# Oracle® Communications Network Integrity

MIB-II SNMP Cartridge Guide

Release 7.3.2

**E66042-01**

May 2016

ORACLE®

Oracle Communications Network Integrity MIB-II SNMP Cartridge Guide, Release 7.3.2

E66042-01

# Contents

## 5 About Design Studio Construction



## 6 About Design Studio Extensions

# Preface

This guide describes the functionality and design of the Oracle Communications Network Integrity Management Information Base (MIB) II SNMP cartridge.

## Audience

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

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

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

- *Network Integrity Concepts*
- *Network Integrity Developer's 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 |
|---|---|---|
| E66042-01 | May 2016 | Initial release. |

# 1

# Overview

This chapter provides an overview of the Oracle Communications Network Integrity MIB-II SNMP cartridge.

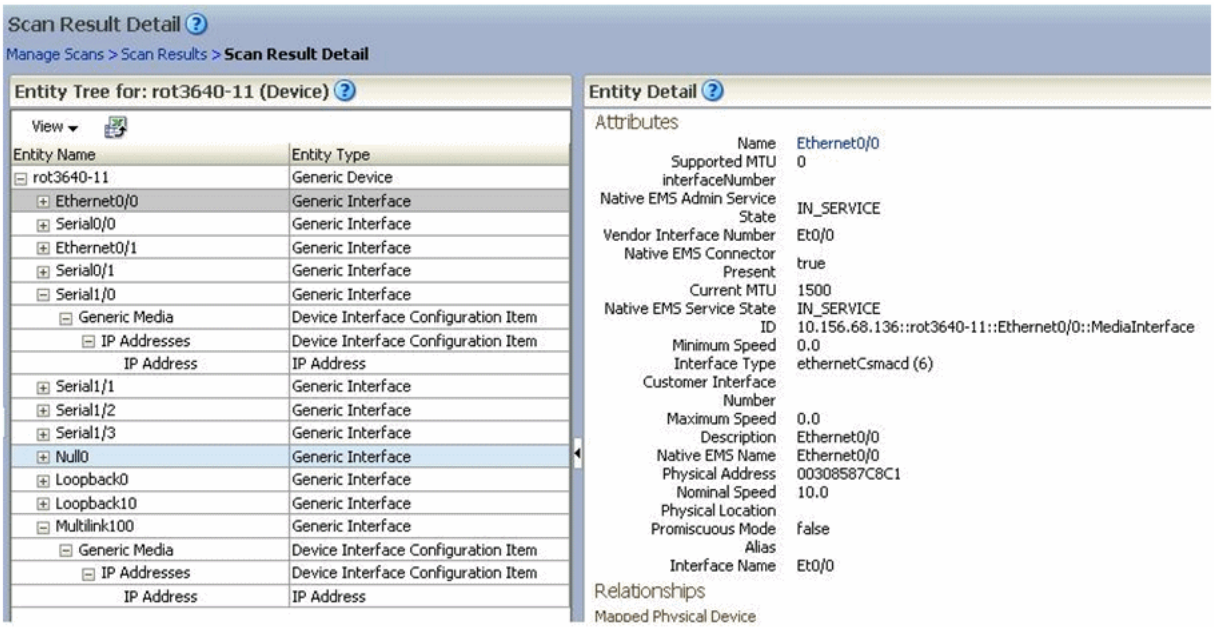
## About the MIB-II SNMP Cartridge

The MIB-II SNMP cartridge enables you to discover devices in your network using the SNMP protocol.

The cartridge produces a logical device hierarchy that represents the discovered device and includes a logical device instance and child interface(s) or subInterface instance(s) (collectively called interfaces).

Figure 1–1 shows a discovered logical device.

***Figure 1–1 Sample Discovered Logical Device***




## About Cartridge Dependencies

This section provides information on dependencies that the MIB-II SNMP cartridge has on other cartridges.

### Run-time Dependencies

For the MIB-II SNMP cartridge to work at run time, the Address_Handlers cartridge must be deployed to Network Integrity.

