Oracle® Communications Network Integrity

Cisco Router and Switch SNMP Cartridge Guide

Release 7.3.2

E66036-01

May 2016

ORACLE®

Oracle Communications Network Integrity Cisco Router and Switch SNMP Cartridge Guide, Release 7.3.2

E66036-01

Contents

5 About Model Correction

6 About Design Studio Construction

Preface

This guide describes the functionality and design of the Oracle Communications Network Integrity Cisco Router and Switch 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.

Document Revision History

The following table lists the revision history for this guide:

Version	Date	Description
E66036-01	May 2016	Initial release.

1

Overview

This chapter provides an overview of the Oracle Communications Network Integrity Cisco Router and Switch SNMP cartridge.

This chapter contains the following sections:

About the Cisco Router and Switch SNMP Cartridge

The Cisco Router and Switch SNMP cartridge provides Cisco-specific functionality including:

- Generic management information base (MIB) MIB-II logical discovery and modeling
- Cisco logical discovery and modeling of Frame Relay, asynchronous transfer mode (ATM), and virtual local area network (VLAN) media configurations
- Cisco 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, subinterfaces (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.

Figure 1–1 shows a sample discovered physical device hierarchy.

Figure 1–1 Sample Physical Device Hierarchy

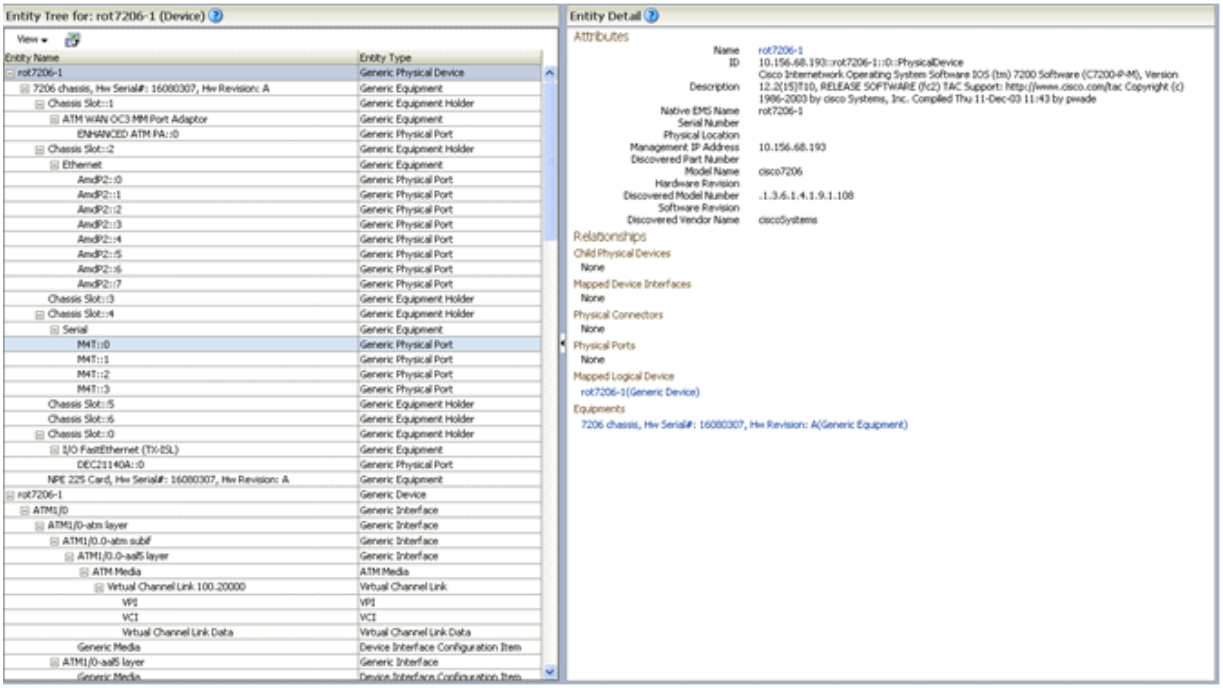


Figure 1–2 shows a sample discovered ATM logical device hierarchy.

Figure 1–2 ATM Logical Device Hierarchy

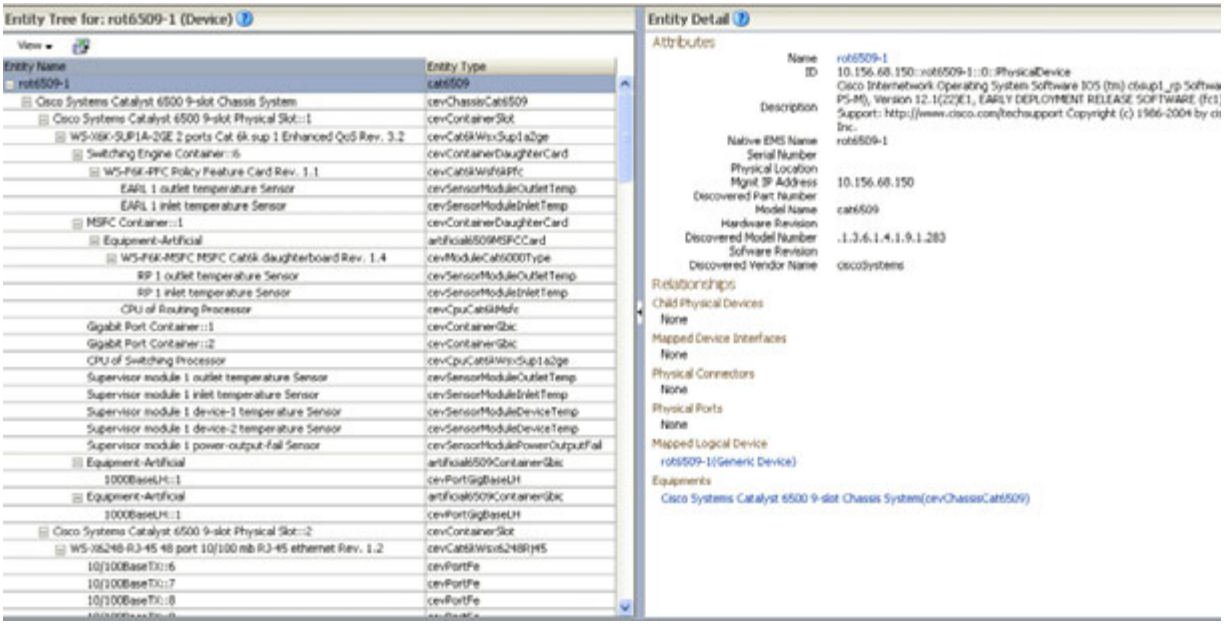


Figure 1–3 shows a sample Frame Relay hierarchy.

