

Oracle® Retail Integration Cloud Service

Integration Gateway Services Guide

Release 19.1.000

F31984-01

August 2020

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 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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via**[™] licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex**[™] licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR

Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments	vii
Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	ix
Customer Support	x
Improved Process for Oracle Retail Documentation Corrections	x
Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)	x
Conventions	xi
1 Introduction	
Prerequisites	1-1
Prepare Oracle WebLogic 12.2.1.4.0 Server	1-1
Create the RIB IGS WebLogic 12.2.1.4.0 Managed Server	1-2
2 Install Integration Gateway Services	
Run the IGS under RIB_HOME	2-1
Verify the IGS Application Installation Using the Administration Console	2-2
3 Customize Integration Gateway Services	
Prerequisites	3-1
RSE and AG-Overview and Relationship	3-2
PL/SQL Provider	3-2
JavaEE Provider	3-3
Artifact Generator Inputs and Outputs	3-3
RSE Inputs and Outputs	3-3
Extend an Existing IGS Web Service Provider	3-4
Use Case Scenario	3-4
Approach	3-4
Artifact Generator Work	3-4
Prepare RSE	3-5
Deploy the Customized IGS	3-6
Test the Customized IGS	3-7

Add a New Service to IGS	3-7
Prepare RSE.....	3-7
Deploy the Customized IGS	3-9
Test the Customized IGS.....	3-9

4 IGS and Clustering (High Availability)

Thread Value	4-1
Routing Information.....	4-1

5 Secure Integration Gateway Services

Send Us Your Comments

Oracle Retail Integration Cloud Service Integration Gateway Services Guide, Release 19.1.000.

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.

Preface

The purpose of this document is to introduce you to Integration Gateway Services (IGS), a component that provides an integration infrastructure for external (third party) connectivity to the Oracle Retail Integration Bus (RIB).

Audience

This guide is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementation staff

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.

Related Documents

For more information, see the following documents in the Oracle Retail Integration Bus documentation set:

- *Oracle Retail Integration Bus Implementation Guide*
- *Oracle Retail Integration Bus Installation Guide*
- *Oracle Retail Integration Bus Operations Guide*
- *Oracle Retail Integration Bus Release Notes*
- *Oracle Retail Integration Bus Hospital Administration Guide*
- *Oracle Retail Integration Bus Security Guide*
- *Oracle Retail Integration Bus Support Tools Guide*

- *Oracle Retail Integration Bus Java Messaging Service (JMS) Console Guide*
- *Oracle Retail Enterprise Integration Guide*
- *Oracle Retail Functional Artifacts Guide*
- *Oracle Retail Functional Artifact Generator Guide*
- *Oracle Retail Service-Oriented Architecture Enabler Tool Guide*
- *Oracle Retail Integration Bus Data Model*
- *Oracle Retail Payload Mapper Guide*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Help Center

(docs.oracle.com)

Oracle Retail product documentation is also available on the following Web site:

<https://docs.oracle.com/en/industries/retail/index.html>

(Data Model documents can be obtained through My Oracle Support.)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	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.

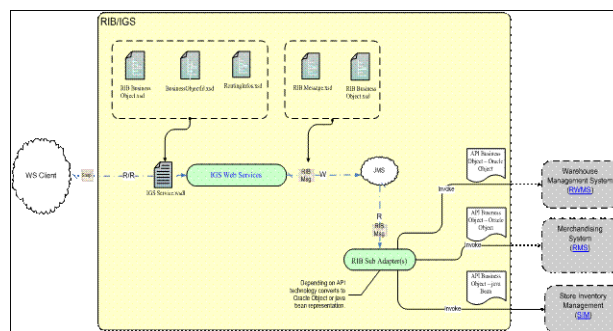
Introduction

The Integration Gateway Services (IGS) component provides an integration infrastructure for external (third party) connectivity to the Oracle Retail Integration Bus (RIB) in the form of a tested set of Web service providers and the configurations to connect to RIB and to ease the integration to RIB interfaces and RIB payloads.

The IGS component provides the integration to these RIB interfaces through standard request/response Web services using only the standard XSD based RIB message payloads.

There are 19 RIB message family interfaces included in IGS. They are selected as the interfaces most commonly used for custom integration to legacy systems. A Web service corresponds to each of the selected message families. Each service exposes the message types supported by the RIB message family.

The RIB Integration Gateway Services (IGS) component is an optional sub system and should be installed only after the core RIB components have been installed and verified. The following diagram describes the IGS logical architecture.



Prerequisites

The RIB Integration Gateway Service (IGS) component requires Oracle WebLogic Server 12c Release 12.2.1.4.0 and Java 8 64 bit with latest security updates.

