

# Oracle® Communications Network Integrity

## IMS FTP Discovery and UIM Integration Cartridge Guide



Release 8.0  
G34164-01  
October 2025

ORACLE®

Copyright © 2024, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## About This Content

---

### 1 Overview

---

About the IMS Discovery Cartridge	1
About Cartridge Dependencies	2
Run-Time Dependencies	2
Design-Time Dependencies	2
Opening the Cartridge Files in Design Studio	2
Building and Deploying the Cartridge	2

### 2 About the Cartridge Components

---

Discover IMS Network	1
IMS Network File Transfer	1
Scan Parameter Group	1
IMS Network File Parser	2
IMS Network Device Modeler	3
Detect IMS UIM Discrepancies	3
Resolve IMS UIM	3

### 3 Using the Cartridge

---

Creating a Discover IMS Network Discovery Scan	1
Populating UIM with Discovered Data	1

### 4 About Cartridge Modeling

---

IMS Discovery Cartridge Representation	1
Specifications	2
Physical Device	2
Equipment	3
Equipment Holder	3
Physical Port	4

Logical Device	4
Device Interface	5
Device Interface Configuration Item	6
Specification Cardinality	6
Equipment Visual Specifications	6
Field Mapping	6
Logical Mapping	6
Logical Device Mapping	7
Media Interface Mapping	7
Device Interface Configuration Item Mapping	9
Physical Mapping	9
Physical Device	9
Equipment Shelf	10
Equipment Holder	11
Equipment Card	11
Physical Port	12

## 5 About Design Studio Construction

---

Model Collections	1
Logical Specification Lineage for Devices	1
Physical Specification Lineage for Devices	2
Discovery Action	2
Discovery Processors	2
Discrepancy Detection Action	3
Discrepancy Resolution Action	3

## 6 About IMS Network FTP Input File Structure

---

NE Report File	1
Shelf Report File	2
Frame Report File	3
Board Report File	5
Module Report File	7
Port Report File	9

## 7 About Design Studio Extension

---

Adding a New Vendor	1
Adding a New Device Type under Existing Vendors	1

# About This Content

This guide describes the functionality and design of the Oracle Communications IMS Discovery 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 extend similar cartridges.

Developers should have a good working knowledge of FTP and FTP operations, specifications, Network Integrity, UIM, and the use of Oracle Communications Service Catalog and Design - 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 File Transfer and Parsing Guide*
- *Oracle Communications Network Integrity UIM Integration 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.

## Conventions

The following text conventions are used in this document.

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# 1

## Overview

This chapter provides an overview of the Oracle Communications Network Integrity IMS Discovery cartridge.

This chapter contains the following sections:

- [About the IMS Discovery Cartridge](#)
- [About Cartridge Dependencies](#)
- [Opening the Cartridge Files in Design Studio](#)
- [Building and Deploying the Cartridge](#)

## About the IMS Discovery Cartridge

The IMS Discovery Cartridge supports the modeling of devices in IMS Network and provides the following functionalities:

- Read and collect data from files provided using FTP.
- Physical discovery and modeling of devices.
- Discrepancy detection and resolution on modeled data.

This cartridge produces both logical and physical device hierarchies that represent a discovered device. The logical hierarchy includes a logical device, child interfaces, sub-interfaces (collectively called interfaces), and device interface configurations. The physical hierarchy includes physical devices, equipment, equipment holders, and physical ports. In addition, the 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.

The cartridge discovers the following network entities:

- NE
- Frame
- Shelf
- Module
- Port
- Interface

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 IMS domain devices only and attempts to discover other devices result in a scan failure.

In addition to a discovery action, this cartridge provides discrepancy detection and discrepancy resolution for integration with Unified Inventory Management.

Discrepancy detection provides the mechanism to allow a filtered comparison of logical and physical device trees between what is discovered and what is imported from UIM.

For more information about discrepancy detection actions and processors, see "About Discrepancy Detection Actions" in *Network Integrity Developer's Guide*.

The discrepancy resolution action enables the discovered logical and physical device trees to be created and updated in UIM.

For **FTP**, modeling of the device fails if it does not follow the FTP file structure properly. For more information about IMS Network FTP Input file structure, see "[About IMS Network FTP Input File Structure](#)".

## About Cartridge Dependencies

This section provides information on dependencies that the IMS Discovery 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 IMS Discovery cartridge has the following dependencies:

- Address\_Handlers
- NetworkIntegritySDK
- ora\_ni\_uim\_device
- ora\_uim\_model
- UIM\_Integration\_Cartridge

## Opening the Cartridge Files in Design Studio

To review and extend the IMS Discovery cartridge, you must first download the Oracle Communications IMS Discovery Cartridge software from the Oracle software delivery website:

<https://edelivery.oracle.com>

The software contains the IMS Discovery cartridge ZIP file, which has the following structure:

- \UIM\_Cartridge\_Projects\
- \Network\_Integrity\_Cartridge\_Projects\
- IMS-Discovery-Cartridge-R7.iar

See *Design Studio Modeling Network Integrity* and "Using Design Studio to Extend Network Integrity" in *Network Integrity Developer's Guide* for information about opening files in Design Studio.

## Building and Deploying the Cartridge

See *Design Studio Modeling Network Integrity* 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 IMS Discovery cartridge.

The IMS Discovery Cartridge contains the following actions:

- Discover IMS Network
- Detect IMS UIM Discrepancies
- Resolve IMS UIM

### Discover IMS Network

The Discover IMS Network action scans 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.

The Discover IMS Network action contains the following processors run in the following order:

- [IMS Network File Transfer](#)
- [IMS Network File Parser](#)
- [IMS Network Device Modeler](#)

### IMS Network File Transfer

IMS Network File Transfer is a file transfer processor. See "Overview" in *Network Integrity File Transfer and Parsing Guide* for more information on File Transfer processors.