### Design Studio Dependencies

The MIB-II SNMP cartridge has the following dependencies:

- Address_Handlers
- NetworkIntegritySDK
- ora_ni_uim_device
- ora_uim_model

## Opening the Cartridge Files in Design Studio

To review and extend the Network Integrity MIB-II SNMP cartridge, you must first download the Oracle Communications Network Integrity MIB-II SNMP Cartridge software from the Oracle software delivery website:

https://edelivery.oracle.com

The software contains the MIB-II SNMP cartridge ZIP file, which has the following structure:

- **\UIM_Cartridge_Projects**
- **\Network_Integrity_Cartridge_Projects**
- **\SNMP_MIBs**
- **Address_Handlers-R7.jar**
- **MIB_II_SNMP_Cartridge-R7.jar**

The **\UIM_Cartridge_Projects\ora_ni_uim_device** project contains the model project used by the MIB-II SNMP cartridge.

The **\Network_Integrity_Cartridge_Projects\MIB_II_SNMP_Cartridge** project contains the extensible Design Studio files.

---

**Note:** When importing MIB files into Design Studio, ensure that the MIB files do not have any filename extension. For example, rename the **FileMIB1.txt** file to **FileMIB1** before importing it into Design Studio.

---

See the Design Studio online Help and *Network Integrity Developer's Guide* for information about opening files in Design Studio. See *Network Integrity Concepts* for guidelines and best practices for extending cartridges.

## Building and Deploying the Cartridge

The MIB-II SNMP cartridge does not include any extendable cartridges.

See 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 MIB-II SNMP cartridge.

The MIB-II SNMP cartridge contains the following actions:

- Discover MIB-II SNMP Action

## Discover MIB-II SNMP Action

The Discover MIB-II SNMP action discovers a device and creates its hierarchical model.

The MIB-II SNMP cartridge is designed to be used on a standalone basis to display the logical device hierarchy in Network Integrity. The SNMP cartridge provides no integration with other products but can be extended.

The Discover MIB-II SNMP action contains the following processors run in the following order:

1. MIB-II Properties Initializer Processor
2. MIB-II SNMP Collector Processor
3. MIB-II SNMP Modeler Processor

Figure 2–1 illustrates the processor workflow of the Discover MIB-II SNMP action.

*Figure 2–1 Discover MIB-II SNMP Action Processor Workflow*


## MIB-II Properties Initializer Processor

The MIB-II Properties Initializer processor produces the following data sets:

- `snmpVendorNameMap`: Contains a snapshot of industry enterprise numbers to help identify devices in the network.
- `snmpIfTypeMap`: Contains a snapshot of ifTypes to help identify interface types in the network.

Table 2–1 shows a fragment of each data set output from the MIB-II Properties Initializer.

*Table 2–1 MIB-II Properties Initializer Fragment*

| Sample snmpIfTypeMap | Sample snmpVendorNameMap |
|---|---|
| 1: other (1) | 0 = Reserved |
| 2: regular1822 (2) | 1 = NxNetworks |
| 3: hdh1822 (3) | 2 = IBM |
| 4: ddnX25 (4) | 3 = Carnegie Mellon |
| 5: rfc877x25 (5) | 4 = UNIX |
| 6: ethernetCsmacd (6) | 5 = ACC |
| 7: iso88023Csmacd (7) | 6 = TWG |
| 8: iso88024TokenBus (8) | 7 = CAYMAN |
| 9: iso88025TokenRing (9) | 8 = PSI |
| 10: iso88026Man (10) | 9 = ciscoSystems |
| 251: vdsl2 (251) | 34730 = FRANCILIENNE D'INGENIERIE ET DE SERVICES INFORMATIQUES SAS |

The content of these files may change from time to time, so 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 Extensions" for more information.

## MIB-II SNMP Collector Processor

The MIB-II SNMP Collector processor collects SNMP variables from a device. See "About Poll Lists" for more information.

## MIB-II SNMP Modeler Processor

The MIB-II SNMP Modeler processor models the data that is collected by the MIB-II SNMP Collector processor. Modeling includes building the hierarchical relationship of logical device and child interfaces.

