# Oracle® Retail Integration Cloud Service Implementation Guide–Concepts



Release 24.0.201.0 F95627-01 April 2024

ORACLE

Oracle Retail Integration Cloud Service Implementation Guide-Concepts, Release 24.0.201.0

F95627-01

Copyright © 2024, 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

#### Send Us Your Comments

#### Preface

Audience	vii
Documentation Accessibility	vii
Customer Support	vii
Improved Process for Oracle Retail Documentation Corrections	vii
Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)	viii
Conventions	viii

#### <u>1</u> Introduction

#### 2 Core Concepts

Key Functional Requirements	2-1
Guaranteed Once-and-Only-Once Successful Delivery	2-1
Preservation of Publication Sequence	2-1
Message Family and Message Types	2-2
Foundation Messages	2-2
Transactional Messages	2-3
RIB Message Envelope and Payloads	2-3
Message Life Cycle	2-3
Messaging Components	2-5
RIB Subsystem Components	2-5
Adapters	2-5
JMS Domains, Destinations, Subscriptions	2-6
JMS Message Selector	2-6
Additional RIB JMS Message Properties	2-7
Simple Message Flow	2-9
The RIB Hospital	2-9
RIB Hospital Dependency Check	2-10



RIB Hospital Insert	2-10
RIB Hospital Tables	2-11
RIB Hospital Retry	2-12
PUB Retry Adapter	2-12
Hospital Attempt (Retry) Count	2-15
JMS Delivery Count	2-15

#### 3 Cloud

Configuring RIB-RWMS for Hybrid Cloud Deployment Topology	
Installation and Setup instructions for RIB-RWMS Secondary (On-Premise)	3-2
Installation Prerequisite	3-3
Prepare the WebLogic Server	3-3
Creating Required RCU Schema Using the Repository Creation Utility	3-9
Creating a WebLogic Domain with wls Policy	3-14
Steps for ear Deployment	3-28

### 4 RIB Self-Service Enablement

Provisioning RIB-Adapters	4-4
How to Remove Dynamic Adapters Selection in RIB-RMS	4-7
Provisioning System Options	4-8
Provisioning InjectorService URL	4-9
RIB ServiceMonitor	4-10

#### 5 Performance

Performance Factors	5-1
Performance and Parallel Logical Channels	5-1

### 6 Security

RIB Application Administrators Security Domain	6-1
Integration with SIOCS	6-1
Integration with ROB	6-4

### 7 Integration with Fusion Middleware

General RIB to Fusion Middleware Architecture	7-1
---	-----



### 8 Integration with External Applications

mplementing RIB-EXT	8-1
How to Send/Receive Messages to/from the RIB System	8-2
External Application as a Publisher (rest-app) using OAuth2	8-2
Create OAuth2 Client Application in IDCS	8-3
External Application as a Subscriber (rest-app)	8-8
How to implement Injector Service (Service Contract) using ReST	8-9
How to Secure Injector Service with Oauth2	8-9
RIB-EXT Side of Configuration to Point to External Application	8-10
How to switch Injector Service app Type at Runtime	8-14
How to Change rib-ext injector-service-app-type from REST to SOAP	8-14
How to change rib-ext injector-service-app-type from SOAP to ReST	8-16
Error Handling	8-17
Monitoring Integration	8-17

### A Appendix - Sample Files

Sample Application.wadl File	A-1
Sample Resource Class	A-2
ApplicationMessages.xsd	A-3
payload.properties	A-5
Sample Request/Response for ReST Injector Service	A-5



### Send Us Your Comments

#### Oracle Retail Integration Cloud Service Implementation Guide-Concepts

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 new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retaildoc\_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 www.oracle.com.



### Preface

The Oracle Retail Integration Bus Implementation Guide provides detailed information that is important when implementing RIB.

### Audience

The Implementation Guide is intended for the Oracle Retail Integration Bus application integrators and implementation staff, as well as the retailer's IT personnel.

### **Documentation Accessibility**

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc</a>.

#### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs</a> if you are hearing impaired.

### **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.
italic	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 Introduction

RIB acts as a shared communication layer for connecting various Oracle Retail applications and external applications throughout an enterprise computing infrastructure. It supplements the core asynchronous messaging backbone with additional application functionality such as intelligent transformation, routing and error handling.

Communication across the RIB is via xml messages (payloads). These payloads describe the retail business objects (such as items, purchase orders, suppliers, and so on) in a standard way and are governed by RIB on behalf of the Oracle Retail applications.

RIB architecture is based on standard Java EE components and the Java Message Service (JMS). JMS is an integral part of the Java EE (Java Enterprise Edition) Technology stack.

The Integration Gateway Services (IGS) and RIB-ext components provides an integration infrastructure for external system (3rd 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.

The issues and considerations needed to properly deploy and configure the integration solution within an enterprise are the subject of this guide.



## 2 Core Concepts

The RIB is designed as an asynchronous publication and subscription messaging integration architecture. This allows the decoupling of applications and their systems. For example, a publishing application need not know about the subscribing applications, other than the requirement that at least one durable subscriber must exist. It decouples the systems operationally. Once a subscriber is registered, the RIB persists all published messages until all subscribers have seen them.

The publishing adapter does not know, or care, how many subscribers are waiting for the message, what types of adapters the subscribers are, what the subscribers' current states are (running or stopped), or where the subscribers are located. Delivering the message to all subscribing adapters is the responsibility of the RIB with the help of the underlying JMS server.

Physically, the message must reside somewhere so that it is available until all subscribers have processed it. The RIB uses the JMS specification for its messaging infrastructure. The JMS accepts the message from the publisher and saves it to stable storage, a JMS topic, until it is ready to be picked up by a subscriber. In all cases, message information must be kept on the JMS until all subscribers have read and processed it.

The RIB interfaces are organized by message family. Each message family contains information specific to a related set of operations on a business entity or related business entities. The publisher is responsible for publishing messages in response to actions performed on these business entities in the same sequence as they occur.

Each message family has specific message payloads based on agreed upon business elements between the Oracle Retail applications.

### **Key Functional Requirements**

The design and architecture of the RIB infrastructure is based on two key requirements driven by the Oracle Retail application business model.

### Guaranteed Once-and-Only-Once Successful Delivery

The RIB must preserve and persist all business events (messages) until all applications (subscribers) have looked at the message and have successfully consumed it or decided they do not care about that event (message). In other words, RIB must deliver to every subscriber all messages except those filtered as per a subscribing application's requirements.

A business event (message) must be redelivered to the consumer application if the business event (message) was not consumed successfully. The redelivery process is bound by the same rules of sequencing as normal (non-redelivered) business event (message).

### Preservation of Publication Sequence

The business event (message) must be delivered to all the subscribing applications in the order (FIFO) the business event (messages) was published by the publishing application.



To enable this, the publishing application defines a business object ID whose existence informs RIB that this and all subsequent messages with the same business object ID have to be processed in order. Business event (message) ordering (FIFO) is assured only for messages with the same business object ID within the same message family.

### Message Family and Message Types

The RIB messaging adapters and payloads are designed around the concept of a message family.

Each RIB message belongs to a specific message family. Each message family contains information specific to a related set of operations on a business entity or related business entities. The publisher is responsible for publishing messages in response to actions performed on these entities in the same sequence as they occur.

One example of a message family is the Order message family used to contain information about purchase order events.

A message family may contain multiple message types. Each message type encapsulates the information specific to a business entity within one or more business events. For example, the Order message family is published for events such as Create PO Header, Create PO Detail, Update PO Header, or Delete PO Detail.

A single business event, such as updating a purchase order, may involve multiple business entities, such as a line item within the purchase order.

Because a single business event may involve multiple business entities, the application may publish messages for this event from multiple message families for a single business transaction. More than one message type within a message family may also be created.

There are two broadly defined types of functional interfaces in the RIB (message families): foundation data and transactional data.

#### **Foundation Messages**

After populating application tables with initial company seed data, item foundation information is needed. Foundation messages are defined as those with payload that carry basic product data.

This table is an example from the Oracle Retail Integration Bus Integration Guide.

Functional Area	Publishing Applications	Subscribing Applications
Items	RMS	RWMS, SIM
Item Locations	RMS	SIM
Locations	RIB	RWMS
Stores	RMS	RWMS, SIM
Vendor	RMS	RWMS, SIM
Warehouses	RMS	RWMS, SIM



#### **Transactional Messages**

After populating application tables with initial seed data and after all required item foundation data messages have been subscribed to, all applications are prepared to publish and subscribe transaction data messages. Transactional messages communicate business events involving two or more organizations within a retail supply chain, for instance, among Oracle Retail Merchandising System (RMS), Oracle Retail Store Inventory Management (SIM), and Oracle Retail Warehouse Management System (RWMS), external suppliers and financial systems.

Functional Area	Publishing Applications	Subscribing Applications
Allocations	RMS	RWMS, SIM
Appointments	RWMS	RMS, SIM
ASN Outbound	RWMS, SIM, RMS, RFM	RMS, SIM, RWMS,
ASN Inbound	RWMS, External, RMS RFM	RMS, SIM, RWMS
Inventory Adjustments	RWMS, SIM	RMS
Inventory Request	SIM	RMS
Receipts	RWMS, SIM	RMS
Purchase Order	RMS, SIM	RWMS, SIM
Stock Order Status	RWMS, SIM	RMS, SIM
Transfers	RMS	RWMS, SIM

This table is an example from the Oracle Retail Integration Bus Integration Guide.

### **RIB Message Envelope and Payloads**

Whenever a publishing application adapter publishes a message, it wraps the message in an envelope known as the RIB message envelope. The envelope is a standard message delivery format where the message information, the data payload, is contained within the overall delivery information. The envelope itself provides information that the RIB uses, such as RIB hospital information and routing information.

#### Note:

Payloads do not support time zone formats.

### Message Life Cycle

The publishing application is responsible for creating the initial message contents. The RIB publishing adapter publishes it to the JMS Server and makes it available to any JMS subscribers. The RIB knows what subscribers are to receive the message due to the RIB configuration—this configuration associates a set of subscribers to each publisher and message family combination.



For PL/SQL Applications, database tables associated with the publishing application typically stage message information. One or more RIB publishing adapters poll the application via a stored procedure call. For Java EE Applications, the application calls a RIB Enterprise Java Bean (EJB) with the payload information to be published. Similarly, SOAP Applications calls with the payload information in the request to be published.

The message resides on a Java Message Service (JMS) immediately after publication. The JMS topic provides stable storage for the message in case a system crash occurs before all message subscribers receive and process it.

A fundamental RIB system requirement is that a message must be delivered to and processed successfully exactly once by each subscriber. Furthermore, all work performed by the subscriber and the RIB must be atomically committed or rolled back, even if the JMS server is on a remote host. The standard way to perform this is by using an XA compliant interface and two-phase commit protocol.

After initial publication, a message may undergo a series of transformation, filtering, or routing operations. A RIB component that implements these operations is known as a Transformation and Address Filter/Router (TAFR) component. TAFR is the acronym for Transform, Address, Filter, and Route. A TAFR is completely internal to the RIB and does not reside in either the publishing or subscribing application. The RIB performs these intermediate transformation and routing operations on some messages before making them available to the subscribing application.

A single TAFR may only transform a given message, only filter the message, only route it, or combine any of the three operations.

- Transform A message may be transformed from one message type into another, for example, WH (warehouse) from RMS to Location for RWMS.
- Filter A message may be filtered. Filtering can occur based on message type or based on content.
- Route A TAFR may route a message. For example, whenever a stock order message is published for a warehouse with an instance of RWMS, the TAFR routes it to the particular RWMS instance from where the stock will be fulfilled and not to warehouses that do not stock the order's items.

TAFR operations are specific to the set of subscribers to a specific message family. Multiple TAFRs may process a single message for a specific subscriber and different specific TAFRs may be present for different subscribers. Different sets of TAFRs are necessary for different message families. If all subscribers to a message can process all messages within a message family without any TAFR operations, then no TAFR components are needed.

Message processing continues until a subscribing adapter successfully processes the message or determines that no subscriber needs this message.

When a subscriber gets a message to be processed, the adapter checks to see if the RIB Hospital contains any messages associated with the same entity as the current message. If so, then the adapter places the current message in the hospital as well. This is to ensure messages are always processed in the proper sequence. If proper sequencing is not maintained, the subscribing application's data can be corrupted.

If an error occurs during message processing, the subscribing adapter notes this internally and rolls back all database work associated with the message. When the message is re-processed (because it has yet to be processed successfully), the adapter now recognizes this message is problematic and checks it into the hospital. If



adding the message to error hospital fails, the subscribing adapter writes the message to the file system. This becomes a poison message (.xml).

After a message is checked into the RIB Hospital, a retry adapter extracts the message from the hospital and re-publishes it to the JMS topic for reprocessing. The message remains in the hospital during all re-tries until the subscribing adapter successfully processes it. Subscribing retry adapter also processes the poison message. It extracts the message from the poison-message file and adds it to the error hospital to be retried. The poison message file will be renamed to processed message (.processed). If the retry adapter fails to process the poison message, the file is moved to human-workflow file (.humanworkflow).

The unprocessed poison messages should be corrected with a human intervention. They are made available in object storage bucket at a regular interval. These messages should be downloaded from object-store, corrected and uploaded back to object store. RIB will process these uploaded messages through subscriber retry adapter.

### **Messaging Components**

The RIB is a messaging system made-up of components that are packaged and shipped as an integration solution between the Oracle Retail applications. The application boundary between RIB and Oracle Retail applications can be confusing at times, so this section defines the RIB components and their responsibility and ownership. A diagram illustrating the RIB integration message flow follows:

#### **RIB Subsystem Components**

This section describes the components of the RIB subsystem.

#### Adapters

A RIB adapter is a component that coordinates business event (message) generation and processing with the respective Oracle Retail application interface. Each adapter in the RIB is created to handle a specific functional interface. RIB adapters are developed using Enterprise Java Beans (EJB) components architecture, subscribing adapters use Message Driven Beans (MDBs) and publishing adapters use Stateless Session Beans (SLSBs).

RIB provides four types of adapters that Oracle Retail applications can exploit to integrate with one another. These adapter types are: publisher, subscriber, TAFR, and hospital retry. They have been built using different technologies based on their particular needs.

Subscriber and TAFR adapters use Message Driven Bean (MDB) technology to register with JMS topics and receive messages for further processing.

Publisher and hospital retry adapters make use of the Java SE (Standard Edition) timer facility to schedule repetitive events that trigger calls to Stateless Session Beans (SLSBs) to query application tables for messages to publish to the JMS server.

As stated in the introduction, a fifth type of adapter exists for publishing messages in a pushing fashion. The Oracle Retail applications invoke this adapter at will for publishing messages.

These adapters have not been considered part of the scope of this technical document in regard to providing a mechanism for starting and stopping them.

Due to the variety of technologies used by the adapters, the goal of this technical design has been to isolate users from these differences and provide them with a common management



interface that can be used to control the state of the adapters. During the last few years, the Java Management Extensions (JMX) specification has become a well known standard that defines the management layer for enterprise Java applications. JMX defines standard methodologies for declaring enterprise application components as manageable resources that can be exposed in a consistent way such that any JMX compliant management application can access and provide means for control.

#### JMS Domains, Destinations, Subscriptions

JMS defines two types of messaging domains: point-to-point and publish/subscribe. RIB uses publish/subscribe types of messaging domains for all its communication. Publish/subscribe is a one-to-many type of message distribution model where one source application en-queues the message and many destination applications can dequeue the same message and process independently of the other peer applications. In publish/subscribe the destinations are known as topics, the en-queue application is known as publisher, and the de-queue is known as subscriber. Unlike point-to-point, in publish/subscribe the publisher and subscriber are totally ignorant of each other and do not and should not know about each other's existence. The JMS Topics retain the messages only as long as it takes to distribute them to current active (running) subscribers. There is also a timing dependency between publishers and subscribers.

A client that subscribes to a topic can consume only messages published after the client has created a subscription, and the subscriber must continue to be active in order for it to consume messages. The JMS specification relaxes this timing dependency to some extent by allowing clients to create durable subscriptions. By creating durable subscriptions the JMS server will continue to hold the messages for all registered subscribers for that topic until the subscriber consumes the message or deletes the subscription.

There are two types of subscribers, non-durable and durable subscribers. The RIB uses only durable subscribers which allow the Oracle Retail edge applications to be in up or down state independently but still not lose any messages and catch up when the application comes back up. Every subscribing RIB adapter registers its durable subscriber with a subscription name that contains its rib-<app> application name and the adapter name in it.

RIB defines logical grouping of retail specific business objects (BO) and business functions in a concept called message family. For every message family there is a corresponding JMS topic. These JMS topics are used as communication pipelines between the source and destination Oracle Retail applications for exchanging the business objects.

The list of JMS topics used by RIB components is detailed in the Reports section of the *Oracle Retail Integration Bus Integration Guide*.

#### JMS Message Selector

A key aspect of the JMS usage that the RIB relies on is the attachment of message properties to published messages and the use of selectors by message subscribers. Message properties are used to convey information about the message outside of the actual message data to establish a logical channel for messages.

JMS message selectors are used by the RIB to filter the messages that each subscriber picks up. In other words, using the message properties, selectors act as a filter to weed out messages a subscriber should not process.



The message property set and used by the RIB messages is called threadValue. The thread value is associated with a logical channel of a message stream. All messages for a specific family with a specific business object ID always contain the same threadValue property. This, combined with the standard first in, first out (FIFO) message ordering on the topic, is integral to message sequencing. Messages with different threadValue properties are not guaranteed to be processed in the same relative order as publishing.

Messages published without JMS Message Property present will not be picked up by the standard subscribing RIB adapters.

Pseudo code for message selector:

(

```
(
    (appName is not null) AND
    (appName == $APP NAME)
 ) AND
  (
    (retryLocation is not null) AND
    (retryLocation LIKE $ADP CLASS DEF)
 )
) OR
(
 (
   (appName is null) OR
   (appName != $APP NAME)
 ) AND
 (
   (retryLocation is null) OR
   (retryLocation LIKE $ADP CLASS DEF)
)
) AND
(threadValue == $ADP INSTANCE NUMBER)
```

#### Additional RIB JMS Message Properties

Every message published by the rib-<app> applications includes a number of JMS user defined header properties. In the current release, these properties are only set, not used by any RIB components. In the future, these properties will be used for intelligent performance enhancement and optimization and for traceability and auditability of RIB messages.

The message properties are as follows:

Property Name: appName

Type: java.lang.String

Required Property: false

Example: appName=rib-rms

Description: The appName property contains the rib-<app> application name that published this particular message.

Property Name: adapterInstance

Type: java.lang.String

Required Property: false

Example: adapterInstance=Item\_pub\_1



Description: The adapterInstance property contains the rib-<app> adapter instance name that published this particular message.

Property Name: family

Type: java.lang.String

Required Property: false

Example: family=Item

Description: The family property contains the name of the RIB family name to which the message belongs.

Property Name: needMessageOrderPreservation

Type: boolean

Required Property: false

Example: needMessageOrderPreservation=true

Description: This property will have a value of true if any ribMessage node within the RibMessages xml has a message that has businessObjectId set. This property will allow us to take advantage of the fact that now we know which messages need message order preserving at JMS header level (without opening the message). In the future, we will be able to take advantage of that information, make our processing parallel, and get better throughput without losing message sequencing.