The file transfer processor outputs a single parameter. This output parameter holds a collection of file objects, each of which points to the local version of the transferred file. This collection of files is used as input to the next processor IMS Network File Parser.

The name of the output parameter is system-generated and based on the name of the file transfer processor. For example, the Sample File Transfer processor outputs `sampleFileTransferFileCollection`.

The type of the output parameter is always `java.util.Collection<java.io.File>`.

### Scan Parameter Group

When a cartridge is deployed with scan parameter groups that are generated from a file transfer processor, they appear in the Network Integrity UI as scan parameters. IMS Network File Transfer Scan Parameters are added as scan parameter to IMS Network File Transfer processor.

[Table 2-1](#) lists the scan parameters available during scan creation.



**Table 2-1 Scan parameters available during scan creation**

Parameter	Default Value	Mandatory	Description
File Pattern	N/A	No	A pattern to match file names. The pattern supports wildcard characters. The supported wildcard characters are "*", "%", and "_". "*" and "%" represent a match of zero or more characters. "_" represents a match of any single character. Wildcard characters can be escaped with a backslash.
Password	N/A	No	The password to connect to the remote location.
Port	N/A	No	The port is used to connect to the remote server. The default is 21 for FTP and 22 for SFTP.
Rename Suffix	processed	No	The suffix to add to the source file if the source file management characteristic has a value of Rename.
Session Timeout (sec)	60	No	The amount of time in seconds before an idle connection is timed out. The valid range is from 1 to 3600.
Source File Management	Rename	No	Select the action to take on source files when the file transfer is complete. Options are: Delete, Rename, Nothing.
Transfer Type	SFTP	Yes	Select how files should be transferred: FTP, SFTP, Local.
User Name	N/A	No	The user name to connect to the remote location.

## IMS Network File Parser

IMS Network File Parser is a discovery processor. This processor accepts the list of files from IMS Network File Transfer processor as an input and is used to parse files and collect required data.

Properties file available in cartridge: `IMS_Discovery_Cartridge/src/imsnetworkfiles.properties` contains key value pairs of file names that should be considered.

The sample `imsnetworkfiles.properties` is as follows:

```
Board_Information_File_Name=Board_Information_  
Model_Information_File_Name=Model_Information_  
Frame_Information_File_Name=Frame_Information_  
Port_Information_File_Name=Port_Information_  
NE_Information_File_Name=NE_Information_  
Shelf_Information_File_Name=Shelf_Information_
```

Key **Board\_Information\_File\_Name** contains value **Board\_Information\_**. This specifies that only files with names starting with "**Board\_Information\_**" will be considered for discovery and modeling Boards. The same rule is applicable for all other types of files as well.

IMS Network File Parser processor generates collection of `frameObjects`, `neObjects`, `shelfObjects`, and `portObjects` as output.

## IMS Network Device Modeler

IMS Network Device Modeler is a discovery processor. It accepts collection of objects generated by IMS Network File Parser as input. This processor models the collected data into physical and logical devices.

## Detect IMS UIM Discrepancies

The Detect IMS UIM Discrepancies action detects discrepancies between discovery scan results of action Discover IMS Network and data imported from UIM.

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

The Detect IMS UIM Discrepancies action contains the following processors run in the following order:

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

## Resolve IMS UIM

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

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

The Resolve IMS UIM action contains the following processors run in the following order:

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

# 3

## Using the Cartridge

This chapter provides instructions for using the Oracle Communications Network Integrity IMS Discovery Cartridge in Network Integrity.

### Creating a Discover IMS Network Discovery Scan

Discover IMS Network scan is used to discover and model data of devices in IMS Network.

This discovery scan discovers network information and creates network entities Physical device, shelf, card, port, interface/sub-interface.

To create a Discover IMS Network discovery scan:

1. Create a new scan.  
See "Creating a Scan" in *Network Integrity Online Help* for more information about creating a new scan.
2. On the **General** tab, select **Discover IMS Network** from the Scan Action list.  
The Scan Type field displays **Discovery**.
3. In the Scan Action Parameters section, configure the scan with appropriate parameters.
4. Save and run the scan.  
The scan discovers and models logical and physical devices.

The scan creates a device model for each logical and physical device.

### Populating UIM with Discovered Data

This procedure describes steps to populate UIM with network data discovered by the Discover IMS Network discovery action.

To populate UIM with discovered network data:

1. Create a new scan.  
See "Creating a Scan" in *Network Integrity Online Help* for more information on creating a new scan.
2. On the **General** tab of the Create Scan page:
  - a. From the Scan Action list, select **Discover IMS Network**.  
The Scan Type field displays **Discovery**.
  - b. Select **Detect Discrepancies**.
  - c. In the Scan Action Parameters area, make any necessary configurations.
3. Save the scan.
4. Run the discovery scan.  
The scan generates Entity+ discrepancies for each discovered device.

5. Right-click on the discrepancies you want to populate into UIM and select **Correct in UIM**.
6. Click **Submit**.
7. Verify that UIM is populated with the discovered data.

**Note**

The import action **Import from UIM** available in UIM\_Integration\_Cartridge can be used to import data from UIM.