Before you install this component, read the *Oracle Retail Integration Bus Implementation Guide* for the considerations and planning steps needed for the RIB IGS deployment to the WebLogic 12.2.1.4.0 Server. You should also make sure the environment variable JAVA-HOME points to Java 8 64 bit with latest security updates.

Prepare Oracle WebLogic 12.2.1.4.0 Server

The installation and base configuration of the Oracle WebLogic 12.2.1.4.0 Server is beyond the scope of this document. Work with the Oracle WebLogic 12.2.1.4.0 Server

administration team to determine the physical and logical placement of the RIB IGS component within the WebLogic 12.2.1.4.0 Server deployment.

Create the RIB IGS WebLogic 12.2.1.4.0 Managed Server

Take the following step to prepare the Oracle WebLogic 12.2.1.4.0 Server to install the `igs-service-<version>.ear`:

Create a new managed server in which to deploy `igs-service-<version>.ear`. Oracle recommends (but does not require) that the `.ear` file name be used (without the extension) along with an underscore.

For example, if the `.ear` file name were `igs-service-<version>.ear`, the instance name should be `igs-service_server`.

Add the server start argument for IGS managed server from WLS console where managed server is configured and update the `startWeblogic.sh` and `startManagedWeblogic.sh` script under `$Domain_Home/bin` with below argument.

```
JAVA_OPTIONS="-Doracle.retail.soa.enabler.service.provider.engine.  
ServiceProviderImplLookupFactory.interceptor=com.oracle.retail.igs.integration.  
service.DynamicServiceProviderImpl
```

Install Integration Gateway Services

This chapter discusses steps to install IGS under RIB_HOME (rib-home/tools-home/integration-bus-gateway-services).

Run the IGS under RIB_HOME

Take the following steps to run IGS under RIB_HOME:

1. Download the IntegrationGatewayService<version>ForAll<version>Apps_eng_ga.tar and untar it under rib-home/tools-home.

```
cd rib-home/tools-home/  
tar -xvf IntegrationGatewayService<version>ForAll<version>Apps_eng_ga.tar
```

2. Go to rib-home/tools-home/integration-bus-gateway-services/conf and edit the IgsConfig.properties as follows:
 - a. Change the value of WlsUrl to point to the WebLogic 12.2.1.4.0 server where IGS is going to be deployed. The port in the WlsUrl should be the AdminServer port.
 - b. Change the value of WlsTarget to the managed server name where IGS is going to be deployed (for example, igs-service_server).
3. Go to rib-home/tools-home/integration-bus-gateway-services/bin. Run the igs-install.sh.
4. Restart the WebLogic managed server.
5. After installation, restrict access to the integration-bus-gateway-services directory with this command:

```
chmod -R 700 .
```

All of the items in Step 3 also can be performed separately, as follows.

1. Go to rib-home/tools-home/integration-bus-gateway-services/bin. Run the igs-admin.sh -setup-igs to set up the environment. Running this script verifies whether the attempted IGS installation is from within the rib-home or in standalone mode; the pre-configuration cleanup is based on this mode.

```
sh igs-admin.sh -setup-igs
```

2. Go to rib-home/tools-home/integration-bus-gateway-services/bin. Run the igs-admin.sh -setup-security-credential to set up the WebLogic username and password information in a secure credential store.

```
sh igs-admin.sh -setup-security-credential
```

3. Go to `IGS_HOME/integration-bus-gateway-services/bin`. Run the `igs-admin.sh -prepare` to prepare the `igs-service-<version>.ear`, based on the number of channels and configured AQ JMS.

```
sh igs-admin.sh -prepare
```

4. Go to `rib-home/tools-home/integration-bus-gateway-services/bin`. Run the `igs-admin.sh -configure` to configure the WebLogic 12.2.1.4.0 server with the AQ JMS server information listed in the `rib-deployment-env-info.xml`.

```
sh igs-admin.sh -configure
```

5. Go to `rib-home/tools-home/integration-bus-gateway-services/bin`. Run the `igs-admin.sh -deploy` to deploy the `igs-service-<version>.ear` to the WebLogic 12.2.1.4.0 server.

```
sh igs-admin.sh -deploy
```

6. Restart the WebLogic managed server.

7. Restrict access to the IGS folder with a command like:

```
cd $RIB-HOME/tools-home/integration-bus-gateway-services
```

```
chmod -R 700 .
```

8. If the `igs-service-<version>.ear` must be undeployed, run the `rib-home/tools-home/integration-bus-gateway-services/bin/igs-admin.sh -undeploy` to undeploy an `igs-service-<version>.ear`.

```
sh igs-admin.sh -undeploy
```

Note: The log files are located here: `RIB-HOME/tools-home/integration-bus-gateway-services/log`

If any changes are made to the `rib-deployment-env-info.xml` or the `rib-<app>-adapters.xml`, the `-prepare`, `-configure`, and `-deploy` steps, must be executed.