Figure 1–3 Frame Relay Logical Device Hierarchy

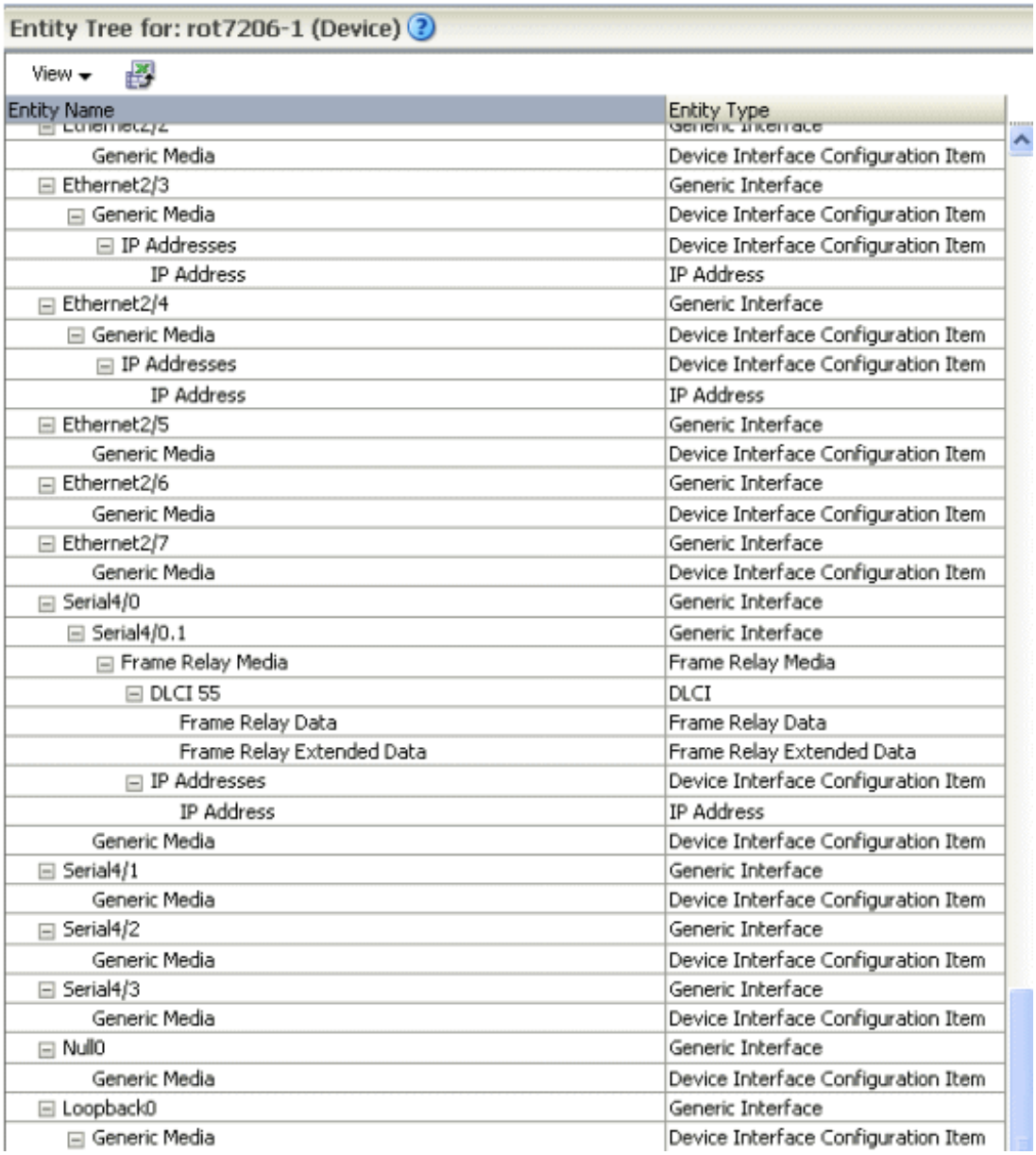
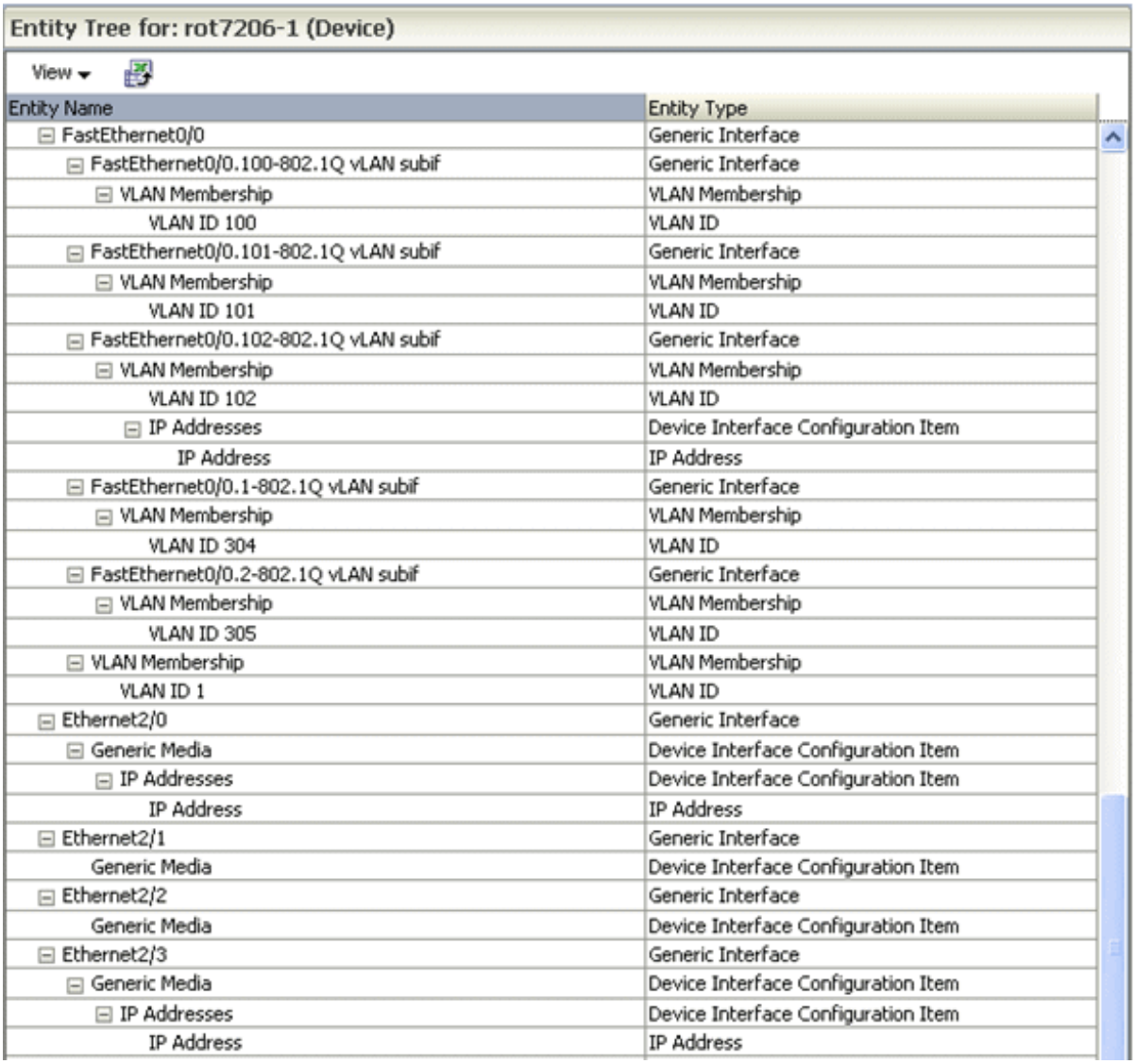


Figure 1–4 shows a sample VLAN hierarchy.

