Oracle® Retail Integration Bus Cloud Service

Installation Guide

Release 19.1.000

F31805-01

August 2020

Oracle Retail Integration Bus Cloud Service Installation Guide, Release 19.1.000

F31805-01

Primary Author: Sanal Parameswaran

Contributing Author: Nathan Young

Contents

Send Us Your Comments

Oracle Retail Integration Bus Cloud Service Installation Guide, Release 19.1.000

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

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

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

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

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

Oracle Retail Installation Guides contain the requirements and procedures that are necessary for the retailer to install Oracle Retail products.

Audience

The Installation Guide is written for the following audiences:

- Database administrators (DBA)
- System analysts and designers
- Integrators and implementation staff

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 19.1) or a later patch release (for example, 19.1.001). If you are installing the base release and additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

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.

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 Technology Network

Oracle Retail product documentation is available on the following web site:

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

(Data Model documents are not available through Oracle Technology Network. You can obtain these documents 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

This manual details the installation of the Retail Integration Bus (RIB). Generally, a RIB installation contains the following components:

- An installation of the Retail Integration RIB Hospital administration (RIHA) tool.
- An installation of the RIB Diagnostics and Monitoring tools.

The RIB includes an optional component, the RIB Integration Gateway Services (IGS) that can be installed as a subsystem to the core RIB. The IGS should be installed after the core RIB components have been successfully installed and tested.

Note: See the "Integration Gateway Services" section in Chapter 3, "Core Concepts," in the *Oracle Retail Integration Bus Implementation Guide* before attempting installation.

It is important to also follow all installation steps of the Oracle Retail Applications that are being connected to the RIB. Failure to follow these may result in a faulty RIB installation. See the installation guides for the relevant Oracle Retail applications for more information.

Note: The instructions provided in this guide apply to a full installation of the RIB. The RIB 19.1.000 application cannot be installed over an existing version, such as 15.x.x.

RIB Installation Master Checklist

This list covers all of the sequential steps required to perform a full installation of the RIB, using a command line installation.

Task	Notes
Prepare the Oracle WebLogic Servers for installation of the RIB Components.	Prerequisite
Prepare the Oracle Database schemas that the RIB will use.	Prerequisite
Prepare the JMS.	Prerequisite
Verify the Applications to which RIB will be integrating are configured appropriately.	

Task	Notes
Information to gather for the Installation	During the prerequisites steps, there is information that should be noted that will be used to configure the RIB during the installation process.
Install the RIB using the RIB App Builder Command Line Tools.	
Verify Application URL settings match RIB installation.	RIB Functional Artifact URL JNDI URL
Complete the setup of RDMT using the same Information to Gather for the Installation.	During either of the installation methods, one of the manual steps will have extracted the RDMT tools to the appropriate directory.
Verify the RIB installation using the RDMT tools.	
Install RIHA.	The RIB Hospital maintenance tool

Note: See Appendix B, “RIB Installation Checklists,” while performing the installation to minimize the chance of errors.

The RIB Integration Gateway Services (IGS) is an optional component and should be installed after the installation and verification of the RIB components.

Task	Notes
Prepare the WebLogic application servers for installation of the IGS component.	This is a mandatory prerequisite.
Information to gather for the Installation	During the RIB component prerequisites steps, there is information that should be noted that will be used to configure the IGS during the installation process.
Install the IGS.	
Verify the IGS installation using the Soap UI tool and test cases.	See Chapter 4 of the *Oracle Retail Integration Bus Operations Guide*.

Note: See Appendix B, “RIB Installation Check Lists,” while performing the installation to minimize the chance of errors.

Technical Specifications

The RIB and Integration Gateway Services have several dependencies on Oracle Retail Application installations, as well as on the Oracle WebLogic Servers. This section covers these requirements.

Requesting Infrastructure Software

If you are unable to find the necessary version of the required Oracle infrastructure software (database server, application server, WebLogic, etc.) on the Oracle Software Delivery Cloud, you should file a non-technical 'Contact Us' Service Request (SR) and request access to the media. For instructions on filing a non-technical SR, see My

Oracle Support Note 1071023.1 - *Requesting Physical Shipment or Download URL for Software Media*.

Check Server Requirements

Note: Oracle Retail assumes that the retailer has applied all required fixes for supported compatible technologies.

Supported On	Versions Supported
Database Server OS	OS certified with Oracle Database 12c (12.x) Enterprise Edition. Options are: ■ Oracle Linux 6 or 7 for x86-64 (Actual hardware or Oracle virtual machine). ■ Red Hat Enterprise Linux 6 or 7 for x86-64 (actual hardware or Oracle virtual machine) ■ IBM AIX 7.1 (actual hardware or LPARs) ■ Solaris 11.2 SPARC (actual hardware or logical domains) ■ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Database Server 12cR1	Oracle Database Enterprise Edition 12cR1 (12.1.0.2) with the following specifications: **Components:** ■ Oracle Partitioning ■ Examples CD **Oneoffs:** ■ 20846438: ORA-600 [KKPAPXFORMFKK2KEY_1] WITH LIST PARTITION ■ 19623450: MISSING JAVA CLASSES AFTER UPGRADE TO JDK 7 ■ 20406840: PROC 12.1.0.2 THROWS ORA-600 [17998] WHEN PRECOMPILING BY 'OTHER' USER ■ 20925154: ORA-39126: WORKER UNEXPECTED FATAL ERROR IN KUPW$WORKER GATHER_ PARSE_ITEMS JAVA ■ 19672263: Patch 19672263: GTT SESSION LEVEL STATISTICS RETURNS ORA-20006 **RAC only:** ■ 21260431: APPSST 12C: GETTING ORA-4031 AFTER 12C UPGRADE ■ 21373473: INSTANCE TERMINATED AS LMD0 AND LMD2 HUNG FOR MORE THAN 70 SECS **Other Components:** ■ Perl interpreter 5.0 or later ■ X-Windows interface ■ JDK 1.8 with latest security updates

Supported On	**Versions Supported**
Database Server 19c	Oracle Database Enterprise Edition 19c (19.3.0.0) with the following components: **Components:** ▪ DB HOME ▪ Examples CD **Other Components:** ▪ Perl Interpreter 5.0 or later ▪ X-Windows interface ▪ JDK 1.8
AQ JMS Server	Oracle Database 12cR1 or 19c
Application Server OS	OS certified with Oracle Fusion Middleware 12.2.1.4.0. Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (actual hardware or Oracle virtual machine) ▪ IBM AIX 7.1 (actual hardware or LPARs) ▪ Solaris 11.2 Sparc (actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Application Server	Oracle Fusion Middleware 12.2.1.4.0 **Components:** ▪ Oracle WebLogic Server 12.2.1.4.0 **Note:** WebLogic 12c domain for RIB needs JRF to be installed. Choose the JRF option at the domain creation. JRF needs an RCU schema for its runtime. This schema must be created prior to domain creation using the RCU tool. **Java:** ▪ JDK 1.8.0+ 64 bit with latest security updates
Minimum required JAVA version for all operating systems	Java: JDK 1.8.0+ 64 bit with latest security updates

Note: By default, JDK is at 1.7. After installing the rdbms binary, apply patch 19623450. Then follow the instructions on *Oracle Database Java Developer's Guide 12c (12 x) Release 1* to change JDK to 1.8 with latest security updates. The document is available at:

http://docs.oracle.com/database/121/JJDEV/chone.htm#JJDEV01000.

Follow-through to complete the post-patch operation.

Note: Use GNU Tar when installing on AIX as other utilities fail when extracting the RIB tarballs.

Note: Bash shell is supported. Using other shells could have adverse effects.

Important: If there is an existing WebLogic installation on the server, you must upgrade to WebLogic 12.2.1.4.0. All middleware components associated with WebLogic server should be upgraded to 12.2.1.4.0.

Back up the weblogic.policy file ($WLS_HOME/wlserver/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.

Additional Requirement if using Oracle RIB Hospital Administration (RIHA)

The RIHA model and view components require ADF runtime to run properly. Verify that ADF runtime 12.2.1 or higher is available in the WebLogic Application Server (12.2.1.4.0) and applied to the domain where RIHA will be installed.

Other Resources

For information about WebLogic Application Server 12.2.1.4.0, see the Oracle WebLogic Server Documentation Library:

https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/index.htm

Note: See also the Oracle Database Administrator's Guide 12c (12.x) and the Oracle WebLogic Application Server 12.2.1.4.0 documentation.

RIB Integration Gateway Services (IGS) Supported Operating Systems

Supported On	Version Supported
Oracle WebLogic Server OS	OS certified with Oracle Fusion Middleware 12.2.1.4.0. Options are: ▪ Oracle Linux 6 or 7 for x86-64 (Actual hardware or Oracle virtual machine) ▪ Red Hat Enterprise Linux 6 or 7 for x86-64 (Actual hardware or Oracle virtual machine) ▪ IBM AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11.2 Sparc (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Oracle WebLogic Server	Oracle WebLogic Server 12.2.1.4.0

RIB and Oracle Database Cluster (RAC)

In this release, rib-<app> uses Oracle Streams AQ as the JMS provider. Oracle Streams AQ is built on top of Oracle database system. Because AQ is hosted by the Oracle database system, RIB can take advantage of database RAC capability for its JMS provider. By using RAC AQ as RIB's JMS provider, you can scale RIB's JMS server vertically and horizontally to meet any retailer's scalability and high availability need.

At runtime, rib-<app> uses the database for keeping track of its RIB Hospital records. These RIB Hospital tables can be hosted by an Oracle RAC database providing high availability and scalability for these RIB Hospital records.

All rib-<app>s use the Oracle type 4 Java Database Connectivity (JDBC) driver to connect to the RIB Hospital database and the AQ JMS server. When the RIB Hospital database and the AQ JMS servers are hosted by an Oracle RAC database, the only configuration change required in rib-<app> is the RAC JDBC connection URL.

Note: RIB supports only the use of the Oracle Type 4 Thin Java Database Connectivity (JDBC) driver (ojdbc8.jar) for all JDBC connections, including RAC.

RIB and Oracle WebLogic Server Cluster

RIB uses JMS server for message transportation between the integrating retail applications. Because RIB must preserve the message publication and subscription ordering, rib-<app>s deployed in Oracle WebLogic Server cannot be configured in an active-active cluster mode. In active-active cluster mode, multiple subscribers and publishers will process messages simultaneously and there will be no way to preserve message ordering.

The rib-<app> can be deployed to a single instance of an Oracle WebLogic server that is clustered (active-passive). In this configuration, even though rib-<app> is deployed in a WebLogic cluster, multiple instances of the same rib-<app> are not running at the same time, as there is only one WebLogic instance where the rib-<app> is deployed. So RIB can still preserve message ordering.

To truly configure rib-<app>s for high availability, the only option is to configure it in active-passive mode.

Note: See Chapter 5, "RIB High Availability Installation Instructions" for more details.

2 Preinstallation Tasks

Before you begin the installation process, read the *Oracle Retail Integration Bus Implementation Guide* to plan a RIB deployment.

Planning may include the decision to employ multiple JMS servers, which can isolate flows for performance and operational QoS. For information, see "Preinstallation Steps for Multiple JMS Server Setup" in this guide.

Determine the UNIX User Account to Install the Software

The user account that installs RIB is an important consideration. Options, pros, and cons are discussed in the *Oracle Retail Integration Bus Implementation Guide*.

Note: See the "Pre-Implementation Considerations" in the *Oracle Retail Integration Bus Implementation Guide*.

rib-home Directory

The RIB software components can be distributed across multiple application servers depending on the deployment option selected, but they are centrally configured and managed.

Note: See the section, "Recommended Deployment Option," in the *Oracle Retail Integration Bus Implementation Guide*.

The location from which all rib-<app> applications are managed is known as rib-home. This directory location (rib-home) contains all the tools and configurations to manage the life cycle and operations of the RIB installation across the enterprise. There must be one rib-home directory for each development, test and production environment. The rib-home directory is not a staging (throw away) directory. It must be available at all times to support the lifecycle management of the RIB system. After initial configuration of the Database server and the Java EE application server, all rib-<app> application level work must be done only from the rib-home directory location.

Note: See the section, "RIB Software Life Cycle," in the *Oracle Retail Integration Bus Implementation Guide*.

Prepare WebLogic Application Server for RIB Components

This section describes the process of preparing the Oracle WebLogic servers to install the rib-<app> Java EE application.

Create the RIB Managed Server Instances

All RIB components are Java EE and run in WebLogic managed server instances in the WebLogic Application Server. The rib-<app> Java EE application runs in its own managed server instance called rib-<app>-server. Each rib-<app> application requires a separate managed server instance that is not shared with any other application. All managed servers can be under one domain. It is a new requirement from 16.0 onwards that the WLS domain for RIB deployment must be a JRF domain. JRF needs RCU.

WebLogic 12.2.1.4.0, needs OWSM for Policy A and C to work.

The OWSM template choice while creating the wls domain, will give an option to deploy wsm-pm application to admin server. This app is required for policies to work. Ensure that wsm-pm is targeted to the admin server and all the managed servers where rib-<app> Java EE applications are going to be deployed.

If http ports are disabled in the server, then wsm-pm app will not be reachable, unless it's configured to use SSL port. To configure SSL ports for wsm-pm, EM has to be deployed.

Note: For additional information about configuring the RIB Domain, see Appendix E, "Appendix: RIB Domain Configuration for Policy A".

Use the following steps to create a new managed server instance for rib-<app> and configure it to RIB requirement.

Note: For information about using commands to create a managed server instance, see the WebLogic Application Server Administrator's Guide 12.2.1.4.0.

Acceptable values for <app> are rms, rwms, tafr, sim, rpm, aip, rob, lgf, ocds, ext, and rfm.

There is one RIB specific managed server instance that must be created regardless of the other application deployment choices.

- rib-func-artifact-server. (This naming convention is recommended, but not required.)

There is one RIB specific managed server instance that must be created depending on the deployment configuration. If RMS is installed with RWMS and/or SIM, the TAFRs must be installed.

- rib-tafr-server. (It is recommended, but not required, that this naming convention be followed).

The following is a list of optional application instances, depending on deployment choices. It is recommended, but not required, that you use the following naming convention:

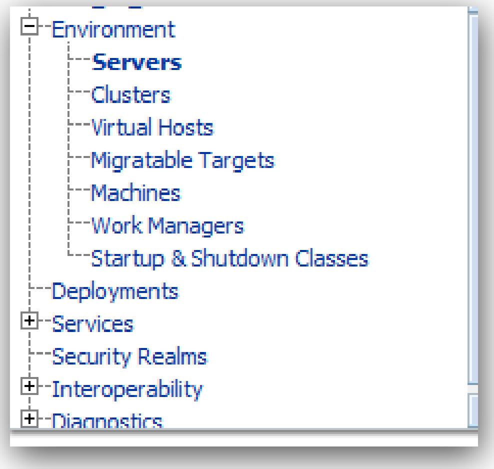
- rib-aip-server
- rib-rfm-server

- rib-rms-server
- rib-rpm-server
- rib-rwms-server
- rib-rob-server
- rib-sim-server
- rib-ext-server
- rib-lgf-server
- rib-ocds-server

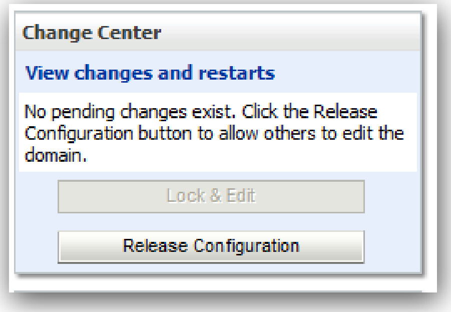
Note: See Oracle WebLogic Server 12.2.1.4.0 documentation for more details on How to Create managed servers.

To create the rib<app>server, complete the following steps:

1. Log in to the WebLogic administration console GUI (http://<host>:<port>/console) as administrator.
2. Using the left side menu, navigate to Environment > Servers.



3. Click **Lock & Edit**.

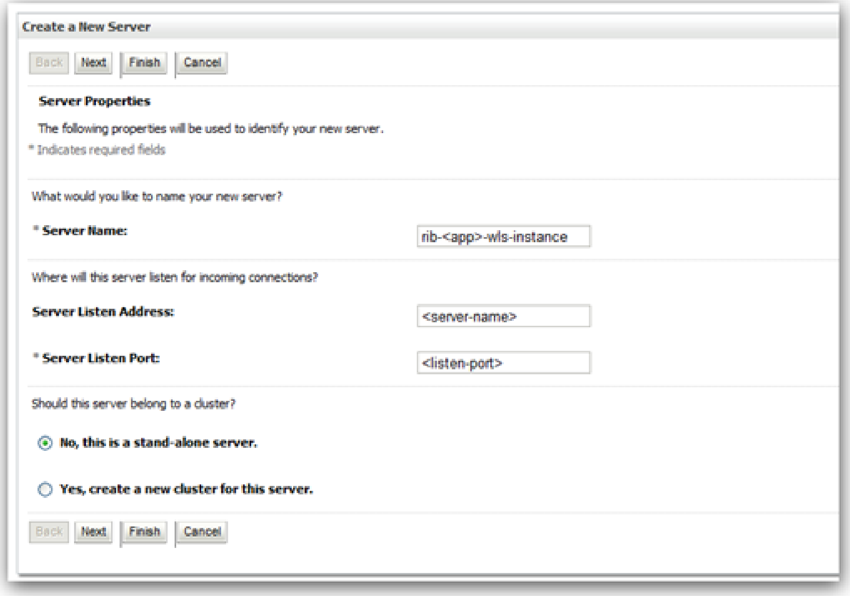


No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

4. Click **New**.
5. Enter the name, port, and listen address of the server instance to be created.

   For example:

   - Server Name: rib-<app>-server
   - Server Listen Address: <server-name>
   - Server Listen Port: <listen-port>



Server Properties

The following properties will be used to identify your new server.

* Indicates required fields

What would you like to name your new server?

* Server Name: rib-<app>-wls-instance

Where will this server listen for incoming connections?

Server Listen Address: <server-name>

* Server Listen Port: <listen-port>

Should this server belong to a cluster?

[✓ No, this is a stand-alone server.]

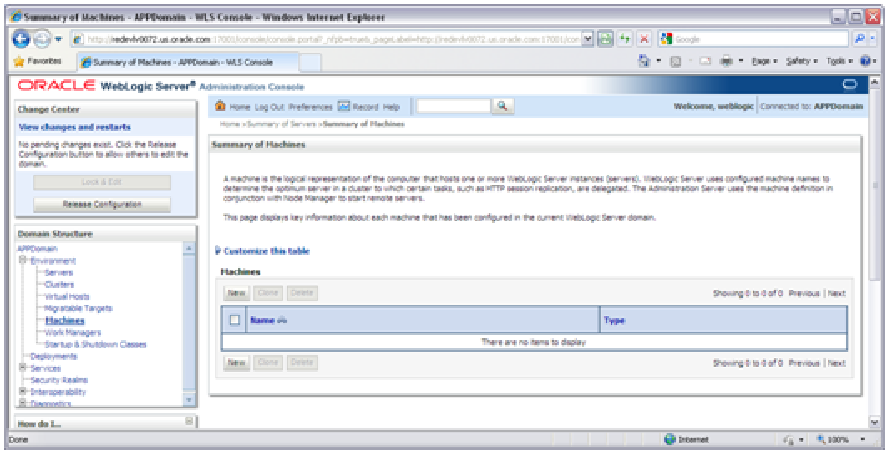
Yes, create a new cluster for this server.

6. Click **Next**. Click **Finish**. Make sure you see this instance listed under Servers

Install NodeManager

Install NodeManager if it was not created during domain install. NodeManager is required so that the managed servers can be started and stopped through the administration console. Only one NodeManager is needed per WebLogic installation.

1. Log in to the administration console.
2. Click **Lock & Edit** and navigate to Environments> Machines.



Summary of Machines

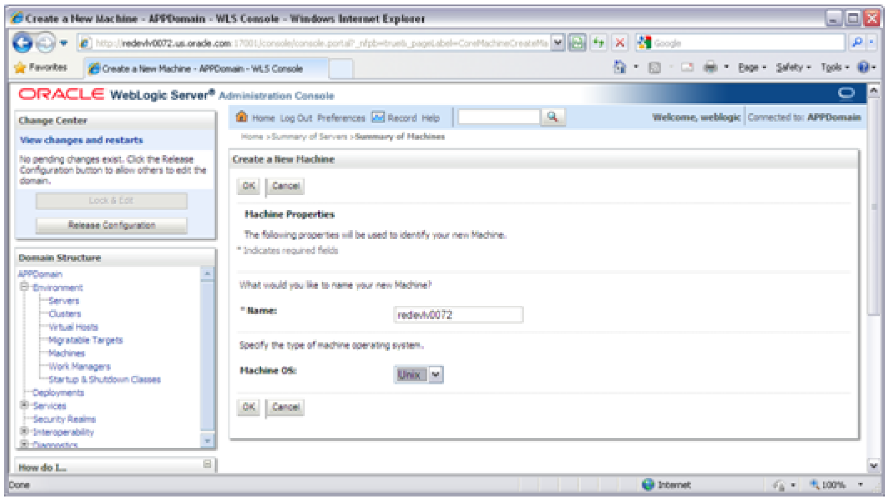
A machine is the logical representation of the computer that hosts one or more WebLogic Server instances (servers). WebLogic Server uses configured machine names to determine the optimum server in a cluster to which certain tasks, such as HTTP session replication, are delegated. The Administration Server uses the machine definition in conjunction with Node Manager to start remote servers.

This page displays key information about each machine that has been configured in the current WebLogic Server domain.

Machines

	Name	Type
There are no items to display		

3. Click **New**.



Create a New Machine

Machine Properties

The following properties will be used to identify your new Machine.

* Indicates required fields

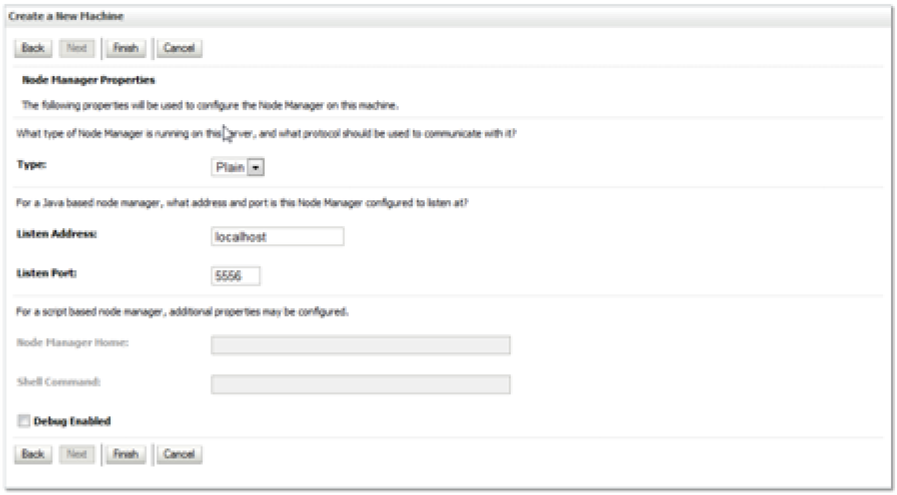
What would you like to name your new Machine?

* Name: redevlv0072

Specify the type of machine operating system.

Machine OS: Unix

4. Set the following variables:
   - Name: Logical machine name
   - Machine OS: UNIX
5. Click **Next**.
6. Update the details below and click the **Finish** button.
   - **Type:** Plain
   - **Listen Address**: For example, ribhost.example.com
   - **Listen Port**: Default port (for example, 5556) or any available port



Create a New Machine

Back | Next | Finish | Cancel

Node Manager Properties

The following properties will be used to configure the Node Manager on this machine.

What type of Node Manager is running on this server, and what protocol should be used to communicate with it?

Type: Plain

For a Java based node manager, what address and port is this Node Manager configured to listen at?

Listen Address: localhost

Listen Port: 5556

For a script based node manager, additional properties may be configured.

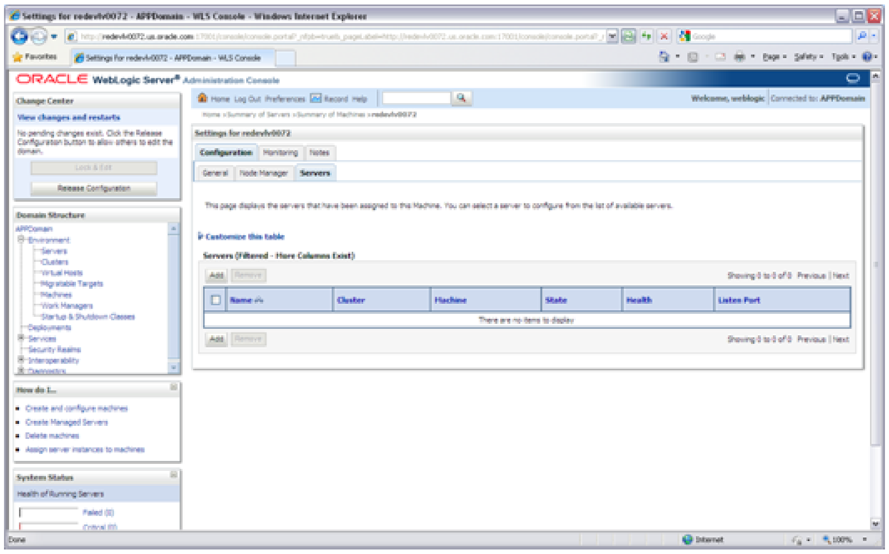
Node Manager Home:

Shell Command:

☐ Debug Enabled

Back | Next | Finish | Cancel

7. Click **Activate Changes**.
8. Click **Lock & Edit**.
9. Navigate to Environments > machines. Click on the machine name and select the Servers tab.



Settings for redevlv0072 - APPDomain - WLS Console - Windows Internet Explorer

ORACLE WebLogic Server® Administration Console

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

APPDomain
- Environment
  - Servers
  - Clusters
  - Virtual Hosts
  - Migratable Targets
  - Machines
  - Work Managers
  - Startup & Shutdown Classes
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...
- Create and configure machines
- Create Managed Servers
- Delete machines
- Assign server instances to machines

System Status

Health of Running Servers

Failed (0)

Critical (0)

Home Log Out Preferences Record Help

Welcome, weblogic | Connected to: APPDomain

Home >Summary of Servers >Summary of Machines >redevlv0072

Settings for redevlv0072

Configuration | Monitoring | Notes

General | Node Manager | Servers

This page displays the servers that have been assigned to this Machine. You can select a server to configure from the list of available servers.

Customize this table

Servers (Filtered - More Columns Exist)

Add | Remove — Showing 0 to 0 of 0 Previous | Next

☐	Name	Cluster	Machine	State	Health	Listen Port
There are no items to display						

Add | Remove — Showing 0 to 0 of 0 Previous | Next

10. Add the managed servers that need to be configured with NodeManager. Save the changes.
11. Click **Add** to repeat for additional servers.
12. Click **Activate Changes**.
13. Start NodeManager from the server using the startNodeManager.sh at $WLS_ HOME/wlserver/server/bin.

Note: To activate changes, the server must be stopped: $WLS_ HOME/user_projects/domains/<RIB_ Domain>/bin/stopManagedWebLogic.sh <rpm>-server ${server_ name}:${server_port}

---

14. Edit the nodemanager.properties file at the following location with the below values:

    $WLS_HOME/user_projects/domains/<RIB_Domain>/nodemanager/nodemanager.properties

    - SecureListener=false
    - StartScriptEnabled=true
    - StartScriptName=startWebLogic.sh

15. NodeManager must be restarted after making changes to the nodemanager.properties file.

    Note: The nodemanager.properties file is created after NodeManager is started for the first time. It will not be available before that point.

Expand the RIB Kernel Distribution

To expand the RIB kernel distribution, complete the following steps:

1. Log in to the UNIX server as the user who will own the RIB development workspace. Create a new directory for the workspace. There should be a minimum of 800 MB of disk space available.
2. Copy the RIB Kernel package (RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar) into the workspace and extract its contents.
3. Extract the jar file using this command:
   $JAVA_HOME/bin/jar -xf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar
4. Change directories to Rib191000ForAll19xxApps/rib-home. This location will be referred to as <RIB_HOME> for the remainder of this chapter.