Property Name: topic

Type: java.lang.String

**Required Property: false** 

Example: topic=etItem

Description: This topic property contains the RIB topic name that this particular message is published to or subscribed from.

Property Name: ribKernelVersion

Type: java.lang.String

Required Property: false

Example: ribKernelVersion=22.0

Description: The system determines the rib kernel jar version number at runtime and includes its value in this JMS property.

Property Name: ribFuncArtifactVersion

Type: java.lang.String

Required Property: false

Example: ribFuncArtifactVersion=22.0

Description: This is a place holder for future enhancement. The idea is the system will somehow determine the runtime payload version and include that information in the message for better compatibility management. This property will be enhanced in a future release.

Property Name: ribMessageCount

Type: int

Required Property: false



Example: ribMessageCount=12

Description: This property contains the number of ribMessage nodes there are in a RibMessages xml message. This value gives us some indication of message aggregation in play. It might be used in the future to better optimize message flow paths based on the size/number of the messages.

Property Name: uuid

Type: java.lang.String

Required Property: false

Example: uuid=116cfabd-8949-4f93-bb61-aaa88e168f30

Description: This property contains a universally unique identifier for every message. This unique identifier will provide better traceability of a message within the JMS system. This property complements the ribMessageID xml element that is there to trace messages within the RIB logs.

### Simple Message Flow

The typical lifecycle of a message through the RIB is as follows:

- The publishing adapter creates the message. The event that triggers the message creation may be a polling operation in case of PL/SQL applications or a synchronous invoke in case of Java EE applications or a request in case of SOAP application. The message is published to a predetermined JMS topic.
- 2. The message is now available for all registered subscribers to the JMS topic for pick up. Subscription is based on the message family.
- 3. Once a subscriber gets the message, it is free to process that message according to its own rules. In the case of a transformer adapter, the adapter can open the message, modify its contents, and then publish the modified message to a new topic. The source topic and destination topic that a TAFR uses must always be distinct/different topics. There may be new subscribers to the modified message, and the scenario is repeated for each of these subscribers.
- 4. When each subscriber has finished (commit) processing a message, the JMS server updates the state of the message to reflect that it has been processed by this subscriber.
- 5. The JMS Server deletes the messages on the topic after delivering it to all the registered subscribers.

Two types of applications require this data and subscribe to it. One type of subscribing application requires a certain transformation be applied to the data, but the other type of subscriber can process the message without any transformations.

### The **RIB** Hospital

The RIB Hospital is a collective term for a set of Java Classes and database tables whose purpose is to provide a mechanism to handle system and business related errors while meeting the fundamental RIB requirements:

- Guaranteed once-and-only-once successful delivery.
- Preservation of publication sequence (even in case of failures).

When a message is processed, the adapter checks to see if the RIB Hospital contains any messages associated with the same businessObjectId as the current message. If so, then the



adapter places the current message in the hospital as well. This is to ensure messages are always processed in the proper sequence. If proper sequencing is not maintained, then the subscribing application's data can get corrupted.

If an error occurs during message processing, the subscribing adapter notes this internally and rolls back all work associated with the message. When the message is re-processed (since it is yet to be processed successfully), the adapter now recognizes this message is problematic and checks it into the hospital.

For Publication, there are some RMS publishers that return an 'H' status to denote a problem creating a new message for a specific business object. This status may be due to database locks being held by on-line users of an Oracle Forms application or it could also be due to some data incompatibility found in the GETNXT() procedure. Whenever a publisher recognizes that a message for a business object cannot be published due to one of these conditions, the message must go into the RIB Hospital.

After a message is checked into the RIB Hospital, a retry adapter extracts the message from the hospital and tries to re-publish it to the integration bus.

#### **RIB Hospital Dependency Check**

The RIB Hospital dependency check logic assumes that each message family has a single unique businessObjectId for all business object entities its messages are associated with. This businessObjectId must be the same for the same business entity across all message types within the message family. If any message for a specific business entity is placed into the RIB Hospital, then the RIB Hospital dependency check logic automatically inserts any subsequent messages for the same business object. This is to preserve the message sequencing and guaranteed exactly once successful message processing. Otherwise, multiple update messages for a business object may be processed in an incorrect order and create incompatibilities between applications.

If the businessObjectid is not set, then there is no dependency check. Not all message families set the businessObjectId or it is not set on all message types. See the Oracle Retail application documentation (for example, "Message Publication and Subscription Designs" in the *Oracle Retail Merchandising System Operations Guide Volume 2*).

#### **RIB Hospital Insert**

In an event of failure during message subscription, the error is flagged within the RIB Hospital software, resulting in rollback of the work done in the retail application, the adapter returns failure so that the database transaction is rolled back as well, and the message is kept on the integration bus topic. This is because subscribing adapters are executed within the context of a distributed transaction, using the XA two-phase commit protocol. This transaction is controlled by the Java EE Application Server. Immediately after the roll back, JMS re-delivers the message back to the subscribing adapter and this time the RIB Hospital software detects the previously flagged message and inserts the message in to the RIB Hospital tables and message is removed from the JMS topic.

When the initial failure occurs while processing the message, the error is flagged within the RIB Hospital software, the adapter returns failure so that the database transaction is rolled back, and the message is kept on the integration bus topic.



#### Note:

The XA interface is a standard protocol between a transaction manager and a database or resource manager. Note that both the JMS topic connection and the database connection must support the XA protocol. For more information regarding the XA standard, see the URL http://www.opengroup.org.

#### **RIB Hospital Tables**

The RIB Hospital tables are:

- RIB\_MESSAGE contains the message payload, all single-field envelope information, and a concatenated string made from <id> tags. It also contains a unique hospital ID identifying this record within the hospital.
- RIB\_MESSAGE\_FAILURE contains all failure information for each time the message was processed.
- RIB\_MESSAGE\_ROUTING\_INFO contains all of the routing element information found in the message envelope.
- RIB\_MESSAGE\_HOSPITAL\_REF contains all of the hospital reference information found in the message envelope.

A database sequence, RIB\_MESSAGE\_SEQ, is used to maintain a unique message number associated with each message placed into the RIB Hospital.



These tables will have been created during the database portion of the Oracle Retail application installation (for example, RWMS, SIM, RPM, AIP, RFM, OMS, or RMS).



The RIB Hospital tables are internal system tables that maintain the RIB runtime state of the system. The entries in these tables must not be manipulated by non RIB tools when the RIB is running.

#### **RIB Hospital Retry**

After a message is inserted into the RIB Hospital, the hospital retry adapter is used to re-post the message to the JMS in order to retry its processing. The assumption is that the error is a transitory one; records locked or there is an external dependency that has not been met. The number of times a message is retried is configurable.

The hospital retry is responsible for maintaining state information for hospital records or what has happened to the record or message information. Each time the message is reprocessed, a record is kept of the event along with the results. The design is to provide a means to halt processing for messages that cause errors while allowing continued processing for the good messages.

One element of this information is whether the message has been queued to the JMS topic for re-try processing. So manually deleting messages from the hospital database using SQL directly may produce severe processing problems. Also, deleting messages directly from the JMS provider may result in a message that is never retried again, as the logic in the retry assumes the message is queued within the JMS.

There are three kinds of hospital retry adapters:

- Sub Retry Adapter
- JMS Retry Adapter
- Pub Retry Adapter

All subscriber side retrying of messages are handled by the Sub Retry Adapter. The Sub Retry Adapter looks at all messages with reason code SUB, then filters and identifies the messages that are ready to be reprocessed, keeping message ordering in mind.

Oracle Retail applications are unaware that the integrations of the business data is happening through a JMS server. RIB abstracts the fact it is using a JMS server from the retail applications. When the JMS server is down or RIB has some problem publishing to the JMS server, RIB will not rollback the transaction as long as it is a recoverable problem. In such situation all messages are inserted to the RIB Hospital with a reason code of JMS and publications continues on. The JMS Retry Adapter retries all messages with reason code of JMS at a later time.

All messages with reason code of PUB are retried by the Pub Retry Adapter. RMS is the only retail application that needs the Pub Retry Adapter.

#### **PUB** Retry Adapter

The following diagrams illustrate how the PUB Retry Adapter works.





RIB PUB\_RETRY Adapter Processing





RIB PUB\_RETRY Adapter Processing





#### Hospital Attempt (Retry) Count

When the message first comes through the subscriber, if there is no businessObjectid, then there is no dependency check performed. If the message cannot be processed, it is then inserted into the hospital with an attempt\_count = 1.

A message that comes through the subscriber, that has a businessObjectid, a dependency check is performed. If there is no dependency and the message cannot be processed, it is then inserted into the hospital with an attempt\_count = 1.

A message that comes through the subscriber that does match the ID and family of another message in the hospital is known to be dependent, so it goes to the hospital immediately, with an attempt\_count = 0.

Exception to this rib-tafr app, in case of rib-tafr attempt\_count is 1, even if the message is inserted into the hospital as a dependent message because tafr adapters work with two topics and message would already be subscribed once by the tafr, therefore it always has attempt\_count=1.

#### JMS Delivery Count

JMSXDeliveryCount is a message property set by AQ JMS. This property is checked to see if the message is being redelivered by the JMS. If the count

MAX\_REDELIVERY\_THRESHOLD (set to 2) is reached, the RIB subscribers assume that the message is being re-delivered; the message will be determined as a poison message. The message is written to the file system (at the same location where application log files are written), and the adapter is shut down in such scenarios. An administrator must decide how this message will be handled.



# 3 Cloud

This chapter describes the RIB cloud.

The following diagram describes a sample hybrid architecture in which some of the retail applications are on-premise and some other (including RIB) are in the cloud. In this architecture, the retail applications RWMS is on-premise, while RIB is on the cloud.



Figure 3-1 Retail Integration Suite - Cloud Architecture

In order to support cloud deployment (including a hybrid cloud), RIB is enhanced with the addition of two Web services. These are injector and publisher Web services that allow retail applications to communicate with other applications.

### Configuring RIB-RWMS for Hybrid Cloud Deployment Topology

RWMS on-premise cannot communicate with RMS and other retail apps, which are all in cloud via RIB. As RIB is already supported in cloud, for enabling the integration of RWMS with all other retail applications which are in the hybrid cloud environment, RIB follows the primary/secondary approach. The secondary resides close to on-prem RWMS, while the primary is on-cloud. Communication between primary and secondary is through web service calls. The RIB-RWMS primary invokes the new web services exposed by secondary RIB-RWMS to send/receive messages to/from other applications on cloud via RIB.



For RIB-RWMS to communicate with RWMS on premise and RIB on cloud, it should be deployed in primary-secondary topology. Hybrid cloud set-up for RWMS involves a two part installation, one for each primary (cloud) and secondary components (onpremise).





Note:

The client-server architecture is only applicable to RIB and RWMS integration, where RIB is deployed on Next Gen SaaS Platform and the legacy RWMS is hosted on on-prem/PaaS.

### Installation and Setup instructions for RIB-RWMS Secondary (On-Premise)

This section describes the installation and setup instructions. This includes the installation prerequisites, preparing the WebLogic server, creating a WebLogic domain, verifying installation of wls policies, extending an existing domain to add wls policies, and deploying the EAR file.

#### Note:

The screen captures included in the following steps are for example only. Therefore, consider the illustrations as guides only; the values shown may not always apply.



#### Installation Prerequisite

The rib-rwms secondary (on-premise) application requires Oracle WebLogic Server 12c (12.2.1.4.0) and must be built with Java 8 (JDK 1.8.0+ 64 bit or later), with the latest security updates.

#### Important:

If there is an existing WebLogic 12.x.x or 10.3.xc installation on the server, you must upgrade to WebLogic 12.2.1.4.0. All middleware components associated with WebLogic server 10.3.6 should be upgraded to 12.2.1.4.0. Back up the weblogic.policy file (\$WLS\_HOME/wlserver/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post-patching installation steps are completed. For upgrading your WebLogic server to 12.2.1.4.0, use the appropriate Upgrade Installer.

#### Prepare the WebLogic Server

Take the following steps to prepare the WebLogic server:

- 1. Find fmw\_12.2.1.4.0\_infrastructure\_Disk1\_lof1.zip and download this file to your system.
- 2. Extract the contents of this zip file to your system. Use the fmw\_12.2.1.4.0\_infrastructure.jar file to run the installer.
- 3. Run the installer by executing the java -jar fmw\_12.2.1.4.0\_infrastructure.jar file. The Welcome window displays.





4. Click Next. The Auto Updates window displays.

are opaaroo	
Welcome         Auto Updates         Installation Location         Installation Type         Prerequisite Checks         Installation Progress         Installation Complete	Ergws



5. Select the appropriate radio button and click **Next**. The Installation Location window displays.

× – Oracle Fusion	n Middleware 12c WebLogic Server and Coherence Installation - Step 3 of 8
Installation Location	
Welcome     Auto Updates     Installation Location     Installation Type     Prerequisite Checks     Installation Summary     Installation Progress     Installation Complete	Oracle Home: <ul> <li>Peature Sets Installed At Selected Oracle Home:</li> <li>Yiew</li> <li>Browse</li> <li>Browse</li></ul>
	Oracle Home may only contain alphanumeric, underscore (_), hyphen (-) or dot(.) characters and it must begin with an alphanumeric character.
Help	< <u>Back</u> Next > Finish Cancel

6. Click **Browse** to select the Oracle Home location where the Weblogic server is to be installed. Click **Next**. The Installation Type window displays.



× – Oracle Fusior	n Middleware 12c WebLogic Server and Coherence Installation - Step 4 of 8	
Installation Type	ORACLE. FUSION MIDDLEWARE	
Installation Type <u>Welcome</u> <u>Auto Updates</u> Installation Location <u>Installation Type</u> <u>Prerequisite Checks</u> <u>Installation Summary</u> <u>Installation Progress</u> <u>Installation Complete</u>		
	<ul> <li>Open Source Components         Fusion Middleware Maven Support 12.2.1.4.0         Oracle Installation Infrastructure         FMW Platform Generic 12.2.1.4.0         OPatch 13.9.4.2.1     </li> </ul>	v
Help	< <u>B</u> ack <u>Next</u> Finish	Cancel

7. Select **Fusion Middleware Infrastructure** and click **Next**. The installer performs the prerequisite checks and ensures all required conditions are satisfied.

rerequisite Checks	_		
<u>Welcome</u> Auto Updates		100%	
Installation Location			
Installation Type	×	Checking operating system certification	
Prerequisite Checks	· · · · · · · · · · · · · · · · · · ·	Checking Java version used to launch the installer	
Installation Summary	II		
Installation Progress			
Installation Complete			
	Stop	Rerun Skip	View Lo
	E Ch	ecking operating system certification ecking lava version used to launch the installer	



8. When the prerequisite check completes successfully, click **Next**. The Installation Summary window displays.

× – Oracle Fusion I	Middleware 12c WebLogic Server and Coherence Installation - Step 6 of 8
Installation Summary	
Welcome         Auto Updates         Installation Location         Installation Type         Prerequisite Checks         Installation Progress         Installation Complete	<ul> <li>Install Oracle Fusion Middleware 12c WebLogic Server and Coherence</li> <li>Installation Location         Oracle Home Location: /home/bhagath/Oracle/Middleware/Oracle_Home         Log File Location: /tmp/Oralnstall2020-06-08_06-13-01PM/install2020-06-08_06-13-01PM         .log</li> <li>Disk Space         Required: 927 MB         Available: 140159 MB</li> <li>Feature Sets to Install         Administration Console Additional Language Help Files 12.2.1.4.0         ClE WLS Config 12.2.1.4.0         Core Application Server 12.2.1.4.0         Coherence Product Files 12.2.1.4.0         WebLogic SCA 12.2.1.4.0         WebLogic Client Jars 12.2.1.4.0         WebLogic Client Jars 12.2.1.4.0         Part Middleware Maven Support 12.2.1.4.0         Part JBC Drivers 12.2.1.4.0         WebLogic Evaluation Database 12.2.1.4.0         WebLogic Evaluation Database 12.2.1.4.0         WebLogic Evaluation Database 12.2.1.4.0         WebLogic Evaluation before starting the installation.         To change the above options before starting the installation, select the option to change in the left         pane or use the Back button.</li> </ul>
Help	< Back Next > Install Cancel

9. Click Install. The Installation Progress window displays.



Installation Progress		
V Welcome	100%	
Installation Location Installation Type Prerequisite Checks Installation Summary Installation Progress	<ul> <li>Prepare</li> <li>Copy</li> <li>Generating Libraries</li> <li>Performing String Substitutions</li> <li>Linking</li> </ul>	
Installation Complete	Image: Setup       Image: Setup	
	View Messages View Successful Tasks Hardware and S Engineered to Wor	View Loc Software k Together

**10.** Click **Next** when the installation completes. The Installation Complete window displays.

× – Oracle Fusion /	Middleware 12c WebLogic Server and Coherence Installation - Step 8 of 8
Installation Complete	
Welcome     Auto Updates     Installation Location     Installation Type     Prerequisite Checks     Installation Summary     Installation Progress     Installation Complete	<ul> <li>Install Oracle Fusion Middleware 12c WebLogic Server and Coherence</li> <li>Installation Location         <ul> <li>Oracle Home Location: /home/bhagath/Oracle/Middleware/Oracle_Home</li> <li>Log File Location: /tmp/Orainstall2020-06-08_06-13-01PM/install2020-06-08_06-13-01PM</li> <li>.log</li> </ul> </li> <li>Feature Sets Installed Successfully         <ul> <li>Administration Console Additional Language Help Files 12.2.1.4.0</li> <li>CIF WLS Config 12.2.1.4.0</li> <li>Core Application Server 12.2.1.4.0</li> <li>Coherence Product Files 12.2.1.4.0</li> <li>WebLogic CSCA 12.2.1.4.0</li> <li>WebLogic CCA 12.2.1.4.0</li> <li>Fusion Middleware Maven Support 12.2.1.4.0</li> <li>FW Platform Generic 12.2.1.4.0</li> <li>OPatch 13.9.4.2.1</li> <li>Third party JDBC Drivers 12.2.1.4.0</li> </ul> </li> <li>WebLogic Evaluation Database 12.2.1.4.0</li> </ul>
	✓ Automatically Launch the Configuration Wizard Oracle Fusion Middleware 12c WebLogic Server and Coherence installation completed successfully
Help	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel



#### Creating Required RCU Schema Using the Repository Creation Utility

To create a schema user for the domain, take the following steps:

1. Run the RCU from the <MW\_HOME>/oracle\_common/bin folder. The Welcome window displays.

	Repository Creation Utility - Step 1 of 8	_ ×
Repository Creation U	tility ORACLE FUSION MIDDLEWARE	<
© Welcome	Welcome to Repository Creation Utility 12.2.1.4.0 for Oracle Fusion Middleware.	
Create Repository	The Repository Creation Utility enables you to create and drop database schemas that	are required
Database Connection Details	for Oracle Fusion Middleware products.	
Select Components		
Schema Passwords		
Map Tablespaces		
y <u>Summary</u>		
Completion Summary		
	Copyright © 1996,2019, Oracle and/or its affiliates. All rights reserved.	
Help	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

2. Click Next and select the Create Repository option.





3. Click **Next**. Enter the database credentials where the schema user has to be created.

epository Creation Ut	ility			(♀)
Velcome Create Repository Database Connection Det Select Components Schema Passwords Map Tablespaces Summary Completion Summary	Database Type: Connection String Format: Connect String Host Name: Port: Service Name: Username: Password: Role:	Oracle Database  Connection Parameters  1521  Normal	Connection String	
	For RAC database, specify VI For SCAN enabled RAC datab	P name or one of the Node na ase, specify SCAN host as Ho	ame as Host name. ost name.	



