Oracle® Communications Network Integrity

CORBA Cartridge Guide

Release 7.3.2

E66039-01

May 2016

This guide explains the functionality and design of the Oracle Communications Network Integrity Cartridge for CORBA (CORBA cartridge).

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

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

- *Network Integrity Developer's Guide*: for an understanding of cartridges.
- *Network Integrity Installation Guide*: for information about deploying and undeploying cartridges.

This guide assumes that you are familiar with:

- Oracle Communications Design Studio Environment and its associated terminology.
- Common object request broker architecture (CORBA) standard and terminology.

Reviewing and Extending in Design Studio

You can download a ZIP file that contains the individual Design Studio files, and you can open these file in Design Studio to review and extend the cartridge.

See *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.

About the CORBA Cartridge

The CORBA cartridge enables CORBA connectivity to Network Integrity, allowing it to use the CORBA protocol to communicate with external systems.

CORBA is a recognized, open (non-vendor specific), and commonly used industry communication standard between systems and devices. This cartridge enables Network Integrity customers to quickly and reliably build deployable cartridges that interact with one or many CORBA systems.

The CORBA cartridge is an abstract cartridge, meaning that Design Studio is used to configure and assemble the run-time cartridge against target systems or devices before it is deployed to the Network Integrity server. Other cartridges can be extended to reference the object request broker (ORB) and NameServer objects produced by the CORBA cartridge to perform discovery, import or discrepancy resolution actions.

Using the CORBA cartridge as a foundation, reusing its functions, and adopting best practices from Oracle, greatly reduce the cost and time to implement base CORBA functions. In addition, you have more time to focus on system specifics or addressing business issues. The included ORB and functions have been extensively tested and documented to deliver rapid value in discovery, import and resolution cartridges.

About Cartridge Dependencies

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

Run-time Dependencies

In order for the CORBA cartridge to work at run time, the AddressHandler cartridge must be installed.

See the Design Studio Help for information about installing the AddressHandler cartridge.

Design-Time Dependencies

The CORBA cartridge has the following dependencies:

- NetworkIntegritySDK
- Address_Handlers

Opening the Cartridge Files in Design Studio

To review and extend the Oracle Communications Network Integrity CORBA cartridge, you must first download the Network Integrity CORBA cartridge software from the Oracle software delivery website:

https://edelivery.oracle.com

The software contains the Oracle Communications Network Integrity CORBA cartridge ZIP file, which has the following structure:

- **\Network_Integrity_Cartridge_Projects\Abstract_CORBA_Cartridge**

The project **Abstract_CORBA_Cartridge** contains the extensible Design Studio files.

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

See the Design Studio Help for information about building and deploying cartridges.

About the Cartridge Components

The CORBA cartridge is an abstract cartridge that is extended by other cartridges that require CORBA connectivity to discover, import, or resolve discrepancies.

This cartridge contains three actions:

- Discover Abstract CORBA
- Import Abstract CORBA

- Resolve Abstract CORBA

Each action is made up of the following processors, run in the following order:

1. Property Initializer
2. Connection Manager

Figure 1 illustrates the processor workflow of the Discover Abstract CORBA action.

Figure 1 Discover Abstract CORBA Action Processor Workflow

Figure 2 illustrates the processor workflow of the Import Abstract CORBA action.

Figure 2 Import Abstract CORBA Action Processor Workflow

Figure 3 illustrates the processor workflow of the Resolve Abstract CORBA action.

Figure 3 Resolve Abstract CORBA Action Processor Workflow

Property Initializer

The Property Initializer processor (CorbaPropertyInitializer) sets the properties needed to initialize CORBA connectivity, and writes them to the CorbaSeed (a JavaBean class). The following properties are set and written by the Property Initializer processor:

- CorbaLoc URL: Obtained from the Request object and used by the "Connection Manager" processor to create the NameServer (NamingContextEXT) object.
- ORB Command Line: Used to create the ORB, made up of command line arguments (string array), supplies the customization properties supplied by the user to initiating the ORB. The ORB command line is a single-space separated list.
- (Optional) OrbProperties: Used to create the ORB. These are customized properties (Datatype: java.util.Properties).
- (Optional) org.omg.CORBA.ORBClass: Used to create the ORB. This class is a Datatype: String.
- (Optional) org.omg.CORBA.ORBSingletonClass: Used to create the ORB. This class is a Datatype: String.
- Naming Service Connection Flag: A Boolean to indicate whether the processor should attempt to obtain the NameServer. Default is true. The output NameServer parameter is null if the flag is set to false.