Configure the rib-<app>-server

To configure the rib-<app>-wls-instance, complete the following steps.

1. Configure the startup script
   a. Take a backup of the script $DOMAIN_HOME/base_domain/bin/startWebLogic.sh
   b. Edit the script $DOMAIN_HOME/base_domain/bin/startWebLogic.sh to add the following attributes. The attributes must be added before the call to start the server.
2. Update $WLS_HOME/<wlserver>/server/lib/weblogic.policy file with the information below.

    Note: If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.

Note: <WEBLOGIC_DOMAIN_HOME> in the following example is the full path of the Weblogic Domain, <managed_server> is the RIB managed server created, and <context_root> correlates to the value entered for the application deployment name/context root of the application that you will supply during installation. Note that the rib-func-artifact-instance does not need to get added to this file. See the example below. There should not be any space between file:<WEBLOGIC_DOMAIN_HOME>.

Note: The path tmp/_WL_user/rib-<app>.ear will not be available before the deployment.

```
grant codeBase "file:
<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/tmp/_WL_user/<context_root>/-"
{
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore.*", "read,write,update,delete";
};
```

An example of the full entry that might be entered is:

```
grant codeBase "file: /u00/webadmin/product/ 19.1.000_RIB/WLS/user_
projects/domains/RIBDomain/servers/rib-rwms-server/tmp/_WL_user/rib-rwms.ear/-"
{
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete"
;
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};
```

Note: Add the path to the patch jars. If any patches are installed into WLS (now or in the future) and this line is not included it could cause the RIB to fail. WLS_HOME refers to the location where Weblogic 12.2.1.4.0 has been installed.

For example:

```
grant codeBase "file:<WLS_HOME>/patch_wls/patch_jars/-" {
permission java.security.AllPermission;
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};

};
```

The AdminServer needs to be bounced after the weblogic.policy file is modified.

3. Start rib-<app>managed server.

   WebLogic managed servers where rib-<app> is deployed can be started two ways.

   Option 1: Run startup scripts through the command line

   1. Log in to the machine where WLS was installed with the operating system user that was used to install the WebLogic Application Server (WLS).
   2. Navigate to DOMAIN_HOME/bin.

      For example:

      $cd product/19.1.000_RIB/WLS/user_projects/domains/RIBDomain/bin
   3. Run the startManagedWebLogic script.

      For example:

      sh startManagedWebLogic.sh rib-rms-wls-instance

   Option 2: Start WebLogic using administration console.

   NodeManager must be running for starting manged server from the console. The nodemanager.properties and startWeblogic.sh must be configured with the properties that have been mentioned above. (See steps 1 and 3 above.)

   > **Note:** RIB applications cannot be deployed from the administration console. They must be run through the command line.

   1. Log in to the WebLogic administration console GUI (http://<host>:<port>/console) as administrator
   2. Using the right side menu, navigate to Environment > Servers
   3. Click rib-<app> managed server.
   4. Click the **Control** tab.
   5. Select the managed server instance that must be started.
   6. Click **Start**.
   7. Repeat this procedure for all rib-<app>managed servers.

3

Database Installation Tasks

There are several tasks that must be performed for RIB and verified in the participating applications.

Oracle Database Schemas

Each Oracle Retail Application has an associated set of RIB Artifacts that must be installed as part of the RIB integration (for example, the RIB Hospital Tables, CLOB API libraries, and Oracle Objects).

- Ensure that these have been installed appropriately, per the individual applications.
- Ensure that the TAFR Hospital user and objects exist.
- Ensure that the RIB user has appropriate access and permissions.

RIB and Multi byte Deployments

If RIB is deployed in an environment where multi byte characters are used in the message data, there are considerations that must be understood. Improper database setup can lead to error messages indicating the inability to insert values that are too long.

Note: See the section, "Pre-Implementation Considerations for Multi byte Deployments," in the *Oracle Retail Integration Bus Implementation Guide*.

These considerations are beyond the scope of the RIB documentation and should be discussed with the site Database Administration team prior to installation.

Verify that Correct RIB Hospital Database Objects are Installed in the Retail Application's Schema

Every rib-<app> application needs a database schema that contains the RIB Hospital tables. Externalizing the RIB Hospital tables from the application database schema is supported.

There are two options:

- rib-<app> can use the respective application database schema to host the RIB hospital tables.

- rib-<app> can have a separate database or a separate schema to host the RIB hospital tables.

Note: The RIB Hospital schema must not be shared across retail applications. Each rib-<app> should have its own RIB hospital tables in both of the options listed above.

These RIB Hospital tables are not installed as part of the RIB installation, but they are installed as part of the Retail applications database schema installation. Verify that the four RIB Hospital tables are already installed in the respective database schema.

Note: See Appendix B, "RIB Installation Checklists."

The database schema for all retail applications must have the database objects defined in the RIB delivered kernel SQL script called 1_KERNEL_CREATE_OBJECTS.SQL.

Note: The 1_KERNEL_CREATE_OBJECTS.SQL script is available in the rib-private-kernel-database-library-<version>.jar file. The rib-private-kernel-database-library-<version>.jar can be found in the rib-home directory structure (/Rib191000ForAll19xxApps/rib-home/integration-lib).

Note: See the section, "RIB App Builder rib-home," in the *Oracle Retail Integration Bus Operations Guide*.

Because these database objects should have already been installed as part of the retail application's installation process, at this point just verify that the four hospital tables and the sequence exist in each application's database schema. Make sure that they have the correct columns to match this release of the RIB.

It is strongly recommended that all applications have a separate RIB Hospital and that they be logically and operationally associated with that application.

Note: See "RIB Software Life Cycle" in the *Oracle Retail Integration Bus Implementation Guide*.

Verify that Database XA Resources are Configured for RIB

RIB uses two phase commit transaction protocol (XA) to maintain consistency between the RIB Hospital database, application database and the JMS server. The Oracle database XA resources must be configured in order to participate in XA transaction. Check to see that the XA scripts have been run on the database to make it XA transaction aware. The initxa.sql script needs to be run before XA transactions will work. These are usually installed by default in 12c (12.x).

Verify that Correct RIB Functional Artifacts Database Objects Are Installed in PL/SQL Applications Database Schema

This section applies to PL/SQL application only, RMS, ORFM, and RWMS.

There are two ways through which PL/SQL applications exchange payload data with RIB:

- Oracle Objects payloads
- CLOB xml parsing and building library

RMS and ORFM use both mechanisms, whereas RWMS uses only Oracle Objects to communicate with RIB.

1. Verify that the RMS, ORFM, and RWMS database schema have the RIB delivered Oracle Objects installed. Oracle Objects are not installed as part of RIB installation. They are installed as part of the retail application database schema installation.
2. Verify that the PL/SQL retail application database schema already has the database objects defined equivalent to the ones defined in the RIB delivered script called InstallAndCompileAllRibOracleObjects.sql.

   Note: See the *Oracle Retail Integration Bus Operations Guide*. InstallAndCompileAllRibOracleObjects.sql script is packaged as a part of rib-public-payload-database-xml-library.zip and is available under <*RIB-HOME*>/application-assembly-home/rib-func-artifacts/.

3. Verify that RMS (not RWMS) database schema has the RIB CLOB XML parsing and building library code installed. These CLOB XML libraries are not installed as part of RIB installation. They are installed as part of the retail application database schema installation. With the recent changes related to decoupling of RICS from RMFCS, there won't be any CLOB xml libraries in RMS. This step is needed for older versions of RMS.
4. Verify that the RMS retail application database schema has all the database objects defined equivalent to the ones defined in the RIB delivered script called 1_CLOB_CREATE_OBJECTS.SQL. This step is not required for 19.1.000 version of RMS.

   Note: See the *Oracle Retail Integration Bus Operations Guide*. 1_CLOB_CREATE_OBJECTS.SQL script is packaged as a part of rib-public-payload-database-xml-library.zip and is available under <*RIB-HOME*>/application-assembly-home/rib-func-artifacts/.

5. Update the RIB functional artifact URL in the RMS table RIB_OPTIONS to point to the location where rib-func-artifact.war will be deployed.

   XML_SCHEMA_BASE_URL=
   http://<hostname>:<port>/rib-func-artifact/payload/xsd

   Where:

   - hostname is the host name where rib-func-artifact.war will be deployed.
   - port is the http port of the WebLogic server where rib-func-artifact.war will be deployed.

Create RIB TAFR RIB Hospital

For RIB, there is a separate RIB Hospital for the rib-tafr application.

1. Create a database user for the rib application rib-tafr.

2. Make sure that the TAFR Hospital user has the proper database permission.

Example TAFR User Create SQL:

CREATE USER <*tafr hosp user*>

IDENTIFIED BY <*tafr hosp password*>

DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP";

GRANT "CONNECT" TO <*tafr hosp user*>;

GRANT "RESOURCE" TO <*tafr hosp user*>;

ALTER USER <*tafr hosp user*>

QUOTA UNLIMITED ON USERS;

The rib-tafr application's database user must have the RIB Hospital tables. To create the RIB Hospital tables, run the 1_KERNEL_CREATE_OBJECTS.SQL script.

Note: The 1_KERNEL_CREATE_OBJECTS.SQL script is available in the rib-private-kernel-database-library-<version>.jar file. The rib-private-kernel-database-library-<version>.jar can be found in the rib-home/integration-lib/ directory structure. Extract the script and provide it to the Database Administrator (DBA) to create the required database objects.

Prepare Oracle AQ JMS Provider

Oracle Streams AQ is the JMS provider that RIB uses for asynchronous communication. It requires Oracle Database Enterprise Edition.

It is strongly recommended that the Oracle Database instance configured as the JMS provider is not shared with any other applications and not be on the same host (physical or logical) with any other applications. The steps included here are those needed to prepare for the installation, there are many architectural issues and operational parameters that must be considered before the installation. These are covered in other RIB documents.

RIB and AQ JMS Database Processes

The RIB's use of the AQ JMS should be understood, and the Oracle Database instance that is configured as the AQ JMS must be configured to support the number of server side user processes needed for the RIB adapters that will be installed and configured in each deployment environment. The number of JMS AQ processes depends on the RIB configuration.

Note: See the section, "Pre-Implementation Considerations - JMS Server Considerations," in the *Oracle Retail Integration Bus Implementation Guide*.

Note: See the section, "Deployment Architectures," in the *Oracle Retail Integration Bus Implementation Guide*. See also the "JMS Provider Management" and "The RIB on AQ JMS" sections in the *Oracle Retail Integration Bus Operations Guide*.

Create the RIB AQ JMS user with the appropriate access and permissions to the Oracle Streams AQ packages. This user must have at least the following database permissions:

- CONNECT
- RESOURCE
- CREATE SESSION
- EXECUTE ON SYS.DBMS_AQ
- EXECUTE ON SYS.DBMS_AQADM
- EXECUTE ON SYS.DBMS_AQIN
- EXECUTE ON SYS.DBMS_AQJMS

Example SQL:

CREATE USER <*rib aq user*> IDENTIFIED BY <*rib aq password*>

DEFAULT TABLESPACE "RETAIL_DATA"

TEMPORARY TABLESPACE "TEMP";

GRANT "CONNECT" TO <*rib aq user*>;

GRANT "RESOURCE" TO <*rib aq user*>;

GRANT CREATE SESSION TO <*rib aq user*>;

GRANT EXECUTE ON "SYS"."DBMS_AQ" TO <*rib aq user*>;

GRANT EXECUTE ON "SYS"."DBMS_AQADM" TO <*rib aq user*>;

GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO <*rib aq user*>;

GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO <*rib aq user*>;

GRANT "AQ_ADMINISTRATOR_ROLE" TO <*rib aq user*>;

ALTER USER <*rib aq user*>

QUOTA UNLIMITED ON RETAIL_DATA;

Note: See also:

Oracle® Database Administrator's Guide 12c Release 1 (12.1.0.2)

Oracle® Streams Advance Queuing User's Guide and Reference 12c Release 1 (12.1.0.2)

---

4 RIB Cloud Support

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

Applications can invoke the new web services to send and receive messages to/from other applications via RIB using SOAP or REST style of webservices. Client applications must use credentials of a user in group ribAdminGroup to call the publisher web service. For consuming messages (using the injector service) applications must create a user in group IntegrationGroup 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.

Following example shows how to configure a hybrid scenario in which SIM is on-premise and RIB is on-cloud.

Configuration for SOAP webservices call:

- In the rib-deployment-env-info.xml file, configure SIM app 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 a copy of the rib-app section for rib-sim (as soap-app). Edit the properties as per rib-sim. Here's an example:

  ```
  <rib-app id="rib-sim" type="soap-app">
           <deploy-in refid="rib-sim-wls1" />
           <rib-admin-gui>

  <web-app-url>http://ribhost.example.com:19206/rib-sim-appserver-gui/index.jsp</
  web-app-url>
              <web-app-user-alias>rib-sim_rib-admin-gui_
  admin-user-name-alias</web-app-user-alias>
              <web-app-user-alias>rib-sim_rib-admin-gui_
  operator-user-name-alias</web-app-user-alias>
              <web-app-user-alias>rib-sim_rib-admin-gui_
  monitor-user-name-alias</web-app-user-alias>
           </rib-admin-gui>
           <error-hospital-database>

  <hosp-url>jdbc:oracle:thin:@simhost.example.com:1521/pdborcl</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>http://simhost.example.com:9001/ApplicationMessageInjectorBean/InjectorSer
vice</url>
              <!-- Supported security policy names =policyC (default) OR
policyA'
               <ws-policy-name>policyC</ws-policy-name>
               <user-alias>rib-sim_ws_security_user-name-alias</user-alias>
            </end-point>
         </app>
      </rib-app>
```

- Make sure the rib-sim_ws_security_user-name-alias user is a member of the IntegrationGroup in the SIM WebLogic domain. Make sure the SIM services are up and running and can be called via SOAP UI using the credentials that will be entered during RIB compilation.
- Compile and deploy RIB.

The following example shows how to configure a hybrid scenario in which ROB (rest application) is on-premise and RIB is on-cloud.

Configuration for REST webservices call:

- In the rib-deployment-env-info.xml file, ROB is configured to be of type "rest-app". Under <app-in-scope-for-integration> ROB is set as rest-app:

  <app id="rob" type="rest-app" />

- Edit the existing rib-app section for rib-rob (as rest-app). Edit the properties as per rib-rob configuration. Here's an example:

```
<rib-app id="rib-rob" type="rest-app">
         <deploy-in refid="rib-rob-wls1" />
         <rib-admin-gui>

<web-app-url>http://ribhost.example.com:19109/rib-rob-appserver-gui/index.jsp</
web-app-url>
            <web-app-user-alias>rib-rob_rib-admin-gui_
admin-user-name-alias</web-app-user-alias>
            <web-app-user-alias>rib-rob_rib-admin-gui_
operator-user-name-alias</web-app-user-alias>
            <web-app-user-alias>rib-rob_rib-admin-gui_
monitor-user-name-alias</web-app-user-alias>
         </rib-admin-gui>
         <error-hospital-database>

<hosp-url>jdbc:oracle:thin:@robhost.example.com:1521/pdborcl</hosp-url>
            <hosp-user-alias>rib-rob_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="rob" type="rest-app">
                <end-point>

<url>http://robhost.example.com:9001/rib-injector-services-web/resources/inject
or/inject</url>
                   <!-- Supported security policy names =policyC (default) OR
policyA OR policyB -->
                   <ws-policy-name>policyC</ws-policy-name>
                   <user-alias>rib-rob_ws_security_user-name-alias</user-alias>
                </end-point>
             </app>
          </rib-app>
```

- Make rib-rob_ws_security_user-name-alias user is a member of the IntegrationGroup in the ROB WebLogic domain. Make sure the ROB rest services are up and running and can be called via SOAP UI using the credentials that will be entered during RIB compilation
- Compile and deploy RIB.

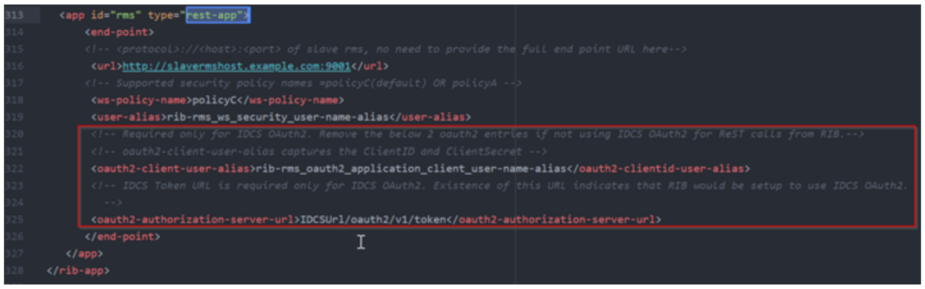
IDCS OAuth2 in RIB

RIB uses IDCS OAuth2 for authentication of ReST calls both inbound and out-bound(**publisher/injector restful services**). Although, the basicAuth will also be supported to maintain backward compatibility and to support on-prem requirements, primary authentication mechanism in the cloud would be OAuth2 using IDCS authenticator. Out of the box configuration would expect OAuth2 to be used.

Installer/ rib-home changes to support IDCS OAuth2

1. **rib-deployment-env-info.xml** has OPTIONAL elements to capture the oauth2 configuration.

   Below is the snapshot of new rib-deployment-env-info.xml for app type "rest-app" such as rib-rms, rib-rob.



```
  <app id="rms" type="rest-app">
     <end-point>
     <!-- <protocol>://<host>:<port> of slave rms, no need to provide the full end point URL here-->
      <url>http://slavermshost.example.com:9001</url>
     <!-- Supported security policy names =policyC(default) OR policyA -->
      <ws-policy-name>policyC</ws-policy-name>
      <user-alias>rib-rms_ws_security_user-name-alias</user-alias>
      <!-- Required only for IDCS OAuth2. Remove the below 2 oauth2 entries if not using IDCS OAuth2 for ReST calls from RIB.-->
      <!-- oauth2-client-user-alias captures the ClientID and ClientSecret -->
      <oauth2-client-user-alias>rib-rms_oauth2_application_client_user-name-alias</oauth2-clientid-user-alias>
      <!-- IDCS Token URL is required only for IDCS OAuth2. Existence of this URL indicates that RIB would be setup to use IDCS OAuth2.
        -->
      <oauth2-authorization-server-url>IDCSUrl/oauth2/v1/token</oauth2-authorization-server-url>
     </end-point>
  </app>
</rib-app>
```

2. If "oauth2-authorization-server-url" exists , this will be treated as OAuth2 setup and only then rib-app-builder prompt user to key in the ClientId/ClientSecret mapped to the ali-as <app-name>_oauth2_application_client_user-name-alias.
3. IDCS token URL will be written to the rib-system.properties at compile time with this format

oauth2.authorization.server.token.url=https://idcs-c3bb6417e4904132beebfa6c440 fcce4.identity.c9dev1.oc9qadev.com/oauth2/v1/token

This URL will be available for post install update using admin GUI. Steps - RIB Admin GUI > Manage Configurations > System Options

At run time, both plsql/java rest injector services (rib-rms master-plsql-app and rib-rob) will use the above property to make the call to IDCS to get the token and use in subsequent inject calls.

5

RIB High Availability Installation Instructions

This chapter provides instruction on RIB HA setup.

Assumptions

- There are two RIB App Server nodes. Nodes must be pre-identified as primary(active) and secondary(passive) node.
- RIB domain is available in both nodes.
- In this chapter, node1 is considered as ACTIVE node. We will assume node1 has WebLogic1.
- In this chapter, node2 is considered as the PASSIVE node. We will assume node2 has WebLogic2.
- Assumption is all credentials (App-DB, EH, AQ JMS etc) are same for rib in node1 and node2.

How to install RIB HA in the ACTIVE node from rib-home

Install using command line tool following the below instructions.

Install Using the RIB App Builder Command Line Tools

1. Download and extract the RibKernel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar.

   e.g tar xvf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar
2. Download RibFuncArtifact<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar and put it in rib-home/download-home/rib-func-artifacts directory.Do not extract the tar file. This will be done by the check-version-and-unpack tool.
3. Download all the RibPak<RIB_{MAJOR|MINOR}_ VERSION>For<RETAIL_APP_NAME><RETAIL_APP_VERSION>_eng_ ga.tar and put it in the rib-home/download-home/all-rib-apps directory. Do not extract the tar file. This will be done by the check-version-and-unpack tool.
4. For Linux and Solaris OS only. Set the JAVA_HOME environment variable. The JAVA_HOME must be set to a JDK1.8.0+64bit with latest security updates, within the1.8codeline.64bit.
5. Run the rib-home/download-home/bin/check-version-and-un pack.sh script from rib-home/download-home/bin directory.

6. Make a copy of rib-home/deployment-home/conf/rib-deployment-env-info.xml file, name the file to rib-home/deployment-home/conf/rib-deployment-env-info.xml.node1
7. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml.node1 file to specify the deployment environment information of weblogic1 in node1.
8. Run rib-home/application-assembly-home/bin/select-deployment-options.sh script from rib-home/application-assembly-home/bin directory. This script is to specify which environment to which you want to deploy.

   example: ./select-deployment-options.sh -deployment-file-extension node1 -set-adapter-status adapter-up
9. Compile: Run the rib-home/application-assembly-home/bin/rib-app-compiler.sh script with setup-security-credential from rib-home/application-assembly-home/bin directory.

   example: ./rib-app-compiler.sh -setup-security-credential
10. Deploy: Execute the rib-home/deployment-home/bin/rib-app-deployer.sh script with the appropriate command line parameter.

    ```
    rib-app-deployer.sh -prepare-jms
    ```
11. Verify: Once the rib- app is deployed, open the rib-admin-gui from a web browser:<http or https://>host:port/rib-<app>-admin-gui

    Eg- http://ribrmshost:7001/rib-rms-admin-gui

All the adapters should be up and running.

How to install RIB HA in the PASSIVE node from rib-home

No need of creating rib-home for each box, same rib-home should be used for deploying in node2. No need of downloading tarballs and running check-version-and-unpack script. Follow these steps:

1. Make a copy of rib-home/deployment-home/conf/rib-deployment-env-info.xml file, name the file to rib-home/deployment-home/conf/rib-deployment-env-info.xml.node2
2. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml.node2 file to specify the deployment environment information of weblogic2 in node2.
3. Run rib-home/application-assembly-home/bin/select-deployment-options.sh script from rib-home/application-assembly-home/bin directory. This script is to specify which environment you want to deploy to

   example: ./select-deployment-options.sh -deployment-file-extension node2 -set-adapter-status adapter-down

   Note: To deploy on secondary node ie. node2 (Passive) use adapter-down, so that all adapters would be down.

4. Run the rib-home/application-assembly-home/bin/rib-app-compiler.sh script without setup-security-credential from rib-home/application-assembly-home/bin directory. assumption is all credentials are same between node1 and node2 rib.

   example: ./rib-app-compiler.sh

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

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

6. Verify: Once the rib- app is deployed, open the rib-admin-gui from a web browser:<http or https://>host:port/rib-<app>-admin-gui

   Eg- http://ribrmshost:7001/rib-rms-admin-gui

All the adapters should be down. Bring down the secondary rib domain after the deployment. Only primary must be running.

Disaster Recovery Steps

When to switch for Active to the Passive Node

- Switch to the passive node only when the active node/weblogic has become corrupt that it can no longer be brought up without rebuilding it. Rebuilding the node/weblogic will take days and we will miss the SLA.
- When primary domain can't be recovered easily, then bring down the domain, make sure that there are no processes still running. Only after this bring up the secondary node.
- Then from the admin GUI bring up the adapters.

How to make node2 ACTIVE from its PASSIVE State

Note: This also means making node1 PASSIVE from its ACTIVE state.

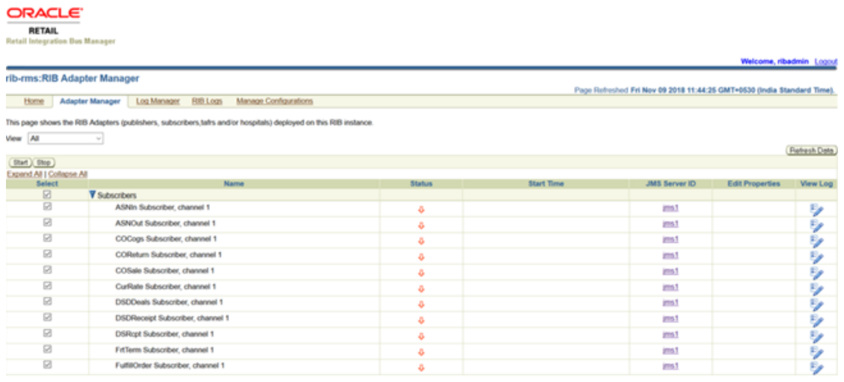
Follow these steps:

1. Bring down the rib domain in WebLogic1 in node1. Shutdown all managed servers and Admin server, make sure there are no processes still running.
2. Bring up the rib domain in Weblogic2 in node2
3. Open the rib-admin-gui from a web browser:

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

   (eg.-http://ribrmshost:37005/rib-rms-admin-gui)

   Then from the rib admin gui bring up the adapters by selecting the checkboxes and clicking the Start button (see screen shot for reference)



rib-rms:RIB Adapter Manager

Page Refreshed Fri Nov 09 2018 11:44:25 GMT+0530 (India Standard Time).

This page shows the RIB Adapters (publishers, subscribers,tafrs and/or hospitals) deployed on this RIB instance.

View: All

Select	Name	Status	Start Time	JMS Server ID	Edit Properties	View Log
✓	Subscribers					
✓	ASNIn Subscriber, channel 1	⇩		jms1		✎
✓	ASNOut Subscriber, channel 1	⇩		jms1		✎
✓	COCogs Subscriber, channel 1	⇩		jms1		✎
✓	COReturn Subscriber, channel 1	⇩		jms1		✎
✓	COSale Subscriber, channel 1	⇩		jms1		✎
✓	CurRate Subscriber, channel 1	⇩		jms1		✎
✓	DSDDeals Subscriber, channel 1	⇩		jms1		✎
✓	DSDReceipt Subscriber, channel 1	⇩		jms1		✎
✓	DSRcpt Subscriber, channel 1	⇩		jms1		✎
✓	FrtTerm Subscriber, channel 1	⇩		jms1		✎
✓	FulfillOrder Subscriber, channel 1	⇩		jms1		✎

6 Run the RIB Application Builder Tools

This chapter provides instructions for running the RIB Application Builder Tools.

> **Note:** If there is an existing WebLogic installation on the server, you must upgrade to WebLogic 12.2.1.4.0. All middleware components associated with WebLogic server should be upgraded to 12.2.1.4.0.
>
> Back up the weblogic.policy file ($WLS_HOME/wlserver/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.

RIB Application Builder Tasks

The RIB application builder can be used to perform any of the tasks below. For a new installation, all tasks are recommended.

- Run the Preparation Phase to unpack files, prepare the workspace, and perform preinstallation verifications.
- Run the Assembly Phase to build the EAR and WAR files for the rib-<app> applications.
- Configure the Advanced Queuing JMS topics for RIB.
- Run the Deployment Phase to deploy the EAR and WAR files to the application servers.
- Restart the WebLogic server.

For more information about the Preparation, Assembly and Deployment Phases, see the *Oracle Retail Integration Bus Operations Guide*.

How to Run the RIB Application Builder

To start the RIB application installation, do the following.

1. Undeploy all old rib-apps and completely remove them from the AdminServer upload directory as follows.
   a. Shut down all the rib-* servers.
   b. Delete all rib-* apps from the deployments menu in WebLogic.

Note: Be sure to remove ONLY the rib-* apps and no others. If other applications are removed, their deployments will break.

c. Remove them from the upload directory if they exist:

```
cd [RIB_DOMAIN]/servers/AdminServer/upload
rm -rf rib-*
```

d. Start the rib-servers up again before starting the installation.