4. Click Next. Specify the prefix to be used for the schema user creation. For example, INT. Select Metadata Services, Weblogic Services, and Oracle Platform Security Services.

<u>Welcome</u> <u>Create Repository</u> Database Connection Details	pecify a unique prefix for all s d manage the schemas later	chemas created in this sess	ion, so you can easily locate, referen
Database Connection Details			
D'attababe commetteren b'ottane	Select existing prefix:	BDI32ABK	
Select Components	Crasta new profix:	DICS	
Schema Passwords	<u>Create new prenx</u> .	Alaka serie ala Gara	a she she take a second s
Man Tablesnases		Alpha numeric only. Canno	t start with a number. No special
Map Tablespaces	Component		Schema Owner
Summary	Oracle AS Repository	Components	
Completion Summary	■ AS Common Schemas		
	Common Infras	tructure Services *	RICS STB
	☑ Oracle Platform	n Security Services	RICS_OPSS
	✓User Messagin	g Service	RICS_UMS
	Audit Services		RICS_IAU
	Audit Services	Append	RICS_IAU_APPEND
	Audit Services	Viewer	RICS_IAU_VIEWER
	Metadata Serv	ices	RICS_MDS
	Weblogic Servi	ces *	RICS_WLS

5. Click Next. Specify the password.



	Repository	Creation Utility - Step 5	5 of 8		-
epository Creation L	Jtility		Ē		7)
Welcome     Create Repository     Database Connection Details     Select Components     Schema Passwords     Map Tablespaces     Summary	Define passwords for <u>Use same password</u> <u>Password</u> : <u>Confirm Password</u> : Use main schema r	main and auxiliary schema us ds for all schemas Alpha numeric only. Cannot st No special characters except: 	ers. art with a n : \$, # :	umber.	
	_ <u>Specify</u> different pa	sswords for all schemas			

6. Click **Next**. The window provides the details of tablespaces created as part of schema creation.

			TODION IM	
<u>Welcome</u> <u>Create Repository</u> <u>Database Connection Details</u>	Default and temporary tablespaces for the selected components appear in the table below. To create new tablespaces or modify existing tablespaces,use the Manage Tablespaces Button Manage Tablespa			
Select Components	Component	Schema Owner	Default Tablesnace	Temp Tablespace
Schema Passwords	Common Infrastructu	DICS STR	*PICS STP	*PICS INS TEMP
Map Tablespaces	Oracle Platform Secu	DICS OPSS	*PICS INS OPSS	*RICS_IAS_TEMP
Summany	User Messaging Serv	RICS LIMS	*RICS IAS LIMS	TRICS_IAS_TEMP
summary	Audit Services	RICS IALL	*RICS IALL	*RICS IAS TEMP
Completion Summary	Audit Services	RICS IALL APPEND	*RICS IALL	*RICS IAS TEMP
	Audit Services Append	RICS IALL VIEWER	*RICS IAU	*RICS IAS TEMP
	Metadata Services	BICS MDS	*RICS MDS	*BICS IAS TEMP
	Weblogic Services	BICS WIS	*RICS WIS	*RICS IAS TEMP
	* Default tablespaces (sp	ecified in the configura	ation files) are to be creat	ed upon confirmation.

7. Click **Next**. The Confirmation window displays.




8. Click OK. The Summary window displays.

	Repository Creation U	Itility - Step 7 of	8	-	2
Repository Creation U	tility				
Welcome     Create Repository     Database Connection Details     Select Components     Schema Passwords     Map Tablespaces     Summary	Database details: Host Name Port Service Name Connected As Operation Prefix for (prefixable) Schema Owne	1521 System and Data ers RICS	a Load concurrently		
Completion Summary	Component Common Infrastructure Services	Schema Owner RICS_STB	Tablespace Type Default Temp Additional	Tablespace Name RICS_STB RICS_IAS_TEMP [None]	ľ
	Oracle Platform Security Services	RICS_OPSS	Default Temp Additional	RICS_IAS_OPSS RICS_IAS_TEMP [None]	11111
	User Messaging Service	RICS_UMS	Default Temp Additional	RICS_IAS_UMS RICS_IAS_TEMP [None]	
	Audit Services	RICS_IAU	Default Temp Additional	RICS_IAU RICS_IAS_TEMP [None]	
	Audit Services Append	RICS_IAU_APPEND	Default Temp	RICS_IAU RICS_IAS_TEMP	-
Help	Save Kesponse File	< 5	ack Next >	<u>C</u> reate Canc	el

9. Click **Create and proceed** to create the schema. This could take a while to complete. The Completion Summary window displays.

	Itility			
Velcome Create Repository Database Connection Details Select Components Schema Passwords Map Tablespaces Summary Completion Summary	Database details: Host Name Port 1521 Service Name Connected As Operation System and Execution Time 2 minutes RCU Logfile //tmp/RCU2 Component Log Directory View Log rcu.log Prefix for (prefixable) Schema Owners	d Data Load concu 20 seconds 020-06-07_23-32_1 020-06-07_23-32_1	rrently 764391877/logs/rcu.log 764391877/logs	
	Component	Status	Time	Loafile(Click to view
	Common Infrastructure Services	Success	00:09.852(sec)	stb.log
	Oracle Platform Security Services	Success	00:20.657(sec)	opss.log
	User Messaging Service	Success	00:20.797(sec)	ucsums.log
	Audit Consistent	C	00.14 000/	
	Audit Services	Success	00:14.029(sec)	iau.log
	Audit Services Audit Services Append Audit Services Viewer	Success Success Success	00:14.029(sec) 00:09.278(sec) 00:09.248(sec)	iau.log iau_append.log iau_viewer.log
	Audit Services Audit Services Append Audit Services Viewer Metadata Services	Success Success Success Success	00:14.029(sec) 00:09.278(sec) 00:09.248(sec) 00:17.136(sec)	iau.log iau_append.log iau_viewer.log mds.log

#### Creating a WebLogic Domain with wls Policy

To create a new WebLogic domain with wls policy, take the following steps:

1. Run config.sh from the <ORACLE\_HOME>/oracle\_common/common/bin folder. The Configuration Type window displays.

I	Fusion Middlewa	are Configuration Wiza	d - Page 1	of 20		_ ×
Configuration Type			ī			
🙊 Create Domain	1					
Templates						
Administrator Account						
Domain Mode and JDK						
Database Configuration Type						
Component Datasources						
JDBC Test	What do you want	to do?				
Advanced Configuration	● <u>C</u> reate a new of	domain				
Managed Servers	O Update an exis	sting domain				
Ulusters						
Server Templates						
Coherence Clusters	Domain Location:	/scratch/u00/webadmin/WL	S12214/user_	projects/domains	/base_d	Browse
Machines						
Virtual Targets						
Partitions						
Deployments Targeting						
Services Targeting						
Configuration Summary	Create a new dom	ain.				
Configuration Progress						
End Of Configuration						
Help			< Back	<u>N</u> ext >	Finish	Cancel



Select Create a new domain, provide Domain Location, and click Next. The Templates window displays. By default, the Basic WebLogic Server Domain [wlserver] checkbox is selected. Select the Oracle JRF [oracle\_common], Oracle Enterprise Manager [em], Oracle WSM Policy Manager [oracle\_common], and Weblogic Advanced WebServices for JAX-WS Extension [oracle\_common] check boxes.

Create Domain	
Templates         Application Location         Administrator Account         Domain Mode and JDK         Database Configuration Type         Component Datasources         JDBC Test         Advanced Configuration         Configuration Summary         Configuration Progress         End Of Configuration	Create Domain Using Product Templates:     Filter Templates:     Type here     Include all gelected templates     Available Templates     Varilable Templates     Oracle Enterprise Manager [em]     Oracle Interprise Manager-Restricted JRF [em]     Oracle User Messaging Service Basic [oracle_common]     Oracle JRF SOAP/JMS Web Services [oracle_common]     Oracle WSM Policy Manager [oracle_common]     Oracle BRF [oracle_common]     Oracle Restricted JRF [oracle_common]     Oracle Doce Doce Doce Doce Doce Doce Doce Doc

- 3. Click Next. The Application location window displays; provide the application location.
- 4. Click Next. The Administrator Account window displays. Enter the user credentials you want to use to log in to the WebLogic Administration Console.



Administrator Account		
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Type Component Datasources JDBC Test Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Name Password Confirm Password Must be the same at at least one number	password. Password must contain at least 8 alphanumeric characters wit

5. Click Next. The Domain Mode and JDK window displays. Set the **Domain Mode** as **Production** and select the JDK version (JDK 1.8 with the latest security updates) you want to use.

1	Fusion Middleware Configuration Wizard - Page 5 of 12
Domain Mode and JDK	
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Type Component Datasources JDBC Test Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Domain Mode         Development         Utilize boot.properties for username and password, and poll for applications to deploy. <ul> <li>Production</li> <li>Require the entry of a username and password, and do not poll for applications to deploy.</li> </ul> JDK <ul> <li>Oracle HotSpot 1.8.0_131 /scratch/u00/webadmin/java_latest/jdkl.8.0_131</li> <li>Other JDK Location:</li> <li>Browse</li> </ul>
Help	< <u>Back</u> <u>N</u> ext > Finish Cancel

- 6. Click **Next**. The Database Configuration Type window displays.
  - a. Select the **RCU Data** radio button.



- b. Select Oracle as the Vendor.
- c. Select Oracle's Driver (Thin) for Service connections; Version 9.0.1 and later as the Driver.
- d. Enter the Service, Host Name, Port, Schema Owner, and <u>Schema Password</u> for the \* STB schema created using RCU.
- e. Click Get RCU Configuration.

The Connection Result Log displays the connection status.

F	usion Middleware Configuration Wizard - Page 6 of 12
Database Configuration 1	
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Ty Component Datasources JDBC Test Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Posion Middlewake         Specify AutoConfiguration Options Using:            • <u>B</u> CU Data <u>Manual Configuration</u> Enter the database connection details using the schema credentials corresponding to Common Infrastructure Services component in the Repository Creation Utility. The Wizard uses this connection to automatically configure the datasources required for components in this domain.         Vendor:       Oracle       Driver: *Oracle's Driver (Thin) for Service connections; •            • Connection Parameters Connection URL String          Host Name:         DBMS/Service:       Port: 1521         Schema Owner:       RICS_STB         Schema Password:       ••••••••         Get RCU Configuration
Help	< Back Next > Einish Cancel

7. Click Next. The JDBC Component Schema window displays.



JDBC Component Schem	a						
Create Domain	Vend	dor:	D	river:			-
Templates	() C	onnection Parameters	Connec	tion <u>U</u> RL String			
	Host	t Name:					
Domain Mode and JDK	DBM	IS/Service:	P	ort:			
Database Configuration Type	Sche	ema Owner:	s	chema Password:			
	10.5253	Convert to Gri	dLink ()Cor	ivert to RAC multi (	data sour	ce () Don't	convert
Advanced Configuration	Edits	Convert to Grives to the data above wi	ll affect all che	overt to RAC multion	data sour able belov	ce () Don't i	convert
Advanced Configuration Configuration Summary Configuration Progress	Edite	Convert to Gri s to the data above wi Component Schema	Il affect all cher DBMS/Service	overt to RAC multi of the tacked rows in the tacked Host Name	data sour able belov Port	ce O Don't Schema Ow	convert Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Edite	Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema	Il affect all cher DBMS/Service SBDB1	cked rows in the ta Host Name blr00abl.in.orac	data sour able belov Port 1521	ce O Don't Schema Ow RICS_STB	Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Edite	Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema WLS Schema	Il affect all cher DBMS/Service SBDB1 SBDB1	cked rows in the ta Host Name blr00abl.in.orac blr00abl.in.orac	data sour able belov Port 1521 1521	Schema Ow RICS_STB RICS_WLS_RU	Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Edite	Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema WLS Schema OWSM MDS Schema	I affect all cher DBMS/Service SBDB1 SBDB1 SBDB1	ked rows in the ta Host Name blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac	data sour able belov Port 1521 1521 1521	Schema Ow RICS_STB RICS_WLS_RU RICS_MDS	Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Edite	Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema WLS Schema OWSM MDS Schema OPSS Audit Schema	I affect all chee DBMS/Service SBDB1 SBDB1 SBDB1 SBDB1	keed rows in the ta Host Name blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac	able belov Port 1521 1521 1521 1521 1521	Schema Ow RICS_STB RICS_WLS_RU RICS_MDS RICS_IAU_APP	Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration		Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema WLS Schema OWSM MDS Schema OPSS Audit Schema OPSS Audit Viewer Sc	Il affect all chee DBMS/Service SBDB1 SBDB1 SBDB1 SBDB1 SBDB1	keet to RAC multi cked rows in the ta Host Name blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac	data sour able below Port 1521 1521 1521 1521 1521	Schema Ow RICS_STB RICS_WLS_RU RICS_MDS RICS_IAU_APP RICS_IAU_VIE\	Schema Pass
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration		Convert to Gri s to the data above wi Component Schema LocalSvcTbl Schema WLS Schema OWSM MDS Schema OPSS Audit Schema OPSS Audit Viewer Sc OPSS Schema	dLink Cor DBMS/Service SBDB1 SBDB1 SBDB1 SBDB1 SBDB1 SBDB1	kert to RAC multi of cked rows in the ta Host Name blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac blr00abl.in.orac	data sour Port 1521 1521 1521 1521 1521 1521 1521	Schema Ow RICS_STB RICS_WLS_RU RICS_IAU_SRU RICS_IAU_VIE\ RICS_IAU_VIE\ RICS_OPSS	Schema Passi

8. Click **Next**. The JDBC Component Schema Test window displays the status on whether the JDBC tests on the schemas were successful.

DBC Component Schema	a Te	st			
Create Domain		Status	Component Schema	JDBC Connection UF	રા
Templates		1	LocalSvcTbl Schema	jdbc:oracle:thin:@// 🖣 🖬	
Application Location		1	WLS Schema	jdbc:oracle:thin:@// 🖬 🖬	1.1.1.4
Administrator Account		ø	OWSM MDS Schema	jdbc:oracle:thin:@//L 💶	6 . I. B.
Administrator Account		4	OPSS Audit Schema	jdbc:oracle:thin:@// 📲 🖬	9
Domain Mode and JDK	•	4	OPSS Audit Viewer Schema	jdbc:oracle:thin:@//h 📫 📥	1.1.1.4
Database Configuration Type		1	OPSS Schema	jdbc:oracle:thin:@//L	6. L. B.
End Of Configuration	Con	ponent	: Schema=LocalSvcTbl Schema		
Advanced Configuration Configuration Summary Configuration Progress End Of Configuration	Cor	Test Se nnection	lected Connections Cancel Testing n Result Log Schema=LocalSvcTbl Schema		
	URL	=jdbc:o			
	Use Pas SQL CFG	r=RICS sword= Test=: FWK-64	STB SELECT 1 FROM DUAL 213: Test Successfull 213: JDBC connection test was successful		1
	Use Pas SQL CFG	r=RICS sword= . Test=: FWK-64 PMK-64	STB SELECT 1 FROM DUAL 213: Test Successful 213: IDBC connection test was successful		

9. Click Next. The Advanced Configuration window displays. Select all the checkboxes, except the Domain Frontend Host Capture and JMS File Store options, in this window.



F	Fusion Middleware Configuration Wizard - Page 9 of 23
Advanced Configuration	
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Tyr Component Datasources JDBC Test Advanced Configuration Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets Partitions Deployments Targeting Services Targeting	<ul> <li>✓ Administration Server Modify Settings</li> <li>✓ Node Manager Configure Node Manager</li> <li>✓ Iopology Add, Delete or Modify Settings for Managed Servers, Clusters, Virtual Targets and Coherence</li> <li>Domain Frontend Host Capture Configure Domain Frontend Host</li> <li>✓ Deployments and Services Target to Servers or Clusters</li> <li>File Store Modify Settings</li> </ul>
Help	< <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel

**10.** Click **Next**. The Administration Server window displays. Enter the Listen Address and the Listen Port details.

Administration Server		i		
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Tyr Component Datasources JDBC Test Advanced Configuration Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets Partitions Deployments Targeting Services Targeting	Server Name AdminServer Listen Address All Local Addresses Listen Port 11001 Enable SSL SSL Listen Port Server Groups Unspecified	5535. and different fro	om SSL listen port and	coherence port



**11.** Click **Next**. The Node Manager window displays. Select the **Node Manager Type** and enter the **Node Manager** credentials.

1	Fusion Middleware Conf	figuration Wizard	- Page 11 o	of 23		_ ×
Node Manager			FU			
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Typ Component Datasources JDBC Test Advanced Configuration Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets Partitions Deployments Targeting Services Targeting	Node Manager Type Per Domain Default Lo Per Domain <u>C</u> ustom Lo Node Manager Home: <u>Manual Node Manager</u> Node Manager Credenti Username: Password: Confirm Password: Must be the same as the part at least one number or spe	cation ocation 12214/user_project Setup ials weblogic  assword. Password n cial character.	s/domains/rics,	_domain/nod	emanager	Browse aracters with
<u>H</u> elp			< <u>B</u> ack	Next >	Einish	Cancel

- 12. Click Next. The Managed Servers window displays.
  - a. Click Add to add a managed server on which you will deploy the application.
  - **b.** Enter the **Server Name**, **Listen Address**, and **Listen Port** for the managed server.
  - c. Set the Server Groups to JRF-MAN-SVR.

Managed Servers			i			$\bigcirc$
Create Domain Templates	🛉 🛃 📳 Cl	one 🔀 <u>D</u> elete			🧳 Dis <u>c</u> ar	rd Changes
Application Location Administrator Account	Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
Domain Mode and JDK	new_ManagedServer_1	All Local Address	11003		Disabled	Unspecifie
Advanced Configuration						
Administration Server Node Manager Managed Servers Clusters						
Administration Server Node Manager Managed Servers Clusters Server Templates						
Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters						
Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines						
Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets						
Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets Partitions						
Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines Virtual Targets Partitions Deployments Targeting						

- **13.** Click **Next**. The Clusters window displays.
  - a. Click Add to add a cluster. This is an optional step in the procedure.

F	usion Middlewar	e Configuration V	/izard - Page 1	3 of 23	
Clusters					
Create Domain	👍 Add 🗙	<u>D</u> elete		6	Dis <u>c</u> ard Changes
Application Location     Administrator Account	Cluster Name	Cluster Address	Frontend Host	Frontend HTTP Port	Frontend HTTPS Port
Domain Mode and JDK     Database Configuration Type					
<u>Component Datasources</u> JDBC Test					
Advanced Configuration					
Node Manager					
Clusters					
<u>Server Templates</u> <u>Coherence Clusters</u>					
Machines Virtual Targets					
Partitions	• •				
<u>Deployments Targeting</u> <u>Services Targeting</u>					
Help			< <u>B</u> ack	<u>N</u> ext > <u>Fini</u>	sh Cancel



- 14. Click Next. The Server Templates window displays.
  - a. Click Add to add a server template. This is an optional step in the procedure.