# 4

## About Cartridge Modeling

The Oracle Communications Network Integrity IMS Discovery Cartridge models collected data according to the Oracle Communications Information Model. Collected data is modeled into the following entities:

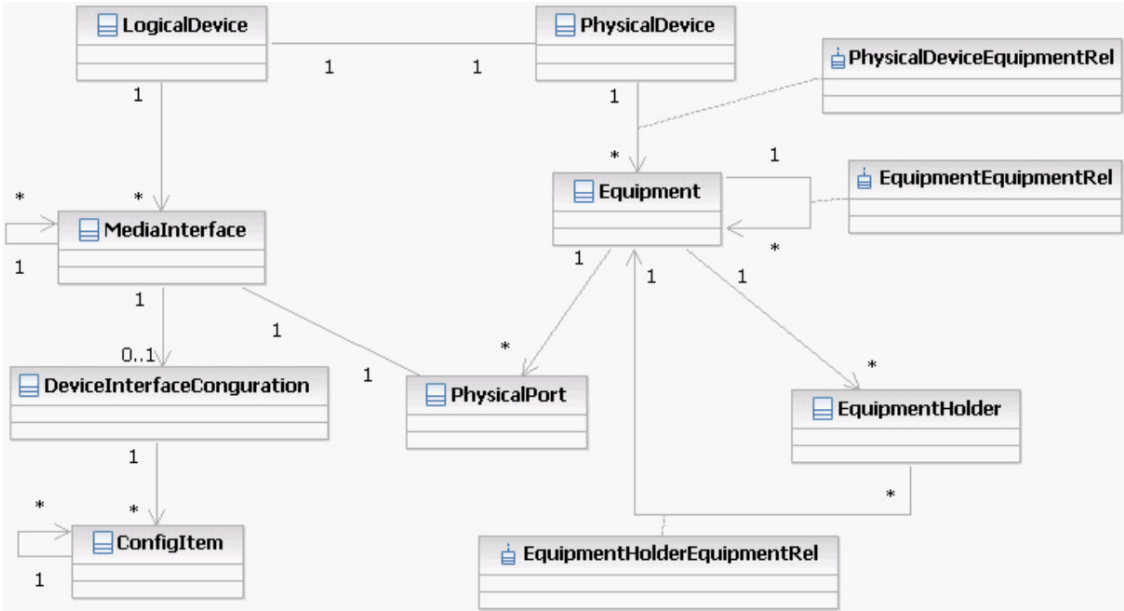
- DeviceInterfaceConfiguration
- DeviceInterfaceConfigurationItem
- Equipment
- EquipmentHolder
- EquipmentEquipmentRel
- EquipmentHolderEquipmentRel
- LogicalDevice
- MediaInterface
- PhysicalDevice
- PhysicalDeviceEquipmentRel
- PhysicalPort

See *Oracle Communications Information Model Reference* for more information about the Information Model.

## IMS Discovery Cartridge Representation

[Figure 4-1](#) depicts the object relationship being rendered.

Figure 4-1 Diagram of Object Relationship being rendered



## Specifications

This section lists the specifications included in the ora\_ni\_uim\_device for devices.

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

Specifications shared with Oracle Communications Unified Inventory Management (UIM) are defined in the ora\_ni\_uim\_device cartridges. These cartridges are used to directly deploy specifications to UIM.

## Physical Device

[Table 4-1](#) shows the list of physical device specifications.

Table 4-1 Physical Device Specifications

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

GenericPhysicalDeviceSpecification is available in ora\_ni\_uim\_device and is used to model all types of IMS devices.

[Table 4-2](#) shows the characteristics applied to the physical device specifications.

**Table 4-2 Physical Device Characteristics**

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

## Equipment

[Table 4-3](#) shows the list of equipment specifications.

**Table 4-3 Equipment Specifications**

Specification	Cartridge	Intended Usage
GenericEquipmentShelfSpecification	ora_ni_uim_device	Used to model all types of devices.
GenericEquipmentSpecification	ora_ni_uim_device	Used to model all types of devices.

GenericEquipmentShelfSpecification and GenericEquipmentSpecification are available in ora\_ni\_uim\_device and used to model all types of IMS devices.

[Table 4-4](#) shows the characteristics applied to equipment specifications.

**Table 4-4 Equipment Characteristics**

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

## Equipment Holder

[Table 4-5](#) shows the list of equipment holder specifications.

**Table 4-5 Equipment Holder Specifications**

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

[Table 4-6](#) shows the characteristics applied to the equipment holder specifications.

**Table 4-6 Equipment Holder Characteristics**

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

## Physical Port

[Table 4-7](#) shows the list of physical port specifications.

**Table 4-7 Physical Port Specifications**

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

[Table 4-8](#) shows the characteristics applied to the physical port specifications.

**Table 4-8 Physical Port Characteristics**

Characteristics	Field Type	Field Content
direction	String	Text
nativeEmsName	String	Text
discoveredVendorName	String	Text
edgePoint	String	Text
tpProtectionAssociation	String	Text

## Logical Device

[Table 4-9](#) shows the list of logical device specifications.

**Table 4-9 Logical Device Specifications**

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



[Table 4-10](#) shows the characteristics applied to the GenericLogicalDeviceSpecification specification.

**Table 4-10 Logical Device Characteristics for GenericLogicalDeviceSpecification specification**

Characteristics	Field Type	Field Content
mgmtIpAddress	String	Text
modelName	String	Text
nativeEmsName	String	Text
nativeEmsAdminServiceState	String	Text
nativeEmsServiceState	String	Text
owner	String	Text
sysObjectId	String	Text
userLabel	String	Text

## Device Interface

[Table 4-11](#) shows the list of device interface specifications.

**Table 4-11 Device Interface Specifications**

Specification	Cartridge	Intended Usage
GenericInterfaceSpecification	ora_ni_uim_device	Used to model all types of cyan devices.

[Table 4-12](#) shows the characteristics applied to the GenericInterfaceSpecification specification.

**Table 4-12 Device Interface Characteristics for GenericInterfaceSpecification Specification**

Characteristics	Field Type	Field Content
ifAlias	String	Text
ifName	String	Text
direction	String	Text
edgePoint	Boolean	Text
ifPromiscuousMode	Boolean	Text
mtuCurrent	Long	Text
mtuSupported	Long	Text
nativeEmsAdminServiceState	String	Text
nativeEmsConnectorPresent	Boolean	Text
nativeEmsName	String	Text
owner	String	Text
physicalAddress	String	Text
userLabel	String	Text

## Device Interface Configuration Item

[Table 4-13](#) shows the list of device interface configuration item specifications.