2. Expand the RIB Kernel distribution as described in Expand the RIB Kernel Distribution.
3. Download the RIB Functional Artifacts distribution (RibFuncArtifact19.1.000ForAll19.1.000Apps_eng_ga.tar), and copy it into the <RIB_HOME>/download-home/rib-func-artifacts directory. Do not untar the file.
4. Download the tar file distributions for each rib-<app> application (RibPak19.1.000For<app>19.1.000_eng_ga.tar) that you will install. Copy the files into the <RIB_HOME>/download-home/all-rib-apps directory. Do not untar the files.
5. Download the RIB Diagnostic and Monitoring Tools (RDMT) package (Rdmt19.1.000ForAll19.x.xApps_eng_ga.tar) and untar it into the <RIB_ HOME>/tools-home directory. Several files will be placed under the rdmt directory when you untar the package. Run the <RIB_ HOME>/tools-home/rdmt/configbuilder.sh script after the RIB installation.
6. For multiple JMS servers only: If your RIB installation includes more than one JMS server, you must complete the additional preinstallation steps in the section, Preinstallation Steps for Multiple JMS Server Setup.
7. Set the JAVA_HOME environment variable. The JAVA_HOME must be set to a JDK 1.8.0+ 64 bit with latest security updates, within the 1.8 code line. 64 bit. For Linux and Solaris OS only.
8. Make sure that all WebLogic instances that you intend to deploy to are currently running.
9. Give execute permissions to rib-home:

   For example, chmod -R 700 rib-home.
10. Change directories to the <RIB_HOME>/download-home/bin directory and run check-version-and-unpack script to extract all the tarball contents.
11. Change directories to the <RIB_HOME>/application-assembly-home/bin directory and rRun the rib-app-compiler.sh script. Provide the needed inputs , compilation prompts for.
12. Change directories to the <RIB_HOME>/deployment-home/bin directory and run the rib-app-deployer.sh script.

    Note: See Appendix A, "Appendix: RIB Application Builder Tools" for more de-tails.

13. Restart the rib-<app>-server. During the installation process a shared library is created that contains the JDBC Driver update. For PL/SQL applications, it is necessary to bounce the WebLogic managed server instance.

14. Manually run the RDMT at this time to verify the installation.

Check the Log Files to Ensure Installation was Successful

To check log files, do the following.

1. Check the log files in <RIB_HOME>/deployment-home/log to ensure that all RIB applications deployed successfully.
2. If errors are encountered, verify that the installer inputs were correct.
3. If all installer inputs were correct, it may be necessary to rerun the rib app builder with the existing <RIB_ HOME>/deployment-home/conf/rib-deployment-env-info.xml file. Running the tools multiple times usually resolves any extraneous errors.

Preinstallation Steps for Multiple JMS Server Setup

Note: Using multiple JMS servers allows for the isolation of flows for performance and operational QoS. For more information, see "JMS Provider Management" section in the *Oracle Retail Integration Bus Operations Guide*.

If your RIB installation will include multiple JMS servers, additional steps are required before you can run the rib application builder.

Note: Do not follow these steps if you are using only one JMS server.

1. Determine the family that must be configured for multiple JMS.
2. Locate the rib-integration-flows.xml inside the rib-func-artifacts.war.

   Examine the rib-integration-flows.xml to identify all the RIB applications participating in the integration flows that must be configured with multiple JMS.
3. Ensure that a new AQ JMS database server (not a schema) is set up. For information see "Prepare Oracle AQ JMS Provider" in this guide.
4. Ensure that any additional AQ JMS are not in the same database server. Each new AQ JMS requires a new database server.
5. Add JMS servers by updating rib-deployment-env-info.xml.
6. In the rib-home, modify the appropriate files for each of the rib-<apps> that participate in the integration flow. Point the adapters to the right JMS server. The following applies to this step:
   - rib-<app>-adapters.xml
   - rib-<app>-adapter-resources.properties

   Note: For more information on this step, see the *Oracle Retail Integration Bus Operations Guide*.
7. Once Step 6 is finished, do the following to complete preinstallation activities:

- Compiles all rib apps ($RIB_HOME/application-assembly-home/bin/rib-app-compiler.sh).
- Runs prepare-jms for the newly-created JMS server ($RIB_HOME/deployment-home/bin/rib-app-deployer.sh -prepare-jms<jms2>). This step configures additional JMS servers.
- Deploys ($RIB_HOME/deployment-home/bin/rib-app-deployer.sh rib-<app>).

8. Restart the WebLogic managed servers.

Preinstallation Steps for Configuring rib-ext

Rib-Ext is a new component added for external apps/3rd party integration with RIB. The RIB-EXT app organizes all publishers and subscribers adapters required by an external system and makes it easy to integrate from thirdparty systems. The list of publisher and subscriber adapters are defined by the customer's implementation team.

Following are the steps required to enable customer's specific integration flows:

1. The install tarball(RibPak19.1.000ForExt19.1.000_eng_ga) includes a version of the configuration file with all adapters enable as well as a copy of the file based on the customer's configuration..
   - rib-ext-adapters.xml (undesired flows can be commented out)
   - rib-ext-adapters_gap.xml (flows have been pre-determined)
2. If required, copy rib-ext-adapters_gap.xml to rib-ext-adapters.xml.
3. Proceed with rib installation.

Preinstallation Steps for Enabling Dynamic Adapter Selection Feature for rib-<app>

This is new feature introduced in this release wherein user can pick a set of adapters at runtime from list of all available adapters for rib-<app>. This feature is enabled by default for rib-ext as it has a big list of adapters.

The following steps are required for enabling the dynamic adapter selection feature in all other rib-<app>:

1. The install tarball(RibPak19.1.000For<app>19.1.000_eng_ga) includes a properties file.
   - rib-<app>.properties (eg- rib-sim.properties)
2. Edit rib-<app>.properties file to add below flag

```
enableDynamicAdapterInstanceSelection=true
```

3. Proceed with RIB installation.

Run RDMT to Verify the Installation

The RIB Diagnostic and Monitoring Tools (RDMT) should be used at this time to verify the RIB installation. See "Diagnostic and Monitoring Tools" in the *Oracle Retail Integration Bus Operations Guide* for how to configure and use the RDMT tools.

7 RIB-RWMS Hybrid Cloud Installation Instructions

This chapter describes the steps that must be completed for rib-rwms hybrid cloud environment setup.

Setup rib-rwms Hybrid Cloud Environment

Prerequisite

- Hybrid Cloud set-up involves 2 parts installation, one each for master (cloud) and slave components (on-prem).
- This document provides specific instructions for rib-rwms installation for hybrid-cloud, for detailed instructions on RIB installation refer to the Chapter 6, "Run the RIB Application Builder Tools".
- Main difference between the regular RIB-RWMS installation and the hybrid cloud RIB-RWMS is the data sources created during the installation needs to point to a set of available database schema for AQ, Error Hospital and the Application DB, as listed below.

Database Schemas

Identify the DB users you will need for the data-sources. You will

need to input these during the compilation step.

- Master rib-rwms app needs a valid AQ schema, a valid Error Hospital schema. Since the master-app does not have access to the on-premises RWMS application schema, we will point the app-db datasource to the Error hospital schema. AQ schema is the usual schema which all rib-apps are connected to including master rib-rwms as well.
- Slave rib-rwms app will be deployed in the on-premise close to RWMS. Slave rib-rwms app needs a valid on-prem RWMS application schema. For the Error Hospital and the AQ data-sources can be pointed to the same RWMS app schema since the slave-app does not know about the on-cloud DB.

Table 7–1 Master and Slave rib-rwms Data-Sources Details

App Type	AQ Schema rib-rwms-jms1-ojmsmanaged-datasource	EH Schema rib-rwms-hosp-managed-datasource	App DB Schema rib-rwms-managed-datasource
Master rib-rwms (master-plsql-app)	A valid AQ . example: rib_aq_schema	A valid EH schema. example: eh_schema_master_rib_rwms	same as EH eh_schema_master_rib_rwms
Slave rib-rwms (slave-plsql-app)	example: rwms_app_schema (RWMS app schema on-prem.)	example: rwms_app_schema (RWMS app schema on-prem.)	example: rwms_app_schema (RWMS app schema on-prem.)

Part 1: RIB_RWMS Master Side Configuration (On Cloud)

Follow the below instructions for installation.

Install Using the RIB App Builder Command Line Tools

1. Download and extract the RibKernel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar.

   e.g tar xvf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar

2. RibFuncArtifact<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar and put it in rib-home/download-home/rib-func-artifacts directory.

   Do not extract the tar file. This will be done by the check-version-and-unpack tool.

3. Download all the RibPak<RIB_{MAJOR|MINOR}_ VERSION>For<RETAIL_APP_NAME><RETAIL_APP_VERSION>_eng_ ga.tar and put it in the rib-home/download-home/all-rib-apps directory.

   Do not extract the tar file. This will be done by the check-version-and-unpack tool.

4. For Linux and Solaris OS only. Set the JAVA_HOME environment variable. The JAVA_HOME must be set to a JDK1.8.0+64bit with latest security updates, within the1.8codeline.64bit.

5. Run the rib-home/download-home/bin/check-version-and-un pack.sh script from rib-home/download-home/bin directory.

6. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml file to specify the deployment environment information.

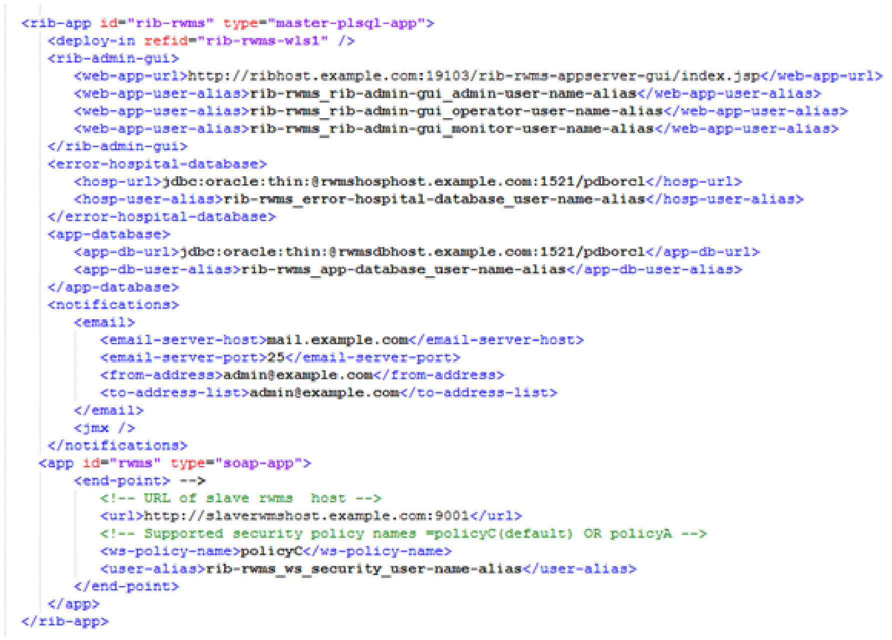
   Under <app-in-scope-for-integration> change RWMS from plsql-app to master-plsql-app:

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

7. Replace the existing rib-app with copy of commented hybrid cloud installation section for rib-rwms.

   Example (this is also available on deployment XML as commented snippet):

Figure 7–1 Commented Hybrid Cloud Installation XML Snippet



```
<rib-app id="rib-rwms" type="master-plsql-app">
   <deploy-in refid="rib-rwms-wls1" />
   <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:19103/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:@rwmshosphost.example.com:1521/pdborcl</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:@rwmsdbhost.example.com:1521/pdborcl</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 slave rwms  host -->
         <url>http://slaverwmshost.example.com:9001</url>
         <!-- Supported security policy names =policyC(default) OR policyA -->
         <ws-policy-name>policyC</ws-policy-name>
         <user-alias>rib-rwms_ws_security_user-name-alias</user-alias>
      </end-point>
   </app>
</rib-app>
```

Note: RWMS retail app type is "soap-app". RIB app (rib-rwms) type is "master-plsql-app".

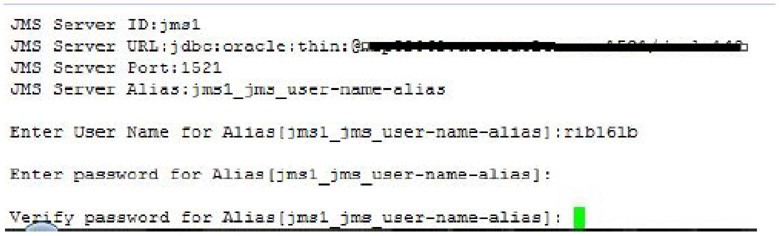
---

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

   example: ./rib-app-compiler.sh -setup-security-credential

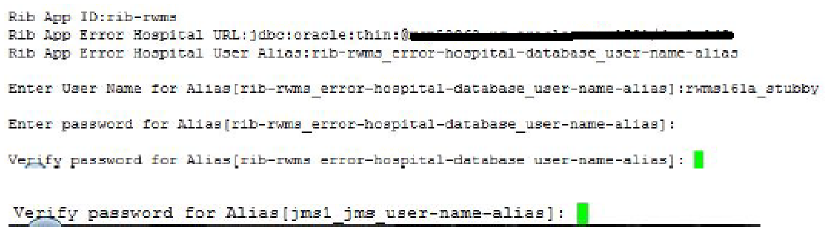
   For rib-rwms give the same schema details for Error Hospital (EH) and app-DB as follows:

Figure 7–2 AQ Details



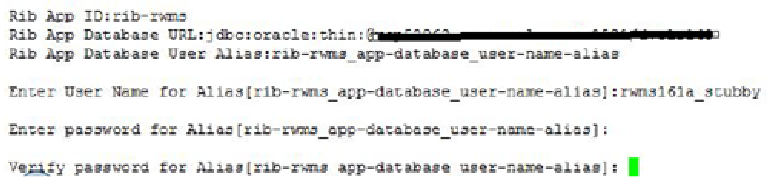
```
JMS Server ID:jms1
JMS Server URL:jdbc:oracle:thin:@[redacted]
JMS Server Port:1521
JMS Server Alias:jms1_jms_user-name-alias

Enter User Name for Alias[jms1_jms_user-name-alias]:rib161b

Enter password for Alias[jms1_jms_user-name-alias]:

Verify password for Alias[jms1_jms_user-name-alias]:
```

Figure 7–3 Error Hospital Details



```
Rib App ID:rib-rwms
Rib App Error Hospital URL:jdbc:oracle:thin:@[redacted]
Rib App Error Hospital User Alias:rib-rwms_error-hospital-database_user-name-alias

Enter User Name for Alias[rib-rwms_error-hospital-database_user-name-alias]:rwms161a_stubby

Enter password for Alias[rib-rwms_error-hospital-database_user-name-alias]:

Verify password for Alias[rib-rwms error-hospital-database user-name-alias]: 

Verify password for Alias[jms1_jms_user-name-alias]: 
```

Figure 7–4 App-DB Details



```
Rib App ID:rib-rwms
Rib App Database URL:jdbc:oracle:thin:@[redacted]
Rib App Database User Alias:rib-rwms_app-database_user-name-alias

Enter User Name for Alias[rib-rwms_app-database_user-name-alias]:rwms161a_stubby

Enter password for Alias[rib-rwms_app-database_user-name-alias]:

Verify password for Alias[rib-rwms app-database user-name-alias]: 
```

9. There are two new webservices added in RIB that allow the master to communicate with slaves.

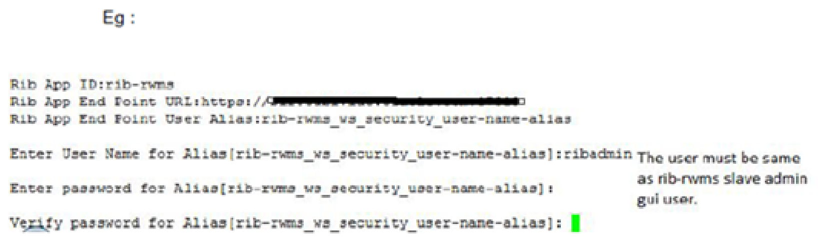
   Publication – RemotePlsqlPublisherComponentServiceBeanService

   Subscription – PlsqlApplicationMessageInjectorServiceBeanService

   RIB supports policyA and poilicyC service provider. By default webservices are secured with policyC (http-or-https-username-token).

   The compilation step prompts you to enter the user/password for the alias " rib-rwms_ws_security_user-name-alias" used for service calls. The credential should be the same as for "rib-rwms_rib-admin-gui_admin-user-name-alias" on the slave side.

Figure 7–5 Compilation Example



```
Eg :

Rib App ID:rib-rwms
Rib App End Point URL:https://[redacted]
Rib App End Point User Alias:rib-rwms_ws_security_user-name-alias

Enter User Name for Alias[rib-rwms_ws_security_user-name-alias]:ribadmin The user must be same
                                                                          as rib-rwms slave admin
Enter password for Alias[rib-rwms_ws_security_user-name-alias]:          gui user.

Verify password for Alias[rib-rwms_ws_security_user-name-alias]: 
```

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

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

Note: <app> must be rwms.

---

11. **Verify:** Once the rib-rwms app is deployed, open the rib-admin-gui from a web browser:

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

Part 2: RIB_RWMS Slave Side Configuration (On Premise)

Install Using the RIB App Builder Command Line Tools

1. Download RIB kernel For RMWS-slave-app (hybrid-cloud) distribution.

   RibKernel19.1.000ForRwmsSlave19.x.xApps_eng_ga.jar

2. Extract contents of jar file.

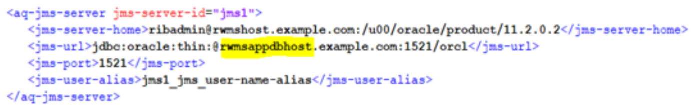
3. Open rib-deployment-env-info.xml found inside ./rib-rwms-slave-home/deployment-home/conf.

4. Edit this file to specify your deployment environment information.

   - Make sure the following entries are present in the <app-in-scope-for-integration> section:

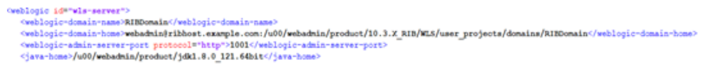
     ```
     <app id="rwms" type=" slave-plsql-app" />
     ```

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



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

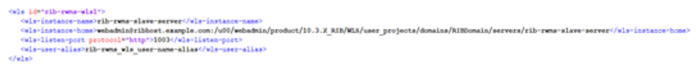
   - Update RIB domain details in weblogic-application-servers section. For example:



```
<weblogic id="wls-server">
    <weblogic-domain-name>RIBDomain</weblogic-domain-name>
    <weblogic-domain-home>webadmin@ribhost.example.com:/u00/webadmin/product/10.3.X_RIB/WLS/user_projects/domains/RIBDomain</weblogic-domain-home>
    <weblogic-admin-server-port protocol="http">1001</weblogic-admin-server-port>
    <java-home>/u00/webadmin/product/jdk1.8.0_121.64bit</java-home>
```

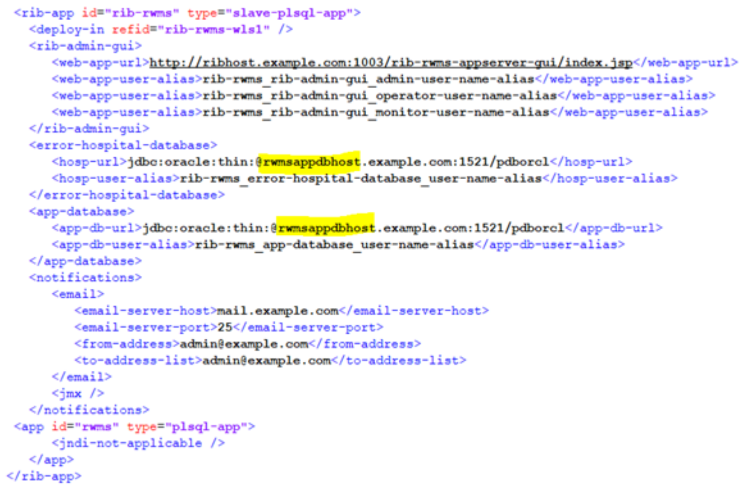
   - Skip updating the rib-func-artifact-server details. Rib-func-artifact deployment is not required for slave.

   - Update RIB-RWMS slave server details. For example:



```
<wls id="rib-rwms-wls1">
    <wls-instance-name>rib-rwms-slave-server</wls-instance-name>
    <wls-instance-home>webadmin@ribhost.example.com:/u00/webadmin/product/10.3.X_RIB/WLS/user_projects/domains/RIBDomain/servers/rib-rwms-slave-server</wls-instance-home>
    <wls-listen-port protocol="http">1003</wls-listen-port>
    <wls-user-alias>rib-rwms_wls_user-name-alias</wls-user-alias>
</wls>
```

   - Make sure datasource url (host, port n service) entries are updated in the rib-app section of rib-rwms slave:



```
<rib-app id="rib-rwms" type="slave-plsql-app">
  <deploy-in refid="rib-rwms-wls1" />
  <rib-admin-gui>
     <web-app-url>http://ribhost.example.com:1003/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:@rwmsappdbhost.example.com:1521/pdborcl</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:@rwmsappdbhost.example.com:1521/pdborcl</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: As the slave app deploys on premise, it will not have access to AQ JMS and Error hospital. Therefore, all the datasources must connect to the RWMS app schema.

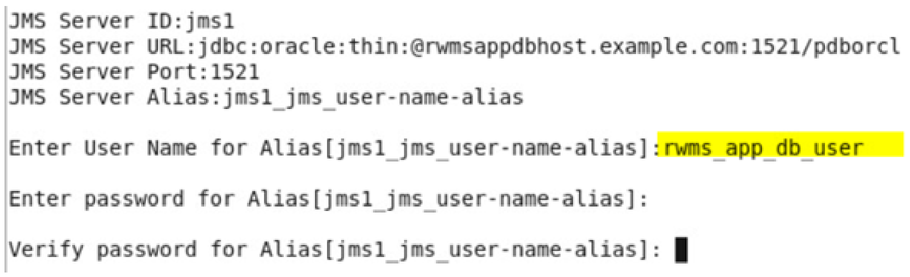
---

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

   Example: ./rib-app-compiler.sh -setup-security-credential

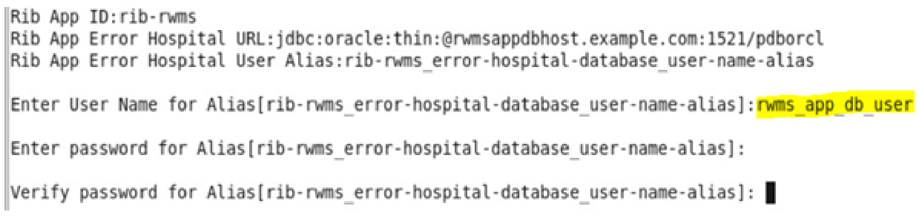
Note: When the slave app deploys on-premise, it will not have access to AQ JMS and Error hospital. Therefore, both datasources must connect to the RWMS app schema.

Figure 7–6 AQ Details



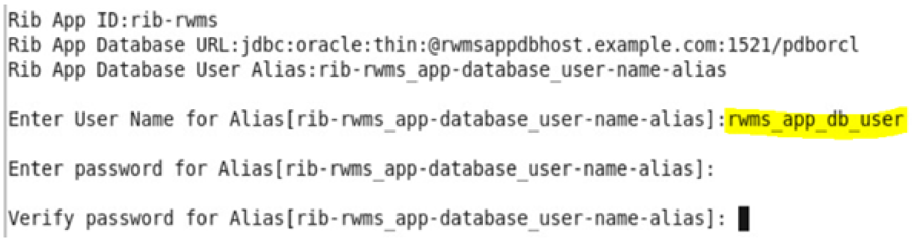
```
JMS Server ID:jms1
JMS Server URL:jdbc:oracle:thin:@rwmsappdbhost.example.com:1521/pdborcl
JMS Server Port:1521
JMS Server Alias:jms1_jms_user-name-alias

Enter User Name for Alias[jms1_jms_user-name-alias]:rwms_app_db_user

Enter password for Alias[jms1_jms_user-name-alias]:

Verify password for Alias[jms1_jms_user-name-alias]: █
```

Figure 7–7 EH Details



```
Rib App ID:rib-rwms
Rib App Error Hospital URL:jdbc:oracle:thin:@rwmsappdbhost.example.com:1521/pdborcl
Rib App Error Hospital User Alias:rib-rwms_error-hospital-database_user-name-alias

Enter User Name for Alias[rib-rwms_error-hospital-database_user-name-alias]:rwms_app_db_user

Enter password for Alias[rib-rwms_error-hospital-database_user-name-alias]:

Verify password for Alias[rib-rwms_error-hospital-database_user-name-alias]: █
```

Figure 7–8 App-DB Details



```
Rib App ID:rib-rwms
Rib App Database URL:jdbc:oracle:thin:@rwmsappdbhost.example.com:1521/pdborcl
Rib App Database User Alias:rib-rwms_app-database_user-name-alias

Enter User Name for Alias[rib-rwms_app-database_user-name-alias]:rwms_app_db_user

Enter password for Alias[rib-rwms_app-database_user-name-alias]:

Verify password for Alias[rib-rwms_app-database_user-name-alias]:
```

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

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

   rib-func-artifact deployment is not required.
   ```
   Verify: Once the rib-rwms slave app is deployed, open the rib-admin-gui from a web browser using the credentials provided during compilation:

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

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

   Example:

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

   https:// ribhost.example.com:17010/PlsqlApplicationMessageInjectorServiceBea
   n/PlsqlApplicationMessageInjectorServiceBeanService?WSDL
   ```

   Sample script to create rib-rwms master AQ schema:

   ```
   CREATE USER <rib aq user> IDENTIFIED BY <rib aq password>
   DEFAULT TABLESPACE "RETAIL_DATA" TEMPORARY TABLESPACE "TEMP";
   GRANT "CONNECT" TO <rib aq user> ;
   GRANT "RESOURCE" TO <rib aq user> ;
   GRANT CREATE SESSION TO <rib aq user> ;
   GRANT EXECUTE ON "SYS"."DBMS_AQ" TO <rib aq user> ;
   GRANT EXECUTE ON "SYS"."DBMS_AQADM" TO <rib aq user> ;
   GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO <rib aq user>;
   GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO <rib aq user>;
   GRANT "AQ_ADMINISTRATOR_ROLE" TO <rib aq user>;
   ALTER USER <rib aq user>
   QUOTA UNLIMITED ON RETAIL_DA
   ```

8 Prepackaged RIB-RWMS Application PAKs

This release contains four PAKs for rib-rwms i.e. rib-rwms, rib-rwms2, rib-rwms3, rib-rwms4. This is to support multiple instances of warehouse. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml to add instances in scope. Under <app-in-scope-for-integration> add the desired instances in scope.

Example-