Coherence Clusters				
Administrator Account		Dis <u>c</u> ard Cha	anges	
Database Configuration Typ	Cluster Name	Cluster Listen Port		
Component Datasources d	efaultCoherenceCluster	7574		
JDBC Test				
Advanced Configuration				
Administration Server				
Node Manager				
Managed Servers				
Clusters				
Server Templates				
Coherence Clusters				
Machines				
Virtual Targets				
Partitions				
Deployments Targeting				
Services Targeting	-			
Configuration Summary				
Configuration Progress				

- **15.** Click **Next**. The Coherence Clusters window displays.
  - a. Add a coherence cluster. This is an optional step in the procedure.

Fusion Middleware Configuration Wizard - Page 15 of 23					
Coherence Clusters	Coherence Clusters				
Templates		🔊 Dis <u>c</u> a	rd Changes		
Application Location	Cluster Name	Cluster Listen Port			
Administrator Account	defaultCoherenceCluster	7574			
Domain Mode and JDK					
Database Configuration Typ					
<u>Component Datasources</u>					
UDBC Test					
Advanced Configuration					
Administration Server					
V Node Manager					
Managed Servers					
Clusters					
Server Templates					
Coherence Clusters					
Machines					
Virtual Targets	***				
Partitions					
Deployments Targeting					
Services Targeting					
Help		< Back Next > Finish	Cancel		



- **16.** Click Next. The Machines window displays.
  - a. Click Add.
  - b. Enter the Name and the Node Manager Listen Address for the managed server.

F	usion Middlewar	e Configu	ration Wiz	ard - Pa	ge 16 of	24	_ ×
Machines					FUSIC		
Administrator Account Domain Mode and JDK Database Configuration Typ	Machine Unix M	achine <mark>( D</mark> elete				🔊 Dis <u>c</u> a	ard Changes
<u>Component Datasources</u> JDBC Test	Name	Enable	Post Bind GID	Enable	Post Bind UID	Node Manager Listen Address	Node Manager
Advanced Configuration	new_UnixMachine_		nobody		nobody	localhost	▼ 5556
Administration Server							
Node Manager							
Managed Servers							
Clusters							
Server Templates							
<u>Coherence Clusters</u>							
Machines							
Assign Servers to Machines							
Virtual Targets							
Partitions							
Deployments Targeting							
Services Targeting							
Configuration Summary							
↓ Configuration Progress 🚽							
Help				< <u>B</u>	ack N	ext > Einish	Cancel

**17.** Click **Next**. The Assign Servers to Machines window displays. Add the Admin Server and the managed server to the computer.



	usion muddeware configuration wizard i ra	lge 17 of 24
Assign Servers to Machir	nes	
Administrator Account Domain Mode and JDK Domain Mode and JDK Database Configuration Tyr Component Datasources JDBC Test Advanced Configuration Administration Server Node Manager Managed Servers Clusters Clusters Server Templates Coherence Clusters Machines Assign Servers to Machi Virtual Targets Partitions Deployments Targeting Services Targeting Configuration Summary Configuration Summary	Servers	Machines UnixMachine Mew_UnixMachine_1 AdminServer new_ManagedServer_1 machine in the right pane. Then use the right e machine.

18. Click Next. The Virtual targets window displays.

a. Click Add to add a Virtual target. This is an optional step in the procedure.

Fu	sion Middleware	Configuration V	Vizard - Page	18 of 24		- 3
Virtual Targets						
Create Domain Templates	👍 Add 🛛 💥	Delete			Dis <u>c</u> ard (	Changes
Application Location Administrator Account	Name	Target	Host Names	URI Prefix	Explicit Port	Port Offset
Domain Mode and JDK Database Configuration Typ						
Component Datasources						
Advanced Configuration						
Node Manager						
<u>Managed Servers</u> <u>Clusters</u>						
<u>Server Templates</u> <u>Coherence Clusters</u>						
Machines						
Virtual Targets	*					
<u>Partitions</u> <u>Deployments Targeting</u>						
Help			< <u>B</u> ac	k <u>N</u> ext >	Einish	Cancel

- **19.** Click **Next**. The Partitions window displays.
  - a. Click Add to add a Partition. This is an optional step in the procedure.



Fusion Middleware Configuration Wizard - Page 19 of 24				
Partitions	and the second se			
Templates	Add Nelete	<b>\$</b>	Dis <u>c</u> ard Changes	
Application Location		Name		
Administrator Account				
Domain Mode and JDK				
Database Configuration	Exe			
Component Datasources				
UDBC Test				
Advanced Configuration				
Administration Server				
Wode Manager				
Managed Servers				
Clusters				
Server Templates				
Coherence Clusters				
<u><u>Machines</u></u>				
Assign Servers to Machin	net			
Virtual Targets	20			
Partitions     Deployments Targeting	<b>.</b>			
Help		< Back Next > Eini	sh Cancel	

20. Click Next. The Deployments Targeting window displays. Select wsm-pm from Deployments and add it to Admin Server in Targets.





21. Click Next. The Services Targeting window displays.



22. Click **Next**. The Configuration Summary window displays. Verify that all information described in this window is accurate.





23. Click **Create**. The Configuration Progress window displays a message when the domain is created successfully.

	usion Middleware Configuration Wizard - Page 23 of 24	_ ×
Configuration Progress		
Create Domain Templates Application Location Administrator Account Domain Mode and JDK Database Configuration Tyr Component Datasources JDBC Test Advanced Configuration Administration Server Node Manager Managed Servers Clusters Server Templates Coherence Clusters Machines	IO0%         IO0%         OPSS Processing         OWSM Processing         Eccurity Processing         Artifacts Generation         String Substitution         Post Processing	
Assign Servers to Machine     Virtual Targets     Partitions     Deployments Targeting		
Help	< Back Next > Finisi	Cancel

24. Click **Next**. The Configuration Success window displays that describes the Domain Location and Admin Server URL once the configuration is complete.



F	24 of 24	1 in [	
End Of Configuration			
Create Domain Templates Application Location Administrator Account	Oracle Weblogic Server Configuration Succeet New Domain Int_domain Creation Succeeded Domain Location Admin Server URL	ded	
Domain Mode and JDK     Database Configuration Typ     Component Datasources     JDBC Test     Advanced Configuration     Administration Server			
Vode Manager Managed Servers Clusters Server Témplates Coherence Clusters			
Machines     Assign Servers to Machinet     Virtual Targets     Partitions     Deployments Targeting     Help	•	ack Next > Finis	Cancel

25. Click Finish to complete creating the WebLogic domain and managed servers.

#### Steps for ear Deployment

- Client connector pak contains the latest v24 rib-rwms application distribution for on-prem installation. Download and extract the RIB kernel for RMWS-secondaryapp RibKernel24.0.000ForRwmsSecondary24.x.xApps eng ga.jar.
- 2. Extract the contents of the jar file.
- Open rib-deployment-env-info.xml found inside ./rib-rwms-secondary-home/ deployment-home/conf.
- 4. Edit this file to specify your deployment environment information.
  - a. Make sure the following entries are present in the <app-in-scope-forintegration> section:

<app id="rwms" type=" slave-plsql-app" />

b. Update the rib-jms-servers section to provide the AQ JMS server details. Because the secondary app deploys on premise, it will not have access to AQ JMS on the cloud. Use RWMS app schema detail for AQ JMS setup. For example:

```
<aq-jms-server jms-server-id="jms1">
<jms-server-home>ribadmin@jms1host.example.com:/u00/oracle/product/11.2.0.2</jms-server-home>
<jms-url>jdbc:oracle:thin:@rwmsappdbhost:1521/service_name</jms-url>
<jms-port>1521</jms-port>
<jms-user-alias>jms1_jms_user-name-alias</jms-user-alias>
</aq-jms-server>
```

c. Update the RIB domain details in the weblogic-application-servers section.



- d. Skip updating the rib-func-artifact-server details. Rib-func-artifact deployment is not required for secondary (on-prem) rib-rwms.
- e. Update RIB-RWMS secondary server details. For example:



f. Make sure the datasource URL (host, port n service) entries are updated in the ribapp section of rib-rwms secondary.



#### Note:

As the secondary app deploys on-premise, it will not have access to AQ JMS and Error hospital. Therefore, all the datasources must connect to the RWMS app schema.

 Compile: Run the rib-home/application-assembly-home/bin/rib-app-compiler.sh script with setup-security-credential from the rib-home/application-assembly-home/bin directory.

Example:

```
./rib-app-compiler.sh -setup-security-credential
```

6. Deploy: Execute the rib-home/deployment-home/bin/rib-app-deployer.sh script with the appropriate command line parameter.

```
rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
rib-func-artifact deployment is not required.
```



7. Verify: Once the rib-rwms secondary app is deployed, open the rib-admin-gui from a web browser using the credentials provided during compilation:

```
<http or https://>host:port/rib-rwms-admin-gui
```

8. Make sure the Publication and Subscription WS are available to use.

Example:

```
https://ribhost.example.com:17010/
RemotePlsqlPublisherComponentServiceBean/
RemotePlsqlPublisherComponentServiceBeanService?WSDL
https:// ribhost.example.com:17010/
PlsqlApplicationMessageInjectorServiceBean/
PlsqlApplicationMessageInjectorServiceBeanService?WSDL
```



# 4 RIB Self-Service Enablement

The Self-service enablement is a feature for provisioning RIB on cloud post deployment only. Because of the promising high availability feature of applications on the cloud environment, this is an essential feature that minimizes the redo of the RIB install cycle post configuration changes to any RIB-app.

The Self-service enablement allows below provisioning in rib-<app>:

Self- Service Feature:	Self Service Feature on RIB-Admin GUI	
Provisioning RIB adapters Choosing the subset of RIB	EXAL For two performances The advances of the second sec	Winner, Westmanner, State State of State S
adapters in scope for integration	Ye B S	

Table 4-1 Self-Service Feature



Self- Service Feature:	Self Service Feature on RIB-Admin	GUI	
Provisioning System Options	rite-est-System Options		
Dynamically	Note Associational Locatories (18.200 Newsyo Configuration (18.500)	Fage Deliverhead Plead Mar 12.1	IST TO THIS DRY ONLY COMMON
modifying	System Reporte Mainter, Reporter		
mounying	KB Properties		
configuratio	anthAltering	specialized and implifi	914
ns via, rib-	i dinaki kali kari kang kang kang kang kang kang kang kang	Contarrer Just Conterner	1812
<app></app>	andreity and the protocol and an	P-6	812
properties	Symposity and a star-meaninger-of-speed	NUMBER OF STREET	(*1*
pioperties	Symposition and any measure of special	DOVUMUP ITANIARP	012
file.	for Children, suite lange removagers of lypers	FOFULAD	0117
Evample	for PE of any weak attent management of open for Teambers, and attent management of open	The second of th	212
	for werder, auto-drop-messages influges	VENDOWOUNDP	211
shown for	to waterable and the nearest of test	1010.00	1912
rib-ext for			
dropping			
aropping	Real Nobel Robes		
messages	ter bestimm acteur blue bestimmer illington	hark	Construction and the Construction of the
for specific	The second se		
types for	INTERVENT CONTRACTOR OF CONTRA	COLUMN C	2(7)
subscription	and an and a feet of the second s	Bateria paramete	011
Subscription	And an an a straight from the second se	An Addressed	214
s. Similarly,	Sector and the party inclusion of the sector and the se	Concept	817
the drop	A service of the serv	of residences	817
ancesem	la facilitado pol especial tem Interferenciam con especial pom	Selection and Selection of Sele	217
timessages	National Andrew Street	3	212
types can	The set of the second	The Art of	217
be		and programmers and the second s	917
configured	ter angen ter	A CONTRACTOR OF CONTRACTOR	917
for other DID	Balance and Bala	the second restore	518
	and and a second	100 000 000 000 100	917
applications	anti Antima di	-	912
like rib-sim.	an contrate support an existing come	Latriana and Art constant	0718 0710
rib_rme and	in American And American	Harry Condition	918
no-mis anu			
so on.			
Note: There			
are other			
infraction			
Infrastructur			
e level			
options that			
ara availabla			
ale available			
only for			
AMS or			
devons			
uevops			
teams to			
configure or			
undate as			
upuaie, as			
snown in the			
screenshot.			

 Table 4-1
 (Cont.) Self-Service Feature



Self Service Feature on RIB-Admin GUI
mail.smtp.fom     admin@oracle.con     @   2       mail.smtp.nost     mail-couler-svc.rgbu-rex.ex     @   2       mail.smtp.not     2501     @   2       mail.smtp.to.list     recipient@example.com     @   2
Pierter         Rest         Rest

 Table 4-1
 (Cont.) Self-Service Feature



Self- Service Feature:	Self Servio	Self Service Feature on RIB-Admin GUI						
RIB ServiceMoni tor Verify	ORACLE ADM. Indiacymic ba Reege							
InjectorServi	rib-ext:File Services Realth D	arth Manon Willion 1	Manual Confidentiation		Page Robertsol We	4 May 20 (1982) (19 (19	D DAT-4830 (mile Namlari	1744
ce	NE this service accessibility are written	NE Ohr under somstellig som verfar hans forar för states ofstigkent anti-services by pilys indep fram .						
	Service Rama	Resulty Palloy		Mas	Proj.	hatus	Passariate	
provisioned in previous	red	proc.	Ny, nyivele del 2014 control del 2016 Epite acceso all'inserante typita typit	10 mil. nr. 10 mil. 10 mil. 10 mil. 10 mil.	296			-
step.	Harton Adapted Research La	stanger Hillion	Manage Configurations   100 Service address					

Table 4-1 (Cont.) Self-Service Feature

## **Provisioning RIB-Adapters**

Every rib-<app> contains a set of publish and subscribing adapters for exchanging messages between retail applications. Subscribing adapters are MDB which are resource intensive. The higher the number of adapters in scope the higher is the resource crunch. In an environment which does not make use of all the publishing and subscribing adapters bundled with the rib-app, the user is allowed to choose a subset of the adapters needed based on the RIB functional flow. This configuration change takes effect dynamically and does not require a redeployment of the rib-app>.

Follow the steps below for configuring the rib-<app> adapters in scope of the integration.

1. By default, dynamic adapter selection feature is enabled for rib-ext. enableDynamicAdapterInstanceSelection flag is applicable ONLY for RIB-EXT and this flag shouldn't be used in any other rib-apps.

enableDynamicAdapterInstanceSelection=true

2. Only if the above property is set to true, the user can select the adapters dynamically. Below is the default landing page when RIB adapters added in scope.

RETAIL Retail Integration Dos Harager					
				Welcome, ribed	inin Legat
rib-sim:RIB Adapter Manager		Page Refeated	Non Oct 09 2017 11:56:53	GBIT+0530 dedia Star	dard Time).
Home Adapter Manager Log Manager BIE Lags Manage Configurations					
This page shows the RD Adapters (sublishers, subscribers, Salts and/or hospitals) deployed on this RD instance. View (A)  (Ranoval Data) (Ranoval Data)					
Expand All I Callapse All		Read Manual	1112 5	Fill Barrowski	
Select Name	Status	Start Time	JMS Server ID	Edit Properties	View Log
V Subscribers					
Request Publishers					
Hote Adapter Menoper Log Manager SBLLags Manage Configurations Copyright 8:217, Once and/or to atlases. All optics sources					



 In the RIB-Admin GUI, the Manage Configuration > Adapter Selection tab provides the list of all available adapters whose subset can be chosen to publish, subscribe and retry rib messages based on rib integration flows.

ORACLE	1
RETAIL Rotal Integration Bas Manager	Walcone diadain Locat
rib-alm:Adapter Selection	1111111, 100001 10000
Page Re	freshed Mon Oct 09 2017 12:29:88 GMT-8530 (India Standard Time).
Harne Adapter Manager Log Manager Bill Logs Menage Configurations	
System Options Intentor Service Adapter Solection	
This may show all the DBL Advances (sublishess subsections table models beneficial waithing one can sub-the advances from list	
The page mosts at the Kol Happen (possible), subscripts and integrant) and any for the adapters from the	
Yes A *	
(Rest) (Care)	
Swand All College All	
Contraction Technique Per	Select Adaptor
V Subactibers	
ASIm sub	
CMPrcDrg_sub	
Diffs sub	
OleS8 seb	8
FullOrd seb	8
herio: mb	
berg sub	
Merchiller sub	8
Order sub	8
Patter sub	8
PmPicDig.wb	8
RTvReg aub	
ReUnitAdi, sub	8
RepProChg sub	8
SOStates sub	8
SeedData_sub	
StockOrder sub	

4. Select the subset of publishing, subscribing and retry adapters depending on the ribintegration-flow in consideration and click **Save**.

Consider the below rib-integration flows:

rib-sim publishing the InvReq message

```
<message-flow id="31">
             <node id="rib-sim.InvReg pub" app-name="rib-sim"
                   adapter-class-def="InvReq_pub" type="DbToJms">
                    <in-db>default</in-db>
                    <out-topic>etInvReq</out-topic>
             </node>
             <node id="rib-ext.InvReq pub" app-name="rib-ext"
                   adapter-class-def="InvReq pub" type="DbToJms">
                    <in-db>default</in-db>
                    <out-topic>etInvReq</out-topic>
             </node>
             <node id="rib-rms.InvReq_sub" app-name="rib-rms"
                   adapter-class-def="InvReq sub" type="JmsToDb">
                    <in-topic>etInvReq</in-topic>
                    <out-db>default</out-db>
             </node>
             <node id="rib-ext.InvReq_sub" app-name="rib-ext"
                   adapter-class-def="InvReq sub" type="JmsToDb">
                    <in-topic>etInvReq</in-topic>
                    <out-db>default</out-db>
             </node>
      </message-flow>
```

#### rib-sim subscribing the ItemLoc message from RMS



```
<node id="rib-ext.ItemLoc pub" app-name="rib-ext"
             adapter-class-def="ItemLoc pub" type="DbToJms">
              <in-db>default</in-db>
              <out-topic>etItemLocFromRMS</out-topic>
       </node>
       <node id="rib-sim.ItemLoc sub" app-name="rib-sim"
             adapter-class-def="ItemLoc sub" type="JmsToDb">
              <in-topic>etItemLocFromRMS</in-topic>
              <out-db>default</out-db>
       </node>
       <node id="rib-rwms.ItemLoc sub" app-name="rib-rwms"
             adapter-class-def="ItemLoc sub" type="JmsToDb">
              <in-topic>etItemLocFromRMS</in-topic>
              <out-db>default</out-db>
       </node>
       <node id="rib-ext.ItemLoc sub" app-name="rib-ext"
             adapter-class-def="ItemLoc sub" type="JmsToDb">
              <in-topic>etItemLocFromRMS</in-topic>
              <out-db>default</out-db>
       </node>
</message-flow>
```

Considering the above flows, select **InvReq\_Pub**, and **ItemLoc\_sub** and both **Hospital** adapters as shown in the image below.

