Oracle® Retail Integration Cloud Service

Implementation Guide–Concepts

Release 21.0.000

F39236-01

May 2021

ORACLE®

Oracle Retail Integration Cloud Service Implementation Guide–Concepts, Release 21.0.000

F39236-01

Primary Author:

Contributing Author:

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

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

Contents

9 RIB Localization - Business Objects

10 Integration with External Applications

Send Us Your Comments

Oracle Retail Integration Cloud Service Implementation Guide–Concepts, Release 21.0.000

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

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

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

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

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the 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: retail-doc_us@oracle.com

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

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

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at 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 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

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

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

Functional Area	Publishing Applications	Subscribing Applications
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.

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

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

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.

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.

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 de-queue 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=19.1

  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=19.1

  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:

1. 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`.

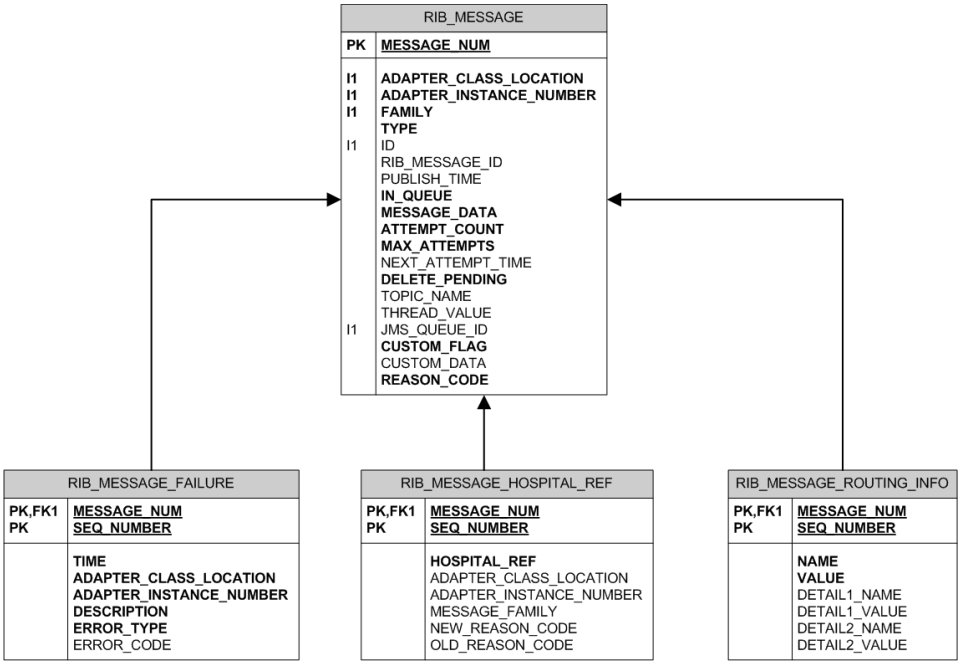
RIB Hospital Tables

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

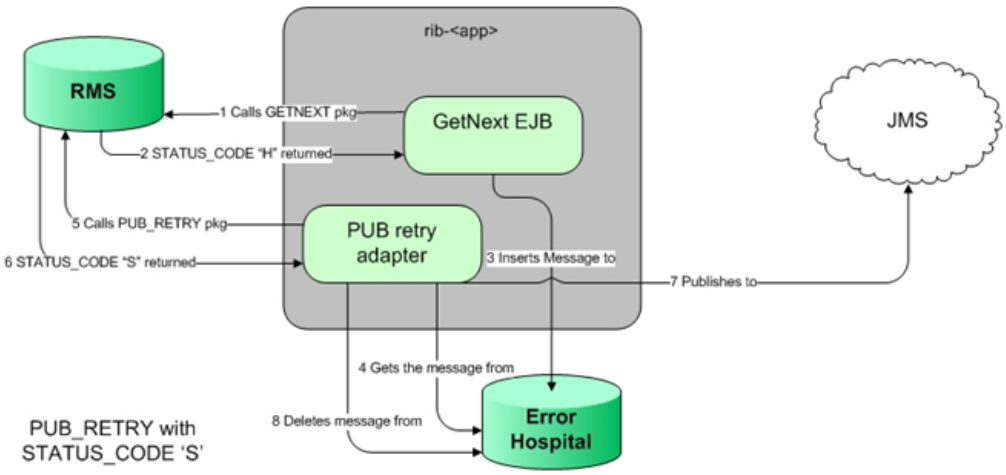
All subscriber side retrying of messages are handled by the Sub Retry Adapter. The Sub Retry Adapter looks at all messages with reason code SUB, then filters and identifies the messages that are ready to be reprocessed, keeping message ordering in mind.

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.

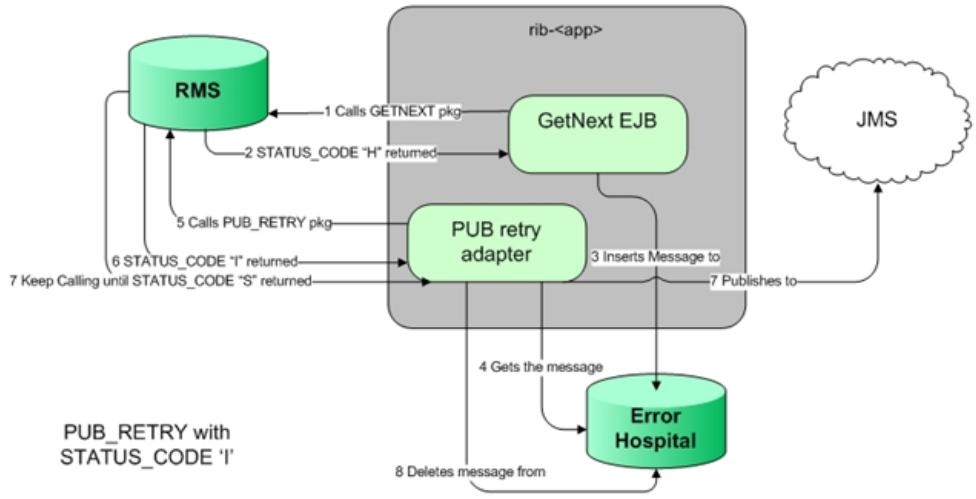
All messages with reason code of PUB are retried by the Pub Retry Adapter. RMS is the only retail application that needs the Pub Retry Adapter.

PUB Retry Adapter

The following diagrams illustrate how the PUB Retry Adapter works.

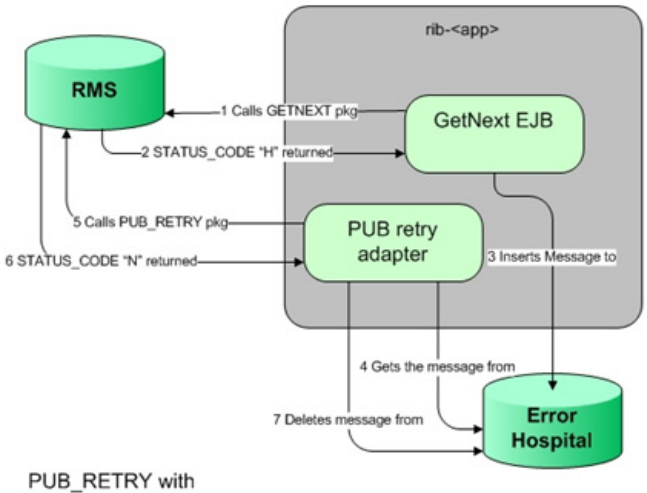


PUB_RETRY with STATUS_CODE 'S'

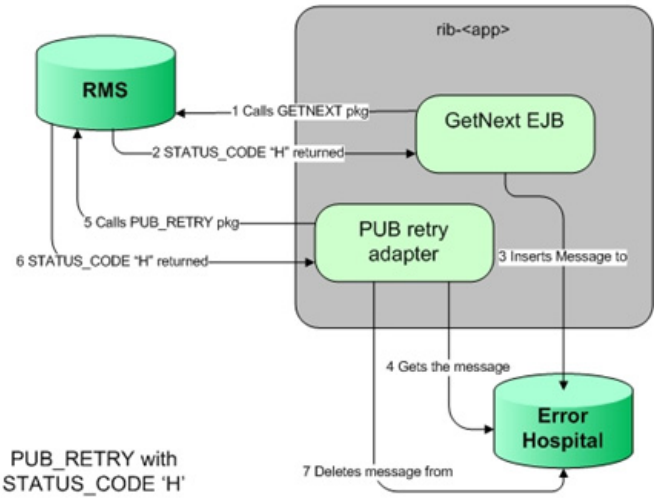


PUB_RETRY with STATUS_CODE 'I'

RIB PUB_RETRY Adapter Processing

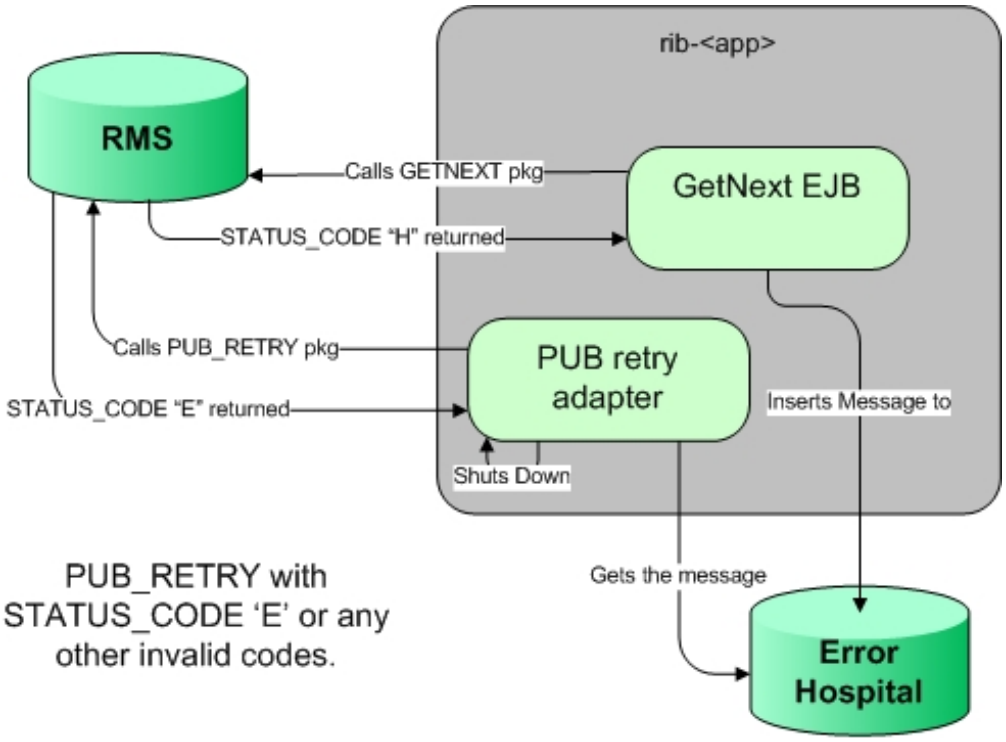


PUB_RETRY with STATUS_CODE 'N'



PUB_RETRY with STATUS_CODE 'H'

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.

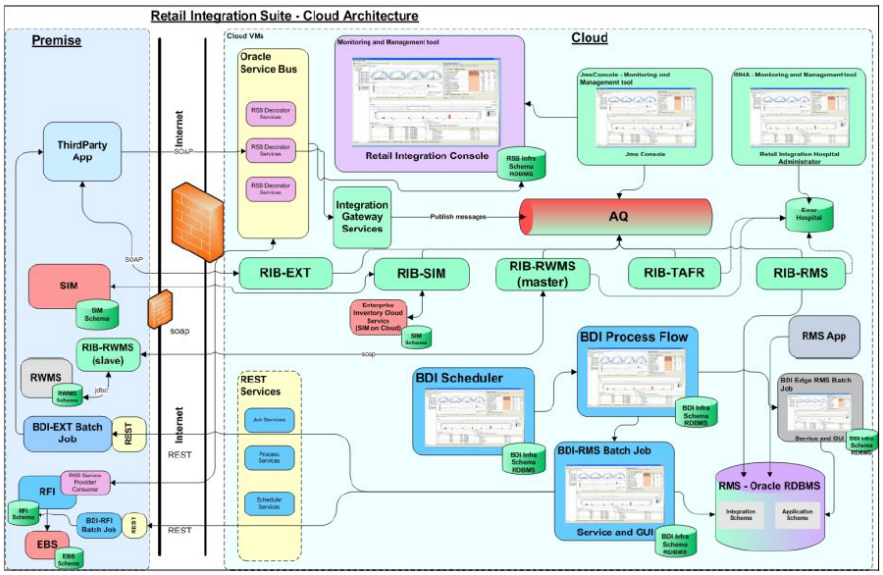
3

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 SIM and RWMS are on-premise, while RIB and BDI are 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.

Applications can invoke the new Web services to send and receive messages to/from other applications via the RIB. Client applications must use credentials of a user with the role ribAdminRole to call the publisher Web service. For consuming messages (using the injector service), applications must create a user with a IntegrationRole role on the server where the retail application is deployed. The rib-<app> must be configured with the same user credentials at install time, so that RIB can call the injector service with the correct credentials.

Configuring RIB-App as a Soap-App for Hybrid-Cloud Deployment

The following example describes the steps to configure a hybrid scenario in which SIM is on-premise and RIB is on the cloud:

- In the rib-deployment-env-info.xml file, configure the SIM application to be of type "soap-app". Under <app-in-scope-for-integration>, change SIM from javaee-app to soap-app:

  <app id="sim" type="soap-app" />

- Replace the existing rib-app section for rib-sim with below content. Edit the properties so that they apply for rib-sim. For example:

```
<rib-app id="rib-sim" type="soap-app">
<deploy-in refid="rib-sim-wls1" />
<rib-admin-gui>
<web-app-url>https://www.example.com<port>/rib-sim-appserver-gui/index.jsp</web
-app-url>         <web-app-user-alias>rib-sim_rib-admin-gui_
user-name-alias</web-app-user-alias>
</rib-admin-gui>
<error-hospital-database>
<hosp-url>jdbc:oracle:thin:@www.example.com:<port>/<service></hosp-url>
<hosp-user-alias>rib-sim_error-hospital-database_
user-name-alias</hosp-user-alias>
</error-hospital-database>
<app-database-not-applicable />
<notifications>
<email>
<email-server-host>mail.example.com</email-server-host>
    <email-server-port>25</email-server-port>
    <from-address>admin@example.com</from-address>
<to-address-list>admin@example.com</to-address-list>
    </email>
    <jmx />
</notifications>
<app id="sim" type="soap-app">
<end-point>
<url>https://www.example.com:<port>/ApplicationMessageInjectorBean/InjectorServ
ice?WSDL</url>
    <ws-policy-name>policyA</ws-policy-name>
         <user-alias>rib-sim_ws_security_user-name-alias</user-alias>
    </end-point>
</app>
</rib-app>
```

- ws-policy-name should be configured with a value “policyA” as RIB supports only this web service security policy.
- Make sure the rib-sim_ws_security_user-name-alias user is a member of the sim_integration_users group in the SIM WebLogic domain. Make sure the SIM services are up and running and can be called via the SOAP UI using the credentials that will be entered during RIB compilation.
- Compile and deploy RIB.

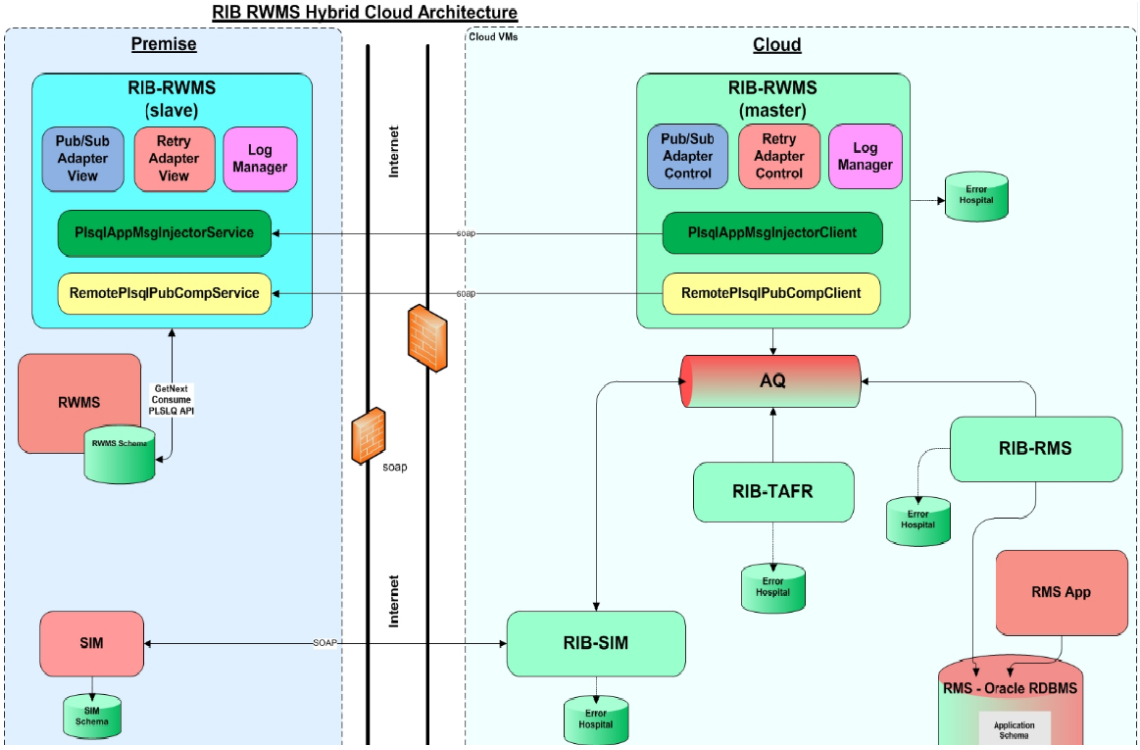
Note: The above configuration pattern is applicable to all rib-<app> except rib-rwms deployments on hybrid cloud.

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 (on-premise).

Figure 3–2 RIB-RWMS Hybrid Cloud Architecture



Configuring RIB-RWMS as Primary Application

The following example describes the steps to configure RIB-RWMS in primary/secondary pattern.

- In the rib-deployment-env-info.xml file, configure the RWMS application to be of type " master-plsql-app". Under <app-in-scope-for-integration>, change RWMS from plsql-app to master-plsql-app.

  ```
  <app id="rwms" type="master-plsql-app" />
  ```

- Replace the existing rib-app section for RIB-RWMS with a copy of the below configuration. Edit the properties so that they apply for RIB-RWMS.

  For example:

```
<rib-app id="rib-rwms" type="master-plsql-app">
    <deploy-in refid="rib-rwms-wls1" />
   <rib-admin-gui>
   <web-app-url>http://host:port/rib-rwms-appserver-gui/index.jsp</web-app-url>
   <web-app-user-alias>rib-rwms_
rib-admin-guiadmin-user-name-alias</web-app-user-alias>
   <web-app-user-alias>rib-rwms_rib-admin-gui_
operator-user-name-alias</web-app-user-alias>
   <web-app-user-alias>rib-rwms_rib-admin-gui_
monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
         <error-hospital-database>
            <hosp-url>jdbc:oracle:thin:@<host>:<port>/<service></hosp-url>
            <hosp-user-alias>rib-rwms_error-hospital-database_
user-name-alias</hosp-user-alias>
         </error-hospital-database>
         <app-database>
            <app-db-url>jdbc:oracle:thin:@<host>:<port>/<service></app-db-url>
            <app-db-user-alias>rib-rwms_app-database_
user-name-alias</app-db-user-alias>
         </app-database>
         <notifications>
            <email>
               <email-server-host>mail.example.com</email-server-host>
               <email-server-port>25</email-server-port>
               <from-address>admin@example.com</from-address>
               <to-address-list>admin@example.com</to-address-list>
            </email>
            <jmx />
         </notifications>
        <app id="rwms" type="soap-app">
            <end-point>
            <!-- URL of secondary rwms  host -->
<url>http://<host-where-rib-rwms-slave-app-is-deployed>:<port></url>
               <!-- Supported security policy names =policyC(default) OR
policyA -->
    <ws-policy-name>policyA</ws-policy-name>
               <user-alias>rib-rwms_ws_security_user-name-alias</user-alias>
            </end-point>
         </app>
      </rib-app>
```

- ws-policy-name should be configured with a value "policyA", as RIB supports only this web service security policy except for rib-oms.
- Make sure the rib-rwms_ws_security_user-name-alias user is same as rib-rwms_ rib-admin-gui_admin-user-name-alias" on secondary side.
- Compile and deploy RIB.

Note: In the above configuration the rib-rwms type is "master-plsql-app", as the primary rib-rwms application will be deployed on cloud. It does not have access to the actual app database schema that is on premise. <app-database> in the above configuration will point to the dummy schema, created during the installation. Refer to *RIB Installation Guide* for more details.

Configuring RIB-RWMS as Secondary Application

- Make the below configuration changes in the rib-home where you extracted RibKernel21.0.000ForRwmsSlave19.x.xApps_eng_ga.jar.
- In the rib-deployment-env-info.xml file, configure the RWMS application to be of type "slave-plsql-app". Under <app-in-scope-for-integration>, change RWMS from plsql-app to slave-plsql-app.

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

- Replace the existing rib-app section for RIB-RWMS with a copy of the below configuration. Edit the properties so that they apply for RIB-RWMS.

  For example:

```
<rib-app id="rib-rwms" type="slave-plsql-app <deploy-in refid="rib-rwms-wls1"
/>
<rib-admin-gui>
<web-app-url>http://<host>: <port>/rib-rwms-appserver-
gui/index.jsp</web-app-url>
<web-app-user-alias>rib-rwms_rib-admin-gui_
admin-user-name-alias</web-app-user-alias>
        <web-app-user-alias>rib-rwms_rib-admin-gui_
operator-user-name-alias</web-app-user-alias>
        <web-app-user-alias>rib-rwms_rib-admin-gui_
monitor-user-name-alias</web-app-user-alias>
 </rib-admin-gui>
<error-hospital-database>
                <hosp-url>jdbc:oracle:thin:@<host>:<port>/<service></hosp-url>
                <hosp-user-alias>rib-rwms_error-hospital-database_
user-name-alias</hosp-user-alias>
        </error-hospital-database>
        <app-database>
<app-db-url>jdbc:oracle:thin:@<host>:<port>/<service></app-db-url>
<app-db-user-alias>rib-rwms_app-database_user-name-alias</app-db-user-alias>
        </app-database>
              <notifications>
              <email>
               <email-server-host>mail.example.com</email-server-host>
               <email-server-port>25</email-server-port>
               <from-address>admin@example.com</from-address>
               <to-address-list>admin@example.com</to-address-list>
             </email>
             <jmx />
           </notifications>
           <app id="rwms" type="plsql-app">
            <jndi-not-applicable />
           </app>
        </rib-app>
```

Note: In the above configuration the RIB app (rib-rwms) type is "slave-plsql-app", as the secondary RIB-RWMS application will be deployed on premise. It does not have access to the actual error hospital and AQ database schema that is on cloud. <error-hospital-database> in the above configuration will point to the dummy schema, created during the installation. Refer to the *RIB Installation Guide* for more details.

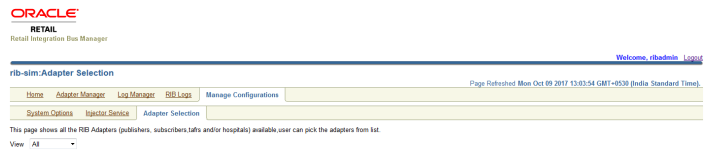
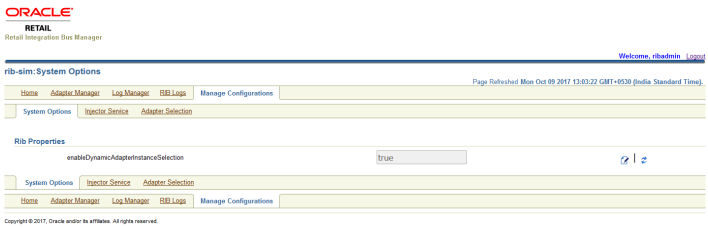
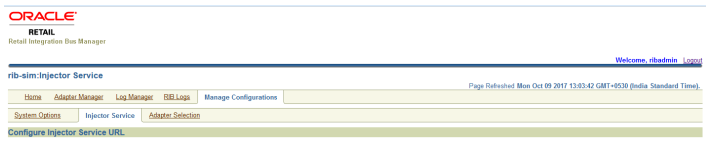
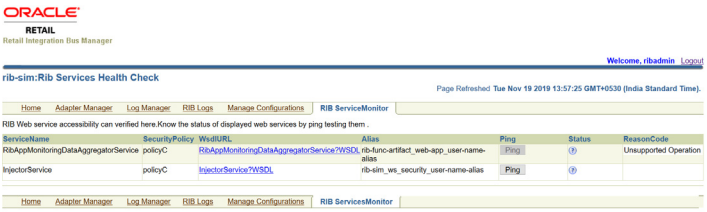
4

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>:

Table 4–1 Self-Service Feature

Self-Service Feature:	Self Service Feature on RIB-Admin GUI
Provisioning RIB adapters Choosing the subset of RIB adapters in scope for integration	
Provisioning System Options Dynamically modifying configurations via, rib-<app> properties file.	
Provisioning Injector Service URL Hook to alternate subscribing retail application installation.	
RIB ServiceMonitor Verify InjectorService provisioned in previous step.	

Provisioning RIB-Adapters

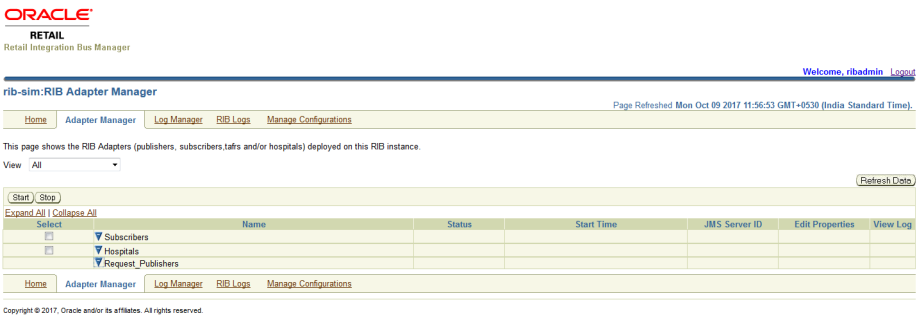
Every rib-<app> contains a set of publish and subscribing adapters for exchanging messages between retail applications. Subscribing adapters are MDB which are resource intensive. The higher the number of adapters in scope the higher is the resource crunch. In an environment which does not make use of 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.

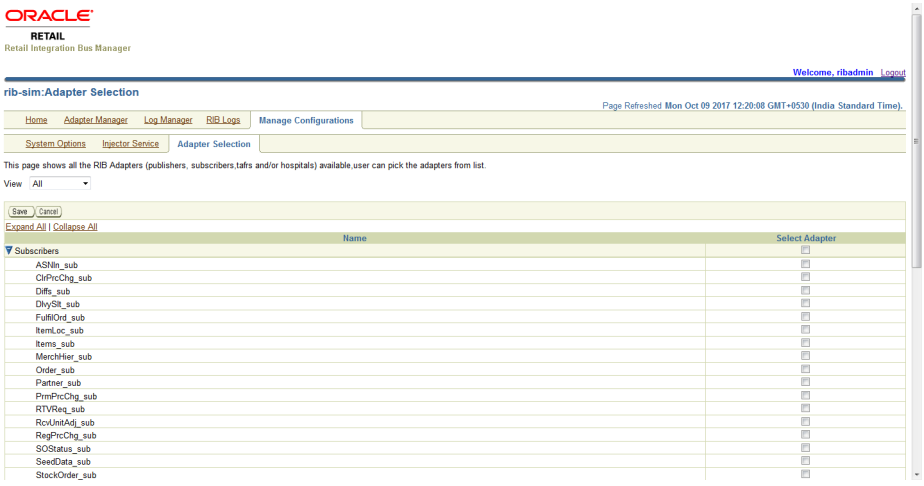
1. For every rib-<app> that needs a dynamic adapter selection to be enabled, add the property below in the rib-<app>. properties file before deploying the app.

   ```
   enableDynamicAdapterInstanceSelection=true
   ```

2. Only if the above property is set to true, the user can select the adapters dynamically. By default, there are no rib-adapters in scope of integration.



3. 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.



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

   Consider the below rib-integration flows:

   rib-sim publishing the ***InvReq*** message

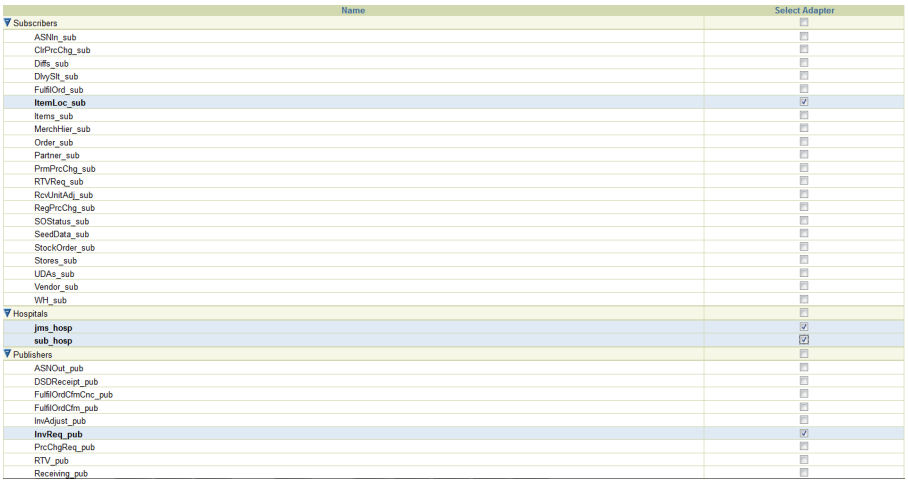
   ```
   <message-flow id="31">
   ```

```
            <node id="rib-sim.InvReq_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

```
<message-flow id="6">
            <node id="rib-rms.ItemLoc_pub" app-name="rib-rms"
                  adapter-class-def="ItemLoc_pub" type="DbToJms">
                   <in-db>default</in-db>
                   <out-topic>etItemLocFromRMS</out-topic>
            </node>
            <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.