```
<app-in-scope-for-integration>
      <app id="rms" type="plsql-app" />
      <!-- The below configuration is for rmws as a master-plsql-app, use this
when RIB is on cloud and RWMS is on-prem-->
      <app id="rwms" type="master-plsql-app" />
      <app id="rwms2" type="master-plsql-app" />
      <app id="rwms3" type="master-plsql-app" />
      <app id="rwms4" type="master-plsql-app" />
   </app-in-scope-for-integration>
```

Compilation and deployment steps remains same as other application PAKs.

9 RIB-RMS Hybrid Cloud Installation Instructions

After decoupling of Retail Integration Cloud Service (RICS) from Retail Merchandise Cloud Service (RMFCS), RICS is going to be standalone offering. RICS would still be required for RMFCS, but not vice versa. RICS supports both RMS on premise and RMS on cloud. RIB is already supported in cloud so for enabling the integration of RMS (on-premise) with all other retail applications which are in hybrid cloud environment, RIB follows master/slave approach. Slave resides close to on-premises RMS while master is on-cloud. Communication between master and slave is through web service calls. RIB-RMS master invokes the web services exposed by slave RIB-RMS to send/receive messages to/from other applications on cloud via RIB.

Setup rib-rms Hybrid Cloud Environment

Prerequisites

- Hybrid Cloud set-up involves 2 parts installation, one each for master (cloud) and slave components (on-premises).
- This document provides specific instructions for rib-rms installation for hybrid-cloud, for detailed instructions on RIB installation refer to the Chapter 6, "Run the RIB Application Builder Tools".
- Main difference between the regular RIB-RMS installation and the hybrid cloud RIB-RMS is the data sources created during the installation needs to point to a set of available database schema for AQ, Error Hospital and the Application DB, as listed below.

Database Schemas

Identify the DB users you will need for the data-sources. You will need to input these during the compilation step.

- Master rib-rms app needs a valid AQ schema, a valid Error Hospital schema. Since the master-app does not have access to the on-premises RMS application schema, we will point the app-db datasource to the Error hospital schema. AQ schema is the usual schema which all rib-apps are connected to including master rib-rms as well.
- Slave rib-rms app will be deployed in the on-premise close to RMS. Slave rib-rms app needs a valid on-prem RMS application schema. For the Error Hospital and the AQ data-sources can be pointed to the same RMS app schema since the slave-app does not know about the on-cloud DB.

App Type	AQ Schema rib-rms-jms1-ojmsmanaged-datasource	EH Schema rib-rms-hosp-managed-datasource	App DB Schema rib-rms-managed-datasource
Master rib-rms (master-plsql-app)	A valid AQ . example: rib_aq_schema	A valid EH schema. example: eh_schema_master_rib_rms	same as EH eh_schema_master_rib_rms
Slave rib-rms (slave-plsql-app)	example: rms_app_schema (RMS app schema on-prem.)	example: rms_app_schema (RMS app schema on-prem.)	example: rms_app_schema (RMS app schema on-prem.)

Part 1: RIB_RMS Master Side Configuration (On Cloud)

Follow the below instructions.

Install Using the RIB App Builder Command LineTools

1. Download and extract the RibKernel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar.

   e.g tar xvf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar

2. Download RibFuncArtifact<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar and put it in rib-home/download-home/rib-func-artifacts directory.

   Do not extract the tar file. This will be done by the check-version-and-unpack tool.

3. Download all the RibPak<RIB_{MAJOR|MINOR}_ VERSION>For<RETAIL_APP_NAME><RETAIL_APP_VERSION>_eng_ ga.tar and put it in the rib-home/download-home/all-rib-apps directory.

   Do not extract the tar file. This will be done by the check-version-and-unpack tool.

4. For Linux and Solaris OS only. Set the JAVA_HOME environment variable. The JAVA_HOME must be set to a JDK1.8.0+64bit with latest security updates, within the1.8codeline.64bit.

5. Run the rib-home/download-home/bin/check-version-and-un pack.sh script from rib-home/download-home/bin directory.

6. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml file to specify the deployment environment information.

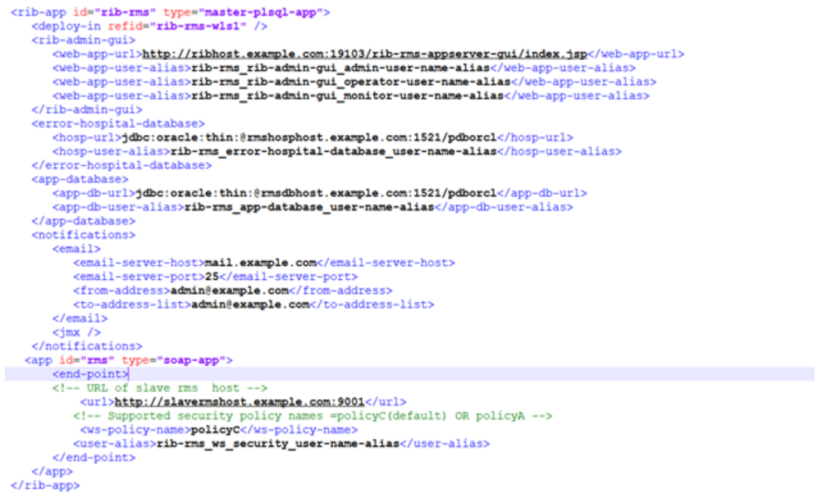
   Under <app-in-scope-for-integration> make sure RMS is set to master-plsql-app:

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

7. Following is the hybrid cloud installation section for rib-rms.

   Example (this is also available on deployment XML):

Figure 9–1 Hybrid Cloud Installation XML Snippet



```
<rib-app id="rib-rms" type="master-plsql-app">
   <deploy-in refid="rib-rms-wls1" />
   <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:19103/rib-rms-appserver-gui/index.jsp</web-app-url>
      <web-app-user-alias>rib-rms_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rms_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rms_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
   <error-hospital-database>
      <hosp-url>jdbc:oracle:thin:@rmshosphost.example.com:1521/pdborcl</hosp-url>
      <hosp-user-alias>rib-rms_error-hospital-database_user-name-alias</hosp-user-alias>
   </error-hospital-database>
   <app-database>
      <app-db-url>jdbc:oracle:thin:@rmsdbhost.example.com:1521/pdborcl</app-db-url>
      <app-db-user-alias>rib-rms_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="rms" type="soap-app">
      <end-point>
      <!-- URL of slave rms  host -->
          <url>http://slavermshost.example.com:9001</url>
         <!-- Supported security policy names =policyC(default) OR policyA -->
          <ws-policy-name>policyC</ws-policy-name>
         <user-alias>rib-rms_ws_security_user-name-alias</user-alias>
      </end-point>
   </app>
</rib-app>
```

Note: RMS retail app type is "rest-app". RIB app (rib-rms) type is "master-plsql-app". RIB will use IDCS OAuth2 for authentication of ReST calls both inbound and outbound(publisher/injector restful services). Although, the basicAuth will also be supported to maintain backward compatibility and to support on-prem requirements, primary authentication mechanism in the cloud would be OAuth2 using IDCS authenticator. Out of the box configuration would expect OAuth2 to be used.

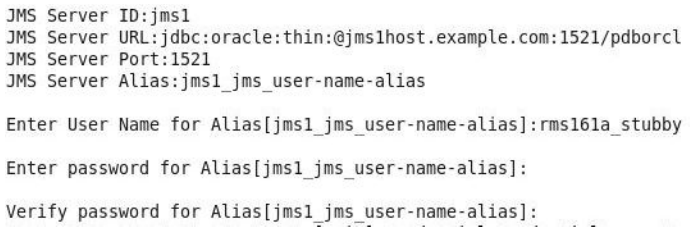
---

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

   Example: ./rib-app-compiler.sh -setup-security-credential

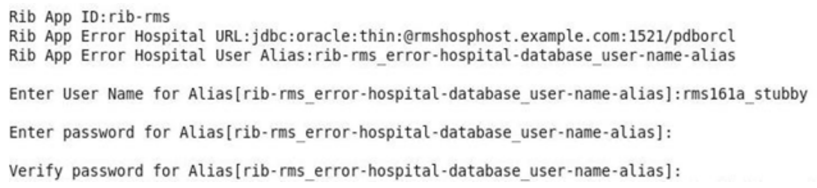
   For rib-rms give the same schema details for Error Hospital (EH) and app-DB

Figure 9–2 Compilation Example - AQ Details



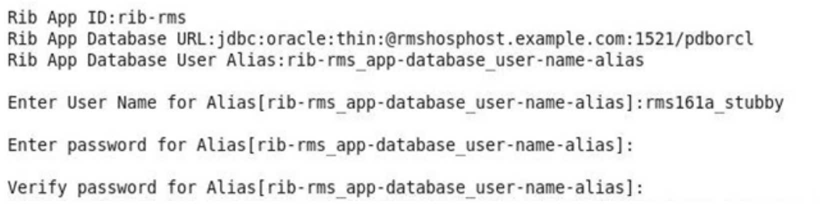
```
JMS Server ID:jms1
JMS Server URL:jdbc:oracle:thin:@jms1host.example.com:1521/pdborcl
JMS Server Port:1521
JMS Server Alias:jms1_jms_user-name-alias

Enter User Name for Alias[jms1_jms_user-name-alias]:rms161a_stubby

Enter password for Alias[jms1_jms_user-name-alias]:

Verify password for Alias[jms1_jms_user-name-alias]:
```

Figure 9–3 Compilation Example - EH Details



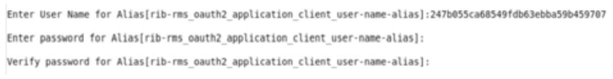
```
Rib App ID:rib-rms
Rib App Error Hospital URL:jdbc:oracle:thin:@rmshosphost.example.com:1521/pdborcl
Rib App Error Hospital User Alias:rib-rms_error-hospital-database_user-name-alias

Enter User Name for Alias[rib-rms_error-hospital-database_user-name-alias]:rms161a_stubby

Enter password for Alias[rib-rms_error-hospital-database_user-name-alias]:

Verify password for Alias[rib-rms_error-hospital-database_user-name-alias]:
```

Figure 9–4 Compilation Example - App DB Details



```
Rib App ID:rib-rms
Rib App Database URL:jdbc:oracle:thin:@rmshosphost.example.com:1521/pdborcl
Rib App Database User Alias:rib-rms_app-database_user-name-alias

Enter User Name for Alias[rib-rms_app-database_user-name-alias]:rms161a_stubby

Enter password for Alias[rib-rms_app-database_user-name-alias]:

Verify password for Alias[rib-rms_app-database_user-name-alias]:
```

Figure 9–5 Compilation Example - Oauth2 Details



```
Enter User Name for Alias[rib-rms_oauth2_application_client_user-name-alias]:247b055ca68549fdb63ebba59b459707
Enter password for Alias[rib-rms_oauth2_application_client_user-name-alias]:
Verify password for Alias[rib-rms_oauth2_application_client_user-name-alias]:
```

9. There are two new webservices added in RIB that allow the master to communicate with slaves.

   Publication - RemotePlsqlPublisherComponentServiceBeanService

   Subscription - PlsqlApplicationMessageInjectorServiceBeanService

   RIB supports policyA and poilicyC service provider. By default webservices are secured with policyC (http-or-https-username-token).

   The compilation step prompts you to enter the user/password for the alias " rib-rms_ws_security_user-name-alias" used for service calls. The credential should be the same as for "rib-rms_rib-admin-gui_admin-user-name-alias" on the slave side.

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

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

    Note: <app> must be rms.

    ---

11. Verify: Once the rib-rms app is deployed, open the rib-admin-gui from a web browser:

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

Part 2: RIB_RMS Slave Side Configuration (On Premises)

Note: This setup is not applicable to fully on-premise topology where RMS and RIB both are on-premise.

Install Using the RIB App Builder Command LineTools

1. Download RIB kernel For RMS-slave-app (hybrid-cloud) distribution.

   RibKernel19.1.000ForRmsSlave19.x.xApps_eng_ga.jar

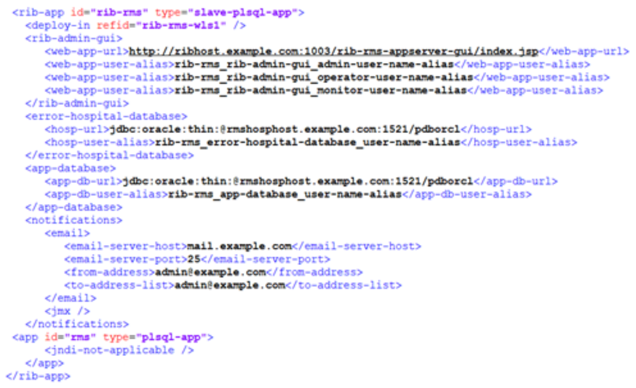
2. Extract contents of jar file.

3. Open rib-deployment-env-info.xml found inside ./rib-rms-slave-home/deployment-home/conf.
4. Edit this file to specify your deployment environment information.

   Make sure the following entries are present in the <app-in-scope-for-integration> section:

   `<app id="rms" type=" slave-plsql-app" />`
   Make sure following entries are present in the rib-app section of rib-rms slave:

Figure 9–6 rib-app Section of rib-rms Slave



```
<rib-app id="rib-rms" type="slave-plsql-app">
  <deploy-in refid="rib-rms-wls1" />
  <rib-admin-gui>
     <web-app-url>http://ribhost.example.com:1003/rib-rms-appserver-gui/index.jsp</web-app-url>
     <web-app-user-alias>rib-rms_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
     <web-app-user-alias>rib-rms_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
     <web-app-user-alias>rib-rms_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
  </rib-admin-gui>
  <error-hospital-database>
     <hosp-url>jdbc:oracle:thin:@rmshosphost.example.com:1521/pdborcl</hosp-url>
     <hosp-user-alias>rib-rms_error-hospital-database_user-name-alias</hosp-user-alias>
  </error-hospital-database>
  <app-database>
     <app-db-url>jdbc:oracle:thin:@rmshosphost.example.com:1521/pdborcl</app-db-url>
     <app-db-user-alias>rib-rms_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="rms" type="plsql-app">
     <jndi-not-applicable />
  </app>
</rib-app>
```

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

   When the slave app deploys on-premise, it will not have access to AQ JMS and Error hospital. Therefore, both datasources must connect to the RMS app schema.
6. Deploy: Execute the rib-home/deployment-home/bin/rib-app-deployer.sh script with the appropriate command line parameter.

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

   rib-func-artifact deployment is not required.
   ```
   Deployed slave app will be in warning state but it will not impact any functionality.
7. Verify: Make sure the Publication and Subscription WS are available to use.

   Example:

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

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

   Sample script to create rib-rms master AQ schema:

   ```
   CREATE USER <rib aq user> IDENTIFIED BY <rib aq password> DEFAULT TABLESPACE
   "RETAIL_DATA" TEMPORARY TABLESPACE "TEMP"; GRANT "CONNECT" TO <rib aq user> ;
   GRANT "RESOURCE" TO <rib aq user> ; GRANT CREATE SESSION TO <rib aq user> ;
   GRANT EXECUTE ON "SYS"."DBMS_AQ" TO <rib aq user> ; GRANT EXECUTE ON
   "SYS"."DBMS_AQADM" TO <rib aq user> ; GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO
   <rib aq user>; GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO <rib aq user>; GRANT "AQ_
   ```

```
ADMINISTRATOR_ROLE" TO <rib aq user>;
ALTER USER <rib aq user>
QUOTA UNLIMITED ON RETAIL_DATA;
```

10

Post-Installation Tasks

This chapter describes the steps that must be completed after installation.

Secure Filesystem

After the RIB installation process is finished, run the following commands from inside rib-home directory.

1. chmod -R go-rwx

   This command revokes read, write, and execute permissions from the group and other users. Only the current user will have read, write, and execute permissions.

2. find . -name "*.sh" -exec chmod u+rwx {} \;

   This command grants to the current user read, write, and execute permission for all executable scripts.

3. The .profile for the OS user for rib-home should include umask 077 set.

4. Go to the $DOMAIN_HOME/servers/$SERVER_NAME folder, which is the managed server home where RIB application is installed, and run this command:

   chmod -R go-rwx .

   This command revokes read, write, and execute permissions from the group and other users. Only the current user will have read, write, and execute permissions.

Oracle Application Tasks

Verify that correct URL's to the RIB Functional Artifacts are configured in the Java EE Applications.

- Functional Artifact URL
- JNDI URL

RDMT Installation

The RIB Diagnostic and Monitoring Tool (RDMT) kit is a collection of command line tools, written in Unix shell script along with supporting Java classes packaged in jar files. There are various tools to address these areas:

- Installation Verification (reports)
- Operations (scanning and monitoring)
- Production (scanning and quick triage)

- Test and Support (scanning and fine grain control)
- AQ JMS support and tools

Installation Steps

Complete the following steps.

1. The RDMT Java support classes require Java 8.0. Installation will perform a check and fail if the path is not correct. Before you begin the installation process, verify that your Java version is correct.
2. Determine the location for installation. The recommended location is to put it in rib-home/tools-home directory. There is an empty rdmt subdirectory already there. This is only a placeholder. However, RDMT can be installed under any user in any directory.
3. Download the tar file (Rdmt19.1.000ForAll19.x.xApps_eng_ga.tar) and extract it (tar xvf Rdmt19.1.000ForAll19.x.xApps_eng_ga.tar).
4. cd to the RDMT directory and execute the configbuilder.sh script supplied with the toolkit (configbuilder.sh).
5. Once executed, it checks if the RDMT has been extracted under rib-home/tools-home directory. If so, it fetches all the necessary configuration information from rib-deployment-env-info.xml present under rib-home/deployment-home/conf directory and it automatically completes the RDMT installation.

   If RDMT was extracted under some other directory with rib-home present on the same server, it prompts for the rib-home path. Provide the same and it fetches all the necessary configuration information from rib-deployment-env-info.xml present under specified rib-home/deployment-home/conf directory and it automatically completes the RDMT installation.

   If rdmt was extracted in a remote server, it prompts for RIB configuration values during setup. The installation script prompts for the configuration settings needed to run the tools in the toolkit.
6. The installation automatically configures for all the rib-<apps> depending upon the applications in scope as defined in rib-deployment-env-info.xml. In case of remote installation, select Yes to configure additional rib<-apps>. It is recommended that you configure all the rib-apps that have been installed in the RIB Installation.
7. Run the RibConfigReport. This report runs a series of tests to validate the RIB components installed.

Information to Gather for Installation in Remote Server

The following are the necessary directory parameters.

Parameters	Setting
RDMT Home Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt/
RDMTLOGS Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt /RDMTLOGS
Temp Files Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt /RDMTLOGS/tmp

Parameters	Setting
RIB App Builder rib-home Directory	/Rib191000ForAll19xxApps/rib-home

The following are parameters for the JMS provider.

Parameter	Setting
AQ JMS User ID	*<rib aq user>*
AQ JMS Password	*<rib aq password>*
JMS Connection URL	jdbc:oracle:thin:@host-name:port:sid

The following are WLS parameters for JMX functions:

Parameter	Setting
WLS/JMX Host	ribhost
WLS Admin Port	8001
WLS Protocol	http or https
WLS Instance Name	rib-rms-server
WLS Instance Port	8002
WLS Protocol	http or https
WLS App Name	rib-rms
WLS User Name	*<weblogic user>*
WLS Password	*<weblogic password>*

The following are parameters for each hospital (RMS, RWMS, SIM, and others).

Parameter	Setting
User Name	*<rms user>*
Password	*<rms password>*
Database URL	jdbc:oracle:thin:@host-name:port:sid

RIB Hospital Administration Tool

This swing based RIB Hospital Administration tool is replaced by a Web application. See Oracle Retail Integration Bus Hospital Administration documentation for end user instructions and details about .ear file deployment in WebLogic Application Server 12.2.1.

11 Retail Integration Console Installation Tasks

Retail Integration Console (RIC) is a visualization tool for Retail Integration. It provides full visibility to the Oracle Retail Integration System in a unified view within the business context of the Oracle Retail applications.

Prerequisites

- RIB must be deployed.
- JMS-Console must be deployed from rib-home/tools-home/.
- rib-home must be accessible to ric-home, in other words both must share the file system.

Deployment Steps

Perform the following procedure to deploy RIC:

1. Download RicKernel19.1.000ForAll19.x.xApps_eng_ga.zip to a location (for example - RIC-APP-BUILDER) on your computer which has your rib-home.
2. Edit the configuration file ric-deployment-env-info.json inside ric-home/conf/ folder.
3. Modify the MiddlewareServerDef and IntegrationProduct with information that is specific to your environment.
   - Set the value of ribEnable property in the configuration file to true.
   - Set the value of ribHome property in the configuration file to point to your rib-home.
   - Set the value of RicAppServer fields to point to the environment where you want to deploy RIC.
   - Set the value of rsbEnable property in the configuration file to false for RIC in RIB only mode.
   - Set the value of bdiEnable property in the configuration file to true.
4. Go to the ric-home/bin/ folder, run the compiler to update the RIC ear as follows:

   ```
   $ sh ric-app-compiler.sh -setup-credentials
   ```
   When prompted by the compiler, enter the user name and password for the Weblogic server and RIC admin user, the RIC admin user will be used to log in RIC.
5. Run the deployer script to deploy RIC and create the user and group on your weblogic server from the same folder as follows:

```
$ sh ric-app-deployer.sh -deploy-ric-app
```

Note: See the section, "Configuration and Deployment," in the *Oracle Retail Integration Console Guide* before attempting installation.

12 Integration Gateway Services Installation Tasks

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

The IGS provides an integration infrastructure for external (third party) system 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. So it should be installed only if there is a requirement to do so.

Prerequisites

The RIB Integration Gateway Service (IGS) component requires Oracle WebLogic Server 12.2.1.4 and Java 8.

Before installation, read the *RIB Implementation Guide* for the considerations and planning steps needed for the RIB IGS deployment to WebLogic Server. Also make sure $JAVA_HOME is pointing to Java 8.

Prepare Oracle WebLogic Server

The installation and base configuration of the Oracle WebLogic Server is beyond the scope of this document. Work with the Oracle WebLogic Server administration team to determine the physical and logical placement of the RIB IGS component within the WebLogic Server deployment.

Create the RIB IGS WebLogic Managed Server

This section describes the process of preparing the Oracle WebLogic Server to install the igs-service.

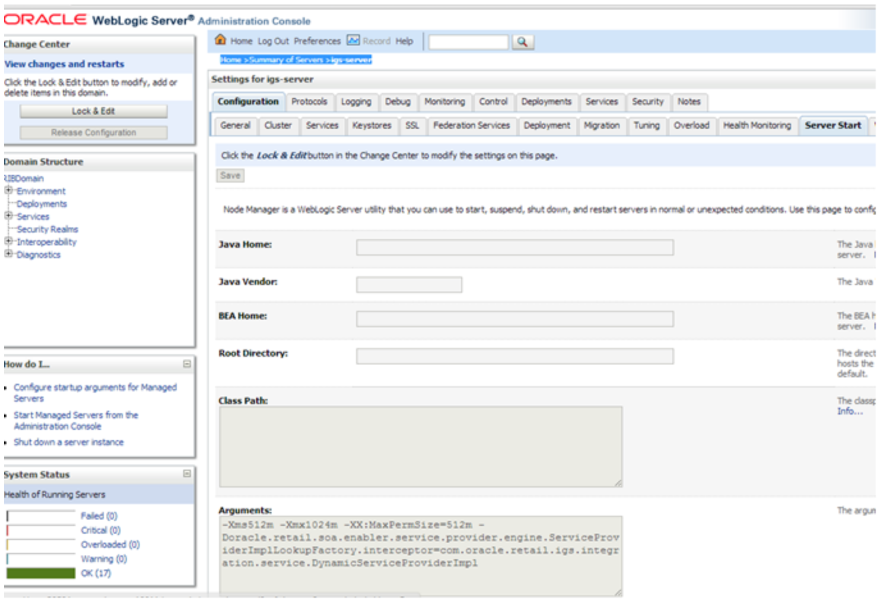
1. IGS ear file should be deployed to a separate managed server.
2. When naming the WebLogic instance, it is recommended (but not required) that the .ear file name is used (without the extension), along with underscore, _server.

   For example, if the .ear file name is igs-service.ear, the instance name would be igs-service_server.
3. Add the server start argument for IGS managed server. This can be done from the WLS console, or in the startWebLogic.sh and startManagedWebLogic.sh scripts.

a. To edit the scripts, add the following to startWeblogic.sh and startManagedWebLogic.sh under $DOMAIN_HOME/bin:

```
JAVA_
OPTIONS="-Doracle.retail.soa.enabler.service.provider.engine.ServiceProvide
rImplLookupFactory.interceptor=com.oracle.retail.igs.integration.service.Dy
namicServiceProviderImpl ${JAVA_OPTIONS}"
```

b. To edit the server start arguments in the WebLogic console, click igs-server -> Server start as below:



ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Home >Summary of Servers >igs-server
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
RIBDomain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Configure startup arguments for Managed Servers
Start Managed Servers from the Administration Console
Shut down a server instance
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (17)
Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Health Monitoring Server Start
Save

Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for igs-server

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to config

Java Home:

Java Vendor:

BEA Home:

Root Directory:

Class Path:

Arguments:

```
-Xms512m -Xmx1024m -XX:MaxPermSize=512m -
Doracle.retail.soa.enabler.service.provider.engine.ServiceProv
iderImplLookupFactory.interceptor=com.oracle.retail.igs.integr
ation.service.DynamicServiceProviderImpl
```

4. Bounce the Admin and managed server before you deploy the IGS.

Prepare Integration Gateway Services (IGS)

The IGS can be installed under $RIB_HOME (rib-home/tools-home/integration-bus-gateway-services) as described below.

Running IGS under $RIB_HOME

To run IGS under $RIB_HOME, complete the following steps:

1. Download the IntegrationGatewayService19.1.000ForAll19.1.000Apps_eng_ga.tar and untar it under rib-home/tools-home.

   ```
   cd rib-home/tools-home/
   IntegrationGatewayService19.1.000ForAll19.1.000Apps_eng_ga.tar
   ```

2. Go to rib-home/tools-home/integration-bus-gateway-services/conf and edit the IgsConfig.properties as follows:

   - Change the value of WlsUrl to point to the WebLogic server where IGS is going to be deployed. The port in the WlsUrl should be the administration port.
   - Change the value of WlsTarget to the instance name where IGS is going to be deployed (for example, igs-service_wls_instance).

3. Go to $IGS_HOME integration-bus-gateway-services/bin. Run the igs-install.sh. Running this script does the following:

- Verifies whether the attempted IGS installation is from within rib-home or in standalone mode; preconfiguration cleanup is based on this mode.
- Asks the user for the WebLogic user name and password and saves it in a secure credential store.

Note: The WebLogic user name used here should be set up with the administrator role.

- Prepares the igs-service.ear, based on the number of channels and the number of configured AQ JMS servers.
- Configures the WebLogic server with the AQ JMS server information listed in the rib-deployment-env-info.xml.
- Deploys the igs-service-ear to the WebLogic server.

4. Restart the WebLogic managed server.

All of the items in Step 4 also can be performed separately, as follows:

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

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

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

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

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

   ```
   sh igs-admin.sh -deploy
   ```
6. Restart the WebLogic managed server.
7. If the igs-service.ear exists, it must be undeployed. Run the rib-home/tools-home/integration-bus-gateway-services/bin/igs-admin.sh -undeploy to undeploy an igs-service.ear.

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

Note: The log files are located here: **$IGS_HOME/ integration-bus-gateway-services/log**

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

Verify the IGS Application Installation Using the Administration Console

To verify the IGS installations using the Oracle WebLogic Administration Console, complete the following steps:

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

1. Navigate to the Deployments page.
2. On the Summary of Deployments page, locate the igs-service.
3. To expand the tree, click the + beside the ig-service.
4. Locate the Web Services section.
5. Click any Web service (for example, ASNInPublishingService) to move to settings for ASNInPublishingService page.
6. Select the Testing tab.
7. To expand the tree, click the + beside the service name.
8. Locate the Test Client link. Move to the WebLogic Test Client page.
9. Select the Ping operation. Enter test data in the string arg0: text box. Click **Ping**.
10. The test page will include the request message and the response message.

Secure IGS Web Services Using the Administration Console

IGS Web services can be secured in two ways. One approach is simple user name and password authentication. For the other approach, passwords are encrypted with certificates.

The following describes both approaches for server-side and client-side setup.

Note: The various policy files that can be used to secure Web services are listed in the ws-policy tab of the Web service in the WebLogic Server Administration Console.

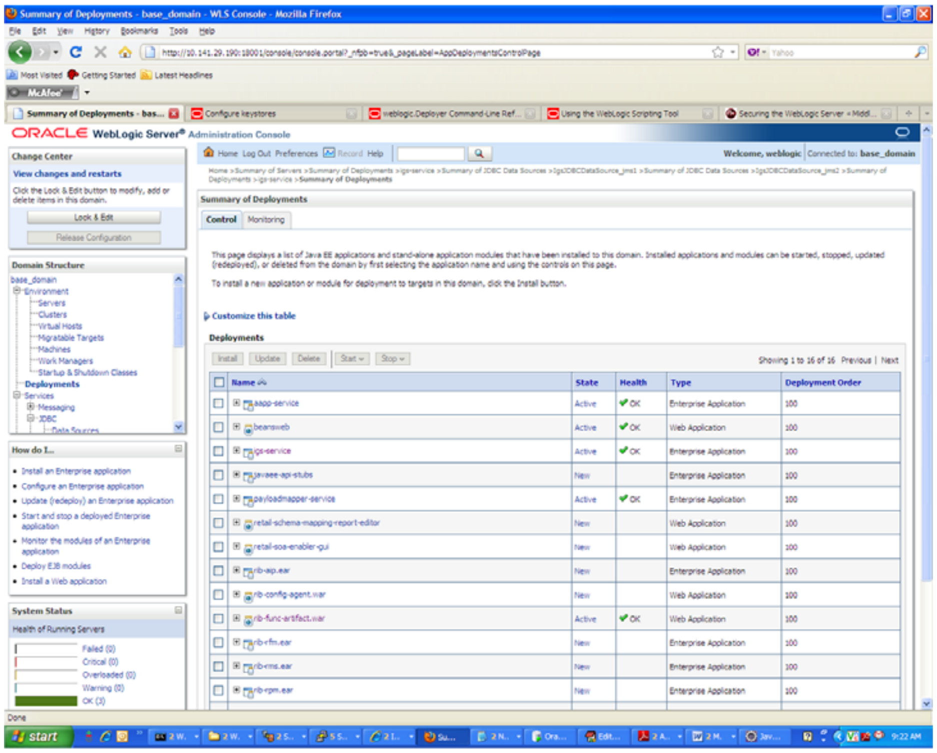
Server-side Setup for User Name and Password Authentication

This section describes the two-step process required for securing Web services on the server side. These steps are performed using the Oracle WebLogic Server Administration Console.