# 3

# About Poll Lists

This chapter provides a poll list for the MIB-II SNMP Collector processor that collects SNMP variables from a device.

## About the MIB-II SNMP Collector Poll List

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

- MIB-II:
  - RFC1213-MIB.mgmt.mib-2.system.sysObjectID
  - RFC1213-MIB.mgmt.mib-2.system.sysDescr
  - RFC1213-MIB.mgmt.mib-2.system.sysName
  - RFC1213-MIB.mgmt.mib-2.system.sysLocation
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifNumber
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifIndex
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifSpeed
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifPhysAddress
  - RFC1213-MIB.mgmt.mib-2.interfaces.ifTable.ifEntry.ifLastChange
  - RFC1213-MIB.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntAddr
  - RFC1213-MIB.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex
  - RFC1213-MIB.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask
  - RFC1213-MIB.mgmt.mib-2.ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr

## About the IF MIB Poll List

The following list shows the IF MIB poll list:

- IF MIB:
  - IF-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName
  - IF-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHighSpeed

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

## About the IP MIB Poll List

The following list shows the IP MIB poll list:

- IP MIB:
  - IP-MIB.mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressPrefix
  - IP-MIB.mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressType
  - IP-MIB.mgmt.mib-2.ip.ipAddressTable.ipAddressEntry.ipAddressIfIndex

# 4

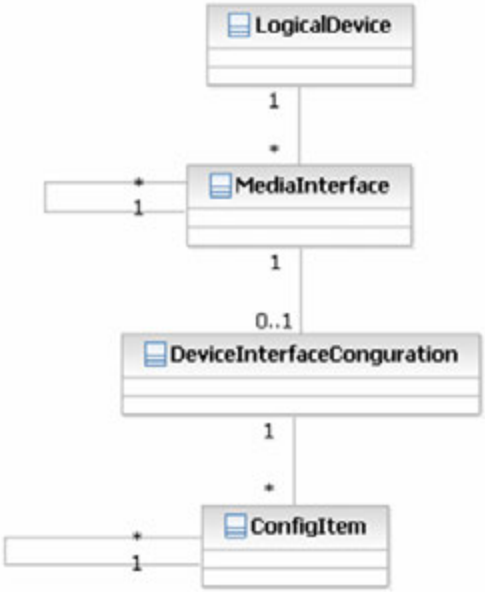
# About Cartridge Modeling

This chapter provides information on modeling the Oracle Communications Network Integrity MIB-II SNMP cartridge.

## MIB-II SNMP Cartridge UML Representation

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

***Figure 4–1 MIB-II SNMP Cartridge UML Representation***




## Hierarchy Mapping

The logical device object is established and seeded with data sourced by RFC1213-MIB.mgmt.mib-2.system tables.

As a device is discovered, the media interface encapsulates common information about that interface. The device interface configuration captures the media type information that decorates the interface with media-specific parameters that define its behavior (that is; Generic, ATM, Frame Relay or VLAN). This cartridge supports only Generic.

Interfaces are established and seeded with data sourced by:

- 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

The interface hierarchy is established by:

- F-MIB.mgmt.mib-2.ifMIB.ifMIBObjects.ifStackTable.ifStackEntry.ifStackStatus

Device interface configuration is established and seeded with data sourced by:

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

# About Oracle Communications Information Model

The logical device and media interface objects are Oracle Communications Information Model 1.0 compliant for static fields. The dynamic fields (sometimes referred to as characteristics) are considered application-specific.

The device interface configuration is considered a mechanism to customize the media interface to show application specific data. See Figure 4–1, "MIB-II SNMP Cartridge UML Representation".

This cartridge supports only generic media.

Other cartridges support other configurations:

- AtmMedia
- FrameRelayMedia
- VLAN Membership

For a listing of the Information Model fields, see "Field Mappings".

# Specifications

This section lists the specifications included in the Network Integrity MIB-II SNMP cartridge.

## Logical Device: deviceGeneric

This specification models a generic Logical Device entity.

This specification is used to model a logical device discovered by Network Integrity.