Verify the IGS Application Installation Using the Administration Console

Take the following steps to verify the IGS installations using the Oracle WebLogic 12.2.1.4.0 Administration Console:

Note: The Test Client link is visible when the server is in Development mode.

1. Navigate to the Deployments page.
2. On the Summary of Deployments page, locate the `igs-service`.
3. To expand the tree, click the + beside the `igs-service`.
4. Locate the Web Services section.
5. Click any Web service (for example, `ASNInPublishingService`) to navigate to the settings for `ASNInPublishingService` page.
6. Select the **Testing** tab.

7. To expand the tree, click the + beside the service name.
8. Locate the Test Client link. Navigate to the WebLogic 12.2.1.4.0 Test Client page.
9. Select the Ping operation. Enter test data in the string arg0: text box. Click **Ping**.
10. The test page includes the request message and the response message.

Customize Integration Gateway Services

This chapter discusses how to use the Oracle Retail SOA Enabler (RSE) and Artifact Generator (AG) tools to customize an Integration Gateway Services (IGS) service. Customizing an IGS service also means customizing the API for RIB and the subscribing application. While this guide focuses only on the IGS end of the process, all activities must be completed for a working message flow.

Prerequisites

For a successful customization, it is important to understand the tools and approaches described in each of the referenced documents. The process is complex and requires knowledge of tools as well as the end-point Oracle Retail application, RIB, and IGS products.

IGS customization requires a number of considerations and planning steps. Planning helps prevent reinstallation or reverting the RIB and IGS due to operational or performance problems.

The following helps to ensure a successful customization of the RIB and the IGS:

- A functional RIB environment without customizations.
- Familiarity with core RIB concepts, components, and architecture, including an understanding of all the following:
 - Oracle database 12c (12 x) triggers, RIB adapters, RIB Message envelope, RIB message payloads, and the functionality of GETNEXT () and CONSUME () stored procedures.
 - The integration message flow paths.
 - The RIB life cycle.
 - The physical and logical requirements and limitations of the RIB components.
 - RIB operational considerations.

The tools used in the customization and extensions of the RIB are documented separately. The primary tools are the Oracle Retail Functional Artifact Generator and the Oracle Retail SOA Enabler.

This guide assumes that you have created the development environment for IGS, and that the RIB installation (rib-home) has working versions of the RSE and Artifact Generator tools.

Note: See the *Oracle Retail Integration Bus Installation Guide*, the *Oracle Retail Integration Bus Operations Guide*, the *Oracle Retail Service-Oriented Architecture Enabler Tool Guide*, and the *Oracle Retail Functional Artifact Generator Guide* for installation and configuration instructions.

RSE and AG-Overview and Relationship

The RSE and AG are companion tools, particularly in the creation and customization of Web service providers. Generally available documents exist for each tool to describe installation, configuration, and use; they include extensive examples. These documents, combined with the *Oracle Retail Integration Bus Implementation Guide*, cover the recommended practices and provide examples for customizing business objects. With each release, Quality Assurance testing is completed on the tools and associated documentation, including each example cited.

The RSE and AG tools work only with Oracle Retail Business Objects, release 13.1.x (and higher). They do not work with objects from earlier releases (for example, 13.0.x and 12.x). This restriction is due to the uptake of Business Object standards that support versioning, and customization and extension.

The tools enforce this separation and will not permit anything except the best practices documented in the Oracle Retail Artifact Standards. This rule helps to preserve changes when a new release is issued. When applying business object upgrades, customers can easily preserve changes by simply applying them over the new base objects.

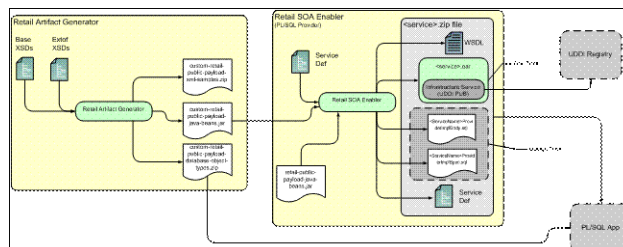
The AG is designed to allow customers to create extensions to base objects using the specified hooks, or to create entirely custom objects that meet Oracle Retail standards. The tools enforce these standards.

Note: See the *Oracle Retail Functional Artifact Guide* for information on standards.

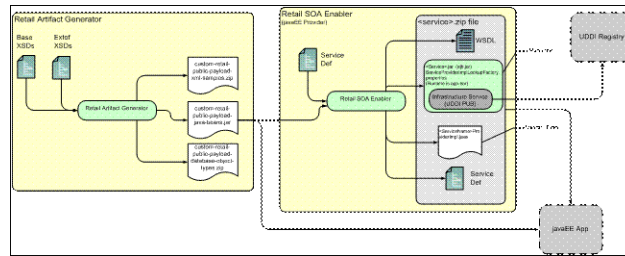
For the creation of a Web service provider, the RSE tool expects as one of its inputs one of the outputs of the AG tool to design time artifacts. There is a runtime relationship as well, as shown in the diagrams below, which illustrates the relationships for a Java EE and PL/SQL Web Service Provider.