Harro	Select Adapter
V Bubschbers	
ASNr_ub	
CiPrtChg_sub	8
Diffs_sub	8
DitySit_sub	10 C
Fallind_sub	10 M
henLoc_sub	(X)
Berra_sub	8
Mexhiler sub	8
Onder sub	8
Patrar_sub	
PrmPrcChg_sub	10 C
RTVReq_sub	8
RevUnitAdj_sub	8
RegPicChg, wb	
SOSiatun_sub	
SeedData_sub	8
StackOnter_sub	8
Stares_sub	8
LCAs_sub	8
Vender_aub	
WH_sub	<b></b>
▼ Haspitals	
jans, hosp	2
sub_hosp	52
V Publishers	
ASNOw, pub	2 C
DSDReceipt_pub	
FallOrdOmCrc_pub	8
FallOrdOm_pub	8
Invitational pub	8
Invited top	2
ProChgReq_pub	
RTV_pub	
Receiving pub	8

5. Verify that the selected adapters are reflected on the Adapter Manager tab. Newly added adapters in scope will be down. Newly added adapters in-scope need a start from the GUI to become ACTIVE as a one time activity; otherwise, newly added adapters won't show up on topic on checking from jms-console and won't even be registered (messages will be lost). Sometimes you need to start adapters 2-3 times because of one known issue where the subscriber registration process is times out. Post start of newly added adapters, ensure adapters are showing up on topic on checking from jms-console. If newly added adapters are not showing up on topic, please try to start them again from the UI. jms-console will not show adapters on topic immediately and there is expected 3 to 5 mins of delay.



RETAIL RETAIL Retail Integration B	a Nonger					
					Welcome, riba	denin Logant
rib-sim:RIB Ada	apter Manager		Page Exhested	las Oct 09 2017 12:54:21	GBIT+0530 (India Sta	edaed Time).
Home Adap	Ner Manager Log Manager RIB Logs Manage Configurations					
This page shows the l View All	This page shows the RIB Adapters (publishers, subscribers,tails and/or hospitals) deplayed on this RIB instance.					
[104] [200						
Expand All I Cellapse	<u>Al</u>	Anne -	davet Times	HAR COMPANY ID	E 45 December	Marca Law
Sefect	V Subscribers	Stotus	Stort Time	JWS Server ID	Edit Properties	View Log
1	ItemLec Subscriber, channel 1	0	Mon Oct 09 11.56.49 IST 2017	jms1		E)
	Thospitals					
	JMS Hospital Retry	÷	Mon Oct 09 11:56:18 IST 2017	imi.	5	B/
11	SUB Hospital Retry	0	Mon Oct 09 11:56:16 (5T 2017	inst	E,	E,
Request Publishers						
	In/Reg Publisher, channel 1	÷		Leni,		5
Harres Adapter Manager Log Manager BB Logs Manage Configurations						

6. All the adapters are in scope by default for rib-<app>:

enableDynamicAdapterInstanceSelection = false

This is the default value for all rib-<app>s except rib-ext for rib-ext following flag is set to true

enableDynamicAdapterInstanceSelection = true

#### Note:

enableDynamicAdapterInstanceSelection flag is not available for end user update. Follow the steps in the next section to disable this flag for other rib-<apps> in case they are enabled.

### How to Remove Dynamic Adapters Selection in RIB-RMS

The concept of Dynamic Adapters Selection applies only to the RIB-EXT application and all other RIB-<apps> such as RIB-RMS, RIB-SIM, RIB-TAFR etc do not support the dynamic adapters. Due to our documentation defect, which has been fixed now, some of our customers have used this feature in non RIB-EXT apps, especially in RIB-RMS, which is unsupported and can cause major issues such as messages piling up on JMS topics and slow down the entire system. customers should remove dynamic selection of the adapters in any rib-app they might have configured it in ex: RIB-RMS/RIB-TAF/RIB-SIM

Steps :

- 1. Log into the RIB-RMS Admin GUI.
- 2. Go to the Adapter Manager page and capture the list of adapters present on the page.
- 3. Go to Manage Configurations -> Adapter Selection and select all the adapters.
- 4. Click **Save**. Make sure all the adapters are displayed in the Adapter Manager page.



rib-rms:Adapter Selection Home Adapter Manager Log Manager RIB Logs Manage Configurations RIB ServiceMonitor	Page Refreshed Fri Apr 19 2024 15:53:01 GMT+0530 (India Standard Tir
System Options Adapter Selection	
This page shows all the RIB Adapters (publishers, subscribers,tafrs and/or hospitals) available,user can pick the adapters from list. View (All )	
Save Cancel	
Expand All   Collapse All	
Name	Select Adapter
▶ Subscribers	
▶ Hospitals	
▶ Publishers	
System Options         Adapter Selection           Home         Adapter Manager         Log Manager         RIB Logs         Manage Configurations         RIB ServiceMonitor	

 Go to Manage Configurations -> System Options and set enableDynamicAdapterInstanceSelection to false.

enableDynamicAdapterInstanceSelection	enableDynamicAdapterInsta	false		2
Co to the Adapter Manager pa	ae and bring down a	II the adaptors th	at do not	

6. Go to the Adapter Manager page and bring down all the adapters that do not belong to the list collected in step 2.

# **Provisioning System Options**

Application specific properties for the rib-<app> are configured in the rib.properties file. When RIB is deployed on cloud, the application specific properties can be configured in the RIB-Admin GUI application. The Manage Configuration > System Options tab allows the user to edit the properties values post deployment. There are some infrastructure level options that are available only for AMS or devops teams to configure or update.

CRACLE RETAR Partial long particular Ban Manager		Webcome, How admin. Lapool
rib-sim:Bystem Options		
Harse Adapter Manager Loc Manager 1983 Loca Manage Configurations 808 ServiceMonfor	Pe	a Trafficiation' Wed Mar 23 2022 13:48:31 GMT+0638 (India Standard Time).
System Options Interfer Service		
Rib Properties		
ts/lis/gpft satisf	true	01 0
aber Publisher Ingl	converse distances	(g   c
devination.extail app.Type	rest-upp	@1#
enable global arnal alort	Tahu	@1#
enableDynamicAdapter/milarce/Selection	Yahee	@1#
tor StockCroter_sub-drop messages of types	SOPARP	@1#
for Vendor_sub-drop-messages of types	VENDORPULARP	@1#
hospital attempt clolary	10	@1#
hospital attempt delayincrement	10	@   #
harqəlal aflamşi max	5	Q1 #
agentica service app Type	rest-opp	@1#
injactor service endpoint un	http://wtse-exc.rgbu-rec	014

Following are the frequently configurable RIB properties:

1. Drop-messages-of-types- for dropping messages for specific types for subscriptions.

for StockOrder_sub.drop-messages-of-types	SOFULREP	212
for Vendor_sub.dtop-messages-of-types	VENDORFULREP	81\$



2. Updating facility\_id and facility\_type for rib-tafr.

facility_id PROD 1	3	815
facility_id PR00.2	1	210
facility_id PROD-3	1	81:
taciity_type default	PROD	810

- 3. A new system option can also be added using 'Add' functionality in UI. Perform the following steps to add the Facilities for rib-tafr.
  - Click the **Add** button.

Rib Properties		
IsRitzAppErsabled	true	214
and a strategy of the	Carl Print, Po. and Trip.	815
<ul> <li>Insert a new Facility ID.</li> </ul>		
For example: key - facility	_id.PROD.12345 value - 1	
Add new property here	fadilty_id.PROD.12345	
Updating injector service url an	nd policy for rest-app.	
injector service endpoint un	http://who-swi.sglucces	21=
injector service security policyname	policyC	@1#
Updating IDCS host URL. This	is needed only for customer own	ed applications using

Oauth for rest call.

# **Provisioning InjectorService URL**

cauth2.dolault.authorizationServerUnl

4.

5.

In the RIB-Admin GUI, the Manage Configuration > Injector Service page allows the user to configure an injector service URL for a customer-owned applications.

https://kks-e99b3f8b00

RETAIL Retail Innegration Bus Manager	Welcome, ricecteadmin, Locou				
ribest Injector Service					
	Page Refreshed Thu Oct 26 2023 10:25:11 GMT+0530 (India Standard Time).				
Home Adapter Manager Log Manager RIB.Logs Manage Configurations RIB.ServiceMonitor					
System Options Injector Service Adapter Selection					
Configure Injector Service WSDL URL (Applicable to SOAP implementation of injector service)					
Current Injector Service URL http://examplehost.com/25704/rb-injector-services-web/nsocurces/injector/inject					
Update connection details for SOAP Injector service					
New Injector Service Host* examplehost.com					
New Injector Service Port* 25704					
New Security Policy Name* policyC ~ Help					
Configure Security details (Applicable for both SOAP/ReST implementation of Injector Service)					
Update security details					
Secured User Alias (nb-ext_ws_security_user-name-alias ) Help					
Secured User Name*					
Secured User Password*					
(Save) Cancel					
System Options Injector Service Adapter Selection					
Home Adapter Manager Log Manager RIB Logs Manage Configurations RIB. ServiceMonitor					
Copyright © 2021, Oracle and/or its affliates. All rights reserved.					



812

Update injector service URL details by providing new host and port details and the user credentials for the service.

## **RIB ServiceMonitor**

Once the RIB integration environment is configured for use by various retail application, as a sanity test the user may need to verify the integration end points. For RIB on cloud, we can ping-test various webservices consumed by RIB using RIB admin GUI.

In RIB Admin GUI, the RibServiceMonitor page lists all the webservices consumed by the rib-application and allows the user to ping the same. The webservices are pingable only if the "ping" operation is supported by the webservice.

ETAL Batal Desgrades Des Manager							
						Welcome, rice admin 1/2	-
rlb-ext:Rib Services Health Check Page Fasheded Med Mar 20 2022 16 Sicile GMT+6638 (India Standard Time).							
Home Adopter Manager Log Manager	RELED Ma	napo Configurations RIB ServiceManitor					
RB Web service accessibility can verified here Kry	w the status of displa	yed web services by ping testing them .					
ServiceName	SecurityPolicy	UNL.	Alles	Fire	Status	ReasonCode	
ryect	poleşA.	htps://egbu-ptse/baot-351.as.oracia.cam-443/rb- injecto-anvions-web/rescarces/injecto/inject	rb-ext_ws_security_care rame-alan	Ping			4
Home Adapter Manager Log Haras	er Riblass	Hanage Configurations RBB Service-Attention					
Home         Advances Management         Loss Management           R83 Web service accossibility can worthed here 90%         Remote Management         Remote Management           report         Loss Management         Loss Management         Remote Management           Etherage         Advances Management         Loss Management         Remote Management           Etherage         Advances Management         Loss Management         Remote Management	r BBLoop Mar write status of depa Recentlyfisliny policyA	son Configuration     RB ServiceManifor     ord exit services by ang Techng Texn .     URL     Miga Inglau che-Beol 351 as neck cam 40mb-     ingcar-services exit resources with tender     Homose Configurations     RB ServicesManifor	Alles rib-est_so_secutity_saw reme alles	Ping	Status (2)	Researcove	



# 5 Performance

# **Performance Factors**

The performance of each of these components is influential in the overall performance of the system:

- The application server(s) topology and configuration.
- The RIB deployment approach.
- The hardware sizing and configuration of the RIB hosts.
- The hardware sizing and configuration of the applications that are connected to the RIB.
- The hardware sizing and configuration of the JMS provider host.
- The hardware sizing and configuration of the RIB Hospitals hosts.

There are other factors that determine the performance of the overall system. Some of these factors in a RIB environment are:

- Number of channels configured
- Number of messages present in the topic
- Size of the message
- Database clustering
- Application Server topology
- Number of TAFRs in the processing of the message
- Message aggregation

See "Performance Considerations" in the Oracle Retail Integration Bus Operations Guide.

#### Note:

For more information, see "Performance Considerations," in the Oracle Retail Integration Bus Operations Guide.

### Performance and Parallel Logical Channels

The RIB must provide guaranteed once and only once processing of business events (messages) across the enterprise. Maintaining the order of business events across the enterprise is critical to data integrity.

To provide guaranteed sequencing of message processing, RIB requires a guaranteed first in, first out (FIFO) messaging system with guaranteed FIFO rollback. That is, when you rollback the message from the consumer you get the same message back the next time so



that it is processed in sequence. JMS Provider provides this FIFO topic and FIFO rollback capability, which enables RIB to guarantee message sequencing.

Processing messages in sequence results in operational overhead, as every message must be checked against the database to find the status of previous messages on which it is dependent (same businessObjectid). Sequencing creates an inherent bottleneck, in that only one message is processed at once. For example, messages can come at the rate of 100 messages per second, but a RIB subscribing adapter can process only one of those messages at a time to preserve the order. To get around this bottleneck and improve performance, RIB provides options for optimization and functionality.

First, RIB processes messages in sequence only when the publishing application wants it to be processed in sequence. The message producer application defines a businessObjectid whose existence informs RIB that this and all subsequent messages with the same businessObjectid have to be processed in order.

Second, parallel logical channels can be created for each message flow paths in the integration system to improve performance. Parallel logical channels are virtual logical message flow paths within the same physical JMS topics. To add additional channels, each adapter participating in a message flow must be configured with additional adapter instances.

Using parallel logical channels is not the solution for all performance problems in the integration system. They can help only when the API for the corresponding applications is written with non-locking logic and concurrency invocation in mind.

Generally, integration for the retail application APIs are the biggest factor for bottlenecks in the overall messaging system throughput. It is not appropriate to start creating parallel logical channels at the first sign of performance problem. It is important to analyze and tune the integration APIs of the retail applications before considering the use of parallel channels.

Using parallel logical channels increases complexity, CPU demands, and memory requirement, resulting in more operational overhead. Use them only when, after all other components are fully tuned, you are still not able to meet your target numbers.



# o Security

Security in the integration layer is a big concern for every retail enterprise. The security system should be open enough to allow trusted remote applications to integrate easily and, at the same time, lock down unauthorized remote access. To address security concerns, RIB utilizes the security modules available in the Oracle middle ware and database systems.

There are two categories of administrators in RIB: RIB System Administrators and RIB Application Administrators. RIB System Administrators are involved in installing, configuring, deploying defect fixes, and making sure that the integration infrastructure is up and running properly. They generally are concerned with the business side of the integration system. Their tasks include bringing up or taking down RIB adapters, and fixing data issues with message payloads using RIHA. There are separate realms, roles, groups, and users defined for each category of RIB administrators.

# **RIB Application Administrators Security Domain**

For each rib-<app>.ear deployed, RIB creates the users belonging to the below groups:

- RicsAdminGroup
- RicsOperatorGroup
- RicsMonitorGroup

The default groups and user that RIB creates must not be deleted or modified.

RIB follows a role-based authorization for allowing valid users to perform a defined set of operations from the rib-admin-gui. The user belonging to each of above groups will be associated with a well defined role and thus able to perform authorized operations only. It is recommended that you have a unique user belonging to each group.

### Integration with SIOCS

- RIB will use IDCS OAuth2 for authentication of ReST calls both inbound and outbound (publisher/injector restful services). The primary authentication mechanism in the cloud is OAuth2 using the IDCS authenticator. Out-of-the-box configuration expects OAuth2 to be used.
- 2. RICS to EICS integration will be a ReST call with OAuth2.
- The EICS injector URL will be auto-wired as part of RICS provisioning. URL will look something like:

```
http://wtss-svc.<SIOCS_SUB-NAMESPACE>.svc.occloud:9999/siocs-int-services/
internal/api/inject
```

4. The RICS IDCS Client ID and Secret are auto-wired with ribsim\_oauth2\_application\_client\_user-name-alias as part of provisioning. These will be used to get the access token for accessing EICS end point.



Note: rib-sim\_ws\_security\_user-name\_alias is for BasicAuth and should be set empty for OAuth2 however auto wiring takes care of setting this alias to empty.

5. IDCS Url is also auto-wired, and is set during RICS provisioning. The URL looks something like:

https://idcs-<TENANT>/oauth2/v1/token

Step	Comment
Access rib-sim admin GUI at https://	
<external-load-< td=""><td>Welcom</td></external-load-<>	Welcom
balancer>/ <sub- namespace&gt;/rib-</sub- 	rib-sim:Home Puge Refrahed Tur Jul 25 2023 18:57:49 GMT-0530 (Inc Home Adatatr. Manager Log Manager BBLogs Manage Configurations RB ServiceMonitor
sim-admin-gui	Status Up Saur Time Fri Jul 21 13:23:06 UTC 2023 Up Time 45482440 ms Host
	Home Adapter Manager Log Manager RIB Logs Manage Configurations RIB ServiceMonitor
	Copyright © 2021, Oracle and/or its affiliates. Al rights reserved.



#### Step Comment Navigate to Manage Configurations-> System Options. Search and verify the following system options: injector.service.a a. ppType : rest-app b. Check the injector.service.e ndpoint.url. URL should be something like: http://wtss-918 912 912 912 912 912 912 912 912 912 svc.<SIOCS SUB-NAMESPACE>.svc. occloud:9999/ siocs-intservices/api/ ribinjector/ inject Look for C. injector.service.se

- c. Look for injector.service.se curity.policyname, policy should be policyC for internal calls.
- d. oauth2.default.aut horizationServer Url : RICS IDCS Host for making call to get the access to-ken.

Step		Comment						
Na Co Inje Ver	vigate to Manage nfigurations-> ector Service. ify the following:							
a.	Current Injector Service URL : should point to correct injector service url.							
b.	rib- sim_ws_security_ username_alias credential should be empty.	Update security details Secured User Alas Secured User Name* Secured User Password*	(rb-sim_ws_security_use	r-namo-alias	Sh	Hale ow Useman	ne	
c.	rib- sim_oauth2_appli cation_client_use r-name-alias credential must be getting populated with client ID and secret.							
Ho the and cor	w to verify whether SIM injector URL d credentials are rect.	Ro-sim/Site Services Reads Check Here: Adecidence: La Manage 198 Jun	Menor Leffanders - Mil Invisionarium		Page Robotics New	Nar 28 2003 10 10 10	iki silin pala ka	-
Na Se	vigate to RIB rvice Monitor Tab	ethints areas accessed; so antar has the star in bacadees have a second star appel price	teplapat web carries to pay solid fram , y me, We lister our spin we spin of the inclusion colored 1990/0000 of service.mission despetialized	tan 1919, p. pada, ne new dia	ing .		-	
a.	Click <b>ping</b> to test the connectivity.							