**Table 4-13 Device Interface Configuration Item Specifications**

Specification	Cartridge	Intended Usage
IMS_Generic_Media	IMS_Discovery_Cartridge	Used to model Device Interface IPAddress Details.

[Table 4-14](#) shows the characteristics applied to IMS\_Generic\_IPAddress device interface configuration item.

**Table 4-14 Device Interface Configuration Item Characteristics for IMS\_Generic\_IPAddress**

Characteristics (IMS_Generic_Media)	Field Type	Field Content
ipAddress	String	Text
prefix	String	Text
ipVersion	Enum	<ul style="list-style-type: none"><li>IPV4</li><li>IPV6</li></ul>

## 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. The user may decide to enrich the technology pack to provide visual effects to the UIM GUI for a given Equipment entity.

## Field Mapping

The IMS Discovery Cartridge supports the following field mappings:

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

## Logical Mapping

The IMS Discovery cartridge supports the following logical mappings:

- Logical Device
- Media Interface
- Device Interface Configuration Item Mapping (IMS\_Generic\_Media)

## Logical Device Mapping

**Table 4-15 Logical Device Mapping**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject name and ipaddress	Text
name	static	neObject name	Text
Description	static	N/A	Text
nativeEMSName	static	neObject name	Text
mgmtIPAddress	dynamic	N/A	Text
sysObjectId	dynamic	N/A	Text
Specification	static	Programmatically applies GenericLogicalDevice specification.	N/A
nativeEmsAdminService State	static	N/A	Enum with the following values: <ul style="list-style-type: none"> <li>• UNKNOWN</li> <li>• IN_SERVICE</li> <li>• OUT_OF_SERVICE</li> <li>• TESTING</li> <li>• IN_MAINTENANCE</li> </ul>
nativeEmsServiceState	static	N/A	Enum with the following values: <ul style="list-style-type: none"> <li>• UNKNOWN</li> <li>• IN_SERVICE</li> <li>• OUT_OF_SERVICE</li> <li>• TESTING</li> <li>• IN_MAINTENANCE</li> </ul>

## Media Interface Mapping

**Table 4-16 GenericInterfaceSpecification Device Interface Mapping**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using portObject Trunk_group_name	Text
Name	static	portObject Trunk_group_name	Text

**Table 4-16 (Cont.) GenericInterfaceSpecification Device Interface Mapping**

Characteristic	Information Model Type	Value	Field Type
description	static	N/A	Text
specification	static	Programmatically applies specification "GenericInterfaceSpecification" .	N/A
interfaceNumber	static	No source	Text
customerInterfaceNumber	static	No source	Text
vendorInterfaceNumber	static	ifName	Text
nativeEmsName	static	Programmatically generated using portObject Trunk_group_name and port nativeEMSName	Text
nativeEmsAdminService State	static	N/A	Enum with the following values: <ul style="list-style-type: none"> <li>UNKNOWN</li> <li>IN_SERVICE</li> <li>OUT_OF_SERVICE</li> <li>TESTING</li> <li>IN_MAINTENANCE</li> </ul>
nativeEmsServiceState	static	N/A	Enum with the following values: <ul style="list-style-type: none"> <li>UNKNOWN</li> <li>IN_SERVICE</li> <li>OUT_OF_SERVICE</li> <li>TESTING</li> <li>IN_MAINTENANCE</li> </ul>
ifType	dynamic	N/A	Text
mtuSupported	static	Nothing available to source the field. Defaults to 0.0	Float
mtuCurrent	static	N/A	Float
physicalAddress	static	N/A	Text
physicalLocation	static	N/A	Text
minSpeed	static	N/A	Float
maxSpeed	static	N/A	Float
nominalSpeed	static	N/A	Float
ifAlias	dynamic	N/A	Text
ifName	dynamic	N/A	Text

## Device Interface Configuration Item Mapping

**Table 4-17 Device Configuration Item (IPv4) Mapping**

Characteristic (IMS Generic Media)	Information Model Type	Value	Field Type
ipAddress	dynamic	Programmatically generated using portObject Remote_URI	Text
prefix	dynamic	Programmatically generated using portObject Second_Remote_URI	Text
ipVersion	dynamic	Set to "IPv4"	Enum <ul style="list-style-type: none"><li>• IPV4</li><li>• IPV6</li></ul>
specification	static	Programmatically applies IMS_Generic_Media specification	N/A

## Physical Mapping

The IMS Discovery cartridge supports the following physical mappings:

- [Physical Device](#)
- [Equipment Shelf](#)
- [Equipment Holder](#)
- [Equipment Card](#)
- [Physical Port](#)

## Physical Device

**Table 4-18 Physical Device**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject Name and ipaddress	Text
name	static	neObject Name	Text
description	static	N/A	Text
nativeEMSName	static	neObject Name	Text
physicalAddress	static	N/A	Text
physicalLocation	static	N/A	Text

**Table 4-18 (Cont.) Physical Device**

Characteristic	Information Model Type	Value	Field Type
specification	static	Programmatically set to GenericPhysicalDeviceSpecification	N/A
discoveredPartNumber	dynamic	N/A	Text
discoveredVendorNumber	dynamic	set to "IMS VENDOR"	Text
discoveredModelNumber	dynamic	N/A	Text
modelName	dynamic	Set to "IMS Device"	Text
mgmtIPAddress	dynamic	N/A	Text
hardwareRev	dynamic	N/A	Text
owner	dynamic	N/A	Text