Note: RSE supports Java EE and PL/SQL. This document uses RSE only for the Java side. The Artifact Generator creates all artifacts for both PL/SQL and Java EE.

PL/SQL Provider



JavaEE Provider



Artifact Generator Inputs and Outputs

The inputs to the AG are the base GA business objects, and the custom or extended base objects. The GA base objects are installed as part of the AG installation and configuration. The creation of extensions or custom XSDs is covered in the *Oracle Retail Functional Artifact Generator Guide*, including:

- Base XSDs and Extof XSDs
- Custom XSDs

The following are AG outputs:

- custom-retail-public-payload-java-beans-<version>.jar
- custom-retail-public-payload-database-object-types-<version>.zip

RSE Inputs and Outputs

The input for creating a service provider is a Service Definition Library XML file containing details about the Web services that must be created, the generally available business objects, and the output of the AG for extensions or custom XSDs.

For example:

The RSE tool references JAXB created java beans based on the BO source schema XSDs. These beans are contained in the retail-public-payload-java-beans-<version>.jar.

The RSE tool uses Oracle Retail BOs from retail-public-payload-java-beans-<version>.jar and custom BOs from custom-retail-public-payload-java-beans-<version>.jar.

The retail-public-payload-java-beans-<version>.jar file is created using the Retail Artifact Generator from the source BO XSDs. The retail-public-payload-java-beans-<version>.jar file is packaged in RSE. It does not need to be updated as part of the customization process.

The retail-public-payload-java-beans-<version>.jar file also contains the source XSDs themselves, which will be used, along with the XSDs in the optional custom-retail-public-payload-java-beans-<version>.jar, by the deployed service to validate all requests and responses against.

Outputs include the following:

- The deployable .ear with the WSDL conforming to the objects referenced in the ServiceProviderDefLibrary.xml.
- The PL/SQL or Java EE API templates that include hooks for talking to the services. These API templates contain placeholders for business logic and references to the Business Objects created by the AG.

- A copy of the ServiceProviderDefLibrary.xml.
- A static version of the service WSDL.
- The decorator services.

Extend an Existing IGS Web Service Provider

The most common use case in customization is to add new elements to existing messages. This section covers that use case. The *Oracle Retail Integration Bus Implementation Guide* covers the same use case with examples for the Oracle Retail Integration Bus (RIB).

The *Oracle Retail Service-Oriented Architecture Enabler Tool Guide* and the *Oracle Retail Functional Artifact Generator Guide* contain detailed examples that are not repeated here. However, they are referred to as needed in the solution approach sections. The IGS component is constructed using the AG and RSE tools, so customization of IGS is a straightforward extension of that process.

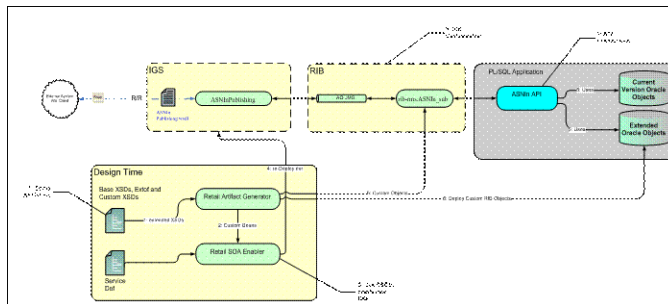
Use Case Scenario

The business has determined that the IGS ASNIn Service must include a new element, Fiscal ID. The Oracle Retail deployment has Oracle Retail RMS as the end-point (Subscriber) and a third party system as the source (Publisher).

The business objects for the IGS Web services are packaged and shipped (as generally available) with the Oracle Retail base objects in RetailFuncArtifact<version>ForAll<version>Apps_eng_ga.tar.

Approach

The following diagram describes the approach.



Artifact Generator Work

The following set of steps assumes installation and configuration of the AG and the RSE in the RIB installed rib-home/tools locations and that the file locations are relative to that installation. For alternative installations and locations, see the *Oracle Retail Artifact Generator Guide* and make adjustments as needed to these steps.

1. Using the Artifact Generator tool, extend the XSDs and generate the custom database 12c (12 x) objects and the custom-beans. Because this is a base object, it will have been installed with the AG tool.

Note: See "Addition of Optional Elements" in Chapter 3 of the *Oracle Retail Artifact Generator Guide*.