Table 4–1 shows the characteristics applied to the generic device specification.

*Table 4–1 Characteristics Applied to the deviceGeneric Specification*

| Characteristics | Field Type | Field Content | Intended Usage/ Notes |
|---|---|---|---|
| mgmtIpAddress | String | Text | Management IP address of discovered device |
| nativeEmsAdminServiceState | Drop Down box | Enum with the following values:<br>■ UNKNOWN<br>■ IN_SERVICE<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_MAINTENANCE | Device administered state |
| nativeEmsServiceState | Drop Down box | Enum with the following values:<br>■ UNKNOWN<br>■ IN_SERVICE<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_MAINTENANCE | Device actual state |
| nativeEmsName | String | Text | Device actual name |
| sysObjectId | String | Text | The vendor's authoritative identification of the network management subsystem contained in the entity. |

## Device Interface: interfaceGeneric

This specification models a generic interface entity.

This specification is used to model an interface discovered by Network Integrity.

Table 4–2 shows the characteristics applied to the interfaceGeneric specification.

*Table 4–2 Characteristics Applied to the interfaceGeneric Specification*

| Characteristics | Field Type | Field Content | Intended Usage/ Notes |
|---|---|---|---|
| nativeEmsName | String | Text | Alternate name this interface could be known by |
| nativeEmsAdminServiceState | String | Enum with the following values:<br>■ UNKNOWN<br>■ IN_SERVICE<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_MAINTENANCE | Interface administered state |
| nativeEmsServiceState | Drop Down box | Enum with the following values:<br>■ UNKNOWN<br>■ IN_SERVICE<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_MAINTENANCE | Interface actual state |

*Table 4–2 (Cont.) Characteristics Applied to the interfaceGeneric Specification*

| Characteristics | Field Type | Field Content | Intended Usage/ Notes |
|---|---|---|---|
| ifType | Drop Down box | Text | IANA iftype |
| mtuSupported | String | Text (numeric) | Max allowed MTU |
| mtuCurrent | String | Text (numeric) | Current Value of MTU |
| physicalAddress | String | Text | Interface's address at its protocol sub-layer |
| physicalLocation | String | Text | Location of interface |
| minSpeed | String | Text (numeric) | Minimum speed of interface |
| maxSpeed | String | Text (numeric) | Max speed of interface |
| nominalSpeed | String | Text (numeric) | Nominal speed of interface |
| ifAlias | String | Text | Alternate name this interface could be known by |
| ifName | String | Text | Alternate name this interface could be known by |

## Other Characteristics

This MIB-II SNMP Cartridge additionally models the following characteristics. These characteristics are used by the Network Integrity Cisco Router and Switch SNMP cartridge which extends the sample technology pack.

Table 4–3 shows the characteristics used by Network Integrity Cisco Router and Switch SNMP cartridge.

*Table 4–3 Characteristics used by ora_ni_uim_cisco_device*

| Characteristics | Field Type | Field Content | Intended Usage/ Notes |
|---|---|---|---|
| discoveredModelNumber | String | Text | All Physical Equipment |
| discoveredPartNumber | String | Text | All Physical Equipment |
| discoveredVendorName | String | Text | All Physical Equipment |
| hardwareRev | String | Text | All Physical Equipment |
| softwareRev | String | Text | All Physical Equipment |
| modelName | String | Text | All Physical Equipment |