## Equipment Shelf

**Table 4-19 Equipment Shelf**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject Name and ipaddress and shelf frame number	Text
name	static	Programmatically set to "shelf" + shelfObject Frame_number	Text
description	static	N/A	Text
nativeEMSName	static	Programmatically set to "shelf" + shelfObject Frame_number	Text
specification	static	Programmatically set to "GenericEquipmentShelf Specification"	N/A
serialNumber	static	Programmatically set using partNumberMappings.properties file	Text
discoveredPartNumber	dynamic	N/A	Text
discoveredVendorName	dynamic	set to "IMS VENDOR"	Text
modelName	dynamic	Set to "VOICE-SHELF"	Text
discoveredModelNumber	dynamic	N/A	Text
owner	dynamic	set to "IMS VENDOR"	Text

## Equipment Holder

**Table 4-20 Equipment Holder**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject Name and ipaddress and slot name.	Text
name	static	Programmatically generated as "slot=" + boardObject slot number + boardObject location	Text
nativeEMSName	static	Programmatically generated using boardObject frame number, board slot number and board location.	Text
owner	static	Set to IMS VENDOR	Text
specification	static	Programmatically set to "GenericEquipmentHolderSpecification"	N/A
discoveredVendorName	dynamic	Set to "IMS VENDOR"	Text
modelName	dynamic	N/A	Text

## Equipment Card

**Table 4-21 Equipment Card**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject Name and ipaddress and card name.	Text
name	static	boardObject Board_type	Text
nativeEMSName	static	Programmatically generated using board frame number, board slot number, board location and board type.	Text
specification	static	Programmatically set to "GenericEquipmentSpecification"	Text
serialNumber	dynamic	N/A	Text
description	dynamic	N/A	Text
discoveredVendorName	dynamic	Set to "IMS VENDOR"	Text

**Table 4-21 (Cont.) Equipment Card**

Characteristic	Information Model Type	Value	Field Type
discoveredModelNumber	dynamic	boardObject module number	Text
discoveredPartName	dynamic	Set based on partNumberMappings.properties and card name.	Text
owner	static	Set to "IMS VENDOR"	Text
modelName	dynamic	Set to "VOICE-CARD"	Text
softwareRev	dynamic	Set to "IMS Module Name".	Text
hardwareRev	dynamic	N/A	Text
userLabel	dynamic	Set based on slot name	Text

## Physical Port

**Table 4-22 Physical Port**

Characteristic	Information Model Type	Value	Field Type
id	static	Programmatically generated using neObject Name and ipaddress and portObject Trunk_No.	Text
name	static	portObject Trunk_group_name	Text
nativeEMSName	static	card nativeEMSName + portObject getTrunk_No	Text
portNumber	static	portObject Local_server_port	Text
vendorPortName	static	N/A	Text
description	static	N/A	Text
customerPortName	static	N/A	Text
specification	static	Programmatically set to "GenericPhysicalPortSpecification"	N/A
discoveredVendorName	dynamic	set to "IMS VENDOR"	Text



# 5

## About Design Studio Construction

This chapter provides information on the composition of the Oracle Communications Network Integrity Discovery IMS cartridge from the Oracle Communications Service Catalog and Design - Design Studio perspective.

### Model Collections

[Table 5-1](#) shows the Discovery IMS Cartridge model collection.

**Table 5-1 Discovery IMS Cartridge Model Collection**

Specification	Information Model Entity Type	Intended Usage/Notes
GenericLogicalDeviceSpecification	LogicalDevice	Represents root object discovered on the network.
GenericInterfaceSpecification	DeviceInterface	Represents interface discovered under GenericLogicalDeviceSpecification. Device Interfaces discovered and modelled are modelled into GenericInterfaceSpecification.
GenericPhysicalDeviceSpecification	PhysicalDevice	Represents any Physical Device discovered on the network.
GenericEquipmentHolderSpecification	EquipmentHolder	Represents Board piece of equipment.
GenericEquipmentShelfSpecification	Equipment	Represents Shelf piece of equipment.
GenericEquipmentSpecification	Equipment	Represents Module piece of equipment.
GenericPhysicalPortSpecification	PhysicalPort	Represents any physical port.

### Logical Specification Lineage for Devices

The example below shows a logical specification lineage for Devices. This lineage shows the intended relationship between specifications.

```
GenericLogicalDeviceSpecification
  [0..*] GenericInterfaceSpecification
    [0..1] IMS_Generic_Media
      [0..1] IMS_IpAddresses
        [0..*] IMS_IpAddress
          IpAddress (characteristic)
          Prefix (characteristic)
          IpVersion (characteristic)
```

# Physical Specification Lineage for Devices

The example below shows a physical specification lineage for Devices. This lineage shows the intended relationship between specifications.

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

## Discovery Action

Discovery IMS Cartridge supports discovery action Discover IMS Network.

[Table 5-2](#) shows the discovery action in the IMS Discovery Cartridge.

**Table 5-2 Discover IMS Network**

Result Category	Address Handler	Scan Parameters	Model	Processors
Device	FileTransferAddressHandler	File Pattern Transfer Type User Name Password Port Rename Suffix Session Timeout (sec) Source File Management <b>Note:</b> These scan parameters must be added for the Create Scan web service request even if the values are left empty.	IMS_Discovery_Cartridge	IMS Network File transfer IMS Network File Parser IMS Network Device Modeller

## Discovery Processors

**Table 5-3 Discover IMS Network Action Processors**

Processor Name	Variable
IMS Network File transfer	Input: N/A Output: imsNetworkFileTransferFileCollection Collection of files read from the scope provided.

Table 5-3 (Cont.) Discover IMS Network Action Processors

Processor Name	Variable
IMS Network File Parser	Input: imsNetworkFileTransferFileCollection Collection of files read from the scope provided. Output: <ul style="list-style-type: none"><li>• frameObjects</li><li>• shelfObjects</li><li>• portObjects</li><li>• neObjects</li></ul> Collections of objects read from file.
IMS Network Device Modeller	Input: <ul style="list-style-type: none"><li>• frameObjects</li><li>• shelfObjects</li><li>• portObjects</li><li>• neObjects</li></ul> Collections of objects read from file. Output: <ul style="list-style-type: none"><li>• physicalDevice This is the Physical Device that is created in the processor.</li><li>• logicalDevice This is the Logical Device that is created in the processor.</li></ul>

## Discrepancy Detection Action

Detect IMS UIM Discrepancies is the action used to perform Discrepancy detection.