Connection Manager

The Connection Manager processor (CorbaConnectionManager) takes the CorbaSeed produced by the Property Initializer processor and initiates it to establish a CORBA connection. It provides the ORB and NameServer object information to any extending cartridges.

This Connection Manager processor performs the following operations:

1. Initiates the ORB using the command line arguments and properties contained in the CorbaSeed.
2. Obtains the NameServer (namingContextExt) object from the CorbaLoc URL.
3. Returns the ORB and NameServer if successful.

The NameServer can be obtained in two ways:

- From the IORFile: The IORFile corresponding to the CORBA server is uploaded while creating the scan a configuration.
- From the CorbaLoc URL: The *n* CorbaLoc URLs corresponding to *n* CORBA servers identified in the scope are added into the scope tab while creating a scan configuration.

CORBA URL Address Validation

All cartridges that extend the AbstractCorbaDiscovery action must provide valid CorbaLOC URL addresses. The CorbaURLAddressHandler address handler validates the addresses entered on the Scope Address page in Network Integrity. The address handler validates that the given address is a properly formatted IPv4 or IPv6 CorbaLoc URL.

To address the problem of bootstrapping and allow for more convenient exchange of human-readable object references, ORB::string_to_object allows URLs in the CorbaLoc formats to be converted into object references. If conversion fails, string_to_object raises a BAD_PARAM exception with one of following standard minor codes, as appropriate.

The CorbaLoc URL scheme provides stringified object references that are more easily manipulated by users than IOR URLs. Currently, CorbaLoc URLs denote objects that

can be contacted only by transport protocols like IIOP or resolve_initial_references (RIR).

Examples of IIOP and RIR based CorbaLoc are as follows:

```
CorbaLoc::555xyz.com/Prod/TradingService
CorbaLoc:iiop:1.1@555xyz.com/Prod/TradingService
CorbaLoc::555xyz.com,:556xyz.com:80/Dev/NameService
CorbaLoc:rir:/TradingService
CorbaLoc:rir:/NameService
CorbaLoc:iiop:192.168.14.25:555/NameService
CorbaLoc::[1080::8:800:200C:417A]:88/DefaultEventChannel
```

Refer to *CORBA Interoperability Specification* for more information about IIOP and RIR formats.

About Design Studio Construction

This section outlines the Design Studio construction for each of cartridge action and its associated processors:

- CORBA Discovery Action
- CORBA Import Action
- CORBA Discrepancy Resolution Action

CORBA Discovery Action

Table 1 outlines the Design Studio construction of the Discover Abstract CORBA action.

Table 1 Discover Abstract CORBA Action Design Studio Construction

Action	Result Category	Address Handler	Scan Parameters	Model	Processors
Discover Abstract CORBA	Device	CorbaURLAddressHandler	N/A	N/A	▪ CORBA Property Initializer ▪ CORBA Connection Manager

Table 2 outlines the Design Studio construction of the processors that belong to the Discover Abstract CORBA action.

Table 2 Discover Abstract CORBA Action Processors Design Studio Construction

Processor	Variable	Notes
CORBA Property Initializer	Input: n/a Output: CorbaSeed	**CorbaSeed.java** is a JavaBean class that contains the ORB properties and arguments.
CORBA Connection Manager	Input: CorbaSeed Output: ORB, namingContextEXT	The ORB is a reference of the local ORB initialized. The namingContextExt is a reference of namingContextExt which is the initial naming context or naming server.

CORBA Import Action

Table 3 outlines the Design Studio construction of the Import Abstract CORBA action.

Table 3 Import Abstract CORBA Action Design Studio Construction

Action	Result Category	Address Handler	Scan Parameters	Model	Processors
Import Abstract CORBA	Device	N/A	N/A	N/A	▪ CORBA Property Initializer ▪ CORBA Connection Manager

Table 4 outlines the Design Studio construction of the processors that belong to the Import Abstract CORBA action.