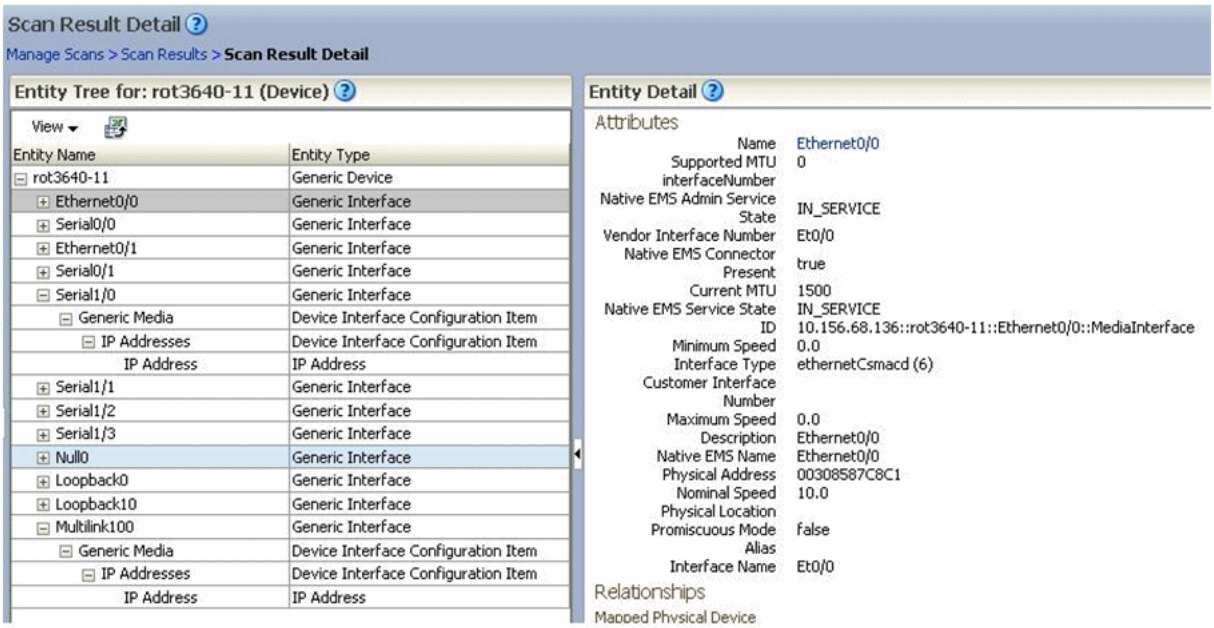
## Specification Lineage

```
deviceGeneric
            [0.100000] interfaceGeneric
                        [0.100000] interfaceGeneric
```

## Sample Logical Device Hierarchy

Figure 4–2 shows a logical device hierarchy.

*Figure 4–2 Sample Logical Device Hierarchy*




# Field Mappings

This section provide information about field mappings used in the cartridge:

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

The following tables describe field mappings used in the cartridge:

- Table 4–4, " Logical Device Mapping"
- Table 4–5, " Media Interface Mapping"
- Table 4–6, " Device Interface Configuration Mapping (IPv4)"
- Table 4–7, " Device Interface Configuration Mapping (IPv6)"
- Table 4–8, " Mapping Table"

# Logical Device Mapping

Table 4–4 shows the logical device mapping for the MIB-II SNMP cartridge.

*Table 4–4 Logical Device Mapping*

| LogicalDevice | Information Model Support | MIB Object | Field Type | 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 specification “deviceGeneric” |
| nativeEmsAdminServiceState | static | N/A | Enum:<br>■ UNKNOWN<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_SERVICE<br>■ IN_MAINTENANCE | No source |
| nativeEmsServiceState | static | N/A | Enum:<br>■ UNKNOWN<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_SERVICE<br>■ IN_MAINTENANCE | No source |
| nativeEmsName | static | sysName | Text | N/A |
| mgmtIpAddress | dynamic | N/A | Text | discoveryAddress |
| sysObjectId | dynamic | sysObjectId | Text | Support legacy systems |

# Media Interface Mapping

Table 4–5 shows the media interface mapping for the MIB-II SNMP cartridge.