5. Verify that the selected adapters are reflected on the Adapter Manager tab and are up and running.

6. Once the subset of adapters chosen for integration are saved, they cannot be undone.

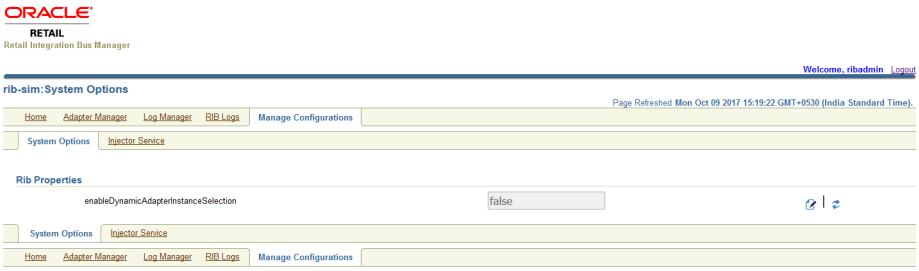
7. To have all the adapters for rib-<app> in scope by default, set the property as follows:

   `enableDynamicAdapterInstanceSelection = false`

   This is the default value for all rib-<app>s except rib-ext.

Provisioning System Options

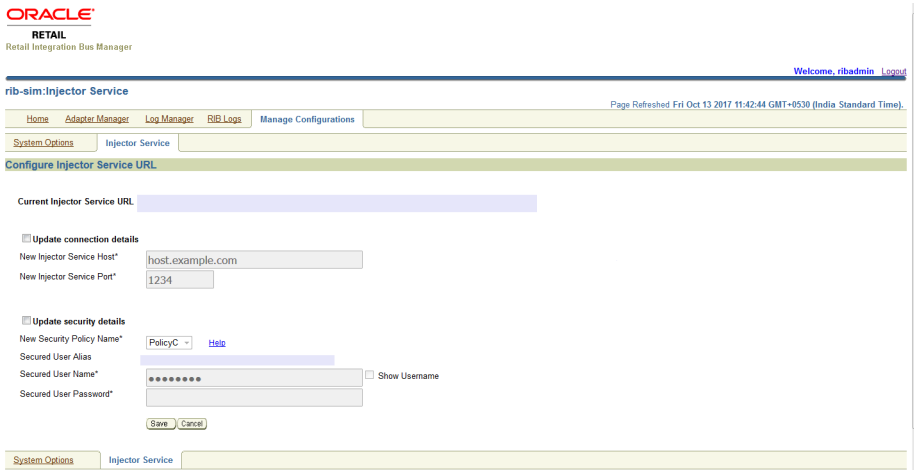
Application specific properties for the rib-<app> are configured in the rib-<app>. properties file. When RIB is deployed on cloud, the application specific properties can be configured in the RIB-Admin GUI application. The Manage Configuration > System Options tab allows the user to edit the properties values post deployment.



Provisioning InjectorService URL

The Oracle Retail application residing on-premise should be deployed as a soap-app to subscribe the messages from RIB on cloud. RIB subscribing adapters call the Injector Service hosted by the on-premise retail application to inject the messages into the retail application database.

In the RIB-Admin GUI, the Manage Configuration > Injector Service page allows the admin user to configure a different injector service URL than the one configured during the deployment. RIB dynamically updates the subscribing adapters to send the messages using the injector service. This injector service URL should be secured, by using policyA or policyC. This secured user, with which the service is secured should belong to the IntegrationGroup in myrealm of the retail application's weblogic domain.

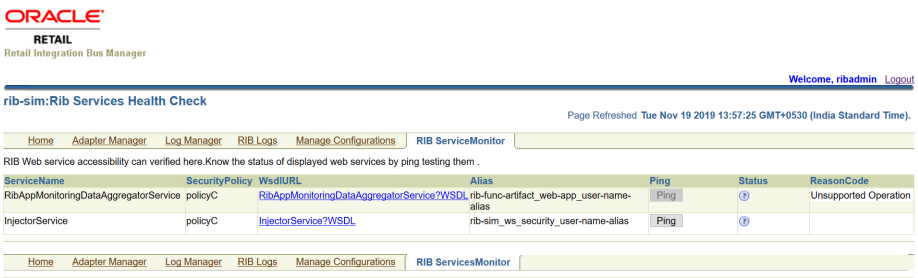


The admin user can edit the existing injector service URL details by providing new host and port details. The user also has to configure the security policy with which the new service configured is secured with, and the user credentials for RIB invoking the service.

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. Also the user can access the WSDL of these webservices. Below figure explains the same.



This page can be used to test the provisioning of InjectorService URL. Only RIBAdmin and RIBOperator users allowed to access the page.

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. See the *Oracle Retail Integration Bus Operations Guide* for how to configure parallel logical channels.

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.

6

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

WebLogic server has a default security realm. For each rib-<app>.ear deployed, RIB creates the users in the default security realm belonging to the below groups:

- ribAdminGroup
- ribOperatorGroup
- ribMonitorGroup

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.

RIB System Administrators Security Domain

The RIB System Administrators focus primarily on managing access to the RIB JMS server, application server instances, RIB Hospital database, and the rib-home workspace. RIB must be deployed with the default WebLogic admin user.

Note: For more information about security, see Oracle Retail Integration Bus Security Guide or see Chapter 7, "RIB Security," in the *Oracle Retail Integration Bus Installation Guide*.

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.

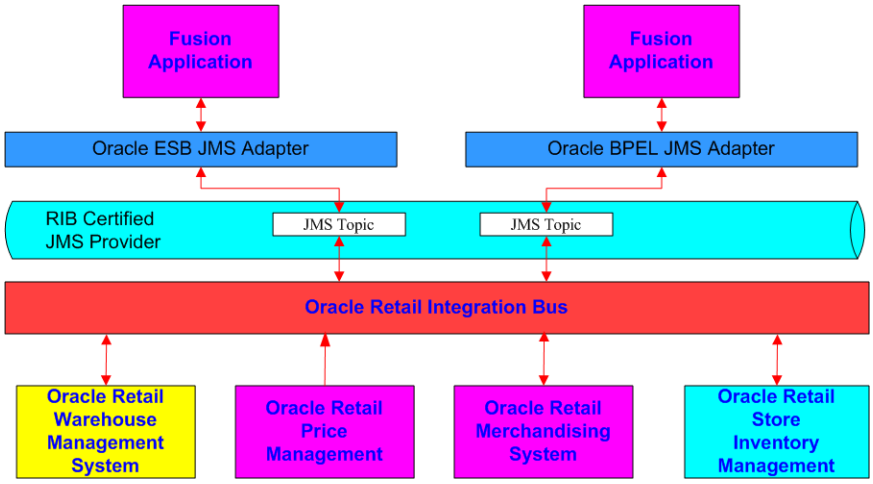
All RIB message payloads are fully standard compliant XSD based. All of the XML payloads are namespace aware and follow the general standards as well as the conventions that make them compatible with other Oracle Fusion products such ESB and BPEL. The payload schema definitions (XSDs) are packaged with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

See the *Oracle Retail Integration Bus Integration Guide* for more information.

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.

General Process of Integration

The general process for custom integration with the RIB:

- Determine the Message Family of interest (such as Items)
- Use the *Oracle Retail Integration Bus Integration Guide* to determine the message payloads and topics involved.
- Configure the JMS Adapter within the tool (ESB/BPEL) to the RIB JMS provider.
- Understand the RIB envelope (RibMessage.xsd) and the message type relationship.

- Understand the payload for each message. These are html-encode inside the RibMessage envelope.
  - The RIB XSDs are included in the *Oracle Retail Integration Bus Integration Guide* as well as the Function Artifacts war file.
- Understand the Oracle Retail Application API mappings. These are included in the *Oracle Retail Integration Bus Integration Guide*. This is important because the XSDs do not reflect the actual optional/mandatory state of an element. For historical reasons (to support previous releases), all elements in the XSD that have been added since RIB version 10.3 have been optional at the message level.
  - The Mapping reports are included with the *Oracle Retail Integration Bus Integration Guide*.
  - Each of the Oracle Retail applications has documentation on the behavior of the API.
- All RIB messages must have the message property threadValue set by publishing applications, and in a multi-channel message flow, the subscriber will need to set the message selector to an appropriate threadValue to maintain message publication sequencing.
  - Understand the relationship between the threadValue and multiple-channels within the RIB. See "Multiple Channels" in the *Oracle Retail Integration Bus Operations Guide*.
- Many of the Message Families have a RIB Component called a TAFR involved. Understand what a TAFR is and how it works within a message flow. This can be very involved in some families, and can actually create additional mandatory elements with a message that may not be obvious. See "Transform, Filtering and Routing" in the *Oracle Retail Integration Bus Operations Guide.*
- The *Oracle Retail Integration Bus Integration Guide* for each family has the general functional specifications for the TAFRs involved with that family.
- Understand the volume characteristics of a message family. The RIB is designed to handle retail volumes, so a poorly designed subscriber can have a huge impact on the JMS. Conversely, a publisher that tries to use the RIB as a bulk transfer mechanism is also inappropriate.

Configure FWM JMS Adapter to RIB AQ JMS

There is nothing special about configuration of the JMS Adapter in either ESB or BPEL to now connect to the Resource Provider configured to the RIB AQ JMS. (See Oracle Service Oriented Architecture Suite tutorials and documentation.) RIB AQ must be configured as foreign JMS, while RIB is deployed on WebLogic server.

For information about configuring foreign JMS adapter, see the WebLogic® Application Server Administrator's Guide 12c Release 1 (12.2.1.4.0).

8

RIB Customization/Extension

The customization of an Oracle Retail Application often drives requirements to customize or extend the messages that flow among the Oracle Retail applications, or to create new message flows to support new business logic.

This section discusses the customization/extension approaches and best practices (from a RIB perspective) for extending base messages, creating new messages and adapters. These are complex topics and should be performed with great care to avoid making future generally available (GA) releases difficult or impossible to accept.

Retailers often modify retail software either in-house or through third-party system integrators. The customization and extension of Oracle Retail base products and messages are not supported by Oracle Retail, including My Oracle Support. This chapter aims to mitigate the risks of unsupported customization by providing guidance and references on how to attempt to customize safely and effectively. The tools and approaches described in this chapter are complex and require a high level of skill and knowledge of the product. Any issues that may arise with custom flows, custom APIs or customized message families are the responsibility of the customer and not Oracle Retail.

Prerequisites for RIB Customization

Customization requires careful consideration and planning for extending the RIB. Planning helps to avoid re-installation or re-architecture because of operational or performance problems.

The following prerequisites help to ensure a successful customization of RIB:

- A functional RIB environment without any customizations.
- Familiarity with the Core RIB Concepts, components, and architecture, including an understanding of all of the following:
  - Oracle database triggers, RIB adapters, RIB Message envelope, RIB Message payloads and the functionality of GETNXT () and CONSUME () stored procedures.
  - Integration message flow paths.
  - RIB life cycle
  - Physical and logical requirements and limitations of the RIB components.
  - RIB operational considerations.

The tools used in the customization and extension of the RIB are separately documented. The primary tools are the Retail Functional Artifact Generator and the

rib-app-builder tools. The message (payload) structure and packaging is covered in the *Oracle Retail Functional Artifacts Guide*.

The following documents are referenced throughout this chapter and are required for the customization effort.

- *Oracle Retail Functional Artifacts Guide*
- *Oracle Retail Functional Artifact Generator Guide*
- *Oracle Retail Integration Bus Operations Guide*

Rules for Customization

Understand the following customization rules.

- Always keep an environment with a base version release to reproduce any base version issues. Only GA base code and messages are supported.
- Always take a backup of the particular files being modified during the customization, to allow for reversal of the changes.
- Always use RIB tools such as RDMT, RIHA, the PL/SQL and Java EE API-simulators (also known as Stubby) and the PLSQL and JavaEE Service Interface Testers to test the customization changes whenever possible.
- Never modify the existing base flows in rib-integration-flows.xml. Modification can cause errors in functionality that is difficult to detect. Also, modifications you make to base flows do not carry over to new releases, nor are they retained when defect fixes are applied to base code and objects.
- When customizing or extending the RIB messages or flows, all publishing and subscribing applications participating in the flow must be considered.
- In scenarios where payload customization or the addition of a new message type for a particular message family is planned, and the flow contains a TAFR, the following rules apply:
  - TAFRs that do not examine RIB Message types/payloads do not require modification.
  - For TAFRs that examine message type/payloads for filtering or transformation purposes, the TAFR implementation code must be changed. If this code is not changed, the messages will fail and land in RIB Error Hospital tables.

Message Family and Message Type Customization

In the RIB, all messages are categorized by message family and message type. A message family is specific to one or more Business Objects. It defines all publishable events occurring on the Business Object(s).

The message type classifies a specific event. For example, the Order message family is designed for messages regarding purchase orders, and the Vendor message family is associated with supplier or vendor information.

Typical message types for a message family includes at least one create, modify, and delete operation.

Note: See "Message Family and Message Types" in Chapter 3.

Adding a New Message Type

To add a new message to an existing message family, the simplest approach is to add a new message type. The first step is to determine and create the payload for the new message type. The message payload must be created following the guideline and packaging rules for RIB messages.

Note: See the *Oracle Retail Functional Artifact Generator Guide* and the *Oracle Retail Functional Artifacts Guide*.

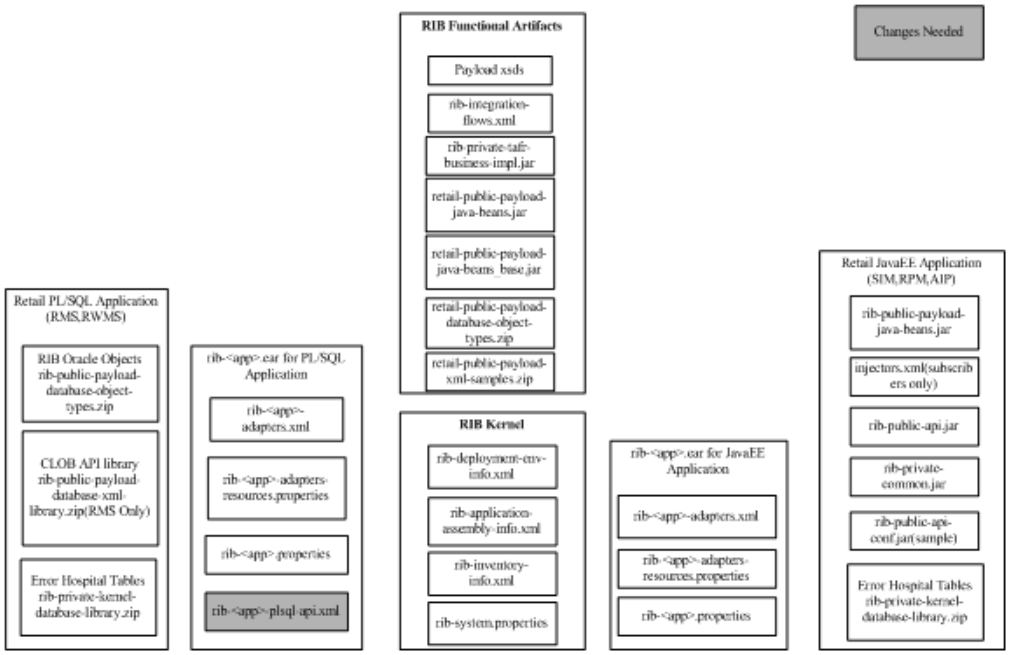
Once the desired payload is ready, follow the appropriate steps for the type of applications in the message family and the message flow.

Message Flows with PL/SQL Applications

The new message type created for an already existing or new message family must be added in the rib-<app>-plsql-api.xml of the subscribing PL/SQL retail application.

Note: No configuration changes are needed in rib-<plsql-app> whenever PL/SQL applications publish a new message type to which no PL/SQL applications subscribe.

The following illustration indicates the files that must be changed inside the RIB infrastructure during the addition of a new message type when a PL/SQL application is involved in the message flow.



Procedure for Adding a New Message Type for PL/SQL Applications

To add a new message type for PL/SQL applications, complete the following steps.

1. Add the new message type in rib-<app>-plsql-api.xml where app = rms, rfm, or rwms, present under <RIB_HOME>/application-assembly-home/rib-<app> directory.

   For example, to add a new message type, DiffGrpFooCre, for the DiffGrp message family using DiffGrpFooDesc as the payload XML that is subscribed by RWMS app: Add the message type under the <adaptorClassDef name="DiffGrp_sub"> of rib-rwms-plsql-api.xml present under <RIB_HOME>/application-assembly-home/rib-rwms as below.

```
> cd <RIB_HOME>/application-assembly-home/rib-rwms
> vi rib-rwms-plsql-api.xml

<adaptorClassDef name="DiffGrp_sub">

          <messageFamily name="DiffGrp">
              <storedProc>
                   <signature>{call RDMSUB_
DIFFGRP.CONSUME(?,?,?,?,?)}</signature>
                   <useFacilityType>true</useFacilityType>
              </storedProc>
              <messageType name="DIFFGRPDEL">
                  <oracleObject>RIB_DiffGrpRef_REC</oracleObject>
              </messageType>
              <messageType name="DIFFGRPDTLCRE">
                  <oracleObject>RIB_DiffGrpDtlDesc_REC</oracleObject>
              </messageType>
              <messageType name="DIFFGRPDTLDEL">
                  <oracleObject>RIB_DiffGrpDtlRef_REC</oracleObject>
              </messageType>
              <messageType name="DIFFGRPHDRCRE">
                  <oracleObject>RIB_DiffGrpHdrDesc_REC</oracleObject>
              </messageType>
              <messageType name="DIFFGRPDTLMOD">
                  <oracleObject>RIB_DiffGrpDtlDesc_REC</oracleObject>
              </messageType>
              <messageType name="DIFFGRPHDRMOD">
                  <oracleObject>RIB_DiffGrpHdrDesc_REC</oracleObject>
              </messageType>
              </messageType name="DIFFGRPFOOCRE">
                  <oracleObject>RIB_DiffGrpFooDesc_REC</oracleObject>
                       </messageType>
             </messageFamily>
       </adaptorClassDef>
```

2. Create a temporary working directory, customization workarea, under <RIB_HOME>/tools-home to perform any customization related tasks.

3. Using the Functional Artifact Generator tool, create custom-retail-public-payload-java-beans-<version>.jar. Copy it to the customization workarea directory created in the previous step.

```
> cd <RIB_HOME>/tools-home/customization-workarea
> cp <RIB_HOME>/application-assembly-home/rib-func-artifacts/
retail-public-payload-java-beans-<version>.jar
```

Note: See the Oracle Retail Functional Artifact Generator Guide for steps to create custom artifacts.

4. Create custom-payload.properties and add the new payload message definitions. The format of the definition is:

   "RIBFAMILY.TYPE=IMPLEMENTATION CLASS NAME"

   ```
   > vi payload.properties (make changes)
   ```

   For example, when adding the new message type, DiffGrpFooCre, under the DiffGrp message family, the custom-payload.properties file is modified as follows:

   DIFFGRP.DIFFGRPFOOCRE=com.oracle.retail.integration.custom.bo.extofdiffgrpfoodesc.v1.ExtOfDiffGrpFooDesc

   For this example, DiffGrpFooCre calls the implementation class, ExtOfDiffGrpFooDesc.

   Note: If there is a TAFR involved in the flow, the appropriate changes must be made to the TAFR to handle the new message types.

5. Add custom-payload.properties to custom-retail-public-payload-java-beans-<version>.jar.

   ```
   > jar -uvf custom-retail-public-payload-java-beans-<version>.jar
   META-INF/custom-payload.properties
   ```

6. Copy the updated custom-retail-public-payload-java-beans-<version>.jar to <RIB_HOME>/application-assembly-home/rib-func-artifacts/ directory.

7. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

   Note: See the *Oracle Retail Integration Bus Operations Guide* - (rib-app-builder tools).

   ```
   > cd <RIB_HOME>/application-assembly-home/bin
   > sh rib-app-compiler.sh
   ```

8. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows:

   ```
   > cd <RIB_HOME>/tools-home/customization-workarea
   > cp <RIB_HOME>/application-assembly-home/rib-func-artifacts/
   retail-public-payload-java-beans-<version>.jar
   ```

   This deploys the rib-func-artifact-war.

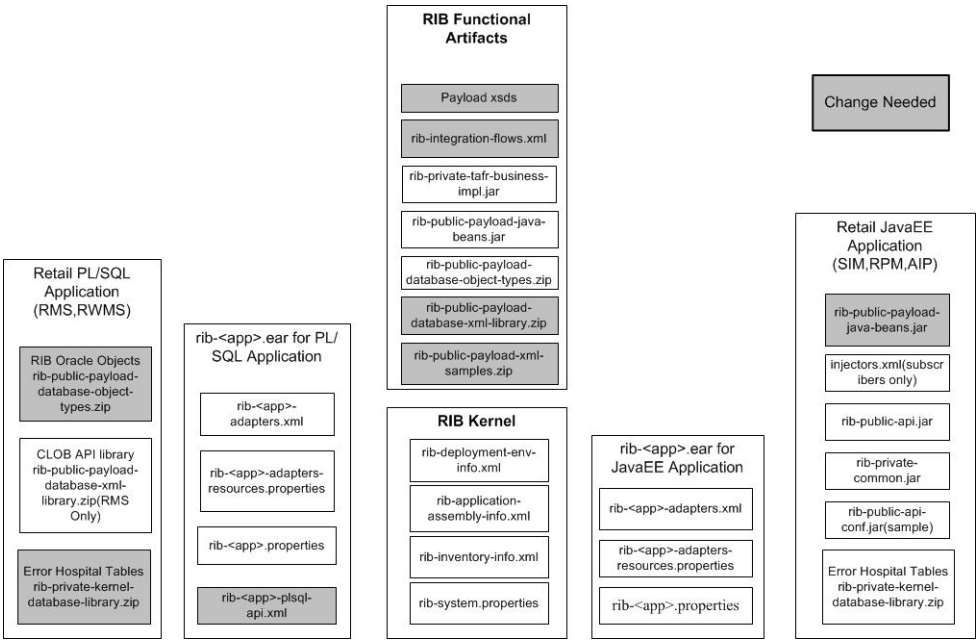
   ```
   > sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
   ```

   This deploys the rib-<app>. Repeat this step for each rib-<app> in scope for this integration environment.

   Note: The <app> must be an RMS, RFM, or RWMS application.

Message Flows with Java EE Applications

The illustration below indicates the files that must be changed inside the RIB infrastructure during the addition of a new message type when a Java EE application is involved in the message flow.



Procedure for Adding a New Message Type for Java EE Applications

1. Create a temporary working directory, customization-workarea, under <RIB_HOME>/tools-home to perform any customization related tasks
2. Go to the customization-workarea directory and create a file called custom-payload.properties.

```
> cd <RIB_HOME>/tools-home/customization-workarea
> vi custom-payload.properties
```

3. Edit the custom-payload.properties created in the step above. The custom-payload.properties would contain the new payload message definitions.

   The format of the definition is:

   "RIBFAMILY.TYPE=IMPLEMENTATION CLASS NAME"

```
> vi custom-payload.properties (make changes)
```

   For example, when adding the new message type, DiffFooCre, under the Diffs message family, the custom-payload.properties file is modified as follows:

   DIFFGRP.DIFFGRPFOOCRE=com.oracle.retail.integration.custom.bo.extofdiffgrpfoodesc.v1.ExtOfDiffGrpFooDesc

   For this example, DiffGrpFooCre calls the implementation class, ExtOfDiffGrpFooDesc.

4. If this involves a customized payload, then copy over the custom-retail-public-payload-java-beans-<version>.jar generated using Functional Artifact Generator tool to customization-workarea directory.

```
> cp
<path-to-the-jar-generated-by-artifact-generator>/custom-retail-public-payload-java-beans-<version>.jar .
For example,:cp <RIB_HOME>/tools-home/retail-func-artifact-gen/dist/custom-retail-public-payload-java-beans-<version>.jar .
and add the custom-payload.properties to the jar
> jar -uvf custom-retail-public-payload-java-beans-<version>.jar custom-payload.properties
```

5. Copy the updated custom-retail-public-payload-java-beans-<version>.jar to <RIB_HOME>/application-assembly-home/rib-func-artifacts/ directory.

   ```
   For example: cp
   <path-to-the-jar-generated-by-artifact-generator>/custom-retail-public-payload-
   java-beans-<version>.jar <RIB_
   HOME>/application-assembly-home/rib-func-artifacts/
   ```

6. Go to <RIB_HOME>/ application-assembly-home/conf and edit rib-application-assembly-info.xml.

   ```
   > cd <RIB_HOME>/ application-assembly-home/conf
   > vi rib-application-assembly-info.xml
   ```

   Add the following line, as shown in the code example below:

   ```
   <include
   name="payload-lib/custom-retail-public-payload-java-beans-<version>.jar"/>
   ```

   Note: You don't have to specify the version of the jar for entry in the rib-application-assembly-info.xml.

   If there is a TAFR involved in the flow, the appropriate changes must be made to the TAFR to handle the new message types. Refer to Message Family and Message Type Customization.

   Example:

   ```
   <rib-app id="rib-tafr" type="javaee-app">
    <ear>
    <classpath>
    <classpath refid="rib-app.global.ejb-jar.classpath"/>
    <fileset dir=".">
    <include name="lib/rib-private-tafr-business-impl.jar"/>
    <include name="lib/rib-custom-tafr-business-impl.jar"/>
    <include
   name="payload-lib/custom-retail-public-payload-java-beans-<version>.jar"/>
    <include name="payload-lib/retail-public-payload-java-beans-<version>.jar"/>
    </fileset>
    </classpath>
    <java-ee-module>
    <web-war/>
    <ejb-jar>
    <classpath>
    <classpath refid="rib-app.global.ejb-jar.classpath"/>
    <fileset dir=".">
    <include name="lib/rib-private-tafr-business-impl.jar"/>
    <include name="lib/rib-custom-tafr-business-impl.jar"/>
    <include
   name="payload-lib/custom-retail-public-payload-java-beans-<version>.jar"/>
    <include name="payload-lib/retail-public-payload-java-beans-<version>.jar"/>
    </fileset>
    </classpath>
    </ejb-jar>
    <jms-jca-connector>
    <classpath refid="rib-app.global.jms-jca-connector.classpath"/>
    </jms-jca-connector>
    </java-ee-module>
    </ear>
   ```

```
<resource>
<resource-path refid="rib-app.global.resource-path"/>
<resource-path>
<fileset dir=".">
<include name="rib-tafr.properties"/>
<include name="rib-tafr-adapters.xml"/>
<include name="rib-tafr-adapters-resources.properties"/>
</fileset>
</resource-path>
</resource>
```

7. Run the rib-app-compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory as follows.

```
>cd <RIB_HOME>/application-assembly-home/bin
>sh rib-app-compiler.sh
```

8. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-func-artifact-war
```

This deploys the rib-func-artifact-war.