Table 5-4 Detect IMS UIM Discrepancies

Result Category	Result Source	Scan Parameters	Model	Processors
All	Discover IMS Network	N/A	IMS_Discovery_Cartridge	This action extends the Abstract Detect UIM Discrepancies action included in the UIM Integration cartridge. See "Abstract Detect UIM Discrepancies Action" in <i>Network Integrity UIM Integration Cartridge Guide</i> for more information.

## Discrepancy Resolution Action

Resolve IMS UIM is the action used to perform Discrepancy Resolution.

Table 5-5    Resolve IMS UIM

Result Category	Result Source	Processors
All	Discover IMS Network	This action extends the Abstract Resolve in UIM action included in the Network Integrity UIM Integration cartridge. For more information, see "Abstract Resolve in UIM Action" in <i>Network Integrity UIM Integration Cartridge Guide</i> .

# 6

## About IMS Network FTP Input File Structure

For IMS Network FTP scan to work properly, files must be populated with proper value.

[Table 6-1](#) shows the list of supported files:

**Table 6-1 List of Supported Files**

FileName	Mandatory	Description
NE_Information_File	Yes	This file contains NetworkElement information.
Shelf_Information_File	Yes	This file contains shelf (equipment) information.
Board_Information_File	Yes	This file contains cards (equipment) information.
Frame_Information_File	Yes	This file contains frame related information.
Model_Information_File	Yes	This file contains module information.
Port_Information_File	Yes	This file contains physical port related information.

### Note

- File naming convention can be changed by making necessary changes inside **imsnetworkfiles.properties** located at **IMS\_Discovery\_Cartridge/src**.
- Each file is structured with a predetermined set of columns, each containing essential device information. Each file contains metadata at the start and the results of the report begins from the eighth line. Subsequent lines contain the specific details pertaining to each device.

## NE Report File

[Table 6-2](#) shows the list of column headers required within the file:

**Table 6-2 List of Column Headers**

Header Name	Description
SoftX3000 Name	Network element name
IP Address	IP address of the element
Result	Result of the element

Sample NE Report File

```
+++      HW-UMS      2023-08-25 08:19:06-07:00 DST
O&M      #256676
%%DSP PATCH:;%%
RETCODE = 0 Network Management Operation Succeed

Operation Summary
-----
Name                      IP Address      Result

Sample1                   5.0.0.1        Success
(Result number = 1)

---      END
```

Shelf Report File

[Table 6-3](#) shows the list of column headers required within the file:

Table 6-3 List of Column Headers

Header Name	Description
Frame number	Id of the shelf (ranges from 0 to n-1 for n number of shelves)
Shelf number	Shelf number
Position number	Position number of shelf

Sample Shelf Report File

```
+++      LOCAL      2023-08-25 10:25:56-15:00 DST
O&M      #8435619
%%LST FRM:;%%
RETCODE = 0 Operation succeeded

Frame information
-----
Frame number  Shelf number  Position number

0             0             2
1             0             3
2             1             0
3             1             1
4             1             2
5             1             3
6             2             0
7             2             1
8             2             2
9             2             3
(Number of results = 10)

---      END
```

## Frame Report File

This file contains sections for each frame available in Shelf report file. Each section contains details about slots of that particular frame.

[Table 6-4](#) shows the list of column headers required within the file:

**Table 6-4 List of Column Headers**

Header Name	Description
Frame	ID of the frame.
Slot	ID of the slot.
FrontBrd	Front board
FrontBrdState	State of front board
UpperPMC	Upper PMC
LowerPMC	Lower PMC
BackBrd	Back board
BackBrdState	State of back board

### Sample Frame Report File

```
+++    LOCAL          2023-08-25 10:25:56-15:00 DST
O&M    #8435619
%%DSP FRM:;%
RETCODE = 0  Operation succeeded
```

#### Frame 0 Information

```
-----
Frame  Slot  FrontBrd  FrontBrdState  UpperPMC  LowerPMC
BackBrd  BackBrdState

0        0      IFMI      Standby ok      Uninstalled
Uninstalled  BFII      Normal
0        1      IFMI      Main ok         Uninstalled
Uninstalled  BFII      Normal
0        2      FCCU      Standby ok      Uninstalled
Uninstalled  NULL      Uninstalled
0        3      FCCU      Main ok         Uninstalled
Uninstalled  NULL      Uninstalled
0        4      FCCU      Standby ok      Uninstalled
Uninstalled  NULL      Uninstalled
0        5      FCCU      Main ok         Uninstalled
Uninstalled  NULL      Uninstalled
0        6      SMUI      Main ok         Uninstalled
Uninstalled  SIUI      Main ok
0        7      NULL      Uninstalled     Uninstalled
Uninstalled  HSCI      Normal
0        8      SMUI      Standby ok      Uninstalled
Uninstalled  SIUI      Standby ok
0        9      NULL      Uninstalled     Uninstalled
Uninstalled  HSCI      Normal
0       10      BSGI      Standby ok      Uninstalled
```