*Table 4–5 Media Interface Mapping*

| Media Interface | Information Model Support | MIB Object | Field Type | Notes |
|---|---|---|---|---|
| Id | static | N/A | Text | Programmatically generated asMgmtIPAddress::sysName::ifDesc::"MediaInterface" |
| Name | static | ifDescr | Text | N/A |
| Description | static | ifDescr | Text | N/A |
| Specification | static | N/A | N/A | Programmatically applies specification "interfaceGeneric" |
| interfaceNumber | static | N/A | Text | No source |
| customerInterfaceNumber | static | N/A | Text | No source |

*Table 4–5 (Cont.) Media Interface Mapping*

| Media Interface | Information Model Support | MIB Object | Field Type | Notes |
|---|---|---|---|---|
| vendorInterfaceNumber | static | ifName | Text | N/A |
| nativeEmsName | static | ifDesc | Text | Field must be unique. ifDescr guarantees uniqueness in device. |
| nativeEmsAdminServiceState | static | ifAdminStatus | Enum:<br>■ UNKNOWN<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_SERVICE<br>■ IN_MAINTENANCE | Mapped.<br>See "Mapping Table" for more information. |
| nativeEmsServiceState | static | ifOperStatus | Enum:<br>■ UNKNOWN<br>■ OUT_OF_SERVICE<br>■ TESTING<br>■ IN_SERVICE<br>■ IN_MAINTENANCE | Mapped.<br>See "Mapping Table" for more information. |
| ifType | dynamic | ifType | Text | Mapped from IANA MIB using properties file "snmpIfTypeMap".<br>Look up returning null results in value of "n" |
| mtuSupported | static | N/A | Float | No source. 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 | No source. Defaults to 0.0 |
| maxSpeed | static | N/A | Float | No source. Defaults 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 |

# Device Interface Configuration Mapping (IPv4)

Table 4–6 shows the device interface configuration mapping for Internet Protocol, Version 4, for the MIB-II SNMP cartridge.

*Table 4–6 Device Interface Configuration Mapping (IPv4)*

| Generic Media | Information Model Support | MIB Object | Field Type | 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 specification "GenericMedia" |

# Device Interface Configuration (IPv6)

Table 4–7 shows the device interface configuration for the Internet Protocol, Version 6 for the MIB-II SNMP cartridge.

*Table 4–7 Device Interface Configuration Mapping (IPv6)*

| Generic Media | Information Model Support | MIB Object | Field Type | Notes |
|---|---|---|---|---|
| ipAddress | dynamic | ipAddressIfIndex | Text | For IP v6 addresses, the actual IP is derived from the index since ipAddressAddr is not accessible. |
| prefix | dynamic | ipAddressPrefix | Text | N/A |
| ipVersion | dynamic | ipAddressType | Enum {IPV4 \| IPV6} | Programmatically set to IPV6 |
| Specification | static | N/A | N/A | Programmatically applies specification "GenericMedia" |

# Mapping Table

Table 4–8 shows the mapping of ifOperStatus/ifAdminStatus to nativeEms(Admin)ServiceState.

*Table 4–8 Mapping Table*

| ifOperStatus | ifAdminStatus | nativeEmsServiceState<br>nativeEmsAdminState |
|---|---|---|
| 1: up | 1: up | IN_SERVICE |
| 2: down | 2: down | OUT_OF_SERVICE |
| 3: testing | 3: testing | TESTING |
| 4: unknown | N/A | UNKNOWN |
| 5: dormant | 2: down | OUT_OF_SERVICE |
| 6: notPresent | N/A | UNKNOWN |
| 7: lowerLayerDown | 2: down | OUT_OF_SERVICE |
| N/A | N/A | IN_MAINTENANCE |

# 5

# About Design Studio Construction

This chapter provides information on using Oracle Communications Design Studio to construct for the Oracle Communications Network Integrity MIB-II SNMP cartridge.

## Model Collection

Table 5–1 shows the model collection for the MIB-II SNMP cartridge.

***Table 5–1 MIB-II Model Collection***

| Specification | Notes |
|---|---|
| deviceGeneric | Intended to represent any root object discovered on the network |
| interfaceGeneric | Intended to represent any interface discovered under deviceGeneric |
| GenericMedia | Intended to represent IpAddresses that are applied to a interfaceGeneric |
| IPAddresses | Container of IP addresses |
| IPAddress | Container of IP address details |

## Specification Lineage

This section outlines the specification lineage.

```
deviceGeneric
   [0..*] interfaceGeneric
      [0..1] GenericMedia
         [0..1] IP Addresses
            [0..*] IpAddress
                   IpAddress (characteristic)
                   Prefix (characteristic)
                   IpVersion (characteristic)
```

## Discovery Action

Table 5–2 shows the discovery action for the MIB-II SNMP cartridge.