2. Locate the correct ExtOfASNInDesc.xsd file.

```
find input-xsd -name ExtOfASNInDesc.xsd
```

3. Edit the ExtOfASNInDesc.xsd file to add the following:

```
<xs:element name="ExtOfASNInDesc">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="fiscal_id" type="varchar23">
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="varchar23">
  <xs:restriction base="xs:string">
    <xs:maxLength value="5"/>
  </xs:restriction>
</xs:simpleType>
```

4. Create the Java Payload and OO.

```
Type java -classpath ./integration-lib/*: groovy.ui.GroovyMain
com/oracle/retail/integration/artifact/generator/GenArtifacts.groovy -g
generateCustom
```

5. For the IGS effort, the focus is on using the custom-retail-public-payload-java-beans-<version>.jar in the IGS application.

Note: The custom RIB objects are used in the application and RIB. See the *Oracle Retail Integration Bus Implementation Guide* for more information on customization of RIB. Both RIB and IGS must be customized with the same custom-retail-public-payload-java-beans-<version>.jar file for a working message flow.

Prepare RSE

Take the following steps to prepare the RSE tool:

1. Merge the IGS GA business objects with the new AG generated custom business objects and regenerate the IGS ear file.
 - a. Create a customization location and sub-folders (for example, /workarea/igs-customization).


```
Create /workarea/igs-customization/jar_update_area
Create /workarea/igs-customization/input_file
Create /workarea/igs-customization/output_file
```
 - b. Extract the needed files from the igs-service-<version>.ear file.


```
jar xvf igs-service-<version>.ear
Copy lib/ igs-services-impl-<version>.jar to
/workarea/igs-customization/input_file.
Copy payload-lib/custom-igs-bo-<version>.jar to
/workarea/igs-customization/input_file folder.
```
 - c. Extract the needed files from the igs-service-ejb.jar file.

```
jar xvf igs-service-ejb.jar
```

Copy META-INF/com.oracle.retail.soa.enabler.ServiceDef.xml to
/workarea/igs-customization/input_file.

- d. Copy custom-retail-public-payload-java-beans-<version>.jar generated by the Functional Artifact Generator, to /workarea/igs-customization/jar_update_area.
 - e. Change directory to /workarea/igs-customization/jar_update_area.
 - f. Extract the contents of custom-igs-bo-<version>.jar

```
jar -xf ../input_file/custom-igs-bo-<version>.jar
```
 - g. Update the custom-retail-public-payload-java-beans-<version>.jar file with the contents of custom-igs-bo-<version>.jar

```
jar -ufM custom-retail-public-payload-java-beans-<version>.jar co* ME*
```
2. Merge the IGS GA business objects with the new AG generated custom business objects and regenerate the IGS ear file.
 - a. Go to RSE Service Provider tab and choose the following options:
Web Service Type = SOAP
Service Provider Type = JavaEE
Service def library XML =
/workarea/igs-customization/input_file/com.oracle.retail.soa.enabler.ServiceDef.xml
Custom BO jar =
/workarea/igs-customization/jar_update_area/custom-retail-public-payload-java-beans-<version>.jar
Service impl jar =
/workarea/igs-customization/input_file/igs-services-impl-<version>.jar
 - b. Do not provide a value for localization BO jar.
 - c. Click **Generate**.
 - d. Save the output igs_JavaEEServiceProvider.zip to
/workarea/igs-customization/output_file/
 - e. Extract igs_JavaEEServiceProvider.zip in current directory (output_file).
Go to output_file directory
unzip igs_JavaEEServiceProvider.zip -d igs_JavaEEServiceProvider

Deploy the Customized IGS

Take the following steps to deploy the igs-service-<version>.ear from rib-home:

1. Copy /workarea/igs-customization/output_file/igs_JavaEEServiceProvider/igs-service-<version>.ear to the rib-home/tools-home/integration-bus-gateway-services/ear folder.
2. Change directory to rib-home/tools-home/integration-bus-gateway-services/bin.
3. Run igs installer: igs-install.sh.

4. Restart the WebLogic managed server.
5. Restrict access to the IGS folder with a command like:


```
cd $RIB-HOME/tools-home/integration-bus-gateway-services
chmod -R 700 .
```

Note: Make sure the WebLogic 12.2.1.4.0 Server is up and running before attempting to install igs-service-<version>.ear.

Test the Customized IGS

The following steps are covered in detail in other RIB documentation.

Note: The use of Hermes JMS is covered in the My Oracle Support document, *How to Connect Hermes JMS to Oracle AQ* (ID 837999.1).

Take the following steps to test the customized IGS:

1. Examine the new customized ASNIn WSDL.
2. Start HermesJMS and browse AQ JMS topic etASNIn.
3. Use SOAP user interface to publish a customized ASNInDesc message.
4. View the customized message in AQ using Hermes.