Uninstalled		NULL	Uninstalled	
0	11	BSGI	Main ok	Uninstalled
Uninstalled		NULL	Uninstalled	
0	12	CDBI	Standby ok	Uninstalled
Uninstalled		NULL	Uninstalled	
0	13	CDBI	Main ok	Uninstalled
Uninstalled		NULL	Uninstalled	
0	14	MSGI	Standby ok	Uninstalled
Uninstalled		NULL	Uninstalled	
0	15	MSGI	Main ok	Uninstalled
Uninstalled		NULL	Uninstalled	
0	16	ALUI	Normal	Uninstalled
Uninstalled		NULL	Uninstalled	
0	17~18	UPWR	Normal	Uninstalled
Uninstalled		UPWR	Normal	
0	19~20	UPWR	Normal	Uninstalled
Uninstalled		UPWR	Normal	

## Frame 1 Information

-----

Frame	Slot	FrontBrd	FrontBrdState	UpperPMC	LowerPMC
BackBrd	BackBrdState				
1	0	CDBI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	1	CDBI	Standby ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	2	FCCU	Standby ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	3	FCCU	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	4	FCCU	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	5	FCCU	Standby ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	6	SMUI	Standby ok	Uninstalled	
Uninstalled		SIUI	Standby ok		
1	7	NULL	Uninstalled	Uninstalled	
Uninstalled		HSCI	Normal		
1	8	SMUI	Main ok	Uninstalled	
Uninstalled		SIUI	Main ok		
1	9	NULL	Uninstalled	Uninstalled	
Uninstalled		HSCI	Normal		
1	10	MSGI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	11	MSGI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	12	MSGI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	13	MSGI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	14	BSGI	Main ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	15	BSGI	Standby ok	Uninstalled	
Uninstalled		NULL	Uninstalled		
1	16	ALUI	Normal	Uninstalled	
Uninstalled		NULL	Uninstalled		



1	17~18	UPWR	Normal	Uninstalled
Uninstalled	UPWR	Normal		
1	19~20	UPWR	Normal	Uninstalled
Uninstalled	UPWR	Normal		

# Board Report File

Table 6-5 shows the list of column headers required within the file:

Table 6-5 List of Column Headers

Header Name	Description
Frame number	ID of the frame.
Slot number	ID of the slot.
Location	Location of slot.
Board type	Type of board.
Module number	Module number
Assistant slot number	Assistant slot number

## Sample Board Report File

```
+++      LOCAL      2023-08-25 10:25:56-15:00 DST
O&M      #8435619
%%LST BRD:;%%
RETCODE = 0  Operation succeeded

Board information
-----
Frame number  Slot number  Location  Board type  Module number  Assistant
slot number

0              0          Front    IFMI        132
1
0              0          Back     BFII        Invalid
Invalid
0              1          Front    IFMI        132
0
0              1          Back     BFII        Invalid
Invalid
0              2          Front    FCCU        22
3
0              3          Front    FCCU        22
2
0              4          Front    FCCU        23
5
0              5          Front    FCCU        23
4
0              6          Front    SMUI        2
8
0              6          Back     SIUI        Invalid
Invalid
0              7          Back     HSCI        Invalid
Invalid
```

0	8	Front	SMUI	2
6				
0	8	Back	SIUI	Invalid
Invalid				
0	9	Back	HSCI	Invalid
Invalid				
0	10	Front	BSGI	136
11				
0	11	Front	BSGI	136
10				
0	12	Front	CDBI	102
13				
0	13	Front	CDBI	102
12				
0	14	Front	MSGI	211
15				
0	15	Front	MSGI	211
14				
0	16	Front	ALUI	Invalid
Invalid				
0	17	Front	UPWR	Invalid
Invalid				
0	17	Back	UPWR	Invalid
Invalid				
0	19	Front	UPWR	Invalid
Invalid				
0	19	Back	UPWR	Invalid
Invalid				
1	0	Front	CDBI	103
1				
1	1	Front	CDBI	103
0				
1	2	Front	FCCU	24
3				
1	3	Front	FCCU	24
2				
1	4	Front	FCCU	25
5				
1	5	Front	FCCU	25
4				
1	6	Front	SMUI	3
8				
1	6	Back	SIUI	Invalid
Invalid				
1	7	Back	HSCI	Invalid
Invalid				
1	8	Front	SMUI	3
6				
1	8	Back	SIUI	Invalid
Invalid				
1	9	Back	HSCI	Invalid
Invalid				
1	10	Front	MSGI	210
Invalid				
1	11	Front	MSGI	207
Invalid				

1	12	Front	MSGI	206	
Invalid					
1	13	Front	MSGI	209	
Invalid					
1	14	Front	BSGI	137	
15					
1	15	Front	BSGI	137	
14					
1	16	Front	ALUI	Invalid	
Invalid					
1	17	Front	UPWR	Invalid	
Invalid					
1	17	Back	UPWR	Invalid	
Invalid					
1	19	Front	UPWR	Invalid	
Invalid					
1	19	Back	UPWR	Invalid	Invalid

## Module Report File

[Table 6-6](#) shows the list of column headers required within the file:

**Table 6-6 List of Column Headers**

Header Name	Description
Module No.	Module number
Module type	Type of module
Software type	Type of software
Version	Version
Patch unit	Number of patch units
Unit in activated state	Number of units in activated state
Unit in confirmed state	Number of units in confirmed state

### Sample Module Report File

```
+++    LOCAL          2023-08-25 10:25:56-15:00 DST
O&M    #8435619
%%DSP PATCH:;%%
RETCODE = 0  Operation succeeded

Display patch information
-----
Module No.  Module type  Software type  Version
Patch unit  Unit in activated state  Unit in confirmed state

  2          Main       SMUI           V11           67
0                                     67
  3          Main       SMUI           V11           67
0                                     67
  4          Main       SMUI           V11           67
0                                     67
  5          Main       SMUI           V11           67
0                                     67
```