*Table 5–2 Discover MIB-II SNMP Action*

| Result Category | Address Handler | Scan Parameters | Model | Processors |
|---|---|---|---|---|
| Device | IPAddressHandler | ▪ version, Enum {v1, v2c, v3}<br>▪ port, String<br>▪ snmpReadCommunity, String<br>▪ snmpTimeout, String<br>▪ snmpRetries, String<br>▪ username, String<br>▪ contextName, String<br>▪ authProtocol, Enum {MD5, SHA}<br>▪ authPassword, String<br>▪ privacyProtocal Enum {DES}<br>▪ privacyPassword, String | MIB-II Model | ▪ MIB-II Properties Initializer<br>▪ MIB-II SNMP Collector<br>▪ MIB-II SNMP Modeler |

Figure 5–1 depicts the Discover MIB-II SNMP action chain.

*Figure 5–1 Discover MIB-II SNMP Action Chain*


# Discovery Processors

Table 5–3 shows the discovery processors for the MIB-II SNMP cartridge.

*Table 5–3 Discovery Processors*

| Processor Name | Variable |
|---|---|
| MIB-II Properties Initializer | Input: N/A<br>Output:<br>▪ snmpIfTypeMap<br>Property map containing listing of ifTypes to string name.<br>▪ snmpVendorNameMap<br>Property map containing listing of sysObjectId suffixes to vendorName. |
| MIB-II SNMP Collector | Input: N/A<br>Output:<br>▪ mibiisnmpCollectorResponseDocument (implicit)<br>Polled SNMP data, see "About Poll Lists". |
| MIB-II SNMP Modeler | Input: mibiisnmpCollectorResponseDocument, snmpIfTypeMap<br>Output:<br>▪ deviceInterfaceMap<br>A map that contains interfaces with IfIndex as key.<br>▪ logicalDevice<br>This is the logical device that was created in the MIB-II Modeler. |

The MIB-II SNMP Collector polling the SNMP variables implicitly outputs this data in mibiisnmpCollectorResponseDocument. This output field is not seen explicitly in Design Studio. This document is available to the MIB-II SNMP Modeler processor.

## Property File Sample

Table 5–4 shows a sample of the property files being loaded by the MIB-II Properties Initializer.

*Table 5–4 Sample Property Files Loaded by MIB-II Properties Initializer*

| Sample snmpIfTypeMap | Sample snmpVendorNameMap |
|---|---|
| 1: other (1) | 0 = Reserved |
| 2: regular1822 (2) | 1 = NxNetworks |
| 3: hdh1822 (3) | 2 = IBM |
| 4: ddnX25 (4) | 3 = Carnegie Mellon |
| 5: rfc877x25 (5) | 4 = UNIX |
| 6: ethernetCsmacd (6) | 5 = ACC |
| 7: iso88023Csmacd (7) | 6 = TWG |
| 8: iso88024TokenBus (8) | 7 = CAYMAN |
| 9: iso88025TokenRing (9) | 8 = PSI |
| 10: iso88026Man (10) | 9 = ciscoSystems |
| 251: vdsl2 (251) | 34730 = FRANCILIENNE D'INGENIERIE ET DE SERVICES INFORMATIQUES SAS |

**Note:** The **snmpVendorNameMap** is not used in this cartridge but is initialized for extension cartridges to make use of this map.

# 6

# About Design Studio Extensions

This chapter provides information about Oracle Communications Design Studio extensions to the Oracle Communications Network Integrity MIB-II SNMP cartridge.

## Updating Property Files

To update the property files (if new interface types or new vendors are introduced), the user can extend **Discover MIB-II SNMP** and add a new **Post** processor, which takes as input, snmpIfTypeMap or snmpVendorNameMap.

Source code can then update the map either programmatically or by loading a new property file and then (re)write into the map. For further information on extensibility, see *Network Integrity Developer's Guide*.