```
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
```

This deploys the rib-<app>. Repeat this step for all rib-<app> that is in scope for this integration environment.

Note: The <app> must be a TAFR, SIM, AIP, or RPM application.

Note: To verify the addition of a new message type for a message family, see "Verifying the New Message Type".

Creating a New Message Family

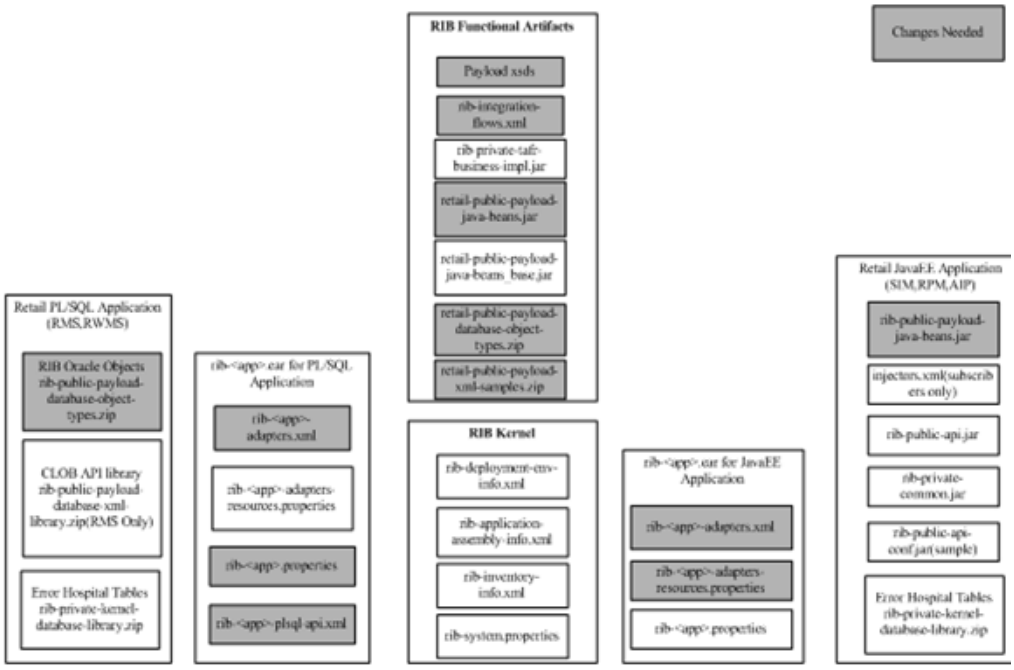
In RIB, all messages are categorized by message family and message type. One option for customizing the RIB is to create a new message family with a new publishing adapter and a new subscribing adapter.

Additional Rules

- If the new message family also corresponds to a topic, it is recommended that the customization also include the creation of a new topic for that family.
- A publishing adapter cannot publish to more than one JMS topic.
- A subscribing adapter cannot subscribe to more than one JMS topic.
- The first custom message flow must start with 901, with each subsequent custom message flow id increasing by one from 901. For example, 901, 902, 903, and so on.
- Each customized message flow id should be unique and must follow the sequence.

A new message family requires new (or custom) Oracle Retail Application side API(s). Each API should be written, installed and tested independently, and then connected to the custom message family flows.

The following illustration indicates the files that require changes during the addition of a new message family inside the RIB infrastructure:



Procedure for Adding a New Message Family

To add a new message family, complete the following steps.

1. Create a temporary working directory, customization-workarea, under <RIB_HOME>/tools-home to perform any customization related tasks.
2. Copy the rib-func-artifact.war present under <RIB_HOME>/application-assembly-home/rib-func-artifacts/ directory into <RIB_HOME>/tools-home/ customization-workarea/ directory.

   ```
   > cd <RIB_HOME>/application-assembly-home/rib-func-artifacts
   > cp rib-func-artifact.war <RIB_HOME>/tools-home/ customization-workarea
   ```

3. Extract the rib-integration-flows.xml from the copied rib-func-artifact.war requiring modification.

   ```
   > cd <RIB_HOME>/tools-home/ customization-workarea
   > jar -xvf rib-func-artifact.war integration/rib-integration-flows.xml
   ```

4. Define the entire flow for the particular message family in rib-integration-flows.xml present under /integration/ directory of <RIB_HOME>/tools-home/ customization-workarea.

   The first custom message flow should always begin with <message-flow id="901">. Each customized message flow id should be unique and must follow the sequence. Adding a new customized message flow with a message-flow ID between 1 and 900 is not recommended, as this range is reserved for adding base flows in higher versions of RIB.

   For example, when adding a new message family, Foo, that flows from the RMS application to the RWMS application, the flow is defined in rib-integration-flows.xml as follows:

   ```
   <message-flow id="901">
         <node id="rib-rms.Foo_pub" app-name="rib-rms"
               adapter-class-def="Foo_pub" type="DbToJms">
               <in-db>default</in-db>
               <out-topic>etFooFromRMS</out-topic>
   ```

```
            </node>
            <node id=" rib-rwms.Foo_sub" app-name=" rib-rwms"
                 adapter-class-def="Foo_sub" type="JmsToDb">
                  <in-topic>etFooFromRMS</in-topic>
                 <out-db>default</out-db>
        </node>
</message-flow>
```

The convention is as follows:

- node id = rib-<app>.<family>_pub or = rib-<app>.<family>_sub or could be external-system.<family>_pub or external-system.<family>_sub.
- app-name = rib-<app> is the application name. The <app> is one of the following: rms, rwms, sim, rpm aip, orfm, rob, ocds, lgf, or tafr-- or external-system.
- adapter-class-def = <family>_pub or <family>_sub.
- type = DbToJms, JmsToDb, or JmsToJms.
- <in-db> means the source of the message is a database.
- <out-db> means the destination of the message is a database.
- <out-topic> is the topic name to which the message is published.
- <in-topic> is the topic name from which the message is consumed.

5. Replace the previous existing rib-integration-flows.xml with the changed rib-integration-flows.xml in the /integration/ directory of rib-func-artifact.war under <RIB_HOME>/tools-home/customization-workarea/ directory and generate the rib-func-artifact.war as follows.

```
> cd <RIB_HOME>/tools-home/customization-workarea
> jar -uvf rib-func-artifact.war integration/rib-integration-flows.xml
```

6. Create a new publishing adapter, subscribing adapter and TAFR adapter (only if necessary), depending on the requirement for the new message family as explained later in this chapter.

   Note: See "Adding New Adapters."

7. Create the message family XSD.

   Note: See the *Oracle Retail Functional Artifact Guide* for information about adding a new payload.

   The newly created XSD should conform to the Meta schema, IntegrationMetaschema.xsd. The artifact generator tool checks the validity of the schema before generating any artifacts. If the schema is not compliant with the IntegrationXmlMetaSchema, the artifact generator fails.

8. Create a new message type.

   Note: See "Adding a New Message Type."

9. Edit the custom-payload.properties file present in /conf directory of Rib Artifact Generator tool installation. The custom-payload.properties contains the new payload message definitions. The format of the definition is:

```
"RIBFAMILY.TYPE=IMPLEMENTATION CLASS NAME"
> cd conf
> vi custom-payload.properties (make changes)
```

For example, when adding a new message type, FooCre, (under the Foo message family) that calls the implementation class, FooDesc, the custom-payload.properties file is modified as follows:

```
FOO.FOOCRE=com.oracle.retail.integration.custom.bo.extoffoodesc.v1.ExtOfFooDesc
```

Note: See the RibMessages.xsd bundled inside rib-func-artifact.war for the maximum supported length for message type.

10. Run the Artifact Generator to generate functional artifacts.

```
> $GROOVY_HOME/bin/groovy
com.oracle.retail.integration.artifact.generator.GenArtifacts.groovy -g
generateCustom
```

Upon completion of this step, the generated artifacts are in the appropriate ./output*/dist folders:
custom-retail-public-payload-database-object-types-<version>.jar and custom-retail-public-payload-java-beans-<version>.jar

11. Copy these newly generated artifacts from the appropriate ./output*/dist folders to <RIB_HOME>/application-assembly-home/rib-func-artifacts/ directory: custom-retail-public-payload-database-object-types-<version>.jar and custom-retail-public-payload-java-beans-<version>.jar.

12. New entries may be needed in RIB_SETTINGS in the RMS application database to reference the new message family only if the RMS application is in scope.

13. Run the rib-app-builder compile: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh
```

14. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows to create the new topic (etFooFromRMS) in the flow. (The prepare jms step is not destructive, so even if it is run again it would remove all the topics and recreate them.)

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -prepare-jms
```

15. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-func-artifact-war
```

This deploys the rib-func-artifact.war.

```
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
```

The rib-<app> is deployed. Repeat this step for each rib-<app> in scope for this integration environment.

Note: The <app> value must be rms, rwms, tafr, sim, rfm, aip, rob, ocds, lgf, or rpm.

Note: To verify the addition of a new message family, see "Verifying the New Message Family."

Adding New 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.

Note: See "Adapters" in Chapter 3.

Adding the Custom Adapter to the rib-integration-flows.xml File

While adding a custom publishing, subscribing or TAFR adapter, it is necessary to add or modify the message flows to which you are adding a custom adapter in the rib-integration-flows.xml, update the rib-func-artifact.war, and deploy the updated rib-func-artifact.war.

Example: Adding a new publisher Foo_pub that publishes a message for a message family Foo that flows from RMS to RWMS. We need to define the flow in rib-integration-flows.xml.

```
<message-flow id="901">
          <node id="rib-rms.Foo_pub" app-name="rib-rms"
               adapter-class-def="Foo_pub" type="DbToJms">
              <in-db>default</in-db>
              <out-topic>etFooFromRMS</out-topic>
         </node>
      <node id=" rib-rwms.Foo_sub" app-name=" rib-rwms"
              adapter-class-def="Foo_sub" type="JmsToDb">
               <in-topic>etFooFromRMS</in-topic>
               <out-db>default</out-db>
          </node>
    </message-flow>
```

Procedure for Adding the Flow to the rib-integration-flows.xml File

To add the flow to the rib-integration-flows.xml file, complete the following steps:

Note: Before adding the above flow to the rib-integration-flows.xml flow, it is recommended that a temporary working directory ("customization-workarea" under <RIB_HOME>/tools-home) be created. This directory can be used for performing any customization related tasks.

1. Copy the rib-func-artifact.war from <RIB_HOME>/application-assembly-home/rib-func-artifacts to <RIB_HOME>/tools-home/customization-workarea/ directory.

   ```
   > cd <RIB_HOME>/ application-assembly-home/rib-func-artifacts
   > cp rib-func-artifact-<version>.war  <RIB_
   HOME>/tools-home/customization-workarea
   ```

2. Extract the rib-integration-flows.xml requiring modification from the copied rib-func-artifact.war as follows:

   ```
   > jar -xvf rib-func-artifact-<version>.war integration/
   rib-integration-flows.xml
   ```

3. Add the flow shown above to the rib-integration-flows.xml.

4. Update the rib-func-artifact-<version>.war with the modified rib-integration-flows.xml.

   ```
   > jar -uvf rib-func-artifact-<version>.war integration/
   rib-integration-flows.xml
   ```

5. Copy the rib-func-artifact-<version>.war from <RIB_HOME>/tools-home/customization-workarea to <RIB_HOME>/application-assembly-home/rib-func-artifacts/ directory.

   ```
   > cd <RIB_HOME>/tools-home/customization-workarea
   > cp rib-func-artifact.war  <RIB_HOME>/
   application-assembly-home/rib-func-artifacts
   ```

6. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

   ```
   > cd <RIB_HOME>/application-assembly-home/bin
   > sh rib-app-compiler.sh
   ```

7. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

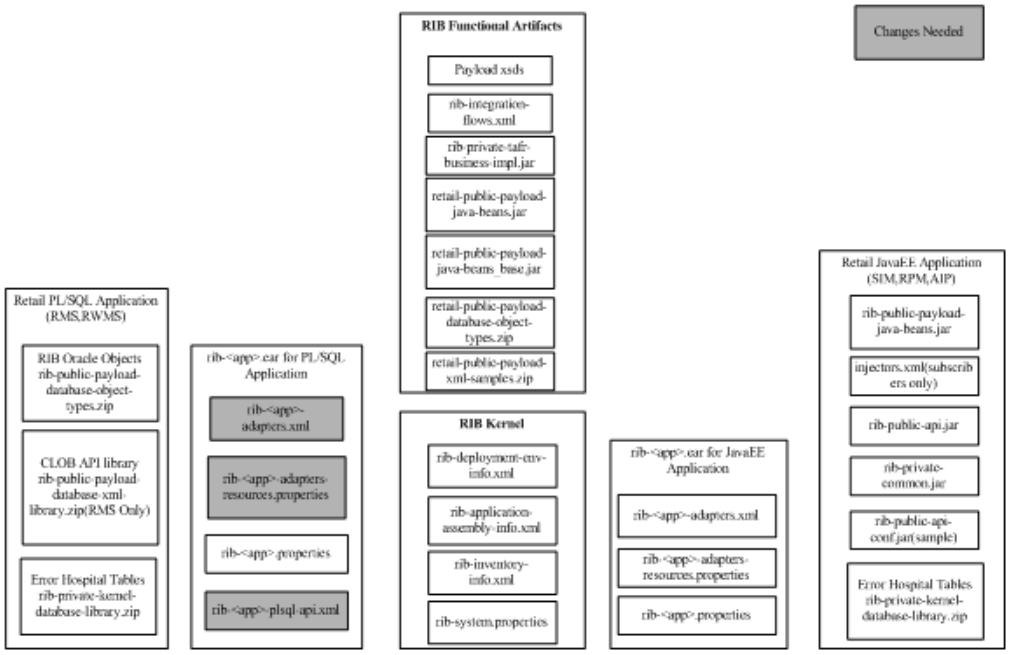
   ```
   > cd <RIB_HOME>/deployment-home/bin
   > sh rib-app-deployer.sh -deploy-rib-func-artifact-war
   ```

   The rib-func-artifact.war is deployed.

Adding a Publishing Adapter for PL/SQL Applications

Publishing adapters create messages from the information captured by the applications. These publishing adapters are designed to publish events for a single message family and are specific to an Oracle Retail application. This section explains how to create a new publishing adapter for a message family for a PL/SQL application (such as RMS and RWMS).

The illustration below indicates the files that require changes inside the RIB infrastructure for the addition of a new publishing adapter for a PL/SQL application:



Procedure for Adding a Publishing Adapter for PL/SQL Applications

To add a publishing adapter for PL/SQL applications, complete the following steps.

1. Identify the flow to which the new adapter in being added.
2. Define the name of the publishing adapter. It should always follow the naming convention, RIBFAMILY_pub_ADAPTER INSTANCE NO.
3. Define the particular publishing adapter in rib-<app>-adapters.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, RFM, or RWMS. The customer also must mention a custom attribute equal to "true" whenever a new customized publishing adapter is added.

   For example, a new publishing adapter, Foo_pub_1, (for the Foo message family) is defined in rib-<app>-adapters.xml as follows:

   ```
   <timer-driven id="Foo_pub_1" initialState="running" timeDelay="10"
   custom="true">
           <timer-task>
               <class name="com.retek.rib.app.getnext.impl.GetNextTimerTaskImpl"/>
               <property name="maxChannelNumber" value="1" />
           </timer-task>
   </timer-driven>
   ```

4. Define the particular publishing adapter in rib-<app>-adapters-resources.properties under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, ORFM, or RWMS.

   ```
   Foo_pub_1.name=Foo Publisher, channel 1
   Foo_pub_1.desc=Publisher for the Foo family through channel 1.
   ```

5. Define the particular publishing adapter in rib-<app>-plsql-api.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, ORFM, or RWMS, as shown in the example below.

Note: The signature of the stored procedure should come from the corresponding PL/SQL applications.

```
<adaptorClassDef name="Foo_pub">
 <class>com.retek.rib.collab.general.OracleObjectPublisherComponentImpl</class>
        <messageFamily name="Foo">
       <storedProc>
       <signature>{call RMSMFM_FOO.GETNXT(?,?,?,?,?,?,?,?)}</signature>
       <storedProc>
       </messageFamily>
 </adaptorClassDef>
```

6. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh
```

7. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
```

   The <app> is deployed.

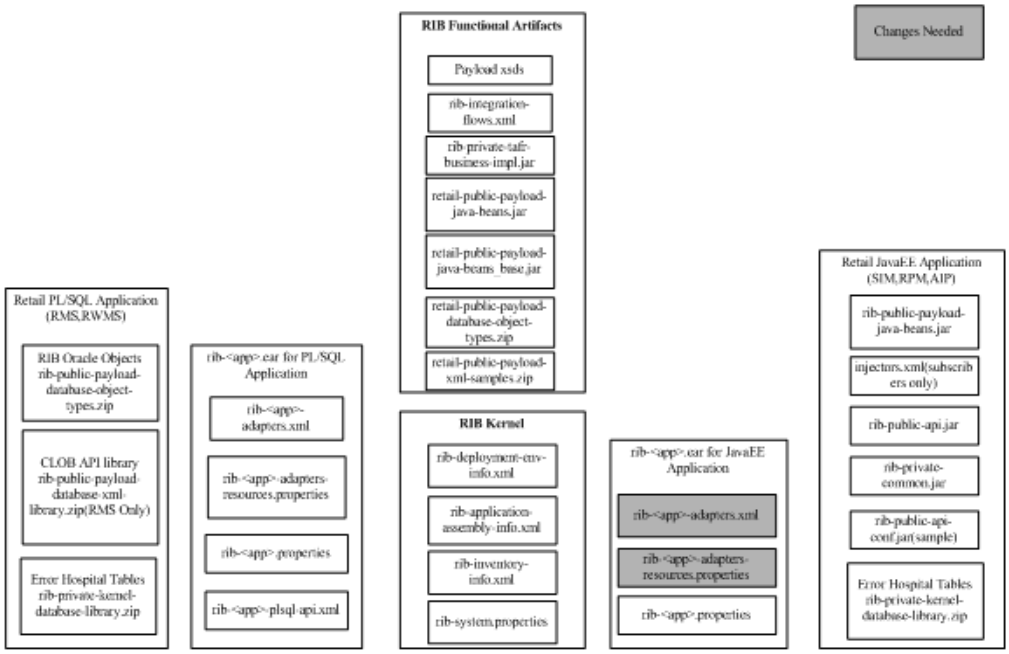
8. Make the required changes to the rib-integration-flows.xml. See "Adding the Custom Adapter to the rib-integration-flows.xml File."

Note: To verify the addition of the new adapter, see "Verifying the New Publishing Adapter."

Adding a Publishing Adapter for Java EE Applications

Publishing adapters create messages from the information captured by the applications. These publishing adapters are designed to publish events for a single message family and are specific to an Oracle Retail application. This section explains how to create a new publishing adapter for a message family for a Java EE application, such as RPM, AIP, or SIM.

The illustration below indicates the files that require changes inside the RIB infrastructure for the addition of a new publishing adapter for a Java EE application.



Procedure for Adding a Publishing Adapter for Java EE Applications

To add a publishing adapter for Java EE applications, complete the following steps.

1. Identify the flow to which the new adapter in being added.
2. Define the name of the publishing adapter. It should always follow the naming convention, RIBFAMILY_pub_ADAPTER INSTANCE NO.
3. Define the particular publishing adapter in rib-<app>-adapters.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to RPM, AIP, or SIM. The customer also must mention a custom attribute equal to "true" whenever a new customized publishing adapter is added.

   For example, a new publishing adapter, Foo_pub_1, (for the Foo message family) is defined in rib-<app>-adapters.xml as follows:

   ```
   <request-driven id=" Foo_pub_1" initialState="notConfigurable" custom="true" />
   ```

4. Define the particular publishing adapter in rib-<app>-adapters-resources.properties under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to RPM, AIP, or SIM.

   ```
   Foo_pub_1.name=Foo Publisher, channel 1
   Foo_pub_1.desc=Publisher for the Foo family through channel 1.
   ```

5. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

   ```
   > cd <RIB_HOME>/application-assembly-home/bin
   > sh rib-app-compiler.sh
   ```

6. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

   ```
   > cd <RIB_HOME>/deployment-home/bin
   > sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
   ```

The <app> is deployed.

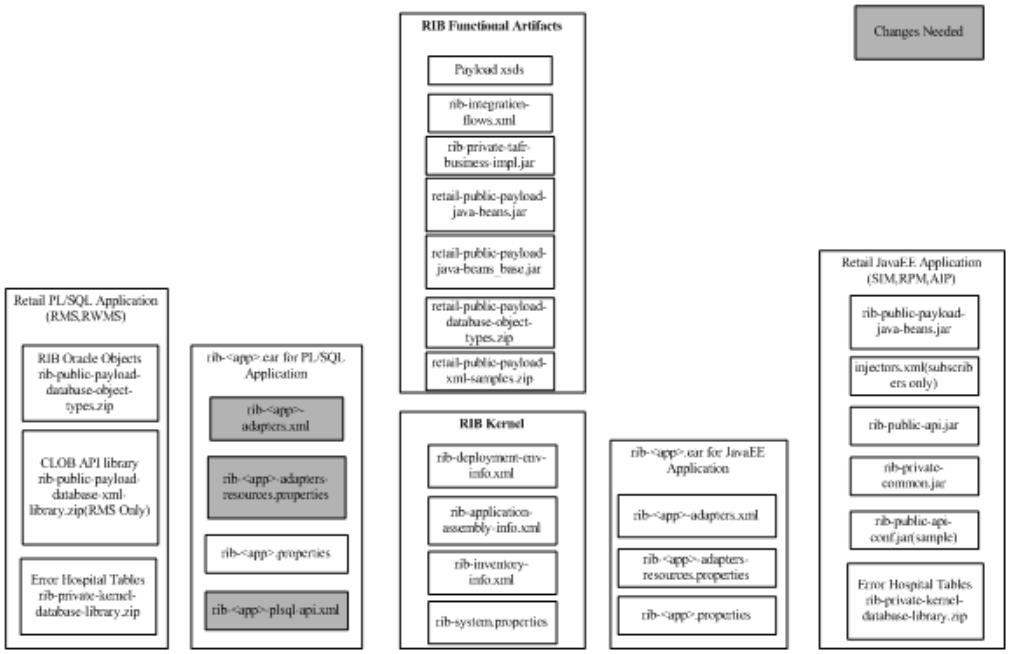
7. Make the required changes to the rib-integration-flows.xml. See "Adding the Custom Adapter to the rib-integration-flows.xml File."

Note: To verify the addition of the new adapter, see "Verifying the New Publishing Adapter."

Adding a Subscriber Adapter for PL/SQL Applications

Subscribing adapters are specific to Oracle Retail and designed to consume all messages from a specific message family.

The illustration below indicates the files that require changes inside the RIB infrastructure for the addition of a new subscriber adapter (for a particular message family) for PL/SQL applications, such as RMS, ORFM, or RWMS.



Procedure for Adding a New Subscribing Adapter for a PL/SQL Application

To add a new subscribing adapter for a PL/SQL application, complete the following steps.

1. Identify the flow to which the new adapter is being added.
2. Define the name of the subscribing adapter. It should always follow the naming convention, "RIBFAMILY_sub_ADAPTER INSTANCE NO".
3. Define the particular subscribing adapter in rib-<app>-adapters.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, ORFM, or RWMS. The customer also must mention a custom attribute equal to true whenever a new customized subscribing adapter is added.

   For example, a new subscribing adapter, Foo_sub_1, (for the Foo message family) is defined in rib-<app>-adapters.xml as follows:

```
<message-driven id="Foo_sub_1" initialState="running" custom="true"/>
```

4. Define the particular subscribing adapter in rib-<app>-adapters-resources.properties under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, ORFM, or RWMS.

```
Foo_sub_1.name=Foo Subscriber, channel 1
Foo_sub_1.desc=Subscriber for the Foo family through channel 1.
```

5. Define the particular subscribing adapter in rib-<app>-plsql-api.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to either RMS, ORFM or RWMS, as shown in the example below.

Note: The signature of the stored procedure should come from the corresponding PL/SQL applications.

```
<adaptorClassDef name="Foo_sub">
<class>com.retek.rib.collab.general.OracleObjectSubscriberComponentImpl</class>
      <messageFamily name="Foo">
            <storedProc>
            <signature>{callRMSSUB_FOO.CONSUME(?,?,?,?)}</signature>
            </storedProc>
      <messageType name=" FOOCRE">
            <oracleObject>RIB_FooDesc_REC</oracleObject>
      </messageType>
      <messageType name=" FooMOD">
            <oracleObject>RIB_FooDesc_REC</oracleObject>
      </messageType>
      <messageType name=" FooDEL">
             <oracleObject>RIB_FooRef_REC</oracleObject>
      </messageType>
      </messageFamily>
  </adaptorClassDef>
```

6. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh
```

7. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
```

   The <app> is deployed.

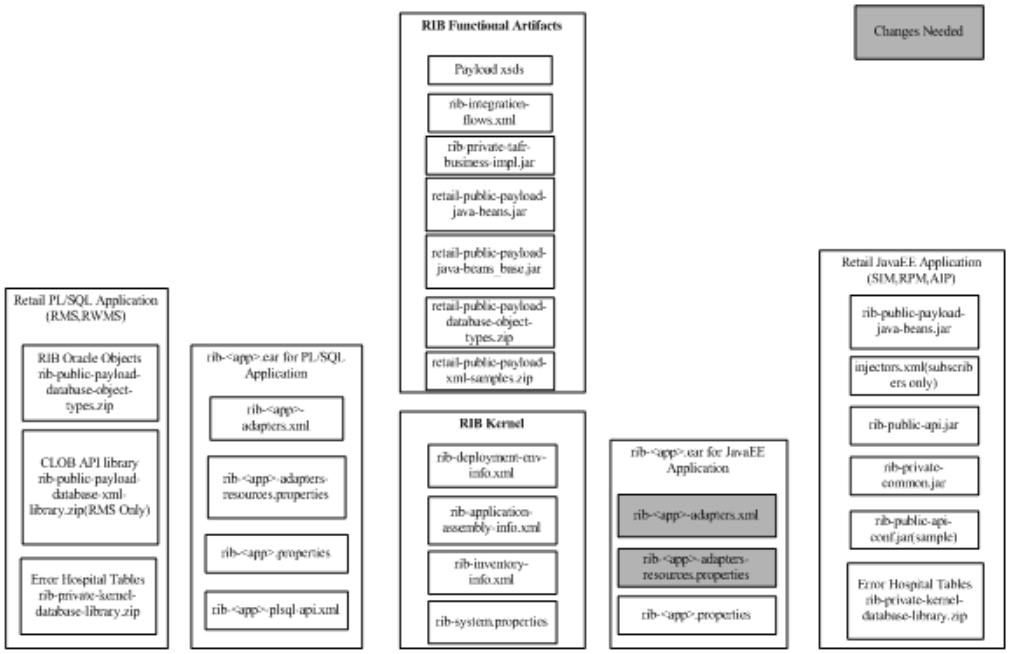
8. Make the required changes to the rib-integration-flows.xml. See "Adding the Custom Adapter to the rib-integration-flows.xml File."

Note: To verify the addition of the new adapter, see "Verifying the New Subscribing Adapter."

Adding a Subscribing Adapter for Java EE Applications

Subscribing adapters are specific to Oracle Retail and designed to consume all messages from a specific message family.

The illustration below indicates the files that require changes inside the RIB infrastructure for the addition of a new subscriber adapter (for a particular message family) for Java EE applications, such as RPM, AIP, or SIM.



Procedure for Adding a New Subscribing Adapter for a Java EE Application

1. Identify the flow to which the new adapter is being added.
2. Define the name of the subscribing adapter. It should always follow the naming convention, RIBFAMILY_sub_ADAPTER INSTANCE NO.
3. Define the particular subscribing adapter in rib-<app>-adapters.xml under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to SIM, RPM, or AIP. The customer also must mention a custom attribute equal to "true" whenever a new customized subscribing adapter is added.

   For example, a new subscribing adapter, Foo_sub_1, (for the Foo message family) is defined in rib-<app>-adapters.xml as follows:

   ```
   <message-driven id="Foo_sub_1" initialState="running" custom="true"/>
   ```

4. Define the particular subscribing adapter in rib-<app>-adapters-resources.properties under <RIB_HOME>/application-assembly-home/rib-<app>, where <app> refers to SIM, RPM, or AIP.

   ```
   Foo_sub_1.name=Foo Subscriber, channel 1
   Foo_sub_1.desc=Subscriber for the Foo family through channel 1.
   ```

5. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

   ```
   > cd <RIB_HOME>/application-assembly-home/bin
   ```

```
> sh rib-app-compiler.sh
```

6. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-<app>
```

   The <app> is deployed.

7. Make the required changes to the rib-integration-flows.xml. See "Adding the Custom Adapter to the rib-integration-flows.xml File."

> **Note:** To verify the addition of the new adapter, see "Verifying the New Subscribing Adapter."

Custom TAFR Adapters

Transformation Address Filters/Router (TAFR) adapters transform message data and route messages. Multiple, message family specific TAFRs have already been implemented. Different TAFR adapters may be active on different message families or on the same message family depending on the needs of an application. Not all message families require TAFRs.

TAFR Considerations

The following topics should be considered before writing a customized TAFR implementation for transformation, filtering or routing.

Transformation

Transformation is handled in the TAFR implementation class. Here is an example method of a TAFR that handles transformation:

```
public RibMessage transformRibMessage(RibMessage inMsg) throws TafrException {
// Transforms the incoming RibMessage into an outgoing RibMessage
RibMessage newMsg = transform(inMsg);
return newMsg; }
```

Filtering Configuration

Filtering configuration involves updating the rib-tafr.properties file with the appropriate information. The property follows the usual properties naming convention (name=value). The property used for filtering is:

```
"for.<tafr name>_tafr.drop-messages-of-types"
```

Example:

```
for.ItemsToItemsISO_
tafr.drop-messages-of-types=ISCDimCre,ISCDimMod,ISCDimDel,ItemImageCre,ItemImageMo
d,ItemImageDel,ItemUdaDateCre,ItemUdaDateMod,ItemUdaDateDel,ItemUdaFfCre,ItemUdaFf
Mod,ItemUdaFfDel,ItemUdaLovCre,ItemUdaLovMod,ItemUdaLovDel
```

This property should be read as, "for ItemsToItemsISO TAFR, drop these message types." A comma delimits the message types. If customization is required, rib-tafr.properties files must be updated for filtering to take place.

Routing

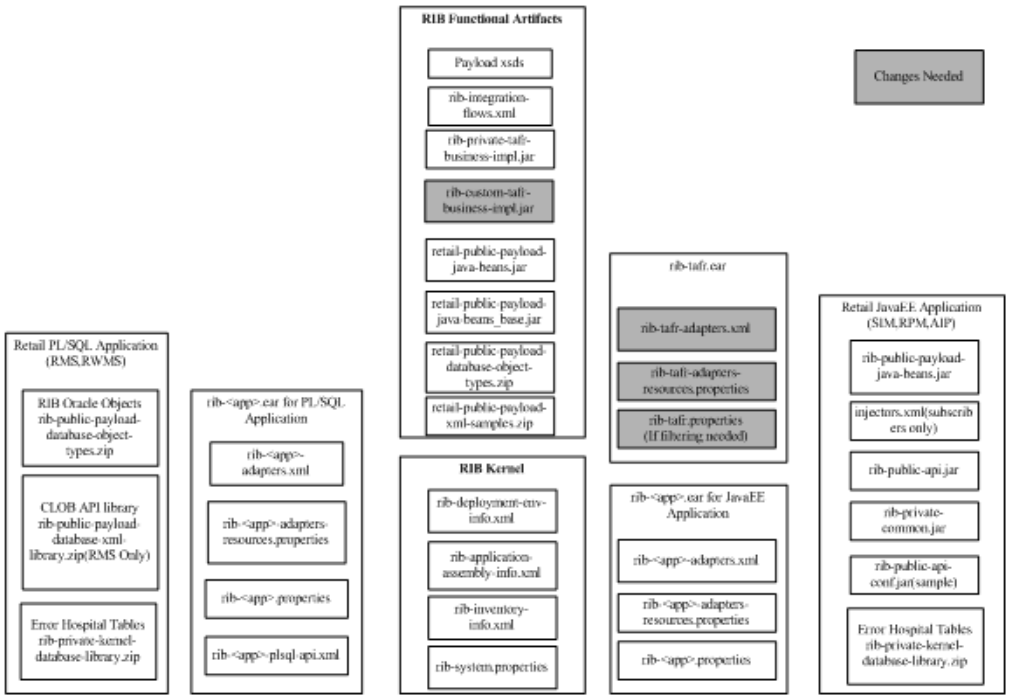
Routing is enabled by default for TAFRs; the RIB infrastructure handles this routing. If a TAFR requires routing based on message content, implementation classes override the following method.

