be added for the Create Scan web service request even if the values are left empty.	Generic_SNMP_Cartridge	• CPU Property Initializer (inherited) • Generic Device CPU Set Processor • CPU Utilization Compare Processor (inherited) • MIB II Properties Initializer (inherited) • DI Name remodel Initializer (inherited) • MIB II SNMP Collector (inherited) • MIB II SNMP Modeler (inherited) • Generic SNMP Logical Collector • Generic SNMP Logical Modeler • DI Name Remodeler (inherited) • Generic SNMP Physical Collector • Generic SNMP Physical Modeler

Discovery Processors

Table 7-4 shows the processors of the Discover Generic SNMP action.

Table 7-4 Discovery Processors

Processor Name	Variable
CPU Property Initializer	Input: N/A Output: cpuProperties This class is used to initialize and set the default CPU value for the device.
Generic Device CPU Set Processor	Input: Custom CPU Collector processor Response Output: N/A
CPU Utilization Compare Processor	Input: cpuProperties This class is used to initialize and set the default CPU value for the device. Output: N/A

Table 7-4 (Cont.) Discovery Processors

Processor Name	Variable
MIB-II PropertiesInitializer	Input: N/A Output: • snmpIfTypeMap Property map containing a listing of ifTypes to string name. • snmpVendorNameMap Property map containing listing of sysObjectId suffixes to vendorName.
DI Name remodel Initializer (inherited)	Input:N/A Output: remodelerProperties Properties object contains device interface re-modeling information.
MIB II SNMP Collector	Input: N/A Output: mibiisnmpCollectorResponseDocument (implicit) For more information about polled SNMP data, see "About Poll Lists".
MIB II SNMP Modeler	Input: • mibiisnmpCollectorResponseDocument Discovered SNMP data produced by the MIB-II SNMP Collector. • snmpIfTypeMap Property map containing listing of ifTypes to string name. Output: • deviceInterfaceMap A map that contains interfaces with IfIndex as key. • logicalDevice The logical device that was created by the MIB-II Modeler processor.
Generic SNMP Logical Collector	Input: N/A Output: genericSNMPLogicalCollectorResponseDocument (implicit) For more information about polled SNMP data, see "About Poll Lists".

Table 7-4 (Cont.) Discovery Processors

Processor Name	Variable
Generic SNMP Logical Modeler	Input: • genericSNMPLogicalCollectorResponseDocument SNMP discovered data produced by the Generic SNMP Logical Collector. • deviceInterfaceMap A map that contains interfaces with IfIndex as key. • logicalDevice This is the logical device that was created in the MIB-II Modeler. • mibiisnmpCollectorResponseDocument SNMP discovered data produced by the MIB-II SNMP Collector. Output: genericLogicalDevice This is the logical device that was created in the Generic SNMP Logical Modeler.
DI Name Remodeler	Input: • snmpIfTypeMap Property map containing a listing of ifTypes to string name. • snmpVendorNameMap Property map containing listing of sysObjectId suffixes to vendorName. • logicalDevice The logical device that was modified by Generic SNMP Logical Modeler. • deviceInterfaceMap A map that contains interfaces with IfIndex as key. • remodelerProperties Properties object contains device interface Re-Modeling information. Output: N/A
Generic SNMP Physical Collector	Input: N/A Output: genericSNMPPhysicalCollectorResponseDocument (implicit) For more information about polled SNMP data, see "About Poll Lists".

Table 7-4 (Cont.) Discovery Processors

Processor Name	Variable
Generic SNMP Physical Modeler	Input: • genericSNMPPhysicalCollectorResponseDocument SNMP discovered data produced by the Generic SNMP Physical Collector. • logicalDevice This is the logical device that was created in the MIB-II Modeler. • mibiisnmpCollectorResponseDocument SNMP discovered data produced by the MIB-II SNMP Collector. Output: physicalDevice This is the physical device that is created in this Generic SNMP Physical Modeler.

Discrepancy Detection Action

The Detect Generic Device UIM Discrepancies action is used to perform discrepancy detection.

Table 7-5 Detect Generic Device UIM Discrepancies

Result Category	Results Source	Scan Parameters	Model	Processors
All	Discover Generic SNMP Device	N/A	Generic_SNMP_Cartridge	This action extends the Abstract Detect UIM Discrepancies action included in the UIM Integration cartridge. For more information, see "Overview" in *Network Integrity UIM Integration Cartridge Guide*.