Figure 1–4 VLAN Logical Device Hierarchy



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 Cisco physical discovery, the ENTITY-MIB (RFC 2737) is required.
- For Cisco ATM discovery, the ATM-MIB (RFC 2515) is required.
- For Cisco Frame Relay discovery, the following MIBs are required:
  - CISCO-FRAME-RELAY-MIB
  - RFC1315-MIB

- For Cisco VLAN discovery, the following MIB is required:
  - CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB

The discovery of a specific device fails if the target device does not support MIB-II or the device vendor type is not Cisco. 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. This cartridge is designed to discover Cisco devices only, and attempts to discover non-Cisco devices result in a scan failure.

CPU Utilization-enabled Discovery

The Cisco Router and Switch SNMP cartridge enables you to discover devices based on their CPU utilization by setting a threshold value (in percentage) in the Discover Generic Cisco 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 Cisco 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 *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 Cisco 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 the Network Integrity Help for information about creating a scan.

About Cartridge Dependencies

This section provides information on dependencies that the Cisco Router and Switch 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 Cisco Router and Switch SNMP cartridge has the following dependencies:

- Address_Handlers
- Cisco_Model
- MIB_II_SNMP cartridge
- NetworkIntegritySDK
- ora_ni_uim_device
- ora_uim_model

Opening the Cartridge Files in Design Studio

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

https://edelivery.oracle.com

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

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

See the Design Studio online Help and *Oracle Communications Network Integrity Developer's Guide* for information about opening files in Design Studio.

Building and Deploying the Cartridge

See the Design Studio 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 Cisco Router and Switch SNMP cartridge.

The Cisco Router and Switch SNMP cartridge contains the following actions:

- Discover Generic Cisco SNMP Action

Discover Generic Cisco SNMP Action

The Discover Generic Cisco SNMP action scans Cisco 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 *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 *Network Integrity Developer's Guide*.

The Discover Generic Cisco SNMP action contains the following processors run in the following order:

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

Figure 2–1 illustrates the processor workflow of the Discover Generic Cisco SNMP action.

Figure 2–1 Discover Generic Cisco SNMP Action Processor Workflow

Cisco SNMP Properties Initializer

This processor produces the following data sets:

- ciscoProductsMap
- ciscoVendorNumbers
- ciscoVendorTypesMap

ciscoProductsMap contains a mapping from the vendor-specific portion of the sysObjectId to Cisco device model name.

Example 2–1 provides an example of the ciscoProductsMap file contents.

Example 2–1 ciscoProductsMap File Content Sample

```
10 = cisco2000
     10 - represents the device specific portion of the sysObject (for example,
1.3.6.1.4.1.9.1.10)
     cisco2000 - represents the model name
```

ciscoVendorNumbers contains a list of supported Cisco vendor numbers.

ciscoVendorTypesMap contains a mapping from entPhysicalVendorType to the Cisco equipment part name.

Example 2–2 provides an example of the ciscoVendorTypesMap file contents.

Example 2–2 ciscoVendorTypesMap File Content Sample

```
1.8 = cevOtherCscLink       1.8 - represents the device specific portion of the
sysObjectId (i.e. .1.3.6.1.4.1.9.12.3.1.1.8)       cevOtherCscLink - represents
the equipment part name.
```

Table 2–1 shows a fragment of each data set provided by the Cisco SNMP Properties Initializer:

Table 2–1 Sample Output for Cisco SNMP Properties Initializer

Sample ciscoProductsMap	Sample ciscoVendorNumbers	Sample ciscoVendorTypesMap
1 = ciscoGatewayServer	9	1 = cevOther
2 = ciscoTerminalServer	4857	1.1 = cevOtherUnknownCard
3 = ciscoTrouter	5771	1.2 = cevOtherMoH
4 = ciscoProtocolTranslator	5842	1.3 = cevNmeApaLink
5 = ciscoIGS	7003	1.5 = cevOtherSce8000Scm
6 = cisco3000	N/A	1.6 = cevOtherSce8000Sip
7 = cisco4000	N/A	1.8 = cevOtherCscLink
8 = cisco7000	N/A	2 = cevUnknown
9 = ciscoCS500	N/A	3 = cevChassis
10 = cisco2000	N/A	3.1 = cevChassisUnknown
1294 = ciscoCDScde250	N/A	12.1 = cevMidplaneUmg9820

The content of these property files changes occasionally, and they are maintained as part of cartridge revisions. SDK extensions to this cartridge can update the content of the property files. See "About Design Studio Extension."

Cisco CPU Collector

This processor collects the CPU busy percentage value over the past one minute from the device using the cpmCPUTotal1minRev poll list. See "Cisco CPU Collector Poll List" for more information.

Device CPU Set Processor

This processor is used to set the device's CPU utilization threshold value, which is collected by the Cisco CPU Collector.

A device can have multiple CPUs associated with it. In such situations, the average of the usage percentage of all the CPUs is considered and the average value is compared with the user-specified CPU utilization threshold value.

This processor uses the **cpuProperties** file to set the CPU value of the device in the deviceCPUValue variable. Before extending the CPU utilization functionality, ensure that the device CPU value is set in the deviceCPUValue variable. This value is used as an input for the next processor to compare the CPU utilization value specified by the user and that of the device.

Cisco SNMP Logical Collector

This processor collects Frame Relay, ATM and VLAN media data from the device. See "About Poll Lists."

Cisco SNMP Physical Collector

This processor collects the physical aspects (such as Chassis, Container, Module, Port) of the device. See "About Poll Lists."

Cisco SNMP Logical Modeler

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

Cisco SNMP Physical Modeler

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

3

About Poll Lists

This chapter provides poll lists for processors in the Oracle Communications Network Integrity Cisco Router and Switch SNMP cartridge.