Attach Policy File to the Web Service

The usernametoken.xml contains the policy used by the Web service and is found in the META_INF/policies folder in the .ear file. Complete the following steps to attach the policy file to a Web service.

1. In the Summary of Deployments screen, click the application. In the illustration below, the application is igs-service.


Summary of Deployments - base_domain - WLS Console - Mozilla Firefox
ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
How do I...
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

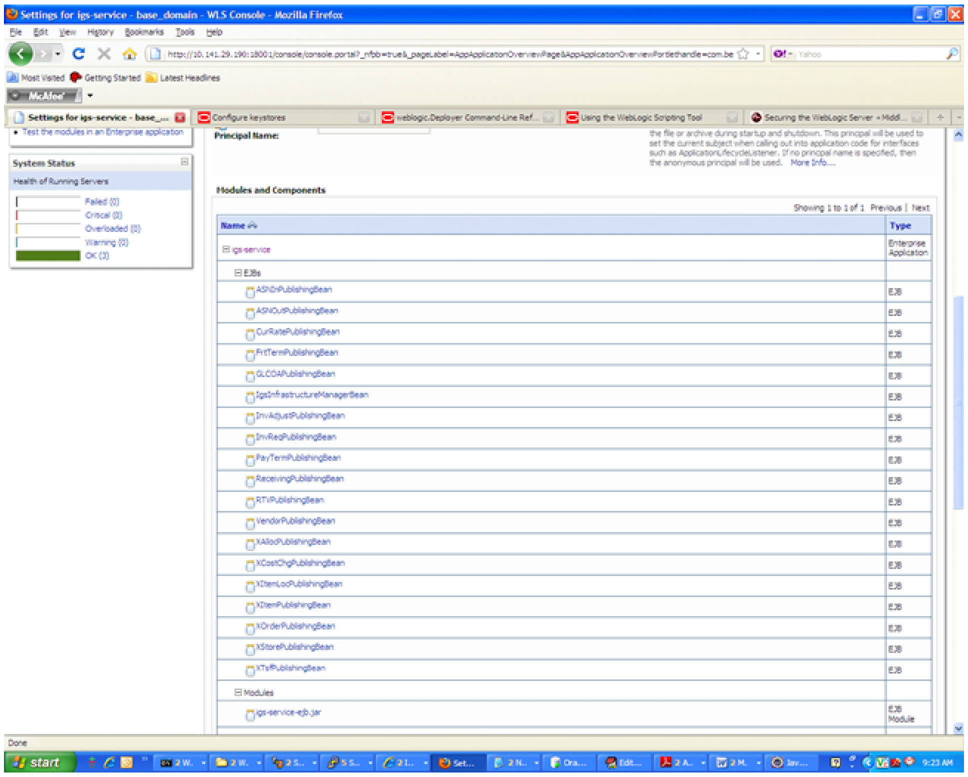
Customize this table

Deployments

Install | Update | Delete | Start | Stop — Showing 1 to 16 of 16 Previous | Next

Name	State	Health	Type	Deployment Order
aapp-service	Active	✓ OK	Enterprise Application	100
beansweb	Active	✓ OK	Web Application	100
igs-service	Active	✓ OK	Enterprise Application	100
javaee-api-stubs	New		Enterprise Application	100
payloadmapper-service	Active	✓ OK	Enterprise Application	100
retail-schema-mapping-report-editor	New		Web Application	100
retail-soa-enabler-gui	New		Web Application	100
rib-aip.ear	New		Enterprise Application	100
rib-config-agent.war	New		Web Application	100
rib-func-artifact.war	Active	✓ OK	Web Application	100
rib-rfm.ear	New		Enterprise Application	100
rib-rms.ear	New		Enterprise Application	100
rib-rpm.ear	New		Enterprise Application	100

2. An overview page is displayed, including a list of modules and components installed as part of the application.


Settings for igs-service - base_domain - WLS Console - Mozilla Firefox
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)

Principal Name:

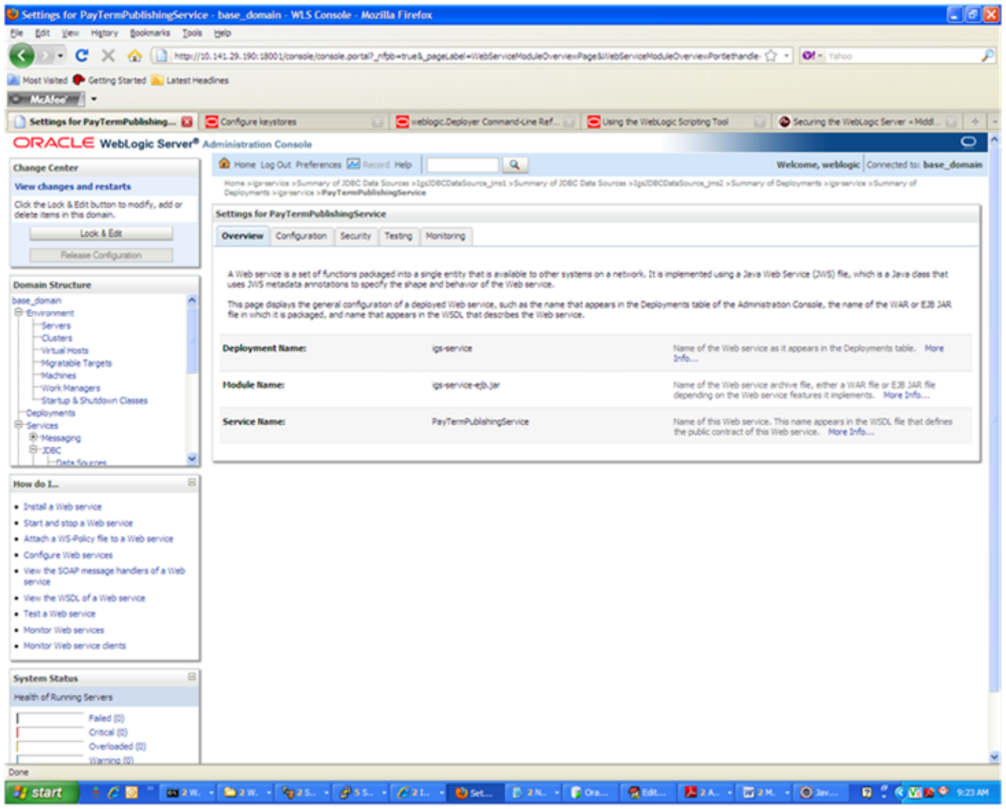
the file or archive during startup and shutdown. This principal will be used to set the current subject when calling out into application code for interfaces such as ApplicationLifecycleListener. If no principal name is specified, then the anonymous principal will be used. More Info...

Modules and Components

Showing 1 to 1 of 1 Previous | Next

Name	Type
igs-service	Enterprise Application
EJBs	
ASNInPublishingBean	EJB
ASNOutPublishingBean	EJB
CurRatePublishingBean	EJB
FrtTermPublishingBean	EJB
GLCOAPublishingBean	EJB
IgsInfrastructureManagerBean	EJB
InvAdjustPublishingBean	EJB
InvReqPublishingBean	EJB
PayTermPublishingBean	EJB
ReceivingPublishingBean	EJB
RTVPublishingBean	EJB
VendorPublishingBean	EJB
XAllocPublishingBean	EJB
XCostChgPublishingBean	EJB
XItemLocPublishingBean	EJB
XItemPublishingBean	EJB
XOrderPublishingBean	EJB
XStorePublishingBean	EJB
XTsfPublishingBean	EJB
Modules	
igs-service-ejb.jar	EJB Module

3. In the Web service list, click the service for which you want to enable security. The following screen is displayed to provide an overview of the Web service.


Settings for PayTermPublishingService - base_domain - WLS Console - Mozilla Firefox
ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
How do I...
System Status
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Overview
Configuration
Security
Testing
Monitoring

Click the Lock & Edit button to modify, add or delete items in this domain.

Home >igs-service >Summary of JDBC Data Sources >IgsJDBCDataSource_jms1 >Summary of JDBC Data Sources >IgsJDBCDataSource_jms2 >Summary of Deployments >igs-service >Summary of Deployments >igs-service >**PayTermPublishingService**

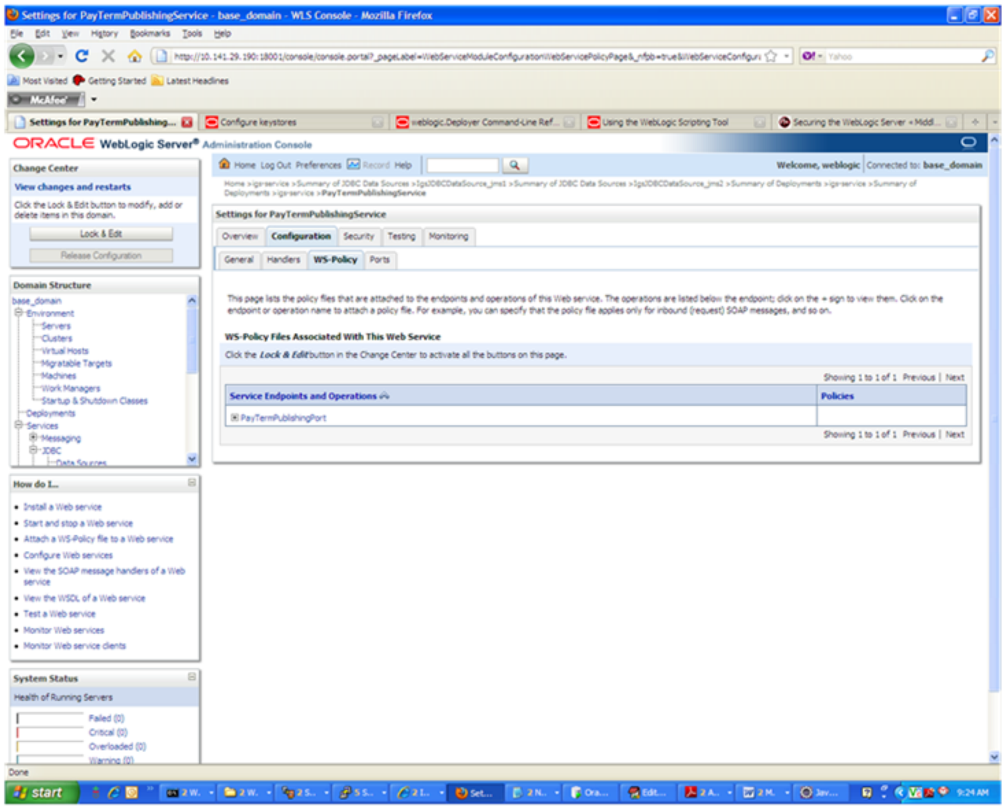
Settings for PayTermPublishingService

A Web service is a set of functions packaged into a single entity that is available to other systems on a network. It is implemented using a Java Web Service (JWS) file, which is a Java class that uses JWS metadata annotations to specify the shape and behavior of the Web service.

This page displays the general configuration of a deployed Web service, such as the name that appears in the Deployments table of the Administration Console, the name of the WAR or EJB JAR file in which it is packaged, and name that appears in the WSDL that describes the Web service.

Deployment Name:	igs-service	Name of the Web service as it appears in the Deployments table. More Info...
Module Name:	igs-service-ejb.jar	Name of the Web service archive file, either a WAR file or EJB JAR file depending on the Web service features it implements. More Info...
Service Name:	PayTermPublishingService	Name of this Web service. This name appears in the WSDL file that defines the public contract of this Web service. More Info...

4. On this overview screen, click the Configuration tab. Click the WS-Policy tab. The Web service port is shown under Service Endpoints and Operations.



Settings for PayTermPublishingService

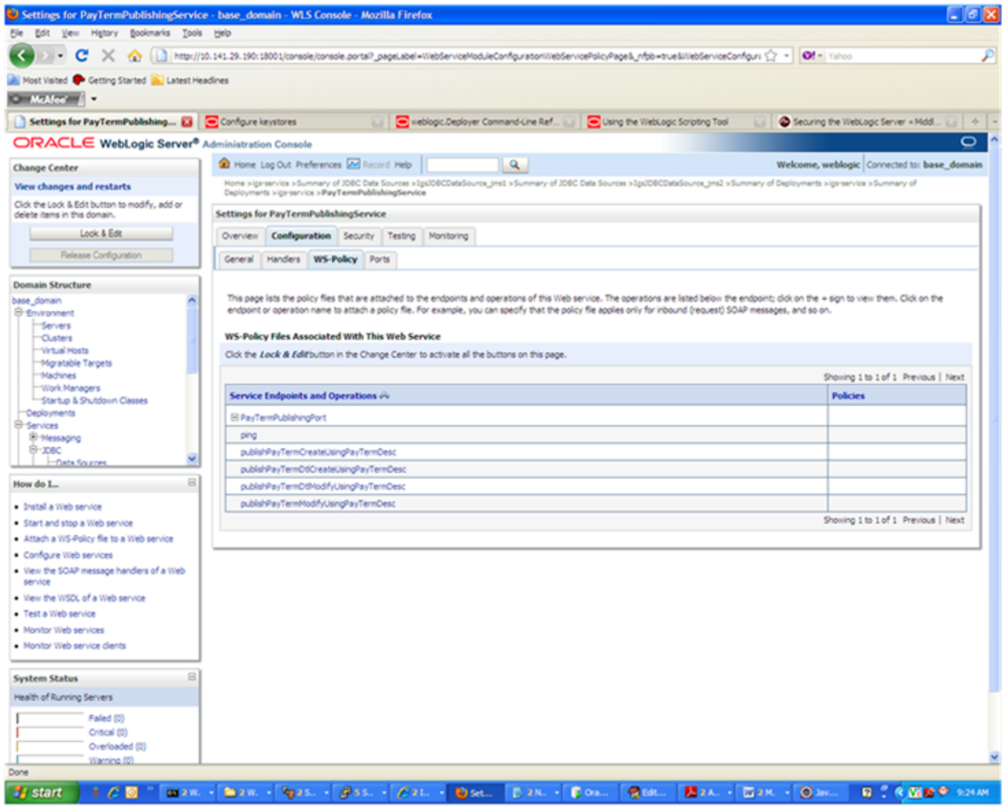
This page lists the policy files that are attached to the endpoints and operations of this Web service. The operations are listed below the endpoint; click on the + sign to view them. Click on the endpoint or operation name to attach a policy file. For example, you can specify that the policy file applies only for inbound (request) SOAP messages, and so on.

WS-Policy Files Associated With This Web Service

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

Service Endpoints and Operations	Policies
⊞ PayTermPublishingPort	

5. Click the plus sign next to the port name. The Web service operations are displayed.


Settings for PayTermPublishingService - base_domain - WLS Console - Mozilla Firefox
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
How do I...
System Status
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Overview Configuration Security Testing Monitoring
General Handlers WS-Policy Ports

Settings for PayTermPublishingService

This page lists the policy files that are attached to the endpoints and operations of this Web service. The operations are listed below the endpoint; click on the + sign to view them. Click on the endpoint or operation name to attach a policy file. For example, you can specify that the policy file applies only for inbound (request) SOAP messages, and so on.

WS-Policy Files Associated With This Web Service

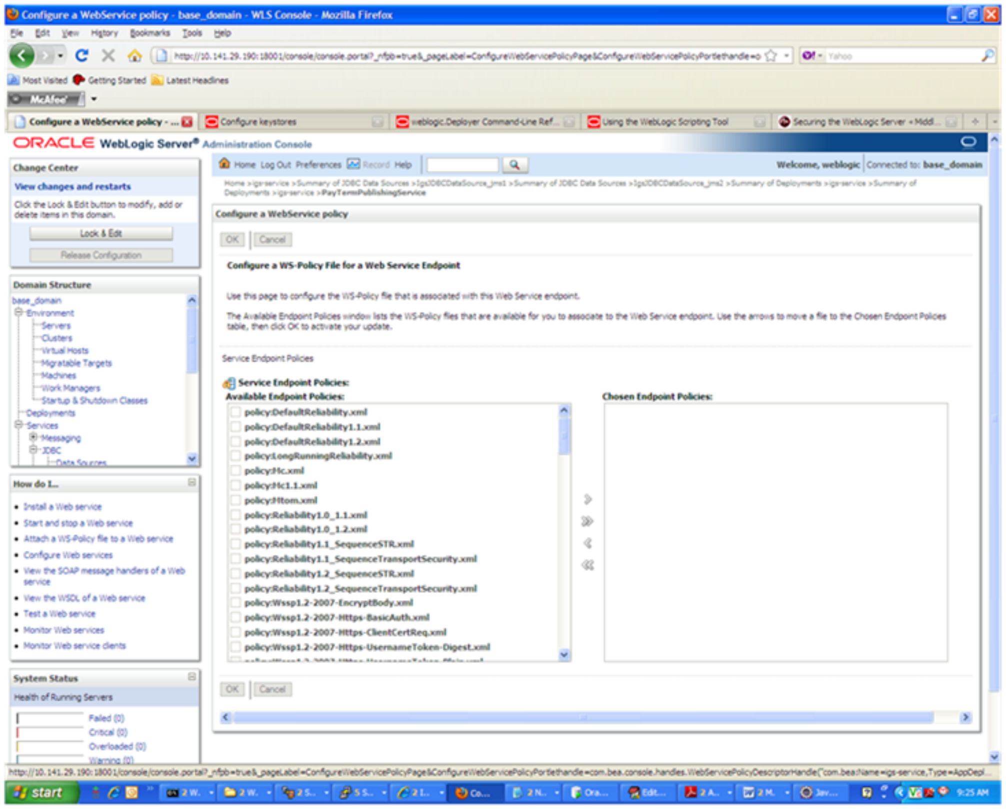
Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

Showing 1 to 1 of 1 Previous | Next

Service Endpoints and Operations	Policies
PayTermPublishingPort	
ping	
publishPayTermCreateUsingPayTermDesc	
publishPayTermDtlCreateUsingPayTermDesc	
publishPayTermDtlModifyUsingPayTermDesc	
publishPayTermModifyUsingPayTermDesc	

Showing 1 to 1 of 1 Previous | Next

6. You can secure all the Web service operations at once or select only the operations you want to secure. Click the name of the port. On the Configure a Web Service policy screen, you can attach the policy file to the Web service. Select WebLogic and click the **Next** button.



Configure a WebService policy

OK Cancel

Configure a WS-Policy File for a Web Service Endpoint

Use this page to configure the WS-Policy file that is associated with this Web Service endpoint.

The Available Endpoint Policies window lists the WS-Policy files that are available for you to associate to the Web Service endpoint. Use the arrows to move a file to the Chosen Endpoint Policies table, then click OK to activate your update.

Service Endpoint Policies

Service Endpoint Policies:

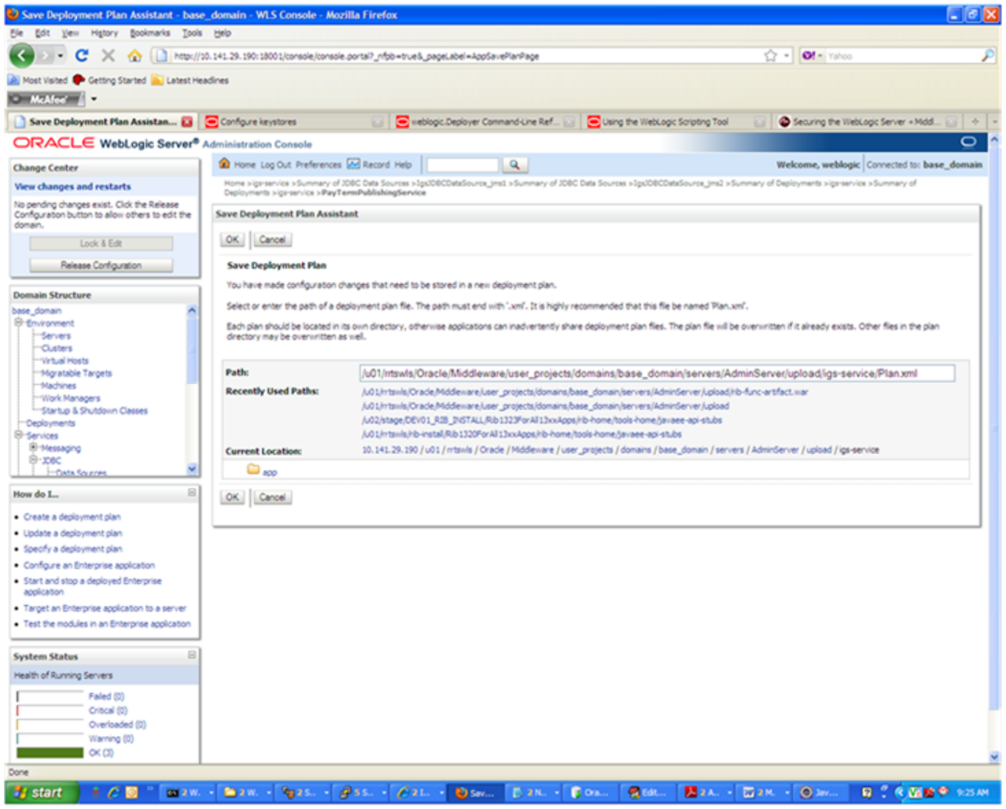
Available Endpoint Policies:

- policy:DefaultReliability.xml
- policy:DefaultReliability1.1.xml
- policy:DefaultReliability1.2.xml
- policy:LongRunningReliability.xml
- policy:Mc.xml
- policy:Mc1.1.xml
- policy:Mtom.xml
- policy:Reliability1.0_1.1.xml
- policy:Reliability1.0_1.2.xml
- policy:Reliability1.1_SequenceSTR.xml
- policy:Reliability1.1_SequenceTransportSecurity.xml
- policy:Reliability1.2_SequenceSTR.xml
- policy:Reliability1.2_SequenceTransportSecurity.xml
- policy:Wssp1.2-2007-EncryptBody.xml
- policy:Wssp1.2-2007-Https-BasicAuth.xml
- policy:Wssp1.2-2007-Https-ClientCertReq.xml
- policy:Wssp1.2-2007-Https-UsernameToken-Digest.xml

Chosen Endpoint Policies:

OK Cancel

7. From the Available Endpoint Policies list, select policy:usernametoken.xml. Click the right arrow to move it to the drop down list below Chosen Endpoint Policies. Click **OK**. The Save Deployment Plan Assistant screen is displayed.



Save Deployment Plan Assistant

Save Deployment Plan

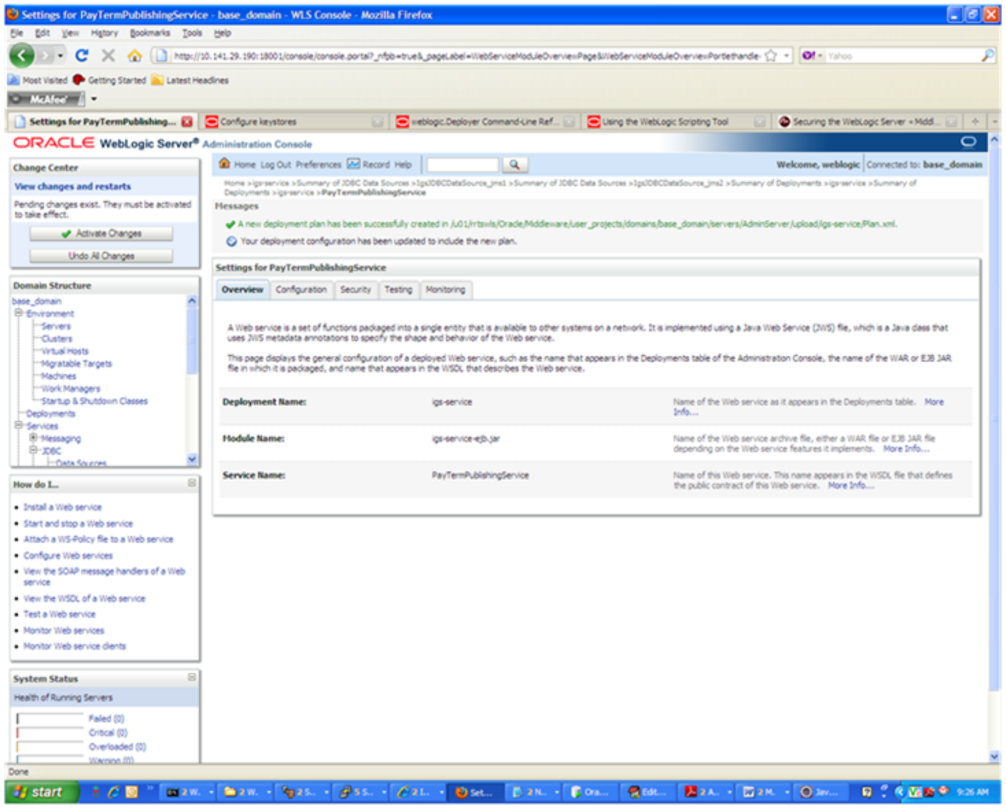
You have made configuration changes that need to be stored in a new deployment plan.

Select or enter the path of a deployment plan file. The path must end with '.xml'. It is highly recommended that this file be named 'Plan.xml'.

Each plan should be located in its own directory, otherwise applications can inadvertently share deployment plan files. The plan file will be overwritten if it already exists. Other files in the plan directory may be overwritten as well.

Path:	/u01/rrtswls/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/upload/igs-service/Plan.xml
Recently Used Paths:	/u01/rrtswls/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/upload/rib-func-artifact.war
	/u01/rrtswls/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/upload
	/u02/stage/DEV01_RIB_INSTALL/Rib1323ForAll13xxApps/rib-home/tools-home/javaee-api-stubs
	/u01/rrtswls/rib-install/Rib1320ForAll13xxApps/rib-home/tools-home/javaee-api-stubs
Current Location:	10.141.29.190 / u01 / rrtswls / Oracle / Middleware / user_projects / domains / base_domain / servers / AdminServer / upload / igs-service

8. At the bottom of the Save Deployment Plan Assistant screen, click **OK**. The following screen is displayed, including status messages near the top.



Change Center

View changes and restarts

Pending changes exist. They must be activated to take effect.

Messages

✓ A new deployment plan has been successfully created in /u01/rrtswls/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/upload/igs-service/Plan.xml.

Your deployment configuration has been updated to include the new plan.