6	Main	FCCU	V11	67
0		67		
7	Main	FCCU	V11	67
0		67		
8	Main	FCCU	V11	67
0		67		
9	Main	SMUI	V300R010C05SPH172	
67	0		67	
10	Main	SMUI	V300R010C05SPH172	
67	0		67	
11	Main	SMUI	V300R010C05SPH172	
67	0		67	
22	Main	FCCU	V300R010C05SPH172	
343	0		343	
23	Main	FCCU	V300R010C05SPH172	
343	0		343	
24	Main	FCCU	V300R010C05SPH172	
343	0		343	
25	Main	FCCU	V300R010C05SPH172	
343	0		343	
26	Main	FCCU	V300R010C05SPH172	
343	0		343	
27	Main	FCCU	V300R010C05SPH172	
343	0		343	
28	Main	FCCU	V300R010C05SPH172	
343	0		343	
29	Main	FCCU	V300R010C05SPH172	
343	0		343	
30	Main	FCCU	V300R010C05SPH172	
343	0		343	
31	Main	FCCU	V300R010C05SPH172	
343	0		343	
32	Main	FCCU	V300R010C05SPH172	
343	0		343	
33	Main	FCCU	V300R010C05SPH172	
343	0		343	
34	Main	FCCU	V300R010C05SPH172	
343	0		343	
35	Main	FCCU	V300R010C05SPH172	
343	0		343	
36	Main	FCCU	V300R010C05SPH172	
343	0		343	
37	Main	FCCU	V300R010C05SPH172	
343	0		343	
38	Main	FCCU	V300R010C05SPH172	
343	0		343	
39	Main	FCCU	V300R010C05SPH172	
343	0		343	
40	Main	FCCU	V300R010C05SPH172	
343	0		343	
41	Main	FCCU	V300R010C05SPH172	
343	0		343	
42	Main	FCCU	V300R010C05SPH172	
343	0		343	
43	Main	FCCU	V300R010C05SPH172	
343	0		343	

44	Main	FCCU	V300R010C05SPH172
343	0		343
45	Main	FCCU	V300R010C05SPH172
343	0		343
46	Main	FCCU	V300R010C05SPH172
343	0		343
47	Main	FCCU	V300R010C05SPH172
343	0		343
48	Main	FCCU	V300R010C05SPH172
343	0		343
49	Main	FCCU	V300R010C05SPH172
343	0		343
50	Main	FCCU	V300R010C05SPH172
343	0		343
51	Main	FCCU	V300R010C05SPH172
343	0		343

# Port Report File

[Table 6-7](#) shows the list of column headers required within the file:

**Table 6-7 List of Column Headers**

Header Name	Description
Trunk No	Module number
Trunk group name	Type of module
IFM Module	Type of software
Local server port	Local server port
Remote URI	Remote URI <IP address>:<port>
Master/Slave type	Type primary/secondary
Second Remote URI	Second remote URI
Second Master/Slave type	Type primary/secondary
Techprefix	Tech prefix
IsDeleteTechprefix	True/false
Protocol Type	Type of protocol

## Sample Port Report File

```
+++ LOCAL 2023-08-25 10:25:56-15:00 DST
O&M #8435619
%%LST SIPIPPAIR:;%
RETCODE = 0 Operation succeeded

SIPTG ip pair information
-----
Trunk No Trunk group name IFM Module
Local server port Remote URI Master/Slave type Second Remote
URI Second Master/Slave type Techprefix IsDeleteTechprefix Protocol
Type

5 intx_SAMPLE_XX_sip 132
5060 500.100.10.100:5060 Master
```

255.255.255.255:65535	Slave	NULL	
True	UDP		
14	CTE_SAMPLE_XX_MTY_SIP		132
5060	500.100.10.101:5060	Master	
255.255.255.255:65535	Slave	NULL	
True	UDP		
15	CTE__SAMPLE__SIP		132
5060	500.100.10.102:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
16	ISF_SAMPLE__LOC		132
5060	500.100.10.103	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
20	CTE_SAMPLE____SIP		132
5060	500.100.10.104:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
21	CTE_SAMPLE____SIP		132
5060	500.100.10.105:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
22	P_AGNYSAMPLESIP		132
5060	500.100.10.106:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
23	PBA_IMS_ALU_2		132
5060	500.100.10.107	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
24	PBA_SAMPLE____SIP		132
5060	500.100.10.108	Master	
255.255.255.255:65535	Slave	NULL	
True	UDP		
33	PVR_SAMPLE____SIP		132
5060	500.100.10.109:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
34	SCOT_SAMPLE__SIP		132
5060	500.100.10.110:5060	Master	
255.255.255.255:65535	Slave	NULL	
True	UDP		
35	CTE_SAMPLE_SIP		132
5060	500.100.10.111:5060	Master	
255.255.255.255:65535	Slave	NULL	
True	UDP		
42	CTE_SAM_NOV1_SIP		132
5060	500.100.10.112:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
43	CTE_SAM_NOV2_SIP		132
5060	500.100.10.113:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
44	PVR_SAMPLETECAS_OUT_SIP		132
5060	500.100.10.114:5060	Master	

---

255.255.255.255:65535	Slave	NULL	
True	UDP		
46	CTE_SAMPLE_T_SIP		132
5060	500.100.10.115:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
47	CTE_SAMPLEL_SIP2		132
5060	500.100.10.116:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		
49	CUCMX_SAMPLE_SIP		132
5060	500.100.10.117:5060	Master	
255.255.255.255:65535	Slave	NULL	
False	UDP		

# About Design Studio Extension

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

## Adding a New Vendor

To model a device from a different vendor in IMS domain, the cartridge defines an action Discover IMS Network to handle discovery and modeling of data. To discover and model data of a new vendor, a new action can be created, or existing action can be extended, and existing processors can be imported for logical and physical discovery. Additional processors can also be created accordingly to collect and model data of the new vendor.

## Adding a New Device Type under Existing Vendors

To discover and model a new device for IMS Network, IMS Network Device Modeller implementation can be updated with logic to support modeling of the new device type. An additional change can be included in IMS Network File Parser to read data from files if required.

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