Cisco CPU Collector Poll List

The following list shows the poll lists for the Cisco CPU Collector:

- CISCO-PROCESS-MIB
  - cpmCPUTotal5secRev: The overall CPU busy percentage over the past five seconds. This object deprecates the object cpmCPUTotal5sec and increases the value range to (0..100).
  - cpmCPUTotal1minRev: The overall CPU busy percentage over the past one minute. This object deprecates the object cpmCPUTotal1min and increases the value range to (0..100).
  - cpmCPUTotal5minRev: The overall CPU busy percentage over the past five minutes. This object deprecates the object cpmCPUTotal5min and increases the value range to (0..100).

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

Cisco SNMP Logical Collector Poll List

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

- ATM-MIB
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmVclTable.atmVclEntry.atmVclAdminStatus
  - .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.atmVccAal5EncapsType
  - .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccCrcErrors
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceMaxVpcs
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceMaxVccs
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceMaxActiveVpiBits

  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceMaxActiveVciBits
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceIlmiVpi
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceIlmiVci
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceAddressType
  - .mgmt.mib-2.atmMIB.atmMIBObjects.atmInterfaceConfTable.atmInterfaceConfEntry.atmInterfaceAdminAddress
  - .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccSarTimeOuts
  - .mgmt.mib-2.atmMIB.atmMIBObjects.aal5VccTable.aal5VccEntry.aal5VccOverSizedSDUs
- Cisco-Frame-Relay-MIB
  - .private.enterprises.cisco.ciscoMgmt.ciscoFrameRelayMIB.ciscoFrMIBObjects.cfrCircuitObjs.cfrCircuitTable.cfrCircuitEntry.cfrCircuitType
  - .private.enterprises.cisco.ciscoMgmt.ciscoFrameRelayMIB.ciscoFrMIBObjects.cfrCircuitObjs.cfrExtCircuitTable.cfrExtCircuitEntry.cfrExtCircuitSubifIndex
  - .private.enterprises.cisco.ciscoMgmt.ciscoFrameRelayMIB.ciscoFrMIBObjects.cfrSvcObjs.cfrSvcTable.cfrSvcEntry.cfrSvcThroughputIn
  - .private.enterprises.cisco.ciscoMgmt.ciscoFrameRelayMIB.ciscoFrMIBObjects.cfrSvcObjs.cfrSvcTable.cfrSvcEntry.cfrSvcCommitBurstIn
  - .private.enterprises.cisco.ciscoMgmt.ciscoFrameRelayMIB.ciscoFrMIBObjects.cfrSvcObjs.cfrSvcTable.cfrSvcEntry.cfrSvcExcessBurstIn
- RFC1315-MIB
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitIfIndex
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitDlci
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitState
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitCreationTime
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitLastTimeChange
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitCommittedBurst
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitExcessBurst
  - .mgmt.mib-2.transmission.frame-relay.frCircuitTable.frCircuitEntry.frCircuitThroughput
- VLAN-CISCO-IFTABLE-RELATIONSHIP-MIB
  - private.enterprises.cisco.ciscoMgmt.ciscoVlanIfTableRelationshipMIB.cviMIBObjects.cviGlobals.cviVlanInterfaceIndexTable.cviVlanInterfaceIndexEntry.cviRoutedVlanIfIndex

Cisco SNMP Physical Collector Poll List

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

- Entity-MIB
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalDescr
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalVendorType
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalContainedIn
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalClass
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalParentRelPos
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalFirmwareRev
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalSoftwareRev
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalHardwareRev
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalSerialNum
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalMfgName
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalModelName
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalAlias
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalAssetID
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityPhysical.entPhysicalTable.entPhysicalEntry.entPhysicalIsFRU
  - .mgmt.mib-2.entityMIB.entityMIBObjects.entityMapping.entAliasMappingTable.entAliasMappingEntry.entAliasMappingIdentifier

4

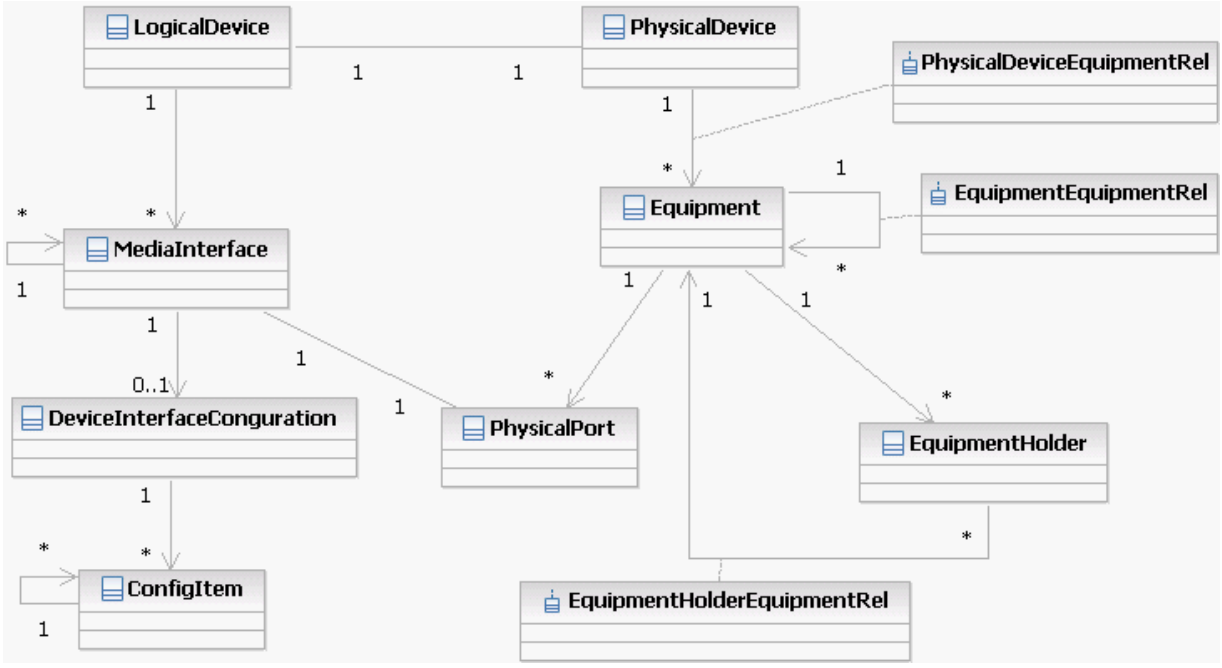
About Cartridge Modeling

This chapter provides information on modeling the Network Integrity Cisco Router and Switch SNMP cartridge.

Cisco Router and Switch SNMP Cartridge UML Representation

Figure 4–1 displays a unified modeling language (UML) diagram depicting the object relationship being rendered.

Figure 4–1 Cisco Router and Switch 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, Frame Relay, or VLAN).

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 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".

The Frame Relay device interface configuration is established and seeded with data sourced from the CISCO-FRAME-RELAY-MIB and RFC1315-MIB. See "About Poll Lists".

The VLAN device interface configuration is established and seeded with data sourced from the VLAN-CISCO-IFTABLE-RELATIONSHIP-MIB. See "About Poll Lists".