Add a New Service to IGS

This section discusses the procedure to add an entirely new Web service to the IGS. This guide assumes that the family for which a new Web service is to be created is already available in the RIB systems, and edge applications can subscribe the messages from those family topics.

Consider the scenario, business decided to add a new service for the family "CoReturn" to IGS, which is not currently available.

Prepare RSE

Take the following steps to prepare the RSE tool:

1. Create a customization location and sub-folders (for example, /workarea/igs-customization).
 - Create /workarea/igs-customization/input_file.
2. Extract the needed files from the igs-service-<version>.ear file. This file is located in the "ear" folder of the IGS archive.
 - ```
jar xvf igs-service-<version>.ear
```

```
Copy lib/ igs-services-impl-<version>.jar to
/workarea/igs-customization/input_file.
```
3. Copy payload-lib/custom-igs-bo-<version>.jar to /workarea/igs-customization/input\_file.
4. Edit the serviceDef file to add new service for the family "CoReturn".

- `jar xvf igs-service-ejb-<version>.jar`
- Copy META-INF/com.oracle.retail.soa.enabler.ServiceDef.xml to /workarea/igs-customization/input\_file.
- Edit the copied ServiceDef.xml file to add new service. In this use case for the family "CoReturn" following service definition to be added.

```

<service name="CoReturnPublishing">
 <documentation>An external system will publish CoSale onto the RIB
system. RMS will subscribe to CoReturn information and updates the database
tables depending upon the validity of the records enclosed within the
message.
 Topic Name : etCOReturn
 Subscribers : COReturn_sub (RMS).
 </documentation>
 <operation name="publishCustRetSaleCreateUsing">
 <documentation> This Operation invokes the publishing of
"Create" event for the CoSale.</documentation>
 <input type="CustRetSaleDesc">
 <documentation>
 CustRetSaleDesc RBO complies with the XML schema
CustRetSaleDesc.xsd, for more information on RBO structure and semantics
please refer to the associated XSD.
 </documentation>
 <header headerName="businessObjectId"
type="com.oracle.retail.integration.bus.gateway.services.businessobjectid.v
1.BusinessObjectId" headerRequired="false"/>
 </input>
 <output type="ServiceOpStatus" custom="true">
 <documentation>
 ServiceOpStatus object represents the acknowledgment from
the service. It encloses the information about the Successful or Failed
status of the publish event.
 </documentation>
 </output>
 <fault faultType="PublishingWSFaultException">
 <documentation>Throw this exception when a "soap:Server"
side publish problem occurs.</documentation>
 </fault>
 <fault faultType="IllegalStateWSFaultException" >
 <documentation>Throw this exception when an unknown
"soap:Server" side problem occurs.</documentation>
 </fault>
 </operation>
 </service>

```

5. Extract the ServiceConfig.properties file from igs-services-impl-<version>.jar  
`jar -xvf igs-services-impl-<version>.jar ServiceConfig.properties`
6. Edit ServiceConfig.properties file located in igs-services-impl-<version>.jar to add jms topic for the new service to publish messages. The topic name corresponding to a family can be found at rib-integration-flows.xml file. This xml is available with rib functional artifact deployment. The link could be as, <http://host:port/rib-func-artifact/integration/rib-integration-flows.xml>. In the case of the COReturn service, the topic is etCOReturn.
7. Update the igs-services-impl-<version>.jar with updated ServiceConfig.properties.  
`jar -ufm igs-services-impl-<version>.jar ServiceConfig.properties.`

8. Generate javaEEServiceProvider archives using updated serviceDef file. Go to deployed RSE tool. Select RSE Service Provider tab and choose the following options:
  - Web Service Type = SOAP
  - Service Provider Type = JavaEE.
  - Service def library XML =  
/workarea/igs-customization/input\_file/com.oracle.retail.soa.enabler.ServiceDef.xml
  - Custom BO jar =  
/workarea/igs-customization/input\_file/custom-igs-bo-<version>.jar
  - Service impl jar =  
/workarea /igs-customization/input\_file/igs-services-impl-<version>.jar
9. Do not provide a value for localization BO jar.
10. Click **Generate**.
11. Save the output igs\_JavaEEServiceProvider.zip to  
/workarea /igs-customization/output\_file/
12. Extract igs\_JavaEEServiceProvider.zip in current directory (output\_file).
  - Go to output\_file directory.
  - unzip igs\_JavaEEServiceProvider.zip -d igs\_JavaEEServiceProvider

## Deploy the Customized IGS

Take the following steps to deploy igs-service-<version>.ear from rib-home:

1. Copy /workarea/igs-customization/output\_file/igs\_JavaEEServiceProvider/igs-service-<version>.ear to the rib-home/tools-home/integration-bus-gateway-services/ear folder.
2. Change directory to rib-home/tools-home/integration-bus-gateway-services/bin.
3. Run igs installer: igs-install.sh.
4. Restart the WebLogic managed server.
5. Restrict access to the IGS folder with a command like:  
cd \$RIB-HOME/tools-home/integration-bus-gateway-services  
chmod -R 700 .