#### Integration with ROB

- 1. RICS to ROB integration is Rest call, Oauth2 Authorization.
- 2. The integration is configured between ROB and RICS via the ReST service (which is HTTPS).
- 3. ROB injector URL looks something like this:

https://<external-load-balancer>/<rob-sub-namespace>/rib-injector-servicesweb/orcos/resources/injector/inject

- 4. The OB IDCS app Client ID and Secret will be used to get the access token for accessing ROB end point.
- 5. The IDCS Url is set during RICS provisioning. The URL looks something like:

https://idcs-<TENANT>/oauth2/v1/token



#### Note:

<code>rib-rob\_ws\_security\_user-name\_alias</code> is for <code>BasicAuth</code> and <code>should</code> be set empty for <code>OAuth2</code>

Table 6-1 Integrating with ROB

Category	Steps	Comment	
Access RIB Admin GUI	Access the rib admin GUI at https:// <external load-balancer&gt;/rib-rob- admin-gui Log in with the admin user.</external 	Image: State of the state o	
Verify Configuratio n and	Navigate to Manage Configurations -> System options	Millioner         Millioner         012	-
update	<ol> <li>Search for and verify the following destination.retail.a ppType: rest-app</li> </ol>	Image: Section of the section of t	
	2. Check the value fo InjectorService URL (injector.service.en dpoint.url).URL should look something like this https:// omni.retail.us- phoenix-1.ocs.oc- test.com/rgbu- omni-rgbu-stg83- obcs/rib-injector- services-web/ orcos/resources/ injector/inject	r	
	<ol> <li>Security Policy (injector.service.se curity.policyname): policyA</li> </ol>		
	<ol> <li>IDCS OAuth Server URL (oauth2.default.aut horizationServerUr ): https://<idcs- tenant&gt;/oauth2/v1/ token</idcs- </li> </ol>	1	

Category	Ste	eps	Comment			
Verify username and	Navigate to Manage Configurations - >		Configure Injector Service WSDL URL (Applicable to SOAP implementation of injector service)			
password	1.	Choose rib- rob_ws_security_u ser_name_alias from dron down	Current injector Service URL       http://examplehost.com:25704/nb-injector-services-web/resourcesi/njector/inject         Update connection details for SOAP Injector service         New Injector Service Host*       examplehost.com         New Injector Service Port*       25704         New Security Policy Name*       policyC >> Help			
	2.	Set username and password to be empty.	Configure Security details (Applicable for both SOAP/ReST implementation of Injector Service) Update security details Secured User Name* Secured User Name* Secured User Password* Save Cance			
Verify ClientID and Secret Navigate to Manag Configurations - > Injector Service Choose rib- rob_oauth2_applic _client_user-name from drop down an verify details		vigate to Manage nfigurations - > ector Service oose rib- _oauth2_application ient_user-name-alias m drop down and ify details	Image: Configure Injector Service         Adapter Selection           System Options         Injector Service           Adapter Selection         Configure Injector Service (Adapter Selection)           Configure Injector Service WSDL URL (Applicable to SOAP Implementation of Injector service)         Current Injector Service USDL URL (Applicable to SOAP implementation of Injector service)			
	1.	Verify a valid Client ID in username is set.	Update connection details for SOAP Injector service       New Injector Service Host*     examplehost.com       New Injector Service Port*     25704       New Security Policy Name*     policyC ×			
	2.	Verify a valid Client Secret in password is set.	Configure Security details (Applicable for both SOAP/ReST implementation of Injector Service)  Update security details Secured User Allas  New Cancer  Secured User Passwort*  Secured User Cancer  Secured User Cancer			
Ping test	Navigate to Manage Configurations -> RIB Service Monitor					
	1.	Click on ping	And the additional and the additional a Additional additional additionadditional additionadditional additional additional additionad			
	2.	It should return success	na menanat unitara Rua tan dan Abda			

#### Table 6-1 (Cont.) Integrating with ROB


Category	Steps	Comment
Verify	How to verify if the ROB	1. Get the ROB Client ID and secret.
credentials	credentials are correct.	<ol> <li>Execute the following curl commands for grant_type cli- ent_credentials:</li> </ol>
		ClientId=RGBU_RICS_STG83_APPID ClientSecret=776381f5-88f5-4995-aa57-ecc7b7a1a8d7 IDCSUrl=https:// idcs-24e4baae56764e91be371e6a2060d66e.identity.c9dev 2.oc9qadev.com AccessToken=\$(curl -i -X POST \ user \$ClientId:\$ClientSecret \ -H "Content-Type: application/x-www-form- urlencoded;charset=UTF-8" \ \$IDCSUrl/oauth2/v1/token \ -d "grant_type=client_credentials&scope=urn:opc:idm:_m yscopes_"   grep -o -P '(?<=access_token":").*(? =","token_type)') echo \$AccessToken ribExtServiceUrl=https://omni.retail.us- phoenix-1.ocs.oc-test.com/rgbu-omni-rgbu-stg83-obcs/ rib-injector-services-web/orcos/resources/injector/ ping curl -ivkLnoproxy '*' -H "Authorization: Bearer \$AccessToken" -H "Content-Type: application/ xml" -X GET \$ribExtServiceUrl if you get a 200 response, then the configuration is correct if you get 401 unauthorized, then Client ID and secret are incorrect

 Table 6-1 (Cont.) Integrating with ROB



# 7 Integration with Fusion Middleware

RIB is certified on the Oracle Fusion Middleware Application Server. All RIB publishers, subscribers, and TAFRs are Java EE standard components (EJBs and MDBs) that are deployed and managed by the WebLogic Application Server in managed instances. This means that the RIB can be deployed into an existing Fusion Middleware architecture without any changes.

All RIB message payloads are fully standard compliant XSD based. All of the XML payloads are namespace aware and follow the general standards as well as the conventions that make them compatible with other Oracle Fusion products such ESB and BPEL. The payload schema definitions (XSDs) are packaged with each release along with sample messages.

The recommended approach for integration between the RIB and Oracle Fusion Middleware products is at the JMS topic level. Any standards compliant tool or product that can interface to the JMS and subscribe and publish messages can be integrated with the RIB.

There are some key functional requirements that an integrating application must follow. It must have the ability to do the following:

- Connect to a standard JMS and publish to a topic.
- Create a durable subscriber to a RIB JMS topic
- Set user-defined message properties.
- Encode and decode RIB payloads embedded within the RIB message envelope.

### General RIB to Fusion Middleware Architecture



The Oracle Fusion Middleware products, such as ESB and BPEL, use a common standard JMS Adapter. This adapter can be used to connect to the RIB certified JMS Provider and topics.

The JMS topics that the RIB creates for publication and subscription are detailed in the *Oracle Retail Integration Bus Integration Guide*, along with all of the message payloads for each message family.

The RIB html encodes each message payload and inserts it into the RIB messages envelope. Each message has a JMS user-defined property called threadValue that is required to be set on all in-bound messages. In a multi-channel message flow, the subscriber will need to set the message selector to an appropriate threadValue to maintain message publication sequencing.

The xml schema definitions for the payloads and the RIB Messages envelopes are packaged and shipped with the RIB.

The RIB JMS topic names and message flows between the RIB adapters for each of the Oracle Retail applications are defined in the rib-integration-flows.xml file. This file is the single source of truth that the RIB release uses at configuration and run-time. It is required to be accessible within each RIB deployment: http://<server>:<port>/rib-func-artifact/rib-integration-flows.xml. During installation and configuration, this file is deployed as a part of the functional artifact war file.



# Integration with External Applications

RIBforExt is the Oracle Retail Enterprise Integration component designed to address the connectivity requirements for 3rd Party integrations in a hybrid cloud topology where the RIB is deployed in the Retail Integration Cloud Services.

In a hybrid cloud scenario customers no longer have access to RIB's JMS server and cannot directly publish and subscribe to messages on the JMS topics. The RIB-EXT app is designed to fill that gap, it provides Web Service based APIs to publish to and subscribe from the RIB's JMS from third party systems.

RIBforEXT has all of the RIB flows available for the deployment time configuration based on the customer use cases.

RIB-EXT out-of-the-box provides the complete set of publishers, subscribers and retry adapters needed for the external application to integrate with Oracle Retail applications using RIB infrastructure.

The selective list of publisher and subscriber adapters needed by each specific external application is defined by the customer's implementation team.



## Implementing RIB-EXT

RIB-EXT is an Oracle Retail Integration Application that provides necessary communication channel for external applications to publish and consume message from RIB's JMS on cloud and premise.



#### Note:

BasicAuth will no longer be supported starting from the RICS v24 release. RICS will enforce OAuth2 as the required authentication mechanism using the IDCS authenticator. Oauth2 is being enforced for authorization for ongoing security reasons and to ensure the customer stays within their OCI IAM limits. Customer/SI partner are advised to prepare for this change and implement OAuth2 for both inbound and outbound calls via RIB-EXT.

RICS is also enforcing environment specific Oauth scope for authorization of inbound web service calls (RIB-EXT). The scope pattern that is used in the RICS IDCS app creation template is rgbu:rics:RICS-<ENVIRONMENT> where ENVIRONMENT is the environment type (STG, PRD, UAT, DEV1, DEV2, and so on). For details Refer Section: Create OAuth2 Client Application in IDCS.

### How to Send/Receive Messages to/from the RIB System

For third-party integration, RIB-EXT provides ReST API's for external applications to send and receive data from the RIB system. The following sections cover the implementation details.

### External Application as a Publisher (rest-app) using OAuth2

The end point of publishing service follows below pattern:

Resource	HTTP Method	Endpoint
Ping	GET	GET https:// <external_lb_url>/<rics-sub- namespace&gt;/rib-ext-services- web/resources/publisher/ping</rics-sub- </external_lb_url>
Publish	POST	https:// <external_lb_url>/ <rics-sub-namespace>/rib-ext- services-web/resources/ publisher/publish</rics-sub-namespace></external_lb_url>

#### Table 8-1 Publishing Service Pattern

- RIB-EXT publishing service REST endpoints are protected using OAuth2 tokenbased authentication meaning end points are accessible by sending along an access token.
- Scope will be used for authorization of REST services. Scope for RICS is in the following format- rgbu:rics:RICS-<Environemnt Type><Environment Index> (that is, rgbu:rics:RICS-DEV1).
- Client Credentials grant type is supported.

For getting access to RICS publishing service you need to create a client app in IDCS. IDCS app generates an access token that will be used for making publishing service calls. Follow steps for creating the client app in IDCS.



# Create OAuth2 Client Application in IDCS

Use Retail Home for creating the client app in IDCS. Once app is created you will get client id and client secret both of them necessary to get access token. Follow the instructions below for generating the access token and making service call using OAuth2 token.

1. Login into retail home as retail home administrator.

	<b>ORACLE</b> CIO rgbudevsandbox	ud
	Oracle Cloud Account Sign In	
User Name		
RETAIL_HOME_A	DMIN_USER	
Password		
•••••		
	Sign In	
	Need help signing in? Click he	ere
		Cookie Preferences

2. In retail home screen click on Settings menu icon on the left and then click on **Application Administration**.

=	O Retail	
88	Settings	
☆	Q. Bearch for a setting	e Admin User RETAIL_HOME_ADMIN
ţ	User Interface >	
Ô	Dashboards and Reports	Merchandising Metrics
000	Application Administration >	
	Resource Bundles >	
	Manage Notifications	
	Integration Status	
	Application Properties	
	Import/Export	
٢		

3. On the Application Administra-tion menu click on Application Navigator Setup. Notice all the hosted applications are listed here with their application and platform service url.

#### Settings -> Application Administration->Application Navigator Settings

	avigator Lir	nks for Retail Hor	ne					
				(2 mm)				
ľ	Features	Application Name	Color Set	Application Code	Ossistic Roles Import	Platform Service	Seeded	Roles
	88 0 ☆ ●	A Assortment Planning	Chestrut	APOFSLCS			Yes	
	語口☆●	Retail Demand Foreca	Chestnut	RDFCS			Yes	
	88 0 ☆ ⊕	名 Oracle Retail Home	Chestnut	RH	https://home.retail.us-phoenix-l	https://home.retail.us-phoenix-1.	Yes	
	88 0 ☆ ⊕	名 Store Inventory Opera	Chestrut	SIDCS	https://rex.retail.us-phoenix-loc	https://rex.retail.us-phoenix-1.oc	No	
	出心なる	유 Retail Merchandising	Chestnut	Rms	https://rex.retail.us-phoenix-l.oc	https://rex.retail.us-phoenix-l.oc	No	
	器 🗘 🎄	只 Rics	Mango	Plics	https://rex.retail.us-phoenix-1.oc	https://rex.retall.us-phoenix-1.oc	No	
	88 0 ☆ ⊕	A Customer Engagemen	Cyan	ORCE	https://res.retail.us-phoenis-1.oc	https://rex.retail.us-phoenix-1.oc	Ves	
	開日会会	8. Customer Engagemen	Jungle	ORCE	https://vex.retail.us-phoenix-1.oc	https://www.netall.us-phoenix-1.pc	No	

- 4. Look for application with name RICS. If you are not seeing RICS application try refreshing seed. Steps
  - a. Select the row with the application code as Rms.
  - b. Click the **Refresh Seed Data** button on top right corner of the menu.
  - c. Wait for some time and refresh the screen.
  - d. RICS should reflect now.



<ul> <li>Real Home Application</li> </ul>				χ		
New Ballout 1 Application Recognition	ing a					
Nanigatar Links for Rotal Hom	-					
+b/	× 8 8		Contractions inquiri fait-integring Select Sections			
Pasture Application Name	Calor Ref. A	Sphate Lab. 1	upfluation Units	Return Lexis	Instal	Rates
III G to # A Control Transmer	Creatinal 0	982 14	ten, trea estatue phonen han an instrum captures gins right country	Next the rest of series for an instrum, give an give right and findings for the rise	14	
図 G ☆ & A Machine	Castral B	trus ta	ten/instation phones lances tencom/phone-glo-appliteda/fen/heas/feafiline	https://www.iwadawadawadawa.com/agtureuragturegtir.nku/htta/hetarcher.com/	10.	
第 ○ 会 ※ 人 set5	Cantal 9	2013. ht	tys//iss.stalius.phoetic lacsar/tetricer//ghi-res-ghi-righl-isse/item-sell-thet/	https://www.iwadwa.elsenie/issues/issues/issues/issues/insues/insues/insues/insues/issue	*	
図 Q 会 参 人 Proce Ground and	Chestrad 0	Dam M	ips//herearisitary/serie/lasses initian/gits corresponded sight pary/MARIE	Next (Next Held or phonic has a fet conjugation or parent) sight per, Realized Peter	The local division of	
III 0 0 0 A Assertment and Spec	Contral 18	60			10	
III 0 0 0 A On-Second	Contrat 0	90			764	
III C C D A Parata and Anti-	Castral II	840			760	
98 C C C R R Homeson	Channel 1		ten jigtu pis hait Milaa anaksan jarahtin		10	

5. If RICS application is not reflecting even after following step 4. Select the row with the application code as Rms and click on the Actions menu on top left. Select Create IDCS OAuth 2.0 Client. A dialog will open for entering oauth2 client details.

#### Note:

Create IDCS OAuth 2.0 Client option is available only for applications those have platform service URL mentioned. RICS is making use of merch platform service as both the apps are sharing same IDCS tenancy.

=	C Retail				A shivekumer 85	TAIL,HOME,ADMIN,USER 👻 ()	D -
C Application Administra	Home Dashboard × Welcome	× Ар	plication Navigato	r Setup X			
Q. Search for a setting	Navigator Links for Ret	ail Hon	ne				
Customer Modules Man	Actions • View • +	D: /	× a	🖸 Detach 🛔 🛉	Disable Roles Import P	tole Mapping Refresh Seed Data	
Application Dashboard	Add	ю	Color Set	Application Code	Application Link	Platform Service	s
Application Navigator Se	Duolicate		Chestnut	ALC			٧
	Coprome	ħ	Chestnut	Reim			Y
	Edit		Chestnut	Resa			¥
	Delete	ng	Chestnut	Rms	https://rex.retail.us-phoenix-1.oc	https://rex.retail.us-phoenix-Loc	e Y
			Chestnut	Rpm			γ
	Refresh	Cloud Ser	Chestnut	MEPRCS			¥
	Manua	loud Serv	Chestnut	MFPCCS			٧
	Move up	Planning	Chestnut	APOFSLCS			γ
	Move down	_					
	Refresh Seed Data						
	Create IDCS OAuth 2.0 Client					Pub	lish

6. Skip this step if RICS application is not showing up. One of either Step 5 or Step 6 needs to be followed.

Select a row with application code as RICS. Click on the Actions menu on top left and select Create IDCS OAuth 2.0 Client. A dialog will open for entering oauth2 client details.



	igator Links for Ret	ail Hom	e				
Actio	na 🔻 View 💌 🕂	D: /	× 2 B	Detech 🛉 🖶 Dtaebb	e Roles Import Role Mapping	Refresh Seed Data	
	Add	et	Application Code	Application Link	Platform Service	Seeded	Roles
	Dualizata	estrut	APOFSLCS			Yes	
	Dopicale	estnut	RDFCS			Yes	
	Edit	estnut	RH	https://home.retail.us-phoenix-1	https://home.retail.us-phoenix-1	Yes	
	Delete	estrut	SIDCS	https://rex.retail.us-phoenix-l.oc	https://rex.retail.us-phoenix-1.oc	No	
		estrut	Rms	https://rex.retail.us-phoenix-Loc	https://rex.retail.us-phoenix-1.oc	No	
	Refresh	ngo	Rics	https://rex.retail.us-phoenix-l.oc	https://rex.retail.us-phoenix-Loc	No	
	Mousur	an	ORCE	https://rex.retail.us-phoenix-1.oc	https://rex.retail.us-phoenix-1.oc	Yes	
	move up	ugle	ORCE	https://rex.retail.us-phoenix-l.oc	https://rex.retail.us-phoenix-1.oc	No	

7. This dialog takes the following values:

App Name is 2-100 characters and will be used as the name in IDCS. Provide unique application name.

**Description** is a detailed description of the application.

Scope: <Custom environment-specific scope>

The scope pattern that is used in the RICS IDCS app creation template is rgbu:rics:<SERVICETYPE>-<ENVIRONMENT> where SERVICETYPE is RICS and ENVIRONMENT is the environment type (STG, PRD, UAT, DEV1, DEV2, and so on).

For example:

"scope": "rgbu:rics:RICS-PRD""scope": "rgbu:rics:RICS-STG"

Submitting this form will pro	vide you with a Client ID and Client Secret which will be displayed one time or
App Name	RICS_TEST
Description	Testing oauth client app
Scope 1	rgbu:rics:RICS-STG99
	+ Add Scope



8. When the application is created, another dialog will open to show the client ID and client secret of the new application. These values should be copied down to a safe location, as they will only be shown once. Retail Home cannot retrieve the credentials again after the dialog is closed.