Oracle Communications Information Model Information

All entities shown in Figure 4–1 (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 Cisco Router and Switch SNMP cartridge supports the following configurations:

- Generic Media
- AtmMedia
- FrameRelayMedia
- VLAN Membership

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

Specifications

This section lists the specifications included in the UIM Sample Technology Pack for Cisco Devices.

Specification Naming

The specifications with the name pattern cev*XXXX* are Cisco product names derived from the **CISCO_ENTITY_VENDORTYPE-OID-MIB** file.

The specifications with the name pattern artificial*XXXX* are Network Integrity application specific and are used to compensate a hierarchy when the discovered device renders a non Information Model compliant tree. See *Oracle Communications*

Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide for information about model correction.

PhysicalDevice

This specification models a physical device entity.

Table 4–1 shows the specifications for physical Device.

Table 4–1 Physical Device Specifications

Specification	Intended Usage/ Notes
cisco3640	Models a subset of 3640 devices.
cat6509	Models a subset of 6509 devices.
cisco7206VXR	Models a subset of 7206 devices.

Table 4–2 shows the characteristics applied to the physical device specifications. See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* for information about how these fields relate to the Information Model and how they are populated.

Table 4–2 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

See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* for information about these fields.

Equipment

The following list shows Equipment specifications:

- artificial6509ContainerGbic
- artificial6509MSFCCard
- artificial6509PowerSupplyHolderCard
- cevBackplaneCat6000
- cevBackplaneCat6500
- cevCat6kWsc6000cl

- cevCat6kWsc6kvtt
- cevCat6kWsf6kPfc
- cevCat6kWsx6248Rj45
- cevCat6kWsx6348Rj45
- cevCat6kWsxSup1a2ge
- cevChassis3640
- cevChassis7206Vxr
- cevChassisCat6509
- cevCpu7200Npe300
- cevCpuCat6kMsfc
- cevCpuCat6kWsxSup1a2ge
- cevFanWSC6k9SlotFan
- cevModuleCat6000Type
- cevModuleUnknownCard
- cevModuleVipPortAdapters
- cevPa8e
- cevPaAtmdxMmOc3
- cevPaE3MuxCbr120e1
- cevPaE3MuxCbr120e1
- cevPmCpm2e2w
- cevPmM4t
- cevPowerSupplyAC1360W
- cevPowerSupplyC7200AC
- cevSensorClock
- cevSensorFanTrayStatus
- cevSensorModuleDeviceTemp
- cevSensorModuleOutletTemp
- cevSensorModulePowerOutputFail
- cevSensorPSFan
- cevSensorPSInput
- cevSensorPSOutput
- cevSensorVtt
- cevWicSerial1t

Table 4–3 shows the characteristics applied to the Equipment specifications. See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* for information about how these field relate to the Information Model and are populated.

Table 4–3 Equipment 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

See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* to understand how these fields are populated.

Equipment Holder

The following list shows Equipment Holder specifications:

- artificial6509PowerSupplySlot
- cevContainerClock
- cevContainerDaughterCard
- cevContainerFanTraySlot
- cevContainerGbic
- cevContainerPowerSupplyBay
- cevContainerSlot
- cevContainerVtt

Table 4–4 shows the characteristics applied to the EquipmentHolder specifications. See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* for information about how these fields relate to the Information Model and are populated.

Table 4–4 EquipmentHolder 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

See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* to understand how these fields are populated.

Physical Port

The following list shows Physical Port specifications:

- cevPortAMDP2
- cevPortDCUATMPort
- cevPortFe
- cevPortFEIP
- cevPortGigBaseLH
- cevPortMueslix
- cevPortQuiccSerial
- cevPortUnknown

Table 4–5 shows the characteristics applied to the PhysicalPort specifications. See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* for information about how these fields relate to the Information Model and are populated.

Table 4–5 PhysicalPort 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

See *Network Integrity Cisco Router and Switch UIM Integration Cartridge Guide* to understand how these fields are populated.

Specification Lineages

The UIM Sample Technology Pack for Cisco Devices contains the following specification lineages:

- cisco3640
- cisco7206VXR
- cat6509

cisco3640

```
cisco3640
    cevChassis3640
          cevContainerSlot
                cevPmCpm2e2w
                      cevPortAMDP2
                      cevContainerDaughterCard
                            cevWicSerial1t
                                  cevPortQuiccSerial
                cevPmM4t
                      cevPortMueslix
```

cisco7206VXR

```
cisco7206VXR
    cevChassis7206Vxr
          cevCpu7200Npe300
          cevContainerSlot
                cevModuleUnknownCard
                      cevPortFEIP
                cevPaAtmdxMmOc3
                      cevPortUnknown
                cevModuleVipPortAdapters
                      cevPortMueslix
                cevPaE3MuxCbr120e1
                      cevPortUnknown
                      cevPortDCUATMPort
                cevPowerSupplyC7200AC
                cevPa8e
                      cevPortAMDP2
```

cat6509

```
cat6509
    cevChassisCat6509
          cevBackplaneCat6000
                cevContainerVtt
                      cevCat6kWsc6kvtt
                            cevSensorVtt
                cevSensorVtt
                cevContainerClock
                      cevCat6kWsc6000cl
                            cevSensorClock
                cevSensorClock
          cevBackplaneCat6500
                cevContainerVtt
                      cevCat6kWsc6kvtt
                            cevSensorVtt
                cevSensorVtt
                cevContainerClock
                      cevCat6kWsc6000cl
                            cevSensorClock
                cevSensorClock

          cevContainerFanTraySlot
                cevFanWSC6k9SlotFan
                      cevSensorFanTrayStatus
          cevContainerPowerSupplyBay
                artificial6509PowerSupplyHolderCard
```

```
                  artificial6509PowerSupplySlot
                        cevPowerSupplyAC1360W
                              cevSensorPSFan
                              cevSensorPSInput
                              cevSensorPSOutput
cevContainerSlot
      cevCat6kWsxSup1a2ge
            cevCpuCat6kWsxSup1a2ge
            cevSensorModuleOutletTemp
            cevSensorModulePowerOutputFail
            cevSensorModuleInletTemp
            cevSensorModuleDeviceTemp
            cevContainerDaughterCard
                  cevWicSerial1t
                        cevPortQuiccSerial
                  cevCat6kWsf6kPfc
                        cevSensorModuleOutletTemp
                        cevSensorModuleInletTemp
                  artificial6509MSFCCard
                        cevModuleCat6000Type
                              cevSensorModuleOutletTemp
                              cevSensorModuleInletTemp
                        cevCpuCat6kMsfc
            artificial6509ContainerGbic
                  cevPortGigBaseLH
      cevCat6kWsx6248Rj45
            cevSensorModulePowerOutputFail
            cevSensorModuleInletTemp
            cevSensorModuleDeviceTemp
            cevPortFe
      cevCat6kWsx6348Rj45
            cevSensorModulePowerOutputFail
            cevSensorModuleInletTemp
            cevSensorModuleDeviceTemp
            cevPortFe
```

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.