---



---

**Note:** Make sure WebLogic 12.2.1.4.0 Server is up and running before attempting to install igs-service-<version>.ear.

---



---

## Test the Customized IGS

The following steps are covered in detail in other RIB documentation.

---



---

**Note:** The use of Hermes JMS is covered in the My Oracle Support document, *How to Connect Hermes JMS to Oracle AQ* (ID 837999.1).

---



---

Take the following steps to test the customized IGS:

1. Examine the new COReturn WSDL.
2. Start HermesJMS and browse AQ JMS topic etCOReturn.
3. Use SOAP user interface to publish a COReturnDesc message.
4. View the message in AQ using Hermes.

---

## IGS and Clustering (High Availability)

Integration Gateway Services are stateless and hence can be deployed in clustered environments.

### Thread Value

The thread value is managed by IGS. The max channel number is configured in ServiceConfig.properties located in igs-services-impl-<version>.jar, and is updated by the igs-install.sh script, or the igs-admin.sh -prepare script.

### Routing Information

IGS services should be called with routing information in the Soap header. RIB uses this information along with the business object Id to route the service calls to various destinations. Here is an example of an IGS SOAP message with Routing Information in the header. The Routing Information snippet is in bold letters.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://www.oracle.com/retail/integration/bus/gateway/services/BusinessObjectId/v1"
xmlns:v11="http://www.oracle.com/retail/integration/bus/gateway/services/RoutingInfos/v1"
xmlns:v12="http://www.oracle.com/retail/igs/integration/services/ASNOutPublishingService/v1"

xmlns:v13="http://www.oracle.com/retail/integration/base/bo/ASNOutDesc/v1">
 <soapenv:Header>
 v1:BusinessObjectId>98766v1:BusinessObjectId>
 v11:RoutingInfos <!--Zero or more repetitions: -->
 v11:routingInfo>
 name>from_phys_loc
 value>1
 v11:detail>
 v11:dtl_name>from_phys_loc_typev11:dtl_name>
 v11:dtl_value>wv11:dtl_value>
```

```
 </v11:detail>
 </v11:routingInfo>
<v11:routingInfo>
 <name>to_phys_loc</name>
 <value>s_loc_1</value>
 <v11:detail>
 <v11:dtl_name>to_phys_loc_type</v11:dtl_name>
 <v11:dtl_value>s</v11:dtl_value>
 </v11:detail>
</v11:routingInfo>
<v11:routingInfo>
 <name>facility_type</name>
 <value>PROD</value>
</v11:routingInfo>
</v11:RoutingInfos>
</soapenv:Header>
<soapenv:Body>
 <v12:publishASNOutCreateUsingASNOutDesc> <!--Optional: -->
 <v13:ASNOutDesc>
 <v13:schedule_nbr>8</v13:schedule_nbr>
 <v13:auto_receive>a</v13:auto_receive>
 <v13:to_location>jqofVPæ±</v13:to_location>
 <v13:from_location>DxZvzWæ±</v13:from_location>
 <v13:asn_nbr>pBLuagRBLOIbAjLNzNHqkLClwæ±</v13:asn_nbr>
 <v13:asn_type>a</v13:asn_type>
 <v13:container_qty>6</v13:container_qty>
 <v13:bol_nbr>ySwcdqfJLoZDDæ±</v13:bol_nbr>
 <v13:shipment_date>2008-12-23T17:27:51</v13:shipment_date>
 <v13:est_arr_date>2008-12-23T17:27:51</v13:est_arr_date>
 <v13:ship_address1>HwLlbGroæ±</v13:ship_address1>
 <v13:ship_address2>KjVlæ±</v13:ship_address2>
 <v13:ship_address3>æ±</v13:ship_address3>
 <v13:ship_address4>plæ±</v13:ship_address4>
 <v13:ship_address5>æ±</v13:ship_address5>
 <v13:ship_city>zeæ±</v13:ship_city>
 <v13:ship_state>a</v13:ship_state>
 <v13:ship_zip>lQXoaWhzIewLSHMaYSiYAbvCEIæ±</v13:ship_zip>
```