```
public void routeRibMessage(RibMessage newMsg,MessageRouterIface router) throws
TafrException {
    router.addMessageForTopic(eventType, newMsg);
}
```

Adding a New TAFR Adapter

This section explains how to create a new TAFR adapter for a particular message family.

The illustration below indicates the files that require changes inside the RIB infrastructure during the addition of a new TAFR adapter to a message family.



Procedure for Adding a New TAFR Adapter

To add a new TAFR adapter, complete the following steps.

1. Identify the flow to which the new adapter is being added.
2. Define the name of the TAFR adapter. It should always follow the naming convention, RIBFAMILY_tafr_ADAPTER INSTANCE NO.
3. Define the corresponding implementation class name the TAFR needs to call.
4. Write the implementation class for the TAFR.

Custom TAFR Implementation

The default implementation of a TAFR implements the following interface in the RIB infrastructure:

```
package com.retek.rib.collab.tafr;

import com.retek.rib.domain.ribmessage.bo.RibMessage;

public interface TafrIface {
@return ribMessage that has been modified from the original one
public RibMessage transformRibMessage(RibMessage ribMsgIn) throws TafrException;

/**
 * Filters message or messages contents accordingly. It is possible that
 * this method could filter away the entire message thus returning null
 * from this method.
 *
 * @param ribMsg
 * @return ribMessage that may have been modified from the original one
 *  passed in or null.
 */
public RibMessage filterRibMessage(RibMessage ribMsgIn) throws TafrException;

/**
 * Routes the message to the appropriate topic for publication.
 *
 * @param ribMsg RibMessage to be routed to the appropriate topic.
 */
public void routeRibMessage(RibMessage ribMsgIn, MessageRouterIface
router) throws TafrException;

public void processRibMessage(RibMessage ribMsgIn, MessageRouterIface
router) throws TafrException;
}
```

Procedure for Completing Custom TAFR Implementation

To complete custom TAFR implementation, do the following.

1. First check if the default implementation that comes with the RIB infrastructure is appropriate.
2. Create a rib-custom-tafr-business-impl-<version>.jar containing the customized implementation class for the specific message family and replace the same under <RIB_HOME>/application-assembly-home/rib-func-artifacts.

   Note: See the My Oracle Support document, "How to Create a Custom TAFR Implementation."

3. Define the particular TAFR adapter in rib-tafr-adapters.xml under <RIB_HOME>/application-assembly-home/rib-tafr. The customer must mention a custom attribute equal to "true" whenever a new customized TAFR adapter is added.

   For example, when adding a new TAFR adapter, Foo_tafr_1, for a Foo message family, the implementation class written is SampleToSampleWH. It is under the package com.retek.rib.collab.tafr.bo.impl inside rib-custom-tafr-business-impl-<version>.jar and should be defined in rib-tafr-adapters.xml as shown below:

   ```
   <message-driven id="Foo_tafr_1" initialState="running" tafr-business-
   impl="com.retek.rib.collab.tafr.bo.impl.SampleToSampleWH" custom="true" />
   ```

4. Define the particular TAFR adapter as below in rib-tafr-adapters-resources.properties under <RIB_HOME>/application-assembly-home/rib-tafr:

```
Foo_tafr_1.name=Foo TAFR, channel 1
Foo_tafr_1.desc=TAFR for the Foo family through channel 1.
```

5. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh
```

6. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-tafr
```

7. Make the required changes to the rib-integration-flows.xml. See "Adding the Custom Adapter to the rib-integration-flows.xml File."

> **Note:** To verify the addition of the new TAFR adapter, see the section, "Verifying the New TAFR Adapter."

Changing an Existing TAFR Adapter

If there is a need to add more functionality than what is already provided for an existing TAFR, a class can be added to extend from the original TAFR class.

To change an existing TAFR adapter, complete the following steps.

1. Identify the TAFR to which more functionality should be added.
2. Define the corresponding implementation class name the TAFR needs to call. This class should extend from the original TAFR implementation class.

   For example, if additional functionality is required for the ASNOutToASNIn_tafr_1 TAFR, for which the implementation class is ASNOutToASNInLocFromRibBOImpl, a new class can be written for the additional functionality that extends from ASNOutToASNInLocFromRibBOImpl. Also, if additional functionality is needed for the transformation of the message, call the transform method of the ASNOutToASNInLocFromRibBOImpl class and write the additional code/logic.

> **Note:** For information on how to write the implementation class., see the My Oracle Support document, "How to Create a Custom TAFR Implementation."

3. Write the implementation class for the TAFR.
4. Create a rib-custom-tafr-business-impl-<version>.jar containing the implementation class and place the same under <RIB_HOME>/application-assembly-home/rib-func-artifacts.

> **Note:** For more information on how to create the rib-custom-tafr-business-impl-21.0.000.jar, see the My Oracle Support document, "How to Create a Custom TAFR Implementation."

5. Replace the name of the implementation class with the new class name in the rib-tafr-adapters.xml as shown below.

   For example, if the name of the new class name is CustomASNOutToASNInLocFromRibBOImpl, the entry in rib-tafr-adapters.xml should be:

   ```
   <message-driven id="ASNOutToASNIn_tafr_1" initialState="running"
   tafr-business-impl=" com.retek.rib.collab.tafr.bo.impl.
   CustomASNOutToASNInLocFromRibBOImpl "  custom ="true"/>
   ```

6. Run the rib-app-builder compiler: Run the rib-app-compiler.sh script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

   ```
   > cd <RIB_HOME>/application-assembly-home/bin
   > sh rib-app-compiler.sh
   ```

7. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows.

   ```
   > cd <RIB_HOME>/deployment-home/bin
   > sh rib-app-deployer.sh -deploy-rib-app-ear rib-tafr
   ```

Adding a New rib-<app>

One aspect of RIB customization/extension includes adding a new rib-app for a new application which user wants to integrate using RIB. This new app can be a PLSQL application, JavaEE application or a SOAP application. This section discusses the general steps required to add a new rib-app followed by the detailed steps for adding a rib-app specific to PLSQL Application, JavaEE Application or a SOAP application.

General steps to add a new rib-<app> are as follows:

1. Create a rib-<app> folder for new application in rib-home/application-assembly -home. Add rib-<app>-adapters.xml, rib-<app>-adapters-resources.properties, rib-<app>.properties. If new app is a plsql-app, add rib-<app>-plsql-api.xml too.
2. Update rib-<app>-adapters.xml with information about subscriber, publisher and hospital adapters.
3. Update rib-<app>-adapters-resources.properties with name, description of subscribers, publishers and hospital retriers.
4. Update rib-application-assembly-info.xml with new rib-app information under <rib-applications>.

   rib-<app>.properties file cannot be left empty, add some text (example below) in case nothing specific to rib-<app> in properties file:

   ```
   ###################################################

   # rib-<app> application specific properties go here.#

   # All properties have default values, add the      #

   # property here only if the default value does not  #

   # suit your environment.                          #
   ###################################################
   ```

5. Update the various sections of the deployment-home/conf/rib-deployment-env-info.xml file to include information about the new rib-<app>.
6. Update the rib-integration-flows.xml file to add the publisher and subscriber flow information for the new rib-app.

7. Follow RIB installation defined lifecycle steps to complete compilation and deployment of rib-<app>.ear

Adding a new PLSQL rib-<app>

Following section lists down the detailed steps required to create a new PLSQL rib-<app> corresponding to a PLSQL application named foo. The application name (foo) is for sample purposes only. The new rib application will be named rib-foo, (i.e. rib-<app> where <app> = foo). Subscribing adapter will be for family Banner and will subscribe from jms topic etBannerFromRMS. The publishing adapter will be for family CurRate and will publish to jms topic etEXTCurRate.

The publishing adapter and subscribing adapter and business objects that are used, while actual, are illustrative only and chosen for their simplicity in the example and should be replaced with the actual ones that match the business case.

Note: The application (foo.ear) that integrates with the rib-foo is designed and developed to satisfy whatever business requirements have driven the need for a new rib-<app> and is beyond the scope of this document.

Note: It is assumed that rib-home is already in place and working properly without the rib-foo application.

The following are the steps to create a new rib-foo.ear application that can communicate with a new application (called foo.ear in this example) designed to meet the business objective.

1. Go to rib-home and create a directory rib-foo under application-assembly-home.

```
> cd rib-home
> mkdir application-assembly-home/rib-foo
```

2. Add rib-foo-adapters.xml, rib-foo-adapters-resources.properties, rib-foo.properties and rib-foo-plsql-api.xml to rib-foo folder created above.

```
> touch application-assembly-home/rib-foo/rib-foo-adapters.xml
application-assembly-home/rib-foo/rib-foo-adapters-resources.properties
application-assembly-home/rib-foo/rib-foo.properties
application-assembly-home/rib-foo/rib-foo-plsql-api.xml
```

3. Add rib adapter (subscriber, publisher and hospital) details to the file application-assembly-home/rib-foo/rib-foo-adapters.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<rib-adapters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="rib-adapters.xsd" appName="rib-foo">
   <subscribers>
      <message-driven id="Banner_sub_1" initialState="running" />
   </subscribers>
   <publishers>
     <timer-driven id="CurRate_pub_1" initialState="running" timeDelay="10" >
     <timer-task>
     <class name="com.retek.rib.app.getnext.impl.GetNextTimerTaskImpl"/>
     <property name="maxChannelNumber" value="1" />
     </timer-task>
    </timer-driven>
```

```
   </publishers>
   <hospitals>
      <timer-driven id="sub_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="SUB" />
         </timer-task>
      </timer-driven>
      <timer-driven id="jms_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="JMS"/>
         </timer-task>
      </timer-driven>
   </hospitals>
</rib-adapters>
```

4. Add publisher, subscriber and hospital retrier details to the file application-assembly-home/rib-foo/rib-foo-adapters-resources.properties.

```
#
# If this changes, ManagedAdaptersResourcesPropertiesTest will need to
# change accordingly.
#
sub_all.name=Subscribers
sub_all.desc=Manages all subscribers at the same time.
Banner_sub_1.name=Banner Subscriber, channel 1
Banner_sub_1.desc=Subscriber for the Banner family through channel 1.
CurRate_pub_1.name=CurRate Publisher, channel 1
CurRate_pub_1.desc=Publisher for the CurRate  family through channel 1.
hosp_all.name=Hospital Retriers
hosp_all.desc=Manages all hospital retriers at the same time.
sub_hosp_0.name=SUB Hospital Retry
sub_hosp_0.desc=Inject messages into from the Error Hospital.
jms_hosp_0.name=JMS Hospital Retry
jms_hosp_0.desc=Re-publish messages from to JMS after JMS is brought back up.
```

5. Put the following xml content into rib-foo-plsql-api.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rib-app-plsql-api
xmlns="http://www.oracle.com/retail/integration/rib/rib-app-plsql-api"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oracle.com/retail/integration/rib/rib
-app-plsql-api.xsd">
<adaptorClassDef name="Banner_sub">
<class>com.retek.rib.collab.general.CLOBSubscriberComponentImpl</class>
<messageFamily name="Banner">
<storedProc>
<signature>{call RMSSUB_BANNERCRE.CONSUME(?, ?, ?)}</signature>
</storedProc>
</messageFamily>
</adaptorClassDef>
<adaptorClassDef name="CurRate_pub">
<class>com.retek.rib.collab.general.CLOBPublisherComponentImpl</class>
<messageFamily name="CURRATE">
<storedProc>
<signature>{call RMSMFM_CURRATE.GETNXT(?,?,?,?,?,?)}</signature>
<outParameter index="5">
<type>
<value>NUMERIC</value>
```

```
</type>
<!--NUMERIC, VARCHAR, INTEGER, FLOAT, DATE -->
<toJavaField>ID</toJavaField>
</outParameter>
<outParameter index="6">
<type>
<value>NUMERIC</value>
</type>
<!--NUMERIC, VARCHAR, INTEGER, FLOAT, DATE -->
</outParameter>
</storedProc>
</messageFamily>
</adaptorClassDef>
</rib-app-plsql-api>
```

6. Update rib-application-assembly-info.xml under application-assembly-home/conf. Near the end (before </rib-applications>), append the following section to application-assembly-home/conf/rib-application-assembly-info.xml.

```
       <rib-app id="rib-foo" type="plsql-app">
            <ear>
                <classpath refid="rib-app.global.ear.classpath" />
                <java-ee-module>
                    <web-war />
                    <ejb-jar>
                        <classpath refid="rib-app.global.ejb-jar.classpath" />
                    </ejb-jar>
                    <jms-jca-connector>
                        <classpath
refid="rib-app.global.jms-jca-connector.classpath" />
                    </jms-jca-connector>
                </java-ee-module>
            </ear>
            <resource>
                <resource-path refid="rib-app.global.resource-path" />
                <resource-path>
                    <fileset dir=".">
                        <include name="retail/remote_service_locator_info_
ribserver.xml"
/>
                        <include name="rib-foo.properties" />
                        <include name="rib-foo-adapters.xml" />
                        <include name="rib-foo-adapters-resources.properties"
/>
                        <include name="rib-foo-plsql-api.xml" />
                    </fileset>
                </resource-path>
            </resource>
        </rib-app>
```

7. Update the various sections of the deployment-home/conf/rib-deployment-env-info.xml file.

   a. Update <app-in-scope-for-integration> section to add the new application.

      Add the following XML section under <app-in-scope-for-integration>.

      ```
      <app id="foo" type="plsql-app"/>
      ```

b. Define the WebLogic Server information for the foo application. Add the following XML section under the <weblogic> (before </weblogic>). Refer to the Oracle Retail Integration Bus Installation Guide for details about the WLS path and ports.

```
<wls id="rib-foo-wls1">
        <wls-instance-name>rib-foo-server</wls-instance-name>
<wls-instance-home>webadmin@ribhost.example.com://u01/rrtswls/Oracle/Middle
ware/user_projects/domains/base_
domain/servers/rib-foo-server</wls-instance-home>
        <wls-listen-port protocol="http">7003</wls-listen-port>
        <wls-user-alias>rib-foo-wls-user-alias</wls-user-alias>
</wls>
```

c. Add the following XML section under the <rib-applications> (before </rib-applications>) section. In the jndi/url xml tag section, point it to the location where foo.ear (not rib-foo.ear) is deployed. Refer to Oracle Retail Integration Bus Installation Guide for details.

```
<rib-app id="rib-foo" type="plsql-app">
        <deploy-in refid="rib-foo-wls1" />
        <rib-admin-gui>
<web-app-url>http://ribhost.example.com:7003/rib-foo-appserver-gui/index.js
p</web-app-url>
          <web-app-user-alias>rib-foo_rib-admin-gui_
web-app-user-alias</web-app-user-alias>
        </rib-admin-gui>
        <error-hospital-database>
<hosp-url>jdbc:oracle:thin:@hospdbhost.example.com:1521:orcl</hosp-url>
           <hosp-user-alias>rib-foo_error-hospital-database_
user-name-alias</hosp-user-alias>
        </error-hospital-database>
        <app-database>
<app-db-url>jdbc:oracle:thin:@hospdbhost.example.com:1521:orcl</app-db-url>
           <app-db-user-alias>rib-foo_app-database_
user-name-alias</app-db-user-alias>
        </app-database>
        <notifications>
          <email>
              <email-server-host>mail.example.com</email-server-host>
              <email-server-port>25</email-server-port>
              <from-address>admin@example.com</from-address>
              <to-address-list>admin@example.com</to-address-list>
          </email>
          <jmx/>
        </notifications>
        <app id="foo" type="plsql-app">
          <jndi-not-applicable/>
        </app>
      </rib-app>
```

8. Update the rib-integration-flows.xml to add the publisher and subscriber flow information for the app.

a. Create a customization-workarea directory under rib-home, extract rib-integration-flows.xml present in rib-func-artifact-<version>.war.

> mkdir customization-workarea

> cd customization-workarea

> jar xf ../
application-assembly-home/rib-func-artifacts/rib-func-artifact-<version>.war
integration/rib-integration-flows.xml

b. Edit integration/rib-integration-flows.xml to add message flow details for new app (rib-foo).

a. Add the following xml section as the last node of message-flow ID number 2. Search for Banner to take you to the right message-flow.

```
<node id="rib-foo.Banner_sub" app-name="rib-foo"
       adapter-class-def="Banner_sub" type="JmsToDb">
    <in-topic>etBannerFromRMS</in-topic>
      <out-db>default</out-db>
</node>
```

b. Add the following XML section as the first node of message-flow ID number 40. Search for CurRate to take you to the right message-flow.

```
<node id="rib-foo.CurRate_pub" app-name="rib-foo"
                adapter-class-def="CurRate_pub" type="DbToJms">
<in-db>default</in-db>
<out-topic>etEXTCurRate</out-topic>
</node>
```

c. Add updated rib-integration-flows.xml to rib-func-artifact-<version>.war.

```
> jar  uvf
../application-assembly-home/rib-func-artifacts/rib-func-artifact-<version>
.war integration/rib-integration-flows.xml
```

d. Cleanup and remove the temporary working directory

```
> cd ..
> rm -rf customization-workarea
```

9. Run the rib-app-builder compiler: Run the rib-app-builder script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
>sh rib-app-compiler.sh -setup-security-credential
```

10. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows to register the new publishing and subscribing adapters in the flow. (The prepare jms step is not destructive, so even if it is run again it will just remove all the topics and recreate them.)

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -prepare-jms
```

11. Run the rib-app-builder deployer: Run the rib-app-deployer script from <RIB_HOME>/deployment-home/bin directory as follows:

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-func-artifact-war
```

This deploys the rib-func-artifact-war

```
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-foo
```

This deploys the new javaee rib-<app>.

Adding a New JavaEE rib-<app>

Following section lists down the detailed steps required to create a new JavaEE rib-<app> corresponding to a JavaEE application named foo. The application name (foo) is for sample purposes only. The new rib application will be named rib-foo, (i.e. rib-<app> where <app> = foo). Subscribing adapter will be for family Banner and will subscribe from jms topic etBannerFromRMS. The publishing adapter will be for family CurRate and will publish to jms topic etEXTCurRate.

The publishing adapter and subscribing adapter and business objects that are used, while actual, are illustrative only and chosen for their simplicity in the example and should be replaced with the actual ones that match the business case.

Note: The application (foo.ear) that integrates with rib-foo is designed and developed to satisfy whatever business requirements have driven the need for a new rib-<app> and is beyond the scope of this document.

Note: It is assumed that rib-home is already in place and working properly without the rib-foo application.

The following are the steps to create a new rib-foo.ear application that can communicate with a new application (called foo.ear in this example). Design to meet the business objective.

1. Go to rib-home and create a directory rib-foo under application-assembly-home.

```
> cd rib-home
> mkdir application-assembly-home/rib-foo
```

2. Add rib-foo-adapters.xml, rib-foo-adapters-resources.properties and rib-foo.properties to rib-foo folder created above.

```
> touch application-assembly-home/rib-foo/rib-foo-adapters.xml
application-assembly-home/rib-foo/rib-foo-adapters-resources.properties
application-assembly-home/rib-foo/rib-foo.properties
```

3. Add rib adapter (subscriber, publisher and hospital) details to the file application-assembly-home/rib-foo/rib-foo-adapters.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<rib-adapters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="rib-adapters.xsd" appName="rib-foo">
   <subscribers>
      <message-driven id="Banner_sub_1" initialState="running" />
   </subscribers>
   <publishers>
      <request-driven id="CurRate_pub_1" initialState="notConfigurable" />
   </publishers>

   <hospitals>
      <timer-driven id="sub_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="SUB" />
         </timer-task>
      </timer-driven>
```

```
      <timer-driven id="jms_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="JMS"/>
         </timer-task>
      </timer-driven>
   </hospitals>
</rib-adapters>
```

4. Add publisher, subscriber and hospital retrier details to the file application-assembly-home/rib-foo/rib-foo-adapters-resources.properties.

```
#
# If this changes, ManagedAdaptersResourcesPropertiesTest will need to
# change accordingly.
#
sub_all.name=Subscribers
sub_all.desc=Manages all subscribers at the same time.
Banner_sub_1.name=Banner Subscriber, channel 1
Banner_sub_1.desc=Subscriber for the Banner family through channel 1.
CurRate_pub_1.name=CurRate Publisher, channel 1
CurRate_pub_1.desc=Publisher for the CurRate  family through channel 1.
hosp_all.name=Hospital Retriers
hosp_all.desc=Manages all hospital retriers at the same time.
sub_hosp_0.name=SUB Hospital Retry
sub_hosp_0.desc=Inject messages into from the Error Hospital.
jms_hosp_0.name=JMS Hospital Retry
jms_hosp_0.desc=Re-publish messages from to JMS after JMS is brought back up.
```

rib-<app>.properties file cannot be left empty, add some text (example below) in case nothing specific to rib-<app> to set in properties file:

```
#######################################################

# rib-<app> application specific properties go here.#

# All properties have default values, add the      #

# property here only if the default value does not  #

# suit your environment.                    #

#######################################################
```

5. Update rib-application-assembly-info.xml under application-assembly-home/conf. Near the end (before </rib-applications>), append the following section to application-assembly-home/conf/rib-application-assembly-info.xml.

```
       <rib-app id="rib-foo" type="javaee-app">
            <ear>
                <classpath refid="rib-app.global.ear.classpath" />
                <java-ee-module>
                    <web-war />
                    <ejb-jar>
                        <classpath refid="rib-app.global.ejb-jar.classpath" />
                    </ejb-jar>
                    <jms-jca-connector>
                        <classpath
refid="rib-app.global.jms-jca-connector.classpath" />
                    </jms-jca-connector>
                </java-ee-module>
```

```
            </ear>
            <resource>
                <resource-path refid="rib-app.global.resource-path" />
                <resource-path>
                    <fileset dir=".">
                        <include name="retail/remote_service_locator_info_
ribserver.xml"
/>
                        <include name="rib-foo.properties" />
                        <include name="rib-foo-adapters.xml" />
                        <include name="rib-foo-adapters-resources.properties"
/>
                    </fileset>
                </resource-path>
            </resource>
        </rib-app>
```

6. Update the various sections of the deployment-home/conf/rib-deployment-env-info.xml file.

   a. Update <app-in-scope-for-integration> section to add the new application.

      Add the following XML section under <app-in-scope-for-integration>.

      ```
      <app id="foo" type="javaee-app"/>
      ```

   b. Define the WebLogic Server information for the foo application. Add the following XML section under the <weblogic> (before </weblogic>). Refer to the Oracle Retail Integration Bus Installation Guide for details about the WLS path and ports.

      ```
      <wls id="rib-foo-wls1">
               <wls-instance-name>rib-foo-server</wls-instance-name>

      <wls-instance-home>user@ribhost.example.com://u01/rrtswls/Oracle/Middleware
      /user_projects/domains/base_
      domain/servers/rib-foo-server</wls-instance-home>
               <wls-listen-port protocol="http">7003</wls-listen-port>
               <wls-user-alias>rib-foo-wls-user-alias</wls-user-alias>
      </wls>
      ```

   c. Add the following XML section under the <rib-applications> (before </rib-applications>) section. In the jndi/url xml tag section, point it to the location where foo.ear (not rib-foo.ear) is deployed. Refer to Oracle Retail Integration Bus Installation Guide for details.

      ```
      <rib-app id="rib-foo" type="javaee-app">
      <deploy-in refid="rib-foo-wls1"/>
      <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:7003/rib-foo-appserver-gui/index.js
      p</web-app-url>
      <web-app-user-alias>
      rib-foo_rib-admin-gui_web-app-user-alias
      </web-app-user-alias>
      </rib-admin-gui>
             <error-hospital-database>
      <hosp-url>
      jdbc:oracle:thin:@hospdbhost.example.com:1521:orcl</hosp-url>
      <hosp-user-alias>rib-foo_error-hospital-database_
      user-name-alias</hosp-user-alias>
             </error-hospital-database>
              <app-database-not-applicable/>
      ```

```
        <notifications>
          <email>
            <email-server-host>mail.example.com</email-server-host>
            <email-server-port>25</email-server-port>
            <from-address>admin@example.com</from-address>
            <to-address-list>admin@example.com</to-address-list>
         </email>
         <jmx/>
         </notifications>
         <app id="foo" type="javaee-app">
            <jndi>
             <url>t3://foohost.example.com:7002/foo</url>
<factory>weblogic.jndi.WLInitialContextFactory</factory>
              <user-alias>foo_jndi_user-name-alias</user-alias>
            </jndi>
          </app>
</rib-app>
```

7. Update the rib-integration-flows.xml to add the publisher and subscriber flow information for the app.

   a. Create a customization-workarea directory under rib-home, extract rib-integration-flows.xml present in rib-func-artifact-<version>.war.

      > mkdir customization-workarea

      > cd customization-workarea

      > jar xf ../ application-assembly-home/rib-func-artifacts/rib-func-artifact.war integration/rib-integration-flows.xml

   b. Edit integration/rib-integration-flows.xml to add message flow details for new app (rib-foo).

      a. Add the following xml section as the last node of message-flow ID number 2. Search for Banner to take you to the right message-flow.

```
<node id="rib-foo.Banner_sub" app-name="rib-foo"
       adapter-class-def="Banner_sub" type="JmsToDb">
      <in-topic>etBannerFromRMS</in-topic>
      <out-db>default</out-db>
</node>
```

      b. Add the following XML section as the first node of message-flow ID number 40. Search for CurRate to take you to the right message-flow.

```
<node id="rib-foo.CurRate_pub" app-name="rib-foo"
                adapter-class-def="CurRate_pub" type="DbToJms">
<in-db>default</in-db>
<out-topic>etEXTCurRate</out-topic>
</node>
```

   c. Add updated rib-integration-flows.xml to rib-func-artifact-<version>.war.

```
> jar  uvf
../application-assembly-home/rib-func-artifacts/rib-func-artifact-<version>
.war integration/rib-integration-flows.xml
```

   d. Cleanup and remove the temporary working directory

```
> cd ..
> rm -rf customization-workarea
```

8. Run the rib-app-builder compiler: Run the rib-app-builder script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh -setup-security-credential
```

9. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows to register the new publishing and subscribing adapters in the flow.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -prepare-jms
```

10. Run the rib-app-builder deployer: Run the rib-app-deployer script from <RIB_HOME>/deployment-home/bin directory as follows:

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-func-artifact-war
This deploys the rib-func-artifact-war
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-foo
This deploys the new javaee rib-<app>.
```

Adding a New SOAP rib-<app>

Following section lists down the detailed steps required to create a new SOAP rib-<app> corresponding to a SOAP application named foo. The application name (foo) is for sample purposes only. The new rib application will be named rib-foo, (i.e. rib-<app> where <app> = foo). Subscribing adapter will be for family Receiving and will subscribe from jms topic etReceiving. The publishing adapter will be for family FulfilOrder and will publish to jms topic etFulfillOrder.

The publishing adapter and subscribing adapter and business objects that are used, while actual, are illustrative only and chosen for their simplicity in the example and should be replaced with the actual ones that match the business case.

Note: The application (foo.ear) that integrates with the rib-foo is designed and developed to satisfy whatever business requirements have driven the need for a new rib-<app> and is beyond the scope of this document.

Note: It is assumed that rib-home is already in place and working properly without the rib-foo application.

The following are the steps to create a new rib-foo.ear application that can communicate with a new application (called foo.ear in this example) designed to meet the business objective.

1. Go to rib-home and create a directory rib-foo under application-assembly-home.

```
> cd rib-home
> mkdir application-assembly-home/rib-foo
```

2. Add rib-foo-adapters.xml, rib-foo-adapters-resources.properties and rib-foo.properties to rib-foo folder created above.

```
> touch application-assembly-home/rib-foo/rib-foo-adapters.xml
```

```
application-assembly-home/rib-foo/rib-foo-adapters-resources.properties
application-assembly-home/rib-foo/rib-foo.properties
application-assembly-home/rib-foo/rib-foo-plsql-api.xml
```

3. Add rib adapter (subscriber, publisher and hospital) details to the file application-assembly-home/rib-foo/rib-foo-adapters.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<rib-adapters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="rib-adapters.xsd" appName="rib-foo">
   <subscribers>
     <message-driven id="Receiving_sub_1" initialState="running" />
   </subscribers>
   <publishers>
     <request-driven id="FulfilOrd_pub_1" initialState="notConfigurable" />
   </publishers>
   <hospitals>
      <timer-driven id="sub_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="SUB" />
         </timer-task>
      </timer-driven>
      <timer-driven id="jms_hosp_0" initialState="running" timeDelay="10"  >
         <timer-task>
            <class name="com.retek.rib.j2ee.ErrorHospitalRetryTimerTask"/>
            <property name="reasonCode" value="JMS"/>
         </timer-task>
      </timer-driven>
   </hospitals>
</rib-adapters>
```

4. Add publisher, subscriber and hospital retrier details to the file application-assembly-home/rib-foo/rib-foo-adapters-resources.properties.

```
#
# If this changes, ManagedAdaptersResourcesPropertiesTest will need to
# change accordingly.
#
sub_all.name=Subscribers
sub_all.desc=Manages all subscribers at the same time.
Receiving_sub_1.name=Receiving Subscriber, channel 1
Receiving_sub_1.desc=Subscriber for the Receiving family through channel 1.
FulfilOrd_pub_1.name=FulfillOrder Publisher, channel 1
FulfilOrd_pub_1.desc=Publisher for the FulfilOrd  family through channel 1.
hosp_all.name=Hospital Retriers
hosp_all.desc=Manages all hospital retriers at the same time.
sub_hosp_0.name=SUB Hospital Retry
sub_hosp_0.desc=Inject messages into from the Error Hospital.
jms_hosp_0.name=JMS Hospital Retry
jms_hosp_0.desc=Re-publish messages from to JMS after JMS is brought back up.
```

5. Update rib-application-assembly-info.xml under application-assembly-home/conf. Near the end (before </rib-applications>), append the following section to application-assembly-home/conf/rib-application-assembly-info.xml.