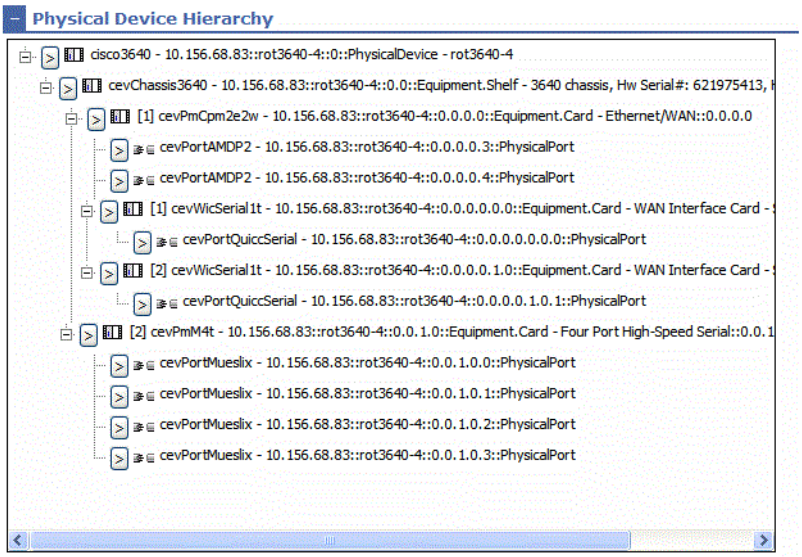
Equipment Visual Specifications

The visual facility on the Equipment specifications is not used. It is left to the customer to decide if they want to enrich the technology pack to provide visual effects to the UIM GUI for a given Equipment entity.

Sample Physical Device Hierarchy

Figure 4–2 shows a physical device hierarchy.

Figure 4–2 Sample Physical Device Hierarchy



Field Mapping

The Cisco Router and Switch 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 getter and setters but generically has get and set characteristics method holding a HashSet of entries.

Logical Mapping

The Cisco Router and Switch SNMP cartridge supports the following logical mappings:

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

LogicalDevice

Table 4–6 shows characteristics for the LogicalDevice specification.

Table 4–6 LogicalDevice Characteristics

Characteristics (LogicalDevice)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
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
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 4–7 shows characteristics for the MediaInterface specification.

Table 4–7 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

Table 4–7 (Cont.) MediaInterface Characteristics

Characteristics (MediaInterface)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
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 4–17.
nativeEmsServiceState	static	ifOperStatus	Enum with the following values: ■ UNKNOWN ■ IN_SERVICE ■ OUT_OF_ SERVICE ■ TESTING ■ IN_ MAINTENANCE	Mapped, see Table 4–17.
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

Table 4–7 (Cont.) MediaInterface Characteristics

Characteristics (MediaInterface)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
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 4–8 shows characteristics for the DeviceInterfaceConfigurationItem (IPv4) specification.

Table 4–8 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 4–9 shows characteristics for the DeviceInterfaceConfigurationItem (IPv6) specification.

Table 4–9 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 (Frame Relay)

Table 4–10 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (FrameRelayMedia) specification.

Table 4–10 Characteristics for the DeviceInterfaceConfigurationItem (FrameRelayMedia) Specification

Characteristics (FrameRelayMedia)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
DLCI	dynamic	Table index	Text	Parsed from the table index, which is in the format: <ifindex>.<DLCI>.

Table 4–11 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (FrameRelayData) specification.

Table 4–11 Characteristics for the DeviceInterfaceConfigurationItem (FrameRelayData) Specification

Characteristics (FrameRelayData)	Information Model Support	MIB Object	Field Type
frCircuitIfIndex	dynamic	frCircuitIfIndex	Text
frCircuitDlci	dynamic	frCircuitDlci	Text
frCircuitState	dynamic	frCircuitState	Text
frCircuitCreationTime	dynamic	frCircuitCreationTime	Text
frCircuitLastTimeChange	dynamic	frCircuitLastTimeChange	Text
frCircuitCommittedBurst	dynamic	frCircuitCommittedBurst	Text
frCircuitExcessBurst	dynamic	frCircuitExcessBurst	Text
frCircuitThroughput	dynamic	frCircuitThroughput	Text

Table 4–12 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (FrameRelayExtendedData) specification.

Table 4–12 Characteristics for the DeviceInterfaceConfigurationItem (FrameRelayExtendedData) Specification

Characteristics (FrameRelayExtendedData)	Information Model Support	MIB Object	Field Type
cfrCircuitType	dynamic	cfrCircuitType	Text
cfrExtCircuitSubIfIndex	dynamic	cfrExtCircuitSubifIndex	Text
cfrSvcThroughputIn	dynamic	cfrSvcThroughputIn	Text
cfrSvcCommitBurstIn	dynamic	cfrSvcCommitBurstIn	Text
cfrSvcExcessBurstIn	dynamic	cfrSvcExcessBurstIn	Text

DeviceInterfaceConfigurationItem Mapping (ATM Media)

Table 4–13 shows characteristics for the DeviceInterfaceConfigurationItem mapping (ATM media) specification.

Table 4–13 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 4–14 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (VirtualChannelLinkData) specification.

Table 4–14 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 4–15 shows characteristics for the DeviceInterfaceConfigurationItem Mapping (AtmInterfaceConfigurationData) specification.

Table 4–15 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

DeviceInterfaceConfigurationItem Mapping (VLAN)

Table 4–16 shows characteristics for the DeviceInterfaceConfigurationItem mapping (VLAN) specification.

Table 4–16 Characteristics for the DeviceInterfaceConfigurationItem (VLAN) Specification

Characteristics (VLAN ID)	Information Model Support	MIB Object	Field Type
cviRoutedVlanIfIndex	dynamic	cviRoutedVlanIfIndex	Text

Mapping Table

Table 4–17 shows a mapping table.

Table 4–17 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:

- Mapping From Cisco to Information Model Nomenclature
- PhysicalDevice
- Equipment
- EquipmentHolder
- PhysicalPort

Mapping From Cisco to Information Model Nomenclature

Table 4–18 lists the Cisco to Information Model mapping 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 4–18 Cisco to Information Model Physical 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 4–19 shows the characteristics for the PhysicalDevice specification.

Table 4–19 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	N/A	Programmatically applies physicalDeviceGeneric specification.
discoveredModelNumber	dynamic	sysObjectId	Text	N/A
discoveredVendorName	dynamic	N/A	Text	Mapped using the snmpVendorNameMap which is created by the MIB II Properties Initializer. See "Determining the Value of the discoveredVendorName Field".
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	Mapped using the ciscoProductsMap, which is created by the Cisco SNMP Properties Initializer. See "Determining the Value of the modelName Field".
mgmtIpAddress	dynamic	N/A	Text	discoveryAddress
nativeEmsName	static	sysName	Text	N/A

Equipment

Table 4–20 shows characteristics for the Equipment specification.