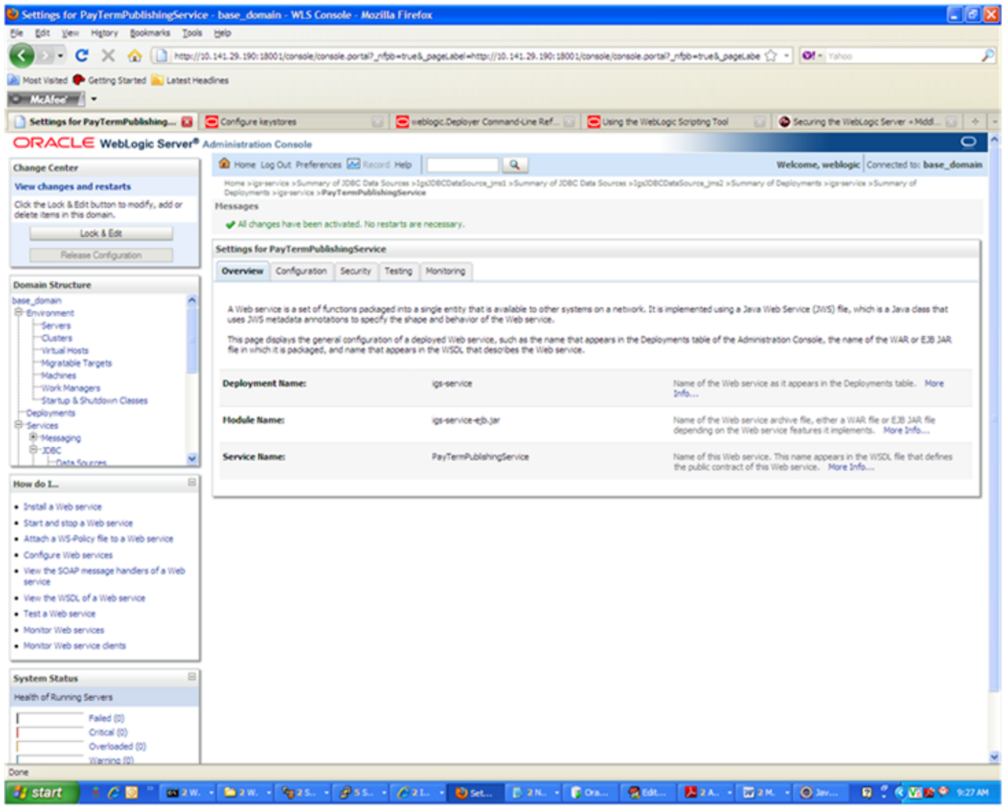
Settings for PayTermPublishingService

A Web service is a set of functions packaged into a single entity that is available to other systems on a network. It is implemented using a Java Web Service (JWS) file, which is a Java class that uses JWS metadata annotations to specify the shape and behavior of the Web service.

This page displays the general configuration of a deployed Web service, such as the name that appears in the Deployments table of the Administration Console, the name of the WAR or EJB JAR file in which it is packaged, and name that appears in the WSDL that describes the Web service.

Deployment Name:	igs-service	Name of the Web service as it appears in the Deployments table. More Info...
Module Name:	igs-service-ejb.jar	Name of the Web service archive file, either a WAR file or EJB JAR file depending on the Web service features it implements. More Info...
Service Name:	PayTermPublishingService	Name of this Web service. This name appears in the WSDL file that defines the public contract of this Web service. More Info...

9. Click **Activate Changes**. The following screen is displayed.



Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Messages

✓ All changes have been activated. No restarts are necessary.

Settings for PayTermPublishingService

A Web service is a set of functions packaged into a single entity that is available to other systems on a network. It is implemented using a Java Web Service (JWS) file, which is a Java class that uses JWS metadata annotations to specify the shape and behavior of the Web service.

This page displays the general configuration of a deployed Web service, such as the name that appears in the Deployments table of the Administration Console, the name of the WAR or EJB JAR file in which it is packaged, and name that appears in the WSDL that describes the Web service.

Deployment Name:	igs-service	Name of the Web service as it appears in the Deployments table. More Info...
Module Name:	igs-service-ejb.jar	Name of the Web service archive file, either a WAR file or EJB JAR file depending on the Web service features it implements. More Info...
Service Name:	PayTermPublishingService	Name of this Web service. This name appears in the WSDL file that defines the public contract of this Web service. More Info...

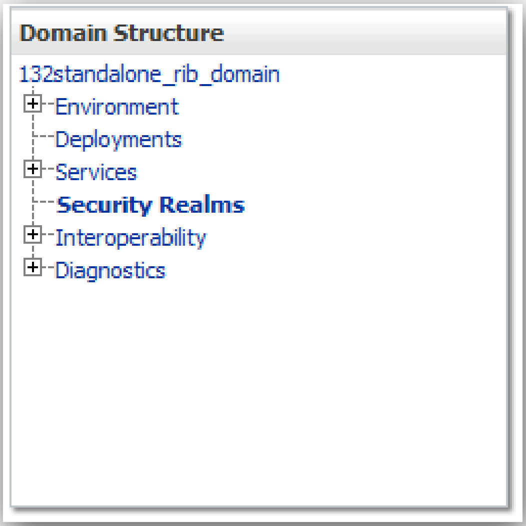
10. Under the Testing tab, on the Web Service page, click the WSDL to view the details of the policy just added to the Web service. The WSDL contains information similar to the following.

```
<?xml version='1.0' encoding='UTF-8'?>
 <definitions
xmlns:tns="http://www.oracle.com/retail/igs/integration/services/PayTermPublish
ingService/v1"
xmlns:ns1="http://www.oracle.com/retail/integration/bus/gateway/services/Busine
ssObjectId/v1"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns2="http://www.oracle.com/retail/integration/services/exception/v1"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="PayTermPublishingService"
targetNamespace="http://www.oracle.com/retail/igs/integration/services/PayTermP
ublishingService/v1"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wssutil="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecuri
ty-utility-1.0.xsd">
<wsp:UsingPolicy wssutil:Required="true" />
<wsp:Policy wssutil:Id="usernametoken">
<ns0:SupportingTokens
xmlns:ns0="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200512">
<wsp:Policy>
<ns0:UsernameToken
ns0:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200512/Inc
ludeToken/AlwaysToRecipient">
<wsp:Policy>
<ns0:WssUsernameToken10/>
</wsp:Policy>
</ns0:UsernameToken>
</wsp:Policy>
</ns0:SupportingTokens>
</wsp:Policy>
```

Create Roles and Users

This section describes how to add roles and users who can access the Web services. The first step is to add users to the security realm, as described below.

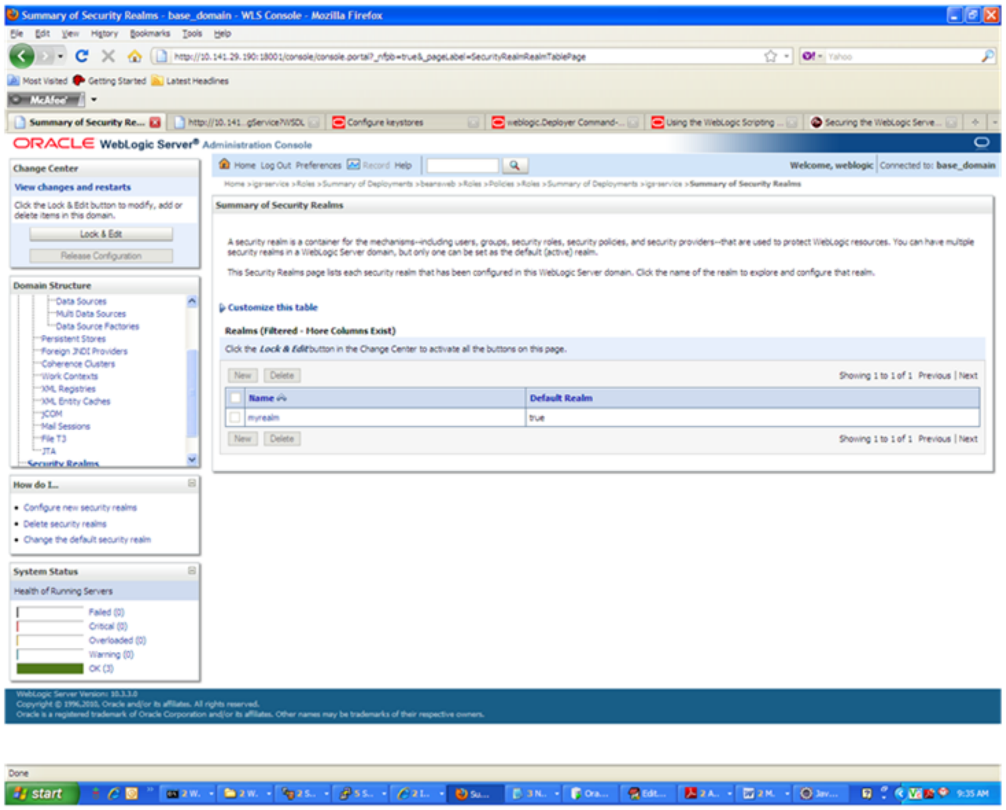
1. In the Domain Structure window of the Oracle WebLogic Services Administration Console, click the Security Realms link.



Domain Structure

132standalone_rib_domain
- Environment
- Deployments
- Services
- **Security Realms**
- Interoperability
- Diagnostics

2. The Summary of Security Realms screen is displayed, including the name of the default realm.



Summary of Security Realms - base_domain - WLS Console - Mozilla Firefox

http://10.141.29.190:18001/console/console.portal?_nfpb=true&_pageLabel=SecurityRealmRealmTablePage

ORACLE WebLogic Server® Administration Console

Welcome, weblogic | Connected to: base_domain

Home >igs-service >Roles >Summary of Deployments >beansweb >Roles >Policies >Roles >Summary of Deployments >igs-service >**Summary of Security Realms**

Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Lock & Edit

Release Configuration

Domain Structure

Data Sources
Multi Data Sources
Data Source Factories
Persistent Stores
Foreign JNDI Providers
Coherence Clusters
Work Contexts
XML Registries
XML Entity Caches
jCOM
Mail Sessions
File T3
JTA
Security Realms

How do I...

- Configure new security realms
- Delete security realms
- Change the default security realm

System Status

Health of Running Servers

Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)

Summary of Security Realms

A security realm is a container for the mechanisms--including users, groups, security roles, security policies, and security providers--that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

Customize this table

Realms (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

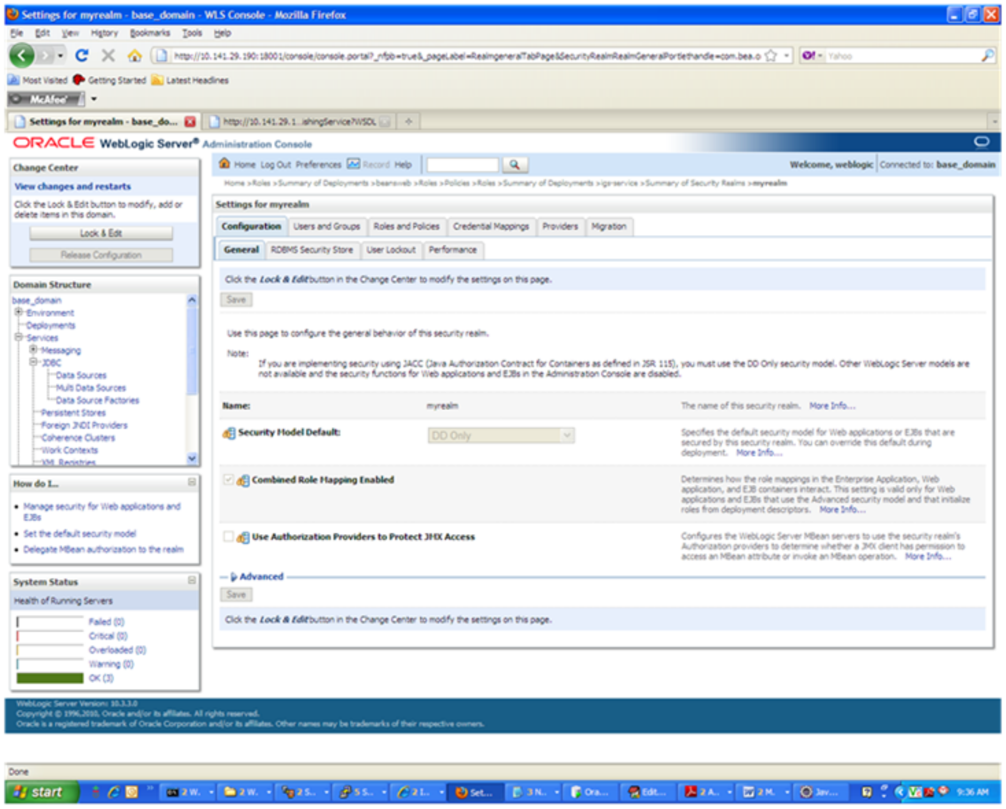
New Delete Showing 1 to 1 of 1 Previous | Next

Name	Default Realm
myrealm	true

New Delete Showing 1 to 1 of 1 Previous | Next

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Click the name of the default realm. The settings for the realm are displayed.



Settings for myrealm

Configuration | Users and Groups | Roles and Policies | Credential Mappings | Providers | Migration

General | RDBMS Security Store | User Lockout | Performance

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

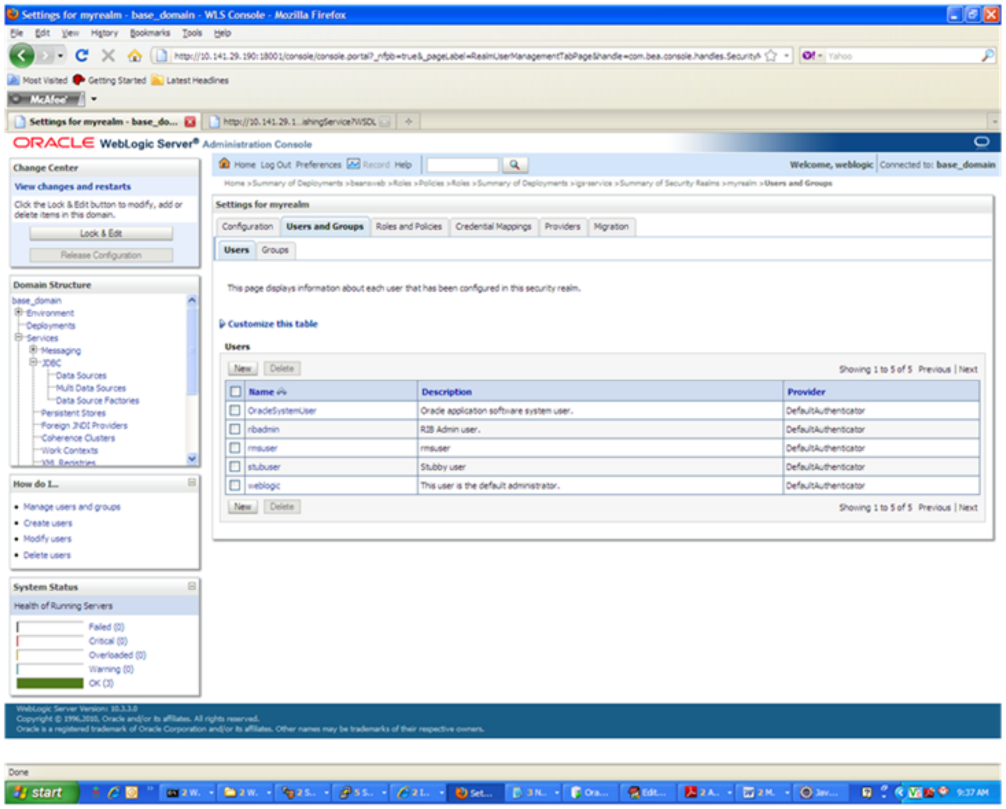
Use this page to configure the general behavior of this security realm.

Note:
If you are implementing security using JACC (Java Authorization Contract for Containers as defined in JSR 115), you must use the DD Only security model. Other WebLogic Server models are not available and the security functions for Web applications and EJBs in the Administration Console are disabled.

Name:	myrealm	The name of this security realm. More Info...
Security Model Default:	DD Only	Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. More Info...
✓ **Combined Role Mapping Enabled**		Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. More Info...
☐ **Use Authorization Providers to Protect JMX Access**		Configures the WebLogic Server MBean servers to use the security realm's Authorization providers to determine whether a JMX client has permission to access an MBean attribute or invoke an MBean operation. More Info...

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

4. On the Settings screen, click the Users and Groups tab.



Settings for myrealm

Configuration | Users and Groups | Roles and Policies | Credential Mappings | Providers | Migration

Users | Groups

This page displays information about each user that has been configured in this security realm.

Customize this table

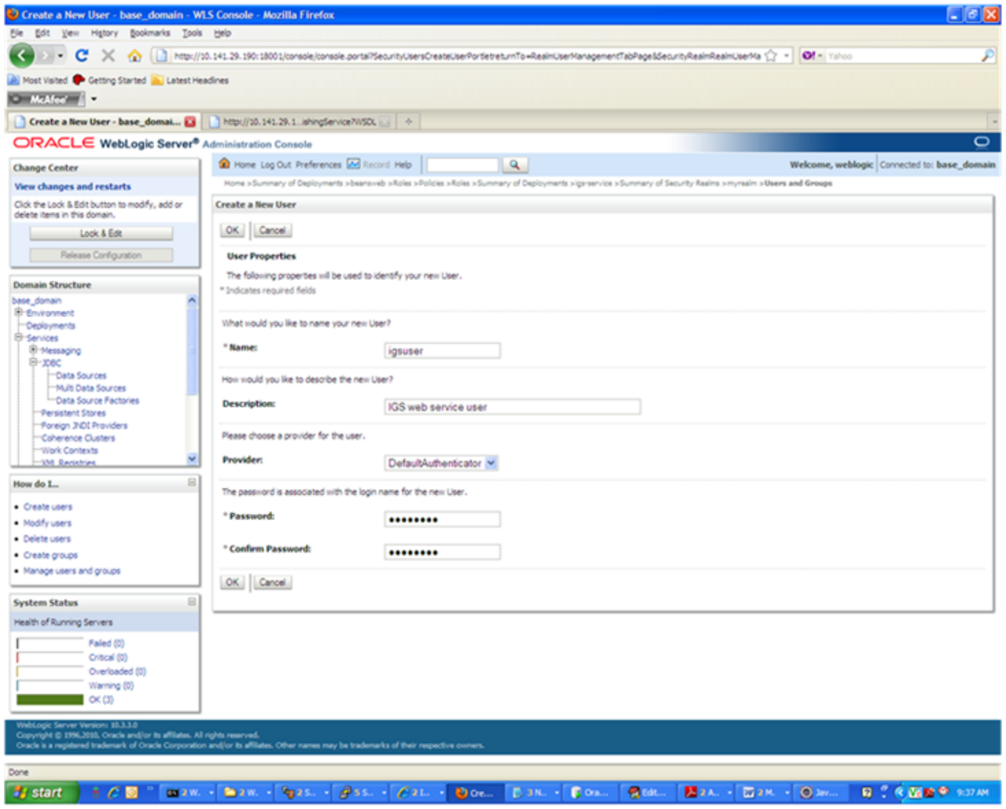
Users

Showing 1 to 5 of 5 Previous | Next

	Name	Description	Provider
☐	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
☐	ribadmin	RIB Admin user.	DefaultAuthenticator
☐	rmsuser	rmsuser	DefaultAuthenticator
☐	stubuser	Stubby user	DefaultAuthenticator
☐	weblogic	This user is the default administrator.	DefaultAuthenticator

Showing 1 to 5 of 5 Previous | Next

5. In the Users and Groups tab, click the Users tab. At the bottom of the Users tab, click **New**. The Create a New User screen is displayed.



Create a New User

User Properties

The following properties will be used to identify your new User.

* Indicates required fields

What would you like to name your new User?

* Name: igsuser

How would you like to describe the new User?

Description: IGS web service user

Please choose a provider for the user.

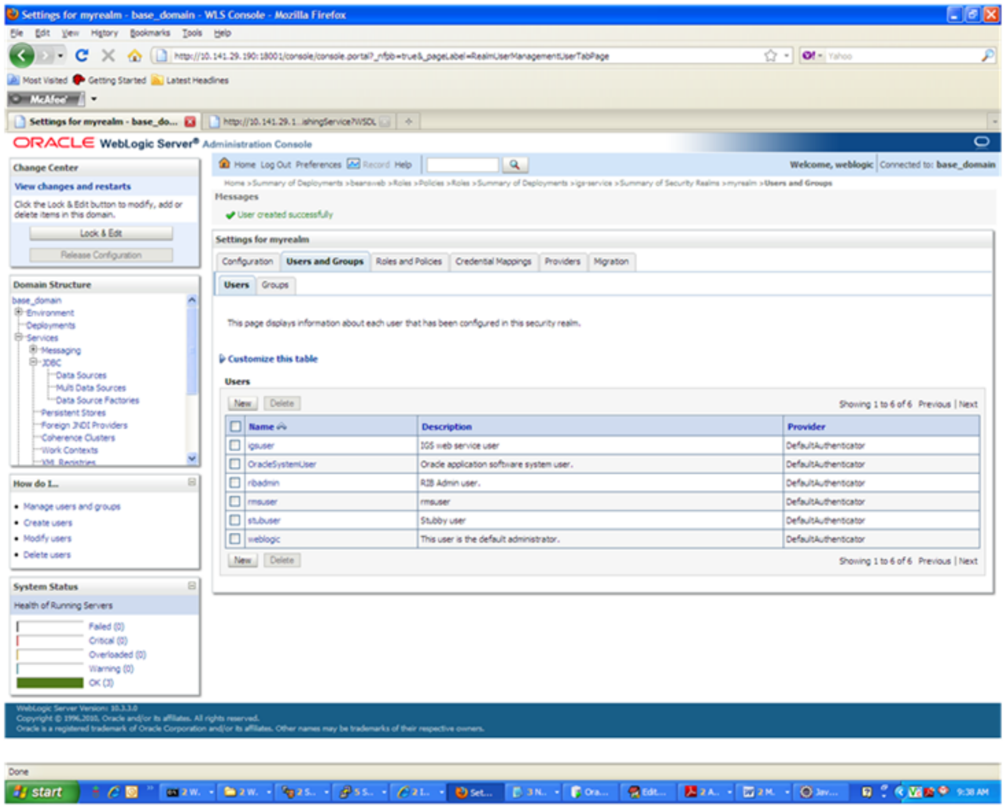
Provider: DefaultAuthenticator

The password is associated with the login name for the new User.

* Password: ••••••••

* Confirm Password: ••••••••

6. In the Create a New User screen, enter a user name and password. Leave the default value for Provider. Click **OK** to save the information. The new user is added to the list of users.



http://10.141.29.190:18001/console/console.portal?_nfpb=true&_pageLabel=RealmUserManagementUserTabPage

Home >Summary of Deployments >beansweb >Roles >Policies >Roles >Summary of Deployments >igs-service >Summary of Security Realms >myrealm >Users and Groups

Messages

✓ User created successfully

Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for myrealm

This page displays information about each user that has been configured in this security realm.

Users

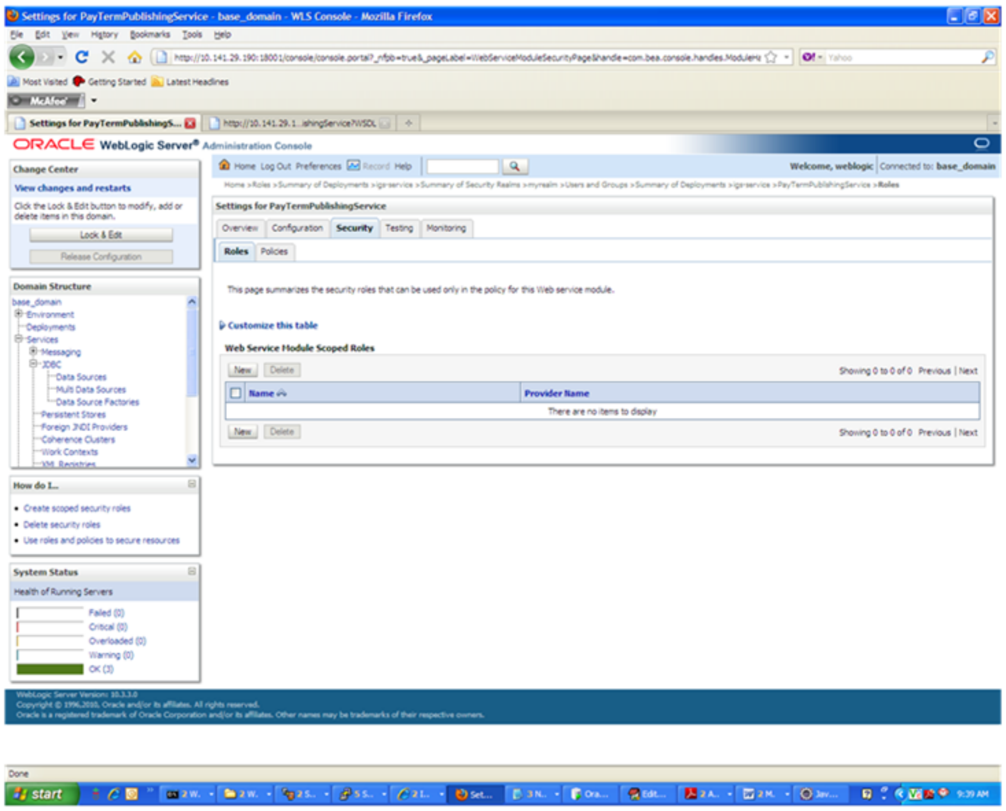
	Name	Description	Provider
☐	igsuser	IGS web service user	DefaultAuthenticator
☐	OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
☐	ribadmin	RIB Admin user.	DefaultAuthenticator
☐	rmsuser	rmsuser	DefaultAuthenticator
☐	stubuser	Stubby user	DefaultAuthenticator
☐	weblogic	This user is the default administrator.	DefaultAuthenticator

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Note: You can add roles from the Roles and Policies tab of the security realm or through the Security tab of the Web service. The following instructions are for creating a role through the Security tab of the Web service.

---

7. Navigate to the Security tab of the Web service. Click the Roles tab.


Settings for PayTermPublishingService - base_domain - WLS Console - Mozilla Firefox

http://10.141.29.190:18001/console/console.portal?_nfpb=true&_pageLabel=WebServiceModuleSecurityPage&handle=com.bea.console.handles.ModuleH

Home >Roles >Summary of Deployments >igs-service >Summary of Security Realms >myrealm >Users and Groups >Summary of Deployments >igs-service >PayTermPublishingService >**Roles**

Welcome, weblogic | Connected to: base_domain

Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for PayTermPublishingService

Overview | Configuration | **Security** | Testing | Monitoring

Roles | Policies

This page summarizes the security roles that can be used only in the policy for this Web service module.

Customize this table

Web Service Module Scoped Roles

Showing 0 to 0 of 0 Previous | Next

	Name	Provider Name
	There are no items to display	

Showing 0 to 0 of 0 Previous | Next

How do I...

- Create scoped security roles
- Delete security roles
- Use roles and policies to secure resources

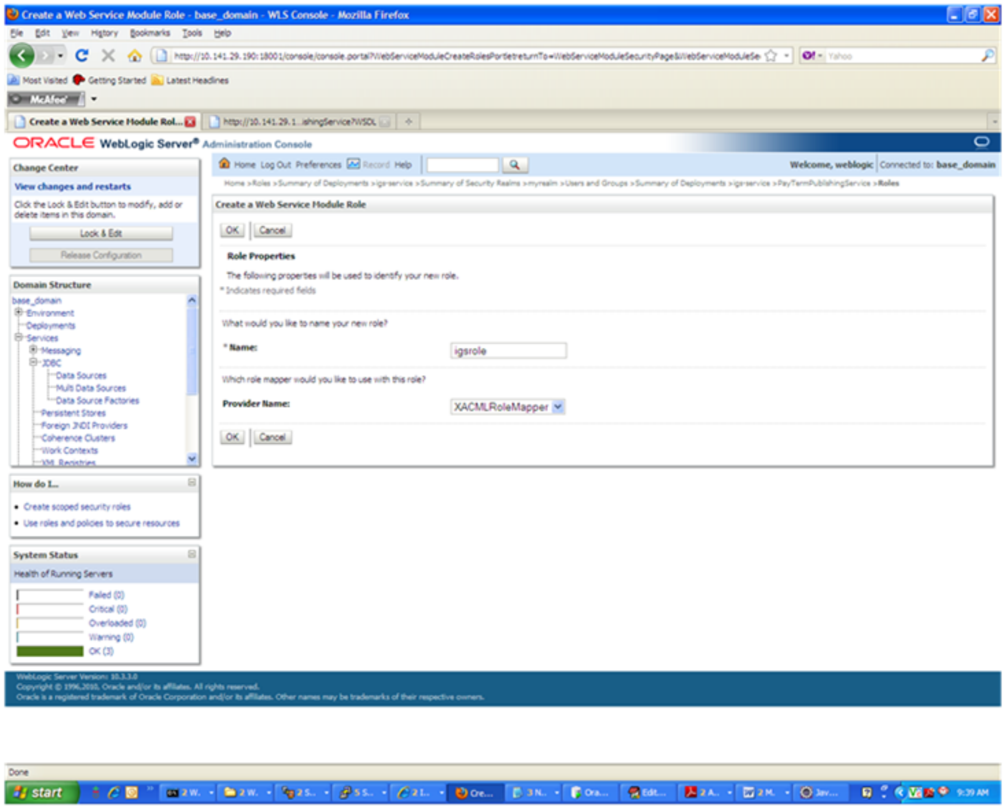
System Status

Health of Running Servers

Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

8. In the Roles tab, click **New**. The Create a Web Service Module role screen is displayed.



Create a Web Service Module Role

Role Properties

The following properties will be used to identify your new role.