```
      <rib-app id="rib-foo" type="soap-app">
           <ear>
               <classpath refid="rib-app.global.ear.classpath" />
               <java-ee-module>
```

```
                    <web-war />
                    <ejb-jar>
                        <classpath refid="rib-app.global.ejb-jar.classpath" />
                    </ejb-jar>
                    <jms-jca-connector>
                        <classpath
refid="rib-app.global.jms-jca-connector.classpath" />
                    </jms-jca-connector>
                </java-ee-module>
            </ear>
            <resource>
                <resource-path refid="rib-app.global.resource-path" />
                <resource-path>
                    <fileset dir=".">
                        <include name="retail/remote_service_locator_info_
ribserver.xml"
/>
                        <include name="rib-foo.properties" />
                        <include name="rib-foo-adapters.xml" />
                        <include name="rib-foo-adapters-resources.properties"
/>
                    </fileset>
                </resource-path>
            </resource>
        </rib-app>
```

rib-<app>.properties file cannot be left empty, add some text (example below) in case nothing specific to rib-<app> to set in properties file

```
#######################################################

# rib-<app> application specific properties go here.#

# All properties have default values, add the      #

# property here only if the default value does not  #

# suit your environment.                  #

#######################################################
```

6. Update the various sections of the deployment-home/conf/rib-deployment-env-info.xml file.

   a. Update <app-in-scope-for-integration> section to add the new application.

      Add the following XML section under <app-in-scope-for-integration>

      ```
      <app id="foo" type="soap-app"/>
      ```

   b. Define the WebLogic Server information for the foo application. Add the following XML section under the <weblogic> (before </weblogic>). Refer to the Oracle Retail Integration Bus Installation Guide for details about the WLS path and ports.

      ```
      <wls id="rib-foo-wls1">
              <wls-instance-name>rib-foo-server</wls-instance-name>
      <wls-instance-home>webadmin@ribhost.example.com://u01/rrtswls/Oracle/Middle
      ware/user_projects/domains/base_
      domain/servers/rib-foo-server</wls-instance-home>
              <wls-listen-port protocol="http">7003</wls-listen-port>
              <wls-user-alias>rib-foo-wls-user-alias</wls-user-alias>
      </wls>
      ```

c. Add the following XML section under the <rib-applications> (before </rib-applications>) section. In the jndi/url xml tag section, point it to the location where foo.ear (not rib-foo.ear) is deployed. Refer to the *Oracle Retail Integration Bus Installation Guide* for details.

```
<rib-app id="rib-foo" type="soap-app">
<deploy-in refid="rib-foo-wls1"/>
<rib-admin-gui>
<web-app-url>https://ribhost.example.com:8108/rib-foo-appserver-gui/index.j
sp</web-app-url>
<web-app-user-alias>rib-foo_rib-admin-gui_
web-app-user-alias</web-app-user-alias>
</rib-admin-gui>
<error-hospital-database>
<hosp-url>jdbc:oracle:thin:@hospdbhost.example.com:1521:orcl</hosp-url>
<hosp-user-alias>rib-foo_error-hospital-database_
user-name-alias</hosp-user-alias>
</error-hospital-database>
<app-database-not-applicable/>
<notifications>
<email>
<email-server-host>mail.example.com</email-server-host>
<email-server-port>25</email-server-port>
<from-address>admin@example.com</from-address>
<to-address-list>admin@example.com</to-address-list>
</email>
<jmx/>
</notifications>
<app id="foo" type="soap-app">
<end-point>
<url>http://hostname:9001/injector-service/InjectorService</url>
<ws-policy-name>policyA/policyB</ws-policy-name>
<user-alias>rib-foo_ws_security_user-name-alias</user-alias>
</end-point>
</app>
</rib-app>
```

7. Update the rib-integration-flows.xml to add the publisher and subscriber flow information for the app.

a. Create a customization-workarea directory under rib-home, extract rib-integration-flows.xml present in rib-func-artifact-<version>.war.

```
> mkdir customization-workarea
> cd customization-workarea
> jar xf ../
application-assembly-home/rib-func-artifacts/rib-func-artifact-<version>.wa
r integration/rib-integration-flows.xml
```

b. Edit integration/rib-integration-flows.xml to add message flow details for new app (rib-foo).

a. Add the following xml section as the last node of message-flow ID number 28. Search for Receiving to take you to the right message-flow.

```
<node id="rib-foo.Receiving_sub" app-name="rib-foo"
                adapter-class-def="Receiving_sub" type="JmsToDb">
       <in-topic>etReceiving</in-topic>
       <out-db>default</out-db>
</node>
```

b. Add the following XML section as the first node of message-flow ID number 65. Search for FulfilOrd to take you to the right message-flow.

```
<node id="rib-foo.FulfilOrd_pub" app-name="rib-foo"
                adapter-class-def="FulfilOrd_pub" type="DbToJms">
      <in-db>default</in-db>
      <out-topic>etFulfillOrder</out-topic>
</node>
```

c. Add updated rib-integration-flows.xml to rib-func-artifact.war.

```
> jar  uvf
../application-assembly-home/rib-func-artifacts/rib-func-artifact-<version>
.war integration/rib-integration-flows.xml
```

d. Cleanup and remove the temporary working directory

```
> cd ..
> rm -rf customization-workarea
```

8. Run the rib-app-builder compiler: Run the rib-app-builder script from <RIB_HOME>/application-assembly-home/bin directory to generate/assemble a rib-<app> and make it ready for deployment.

```
> cd <RIB_HOME>/application-assembly-home/bin
> sh rib-app-compiler.sh -setup-security-credential
```

9. Run the rib-app-builder deployer: Run the rib-app-deployer.sh script from <RIB_HOME>/deployment-home/bin directory as follows to register the new subscribing and publishing adapters in the flow.

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -prepare-jms
```

10. Run the rib-app-builder deployer: Run the rib-app-deployer script from <RIB_HOME>/deployment-home/bin directory as follows:

```
> cd <RIB_HOME>/deployment-home/bin
> sh rib-app-deployer.sh -deploy-rib-func-artifact-war
```

This deploys the rib-func-artifact-war

```
> sh rib-app-deployer.sh -deploy-rib-app-ear rib-foo
```

This deploys the new soap rib-<app>.

Verification of RIB Customizations

This section explains how to verify the various customizations using the RIB diagnostic and test tools, RDMT, the PL/SQL API simulator, and the Java EE API simulator.

These verification tests are described only from a RIB perspective and not as end-to-end testing. They should be considered only as the first step in a process to move the customizations through the RIB life cycle.

The verification steps assume that these RIB tools have already been installed and are in working condition.

Note: See "Testing the RIB" in the *Oracle Retail Integration Bus Operations Guide*.

Verifying the New Message Type

To verify the addition of a new message type under a message family from a RIB perspective, complete the following steps.

1. Log in to the RDMT main menu.
2. Select menu option 3 - PUB/SUB/TAFR Msg Menu.
3. Publish a message using 8 - EJB Publish Utility.
4. Provide the new message type when prompted for the <type> parameter.
5. Use the sample message that was generated using the RIB Artifact Generator tool after adding the new message type for the corresponding message family.
6. Check the corresponding adapter's RIBLOGS to be sure the message was published successfully. The logs are written to the path, <rib-application_instance_home>/<rib-app>/logs/<rib-app>.

   For example, for /home/dev01/Weblogic12.2/Oracle/Middleware/user_projects/domains/rib_domain/servers/rib-rms-server/logs/rib-rms, the RIBLOG filenames are in the format, <adapter-instance-name>.rib.log.

   Example:

   ```
   Alloc_pub_1.rib.log
   ASNIn_sub_1.rib.log
   ```

7. Enable the RIB Audit Logs for all the corresponding adapters involved in the message flow. The auditing feature logs the message as it passes though the RIB infrastructure. This helps the tracing of message content from publication to subscription and all steps, such as a TAFR, in between.

   Note: To enable RIB Audit logs, see the section, "RIB Logging," in the *Oracle Retail Integration Bus Operations Guide* .

8. Check the RIB audit logs for the particular message family adapters (publisher, subscriber, and TAFR if involved) and verify whether the new message type is part of the message header. Also ensure that the message passes successfully through all the adapters involved in the particular message flow.
9. Check whether the new message type was successfully consumed by the subscribing adapter. The CONSUME API call from the subscribing adapter should successfully return the status S.

Verifying the New Message Family

To verify the addition of a new message family in the RIB, complete the following steps:

1. Once RIB is compiled and deployed (after adding a new message family), check whether the new family adapters (publisher, subscriber, and TAFR if involved) are visible through RIB Admin GUI.

   The RIB admin GUI can be accessed via the URL as below.

   ```
   http://<server>.example.com:<http-port>/rib-<app>-admin-gui/
   ```

   - Replace <server> with the name or IP address of the server in the environment where the rib-<app> is deployed.

- Replace <http-port> with the port number that the WebLogic managed server instance is listening on (for example, 7777).
- Replace <app> with rms, tafr, rwms, sim, rfm, aip, lgf, ocds, rob, or rpm.

Note: See the section, "Admin GUI," in the *Oracle Retail Integration Bus Operations Guide*.

2. Log in to the RDMT main menu.
3. Select menu option 3 - PUB/SUB/TAFR Msg Menu.
4. Publish a message using 8 - EJB Publish Utility.
5. Provide the new message family when prompted for the <family> parameter.
6. Use the sample message created by the Functional Artifact Generator.

Note: See the *Oracle Retail Functional Artifact Generator Guide*.

7. Check the corresponding adapter's RIBLOGS to be sure the message was published successfully. The logs are written to the path, <rib-application_managed_server>/logs/<rib-app>.

   For example, for "/u00/webadmin/product/12.2.1.4.0/WLS/user_projects/domains/rib_domain/servers/rib-rms-server/logs/rib-rms" the RIBLOG filenames are in the format, <adapter-instance-name>.rib.log.

   Example:

   ```
   Foo_pub_1.rib.log
   Foo_sub_1.rib.log
   ```

8. Also enable the RIB Audit Logs for all the corresponding adapters involved in the message flow. The auditing feature logs the message as it passes though the RIB infrastructure. This helps the tracing of message content from publication to subscription and all steps, such as a TAFR, in between.

Note: To enable RIB Audit logs, see "RIB Logging" in the *Oracle Retail Integration Bus Operations Guide*.

9. Check the RIB audit logs for the particular message family adapters (publisher, subscriber, and TAFR if involved) and verify whether the new message family is part of the message header. Also ensure that the message passes successfully through all the adapters involved in the particular message flow.

Verifying the New Publishing Adapter

To verify the addition of a new publishing adapter for PL/SQL or Java EE applications, complete the following steps:

1. Once the RIB has been compiled and deployed (after adding a new publishing adapter), check whether the new publishing adapter is visible through RIB Admin GUI.

   The RIB admin GUI can be accessed via the URL as below:

```
http://<server>.example.com:<http-port>/rib-<app>-admin-gui/
```

- Replace <server> with the name or IP address of the server in the environment where the rib-<app> is deployed.
- Replace <http-port> with the port number that the WebLogic managed server instance is listening on (for example, 7777).
- Replace <app> with rms, rwms, sim, rfm, aip, lgf, ocds, rob, or rpm.

Note: See "Admin GUI" in the *Oracle Retail Integration Bus Operations Guide*

2. Log in to the RDMT main menu.
3. Select menu option 3, PUB/SUB/TAFR Msg Menu.
4. Publish a message using 8 - EJB Publish Utility.
5. Use the sample message created by the Functional Artifact Generator for the corresponding message family.
6. Check the corresponding adapter's RIBLOGS to be sure the message was published successfully. The logs are written to the path, <rib-application_managed_server>/logs/<rib-app>.

   For example, for "/u00/webadmin/product/12.2.1/WLS/user_projects/domains/rib_domain/servers/rib-rms-server/logs/rib-rms" the RIBLOG filenames are in the format, <adapter-instance-name>.rib.log.

   Example:

   ```
   Foo_pub_1.rib.log
   ```

7. Also enable the RIB Audit Logs for the corresponding publishing adapter involved in the message flow. The auditing feature logs the message as it passes though the RIB infrastructure. This helps the tracing of message content from publication to subscription.

   Note: To enable RIB Audit logs, see "RIB Logging" in the *Oracle Retail Integration Bus Operations Guide*.

8. Check the RIB audit logs for the particular publishing adapter and verify whether the message content is displayed correctly as published. Also ensure that the message passes successfully through all the adapters involved in the particular message flow.

Verifying the New Subscribing Adapter

To verify the addition of a new subscribing adapter for PL/SQL or Java EE applications, complete the following steps:

1. Once the RIB has been compiled and deployed (after adding a new subscribing adapter), check whether the new subscribing adapter is visible through RIB Admin GUI.

   The RIB admin GUI can be accessed via the URL as below.

   ```
   http://<server>.example.com:<http-port>/rib-<app>-admin-gui/
   ```

- Replace <server> with the name or IP address of the server in the environment where the rib-<app> is deployed.
- Replace <http-port> with the port number that the WebLogic managed server instance is listening on (for example, 7777).
- Replace <app> with rms, rwms, sim, rfm, aip, lgf, ocds, rob, or rpm.

Note: See the section,"Admin GUI," in the *Oracle Retail Integration Bus Operations Guide*.

2. Log in to the RDMT main menu.
3. Select menu option 3 - PUB/SUB/TAFR Msg Menu.
4. Publish a message using 1 - Publish Msg Utility to the topic from which the newly added subscriber has to subscribe the message.
5. Use the sample message.
6. Check the corresponding adapter's RIBLOGS to be sure the message was subscribed from the topic successfully. The logs are written to the path, <rib-application_instance_home>/logs/<rib-app>.

   For example, "/u00/webadmin/product/12.2.1/WLS/user_projects/domains/rib_domain/servers/rib-rms-server/logs/rib-rms" the RIBLOG filenames are in the format, <adapter-instance-name>.rib.log.

   Example:

   ```
   Foo_pub_1.rib.log
   ```

7. Also enable the RIB Audit Logs for the corresponding subscribing adapter involved in the message flow. The auditing feature logs the message as it passes though the RIB infrastructure. This helps the tracing of message content from publication to subscription.

   Note: To enable RIB Audit logs, see "RIB Logging" in the *Oracle Retail Integration Bus Operations Guide*.

8. Check the RIB audit logs for the particular message family's subscribing adapter and verify whether the message content is displayed correctly. Also ensure that the message is subscribed successfully by the subscribing adapter.

Verifying the New TAFR Adapter

To verify the addition of a new TAFR adapter, complete the following steps:

1. Once the RIB has been compiled and deployed (after adding a new TAFR adapter), check whether the new TAFR adapter is visible through RIB Admin GUI.

   The RIB Admin GUI can be accessed via the URL as below.

   ```
   http://<server>.example.com:<http-port>/rib-tafr-admin-gui/
   ```

   - Replace <server> with the name or IP address of the server in the environment where the rib-<app> is deployed.

- Replace <http-port> with the port number that the WebLogic managed server instance is listening on (for example, 7777).

  Note: See the section, "Admin GUI," in the *Oracle Retail Integration Bus Operations Guide*.

2. Log in to the RDMT main menu.
3. Select menu option 3, PUB/SUB/TAFR Msg Menu.
4. Publish a message using 1—Publish Msg Utility to the topic from which the newly added TAFR has to subscribe the message.
5. Use the sample message generated by the RIB Artifact Generator tool for the corresponding message family.
6. Check the corresponding TAFR adapter's RIBLOGS to be sure the message was subscribed by the TAFR from the particular topic and again published to the next destination topic successfully. The logs are written to the path, <rib-application_instance_home>/logs/rib-tafr.

   For example, for "/u00/webadmin/product/12.2.1/WLS/user_projects/domains/rib_domain/servers/rib-rms-server/logs/rib-rms" the RIBLOG filenames are in the format, <adapter-instance-name>.rib.log.

   Example:

   ```
   SampleToSampleWH_tafr_1.rib.log
   ```

7. Also enable the RIB Audit Logs for the corresponding TAFR adapter. The auditing feature logs the message as it passes though the RIB infrastructure. This helps the tracing of message content from publication to subscription.

   Note: To enable RIB Audit logs, see "RIB Logging" in the *Oracle Retail Integration Bus Operations Guide*.

8. Check the RIB audit logs for the particular message family's TAFR adapter and verify whether the message content is displayed correctly.

Prerequisites for RIB Localization

The tools used for localization extension of the RIB are separately documented. The primary tool is the Retail Functional Artifact Generator. The message (payload) structure and packaging is covered in the *Oracle Retail Functional Artifacts Guide.*

The following documents are referenced in this chapter and are required for the localization effort:

- O*racle Retail Functional Artifacts Guide*
- *Oracle Retail Functional Artifact Generator Guide*

9

RIB Localization - Business Objects

The deployment of Oracle Retail applications across the globe often drive requirements to localize the messages that flow among the Oracle Retail applications to support the region specific requirements (such as tax laws).

This chapter provides a detailed description of localization implementation at the Business Objects level. Localization hooks provided in base Business Objects and placeholders for adding implements are covered elsewhere in this guide.

As localization of Business Objects is performed as part of the localization process, where the end-point API's are changed or added to per region specific business requirements, localizations are meant to be performed by Oracle Retail or Partners. The localization points are not for use by customers. However, localization customization ispossible for use by customers.

Prerequisites for RIB Localization

The tools used for localization extension of the RIB are separately documented. The primary tool is the Retail Functional Artifact Generator. The message (payload) structure and packaging is covered in the *Oracle Retail Functional Artifacts Guide.*

The following documents are referenced in this chapter and are required for the localization effort:

- O*racle Retail Functional Artifacts Guide*
- *Oracle Retail Functional Artifact Generator Guide*

Business Objects Localization

Business Object localization is defined as extensions of the base XSDS made by the Oracle Retail or Partners to satisfy region specific business requirements.

Localization Hooks in Base Business Objects

The following table describes the hooks in base Business Objects.

Reference	Description	Example

Every Base BO XSD imports a LocOf<BO> XSD	A new placeholder XSD is created for each base Business Object. Every Business Object XSD imports the LocOf<BO> XSD.	`<xs:import namespace="http://www.oracle.com/retail/integration/custom/bo/ExtOfASNInDesc/v1" schemaLocation="../../../../custom/bo/ExtOfASNInDesc/v1/ExtOfASNInDesc.xsd">`
		`<retailDoc:annotation>`
		`<retailDoc:documentation>It's a referenced element. For detailed description, please refer referenced element doc.</retailDoc:documentation>`
		`</retailDoc:annotation>`
		`</xs:import>`
		`<xs:import namespace="http://www.oracle.com/retail/integration/base/bo/LocOfASNInDesc/v1" schemaLocation="../../../../base/bo/LocOfASNInDesc/v1/LocOfASNInDesc.xsd">`
		`<retailDoc:annotation>`
		`<retailDoc:documentation>It's a referenced element. For detailed description, please refer referenced element doc.</retailDoc:documentation>`
		`</retailDoc:annotation>`
		`</xs:import>`

Every Base BO XSD complex type has a localization point	Within each base Business Object, every complex type contain a reference to a placeholder for that complex type within the LocOf for that Business Object.	`<xs:element name="ASNInItem">`
		`<retailDoc:annotation>`
		`<retailDoc:documentation>Description is not available.</retailDoc:documentation>`
		`</retailDoc:annotation>`
		`<xs:complexType>`
		`<xs:sequence>`
		`...`
		`...`
		`<xs:element maxOccurs="1" minOccurs="0" ref="ExtOfASNInDesc:ExtOfASNInItem">`
		`<retailDoc:annotation>`
		`<retailDoc:documentation>Provide an extension hook to customize ASNInItem.</retailDoc:documentation>`
		`</retailDoc:annotation>`
		`</xs:element>`
		`<xs:element maxOccurs="1" minOccurs="0" ref="LocOfASNInDesc:LocOfASNInItem">`
		`<retailDoc:annotation>`
		`<retailDoc:documentation>Provide an extension hook to localize ASNInItem.</retailDoc:documentation>`
		`</retailDoc:annotation>`
		`</xs:element>`
		`</xs:sequence>`
		`</xs:complexType>`

Region Specific Placeholders

The following table describes region specific placeholders.

Reference	Description	Example
LocOfBO imports All Specific Locals	The LocOf XSD is point where all the specific localization XSDs are imported and referred to. There exists one for each region (e.g. Brazil, India, China, etc.). The ISO standard alpha-2 country code is used as the identifier.	<xs:schema elementFormDefault="qualified" targetNamespace="http://www.oracle.com/retail/integration/base/bo/LocOfASNInDesc/v1" version="1.0" xmlns="http://www.oracle.com/retail/integration/base/bo/LocOfASNInDesc/v1" xmlns:BrASNInDesc="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" xmlns:InASNInDesc="http://www.oracle.com/retail/integration/localization/bo/InASNInDesc/v1" xmlns:retailDoc="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"> <retailDoc:annotation> <retailDoc:documentation>This is root element of this document which contains name space definitions for the document elements.</retailDoc:documentation> </retailDoc:annotation> <xs:import namespace="http://www.oracle.com/retail/integration/localization/bo/InASNInDesc/v1" schemaLocation="../../../../localization/bo/InASNInDesc/v1/InASNInDesc.xsd"> <retailDoc:annotation> <retailDoc:documentation>It's a referenced element. For detailed description, please refer referenced element doc.</retailDoc:documentation> </retailDoc:annotation> </xs:import> <xs:import namespace="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" schemaLocation="../../../../localization/bo/BrASNInDesc/v1/BrASNInDesc.xsd"> <retailDoc:annotation> <retailDoc:documentation>It's a referenced element. For detailed description, please refer referenced element doc.</retailDoc:documentation> </retailDoc:annotation> </xs:import> </xs:schema>

Reference	Description	Example
Each of the specific localization XSDs will resolve the references with the actual localization specific elements.	For each of the complex types in the region specific XSD the place to perform the actual localizations. Localization team add implements here.	<xs:schema elementFormDefault="qualified" targetNamespace="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" version="1.0" xmlns="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" xmlns:retailDoc="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:import namespace="http://www.oracle.com/retail/integration/custom/bo/EOfBrASNInDesc/v1" schemaLocation="../../../../custom/bo/EOfBrASNInDesc/v1/EOfBrASNInDesc.xsd"> <retailDoc:annotation> <retailDoc:documentation>It's a referenced element.</retailDoc:documentation> </retailDoc:annotation> </xs:import> <xs:element name="BrASNInItem"> <xs:complexType> <xs:sequence> ... ... </xs:sequence> </xs:element> </xs:schema>

Localization Customization

The following table describes the hook for localization customization.

Reference	Description	Example
Each specific localization XSD complex type contain a customization hook	In each of the specific localizations, a hook back to the customization of that complex type is referenced	<xs:schema elementFormDefault="qualified" targetNamespace="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" version="1.0" xmlns="http://www.oracle.com/retail/integration/localization/bo/BrASNInDesc/v1" xmlns:retailDoc="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:import namespace="http://www.oracle.com/retail/integration/custom/EOfBrASNInDesc/v1" schemaLocation="../../../../custom/EOfBrASNInDesc/v1/EOfBrASNInDesc.xsd"> <retailDoc:annotation> <retailDoc:documentation>It's a referenced element.</retailDoc:documentation> </retailDoc:annotation> </xs:import> <xs:element name="BrASNInItem"> <xs:complexType> <xs:sequence> ... ... <xs:element maxOccurs="1" minOccurs="0" ref="EOfBrASNInDesc:EOfBrASNInItem"> <retailDoc:annotation> <retailDoc:documentation>Provide an extension hook to customize ASNInItem </retailDoc:documentation> </retailDoc:annotation> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:schema>

Adding Localization Fields

To add the localization specific fields in a placeholder, the first step is to determine the region specific XSD, which is follows naming convention below:

<CC><BO>.xsd (for exampleBrASNInDesc), where CC = 2 char ISO country code

Determine the complex type fields that must be added. Add the fields and generate payloads using the Artifact Generator tool.

Note: See the *Oracle Retail Functional Artifact Generator Guide* and the *Oracle Retail Functional Artifacts Guide.*

Adding Localization Customization Fields

Customization of localization Business Objects follows the same process required for base Business Objects customization.

Packaging

Retail Functional Artifact packaging is extended to provide the regional localization placeholders and the region specific XSD. The placeholders for localization customization also are provided.

Note: See the *Oracle Retail Functional Artifact Generator Guide* and the *Oracle Retail Functional Artifacts Guide.*

10

Integration with External Applications

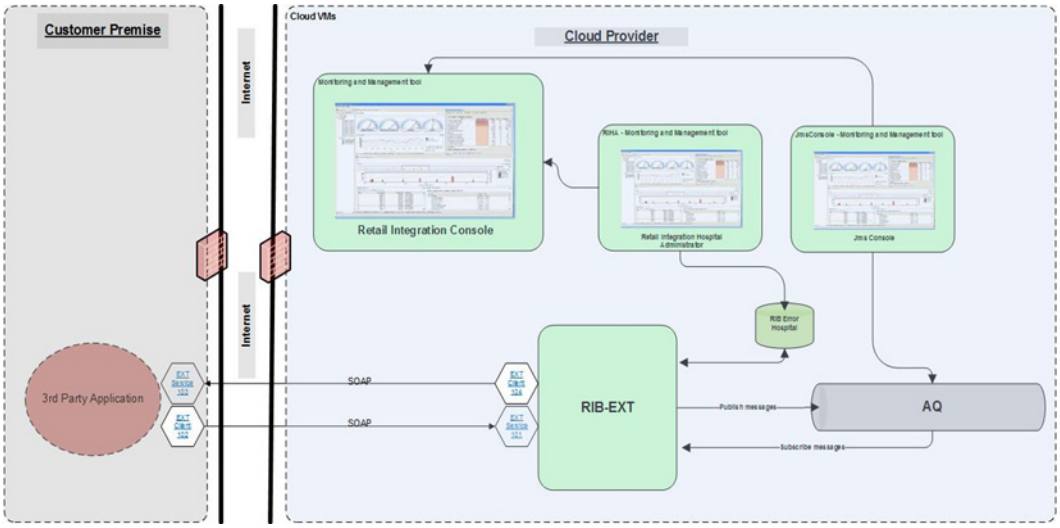
RIBforExt is the Oracle Retail Enterprise Integration component designed to address the connectivity requirements for 3rd Party integrations in a hybrid cloud topology where the RIB is deployed in the Retail Integration Cloud Services.

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: For more information on WDSL's, see the *Oracle Retail Integration Bus Integration Guide* and for information on plugable jar, see the Client Connector For Oracle Retail Integration Cloud Service 21.0.000 (Patch) available on My Oracle Support.

External Application as a Publisher (soap-app)

For external applications to publish to the RIB JMS on cloud, it needs to use a publishing webservice provided by rib-ext .The WSDL URL of publishing service is as follows:

http://<rib-ext-host>:<port>/ApplicationMessagePublishingServiceBean/ApplicationMessagePublishingService?wsdl

An external application can publish messages using the above webservice only when rib-ext is configured as a soap-app.

External Application as a Subscriber (soap-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 SOAP webservice that is made available as a pluggable jar.

Subscriber adapters in rib-ext makes a SOAP call to Injector service to send the message to the external application. The WSDL URL of injector service is as follows:

http://<external-app-host>:<port>/ApplicationMessageInjectorBean/InjectorService?wsdl

The following example describes the steps to configure an external application to publish and subscribe using RIB on cloud:

- Include rib-private-app-plugin-21.0.000.jar in to the external application deployable file for example, ext-app.ear/lib.
- In the rib-deployment-env-info.xml file, configure the EXT application to be of type "soap-app". Under <app-in-scope-for-integration>, change EXT from javaee-app to soap-app:

  <app id="ext" type="soap-app" />
- Replace the existing rib-app section for rib-ext with a copy of the rib-app section for rib-sim (an existing soap-app). Edit the properties so that they apply for rib-ext.

  For example:

```
<rib-app id="rib-ext" type="soap-app">
<deploy-in refid="rib-ext-wls1" />
<rib-admin-gui><message-flow id="901">
     <web-app-url>https://www.example.com:<port>/rib-ext-appserver-gui/index.jsp</web-app-url>
     <web-app-user-alias>rib-ext_rib-admin-gui_
user-name-alias</web-app-user-alias>
</rib-admin-gui>
<error-hospital-database>
<hosp-user-alias>rib-ext_error-hospital-database_
user-name-alias</hosp-user-alias>
</error-hospital-database>
<app-database-not-applicable />
<notifications>
```

```
<email>
<email-server-host>mail.example.com</email-server-host>
    <email-server-port>25</email-server-port>
    <from-address>admin@example.com</from-address>
    <to-address-list>admin@example.com</to-address-list>
    </email>
    <jmx />
    </notifications>
<app id="ext" type="soap-app">
<end-point>
<url>https://www.example.com:<port>/ApplicationMessageInjectorBean/InjectorServ
ice?WSDL</url>
    <ws-policy-name>policyA</ws-policy-name>
       <user-alias>rib-ext_ws_security_user-name-alias</user-alias>
    </end-point>
</app>
</rib-app>
```

- ws-policy-name should be configured with a value “policyA”.
- Make sure the rib-ext_ws_security_user-name-alias user is a member of the ext_integration_users group in the EXT WebLogic domain. Make sure the EXT services are up and running and can be called via the SOAP UI using the credentials that will be entered during RIB compilation.
- Compile and deploy RIB.

External Application as a Publisher (rest-app)

For external applications to publish to the RIB JMS on cloud, it needs to use a publishing webservice provided by rib-ext. The end point of publishing service is as follows:

API	Rest End Point
Application WADL	http://<rib-ext-host>:<port>/rib-<app>-services-web/resources/application.wadl
Ping resource	http://<rib-ext-host>:<port>/rib-<app>-services-web/resources/publisher/ping
Publish resource	http://<rib-ext-host>:<port>/rib-<app>-services-web/resources/publisher/publish

An external application can publish messages using the above webservice only when rib-ext is configured as a rest-app.

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.

Subscriber adapters in rib-ext makes a ReST call to Injector service to send the message to the external application. The End Point of injector service is as follows:

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

The following example describes the steps to configure an external application to publish and subscribe using RIB on cloud:

- Include rib-injector-services-web-21.0.000.war in to the external application deployable file for example, ext-app.ear/lib.
- In the rib-deployment-env-info.xml file, configure the EXT application to be of type "rest-app". Under <app-in-scope-for-integration>, change EXT from javaee-app to rest-app:
- Replace the existing rib-app section for rib-ext with a copy of the rib-app section for rib-rce (an existing rest-app). Edit the properties so that they apply for rib-ext.