Table 4–20 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	N/A	Programmatically applies equipmentGeneric specification.
discoveredModelNumber	dynamic	entPhysicalVendorType	Text	N/A

Table 4–20 (Cont.) Equipment Characteristics

Characteristics (Equipment)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
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	Mapped using the ciscoVendorTypesMap, which is created by the Cisco SNMP Properties Initializer. See "Determining the Value of the modelName Field".
nativeEmsName	static	entPhysicalDescr and absRelativePosition	Text	N/A

EquipmentHolder

Table 4–21 shows characteristics for the EquipmentHolder specification.

Table 4–21 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".
Name	static	entPhysicalDescr and entPhysicalParentRelPos	Text	Programmatically generated as entPhysicalDescr::entPhysicalParentRelPos.
Description	static	entPhysicalDescr	Text	N/A
Specification	static	N/A	N/A	Programmatically applies equipmentHolderGeneric specification.
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

Table 4–21 (Cont.) EquipmentHolder Characteristics

Characteristics (EquipmentHolder)	Information Model Support	MIB Object	Field Type	Intended Usage/ Notes
softwareRev	dynamic	entPhysicalSoftwareRev	Text	N/A
modelName	dynamic		Text	Mapped using the ciscoVendorTypesMap, which is created by the Cisco SNMP Properties Initializer. See "Determining the Value of the modelName Field".
nativeEmsName	static	entPhysicalParentRelPos	Text	N/A

PhysicalPort

Table 4–22 shows characteristics for the PhysicalPort specification.

Table 4–22 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	N/A	Programmatically applies equipmentGeneric specification.
discoveredModelNumber	dynamic	entPhysicalVendorType	Text	N/A
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	N/A	Text	Mapped using the ciscoVendorTypesMap, which is created by the Cisco SNMP Properties Initializer. See "Determining the Value of the modelName Field".
nativeEmsName	static	entPhysicalParentRelPos	Text	N/A

5

About Model Correction

This chapter provides SNMP to Oracle Communications Information Model correction information for the Oracle Communications Network Integrity Cisco Router and Switch SNMP cartridge.

About Model Correction

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

The Cisco Router and Switch 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 Cisco Router and Switch SNMP cartridge applies the algorithm detailed in "About the absRelativePosition Value". No hierarchical change is required.

Equipment Under Physical Port

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

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

the Cisco Router and Switch 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 Cisco device hierarchy, such as the following:

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

the Cisco Router and Switch 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 Cisco device hierarchy, such as the following:

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

the Cisco Router and Switch 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 Cisco device hierarchy, such as the following:

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

the Cisco Router and Switch SNMP cartridge inserts an artificial Equipment entity as the parent to the two equipment as follows:

```
Physical Device
```

```
Equipment-Shelf
      EquipmentHolder
                ***Equipment***
                      Equipment
                      Equipment
```

EquipmentHolder Under Physical Port

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

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

the Cisco Router and Switch 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 Cisco device hierarchy, such as the following:

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

the Cisco Router and Switch SNMP cartridge first converts the Physical Port to an EquipmentHolder, then inserts an artificial Equipment between the EquipmentHolder and the Equipment as follows:

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

6

About Design Studio Construction

This chapter provides information on the composition of the Oracle Communications Network Integrity Cisco Router and Switch SNMP cartridge from the Oracle Communications Design Studio perspective.

Model Collections

Table 6–1 shows the MIB-II model collection used in the Cisco Router and Switch SNMP cartridge.

Table 6–1 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 6–2 shows the Cisco model collection.

Table 6–2 Cisco Model Collection

Specifications	Information Model Entity Type	Intended Usage/ Notes
physicalDeviceGeneric	PhysicalDevice	Represents any Cisco Physical Device discovered on the network.
equipmentHolderGeneric	EquipmentHolder	Represents any equipment container.
equipmentGeneric	Equipment	Represents any piece of equipment (for example, Chassis, Backplane, Module).
physicalPortGeneric	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.

Table 6–2 (Cont.) Cisco Model Collection

Specifications	Information Model Entity Type	Intended Usage/ Notes
VirtualChannelLinkData	DeviceInterfaceConfigurationItem	Contains the Virtual Channel Link characteristics and is contained with the Virtual Channel Link.
AtmInterfaceConfigurationData	DeviceInterfaceConfigurationItem	Contains the ATM configuration applied to an interface.
FrameRelayMedia	DeviceInterfaceConfigurationItem	Represents a Frame Relay media configuration that can be applied to an interface or sub-interface.
DLCI	DeviceInterfaceConfigurationItem	Represents a single DLCI and is contained within a Frame Relay media configuration.
FrameRelayData	DeviceInterfaceConfigurationItem	Contains the Frame Relay characteristics.
FrameRelayExtendedData	DeviceInterfaceConfigurationItem	Contains additional Frame Relay characteristics.
VlanMembership	DeviceInterfaceConfigurationItem	Represent the VLAN Membership media configuration that can be applied to an interface or sub-interface.
Vlan	DeviceInterfaceConfigurationItem	Represents a single VLAN and is contained within a VLAN Membership.

Logical Specification Lineage

Example 6–1 shows a logical specification lineage. This lineage shows the intended relationship between specifications.

Example 6–1 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
           [0..1] FrameRelayMedia
                 [0..1] IP Addresses
                       [0..*] IpAddress
                             IpAddress (characteristic)
                             Prefix (characteristic)
                             IpVersion (characteristic)
                 [0..*] DLCI
                       [0..1] FrameRelayData
                       [0..1] FrameRelatExtendedData
           [0..1] VlanMembership
```

```
[0..1] IP Addresses
      [0..*] IpAddress
             IpAddress (characteristic)
             Prefix (characteristic)
             IpVersion (characteristic)
[0..*] Vlan
```

Physical Specification Lineage

This section provides an example of a physical specification lineage.

Example 6–2 shows a physical specification lineage.

Example 6–2 Physical Specification Lineage