* Indicates required fields

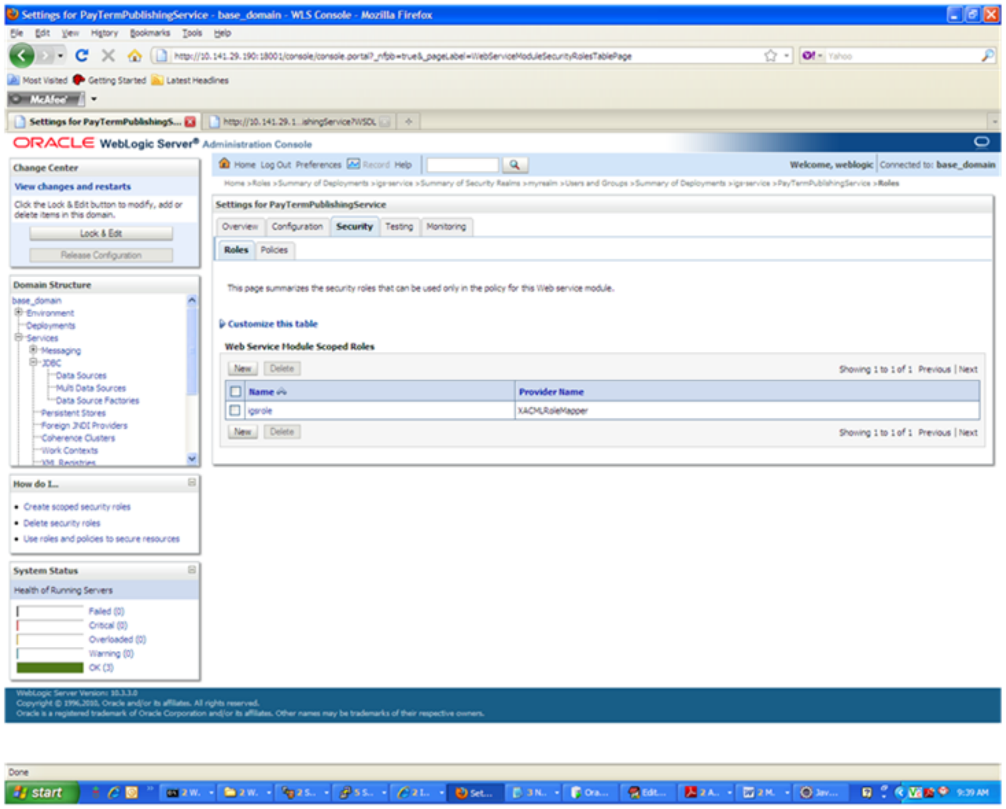
What would you like to name your new role?

* Name: igsrole

Which role mapper would you like to use with this role?

Provider Name: XACMLRoleMapper

9. In the Create a Web Service Module Role screen, enter the role name in the Name field (for example, rmsrole). Leave the default value in the Provider Name field. Click **OK**. The new role is displayed in the Roles tab of the Web service.



Settings for PayTermPublishingService

Overview | Configuration | Security | Testing | Monitoring

Roles | Policies

This page summarizes the security roles that can be used only in the policy for this Web service module.

Customize this table

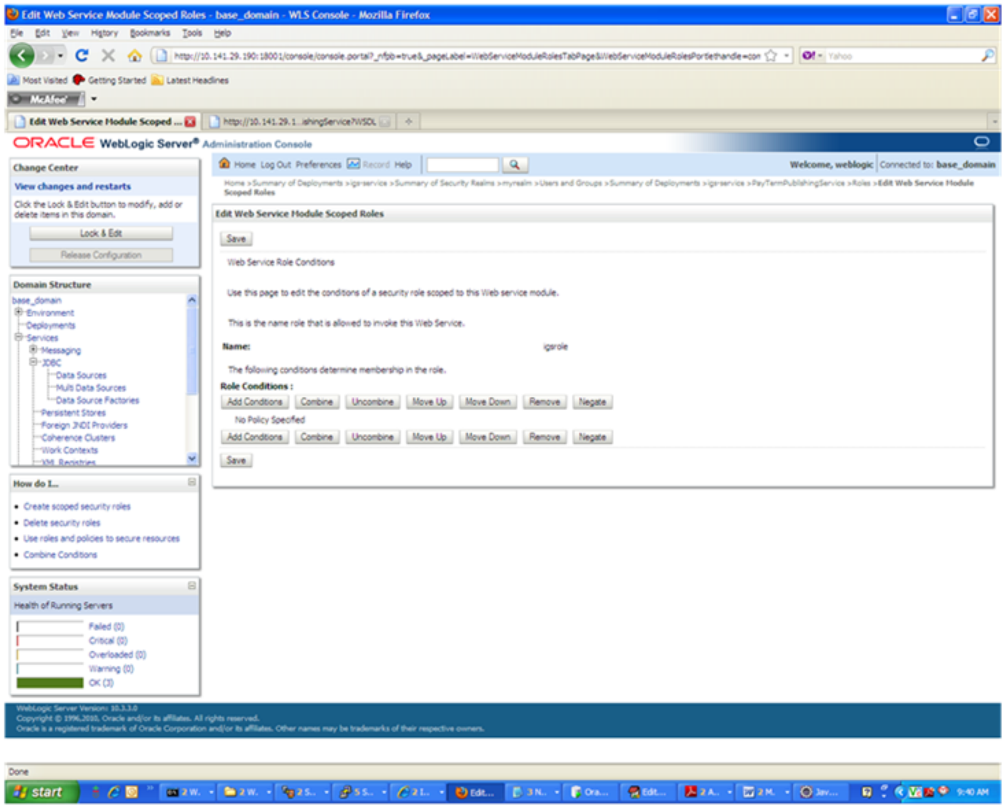
Web Service Module Scoped Roles

Showing 1 to 1 of 1 Previous | Next

Name	Provider Name
igsrole	XACMLRoleMapper

Showing 1 to 1 of 1 Previous | Next

10. To add the user to the role, click the name of the new role in the Roles tab. The Edit Web Service Module Scoped Roles screen is displayed.


Edit Web Service Module Scoped Roles - base_domain - WLS Console - Mozilla Firefox
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
How do I...
System Status
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Save
Add Conditions
Combine
Uncombine
Move Up
Move Down
Remove
Negate

Home >Summary of Deployments >igs-service >Summary of Security Realms >myrealm >Users and Groups >Summary of Deployments >igs-service >PayTermPublishingService >Roles >**Edit Web Service Module Scoped Roles**

Edit Web Service Module Scoped Roles

Web Service Role Conditions

Use this page to edit the conditions of a security role scoped to this Web service module.

This is the name role that is allowed to invoke this Web Service.

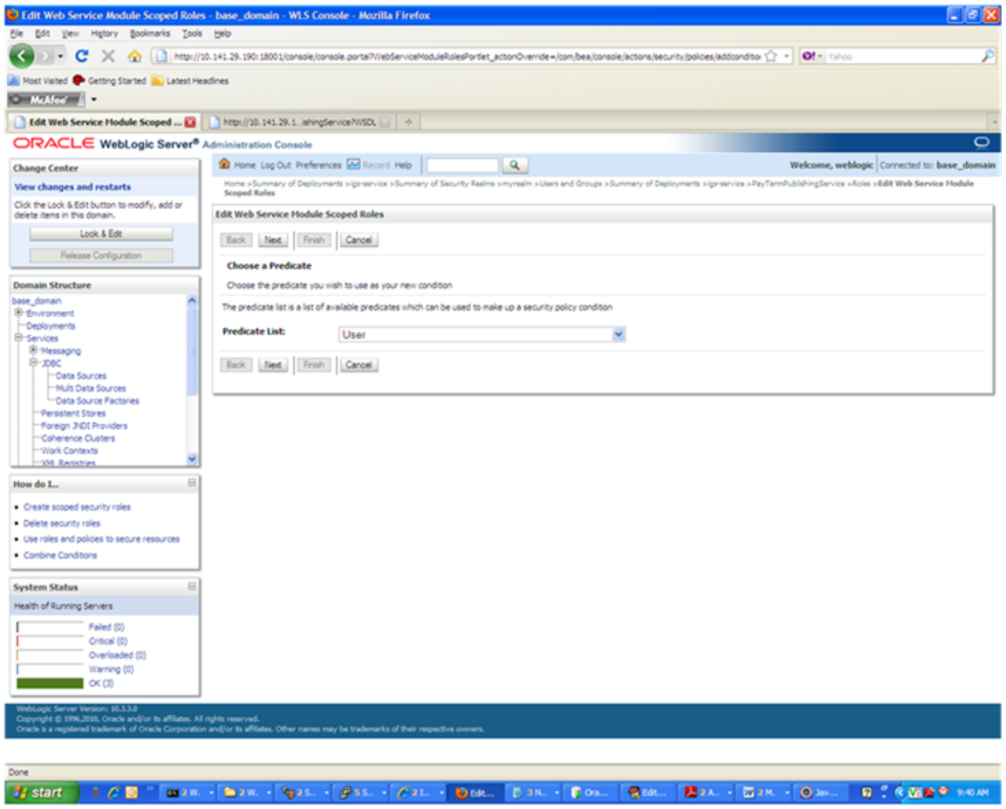
Name: igsrole

The following conditions determine membership in the role.

Role Conditions :

No Policy Specified

11. In the Edit Web Service Module Scoped Roles screen, click **Add Conditions**. The "Choose a Predicate" option is displayed.


Edit Web Service Module Scoped Roles - base_domain - WLS Console - Mozilla Firefox

Home >Summary of Deployments >igs-service >Summary of Security Realms >myrealm >Users and Groups >Summary of Deployments >igs-service >PayTermPublishingService >Roles >**Edit Web Service Module Scoped Roles**

Edit Web Service Module Scoped Roles

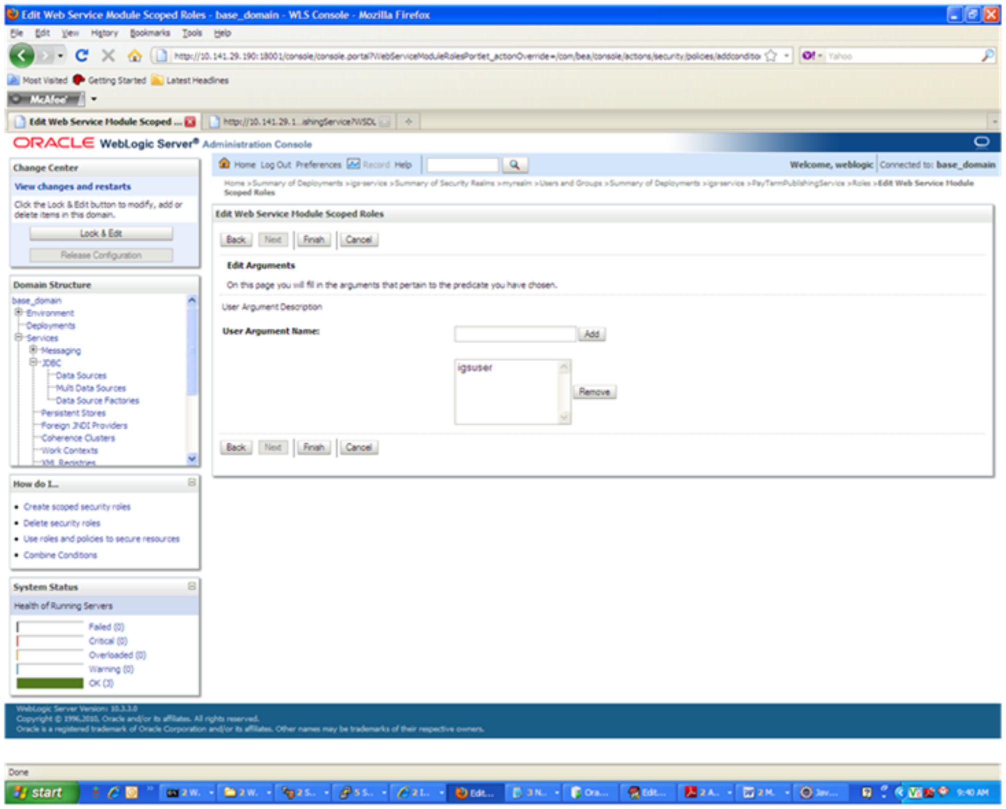
Choose a Predicate

Choose the predicate you wish to use as your new condition

The predicate list is a list of available predicates which can be used to make up a security policy condition

Predicate List: User

12. From the Predicate List, select User. Click **Next**. The Edit Arguments argument is displayed.



Edit Web Service Module Scoped Roles

Edit Arguments

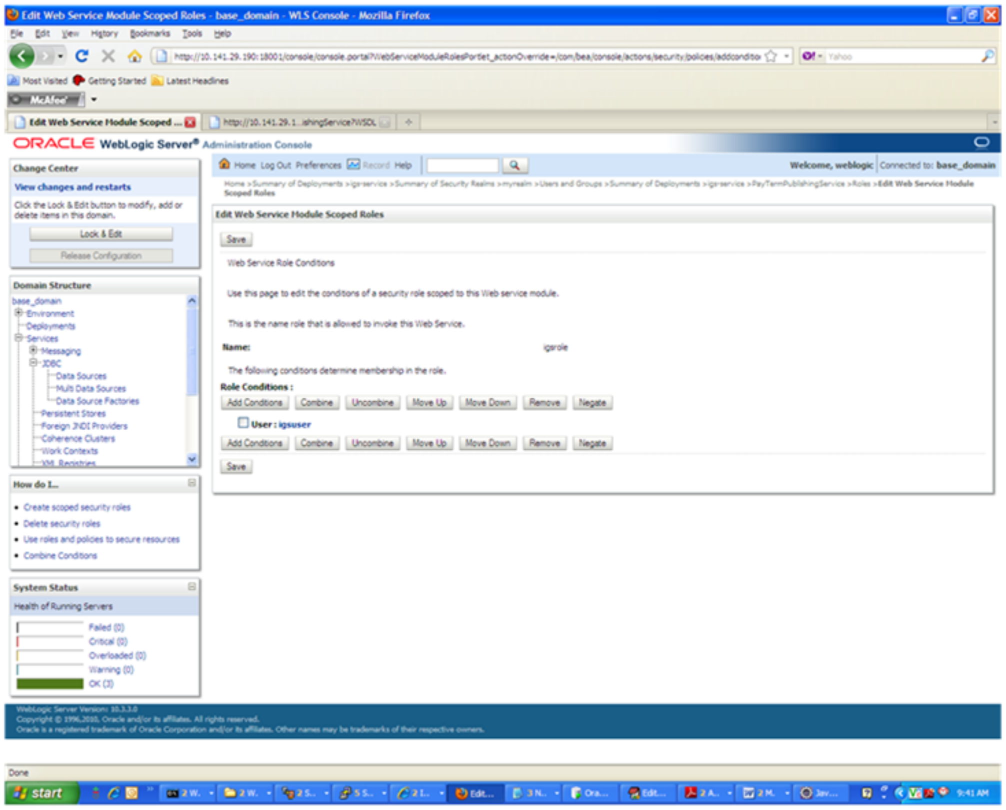
On this page you will fill in the arguments that pertain to the predicate you have chosen.

User Argument Description

User Argument Name:

igsuser

13. In the User Argument Name field, enter the user name created in the security realm. Click **Add**. The name will move down to the box below the Add button. Click **Finish**. The following screen is displayed.


Home >Summary of Deployments >igs-service >Summary of Security Realms >myrealm >Users and Groups >Summary of Deployments >igs-service >PayTermPublishingService >Roles >**Edit Web Service Module Scoped Roles**

Edit Web Service Module Scoped Roles

Web Service Role Conditions

Use this page to edit the conditions of a security role scoped to this Web service module.

This is the name role that is allowed to invoke this Web Service.

Name: igsrole

The following conditions determine membership in the role.

Role Conditions :

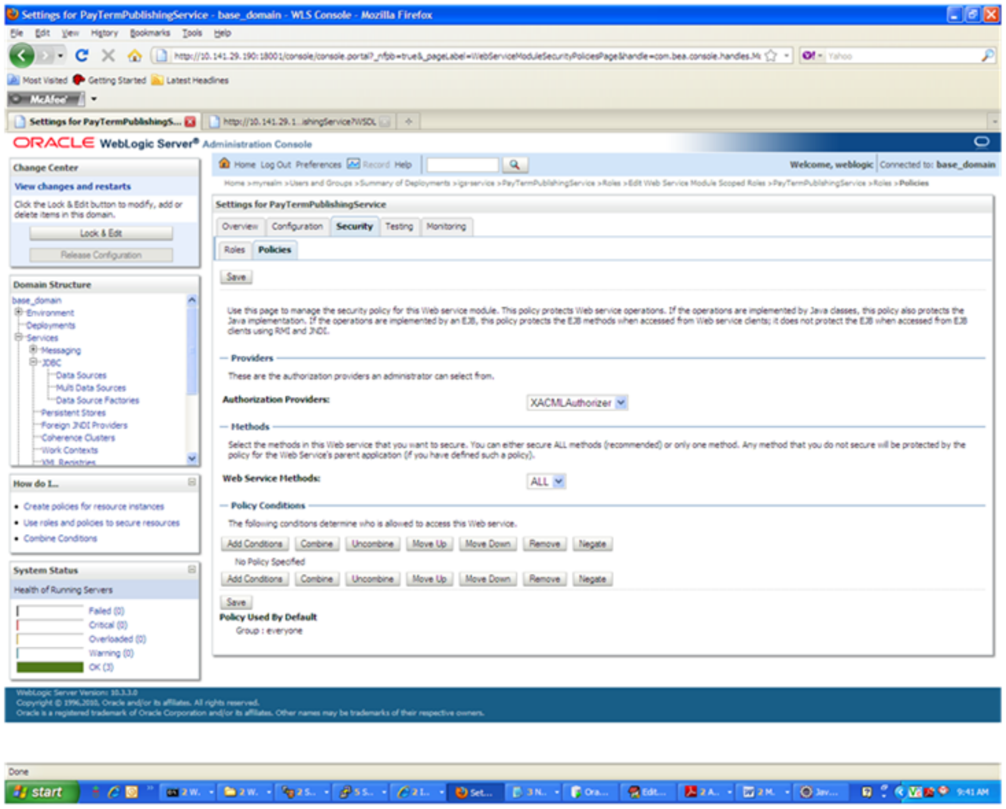
☐ **User : igsuser**

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Edit Web Service Module Scoped Roles - base_domain - WLS Console - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.141.29.190:18001/console/console.portal?WebServiceModuleRolesPortlet_actionOverride=/com/bea/console/actions/security/policies/addconditio
Most Visited Getting Started Latest Headlines
ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Change Center
View changes and restarts
Click the Lock & Edit button to modify, add or delete items in this domain.
Lock & Edit
Release Configuration
Domain Structure
base_domain
Environment
Deployments
Services
Messaging
JDBC
Data Sources
Multi Data Sources
Data Source Factories
Persistent Stores
Foreign JNDI Providers
Coherence Clusters
Work Contexts
XML Registries
How do I...
Create scoped security roles
Delete security roles
Use roles and policies to secure resources
Combine Conditions
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)
Save
Add Conditions
Combine
Uncombine
Move Up
Move Down
Remove
Negate
Done
start

14. Click **Save**. The same screen is displayed with this message near the top: "Changes saved successfully."

15. Return the Security tab of the Web service and click the Policies tab.



Settings for PayTermPublishingService - base_domain - WLS Console - Mozilla Firefox

http://10.141.29.190:18001/console/console.portal?_nfpb=true&_pageLabel=WebServiceModuleSecurityPoliciesPage&handle=com.bea.console.handles.M

Home > myrealm > Users and Groups > Summary of Deployments > igs-service > PayTermPublishingService > Roles > Edit Web Service Module Scoped Roles > PayTermPublishingService > Roles > **Policies**

Settings for PayTermPublishingService

Overview | Configuration | **Security** | Testing | Monitoring

Roles | **Policies**

Use this page to manage the security policy for this Web service module. This policy protects Web service operations. If the operations are implemented by Java classes, this policy also protects the Java implementation. If the operations are implemented by an EJB, this policy protects the EJB methods when accessed from Web service clients; it does not protect the EJB when accessed from EJB clients using RMI and JNDI.

Providers

These are the authorization providers an administrator can select from.

Authorization Providers: XACMLAuthorizer

Methods

Select the methods in this Web service that you want to secure. You can either secure ALL methods (recommended) or only one method. Any method that you do not secure will be protected by the policy for the Web Service's parent application (if you have defined such a policy).

Web Service Methods: ALL

Policy Conditions

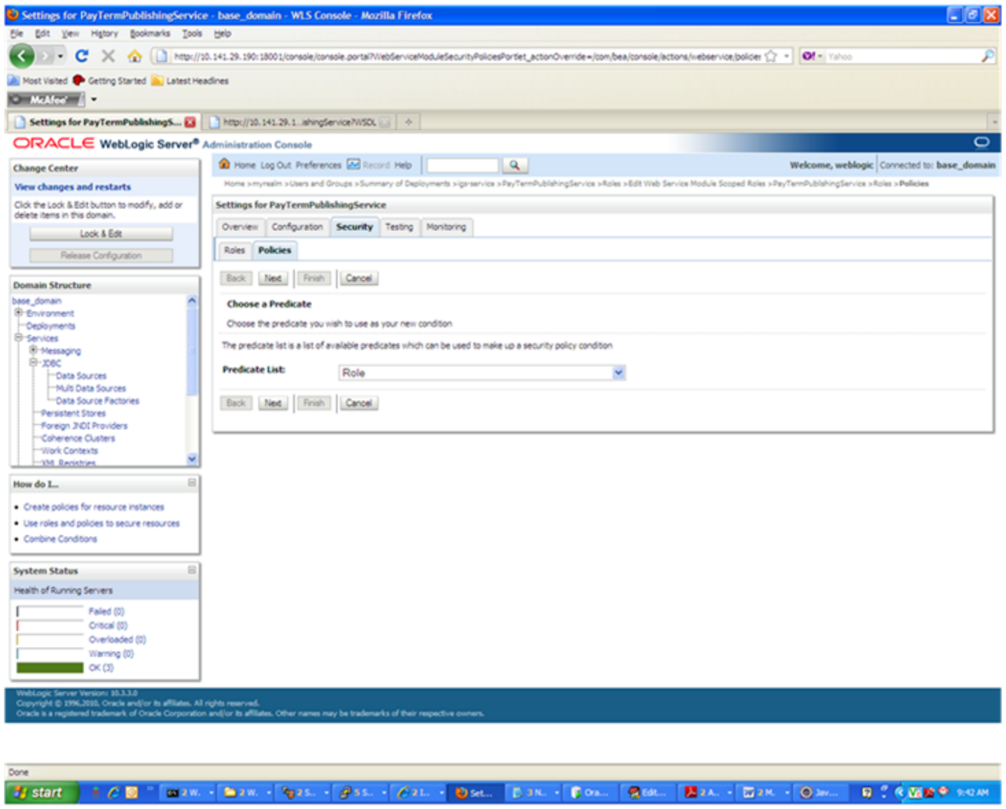
The following conditions determine who is allowed to access this Web service.

No Policy Specified

Policy Used By Default

Group : everyone

16. On the Policies tab, click **Add Conditions**. The "Choose a Predicate" option is displayed.



Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for PayTermPublishingService

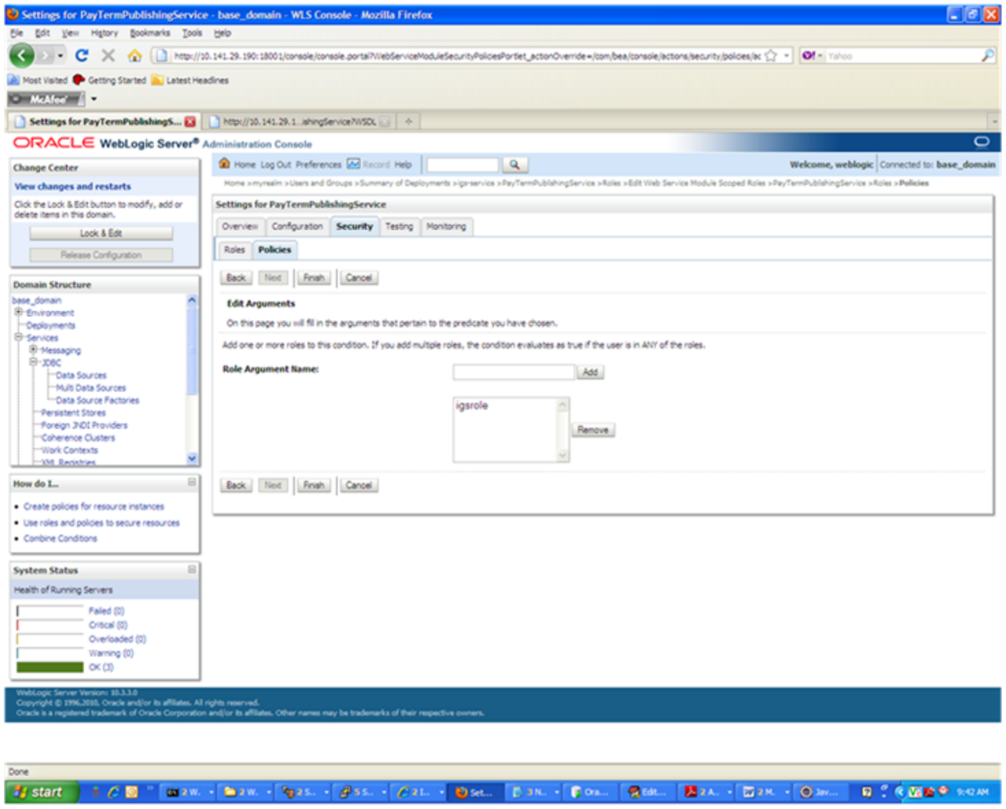
Choose a Predicate

Choose the predicate you wish to use as your new condition

The predicate list is a list of available predicates which can be used to make up a security policy condition

Predicate List: Role

17. From the Predicate List, select Role. Click **Next**. The Edit Arguments option is displayed.



Edit Arguments

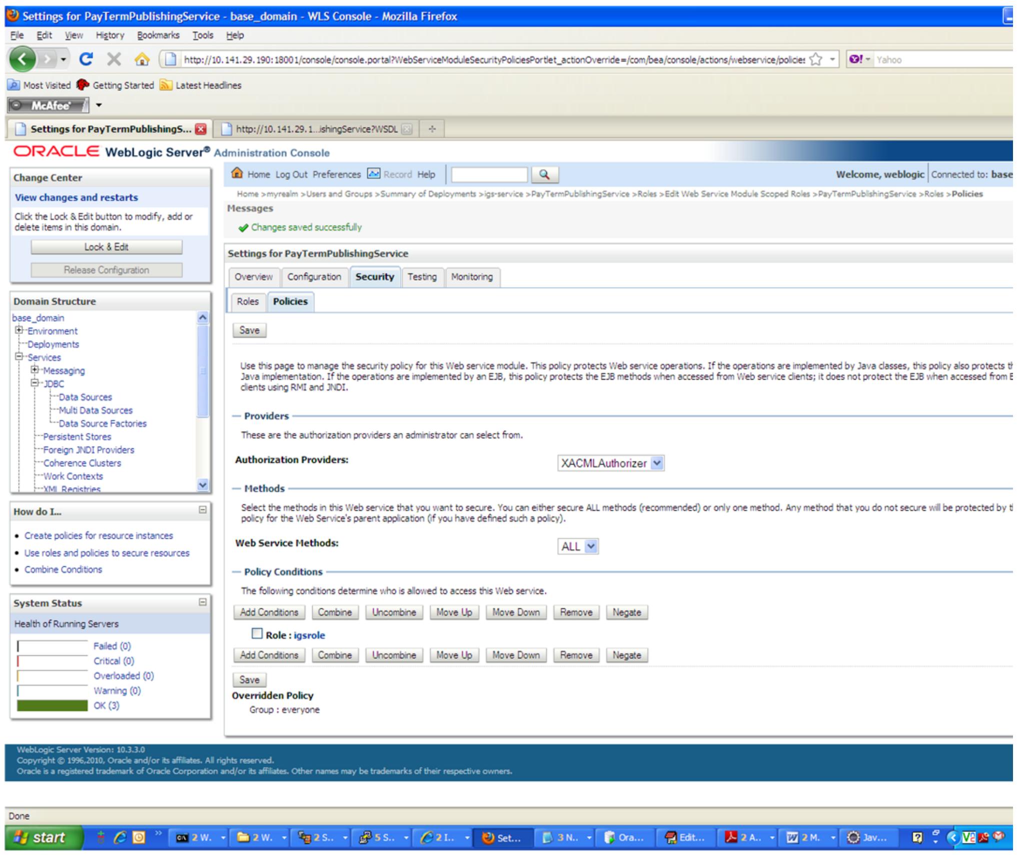
On this page you will fill in the arguments that pertain to the predicate you have chosen.