New IDCS OAuth 2.0	Client
Display Name	RICS_TEST
Client ID	RICS_TEST_APPID
Client Secret	998e1e1d-f146-45a5-a9a1-99785e3ebf43
	Done

9. Client ID and Client Secret from previous step will be used for generating access token.

Sample code for generating Access Token:

```
clientId=RICS_TEST_APPID
clientSecret=998ele1d=f146-45a5-a9a1-99785e3ebf43
idcsUrl=https://idcs-234e8f7334564936aa0ed93f2c39e9ca.identity.pint.oc9qadev.com
scope=rgbu:rics:RICS-STG99
ec=$(echo -n "$clientId:$clientSecret" | base64 -w 0)
AccessToken=$(curl -iv \
-H "Authorization: Basic $ec" \
-H "Authorization: Basic $ec" \
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
--request POST $idcsUrl/oauth2/v1/token \
-d "grant_type=client_credentials&scope=$scope" | grep -o -P '(?
<=access_token":").*(?=","token_type)')</pre>
```

echo \$AccessToken

**10.** Now service call can be made by passing along the access token generated in previous step.

Here is sample curl command with Bearer token and rib-ext publisher ping

```
ribExtServiceUrl=https://rex.retail.us-phoenix-1.ocs.oc-test.com:443/rgbu-rex-eit-
stg99-rics/rib-ext-services-web/resources/publisher/ping
curl -ivkL --noproxy '*' -H "Authorization: Bearer $AccessToken" -H "Content-
Type: application/xml" -X GET $ri-bExtServiceUrl
```

#### Sample response

{"message": "ping() was called with input String of: hello"}

**11.** Publishing a message using access token.

Here is sample curl for publishing a message



```
ribExtServiceUrl=https://rex.retail.us-phoenix-1.ocs.oc-test.com:443/rgbu-
rex-eit-stg99-rics/rib-ext-services-web/resources/publisher/publish
curl -ivkL --noproxy '*' -H "Authorization: Bearer $AccessToken"
"Content-Type: application/xml" -X POST $ribExtServiceUrl --data
'<v1:ApplicationMessages xmlns:v1="http://www.oracle.com/retail/</pre>
integration/rib/ApplicationMessages/v1">
<v1:ApplicationMessage>
<v1:family>InvAdjust</v1:family>
<v1:type>InvAdjustCre</v1:type>
<v1:payloadXml>&lt;InvAdjustDesc xmlns=&quot;http://www.oracle.com/retail/
integration/base/bo/InvAdjustDesc/v1" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/retail/integration/base/bo/
InvAdjustDesc/v1
http://www.oracle.com/retail/integration/base/bo/InvAdjustDesc/v1/
InvAdjustDesc.xsd&quot
;><dc dest id&gt;DC ES&lt;/
dc dest id&qt;<InvAdjustDtl&qt;&lt;item id&qt;Aline&lt;/
item id><adjustment reason code&gt;stri&lt;/
adjustment reason code><unit qty&gt;22.4&lt;/unit qty&gt
;<transshipment nbr&gt;ss&lt;/
transshipment_nbr><from_disposition&gt;ss&lt;/
from disposition><to disposition&gt;sss&lt;/
to disposition><from trouble code&gt;sss&lt;/from trouble code&gt;
<to trouble code&gt;ss&lt;/
to trouble code><from wip code&gt;aaa&lt;/
from wip code><to wip code&gt;sss&lt;/
to wip code><transaction code&gt;4&lt;/
transaction code><user id&gt;TestUser&lt;/user id&gt;
<create date&gt;1999-10-23T20:27:56.32&lt;/
create date><po nbr&gt;PratapOrd96&lt;/po nbr&gt;&lt;doc type&gt;P&lt;/
doc type><aux reason code&gt;string&lt;/aux reason code&gt;
<weight&gt;12.4&lt;/weight&gt;&lt;weight uom&gt;smn;&lt;/
weight uom><unit cost&gt;20.4&lt;/
unit cost><InvAdjustUin&gt;&lt;uin&gt123&lt;/uin&gt;
<status&gt;4&lt;/status&gt;&lt;/InvAdjustUin&gt;&lt;/InvAdjustDtl&gt;&lt;/
InvAdjustDesc></v1:payloadXml>
</vl:ApplicationMessage>
</v1:ApplicationMessages>'
```

#### Sample response

```
{"message": "Publish done"}
```

### External Application as a Subscriber (rest-app)

For an external application to consume the message from the RIB's JMS on cloud, it has to host the Injector Service. Injector Service is a ReST webservice that is made available as a pluggable jar.

A pluggable jar is provided which contains all the wrapper classes to help in implementing injector service. rib-injector-services-web-<version>.war is the pluggable jar which can be included into the external application deployable file for example, ext-app.ear/lib. Once pluggable jar is added, endpoint for injector service will be exposed as follows:

https://<external-app-host>:<port>/ rib-injector-services-web/resources/injector/ inject

Pluggable jar is provided for reference however customer can choose to write



their own injector service by adhering to REST service contract detailed in next section.



### How to implement Injector Service (Service Contract) using ReST

Here is the Rest service contract detail:

1. Keep the path as Injector/inject.

```
@Path("/injector")
```

 Use POST for this service. As the input message object itself has identifier (message type- CRE/MOD) they don't need to use the PUT/PATCH. they can use message type to build the implementation logic.

```
@POST
@Path("/inject")
@Consumes({MediaType.APPLICATION_XML})
```

3. The input would be MediaType.APPLICATION\_XML and the structure would be 'ApplicationMessage' object. (file attached for reference).

- 4. Customer can utilize the payload.properties file for validation of message family and type.
- 5. Return type should be JSON, see below example:

String message = "{\"message\": \"Inject successful.\"}";
return Response.ok(message, MediaType.APPLICATION\_JSON).build();

6. For exception response customer needs to follow the structure of exceptionVO.

### How to Secure Injector Service with Oauth2

Injector service exposed by external service should be secured with OAuth2. This chapters covers the key points that should be taken into consideration while protecting the resources exposed by external application.

#### Prerequisites

IDCS should be same as RICS.



- Use Client Credentials grant type with scope to provide access to resource.
- Following is the screen shot of a sample IDCS app with scope added

	Gapit resurses services disconstration and materialises				
and a local state of the local state	Carlo Card Bernari - 1720 1973 1997				
	RGBU_RICS_DEV23				
ACTVB	Deweigten, Owei, Real Importer Goot Barrie Codene span-1000 - Codene smar IBL - Danlys of My Apen 10 Une col registed service: 50 Bitterie genet as authorizative: Datated		8 8 0 0	pleature izen 🍙 pleature 2003 - alem viges an Ulis - alem model letting califacti 2014 -	
Resources	OAuth configuration				
Application roles	Resource server configuration				
Uses	Resurce sever torigonition for the application is enabled.				
Grape	Configure application APIs that need to be OA	Auth protected			
	Annea tokar angintism (seconda) undefinet Annea tokan raftedit. Districted Primary automate rytoritor				
	Secondary sufferce				
	learning automs				Protocial
			No secondary sur	iena attiet	
	Bourn				
	despa	Protected	Distanta name	Description	Re: 0
	Resource	16			~



### **RIB-EXT Side of Configuration to Point to External Application**

Below are the steps to point rib-ext to the correct injector service.

Table 8-2

Category	Step	Comment
Access RIB Admin GUI	Access the rib admin GUI at https:// <external-load- balancer&gt;/rib-ext-admin-gui Log in with the admin user.</external-load- 	Image: State of the second



Table 8-2 (	Cont.)
-------------	--------

Category	Step	Comment
Verify Configuratio n and update	Navigate to Manage Configurations -> System options Search for and verify the following:	Non-         N           All Market         All
	1. destination.retail.appT pe: rest-app	nonen Brannen Brannen Brannen
	2. Update the value for InjectorService URL (injector.service.endpor int.url). URL should point to inject service provided by external application. (e.g https:// <host:port>/rib- injector-services-web/ resources/injector/ inject</host:port>	
	3. Update the value for Ping Service URL (injector.service.endpotint.ping.url). This URL should point to a ping service provided by an external application. (for example - https:// <host:port>/rib- injector- services-web/ resources/ injector/ping).</host:port>	
	Note: This feature allows users to provide their ping URL, as it can be a freeform URL. Previously, it was assumed ping used host:port/ injector/ping, causing ping to fail when the systems used its own ping URL. Now, the ping feature in rib-ext relies on the ping implemented on the system. Ping is typically used to test the first-time handshake between the service client and	



Table 8-2 (Cont.)

Category	Step	Comment
	the service provider before sending the actual data over to OIC. The fact that data is moving to OIC tells us that the integration is working fine.	a
	<ol> <li>For 3<sup>rd</sup> party integration where the injector service is hosted on OIC/on- prem, the below system property in JAVA_OPTIONS needs to be added oauth2.url.path.wo.vrc =<context of<br="" root="">injector service&gt;</context></li> </ol>	
	Important: context root of injector service is any word in injector service url which can identify service uniquely.	
	Eg- For the following injector service url https:// <external-lb>/ external-injector- services/external/ ribinjector/inject.</external-lb>	
	Java_option would be oauth2.url.path.wo.vrc =ribinjector	
	<ol> <li>Security Policy (injector.service.secur y.policyname) : policy/</li> </ol>	it A
	<ol> <li>IDCS OAuth Server URL (oauth2.default.author zationServerUrl): https://<idcs-tenant>/ oauth2/v1/token</idcs-tenant></li> </ol>	i
	<ol> <li>OAuth2 Token Scope: Update with external application provided scope</li> </ol>	

Category	Step	Comment           ate to Manage         Configure Injector Service WSDL URL (Applicable to SOAP implementation of injector service)           gurations - > Injector         Current Injector Service URL http://examplehost.com/25704/rb-injector-services-web/resources/injector/injector		
Update username and	Navigate to Manage Configurations - > Injector Service			
empty	Update details.	Update connection details for SOAP Injector service New Injector Service Host* examplehost.com New Life Order Dett		
	<ol> <li>Choose "rib- (app)_ws_security_us er-name-alias" as Secured User Alias.</li> <li>Usedate the Oceaned</li> </ol>	New Injector Service Fort 25764 New Security Policy Name* policyC <u>Hale</u> Configure Security details (Applicable for both SOAP/ReST implementation of Injector Service) Update security details Secured User Alia The ext. ws_security_user-name-alias		
	2. Update the Secured User Name with a blank userName.	Secured User Name* Secured User Password* Save Cancel		
	<ol> <li>Update the Secured User Password with a blank password.</li> </ol>			
	4. Click on Save.			
Update ClientID/ Secret	Navigate to Manage Configurations - > Injector Service	Configure Injector Service WSDL URL (Applicable to SOAP implementation of injector service) Current Injector Service URL http://examplehost.com/25704/rib-injector-services-web/resources/injector/inject		
	Update details	Update connection details for SOAP Injector service New Injector Service Host* examplehost.com		
	<ol> <li>Choose "rib- (app)_oauth2_applicati on_client_user-name- alias" as Secured User Alias.</li> </ol>	New Injector Service Port*     25704       New Security Policy Name*     policyC ···· Hale   Configure Security details (Applicable for both SOAP/ReST implementation of Injector Service)       Update security details       Secured User Alias     Int-ext_vs_security_user-name-alias		
	2. Update the Secured User Name with clientID.	Secured User Name* Secured User Password* Save Cancel		
	3. Update the Secured User Password with clientSecret.			
Ping Test	Navigate to Manage Configurations -> RIB Service Monitor	Image: Instantiant Control (Instantiant)         Image: Instantiant)         Image: Instantiant)           Bandwidt Neumann (Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)           Bandwidt Neumann (Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)           Bandwidt Neumann (Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)           Bandwidt Neumann (Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)         Image: Instantiant)		
	1. Click on ping	Na Bak-Salat Lahasi Bi-as Baki-Salata Bi-as Salatakan		
	2. It should return success			

Table 8-2 (Cont.)

Category	Step	Comment
How to verify	Verify if the provided	Execute the following curl commands
injector service details are correct	credentials are correct.	ClientId=56c7eb72f11b43bb98bf2570fa2353eb ClientSecret=bb18aa22-4bb4-41d1-9ed4- fea276651e28 IDCSUrl=https:// idcs-24e4baae56764e91be371e6a2060d66e.ident ity.c9dev2.oc9qadev.com AccessToken=\$(curl -i -X POST \ user \$ClientId:\$ClientSecret \ -H "Content- Type: applica-tion/x-www-form- urlencoded;charset=UTF-8" \ \$IDCSUrl/oauth2/v1/token \ -d "grant_type=client_credentials&scope=urn:op c:idm:_myscopes_"   grep -o -P '(? <=access_token":").*(?=","token_type)') ribExtServiceUrl=https://rgbu-phx- lbext-351.us.oracle.com/rib-injector- services-web/resources/injector/ping curl -ivkLnoproxy '*' -H "Authorization: Bearer \$AccessToken" -H "Content-Type: applica-tion/xml" -X GET \$ribExtServiceUrl

Table 8-2 (Cont.)

### How to switch Injector Service app Type at Runtime

RIB-EXT is a rest-app by default for CFS and expects injector service also to be of ResT type. ONLY for egress/migration customers who already have injector service SOAP implementation in GBUCS they should follow these steps to switch from rest to soap based injector calls and vice-versa.

### How to Change rib-ext injector-service-app-type from REST to SOAP

 Open rib-ext admin gui. Go to Manage Configurations > System Options, observe new prop-erty i.e. injector-service-appType added to allow switching injector service app-type at runtime.

By default rib-ext is deployed as rest-app so injector-service-appType is defaulted to.



		Welcome, rics.admin
st:System Options	Page Robushed Tao Jan N	5 2822 10:53:42 GMT+0538 (India Standard 1
tome Adapter Manager Log Manager HB Logo Manage Configurations RIB ServiceManitor		
System Options Injector Service Adapter Selection		
Properties		
In Filo ApplC nubled	(bran	@   #
alett <sup>e</sup> ublisher Impi	com.netek.nb.ailert.impl.	(8   ¢
destination.retail.appType	nest-app	(g   \$
disabled.ogt.evell.lpdates?coll.depter	Customer_pub,Custome	(2   ¢
enable global email alert	false	(2   ¢
enableOynamicAdapterInstance/Belection	true .	(2   2
for.Alloc_sub.drop-messages-ol-types	ALLOCPULRUP	(g   #
fer:ISORTVReq_sub.dop-messages-ellypes	RTVREQRUIREP	(2   ¢
for EIOStockDrder_sub-drop-messages-of-types	SOPULIEF	(2   ¢
for Remo_sub drop-messages-of-types	TTEMFULREP	(2   ¢
for Onlin_sub.dop-messages-eHypes	POPULIEP	(g   #
for RTVHeq_sab.dop-messages of types	RTVRDQFULRIP	(8   ¢
for Transfern_sub-drop-messages-of-types	TRANSPERFUSION	@   #
for Verstor_sub.chop-messages-of-types	VENDORFLUREP	(2   ¢
for IIHStockOvder_w.b.drop-messages-of-types	SOPULAR	(2   ¢
hospital attempt delay	1.0	(2   ¢
heapital attempt delayincrement	1.0	@   #
heropital attempt max	5	(8   ¢
injector.service.appType	1012-app	@   #
Injector.service.endpoint.er/	http://exampleheait.com	(2   ¢
injector.service.security.policyneme	policy;	(2   ¢
log default file_path	Juli L/wade/user_projec	8 2
mail.amip.from	admin@oracle.com	Q   2
mail settp: host	mail-router-avc.rgba-re-	(2   ¢
mail antip port	2941	Q   #
read overp to list	rtg_dev_as_prpdboracle	814
ner#OfRecards/IoReity	20	Q   2
repare notification lead time	30	212
request domain	neuretailus-phaesio-1.c	014
request referer	resumital us phoenix Lu	214
rib adapter shuldown repave lead time	5	814
rib jans jims 5 hostname	prothest example.com	2 2
rbo jena ji ang ji ang din	1521	014
rikPublisheringi	com.rebek.rdb.d2ee.12ee.	010
war http.port	8050	010
war http://www.http.protecol	Mito	21.4
who walled file locations	Julit/wade/user protect	010
vis valiet map name	all and all all all all all all all all all al	0.1.4
who well at your allow	situat ult una conta-	010
	and the local state of the state	08 I #

- Copyright © 2021, Oracle and/or its alliates. All rights reserved.
- 2. Edit injector-service-appType and update this to soap-app. Save the changes.

fur.WHStockOrder_sub.drop messages of types		SOPULREP	212
hospital attempt delay		10	(2   ¢
hospital attempt delayincrement		10	815
hospital attornpt max		5	812
injector service appType	injector.aervice.appType	(emp-mpl	· · · ·

3. Navigate to Manage Configurations > Injector Service tab. Check for the correctness of injector service URL, ensure it points to correct ext-app injector service.

Update rib-ext\_ws\_security\_user-name-alias with correct username/password needed to make inject call.

ORACLE		
RETAIL Retail Integration Bus Manager		
		Welcome, ricsofsadmin Logou
rib-ext:Injector Service		Pana Refeatured Thu Oct 26 2023 10:25-11 OMT+0550 (India Standard Tima)
Home Adapter Manager Log Manager	er RIB Logs Manage Configurations RIB ServiceMonitor	
System Options Injector Service A	dapter Selection	
Configure Injector Service WSDL URL	(Applicable to SOAP implementation of injector service)	
Current Injector Service URL http://example	ehost.com:25704/rib-injector-services-web/resources/injector/inject	
Update connection details for SOAP Inje	actor service	
New Injector Service Host*	examplehost.com	
New Injector Service Port*	25704	
New Security Policy Name*	policyC ~ Help	
Configure Security details (Applicable	for both SOAP/ReST implementation of Injector Service)	
somgare security astano (rippicano		
Update security details		
Secured User Alias rib-ext_ws_sec	urity_user-name-alias v Help	
Secured Liser Name*	Charlesson	
Secured User Password*	Show Germanie	
(Save) Can	cel	
System Options Injector Service /	Adapter Selection	
Home Adapter Manager Log Manager	er RIB Logs Manage Configurations RIB ServiceMonitor	
Constitute 0 2021 Oracle and/or its affiliates All rivits reserve	4	

#### 4. Update the value for the Ping Service URL

(injector.service.endpoint.ping.url). This URL should point to a ping service WSDL provided by an external application.

#### Note:

This feature allows users to provide their ping URL. The ping feature in rib-ext relies on the ping implemented on the system. Ping is typically used to test the first-time handshake between the service client and the service provider before sending the actual data to OIC. The fact that data is moving to OIC tells us that the integration is working fine.

5. Setup is ready now. Do a ping test from RIB ServiceMonitor tab.

#### How to change rib-ext injector-service-app-type from SOAP to ReST

 Navigate to Manage Configurations > System Options from admin GUI. Look for injector-service-appType, update this property to switch from SOAP to ReST.Save the changes.

hospitol.attempt.delay/increment		10	Q   2
hespital altompt max		5	(2   ¢
injedar nervice app?ype	injector service appTspr	rest-app	- <del></del>
injector service endpoint ut		http://wamplehost.com	8 2

2. Navigate to Injector Service tab. Update host/port and security credentials (ribext\_ws\_security\_user-name-alias) if needed.



ORACLE.		
Retail Integration Bus Manager		
rib-ext:Injector Service		Welcome, ricscfsadmin Logour
Ind-ext.injector dervice		Page Refreshed Thu Oct 26 2023 10:25:11 GMT+0530 (India Standard Time).
Home Adapter Manager Log M	Inager RIB Logs Manage Configurations RIB ServiceMonitor	
System Options Injector Service	Adapter Selection	
Configure Injector Service WSDL U	RL (Applicable to SOAP implementation of injector service)	
Current Injector Service URL http://ex	amplehost.com:25704/rib-injector-services-web/resources/injector/inject	
Update connection details for SOA	P Injector service	
New Injector Service Host*	examplehost.com	
New Injector Service Port-	25704	
New Security Policy Name*	policyC ~ Hele	
Configure Security details (Applica	ble for both SOAP/ReST implementation of Injector Service)	
Update security details		
Secured User Alias rib-ext_wa	security_user-name-alias v Hdg	
Secured Liser Name*	21	
Secured User Password*	Snow Osername	
Save	Cantel	
System Options Injector Service	Adapter Selection	
Home Adapter Manager Log M	nager RIB Loge Manage Configurations RIB ServiceMonitor	
Copyright @ 2021, Oracle and/or its affiliates. All rights n	served.	