```
physicalDeviceGeneric
     [0..*] equipmentGeneric
           [0..*] equipmentHolderGeneric
                 [0..1] equipmentGeneric
           [0..*] physicalPortGeneric
           [0..*] equipmentGeneric
```

Discovery Action

Table 6–3 shows the discovery action in the Cisco Router and Switch SNMP cartridge.

Table 6–3 Discover Generic Cisco 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.	MIB-II Model and Cisco Model	▪ MIB II Properties Initializer ▪ Cisco SNMP Properties Initializer ▪ MIB II SNMP Collector ▪ MIB II SNMP Modeler ▪ Cisco SNMP Logical Collector ▪ Cisco SNMP Physical Collector ▪ Cisco SNMP Logical Modeler ▪ Cisco SNMP Physical Modeler

Figure 6–1 displays the Discover Generic Cisco SNMP action chain.

Figure 6–1 Discover Generic Cisco SNMP Action Flow

Discovery Processors

Table 6–4 shows the processors of the Discover Generic Cisco SNMP action.

Table 6–4 Discover Generic Cisco SNMP Action Processors

Processor Name	Variable
MIB-II Properties Initializer	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.
Cisco SNMP Properties Initializer	Input: N/A Output: ■ ciscoProductMap A mapping from the vendor portion of the sysObjectId to the Cisco Device model name. ■ ciscoVendorNumber A mapping from the vendor specific portion of the sysObjectId to Cisco Device model name. ■ ciscoVendorTypesMap A mapping from entPhysicalVendorType to Cisco equipment part name.
CPU Property Initializer	Input: N/A Output: ■ cpuProperties This class is used to initialize and set the default CPU value for the device.
Cisco CPU Collector	Input: N/A Output: ■ cpmCPUTotal5secRev ■ cpmCPUTotal1minRev ■ cpmCPUTotal5minRev
Device CPU Set Processor	Input: N/A ■ ciscoCPUCollectorResponseDocument SNMP discovered data produced by the Cisco CPU Collector. ■ cpuProperties This class is used to initialize and set the default CPU value for the device. Output: N/A
CPU Utilization Compare Processor	Input: N/A ■ cpuProperties This class is used to initialize and set the default CPU value for the device. Output: N/A
MIB-II SNMP Collector	Input: N/A Output: ■ mibiisnmpCollectorResponseDocument (implicit) Polled SNMP data, see "About Poll Lists".

Table 6–4 (Cont.) Discover Generic Cisco SNMP Action Processors

Processor Name	Variable
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.
Cisco SNMP Logical Collector	Input: N/A Output: ■ ciscoSNMPLogicalCollectorResponseDocument (implicit) Polled SNMP data, see "About Poll Lists".

Table 6–4 (Cont.) Discover Generic Cisco SNMP Action Processors

Processor Name	Variable
Cisco SNMP Physical Collector	Input: N/A Output: ■ ciscoSNMPPhysicalCollectorResponseDocument (implicit) Polled SNMP data, see "About Poll Lists".
Cisco SNMP Logical Modeler	Input: ■ ciscoSNMPLogicalCollectorResponseDocument SNMP discovered data produced by the Cisco SNMP Logical Collector. ■ ciscoVendorNumbers The vendor numbers that are considered Cisco. Only Cisco devices are discovered. ■ 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: ■ ciscoLogicalDevice This is the logical device that was created in the Cisco SNMP Logical Modeler.
Cisco SNMP Physical Modeler	Input: ■ ciscoSNMPPhysicalCollectorResponseDocument SNMP discovered data produced by the Cisco SNMP Physical Collector. ■ ciscoProductsMap A mapping of sysObjectId (vendor-specific portion) to Cisco device model names/numbers. ■ ciscoVendorTypesMap A mapping of entPhysicalVendorType OID to equipment part names/numbers. ■ 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. ■ snmpVendorNameMap Property map containing listing of sysObjectId suffixes to vendorName Output: ■ physicalDevice This is the physical device that is created in this Cisco SNMP Physical Modeler.

7

About Design Studio Extension

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

Adding a New Vendor

For this example, Cisco introduces a new vendor type to represent a new equipment part. Currently, this cartridge defines a map called ciscoVendorTypesMap that contains the equipment part name indexed by the vendor type number, which is a portion of the entPhysicalVendorType OID. The Cisco SNMP Properties Initializer produces this map and makes it available for other processors. To update the map to include a new vendor type number and corresponding equipment part name, you can extend the Discover Generic Cisco SNMP action and add a new Cisco SNMP Post Properties Initializer processor. This initializer takes as input the map (for example, ciscoVendorTypesMap) produced by the Cisco SNMP Properties Initializer. The implementation can then update the map.

For more details regarding extensibility, see *Network Integrity Developer's Guide*.

A

Characteristic Attributes

This appendix describes how to determine the values of the modelName and discoveredVendorName fields.

Determining the Value of the modelName Field

You determine the value of the modelName field for PhysicalDevice using the following algorithm:

- The input is the sysObjectId value in its raw form (for example, .1.3.6.1.4.1.9.1.110) stored in the discoveredModelNumber field.
- It then parses the ninth digit from sysObjectId (for example, 110).
- It then uses the parsed value (for example, 110) as the key to look up the model name from the ciscoProductsMap that is output from the Cisco SNMP Properties Initializer.
- If the key does not exist, discoveredVendorName is set to **Unknown** *(x)* where *x* is the key; for example, **Unknown (110)**.

You determine the value of the modelName field for Equipment, EquipmentHolder, and PhysicalPort using the following algorithm:

- The input is the entPhysicalVendorType OID value in its raw form (for example, .1.3.6.1.4.1.9.12.3.1.9.3.2) stored in the discoveredModelNumber field. This value is in the form of .1.3.6.1.4.1.9.12.3.1.*vvvv.xxxx.yyyy.n.n*….. where *vvvv.xxxx.yyyy* is the vendor type number.
- It then parses out the vendor type number from the entPhysicalVendorType value (for example, 9.3.2).
- It then uses the vendor type number (for example, 9.3.2) as the key to look up the model name from the ciscoVendorTypesMap that is output from the Cisco SNMP Properties Initializer.
- If the key does not exist, the cartridge trims the last digit and try again (for example, from 9.3.2 to 9.3). If the key still does not exist, it trims again (for example, to 9) and try again. If this fails as well, discoveredVendorName is set to **Unknown** *(x)* where *x* is the original key; for example, 9.3.2.

Determining the Value of the discoveredVendorName Field

You determine the value of the discoveredVendorName field using the following algorithm:

- The input is the sysObjectId value in its raw form (for example, .1.3.6.1.4.1.9.1.110) stored in the discoveredModelNumber field.
- It then parses the seventh digit from sysObjectId (for example, 9).
- It then uses the vendor number (for example, 9) as the key to look up the vendor name from the snmpVendorNameMap that is output from the MIB II Properties Initializer.
- If the key does not exist, discoveredVendorName is set to **Unknown** *(x)* where *x* is the key; for example, **Unknown (9)**.

B

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 B–1 shows the applicable SNMP attributes that are used in determining the correct relative position.

Table B–1 SNMP Attributes Used to Determine Correct Relative Position

Index	entPhysicalDesr	entPhysicalParentRelPos	entPhysicalContainedIn
1	3640 chassis, Hw Serial#: 621974280, Hw Revision: 00	-1	0

Table B–1 (Cont.) SNMP Attributes Used to Determine Correct Relative Position

Index	entPhysicalDesr	entPhysicalParentRelPos	entPhysicalContainedIn
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 B–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 B–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 B–2 shows the result.

Table B–2 End Result

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