Table 4 Import Abstract CORBA Action Processors Design Studio Construction

Processor	Variable	Notes
CORBA Property Initializer	Input: n/a Output: CorbaSeed	**CorbaSeed.java** is a JavaBean class that contains the ORB properties and arguments.
CORBA Connection Manager	Input: CorbaSeed Output: ORB, namingContextEXT	The ORB is a reference of the local ORB initialized. The namingContextExt is a reference of namingContextExt which is the initial naming context or naming server.

CORBA Discrepancy Resolution Action

Table 5 outlines the Design Studio construction of the Resolve Abstract CORBA action.

Table 5 Resolve Abstract CORBA Action Design Studio Construction

Action	Result Category	Address Handler	Scan Parameters	Model	Processors
Resolve Abstract CORBA	Device	N/A	N/A	N/A	▪ CORBA Property Initializer ▪ CORBA Connection Manager

Table 6 outlines the Design Studio construction of the processors that belong to the Resolve Abstract CORBA action.

Table 6 Resolve Abstract CORBA Action Processors Design Studio Construction

Processor	Variable	Notes
CORBA Property Initializer	Input: N/A Output: CorbaSeed	**CorbaSeed.java** is a JavaBean class that contains the ORB properties and arguments.
CORBA Connection Manager	Input: CorbaSeed Output: ORB, namingContextEXT	The ORB is a reference of the local ORB initialized. The namingContextExt is a reference of namingContextExt which is the initial naming context or naming server.

About Design Studio Extension

This section gives examples of how to extend the CORBA cartridge. See *Network Integrity Concepts* for guidelines and best practices for extending cartridges.

Creating a CORBA Discovery Action that Extends the CORBA Cartridge

This example explains the high level steps to create a discovery action that extends the Discover Abstract CORBA action and models the collected data into the Oracle Communications Information Model using the CORBA interface.

This example assumes that the CORBA cartridge is loaded in Design Studio and is building without errors.

1. Open Design Studio in the design perspective.
2. Create a new cartridge project.
3. Create a new discovery action.
4. Add the CORBA cartridge project as a dependency to your new cartridge project.
5. Add the Discover Abstract CORBA action to the new discovery action.
6. Change to the Java perspective.
7. Compile the IDL into Java and package it into a JAR file.
8. Copy the JAR file to the **lib** directory of the new cartridge project.
9. Right-click the cartridge project and select **Properties**.

   The Properties for *Cartridge_name* dialog box appears.
10. Click **Java Build Path**.
11. Click the **Libraries** tab.
12. Add the new JAR file to the library.
13. Add a new discovery processor to the discovery action. Make this processor responsible for calling the appropriate CORBA interfaces needed to collect the data.
14. Make the new discovery processor use the ORB and Naming Service (NamingContextExt) objects that are output by the Connection Manager processor as input context parameters.
15. Design an implementation for the new discovery processor that runs one or more CORBA calls to retrieve the intended data.
16. Clean, build, and deploy the cartridge project to Network Integrity.

    The cartridge project creates a cartridge (an IAR file) that can be deployed. A scan can be configured and executed using the produced cartridge.

Initializing a Custom ORB

To support an ORB that does not come with the JDK on Network Integrity, you must extend the CORBA cartridge. The steps below represent an example procedure that uses **JacORB**.

To initialize a custom ORB:

1. Open Design Studio in the design perspective.

2. Create a new cartridge project.
3. Create a new discovery action.
4. Add the CORBA cartridge project as a dependency to your new cartridge project.
5. Add the Discover Abstract CORBA action to the new discovery action.
6. Create a new processor called **JacORB Property Initializer**.
7. Insert the JacORB Property Initializer processor between the Corba Property Initializer processor and the Corba Connection Manager processor.

   This ensures that the output parameter from the preceding processor is used as input for the subsequent processor in the chain.
8. Copy the **JacORB** JAR files to the **lib** directory of the new cartridge project and add them to the buildpath of that project.

   The *Jac ORB Property Initializer* processor sets the properties below to the corbaSeed JavaBean class.

   - **org.omg.CORBA.ORBClass = org.jacorb.orb.ORB**
   - **org.omg.CORBA.ORBSingletonClass = org.jacorb.orb.ORBSingleton**

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
E66039-01	May 2016	Initial release.

Oracle Communications Network Integrity CORBA Cartridge Guide, Release 7.3.2
E66039-01