```

<v13:ship_country_id>a</v13:ship_country_id>
<v13:trailer_nbr>LgvamYcQæ±</v13:trailer_nbr>
<v13:seal_nbr>yThCPXEIæ±</v13:seal_nbr>
<v13:transshipment_
nbr>xMKXnbWifsBSmgPLwLyMxiWLPJæ±</v13:transshipment_nbr>
<v13:ASNOutDistro>
 <v13:distro_nbr>VBoEwQæ±</v13:distro_nbr>
 <v13:distro_doc_type>a</v13:distro_doc_type>
 <v13:cust_order_nbr>jnveQSbVZtlAIAGaQGjisæ±</v13:cust_order_nbr>
 <v13:fulfill_order_nbr>aVjQVDqDPMæ±</v13:fulfill_order_nbr>
 <v13:consumer_direct>a</v13:consumer_direct>
<v13:ASNOutCtn>
 <v13:final_location>PwkJygæ±</v13:final_location>
 <v13:container_id>sjlkOzhQkassAOJoæ±</v13:container_id>
 <v13:container_weight>12.4</v13:container_weight>
 <v13:container_length>12.4</v13:container_length>
 <v13:container_width>12.4</v13:container_width>
 <v13:container_height>12.4</v13:container_height>
 <v13:container_cube>12.2</v13:container_cube>
 <v13:expedite_flag>a</v13:expedite_flag>
 <v13:in_store_date>2008-12-23T17:27:51</v13:in_store_date>
 <v13:tracking_nbr>dQnSDeQCfiytpdMmnPoKTæ±</v13:tracking_nbr>
 <v13:freight_charge>20.4</v13:freight_charge>
 <v13:master_container_id>zfKcepZCSilooFwiaæ±</v13:master_
container_id>
<v13:ASNOutItem>
 <v13:item_id>SRnrJdLRnvtEllijdjHGsaæ±</v13:item_id>
 <v13:unit_qty>12.4</v13:unit_qty>
 <v13:gross_cost>20.4</v13:gross_cost>
 <v13:priority_level>1</v13:priority_level>
 <v13:order_line_nbr>4</v13:order_line_nbr>
 <v13:lot_nbr>eNQKYwMLæ±</v13:lot_nbr>
 <v13:final_location>HEuXHBæ±</v13:final_location>
 <v13:from_disposition>æ±</v13:from_disposition>
 <v13:to_disposition>æ±</v13:to_disposition>
 <v13:voucher_number>LxIZqBAFiyJSæ±</v13:voucher_number>
 <v13:voucher_expiration_date>2008-12-7:27:51</v13:voucher_
expiration_date>

```

```

 <v13:container_qty>6</v13:container_qty>
 <v13:comments> dIeiæ±</v13:comments>
 <v13:unit_cost>20.4</v13:unit_cost>
 <v13:base_cost>20.4</v13:base_cost>
 <v13:weight>12.4</v13:weight>
 <v13:weight_uom>æ±</v13:weight_uom>
 </v13:ASNOutItem>
 <v13:comments> STINVIEZæ±</v13:comments>
 <v13:weight>12.4</v13:weight>
 <v13:weight_uom>æ±</v13:weight_uom>
 <v13:carrier_shipment_nbr>EVHHCpBEjkiFPuvæ±</v13:carrier_
shipment_nbr>
 <v13:original_item_id>ghsKCYMTIOHSMeuVQRzyUæ±</v13:original_
item_id>
 </v13:ASNOutCtn>
 <v13:comments> wgæ±</v13:comments>
</v13:ASNOutDistro>
<v13:comments>OEmCazvSQuVfXXvbSyqæ±</v13:comments>
<v13:carrier_code>æ±</v13:carrier_code>
<v13:carrier_service_code>dqæ±</v13:carrier_service_code> <!--Optional:
-->
 </v13:ASNOutDesc>
</v12:publishASNOutCreateUsingASNOutDesc>
</soapenv:Body>
</soapenv:Envelope>

```



---

---

## Secure Integration Gateway Services

IGS services can be secured by attaching Web service policies to the service. When the services are secured, the service consumers call the services with credentials and/or certificates. If the Retail Service Backbone (RSB) is installed in your enterprise, the IGS services are called via the RSB layer. In that case, the RSB should also be secured. Since the Web service invocation syntax and credentials have to satisfy the security policies at each layer, a set of policies are identified and supported by RSB that are compatible from the consumer to RSB and then to the provider (IGS in this case). These compatible policies are called Policy A and Policy B. Although the services can be secured with any of the Web services policies, the scripting and automation support is available only for policies included in Policy A and Policy B. We believe these configurations satisfy most of the common security requirements of the customers. Any requirements outside these configurations will have to be configured manually.

Policy A is SSL with Username Token. The transport layer of the service invocation uses https for this policy. The consumers will have to provide the username and password for invoking the service. Policy B is message protection with Username Token. Policy B should be called with http. The encryption is at the message level, instead of transport layer.

IGS services can be secured in the same way as any other Web service provided by Oracle Retail is secured.

---

---

**Note:** For more information on securing the IGS services, see the *Oracle Retail Service Backbone Security Guide*.

---

---