  For example:

```
<rib-app id="rib-ext" type="rest-app">
  <deploy-in refid="rib-ext-wls1" />
  <rib-admin-gui><message-flow id="901">
    <web-app-url>https://www.example.com:<port>/rib-ext-appserver-
gui/index.jsp</web-app-url>
    <web-app-user-alias>rib-ext_rib-admin-gui_
user-name-alias</web-app-user-alias>
  </rib-admin-gui>
  <error-hospital-database>

    <hosp-user-alias>rib-ext_error-hospital-database_
user-name-alias</hosp-user-alias>
  </error-hospital-database>
  <app-database-not-applicable />
  <notifications>
    <email>
      <email-server-host>mail.example.com</email-server-host>
      <email-server-port>25</email-server-port>
      <from-address>admin@example.com</from-address>
      <to-address-list>admin@example.com</to-address-list>
    </email>
    <jmx />
  </notifications>
  <app id="ext" type="rest-app">
    <end-point>
      <url>https://www.example.com:<port>/rib-injector-services-
web/resources/injector/inject" </url>
      <ws-policy-name>policyA</ws-policy-name>
      <user-alias>rib-ext_ws_security_user-name-alias</user-alias>
    </end-point>
  </app>
</rib-app>
```

- ws-policy-name should be configured with a value "policyA".
- Make sure the rib-ext_ws_security_user-name-alias user is a member of the ext_integration_users group in the EXT WebLogic domain. Make sure the EXT services are up and running and can be called via the SOAP UI using the credentials that will be entered during RIB compilation.
- Compile and deploy RIB.

How to implement ReST Client to Call the Publisher Service

In order to publish messages to RIB via ReST service, a standard JAX-RS client API can be used. We provide a few helper libraries and payload jars (RBO contracts) to be used

on the classpath of a client application. In other words, these libraries should be packaged as part of the application war/ear file.

Required Libraries are as below.

- application-message-publishing-service-consumer-21.0.000.jar
- commons-logging-1.2.jar
- rib-public-api-21.0.000.jar
- retail-private-int-common-util-21.0.000.jar
- retail-public-payload-java-beans-21.0.000.jar
- retail-public-payload-java-beans-base-21.0.000.jar
- retail-rest-service-common-util-21.0.000.jar
- rib-private-common-21.0.000.jar

API Info	URL
Application WADL	http://<examplehost>:<port>/rib-<app>-services-web/resources/application.wadl
Ping resource	http://<examplehost>:<port>/rib-<app>-services-web/resources/publisher/ping
Publish resource	http://<examplehost>:<port>/rib-<app>-services-web/resources/publisher/publish

See "Sample Data from Integration Monitoring Service" for sample code.

How to implement Injector Service (CONSUME messages from RIB) using ReST

Here is the Rest service contract detail:

1. Keep the path as Injector/inject.

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

2. 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).

```
<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" minOccurs="0"/>
      <xs:element ref="ApplicationMessageRoutingInfo" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="payloadXml" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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.

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.*

A

External LDAP Configuration

WebLogic ships with a default internal Light-weight Directory Access Protocol (LDAP) authentication provider. In an environment where a couple of domains exist, an administrator can set up users and groups in an internal LDAP provider and use these parameters during login and authentication. Alternatively, in an environment that contains multiple domains, managing/maintaining users and groups can be a difficult task. Oracle recommends that you use a centralized LDAP server to manage/maintain the users and groups.

This chapter describes the steps you should take to configure the Oracle Internet Directory (OID) and the Active Directory (AD) LDAP based authentication provider in WebLogic.

Introducing the Oracle Internet Directory (OID)

An online directory is a specialized database that stores and retrieves collections of information about objects. The information can represent any resources that require management, for example:

- Employee names, titles, and security credentials
- Information about partners
- Information about shared resources such as conference rooms and printers

The information in the directory is available to different clients, such as single sign-on solutions, e-mail clients, and database applications. Clients communicate with a directory server by means of the LDAP. The Oracle Internet Directory is an LDAP directory that uses an Oracle database for storage.

Introducing the Microsoft Active Directory (AD)

An Active Directory (AD) is a directory service implemented by Microsoft for Windows domain networks. It is included in most Windows Server operating systems.

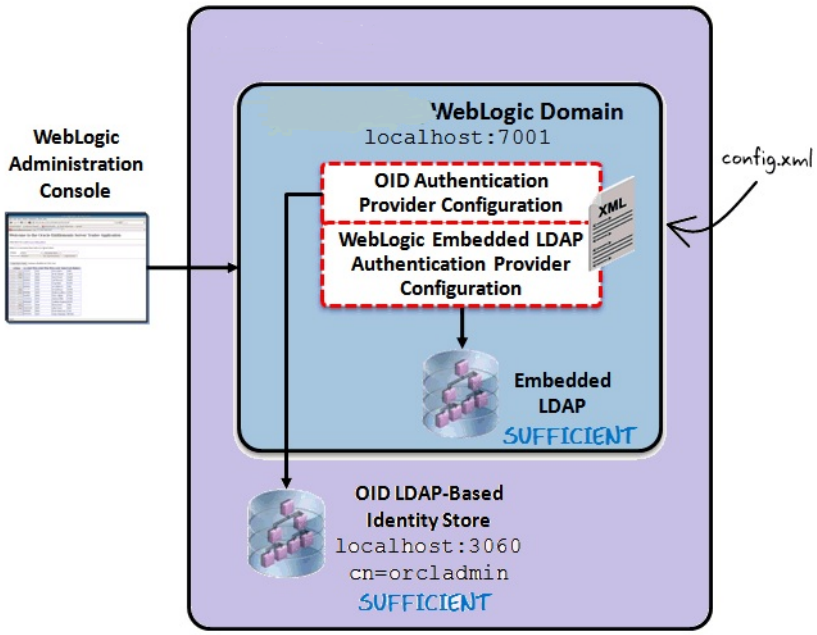
Active Directory is a special-purpose database — it is not a registry replacement. The directory is designed to handle a large number of read and search operations and a significantly smaller number of changes and updates. Active Directory data is hierarchical, replicated, and extensible. Because it is replicated, you do not want to store dynamic data, such as corporate stock prices or CPU performance.

In Windows 2000, Active Directory has three partitions. These are also known as naming contexts: do-main, schema, and configuration. The domain partition contains users, groups, contacts, computers, organizational units, and many other object types. Because Active Directory is extensible, you can also add your own classes and/or

attributes. The schema partition contains classes and attributes definitions. The configuration partition includes configuration data for services, partitions, and sites.

Architecture Overview

The architecture diagram describes the configuration of an OID and AD LDAP-based authentication provider used by applications deployed in an WebLogic server environment.



The diagram displays a sample environment and consists of the following:

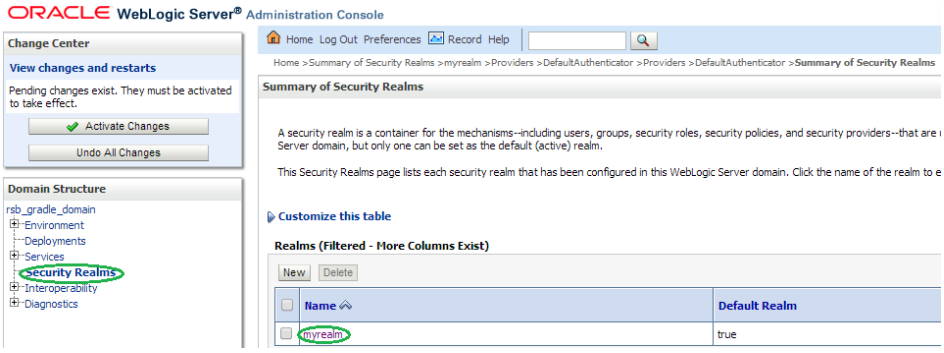
- The WebLogic Server running on port 7001
- The WebLogic Administration Console used to configure authentication providers
- The WebLogic Embedded LDAP server with a control flag setting of SUFFICIENT
- An OID LDAP-based identity store running on port 3060 with a control flag setting of SUFFI-CIENT
- The WebLogic config.xml that stores the authentication provider configuration

By default, the WebLogic server uses a security realm with the name “myrealm” that uses an embedded LDAP server (two default users WebLogic & OracleSystemUser) that acts as data store for Authentication, Authorization, Credential Mapping and Role Mapping Provider.

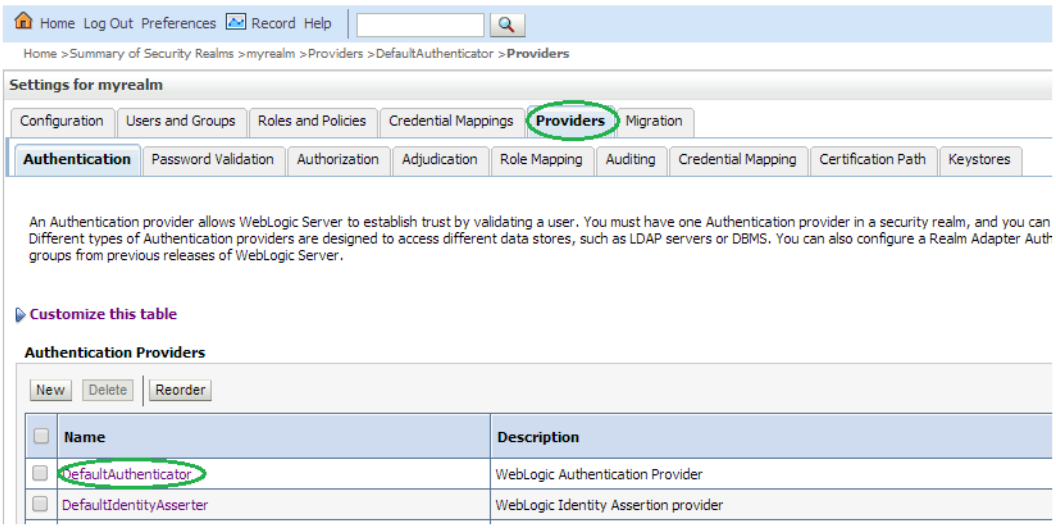
Configuring the Oracle Internet Directory (OID) as an Authentication Provider in WebLogic

To configure the OID as an authentication provider in WebLogic, take the following steps:

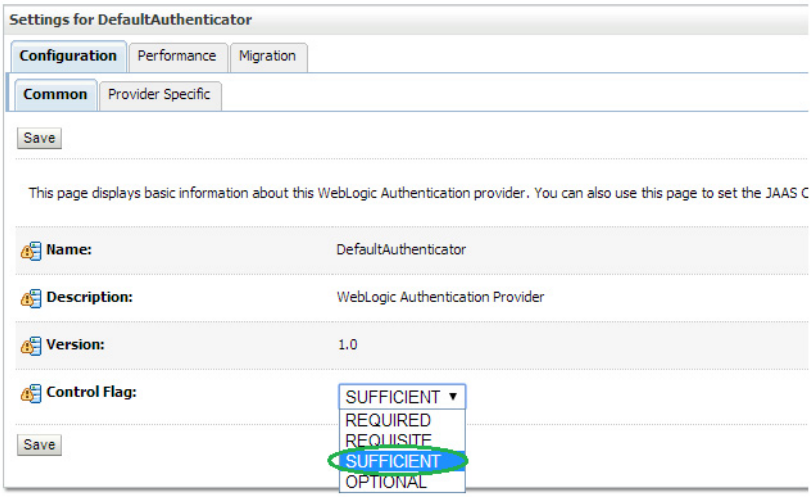
1. Login to **WebLogic Console -> Security Realm -> myrealm**.



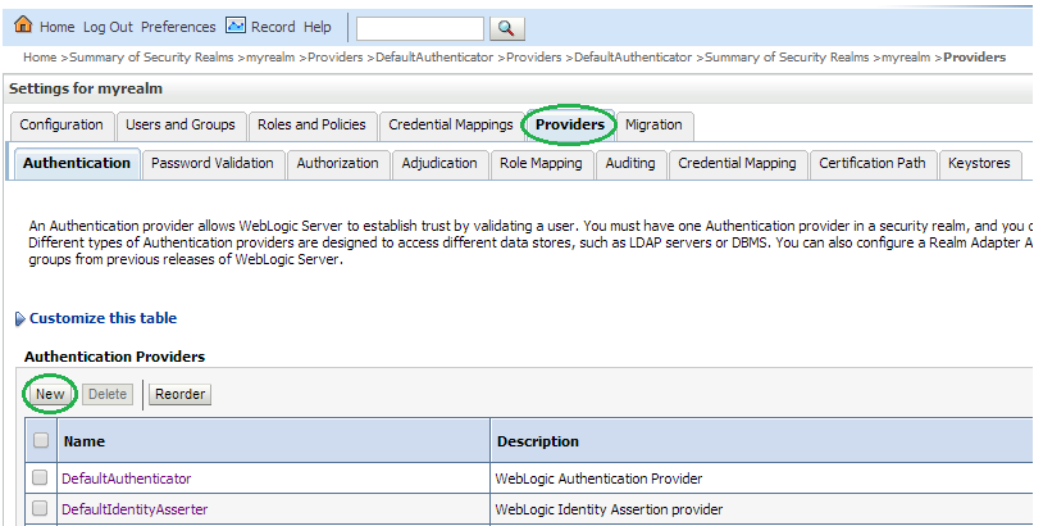
2. Select tab **Providers -> Authentication -> Default Provider (DefaultAuthenticator)**.



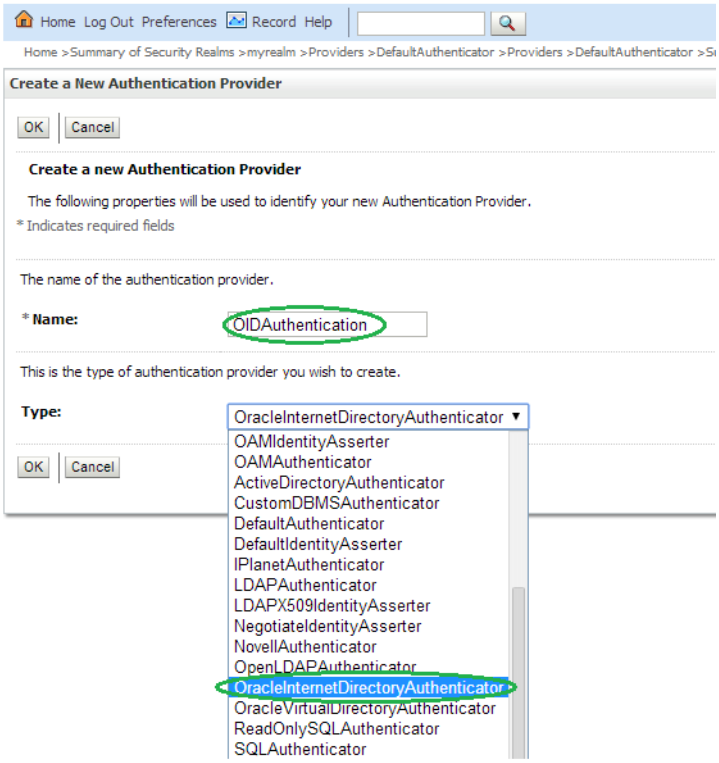
3. Change the **Control Flag** (JAAS Flag) parameter from REQUIRED to SUFFICIENT and click **Save**.



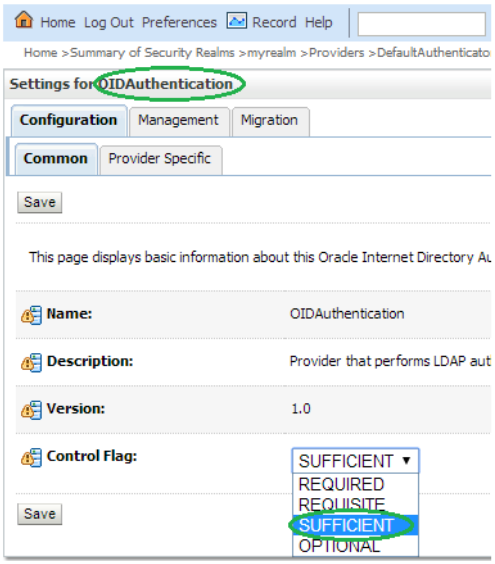
4. Click **New** to add a new Authentication Provider.



5. Enter OIDAuthentication as the **Name** of the new provider. Select OracleInternetDirectoryAuthenticator as **Type** and then click **OK**.



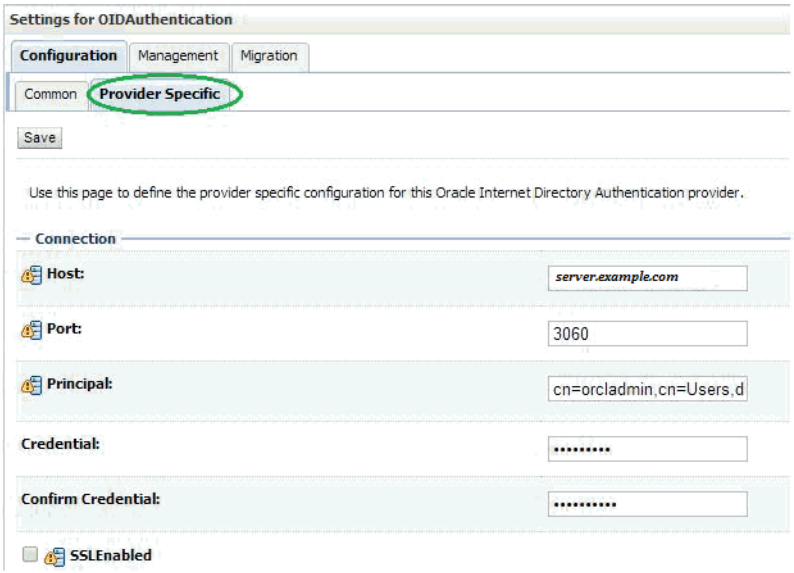
6. Change the **Control Flag** to SUFFICIENT for the OIDAuthentication Provider added and click **Save**.



7. Select the **Provider Specific** tab and enter your OID server details.

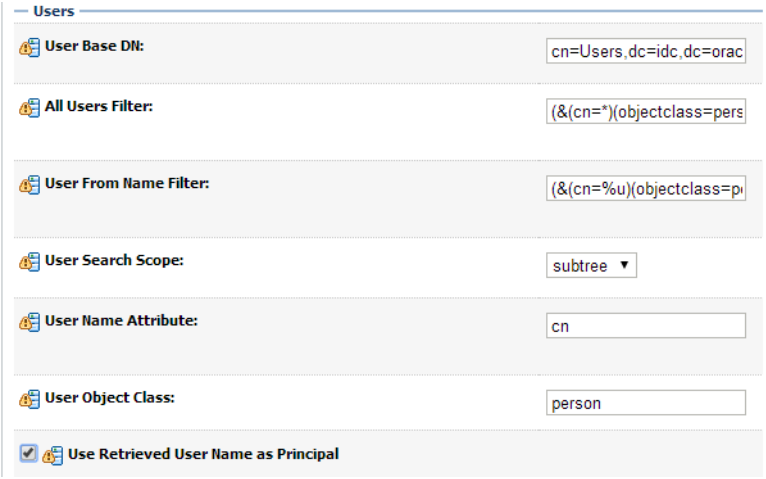
   a. The first section contains the Connection settings for the OID server. Use the appropriate values based on where the OID is hosted and the credentials:

Name	Value	Purpose
Host:	server.example.com	The OID host name
Port:	3060	The standard OID listening port
Principal:	cn=orcladmin,cn=Users,dc=idc,dc=oracle,dc=com	The LDAP user that logs into OID on behalf of your authentication provider
Credentials:		Password for the principal user
Confirm Credentials:		Confirmation of the password
SSL Enabled:	Unchecked	Enables or disables SSL connectivity



b. The second section contains the Users settings for the OID provider. Use appropriate values:

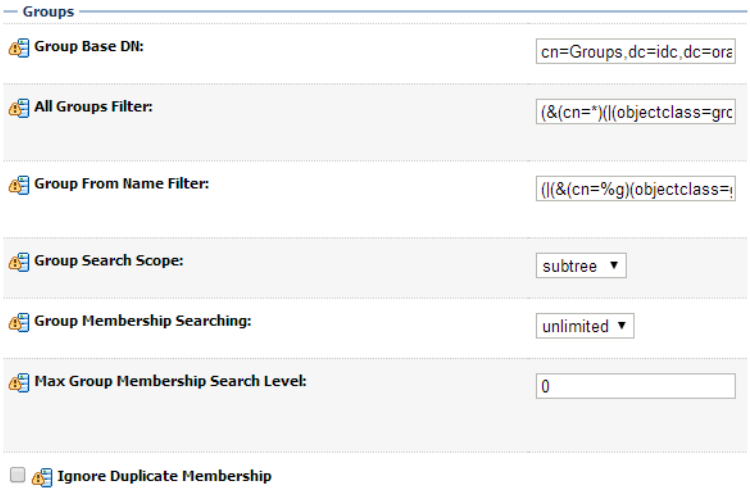
Name	Value	Purpose
User Base DN:	cn=Users,dc=idc,dc=oracle,dc=com	The root (base DN) of the LDAP tree where searches are performed for user data
All Users Filter:	(&(cn=*)(objectclass=person)) -- Leave as default	The LDAP search filter that is used to show all the users below the User Base DN
User From Name Filter:	(&(cn=%u)(objectclass=person)) -- Leave as default	The LDAP search filter used to find the LDAP user by name
User Search Scope:	Leave as default	Specifies how deep in the LDAP tree to search for users
User Name Attribute:	Leave as default	The attribute of the LDAP user that specifies the user name
User Object Class:	Leave as default	The LDAP object class that stores users
Use Retrieved User Name as Principal:	Checked	Specifies if the user name retrieved from the LDAP directory will be used as the Principal in the Subject



c. The third section contains the Groups settings for the OID provider. Use appropriate values:

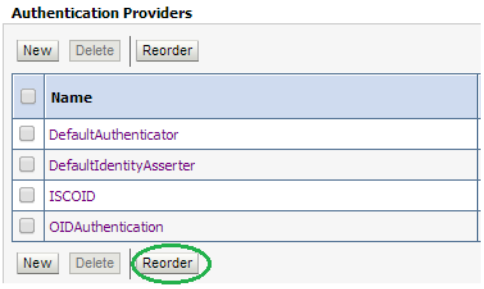
Name	Value	Purpose
Group Base DN:	cn=Groups,dc=idc,dc=oracle,dc=com	The root (base DN) of the LDAP tree where searches are per-formed for group data
All Groups Filter:	(&(cn=*)(|(objectclass=groupofUniqueNames)(objectclass=orcldynamicgroup))) -- Leave as default	The LDAP search filter that is used to show all the groups below the Group Base DN
Group From Name Filter:	(|(&(cn=%g)(objectclass=groupofUniqueNames))(&(cn=%g)(objectclass=orcldynamicgroup))) -- Leave as default	The LDAP search filter used to find the LDAP group by name

Name	Value	Purpose
Group Search Scope:	Leave as default	Specifies how deep in the LDAP tree to search for groups
Group Member-ship Searching:	Leave as default	Specifies whether group searches into nested groups are limited or unlimited
Max Group Member-ship Search Level:	Leave as default	Specifies how many levels of group membership can be searched. This setting is only valid if GroupMembershipSearching is set to limited
Ignore Duplicate Membership:	Unchecked	Determines whether duplicates members are ignored when adding groups.

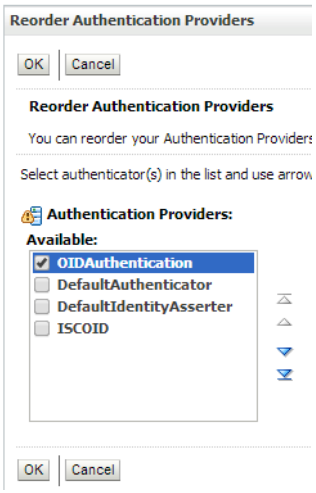


d. Click **Save**.

8. Click **Reorder** to change the order of your configured authentication providers. In order to ensure that the new OID authenticator is recognized as authentication provider, you must reorder your list of authentication providers so that the OID authentication provider is first in the list.



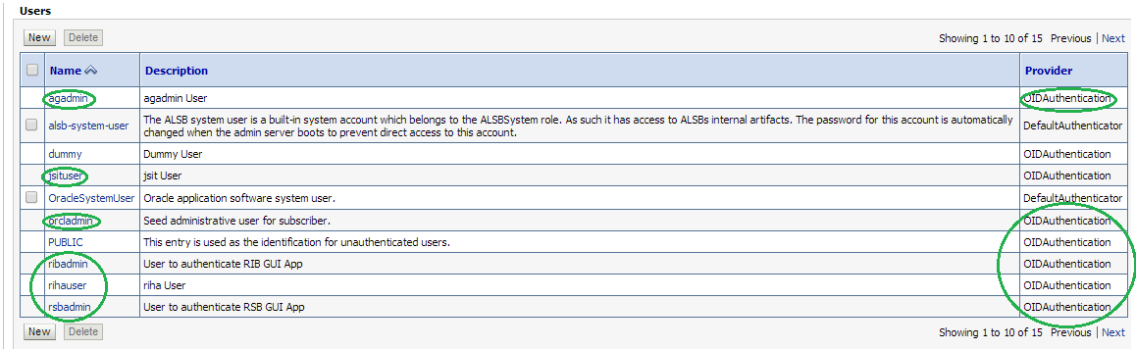
9. Select the OIDAuthentication and use the arrows on the right to move it into the first position. Click **OK**.



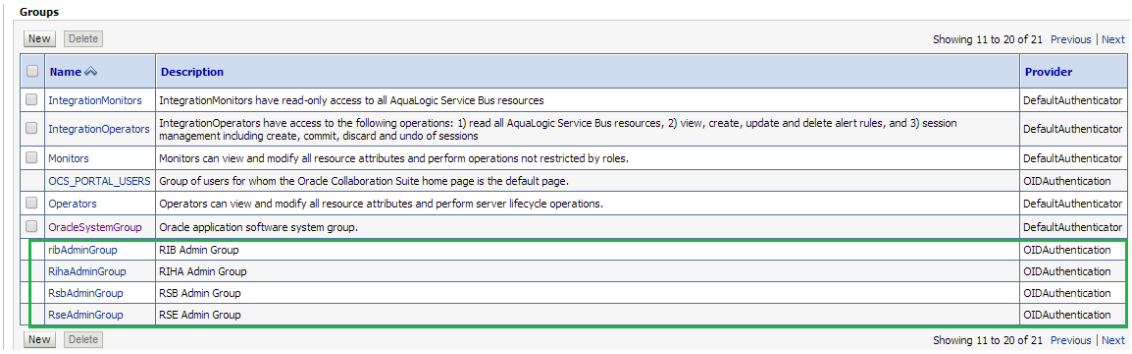
Verifying the Oracle Internet Directory (OID) Configuration

To verify the OID configuration, take the following steps:

1. Restart the WebLogic Server for your changes to take effect.
2. Using the WebLogic Administration Console, select **Security Realms > myrealm > Users and Groups** tab. The Users sub-tab should be selected by default. The circled users are created in OID and can verify the Provider – OIDAuthentication provider.



3. Click the **Groups** tab to see the list of groups the server can see. The highlighted groups are created in OID and can verify the Provider – OIDAuthentication provider.



Using LDIF Scripts to Configure Users and Groups for OID

LDIF scripts can be used to import users and groups into OID. Two sample scripts are supplied below. The scripts contain users and groups for multiple Oracle Retail integration products. You must review and edit the scripts to match your deployment topology and in-scope applications.

Integration-oid-create-groups.ldif

dn: cn=agAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: agAdminGroup

description: ArtifactGenerator Administrator is a group of individuals who can generate artifacts used in the integration products like OracleObject, JavaBeans.

displayname: ArtifactGenerator Administrator

#businessCategory: TBD

uniquemember: cn=agadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=JmsConsoleAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: JmsConsoleAdminGroup

description: JMS Console Administrator is a group of individuals who can perform various administrator task on jmsconsole like publishing message on topic, browsing messages on topic.

displayname: JMS Console Administrator

#businessCategory: TBD

uniquemember: cn=jmsconsoleadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=ribAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: ribAdminGroup

description: RIB Administrator is a group of individuals who can administrator rib-admin-gui. View the adapters state, start/stop adapters, view logs,set the log levels for adapters.

displayname: RIB Administrator

#businessCategory: TBD

uniquemember: cn=ribrmsadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribsimadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribrwmsadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribaipadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=riblgfadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribocdadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribrobadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribtafradmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribrfmadmin,cn=users,dc=us,dc=oracle,dc=com

uniquemember: cn=ribrpmadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=IntegrationGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: IntegrationGroup

description: IntegrationGroup is a group of individuals who can invoke rib interface api inject and publish.

displayname: Integration Group

#businessCategory: TBD

uniquemember: cn=integrationuser,cn=users,dc=us,dc=oracle,dc=com

dn: cn=RihaAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: RihaAdminGroup

description: Riha Admin Group is a group of individuals who can administer rib hospital. Can flush the messages stuck in rib error hospital, can retry the messages,view the messages in error hospital and can edit.

displayname: Riha Administrator

#businessCategory: TBD

uniquemember: cn=rihaadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=RicAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: RicAdminGroup

description: Ric Admin Group is a group of individuals who can administer rib runtime statistics , rsb runtime statistics.

displayname: Ric Administrator

#businessCategory: TBD

uniquemember: cn=ricadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=rseAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: rseAdminGroup

description: Rse Admin Group is a group of individuals who can generate webservice provider , consumer.

displayname: RSE Administrator

#businessCategory: TBD

uniquemember: cn=rseadmin,cn=users,dc=us,dc=oracle,dc=com

dn: cn=RfiAdminGroup,cn=groups,dc=us,dc=oracle,dc=com

objectclass: groupOfUniqueNames

objectclass: orclGroup

objectclass: top

cn: RfiAdminGroup

description: RFI Admin

displayname: RFI Administrator

#businessCategory: TBD

uniquemember: cn=rfiadmin,cn=users,dc=us,dc=oracle,dc=com

Integration-oid-create-users.ldif

dn: cn=agadmin, cn=Users,dc=us,dc=oracle,dc=com

description: A user for the 'AG Admin' role.

objectclass: inetOrgPerson

objectclass: organizationalPerson

objectclass: person

objectclass: top

objectclass: orcluser

objectclass: orcluserV2

objectclass: orclIDXPerson

cn: agadmin

orclsamaccountname: agadmin
sn: agadmin
uid: agadmin
givenname: agadmin
displayname: agadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: agadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=jmsconsoleadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'JMS Console Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: jmsconsoleadmin
orclsamaccountname: jmsconsoleadmin
sn: jmsconsoleadmin
uid: jmsconsoleadmin
givenname: jmsconsoleadmin
displayname: jmsconsoleadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:

telephoneNumber:
facsimileTelephoneNumber:
mail: jmsconsoleadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribrmsadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribrmsadmin
orclsamaccountname: ribrmsadmin
sn: ribrmsadmin
uid: ribrmsadmin
givenname: ribrmsadmin
displayname: ribrmsadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribrmsadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribrpmadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribrpmadmin
orclsamaccountname: ribrpmadmin
sn: ribrpmadmin
uid: ribrpmadmin
givenname: ribrpmadmin
displayname: ribrpmadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribrpmadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribrwmsadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson

cn: ribrwmsadmin
orclsamaccountname: ribrwmsadmin
sn: ribrwmsadmin
uid: ribrwmsadmin
givenname: ribrwmsadmin
displayname: ribrwmsadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribrwmsadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=riblgfadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: riblgfadmin
orclsamaccountname: riblgfadmin
sn: riblgfadmin
uid: riblgfadmin
givenname: riblgfadmin
displayname: riblgfadmin
userpassword: <update your password here>
employeeNumber:
middleName:

orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: riblgfadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribrobadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribrobadmin
orclsamaccountname: ribrobadmin
sn: ribrobadmin
uid: ribrobadmin
givenname: ribrobadmin
displayname: ribrobadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribrobadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribocdsadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribocdsadmin
orclsamaccountname: ribocdsadmin
sn: ribocdsadmin
uid: ribocdsadmin
givenname: ribocdsadmin
displayname: ribocdsadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribocdsadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribtafradmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2

objectclass: orclIDXPerson
cn: ributafradmin
orclsamaccountname: ributafradmin
sn: ributafradmin
uid: ributafradmin
givenname: ributafradmin
displayname: ributafradmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ributafradmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribaipadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribaipadmin
orclsamaccountname: ribaipadmin
sn: ribaipadmin
uid: ribaipadmin
givenname: ribaipadmin
displayname: ribaipadmin
userpassword: <update your password here>
employeeNumber:

middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribaipadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=ribsimadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribsimadmin
orclsamaccountname: ribsimadmin
sn: ribsimadmin
uid: ribsimadmin
givenname: ribsimadmin
displayname: ribsimadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribsimadmin@example.com
postalAddress:
street:
postalCode:
title:

employeeType:

dn: cn=ribrfmadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIB Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: ribrfmadmin
orclsamaccountname: ribrfmadmin
sn: ribrfmadmin
uid: ribrfmadmin
givenname: ribrfmadmin
displayname: ribrfmadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: ribrfmadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=integrationuser, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'Integration' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser

objectclass: orcluserV2
objectclass: orclIDXPerson
cn: integrationuser
orclsamaccountname: integrationuser
sn: integrationuser
uid: integrationuser
givenname: integrationuser
displayname: integrationuser
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: integrationuser@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=rihaadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RIHA Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: rihaadmin
orclsamaccountname: rihaadmin
sn: rihaadmin
uid: rihaadmin
givenname: rihaadmin
displayname: rihaadmin
userpassword: <update your password here>

employeeNumber:

middleName:

orclHireDate:

telephoneNumber:

facsimileTelephoneNumber:

mail: rihaadmin@example.com

postalAddress:

street:

postalCode:

title:

employeeType:

dn: cn=ricadmin, cn=Users,dc=us,dc=oracle,dc=com

description: A user for the 'RIC Admin' role.

objectclass: inetOrgPerson

objectclass: organizationalPerson

objectclass: person

objectclass: top

objectclass: orcluser

objectclass: orcluserV2

objectclass: orclIDXPerson

cn: ricadmin

orclsamaccountname: ricadmin

sn: ricadmin

uid: ricadmin

givenname: ricadmin

displayname: ricadmin

userpassword: <update your password here>

employeeNumber:

middleName:

orclHireDate:

telephoneNumber:

facsimileTelephoneNumber:

mail: ricadmin@example.com

postalAddress:

street:

postalCode:

title:

employeeType:

dn: cn=rseadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RSE Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser
objectclass: orcluserV2
objectclass: orclIDXPerson
cn: rseadmin
orclsamaccountname: rseadmin
sn: rseadmin
uid: rseadmin
givenname: rseadmin
displayname: rseadmin
userpassword: <update your password here>
employeeNumber:
middleName:
orclHireDate:
telephoneNumber:
facsimileTelephoneNumber:
mail: rseadmin@example.com
postalAddress:
street:
postalCode:
title:
employeeType:

dn: cn=rfiadmin, cn=Users,dc=us,dc=oracle,dc=com
description: A user for the 'RFI Admin' role.
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: orcluser

objectclass: orcluserV2

objectclass: orclIDXPerson

cn: rfiadmin

orclsamaccountname: rfiadmin

sn: rfiadmin

uid: rfiadmin

givenname: rfiadmin

displayname: rfiadmin

userpassword: <update your password here>

employeeNumber:

middleName:

orclHireDate:

telephoneNumber:

facsimileTelephoneNumber:

mail: rfiadmin@example.com

postalAddress:

street:

postalCode:

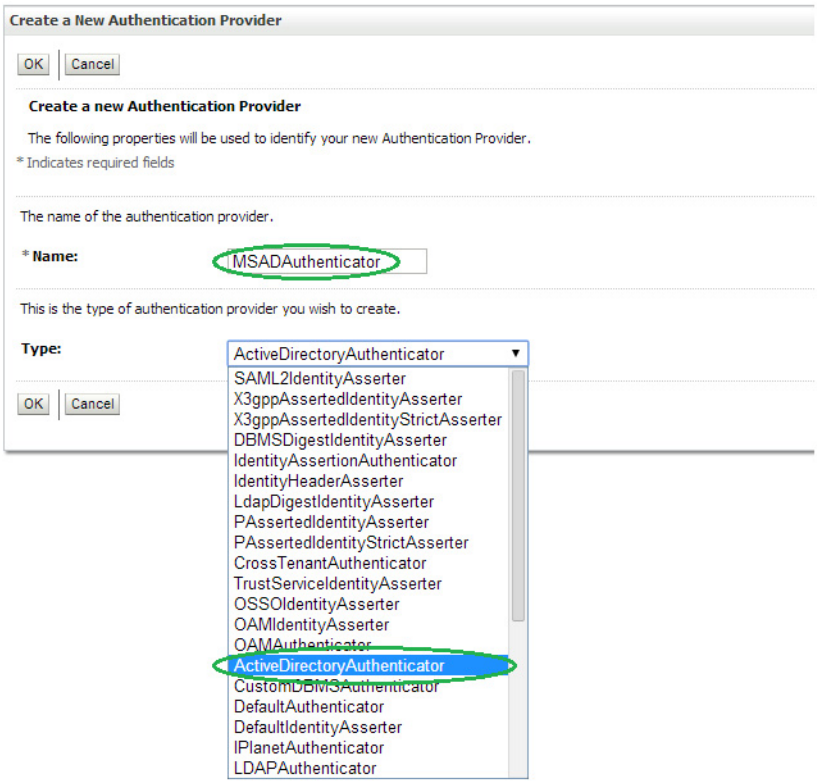
title:

employeeType:

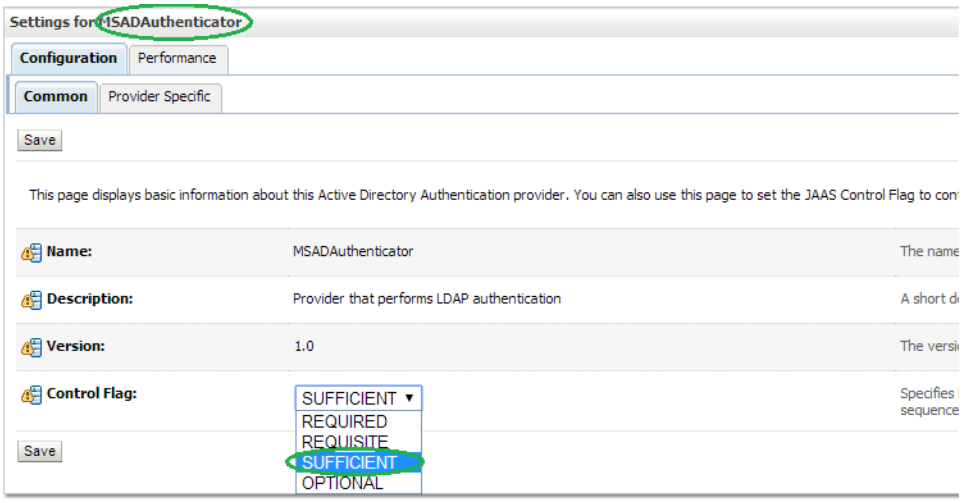
Configuring Active Directory (AD) as an Authentication Provider in WebLogic

To configure the AD as an authentication provider in WebLogic, take the following steps:

1. Login to **WebLogic Console -> Security Realm -> myrealm**.
2. Select tab **Providers -> Authentication -> Default Provider (DefaultAuthenticator)**.
3. Change the **Control Flag** (JAAS Flag) from REQUIRED to SUFFICIENT and click **Save**.
4. Click **New** to add a new Authentication Provider.
5. Enter MSADAuthenticator as the **Name**. Select ActiveDirectoryAuthenticator as the **Type** and click **OK**.



6. Change the **Control Flag** to SUFFICIENT for the MSADAuthenticator Provider added and click **Save**.

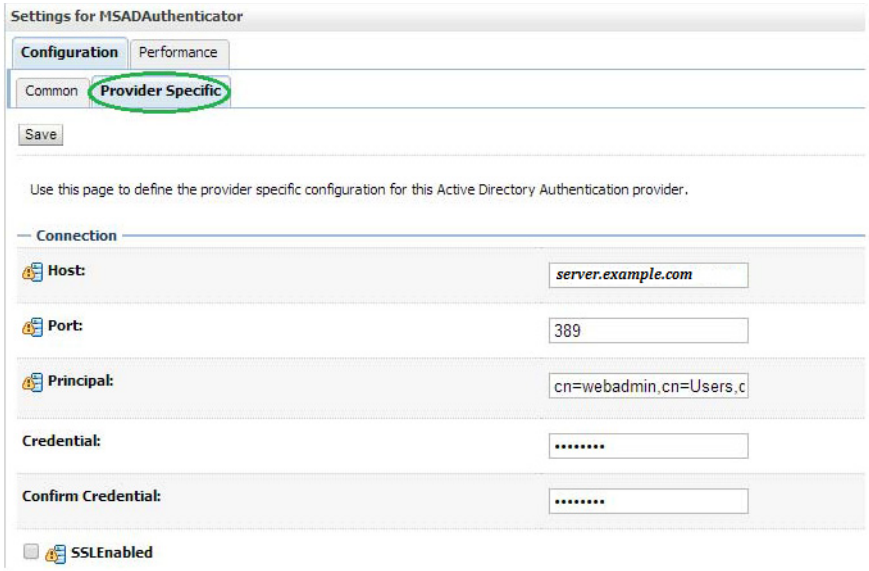


7. Select **Provider Specific** tab and enter the Active Directory (AD) server details.

   a. The first section contains the Connection settings for the AD server. Use appropriate values based on where AD is hosted and the credentials:

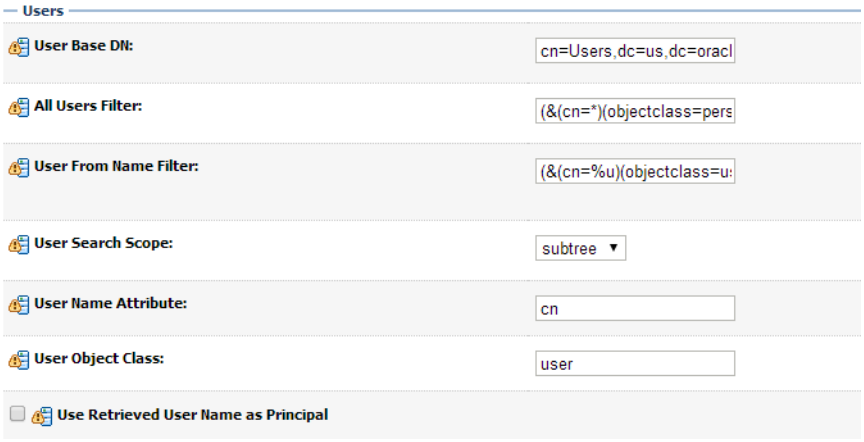
Name	Value	Purpose
Host:	server.example.com	The AD host name
Port:	389	The standard AD listening port

Name	Value	Purpose
Principal:	cn=webadmin,cn=Users, dc=us,dc=oracle,dc=com	The LDAP user that logs into AD on behalf of your authentication provider
Credentials:		Password for the principal user
Confirm Credentials:		Confirmation of the password
SSL Enabled:	Unchecked	Enables or disables SSL connectivity



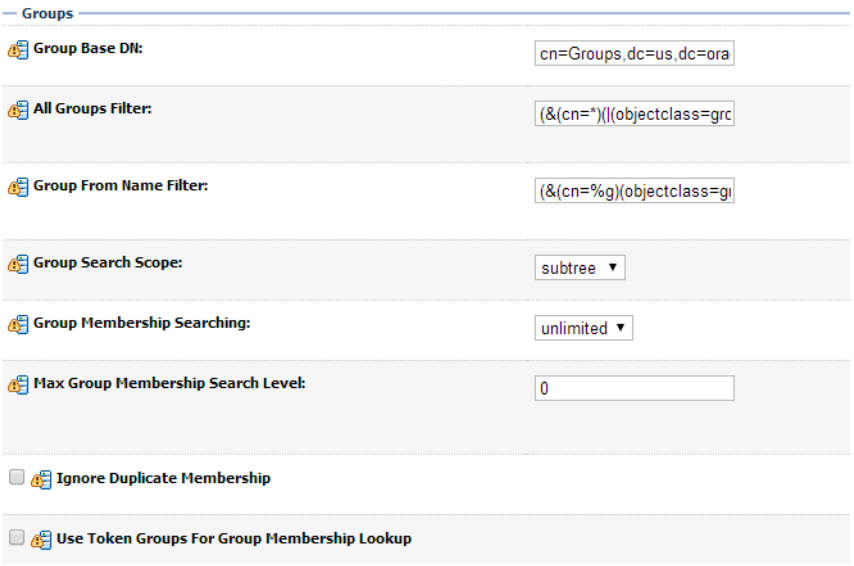
b. The second section contains the Users settings for the AD provider. Use appropriate values:

Name	Value	Purpose
User Base DN:	cn=Users,dc=us,dc=oracle,dc=com	The root (base DN) of the LDAP tree where searches are performed for user data
All Users Filter:	(&(cn=*)(objectclass=person))	The LDAP search filter that is used to show all the users below the User Base DN
User From Name Filter:	(&(cn=%u)(objectclass=user))	The LDAP search filter used to find the LDAP user by name
User Search Scope:	Leave as default	Specifies how deep in the LDAP tree to search for users
User Name Attribute:	Leave as default	The attribute of the LDAP user that specifies the user name
User Object Class:	Leave as default	The LDAP object class that stores users
Use Retrieved User Name as Principal:	Unchecked	Specifies if the user name retrieved from the LDAP directory will be used as the Principal in the Subject



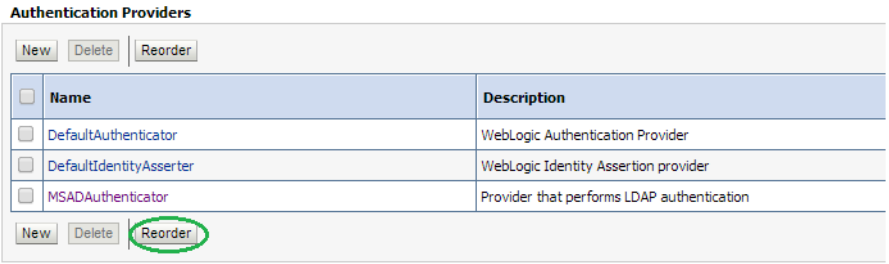
c. The third section contains the Groups settings for the AD provider. Use appropriate values:

Name	Value	Purpose
Group Base DN:	cn=Groups,dc=us,dc=oracle,dc=com	The root (base DN) of the LDAP tree where searches are performed for group data
All Groups Filter:	(&(cn=*)(|(objectclass=group)))	The LDAP search filter that is used to show all the groups below the Group Base DN
Group From Name Filter:	(&(cn=%g)(objectclass=group))	The LDAP search filter used to find the LDAP group by name
Group Search Scope:	Leave as default	Specifies how deep in the LDAP tree to search for groups
Group Member-ship Searching:	Leave as default	Specifies whether group searches into nested groups are limited or unlimited
Max Group Membership Search Level:	Leave as default	Specifies how many levels of group membership can be searched. This setting is only valid if GroupMembershipSearching is set to limited
Ignore Duplicate Membership:	Unchecked	Determines whether duplicates members are ignored when adding groups.

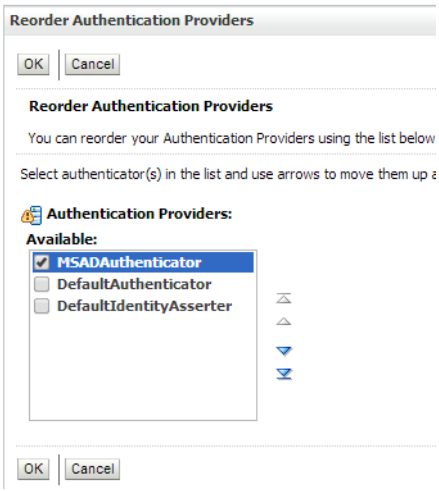


d. Click **Save**.

8. Click **Reorder** to change the order of your configured authentication providers. In order to ensure that MSAD authenticator is recognized as authentication provider, you must reorder your list of authentication providers so that the MSAD authentication provider is first in the list.



9. Select the MSADAuthenticator and use the arrows on the right to move it into the first position. Click **OK**.



Verifying the Active Directory (AD) Configuration

To verify the AD configuration, take the following steps:

1. Restart the WebLogic Server for your changes to take effect.
2. Using the WebLogic Administration Console, select S**ecurity Realms > myrealm > Users and Groups** tab. The Users sub-tab should be selected by default. The circled users are created in AD and can verify the Provider – MSADAuthenticator provider.

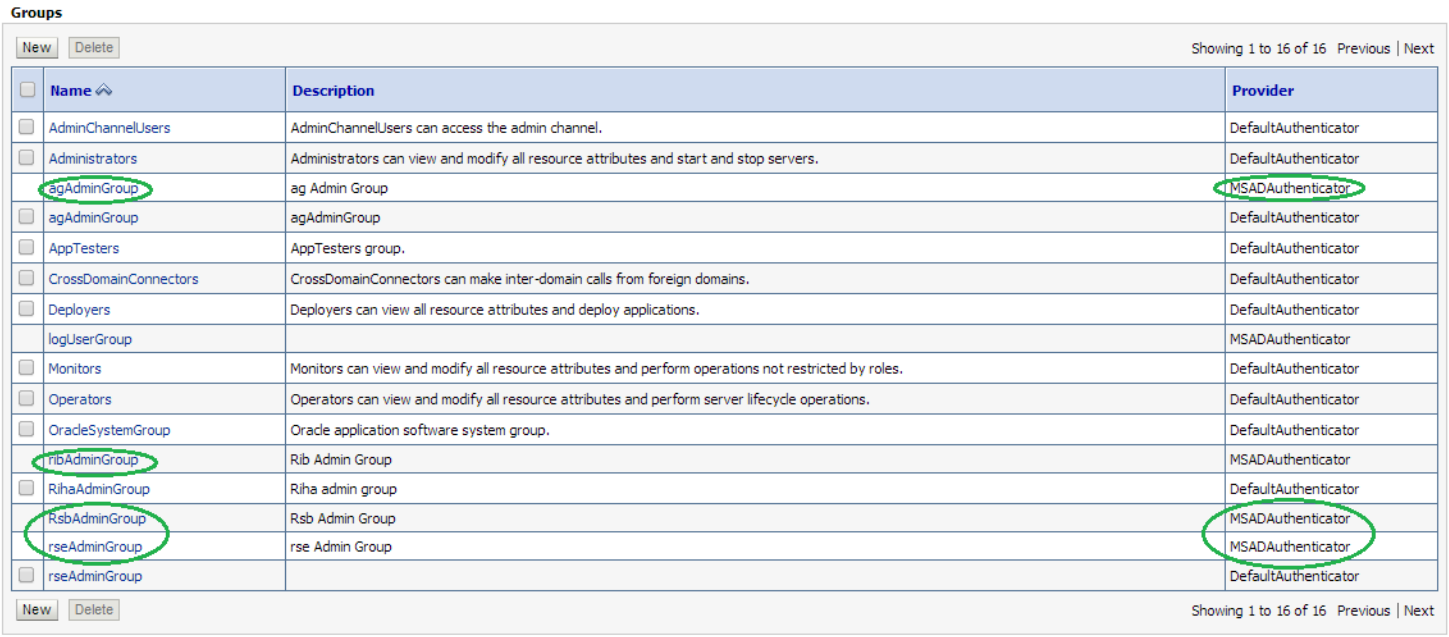
Users

New | Delete | Showing 1 to 18 of 18 Previous | Next

	Name	Description	Provider
	Administrator	Built-in account for administering the computer/domain	MSADAuthenticator
	agadmin	ag admin	MSADAuthenticator
	agadmin	agadmin	DefaultAuthenticator
	devsrvsppt	Oracle Sys Admin Account	MSADAuthenticator
	Guest	Built-in account for guest access to the computer/domain	MSADAuthenticator
	jsituser	jsit user	MSADAuthenticator
	krbtgt	Key Distribution Center Service Account	MSADAuthenticator
	logUser		MSADAuthenticator
	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
	ribadmin	rib admin	MSADAuthenticator
	rihauser	riha user	MSADAuthenticator
	rmsuser		DefaultAuthenticator
	rsbadmin	rsb admin	MSADAuthenticator
	rsbuser	rsb user	MSADAuthenticator
	rseadmin	rse admin	MSADAuthenticator
	user		MSADAuthenticator
	webadmin		MSADAuthenticator
	weblogic	This user is the default administrator.	DefaultAuthenticator

New | Delete | Showing 1 to 18 of 18 Previous | Next

3. Click the **Groups** tab to see the list of groups the server can see. The highlighted groups are created in AD and can verify the Provider – MSADAuthenticator provider.



Groups

New Delete — Showing 1 to 16 of 16 Previous | Next

	Name	Description	Provider
☐	AdminChannelUsers	AdminChannelUsers can access the admin channel.	DefaultAuthenticator
☐	Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
	agAdminGroup	ag Admin Group	MSADAuthenticator
☐	agAdminGroup	agAdminGroup	DefaultAuthenticator
☐	AppTesters	AppTesters group.	DefaultAuthenticator
☐	CrossDomainConnectors	CrossDomainConnectors can make inter-domain calls from foreign domains.	DefaultAuthenticator
☐	Deployers	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
	logUserGroup		MSADAuthenticator
☐	Monitors	Monitors can view and modify all resource attributes and perform operations not restricted by roles.	DefaultAuthenticator
☐	Operators	Operators can view and modify all resource attributes and perform server lifecycle operations.	DefaultAuthenticator
☐	OracleSystemGroup	Oracle application software system group.	DefaultAuthenticator
	ribAdminGroup	Rib Admin Group	MSADAuthenticator
☐	RihaAdminGroup	Riha admin group	DefaultAuthenticator
	RsbAdminGroup	Rsb Admin Group	MSADAuthenticator
	rseAdminGroup	rse Admin Group	MSADAuthenticator
☐	rseAdminGroup		DefaultAuthenticator

New Delete — Showing 1 to 16 of 16 Previous | Next

B

Sample Data from RIB App Monitoring Service

A sample data from RIB App monitoring service is shown below.

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <S:Body>
      <ns0:getRibAppSystemStateResponse
xmlns:ns1="http://www.oracle.com/retail/integration/rib/rib-integration-runtime-in
fo"
xmlns:ns0="http://www.oracle.com/retail/rib/monitor/service/RibAppMonitorService">
         <ns1:rib-app-runtime-info id="rib-tafr" rib-app-status="RUNNING"
up-since="2016-09-09T05:15:22.814-04:00" total-events-count="334203">
            <ns1:rib-adapters>
               <ns1:subscriber id="rib-tafr.SeedData_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="20"
num-ber-of-commits="20" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:30:01.892-04:00"
most-recent-event-adapter-execution-time="10"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="11008"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                        <ns1:between-hours hour="5-6" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="3882"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="11008"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                        <ns1:between-hours hour="5-6" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="3882"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.OrderToOdrWH_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="26"
num-ber-of-commits="26" number-of-rollbacks="2"
most-recent-event-time="2016-09-15T03:49:36.763-04:00"
most-recent-event-adapter-execution-time="32"
```

```
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="22" adapter-execution-max-time="60012"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="22" adapter-execution-max-time="60012"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.Transfers_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="130"
num-ber-of-commits="42" number-of-rollbacks="88"
most-recent-event-time="2016-09-15T04:15:42.550-04:00"
most-recent-event-adapter-execution-time="60012"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="11" adapter-execution-max-time="60022"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        <ns1:between-hours hour="4-5" event-count="47"
adapter-execution-min-time="11" adapter-execution-max-time="60311"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="11" adapter-execution-max-time="60022"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        <ns1:between-hours hour="4-5" event-count="47"
adapter-execution-min-time="11" adapter-execution-max-time="60311"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WOOut_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="7"
num-ber-of-commits="7" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:33:29.768-04:00"
most-recent-event-adapter-execution-time="33"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="7"
adapt-er-execution-min-time="29" adapter-execution-max-time="11635"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="11"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="7"
adapt-er-execution-min-time="29" adapter-execution-max-time="11635"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="11"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WHToLocation_tafr_1"
```

```
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="21"
num-ber-of-commits="21" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:51:42.695-04:00"
most-recent-event-adapter-execution-time="29"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="11" adapter-execution-max-time="45"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="11" adapter-execution-max-time="45"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WHToWHPhys_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="21"
num-ber-of-commits="21" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:51:41.589-04:00"
most-recent-event-adapter-execution-time="28"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="10" adapter-execution-max-time="138"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="10" adapter-execution-max-time="138"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.Partner_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="107"
num-ber-of-commits="37" number-of-rollbacks="70"
most-recent-event-time="2016-09-15T04:04:55.033-04:00"
most-recent-event-adapter-execution-time="60014"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="94"
adapter-execution-min-time="11" adapter-execution-max-time="60017"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        <ns1:between-hours hour="4-5" event-count="13"
adapter-execution-min-time="12" adapter-execution-max-time="60015"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="94"
adapter-execution-min-time="11" adapter-execution-max-time="60017"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        <ns1:between-hours hour="4-5" event-count="13"
```

```
adapter-execution-min-time="12" adapter-execution-max-time="60015"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.StoresToStors_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="32"
num-ber-of-commits="32" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:53:20.954-04:00"
most-recent-event-adapter-execution-time="23"
most-recent-event-integration-api-execution-time="0">
                    <ns1:todays-events>
                       <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="11" adapter-execution-max-time="118"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                    </ns1:todays-events>
                    <ns1:events on="2016-09-15T00:00:00-04:00">
                       <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="11" adapter-execution-max-time="118"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.ASNOutToASNOt_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="82"
num-ber-of-commits="80" number-of-rollbacks="9"
most-recent-event-time="2016-09-15T04:06:13.656-04:00"
most-recent-event-adapter-execution-time="46"
most-recent-event-integration-api-execution-time="1">
                    <ns1:todays-events>
                       <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="12" adapter-execution-max-time="60014"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                       <ns1:between-hours hour="4-5" event-count="7"
adapt-er-execution-min-time="12" adapter-execution-max-time="85"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                    </ns1:todays-events>
                    <ns1:events on="2016-09-15T00:00:00-04:00">
                       <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="12" adapter-execution-max-time="60014"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                       <ns1:between-hours hour="4-5" event-count="7"
adapt-er-execution-min-time="12" adapter-execution-max-time="85"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.WOIn_tafr_1" adapt-er-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.969-04:00" adapter-type="MESSAGE_DRIVEN_
SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="14"
num-ber-of-commits="14" number-of-rollbacks="4"
most-recent-event-time="2016-09-15T03:39:01.327-04:00"
most-recent-event-adapter-execution-time="12"
most-recent-event-integration-api-execution-time="1">
                    <ns1:todays-events>
```

```
                        <ns1:between-hours hour="3-4" event-count="14"
adapter-execution-min-time="12" adapter-execution-max-time="60053"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="14"
adapter-execution-min-time="12" adapter-execution-max-time="60053"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.ItemsToItmTL_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="82"
num-ber-of-commits="82" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:01:45.400-04:00"
most-recent-event-adapter-execution-time="11"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="8389"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="4-5" event-count="39"
adapter-execution-min-time="10" adapter-execution-max-time="6102"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="5-6" event-count="2"
adapt-er-execution-min-time="11" adapter-execution-max-time="13"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="8389"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="4-5" event-count="39"
adapter-execution-min-time="10" adapter-execution-max-time="6102"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="5-6" event-count="2"
adapt-er-execution-min-time="11" adapter-execution-max-time="13"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.SOStatus_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="15"
num-ber-of-commits="15" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:35:32.813-04:00"
most-recent-event-adapter-execution-time="23"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="15"
adapter-execution-min-time="11" adapter-execution-max-time="2612"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="15"
adapter-execution-min-time="11" adapter-execution-max-time="2612"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
```

```
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.Alloc_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="83"
num-ber-of-commits="29" number-of-rollbacks="54"
most-recent-event-time="2016-09-15T03:58:37.051-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="0">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="12" adapter-execution-max-time="60302"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:todays-events>
                  <ns1:events on="2016-09-15T00:00:00-04:00">
                     <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="12" adapter-execution-max-time="60302"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.ASNInToASNInL_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.965-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="10"
num-ber-of-commits="10" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:38:21.401-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="1">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="12" adapter-execution-max-time="3464"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                  </ns1:todays-events>
                  <ns1:events on="2016-09-15T00:00:00-04:00">
                     <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="12" adapter-execution-max-time="3464"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.StoresToLoc_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="32"
num-ber-of-commits="32" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:52:57.036-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="0">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="9" adapter-execution-max-time="246"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                  </ns1:todays-events>
                  <ns1:events on="2016-09-15T00:00:00-04:00">
                     <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="9" adapter-execution-max-time="246"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
```

```
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.ASNOutToASNIn_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="83"
num-ber-of-commits="81" number-of-rollbacks="5"
most-recent-event-time="2016-09-15T04:06:15.481-04:00"
most-recent-event-adapter-execution-time="33"
most-recent-event-integration-api-execution-time="0">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="19" adapter-execution-max-time="60035"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     <ns1:between-hours hour="4-5" event-count="8"
adapt-er-execution-min-time="26" adapter-execution-max-time="41"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:todays-events>
                  <ns1:events on="2016-09-15T00:00:00-04:00">
                     <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="19" adapter-execution-max-time="60035"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     <ns1:between-hours hour="4-5" event-count="8"
adapt-er-execution-min-time="26" adapter-execution-max-time="41"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.OrderToOdrISO_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="26"
num-ber-of-commits="26" number-of-rollbacks="2"
most-recent-event-time="2016-09-15T03:49:37.990-04:00"
most-recent-event-adapter-execution-time="30"
most-recent-event-integration-api-execution-time="0">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="24" adapter-execution-max-time="60012"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:todays-events>
                  <ns1:events on="2016-09-15T00:00:00-04:00">
                     <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="24" adapter-execution-max-time="60012"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                  </ns1:events>
               </ns1:events-processed>
            </ns1:subscriber>
            <ns1:subscriber id="rib-tafr.CustOrder_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
               <ns1:events-processed total-events-count="21"
num-ber-of-commits="7" number-of-rollbacks="14"
most-recent-event-time="2016-09-15T03:39:18.728-04:00"
most-recent-event-adapter-execution-time="38"
most-recent-event-integration-api-execution-time="1">
                  <ns1:todays-events>
                     <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="12" adapter-execution-max-time="60016"
```

```
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                    </ns1:todays-events>
                    <ns1:events on="2016-09-15T00:00:00-04:00">
                       <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="12" adapter-execution-max-time="60016"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.UDAs_tafr_1" adapt-er-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.969-04:00" adapter-type="MESSAGE_DRIVEN_
SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="10"
num-ber-of-commits="10" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:34:45.952-04:00"
most-recent-event-adapter-execution-time="11"
most-recent-event-integration-api-execution-time="0">
                    <ns1:todays-events>
                       <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="11" adapter-execution-max-time="60732"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                    </ns1:todays-events>
                    <ns1:events on="2016-09-15T00:00:00-04:00">
                       <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="11" adapter-execution-max-time="60732"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.RTVReq_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="24"
num-ber-of-commits="24" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:15:52.080-04:00"
most-recent-event-adapter-execution-time="34"
most-recent-event-integration-api-execution-time="1">
                    <ns1:todays-events>
                       <ns1:between-hours hour="3-4" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="11770"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                       <ns1:between-hours hour="5-6" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="667"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                    </ns1:todays-events>
                    <ns1:events on="2016-09-15T00:00:00-04:00">
                       <ns1:between-hours hour="3-4" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="11770"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                       <ns1:between-hours hour="5-6" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="667"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                    </ns1:events>
                 </ns1:events-processed>
              </ns1:subscriber>
              <ns1:subscriber id="rib-tafr.ItemsToItmISO_tafr_1"
adapt-er-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                 <ns1:events-processed total-events-count="82"
num-ber-of-commits="82" number-of-rollbacks="0"
```

```
most-recent-event-time="2016-09-15T05:02:09.784-04:00"
most-recent-event-adapter-execution-time="12"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="915"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="4-5" event-count="38"
adapter-execution-min-time="33" adapter-execution-max-time="3650"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="5-6" event-count="3"
adapt-er-execution-min-time="10" adapter-execution-max-time="13"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="915"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="4-5" event-count="38"
adapter-execution-min-time="33" adapter-execution-max-time="3650"
integra-tion-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        <ns1:between-hours hour="5-6" event-count="3"
adapt-er-execution-min-time="10" adapter-execution-max-time="13"
integra-tion-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:hospital id="rib-tafr.jms_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.971-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
               <ns1:hospital id="rib-tafr.sub_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.971-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
            </ns1:rib-adapters>
            <ns1:error-hospital-db total-messages-in-eh="0"
to-tal-messages-in-eh-due-to-dependency="0"/>
            <ns1:resource-usage>
               <ns1:cpu current="0.010379236"/>
               <ns1:memory current="1.06378035E9" max="1.90893261E9"
free="5.11592832E8"/>
            </ns1:resource-usage>
         </ns1:rib-app-runtime-info>
      </ns0:getRibAppSystemStateResponse>
   </S:Body>
</S:Envelope>
```