Discrepancy Resolution Action

The Resolve Generic Device in UIM action is used to perform discrepancy resolution.

Table 7-6 Resolve Generic Devices in UIM

Result Category	Result Source	Processors
All	Discover Generic SNMP Device	This action extends the Abstract Resolve in UIM action included in the Network Integrity UIM Integration cartridge. For more information, see "Overview" in *Network Integrity UIM Integration Cartridge Guide*.

8

About Design Studio Extension

This chapter provides scenarios for the extensibility of Oracle Communications Network Integrity using Oracle Communications Design Studio.

Adding a New Device Type under Generic

To discover and model a new device type, the Generic SNMP Physical Modeler implementation can be updated with logic to support modeling of the new device type and an additional collector processor can be included to read the data in additional MIBs if required.

For more details regarding extensibility, see "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide*.

A

About the absRelativePosition Value

This appendix explains how to generate the absRelativePosition variable value.

Generating the absRelativePosition Value

The field contains a variable called absRelativePosition which is used to generate a unique value for an entity in a given tree. absRelativePosition is a programmatically generated value and is composed of a prefix and suffix. The suffix is always derived from entPhysicalParentRelPos.

The following is an example of a physical device tree: PD E1 PP1 E2 PP2

The prefix of each entity is derived from the absolute relative position from the root; for example:

PD takes the prefix 0

The child E1 takes the prefix 0:0

The child E2 takes the prefix 0:1, uniquely identifying itself from its sibling.

PP1 takes the prefix 0:0:0

PP2 takes the prefix 0:1:0

Some devices (due to a device reporting error) show that multiple equipment holders or physical ports have entPhysicalParentRelPos value that occupy the same relative position under their parent.

At a high level, the algorithm works by resolving conflicts by increasing the relative position of the subsequent duplicates by 1. This is best described using an example.

Table A-1 shows the applicable SNMP attributes that are used in determining the correct relative position.

Table A-1 SNMP Attributes Used to Determine Correct Relative Position

Index	entPhysicalDesr	entPhysicalParentRelPos	entPhysicalContainedIn
1	3640 chassis, Hw Serial#: 621974280, Hw Revision: 00	-1	0
2	3640 Chassis Slot	0	1
3	Ethernet/WAN	0	2
12	AmdP2	0	3
13	AmdP2	0	3
14	AmdP2	1	3
15	AmdP2	1	3

Where the headings in Table A-1 have the following significance:

- Index: A numeric value used to represent a physical entity and must be unique.
- entPhysicalDesr: A string description of the physical entity.
- entPhysicalParentRelPos: The relative position within the context of its parent.
- entPhysicalContainedIn: The Index of the parent entity denoting the current entity as a child. 0 indicates the root of the physical entity tree.

From Table A-1, notice that AmdP2 at index 12 and 13, and AmdP2 at index 14 and 15 have the same entPhysicalParentRelPos values (that is, 0 and 1 respectively) within the context of its parent, which is Ethernet/WAN.

- Index 12 and 13 would both generate the name AmdP2::0.
- Index 14 and 15 would both generate the name AmdP2::1.

To correct this, the algorithm briefly described above is executed as follows:

- Index 12 is processed first and would generate the name "AmdP2::0".
- Index 13 is processed and it is determined that the entPhysicalParentRelPos is a duplicate. The cartridge increments the value by 1 and generates the name "AmdP2::1". It also flags this entity's position as artificially generated.
- Index 14 is processed and it is determined that an entity already exists with entPhysicalParentRelPos 1, however it was artificially generated. Therefore, Index 14 gets the name "AmdP2::1" and Index 13 gets the name "AmdP2::2".
- Index 15 is processed and it is determine that the entPhysicalParentRelPos is a duplicate. The cartridge increments the last artificially generated value by 1 thereby creating name "AmdP2::3".

  Table A-2 shows the result.

Table A-2 End Result

Index	entPhysicalDesr	Name
12	AmdP2	AmdP2::0
13	AmdP2	AmdP2::2
14	AmdP2	AmdP2::1
15	AmdP2	AmdP2::3