3. Setup is ready now. Do a ping test from RIB Service Monitor tab.

### Error Handling

The RIB infrastructure provides a mechanism called RIB error hospital to handle and manage the error messages. When the publishing or subscription of a message fails in the rib-ext for some reason, it lands in error hospital with a reason code. The retry adapters in the rib-ext application are responsible for retrying the messages in error hospital.

Oracle RIB Hospital Administration (RIHA) is a Weblogic application that allows the management of messages in error hospital. Some of the RIHA operations include:

- Viewing error messages
- Editing error messages
- Retrying error messages
- Stopping error messages

For more information, see the Oracle Retail Integration Bus Hospital Administration Guide.

### Monitoring Integration

To monitor live statistics of various components involved in RIB integration system like RIB adapter, error hospital, JMS server, RTG provides a live monitoring application called the Retail Integration Console (RIC).

The RIC is the user interface application designed to provide a unified view of the RTG integration products within the business context of the Oracle Retail applications. It provides near real time statistics regarding the message flows, JMS topics, historical trends of each message family, performance comparisons, and static information like application configuration.

For more information, see the Oracle Retail Integration Console User Guide.

# Sample Application.wadl File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ns0:application xmlns:ns0="http://wadl.dev.java.net/2009/02">
  <ns0:doc ns1:generatedBy="Jersey: 2.22.4 2016-11-30 13:33:53" xmlns:ns1="http://</pre>
jersey.java.net/"/>
  <ns0:doc ns2:hint="This is simplified WADL with user and core resources only. To get
full WADL with extended resources use the query parameter detail. Link: http://
abc.us.oracle.com:8003/rib-injector-services-web/resources/application.wadl?
detail=true" xmlns:ns2="http://jersey.java.net/"/>
  <ns0:grammars>
    <ns0:include href="application.wadl/xsd0.xsd">
      <ns0:doc title="Generated" xml:lang="en"/>
    </ns0:include>
  </ns0:grammars>
  <ns0:resources base="http://abc.us.oracle.com:8003/rib-injector-services-web/
resources/">
    <ns0:resource path="discover">
      <ns0:method id="discoverAllResources" name="GET">
        <ns0:response>
          <ns0:representation mediaType="application/json"/>
        </ns0:response>
      </ns0:method>
    </ns0:resource>
    <ns0:resource path="/injector">
      <ns0:resource path="/inject">
        <ns0:method id="injectMessage" name="POST">
          <ns0:request>
            <ns0:representation mediaType="application/xml"
element="ns3:ApplicationMessage" xmlns:ns3="http://www.oracle.com/retail/
integration/rib/ApplicationMessages/v1"/>
          </ns0:request>
          <ns0:response>
            <ns0:representation mediaType="*/*"/>
          </ns0:response>
        </ns0:method>
      </ns0:resource>
      <ns0:resource path="/ping">
        <ns0:method id="ping" name="GET">
          <ns0:request>
            <ns0:param name="pingMessage" default="hello" type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" style="query"/>
          </ns0:request>
          <ns0:response>
            <ns0:representation mediaType="application/json"/>
          </ns0:response>
        </ns0:method>
      </ns0:resource>
    </ns0:resource>
  </ns0:resources>
</ns0:application>
```



### Sample Resource Class

```
package com.oracle.retail.rib.integration.services.applicationmessageinjector;
import javax.ejb.EJB;
import javax.ejb.Stateless;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import com.oracle.retail.integration.rib.applicationmessages.v1.*;
import com.retek.rib.binding.exception.InjectorException;
import com.retek.rib.binding.injector.Injector;
import com.retek.rib.binding.injector.InjectorFactory;
import com.retek.rib.domain.payload.PayloadFactory;
import javax.ws.rs.DefaultValue;
import javax.ws.rs.QueryParam;
import javax.ws.rs.core.Response;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import com.oracle.retail.integration.payload.Payload;
@Stateless
@Path("/injector")
public class ApplicationMessageInjectorResource {
    private static Log LOG =
            LogFactory.getLog(ApplicationMessageInjectorResource.class);
    @GET
    @Path("/ping")
    @Produces({MediaType.APPLICATION JSON})
    public Response ping(@DefaultValue("hello") @QueryParam("pingMessage")
String pingMessage) {
        String message = "{\"message\": \"Got " + pingMessage + " from
server.\"}";
        return Response.ok(message, MediaType.APPLICATION JSON).build();
    }
    @POST
    @Path("/inject")
    @Consumes({MediaType.APPLICATION XML})
    public Response injectMessage (ApplicationMessage applicationMessage) throws
InjectorException{
        verifyNotNull(applicationMessage, "applicationMessage");
        invokeInjectForMessageType(applicationMessage.getFamily(),
applicationMessage.getType(), applicationMessage.getBusinessObjectId(),
applicationMessage.getPayloadXml());
        String message = "{\"message\": \"Inject successful.\"}";
        return Response.ok(message, MediaType.APPLICATION JSON).build();
    }
```



```
private void invokeInjectForMessageType(String family, String messageType, String
businessObjectId, String retailPayload)throws InjectorException{
        try {
            verifyNotNull(family, "family");
            verifyNotNull(messageType, "messageType");
            verifyNotNull(retailPayload, "retailPayload");
            Payload payload = PayloadFactory.unmarshalPayload(family, messageType,
retailPayload);
            Injector injector = InjectorFactory.getInstance().getInjector(
      ??
                     family, messageType);
            if (injector == null) {
                final String eMsg = "Unknown message"
                    + " family/type: " + family + "/" + messageType;
                LOG.error(eMsq);
                throw new InjectorException(eMsg);
            if(LOG.isDebugEnabled()){
                LOG.debug("Received inject call for family("+family+")
type("+messageType+") businessObjectId("+businessObjectId+") with payload:\n" +
payload.toString());
            }
            injector.inject(messageType, businessObjectId, payload);
            LOG.debug("Inject call for family("+family+") type("+messageType+")
businessObjectId("+businessObjectId+") return.");
  ??
         } catch (InjectorException e) {
            final String eMsg = "Exception calling inject.";
            LOG.error(eMsg, e);
            throw e;
        }catch (Exception re) {
            final String eMsg = "Exception calling inject.";
            LOG.error(eMsg, re);
            throw new RuntimeException (eMsg, re);
        }
    }
    private void verifyNotNull(Object field, String fieldName) {
      if(field == null) {
        final String eMsg = fieldName + " cannot be null.";
        LOG.error(eMsg);
        throw new IllegalArgumentException(eMsg);
      }
    }
}
```

### ApplicationMessages.xsd



```
ApplicationMessages/v1"
           xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
           xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
           jaxb:extensionBindingPrefixes="xjc"
           jaxb:version="2.0"
           targetNamespace="http://www.oracle.com/retail/integration/rib/
ApplicationMessages/v1"
           elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:appinfo>
            <jaxb:globalBindings
                fixedAttributeAsConstantProperty="false"
                choiceContentProperty="true"
                enableFailFastCheck="true"
                generateIsSetMethod="true"
                enableValidation="true">
                <!--xjc:javaType name="java.util.Calendar"
                               xmlType="xs:dateTime"
adapter="com.oracle.retail.integration.rib.rib_integration_runtime_info.datatypea
dapter.CalendarAdapter"/ -->
                <jaxb:serializable uid="1"/>
            </jaxb:globalBindings>
            <!--jaxb:schemaBindings>
                <jaxb:package
name="com.oracle.retail.integration.rib.ribintegrationruntimeinfo" />
            </jaxb:schemaBindings-->
        </xs:appinfo>
    </xs:annotation>
    <xs:element name="ApplicationMessages">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ApplicationMessage" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ApplicationMessage">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="family" type="string25"/>
                <xs:element name="type" type="string30"/>
                <xs:element name="businessObjectId" type="string255"</pre>
minOccurs="0"/>
                <xs:element ref="ApplicationMessageRoutingInfo" minOccurs="0"</pre>
maxOccurs="unbounded"/>
                <xs:element name="payloadXml" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ApplicationMessageRoutingInfo">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" type="string25"/>
                <xs:element name="value" type="string25"/>
                <xs:element ref="ApplicationMessageRoutingInfoDetail"</pre>
```



```
minOccurs="0" maxOccurs="2"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ApplicationMessageRoutingInfoDetail">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" type="string25"/>
                <xs:element name="value" type="string300"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
  <xs:simpleType name="string255">
    <xs:restriction base="xs:string">
      <xs:maxLength value="255" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string25">
   <xs:restriction base="xs:string">
      <xs:maxLength value="25" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string30">
    <xs:restriction base="xs:string">
      <xs:maxLength value="30" />
    </xs:restriction>
  </xs:simpleType>
    <xs:simpleType name="string300">
    <xs:restriction base="xs:string">
      <xs:maxLength value="300" />
    </xs:restriction>
  </xs:simpleType>
```

</xs:schema>

payload.properties

# payload.properties

ASNIN.ASNINCRE=com.oracle.retail.integration.base.bo.asnindesc.v1.ASNInDesc ASNIN.ASNINDEL=com.oracle.retail.integration.base.bo.asninref.v1.ASNInRef ASNIN.ASNINMOD=com.oracle.retail.integration.base.bo.asnindesc.v1.ASNInDesc

```
WH.WHCRE=com.oracle.retail.integration.base.bo.whdesc.v1.WHDesc
WH.WHDEL=com.oracle.retail.integration.base.bo.whref.v1.WHRef
WH.WHMOD=com.oracle.retail.integration.base.bo.whdesc.v1.WHDesc
```

# Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
http:// localhost:700 1/rib-injector- services- web/ resources/ injector/inject	POST	application/x ml Request are xml only and response are json only.	A valid user that is part of IntegrationGr oup.	<pre><v1:applicationm essage xmlns:v1="http:/ /www.oracle.com/ retail/ integration/rib/ ApplicationMessa ges/v1"&gt; <v1:family>XOrde r</v1:family> <v1:type>XOrderC re</v1:type> <v1:businessobje ctId&gt;592824510<!--<br-->v1:businessObjec tId&gt; <v1:payloadxml>&lt; <xorderdesc xmlns="http :// www.oracle.com/ retail/ integration/ base/bo/ XOrderDesc/ v1" xmlns:ns0=" http:// www.oracle.com/ retail/integra- tion/base/bo/ CustFlexAttriVo/ v1"&gt; <order_no> <order_no> <currency_cod e&gt;USD<!--<br-->currency_code&gt; ; <terms>13&amp; lt;/terms&gt; <not_after_da te&gt;2022-02-0 9T00:00:00Z<!--<br-->not_before_date&amp; gt; <not_after_da te&gt;2022-02-19 T00:00:00Z&lt;/</not_after_da </not_after_da </terms></currency_cod </order_no></order_no></xorderdesc </v1:payloadxml></v1:businessobje </v1:applicationm </pre>	HTTP/1.1 200 OK Date: Thu, 10 May 2018 16:33:11 GMT Content-Length: 33 Content-Type: application/json X-ORACLE-DMS- ECID: 4a8e5d3f-1aae-43 d7-ba84- c6b9c60563c7-000 00039 X-ORACLE-DMS- RID: 0 Set-Cookie: JSES- SIONID=hsFK5jW4B 1QtipC9zhng or1WL7ywxCuxsJeV wdgPpnv6oNUnde! 233126712; path=/; HttpOnly {"message": "Inject successful."}	Success

#### Table A-1 Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
				not after date&g		
				t;		
				<otb date<="" eow="" td=""><td></td><td></td></otb>		
				>2022-02-19T0		
				0:00:00Z </td <td></td> <td></td>		
				otb_eow_date>		
				<status>A&amp;</status>		
				lt;/status>		
				<exchange_rat< td=""><td></td><td></td></exchange_rat<>		
				e>1 </td <td></td> <td></td>		
				exchange_rate>		
				;		
				<pre>&lt;include_on_o</pre>		
				rd_ind>Y </td <td></td> <td></td>		
				include_on_ord_i		
				nd>		
				<written_date< td=""><td></td><td></td></written_date<>		
				>2022-02-09T0		
				0:00:00Z </td <td></td> <td></td>		
				written_date>		
				<xorderdtl&gt< td=""><td></td><td></td></xorderdtl&gt<>		
				;		
				all; itemagt; 1/42		
				50093&10;/		
				flt.locationsat.		
				21£]+•/		
				locationsat:		
				<pre>ilt.unit cost&amp;at</pre>		
				:10%]t:/		
				unit cost&at		
				<origin count<="" td=""><td></td><td></td></origin>		
				ry id&qtUS </td <td></td> <td></td>		
				origin country i		
				d>		
				<supp pack="" si<="" td=""><td></td><td></td></supp>		
				ze>1 </td <td></td> <td></td>		
				supp pack size&g		
				t;		
				<qty_ordered&< td=""><td></td><td></td></qty_ordered&<>		
				gt;2 </td <td></td> <td></td>		
				qty_ordered>		
				<location_typ< td=""><td></td><td></td></location_typ<>		
				e>W </td <td></td> <td></td>		
				location_type>		
				;		
				<reinstate_in< td=""><td></td><td></td></reinstate_in<>		
				d>N </td <td></td> <td></td>		
				reinstate_ind>		
				;		
				<pre>&lt;delivery_dat</pre>		
				e>2022-02-09T		
				00:00:00Z </td <td></td> <td></td>		

#### Table A-1 (Cont.) Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
				<pre>delivery_date&gt; ; &lt;/ XOrderDtl&gt; &lt;orig_ind&gt; &lt;edi_po_ind&gt; &lt;edi_po_ind&gt; &lt;pre_mark_ind &gt;N&lt;/ pre_mark_ind&gt; &lt;/ XOrderDesc&gt;&gt;&lt; /v1:payloadXml&gt; <!-- v1:ApplicationMe ssage--></pre>		

 Table A-1 (Cont.) Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments								
End Point	Method	Media Type	User/ Password If user in not added in IntegrationGr oup	<pre>Request.xml</pre>	Response HTTP/1.1 403 Forbidden Date: Thu, 05 Aug 2021 10:25:26 GMT Content-Length: 1166 Content-Type: text/html; char- set=UTF-8 HTML<br PUBLIC "-//W3C//DTD HTML 4.0 Draft// EN"> <html> <head> <title>Error 403Forbidden<!--/<br-->TITLE&gt; </title></head> <body bgcol-<br="">or="white"&gt; <font FACE=Helvetica&gt;&lt; BR CLEAR=all&gt; <table bor-<br="">der=0 cellspac- ing=5&gt;<tr><td><b R CLEAR=all&gt; <font FACE="Helvetica" COL-OR="black" SIZE="3"&gt;<h2>Err or 403 Forbidden</h2> <!--<br-->TD&gt;</font </b </td></tr> </table> <table bor-<br="">der=0 width=100% cellpad- ding=10&gt;<tr><td VALIGN=top WIDTH=100% BGCOL- OR=white&gt;<font FACE="Courier New"&gt;<font< td=""><td>Comments</td></font<></font </td </tr><tr><td></td><td></td><td></td><td></td><td>integration/ base/bo/WHDesc/</td><td>SIZE="3"&gt;<h3>Fro</h3></td><td></td></tr></table></font </body></html>	<b R CLEAR=all&gt; <font FACE="Helvetica" COL-OR="black" SIZE="3"&gt;<h2>Err or 403 Forbidden</h2> <!--<br-->TD&gt;</font </b 	Comments					integration/ base/bo/WHDesc/	SIZE="3"> <h3>Fro</h3>	
<b R CLEAR=all&gt; <font FACE="Helvetica" COL-OR="black" SIZE="3"&gt;<h2>Err or 403 Forbidden</h2> <!--<br-->TD&gt;</font </b 														
Comments														
				integration/ base/bo/WHDesc/	SIZE="3"> <h3>Fro</h3>									

Table A-1	(Cont.) Sample Request/Respons	e for ReST Injector Service
Table A-1	(Cont.) Sample Requestirespons	Se IUI REST INJECTUI SELVICE

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
				<pre>vl"&gt;&lt; wh&gt;l0<!--<br-->wh&gt;e&gt;g<!--<br-->wHDesc&gt;<!--<br-->v1:payloadXml&gt; <!--<br-->v1:ApplicationMe ssage&gt;</pre>	m RFC 2068 <i>Hypertext Transfer Protocol HTTP/1.1<!--<br-->i&gt;: <font FACE="Helvetica" SIZE="3"&gt;<h4>10. 4.4 403 Forbidden</h4>  <p><font FACE="Courier New"&gt;The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated. If the request method was not HEAD and the server wishes to make public why the request has not been ful-filled, it SHOULD de- scribe the reason for the refusal in the entity. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is ap-plica- ble.</font </p> </font </i>	

  ||  |  |  |  |  |  |  |

Table A-1	(Cont.) Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments

Table A-1	(Cont.)	Sample Rec	uest/Respon	se for ReST Ir	jector Service
-----------	---------	------------	-------------	----------------	----------------



End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
End Point	Method	Media Type	User/ Password Wrong User/ pass	<pre>Request.xml  </pre>	Response HTTP/1.1 401 WWW- Authenticate: Basic realm="Authentic ation required" Content-Type: text/ html;charset=utf -8 Content- Language: en Content-Length: 669 Date: Thu, 05 Aug 2021 05:08:40 GMT Keep-Alive: timeout=20 Connection: keep-alive br html> <html lang="en"&gt;<head> <title>HTTP Status 401 âC" Unauthorized<!--<br-->title&gt;<style type="text/ css"&gt;body {font- family:Tahoma,Ar ial,sans- serif;} h1, h2, h3, b {color:#525D76;} h1 {font- size:16px;} h3 {font- size:12px;} a {color:black;} . line</style </title></head></html 	Comments
				<pre>:// www.oracle.com/ retail/ integration/</pre>	<pre>ground- color:#525D76;bo rder:none;}<!--<br-->style&gt;<!--</pre--></pre>	
				base/bo/WHDesc/	head> <body><h1>H</h1></body>	

#### Table A-1 (Cont.) Sample Request/Response for ReST Injector Service

End Point	Method	Media Type	User/ Password	Request.xml	Response	Comments
				<pre>vl"&gt;&lt; wh&gt;l0<!--<br-->wh&gt;<wh_nam e&gt;g<!--<br-->wh_name&gt;<!--<br-->WHDesc&gt;<!--<br-->v1:payloadXml&gt; <!--<br-->v1:ApplicationMe ssage&gt;</wh_nam </pre>	TTP Status 401 âC" Unauthorized <br h1> <hr class="line" / &gt;<b>Type</b> Status Report<!--<br-->p&gt;<b>Descript ion</b> The request has not been applied because it lacks valid authentication credentials for the target resource.<!--<br-->p&gt;<hr class="line" / &gt;<h3>Apache Tomcat/8.5.64<!--<br-->h3&gt;<!--</td--><td></td></h3></hr </hr 	

Table A-1	(Cont.) Sa	mple Reques	st/Response fo	or ReST	Injector	Service
-----------	------------	-------------	----------------	---------	----------	---------