C

Sample Data from Integration Monitoring Service

A sample data from the integration monitoring service is given below:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <S:Body>
      <ns0:getRibIntegrationSystemStateResponse
xmlns:ns1="http://www.oracle.com/retail/integration/rib/rib-integration-runtime-in
fo"
xmlns:ns0="http://www.oracle.com/retail/rib/monitor/service/RibIntegrationMonitorS
ervice">
         <ns1:rib-integration-runtime-info
data-requested-at="2016-09-15T15:50:44.920-04:00">
            <ns1:rib-app-runtime-info id="rib-rms" rib-app-status="RUNNING"
up-since="2016-09-10T02:56:39.520-04:00" total-events-count="141757">
               <ns1:rib-adapters>
                  <ns1:subscriber id="rib-rms.StkCountSch_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.WOStatus_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.XItemRcls_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.065-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
```

```
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.DSRcpt_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.XCostChg_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.XItem_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.065-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.FulfilOrd_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.SOStatus_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.063-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:subscriber>
                 <ns1:subscriber id="rib-rms.PayTerm_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.063-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                       <ns1:todays-events/>
```

```
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XMrchHr_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.066-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XLocTrt_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.065-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XDiffID_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.065-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.Receiving_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.063-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="26"
number-of-commits="14" number-of-rollbacks="12"
most-recent-event-time="2016-09-15T01:58:38.440-04:00"
most-recent-event-adapter-execution-time="54"
most-recent-event-integration-api-execution-time="535">
                     <ns1:todays-events>
                        <ns1:between-hours hour="0-1" event-count="10"
adapter-execution-min-time="35" adapter-execution-max-time="19019"
integration-api-execution-min-time="2449"
integration-api-execution-max-time="18995"/>
                        <ns1:between-hours hour="1-2" event-count="16"
adapter-execution-min-time="35" adapter-execution-max-time="6531"
integration-api-execution-min-time="353"
integration-api-execution-max-time="6519"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="0-1" event-count="10"
adapter-execution-min-time="35" adapter-execution-max-time="19019"
integration-api-execution-min-time="2449"
integration-api-execution-max-time="18995"/>
                        <ns1:between-hours hour="1-2" event-count="16"
adapter-execution-min-time="35" adapter-execution-max-time="6531"
integration-api-execution-min-time="353"
integration-api-execution-max-time="6519"/>
                     </ns1:events>
```

```
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XOrder_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.066-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.OTB_sub_1" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.063-04:00" adapter-type="MESSAGE_DRIVEN_
SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="2"
number-of-commits="2" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T01:40:48.954-04:00"
most-recent-event-adapter-execution-time="627"
most-recent-event-integration-api-execution-time="619">
                     <ns1:todays-events>
                        <ns1:between-hours hour="1-2" event-count="2"
adapter-execution-min-time="627" adapter-execution-max-time="2070"
integration-api-execution-min-time="619"
integration-api-execution-max-time="2055"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="1-2" event-count="2"
adapter-execution-min-time="627" adapter-execution-max-time="2070"
integration-api-execution-min-time="619"
integration-api-execution-max-time="2055"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XStore_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.066-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XItemLoc_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.065-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.COCogs_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.060-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
```

```
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.ASNIn_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.060-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.FrtTerm_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.XMrchHrRcls_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.066-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.COSale_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.061-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.XOrgHr_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.066-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.GLCOA_sub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="1"
number-of-commits="1" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T02:31:47.147-04:00"
```

```
most-recent-event-adapter-execution-time="603"
most-recent-event-integration-api-execution-time="589">
                        <ns1:todays-events>
                           <ns1:between-hours hour="2-3" event-count="1"
adapter-execution-min-time="603" adapter-execution-max-time="603"
integration-api-execution-min-time="589"
integration-api-execution-max-time="589"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="2-3" event-count="1"
adapter-execution-min-time="603" adapter-execution-max-time="603"
integration-api-execution-min-time="589"
integration-api-execution-max-time="589"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.CurRate_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.061-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.InvReq_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="11"
number-of-commits="6" number-of-rollbacks="5"
most-recent-event-time="2016-09-10T05:28:12.333-04:00"
most-recent-event-adapter-execution-time="45"
most-recent-event-integration-api-execution-time="179">
                        <ns1:todays-events/>
                        <ns1:events on="2016-09-10T00:00:00-04:00">
                           <ns1:between-hours hour="5-6" event-count="11"
adapter-execution-min-time="28" adapter-execution-max-time="511"
integration-api-execution-min-time="45" integration-api-execution-max-time="415"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.XTsf_sub_1" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.066-04:00" adapter-type="MESSAGE_DRIVEN_
SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-rms.Vendor_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events/>
```

```
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.DSDReceipt_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.061-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="51"
number-of-commits="27" number-of-rollbacks="24"
most-recent-event-time="2016-09-14T06:56:22.187-04:00"
most-recent-event-adapter-execution-time="530"
most-recent-event-integration-api-execution-time="472">
                     <ns1:todays-events/>
                     <ns1:events on="2016-09-10T00:00:00-04:00">
                        <ns1:between-hours hour="5-6" event-count="10"
adapter-execution-min-time="35" adapter-execution-max-time="3539"
integration-api-execution-min-time="1486"
integration-api-execution-max-time="3510"/>
                     </ns1:events>
                     <ns1:events on="2016-09-14T00:00:00-04:00">
                        <ns1:between-hours hour="6-7" event-count="41"
adapter-execution-min-time="18" adapter-execution-max-time="17712"
integration-api-execution-min-time="293"
integration-api-execution-max-time="17661"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.COReturn_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.061-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XDiffGrp_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.InvAdjust_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.062-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="1"
number-of-commits="1" number-of-rollbacks="0"
most-recent-event-time="2016-09-14T06:51:42.811-04:00"
most-recent-event-adapter-execution-time="7224"
most-recent-event-integration-api-execution-time="7203">
                     <ns1:todays-events/>
                     <ns1:events on="2016-09-14T00:00:00-04:00">
                        <ns1:between-hours hour="6-7" event-count="1"
adapter-execution-min-time="7224" adapter-execution-max-time="7224"
integration-api-execution-min-time="7203"
integration-api-execution-max-time="7203"/>
                     </ns1:events>
```

```
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.RTV_sub_1" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.063-04:00" adapter-type="MESSAGE_DRIVEN_
SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.ASNOut_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.060-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events/>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.DSDDeals_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.061-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="10"
number-of-commits="5" number-of-rollbacks="5"
most-recent-event-time="2016-09-15T02:15:27.400-04:00"
most-recent-event-adapter-execution-time="41"
most-recent-event-integration-api-execution-time="0">
                     <ns1:todays-events>
                        <ns1:between-hours hour="2-3" event-count="10"
adapter-execution-min-time="23" adapter-execution-max-time="1613"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="2-3" event-count="10"
adapter-execution-min-time="23" adapter-execution-max-time="1613"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-rms.XAlloc_sub_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.064-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="1"
number-of-commits="1" number-of-rollbacks="0"
most-recent-event-time="2016-09-14T12:34:57.301-04:00"
most-recent-event-adapter-execution-time="4684"
most-recent-event-integration-api-execution-time="4662">
                     <ns1:todays-events/>
                     <ns1:events on="2016-09-14T00:00:00-04:00">
                        <ns1:between-hours hour="12-13" event-count="1"
adapter-execution-min-time="4684" adapter-execution-max-time="4684"
integration-api-execution-min-time="4662"
integration-api-execution-max-time="4662"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:publisher id="rib-rms.RcvUnitAdj_pub_1"
```

```
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.ASNOut_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.070-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Vendor_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.072-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.WOOut_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.072-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Stores_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.072-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Diffs_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.070-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.SeedData_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
```

```
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.Partner_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.ItemLoc_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.UDAs_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.072-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.MerchHier_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.DiffGrp_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.070-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.Order_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.071-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                       <ns1:todays-events/>
                    </ns1:events-processed>
                 </ns1:publisher>
                 <ns1:publisher id="rib-rms.SeedObj_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.072-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                    <ns1:events-processed total-events-count="0"
```

```
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.RTVReq_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Items_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.071-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Alloc_pub_1" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.070-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.FulfilOrdCfm_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.DlvySlt_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.071-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Transfers_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.072-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.WH_pub_1" adapter-status="RUNNING"
```

```
data-collection-time="2016-09-15T15:48:00.072-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.WOIn_pub_1" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.072-04:00" adapter-type="TIMER_DRIVEN_
PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:publisher id="rib-rms.Banner_pub_1"
adapter-status="RUNNING" data-collection-time="2016-09-15T15:48:00.070-04:00"
adapter-type="TIMER_DRIVEN_PUBLISHER_TYPE">
                     <ns1:events-processed total-events-count="0"
number-of-commits="0" number-of-rollbacks="0"
most-recent-event-adapter-execution-time="0">
                        <ns1:todays-events/>
                     </ns1:events-processed>
                  </ns1:publisher>
                  <ns1:hospital id="rib-rms.jms_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.073-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
                  <ns1:hospital id="rib-rms.sub_hosp_0" adapter-status="RUNNING"
data-collection-time="2016-09-15T15:48:00.074-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
                  <ns1:hospital id="rib-rms.pub_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.074-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
               </ns1:rib-adapters>
               <ns1:error-hospital-db total-messages-in-eh="54"
total-messages-in-eh-due-to-dependency="32">
                  <ns1:messages-in-eh-for-family family="Receiving"
adapter-class-def="rib-rms_Receiving_sub" error-count="33" dependency-count="32"/>
                  <ns1:messages-in-eh-for-family family="DSDReceipt"
adapter-class-def="rib-rms_DSDReceipt_sub" error-count="2" dependency-count="0"/>
                  <ns1:messages-in-eh-for-family family="INVREQ"
adapter-class-def="rib-rms_InvReq_sub" error-count="2" dependency-count="0"/>
                  <ns1:messages-in-eh-for-family family="InvAdjust"
adapter-class-def="rib-rms_InvAdjust_sub" error-count="5" dependency-count="0"/>
                  <ns1:messages-in-eh-for-family family="CurRate"
adapter-class-def="rib-rms_CurRate_sub" error-count="10" dependency-count="0"/>
                  <ns1:messages-in-eh-for-family family="DSDDEALS"
adapter-class-def="rib-rms_DSDDeals_sub" error-count="1" dependency-count="0"/>
                  <ns1:messages-in-eh-for-family family="SOStatus"
adapter-class-def="rib-rms_SOStatus_sub" error-count="1" dependency-count="0"/>
               </ns1:error-hospital-db>
               <ns1:resource-usage>
                  <ns1:cpu current="0.022521019"/>
                  <ns1:memory current="1.0616832E9" max="1.90893261E9"
free="5.16266752E8"/>
               </ns1:resource-usage>
            </ns1:rib-app-runtime-info>
            <ns1:rib-app-runtime-info id="rib-tafr" rib-app-status="RUNNING"
```

```
up-since="2016-09-09T05:15:22.814-04:00" total-events-count="333275">
               <ns1:rib-adapters>
                  <ns1:subscriber id="rib-tafr.SeedData_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="20"
number-of-commits="20" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:30:01.892-04:00"
most-recent-event-adapter-execution-time="10"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="11008"
integration-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                           <ns1:between-hours hour="5-6" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="3882"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="11008"
integration-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                           <ns1:between-hours hour="5-6" event-count="10"
adapter-execution-min-time="10" adapter-execution-max-time="3882"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.OrderToOdrWH_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="26"
number-of-commits="26" number-of-rollbacks="2"
most-recent-event-time="2016-09-15T03:49:36.763-04:00"
most-recent-event-adapter-execution-time="32"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="22" adapter-execution-max-time="60012"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="22" adapter-execution-max-time="60012"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.Transfers_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="130"
number-of-commits="42" number-of-rollbacks="88"
most-recent-event-time="2016-09-15T04:15:42.550-04:00"
most-recent-event-adapter-execution-time="60012"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="11" adapter-execution-max-time="60022"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
```

```
                        <ns1:between-hours hour="4-5" event-count="47"
adapter-execution-min-time="11" adapter-execution-max-time="60311"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="11" adapter-execution-max-time="60022"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        <ns1:between-hours hour="4-5" event-count="47"
adapter-execution-min-time="11" adapter-execution-max-time="60311"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WOOut_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="7"
number-of-commits="7" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:33:29.768-04:00"
most-recent-event-adapter-execution-time="33"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="7"
adapter-execution-min-time="29" adapter-execution-max-time="11635"
integration-api-execution-min-time="1" integration-api-execution-max-time="11"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="7"
adapter-execution-min-time="29" adapter-execution-max-time="11635"
integration-api-execution-min-time="1" integration-api-execution-max-time="11"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WHToLocation_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="21"
number-of-commits="21" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:51:42.695-04:00"
most-recent-event-adapter-execution-time="29"
most-recent-event-integration-api-execution-time="1">
                     <ns1:todays-events>
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="11" adapter-execution-max-time="45"
integration-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                     </ns1:todays-events>
                     <ns1:events on="2016-09-15T00:00:00-04:00">
                        <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="11" adapter-execution-max-time="45"
integration-api-execution-min-time="0" integration-api-execution-max-time="2"/>
                     </ns1:events>
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:subscriber id="rib-tafr.WHToWHPhys_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                  <ns1:events-processed total-events-count="21"
number-of-commits="21" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:51:41.589-04:00"
```

```
most-recent-event-adapter-execution-time="28"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="10" adapter-execution-max-time="138"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="10" adapter-execution-max-time="138"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.Partner_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="107"
number-of-commits="37" number-of-rollbacks="70"
most-recent-event-time="2016-09-15T04:04:55.033-04:00"
most-recent-event-adapter-execution-time="60014"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="94"
adapter-execution-min-time="11" adapter-execution-max-time="60017"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                           <ns1:between-hours hour="4-5" event-count="13"
adapter-execution-min-time="12" adapter-execution-max-time="60015"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="94"
adapter-execution-min-time="11" adapter-execution-max-time="60017"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                           <ns1:between-hours hour="4-5" event-count="13"
adapter-execution-min-time="12" adapter-execution-max-time="60015"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.StoresToStors_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="32"
number-of-commits="32" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:53:20.954-04:00"
most-recent-event-adapter-execution-time="23"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="11" adapter-execution-max-time="118"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="11" adapter-execution-max-time="118"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
```

```
                  <ns1:subscriber id="rib-tafr.ASNOutToASNOt_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="82"
number-of-commits="80" number-of-rollbacks="9"
most-recent-event-time="2016-09-15T04:06:13.656-04:00"
most-recent-event-adapter-execution-time="46"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="12" adapter-execution-max-time="60014"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="7"
adapter-execution-min-time="12" adapter-execution-max-time="85"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="12" adapter-execution-max-time="60014"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="7"
adapter-execution-min-time="12" adapter-execution-max-time="85"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.WOIn_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="14"
number-of-commits="14" number-of-rollbacks="4"
most-recent-event-time="2016-09-15T03:39:01.327-04:00"
most-recent-event-adapter-execution-time="12"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="14"
adapter-execution-min-time="12" adapter-execution-max-time="60053"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="14"
adapter-execution-min-time="12" adapter-execution-max-time="60053"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.ItemsToItmTL_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="82"
number-of-commits="82" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:01:45.400-04:00"
most-recent-event-adapter-execution-time="11"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="8389"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="39"
adapter-execution-min-time="10" adapter-execution-max-time="6102"
```

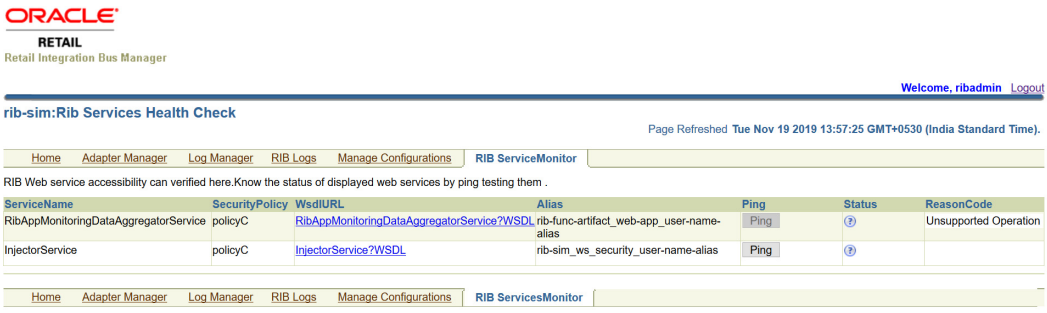
```
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="5-6" event-count="2"
adapter-execution-min-time="11" adapter-execution-max-time="13"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="8389"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="39"
adapter-execution-min-time="10" adapter-execution-max-time="6102"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="5-6" event-count="2"
adapter-execution-min-time="11" adapter-execution-max-time="13"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.SOStatus_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="15"
number-of-commits="15" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:35:32.813-04:00"
most-recent-event-adapter-execution-time="23"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="15"
adapter-execution-min-time="11" adapter-execution-max-time="2612"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="15"
adapter-execution-min-time="11" adapter-execution-max-time="2612"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.Alloc_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="83"
number-of-commits="29" number-of-rollbacks="54"
most-recent-event-time="2016-09-15T03:58:37.051-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="12" adapter-execution-max-time="60302"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="83"
adapter-execution-min-time="12" adapter-execution-max-time="60302"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.ASNInToASNInL_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.965-04:00"
```

```
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="10"
number-of-commits="10" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:38:21.401-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="12" adapter-execution-max-time="3464"
integration-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="12" adapter-execution-max-time="3464"
integration-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.StoresToLoc_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.968-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="32"
number-of-commits="32" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:52:57.036-04:00"
most-recent-event-adapter-execution-time="25"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="9" adapter-execution-max-time="246"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="32"
adapter-execution-min-time="9" adapter-execution-max-time="246"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.ASNOutToASNIn_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="83"
number-of-commits="81" number-of-rollbacks="5"
most-recent-event-time="2016-09-15T04:06:15.481-04:00"
most-recent-event-adapter-execution-time="33"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="19" adapter-execution-max-time="60035"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                           <ns1:between-hours hour="4-5" event-count="8"
adapter-execution-min-time="26" adapter-execution-max-time="41"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="75"
adapter-execution-min-time="19" adapter-execution-max-time="60035"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                           <ns1:between-hours hour="4-5" event-count="8"
adapter-execution-min-time="26" adapter-execution-max-time="41"
```

```
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.OrderToOdrISO_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="26"
number-of-commits="26" number-of-rollbacks="2"
most-recent-event-time="2016-09-15T03:49:37.990-04:00"
most-recent-event-adapter-execution-time="30"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="24" adapter-execution-max-time="60012"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="26"
adapter-execution-min-time="24" adapter-execution-max-time="60012"
integration-api-execution-min-time="0" integration-api-execution-max-time="0"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.CustOrder_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="21"
number-of-commits="7" number-of-rollbacks="14"
most-recent-event-time="2016-09-15T03:39:18.728-04:00"
most-recent-event-adapter-execution-time="38"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="12" adapter-execution-max-time="60016"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="21"
adapter-execution-min-time="12" adapter-execution-max-time="60016"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.UDAs_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.969-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="10"
number-of-commits="10" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T03:34:45.952-04:00"
most-recent-event-adapter-execution-time="11"
most-recent-event-integration-api-execution-time="0">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="11" adapter-execution-max-time="60732"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="10"
adapter-execution-min-time="11" adapter-execution-max-time="60732"
```

```
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.RTVReq_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.967-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="24"
number-of-commits="24" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:15:52.080-04:00"
most-recent-event-adapter-execution-time="34"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="11770"
integration-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                           <ns1:between-hours hour="5-6" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="667"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="11770"
integration-api-execution-min-time="1" integration-api-execution-max-time="3"/>
                           <ns1:between-hours hour="5-6" event-count="12"
adapter-execution-min-time="12" adapter-execution-max-time="667"
integration-api-execution-min-time="1" integration-api-execution-max-time="2"/>
                        </ns1:events>
                     </ns1:events-processed>
                  </ns1:subscriber>
                  <ns1:subscriber id="rib-tafr.ItemsToItmISO_tafr_1"
adapter-status="STOPPED" data-collection-time="2016-09-15T15:48:00.966-04:00"
adapter-type="MESSAGE_DRIVEN_SUBSCRIBER_TYPE">
                     <ns1:events-processed total-events-count="82"
number-of-commits="82" number-of-rollbacks="0"
most-recent-event-time="2016-09-15T05:02:09.784-04:00"
most-recent-event-adapter-execution-time="12"
most-recent-event-integration-api-execution-time="1">
                        <ns1:todays-events>
                           <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="915"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="38"
adapter-execution-min-time="33" adapter-execution-max-time="3650"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="5-6" event-count="3"
adapter-execution-min-time="10" adapter-execution-max-time="13"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:todays-events>
                        <ns1:events on="2016-09-15T00:00:00-04:00">
                           <ns1:between-hours hour="3-4" event-count="41"
adapter-execution-min-time="10" adapter-execution-max-time="915"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="4-5" event-count="38"
adapter-execution-min-time="33" adapter-execution-max-time="3650"
integration-api-execution-min-time="0" integration-api-execution-max-time="1"/>
                           <ns1:between-hours hour="5-6" event-count="3"
adapter-execution-min-time="10" adapter-execution-max-time="13"
integration-api-execution-min-time="1" integration-api-execution-max-time="1"/>
                        </ns1:events>
```

```
                  </ns1:events-processed>
               </ns1:subscriber>
               <ns1:hospital id="rib-tafr.jms_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.971-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
               <ns1:hospital id="rib-tafr.sub_hosp_0" adapter-status="STOPPED"
data-collection-time="2016-09-15T15:48:00.971-04:00" adapter-type="TIMER_DRIVEN_
HOSPITAL_SUB_TYPE"/>
            </ns1:rib-adapters>
            <ns1:error-hospital-db total-messages-in-eh="0"
total-messages-in-eh-due-to-dependency="0"/>
            <ns1:resource-usage>
               <ns1:cpu current="0.010379236"/>
               <ns1:memory current="1.06378035E9" max="1.90893261E9"
free="5.11592832E8"/>
            </ns1:resource-usage>
         </ns1:rib-app-runtime-info>
      </ns1:rib-integration-runtime-info>
   </ns0:getRibIntegrationSystemStateResponse>
</S:Body>
</S:Envelope>
```



D

REST Publisher Client Sample Code

A sample of the REST publisher client code is below:

```
//Import required classes

import com.oracle.retail.integration.base.bo.fulfilorddesc.v1.*
import com.oracle.retail.integration.payload.Payload
import com.retek.rib.domain.payload.PayloadFactory
import
com.oracle.retail.integration.rib.applicationmessages.v1.ApplicationMessage;
import
com.oracle.retail.integration.rib.applicationmessages.v1.ApplicationMessages;
//Create new instance of your FulfilOrdDesc object and populate it.
FulfilOrdDesc fulfilOrdDesc = new FulfilOrdDesc()
fulfilOrdDesc.setCustomerOrderNo(123)
//Get a string version of the payload
String payloadXml = PayloadFactory.marshalPayload(fulfilOrdDesc)
//Prepare the header message section
ApplicationMessages ams = new ApplicationMessages();
ApplicationMessage am = new ApplicationMessage();
am.setFamily("FULFILORD");
am.setType("FULFILORDPOCRE");
am.setBusinessObjectId("abc"); //optional
//Set the payload xml into the message
am.setPayloadXml(payloadXml);
ams.getApplicationMessage().add(am);
//Call rest url with ams
       String ribPublisherRestUrl =
"http://msp8925.us.oracle.com:47019/rib-oms-services-web/resources/publisher/publi
sh"
       Client client = ClientBuilder.newClient();
       WebTarget webTarget = client.target(ribPublisherRestUrl);
       String userName = "ribadmin";
       char[] password = "ribadmin1";

       String usernameAndPassword = userName + ":" + new String(password);
       String authorizationHeaderValue = "Basic " +
java.util.Base64.getEncoder().encodeToString( usernameAndPassword.getBytes() );
       Invocation.Builder invocationBuilder =
webTarget.request().header("Authorization", authorizationHeaderValue);
       Response response = invocationBuilder.post(Entity.entity(ams,
MediaType.APPLICATION_XML));
       log.debug("Publish call response(" + response + ").");
```

Notes/FAQ

Q1. Is using PayloadFactory required? OR can I use standard JAXB?

A1. It's recommended that you use the PaloadFactory as it handles the html encoding that's required while including the payload in the RIBMessages envelope. All RIB connected apps use the API and we maintain it so that if there are any fixes that can be distributed. It also takes care of validation etc and when you're using RBO's it's best to use this class so that the family/type bind to the right JAXB beans

Q2. Can we send multiple ApplicationMessage as part of ApplicationMessages?

A2. Yes, multiple AM's are typically published together each AM corresponds to a RIBMessage node and AMS corresponds to RIBMessages envelope.

Q3. How do I know if my message made it successfully?

A3. You get HTTP status 200 OK when Publish is successful. The client receives a JSON string with a "Message" on it saying "Publish done".

Q4. Where can see the RIB logs for the messages I published?

A4. You can login to RIB-<app> admin GUI and look at the logs as you test to see the messages, errors, and so on. See below.