Add one or more roles to this condition. If you add multiple roles, the condition evaluates as true if the user is in ANY of the roles.

Role Argument Name:

igsrole

18. In the Role Argument Name field, enter the role name created earlier. Click **Add**. The role name will move down to the box below the Add button. Click **Finish** to return to the Policy Conditions screen.
19. Click **Save**. The Policy Conditions screen is displayed with this message near the top: "Changes saved successfully."


Settings for PayTermPublishingService - base_domain - WLS Console - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.141.29.190:18001/console/console.portal?WebServiceModuleSecurityPoliciesPortlet_actionOverride=/com/bea/console/actions/webservice/policies
Most Visited Getting Started Latest Headlines
ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
base_domain
Environment
Deployments
Services
Messaging
JDBC
Data Sources
Multi Data Sources
Data Source Factories
Persistent Stores
Foreign JNDI Providers
Coherence Clusters
Work Contexts
XML Registries
How do I...
Create policies for resource instances
Use roles and policies to secure resources
Combine Conditions
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)
Overview Configuration Security Testing Monitoring
Roles Policies
Save
Add Conditions Combine Uncombine Move Up Move Down Remove Negate
Done

Click the Lock & Edit button to modify, add or delete items in this domain.

Home >myrealm >Users and Groups >Summary of Deployments >igs-service >PayTermPublishingService >Roles >Edit Web Service Module Scoped Roles >PayTermPublishingService >Roles >**Policies**

Messages

✓ Changes saved successfully

Settings for PayTermPublishingService

Use this page to manage the security policy for this Web service module. This policy protects Web service operations. If the operations are implemented by Java classes, this policy also protects th Java implementation. If the operations are implemented by an EJB, this policy protects the EJB methods when accessed from Web service clients; it does not protect the EJB when accessed from E clients using RMI and JNDI.

Providers

These are the authorization providers an administrator can select from.

Authorization Providers: XACMLAuthorizer

Methods

Select the methods in this Web service that you want to secure. You can either secure ALL methods (recommended) or only one method. Any method that you do not secure will be protected by t policy for the Web Service's parent application (if you have defined such a policy).

Web Service Methods: ALL

Policy Conditions

The following conditions determine who is allowed to access this Web service.

☐ **Role :** igsrole

Overridden Policy

Group : everyone

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Client-side Setup for User Name and Password Authentication

The following is sample code for calling a secure IGS Web service.

Note: The following is sample code for invoking the PayTermPublishingService service. When you generate Java consumer for a Web service, the generated jar file contains classes specific to that Web service. Use the appropriate classes in the client code. Service namespace and WSDL location also should be changed accordingly.

```
package com.oracle.retail.rms.client;

import java.net.URL;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;

import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PayTermPubl
ishingPortType;
import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PayTermPubl
ishingService;
import
```

```
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PublishPayT
ermCreateUsingPayTermDesc;
import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PublishPayT
ermCreateUsingPayTermDescResponse;
import com.oracle.retail.integration.base.bo.paytermdesc.v1.PayTermDesc;

import weblogic.wsee.security.unt.ClientUNTCredentialProvider;
import weblogic.xml.crypto.wss.WSSecurityContext;
import weblogic.xml.crypto.wss.provider.CredentialProvider;

import junit.framework.TestCase;


public class PayTermPublishingClient extends TestCase{
       public void testCreatePayTermDesc(){
              try{
                      //qName is namespace of the service
QName qName = new
QName("http://www.oracle.com/retail/igs/integration/services/PayTermPublishingServ
ice/v1"," PayTermPublishingService");

                      // url is the URL of the WSDL of the web service
URL url = new
URL("http://ribhost.example.com:18030/PayTermPublishingBean/PayTermPublishingServi
ce?WSDL");

                     //create an instance of the web service
                     PayTermPublishingServiceservice = new
PayTermPublishingService (url,qName);
                     PayTermPublishingPortType =
service.getPayTermPublishingPort ();

                     //set the security credentials in the service context
                     List credProviders = new ArrayList();
CredentialProvider cp = new ClientUNTCredentialProvider("<rms user>", "<rms
password>");
                     credProviders.add(cp);
Map<String, Object> rc =            ((BindingProvider)port).getRequestContext();
                     rc.put(WSSecurityContext.CREDENTIAL_PROVIDER_LIST,
credProviders);

                     //populate the service method input object
                                  PayTermDesc payTermDesc = new PayTermDesc();
              payTermDesc.setTerms("terms");
              PublishPayTermCreateUsingPayTermDesc payTermCreateDesc = new
PublishPayTermCreateUsingPayTermDesc();
              payTermCreateDesc.setPayTermDesc(payTermDesc);


                    //call the web service
                              PublishPayTermCreateUsingPayTermDescResponse
response =  port.publishPayTermCreateUsingPayTermDesc(payTermCreateDesc,"1");

                   System.out.println("response="+response);
             }catch(Exception e){
                   e.printStackTrace();
             }
      }
}
```

Server-side Setup for Encrypted User Name and Password Token Authentication

WebLogic provides predefined policy files for securing Web services. This section describes the process required to secure a Web service where user name and password are encrypted and signed. Below are the steps to secure the Web service.

1. Follow the steps to attach the policy file to the Web service described in the section, "Attach Policy File to the Web Service," with this exception: In Step 7, select "policy:Wssp1.2-2007-Wss1.1-UsernameToken-Plain-X509-Basic256.xml" (instead of policy:usernametoken.xml). Follow the remaining steps as shown.

   After attaching the policy file, the header for the WSDL of the Web service contains the following.

```
<wsp:UsingPolicy wssutil:Required="true"/>
<wsp:Policy
wssutil:Id="Wssp1.2-2007-Wss1.0-UsernameToken-Plain-X509-Basic256.xml">
<ns1:AsymmetricBinding
xmlns:ns1="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
<wsp:Policy>
<ns1:InitiatorToken>
<wsp:Policy>
<ns1:X509Token
ns1:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/Inc
ludeToken/AlwaysToRecipient">
<wsp:Policy>
<ns1:WssX509V3Token10/>
</wsp:Policy>
</ns1:X509Token>
</wsp:Policy>
</ns1:InitiatorToken>
<ns1:RecipientToken>
<wsp:Policy>
<ns1:X509Token
ns1:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/Inc
ludeToken/Never">
<wsp:Policy>
<ns1:WssX509V3Token10/>
</wsp:Policy>
</ns1:X509Token>
</wsp:Policy>
</ns1:RecipientToken>
<ns1:AlgorithmSuite>
<wsp:Policy>
<ns1:Basic256/>
</wsp:Policy>
</ns1:AlgorithmSuite>
<ns1:Layout>
<wsp:Policy>
<ns1:Lax/>
</wsp:Policy>
</ns1:Layout>
<ns1:IncludeTimestamp/>
<ns1:ProtectTokens/>
<ns1:OnlySignEntireHeadersAndBody/>
</wsp:Policy>
</ns1:AsymmetricBinding>
<ns2:SignedEncryptedSupportingTokens
xmlns:ns2="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
<wsp:Policy>
<ns2:UsernameToken
```

```
ns2:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/Inc
ludeToken/AlwaysToRecipient">
<wsp:Policy>
<ns2:WssUsernameToken10/>
</wsp:Policy>
</ns2:UsernameToken>
</wsp:Policy>
</ns2:SignedEncryptedSupportingTokens>
<ns3:Wss10
xmlns:ns3="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
<wsp:Policy>
<ns3:MustSupportRefKeyIdentifier/>
<ns3:MustSupportRefIssuerSerial/>
</wsp:Policy>
</ns3:Wss10>
</wsp:Policy>
```

2. The key combination used by the client to sign the message is a valid one for the server. The client certificate must be signed with a certificate authority that is trusted by the server.
3. WebLogic instances include a demo CA. The certificate and key for it is in $WLS_HOME/Middleware/wlserver/server/lib/CertGenCA.der and CertGenCAKey.der. The key does not appear to change between WebLogic installations and is trusted by the default DemoTrust store. For this reason, the DemoTrust store must never be enabled in a production environment. Otherwise anybody can become "trusted" fairly easily.
4. WebLogic CertGen command can be used for generating keys of the correct key length and signing them with the demo CA noted above. A client certification/key pair is required to sign the outgoing message and server certificate to encrypt the critical information.

   ```
   java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
   utils.CertGen -certfile ClientCert -keyfile ClientKey -keyfilepass ClientKey
   -cn <rms user>
   ```

   The above command generates the following files.

   - ClientCert.der
   - ClientCert.pem
   - ClientKey.der
   - ClientKey.pem

   The user name is <*rms user*>. Replace it with the user name of the user who will access the Web service.
5. The command below generates the four files that follow it.

   ```
   java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
   utils.CertGen -certfile ServerCert -keyfile ServerKey -keyfilepass ServerKey
   -cn <rms user>
   ```

   - ServerCert.der
   - ServerCert.pem
   - ServerKey.der
   - ServerKey.pem

The user name is <*rms user*>. Replace it with user name of the user who will access the Web service

6. Using the following commands, import the files into key stores.

```
java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile ClientCert.der -keyfile ClientKey.der
-keyfilepass <Client Key Password> -keystore ClientIdentity.jks -storepass
<Client Key Password> -alias identity - keypass <Client Key Password>

java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile ServerCert.der -keyfile ServerKey.der
-keyfilepass <Server Key Password> -keystore ServerIdentity.jks -storepass
<Server Key Password> -alias identity - keypass <Server Key Password>
```

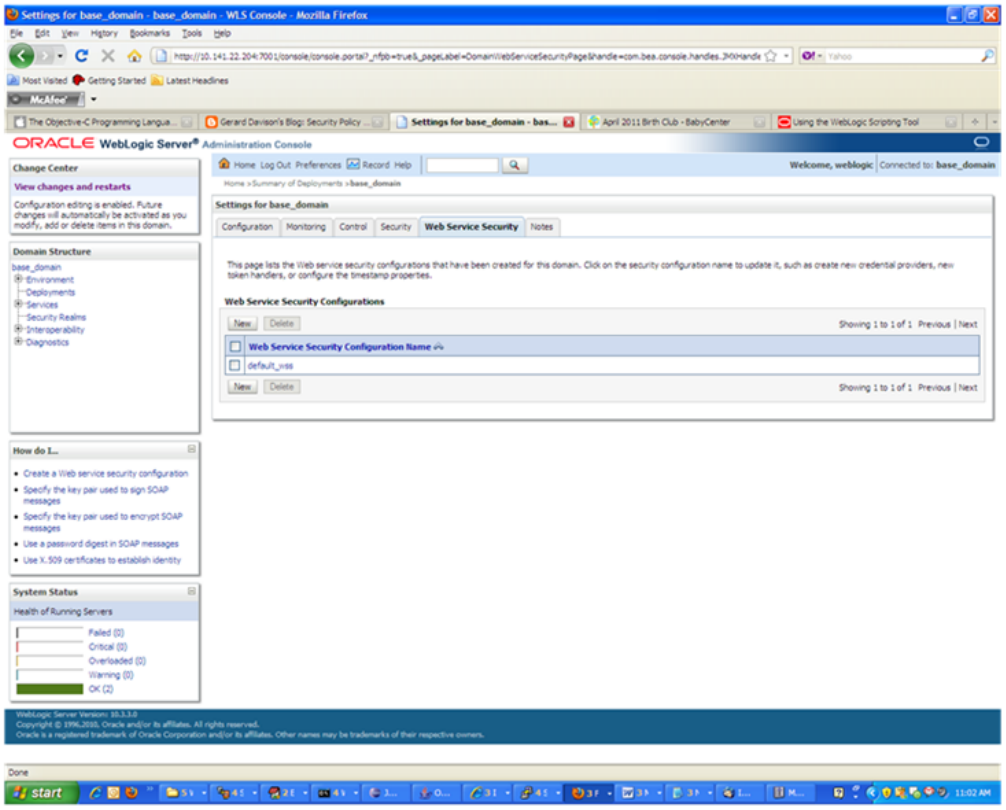
7. Using the script in Appendix: configWss.py, configure the WebLogic server to use the key. Copy the script and save it in the location from which it will run.

```
Java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
weblogic.WLST configWss.py <weblogicuser> <weblogicpassword> <weblogichost>
<weblogic admin port> ServerIdentity.jks ServerKey identity ServerKey
```

For example:

```
Java -classpath $WLS_HOME/Middleware/wlserver/server/lib/weblogic.jar
weblogic.WLST configWss.py <weblogic user> <weblogic password> localhost
7001/home/wls/ServerIdentity.jks ServerKey identity ServerKey
```

8. In the WebLogic logic console, check the Web Service Security tab to verify that the command ran properly. Note that the default_ww configuration is used for all Web services unless otherwise indicated.



ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Home >Summary of Deployments >base_domain
Configuration Monitoring Control Security Web Service Security Notes
New Delete

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

base_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...
- Create a Web service security configuration
- Specify the key pair used to sign SOAP messages
- Specify the key pair used to encrypt SOAP messages
- Use a password digest in SOAP messages
- Use X.509 certificates to establish identity

System Status

Health of Running Servers

Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)

Settings for base_domain

This page lists the Web service security configurations that have been created for this domain. Click on the security configuration name to update it, such as create new credential providers, new token handlers, or configure the timestamp properties.

Web Service Security Configurations

Showing 1 to 1 of 1 Previous | Next

Web Service Security Configuration Name
default_wss

Showing 1 to 1 of 1 Previous | Next

WebLogic Server Version: 10.3.3.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

9. After the certificate setup is completed for the Web service, follow the steps in the "Create Roles and Users" section to create a user in WebLogic to access the Web service.
10. Restart the server. Create a client to invoke the Web service.

Client-side Setup for Encrypted User Name and Password Token Authentication

Below is sample code for calling a Web service that is secured using the policy file, policy:Wssp1.2-2007-Wss1.1-UsernameToken-Plain-X509-Basic256.xm.:

```
package com.test;
import java.net.URL;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.WebServiceRef;

import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PayTermPubl
ishingPortType;
import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PayTermPubl
ishingService;
import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PublishPayT
ermCreateUsingPayTermDesc;
import
com.oracle.retail.igs.integration.services.paytermpublishingservice.v1.PublishPayT
ermCreateUsingPayTermDescResponse;
import com.oracle.retail.integration.base.bo.paytermdesc.v1.PayTermDesc;

import weblogic.security.SSL.TrustManager;
import weblogic.wsee.security.bst.ClientBSTCredentialProvider;
import weblogic.wsee.security.unt.ClientUNTCredentialProvider;
import weblogic.wsee.security.util.CertUtils;
import weblogic.xml.crypto.wss.WSSecurityContext;
import weblogic.xml.crypto.wss.provider.CredentialProvider;
public class Client {
public static void main(String args[]){
try{
//qName is namespace of the service
QName qName = new
QName("http://www.oracle.com/retail/igs/integration/services/PayTermPublishingServ
ice/v1"," PayTermPublishingService");

// url is the URL of the WSDL of the web service
URL url = new
URL("http://<host>:<port>/PayTermPublishingBean/PayTermPublishingService?WSDL");

//create an instance of the web service
PayTermPublishingServiceservice = new PayTermPublishingService(url,qName);
PayTermPublishingPortType = service.getPayTermPublishingPort ();
PayTermDesc payTermDesc = new PayTermDesc();
payTermDesc.setTerms("terms");
PublishPayTermCreateUsingPayTermDesc payTermCreateDesc = new
PublishPayTermCreateUsingPayTermDesc();
payTermCreateDesc.setPayTermDesc(payTermDesc);
String serverCertFile = "D:/head/retail-soa-enabler/dist/client/ServerCert.der";
String clientKeyStore =
"D:/head/retail-soa-enabler/dist/client/ClientIdentity.jks";
String clientKeyStorePass = "ClientKey";
String clientKeyAlias = "identity";
String clientKeyPass = "ClientKey";
List credProviders = new ArrayList();
ClientUNTCredentialProvider unt =  new "<rms user>", "<rms password>";
credProviders.add(unt);
final X509Certificate serverCert =
(X509Certificate)CertUtils.getCertificate(serverCertFile);
serverCert.checkValidity();
CredentialProvider cp = new ClientBSTCredentialProvider(clientKeyStore,
clientKeyStorePass,clientKeyAlias, clientKeyPass, "JKS", serverCert);
credProviders.add(cp);
Map requestContext = ((BindingProvider)port).getRequestContext();
requestContext.put(WSSecurityContext.CREDENTIAL_PROVIDER_LIST, credProviders);
requestContext.put(WSSecurityContext.TRUST_MANAGER,    new TrustManager() {
public boolean certificateCallback(X509Certificate[] chain,int validateErr) {
boolean result = chain[0].equals(serverCert);
return result;
}
});
PublishPayTermCreateUsingPayTermDescResponse response =
port.publishPayTermCreateUsingPayTermDesc(payTermCreateDesc,"1");
System.out.println("response="+response);
}catch(Exception e){
e.printStackTrace();
}
}
}
```

13 RIB Security

This chapter explains how to securely configure Oracle Retail Integration Bus applications and related tools.

Security in RIB Application Builder

RIB Application Builder is a tool for building and deploying RIB applications on the WebLogic server. The rib-deployment-env-info.xml file is the single source of all values used in the RIB App Builder tools. It is the only (or should be the only) file that requires editing.

This file contains all the configuration information required for building RIB applications. Below is a sample for AQ configuration details:

```
<aq-jms-server jms-server-id="jms1">
<jms-server-home>linux1@linux1:/home/oracle/oracle/product/12.1.0.2/db_
1</jms-server-home>
<jms-url>jdbc:oracle:thin:@linux1:1521:ora12c</jms-url>
<jms-port>1521</jms-port>
<jms-user-alias>jms1_user-name-alias</jms-user-alias>
</aq-jms-server>
```

This file does not contain the user name and password for connecting to the application server or databases. Rather, it contains the alias for each user name/password combination. This alias refers to the user name/password stored in a secured wallet file. The wallet file is created when the user runs the application assembly tool during the RIB application building process.

The syntax for the application assembly command is as follows:

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

The argument, setup-security-credential, must be used when running the rib-app-compiler for the first time. It prompts the user to enter user names and passwords required to install RIB components. It stores details as credentials in a wallet file inside the rib-home/deployment-home/conf/security/ directory. The credentials are retrieved and used by the deployer when installing RIB components.

Only the operating system user who created the wallet file with the RIB application assembly tool has read and write access to the file. Other users do not have permission to access the file. The file permissions are set up during the post-deployment phase for RIB applications.

See the "Application Builder" chapter in the *Oracle Retail Integration Bus Operations Guide* for details about the RIB Application Builder.

Note: Users also can change user names and passwords for RIB applications after deploying them. Refer to the section, "setup-security-credential," under "RIB App Builder Tools" in the "Application Builder" chapter in *Oracle Retail Integration Bus Operations Guide* for how to change RIB user names and passwords after deployment.

Security during RIB Deployment Process

Users can run the RIB application assembly tool to build RIB application .ear files. The generated .ear files contain deployment descriptors for data sources used by RIB runtime to connect to the application database and the error hospital database. The deployment descriptors contain the user name for accessing the database, but the passwords are not stored there. During the deployment process for the RIB application, the passwords are read from the wallet file and encrypted using a WebLogic utility. The encrypted passwords are added in a WebLogic deployment plan that is uploaded on the server along with the .ear file.

Security during RIB Runtime

During the runtime process, the RIB application must make JMX calls to the JMX server. WebLogic instance user name and password are required to make connections to the JMX server. This information is stored in a secured wallet file, the path to which is stored in the rib-system.properties file.

For information about the properties in rib-system.properties file, see the "rib-system.properties" section in the "Backend System Administration and Logging" chapter of the *Oracle Retail Integration Bus Operations Guide*.

Only the operating system user who created them has read and write access to the properties files created during the RIB application deployment process. Other users do not have permission to access the files. Permissions are granted during the post deployment phase for RIB applications.

RIB Administration Security

There are two categories of administrators in RIB: RIB System Administrators and RIB Application Administrators. The defined realms, roles, and users differ according to administrator type.

RIB System Administrators install, configure, and deploy defect fixes—and make sure that integration infrastructure is up and running properly.

RIB Application Administrators handle the business side of the integration system. Primarily, they bring RIB adapters up or down and fix data issues with message payloads through RIHA.

RIB Application Administrators Security Domain

The WebLogic server has a default security realm. For each rib-<app>.ear deployed, RIB creates a user in the default security realm. By default, RIB creates a user that belongs to the ribAdminGroup and administrators groups. RIB system administrators can manage rib-<app> application users and access control through the WebLogic Server Administration Console. The default group and user that RIB creates must not be deleted or modified.

The user created in ribAdminGroup has access to the RIB administration GUI. When a RIB application administrator tries to access the RIB administration GUI, a basic authentication screen is displayed, where the user must provide a user name and password for authentication. The user name must be the same as the one created by RIB in ribAdminGroup. When the credentials are verified, the RIB administration GUI home page is displayed.

Multiple User Configuration

To create new users to logon to the RIB Administration GUI, follow these steps:

1. Login to the WebLogic console and navigate to Home >Summary of Security Realms >myrealm >Users and Groups location.
2. Create a new user, for example: testuser.
3. Navigate to the details of the new user.
4. On the Groups tab, choose the created group(ribAdminGroup) from list.

   For example: Home > Summary of Security Realms > Summary of Deployments > Summary of Security Realms > myrealm > Users and Groups > testuser.

RIB System Administrators Security Domain

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

Note: For details on external LDAP configurations, see Appendix A in the *Oracle Retail Integration Bus Implementation Guide*.

Security in RIHA

Oracle Retail Integration Bus Hospital Administration or RIB Hospital Administration (RIHA) is a tool to manage RIB messages in the RIB hospital error tables. It is a Web application that is deployable on the WebLogic server.

For how to set up security for RIHA, see the "Security Setup Guidelines" section in the *Oracle Retail Integration Bus Hospital Administration Guide*.

Security in RDMT

The RIB Diagnostic and Monitoring Took Kit (RDMT) is a collection of command line tools for controlling and monitoring RIB applications. When used from within rib-home, RDMT loads configuration information from the rib-deployment-env-info.xml file. For user name and password information, it reads the wallet file created during the RIB application assembly process.

For information about RDMT, see the "Diagnostic and Monitoring Tools" chapter in the *Oracle Retail Integration Bus Operations Guide*.

Security in PL/SQL Application API Stubs

The plsql-api-stubs is an API simulator designed to act as though RIB is connected to the application, but it can process specific status and other parameters from a "stubbed" application. This set of tools is designed to emulate those applications exposing PL/SQL APIs to RIB, such as RMS, ORFM, and RWMS. The tool reads and writes the user name and password for connecting to the database in a secured wallet file.

Security in Integration Gateway Services

The RIB Integration Gateway Services (IGS) component is a set of standard Simple Object Access Protocol (SOAP) based Web services that provide access to the RIB infrastructure. These Web services are generated using the Oracle Retail Service Enabler Tool. They should be secured after being deployed. For information, see "Secure IGS Web Services Using the Administration Console."

SSL Configuration

Secure Sockets Layer (SSL) provides secure connections by allowing two applications connecting over a network to authenticate each other's identity and encrypting the data exchanged between the applications. Configuring SSL in WebLogic servers in production environments is recommended. See WebLogic documentation for how to configure SSL in WebLogic. Below is the link to documentation for configuring SSL in WeblLogic 12.2.1 server:

https://docs.oracle.com/middleware/1221/wls/SECMG/ssl.htm#CIHBDHEG

Deployment of RIB applications over SSL protocol is supported now by giving protocol values as https in deployment info xml.

Below are the steps for running RIB in SSL environment.

1. Configure SSL in the WebLogic server. (See WebLogic documentation for detailed steps.)
2. Keep the SSL ports of the WebLogic server instances open for RIB deployment. Verify that the SSL port is open: In the WebLogic administration console, go to the Configuration > General page of the server instance. Verify that the "Listen Port Enabled" checkbox is checked and provide listen address to all managed servers/admin server.
3. Make sure that the rib-deployment-env-info.xml file has protocol specified as https and port numbers are https port numbers for WebLogic server instances.
4. While starting managed servers, provide admin server.

   For example: startManagedServer.sh rib-rob-server https://host:port
5. Deploy the RIB applications.
6. Restart the WebLogic managed servers.
7. If required, non-SSL ports can be disabled as follows. In the WebLogic administration console go to the Configuration > General page of the server instance. Uncheck the "Listen Port Enabled" checkbox and check the "SSL Listen Port Enabled" checkbox. This is an optional step and must be done only when all communications with the server are over HTTPS protocol.

Note: Due to known vulnerabilities, Oracle recommends disabling SSLv3 in all products. We recommend using the TLSv1.2 protocol. WebLogic server can be configured to use the TLSv1.2 protocol by adding the following line in the setDomainEnv.sh. Restart the server after making the change.

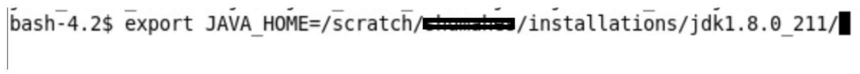
JAVA_OPTIONS=" $JAVA_OPTIONS
-DwebLogic.security.SSL.minimumProtocolVersion=TLSv1.2"

A
Appendix: RIB Application Builder Tools

Perform the following steps to install the RIB application.

1. Make sure that the JAVA_HOME environment variable is set for the user that will be performing these tasks.

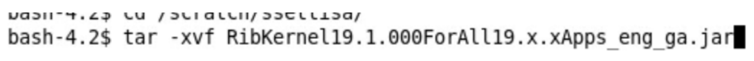
   `echo $JAVA_HOME /usr/bin/java/jdk1.8.0_64`
   Example: export JAVA_HOME=/usr/bin/java/jdk1.8.0_64



```
bash-4.2$ export JAVA_HOME=/scratch/[redacted]/installations/jdk1.8.0_211/
```

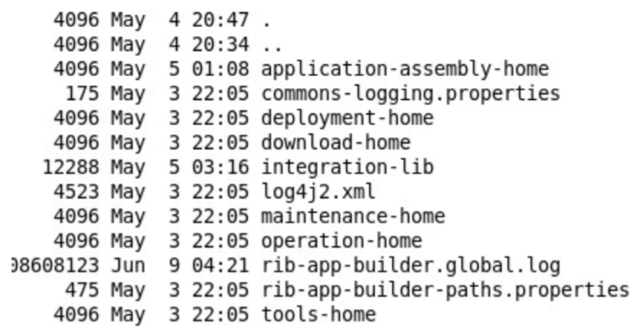
2. Make sure that all RIB WebLogic instances that are to be deployed to are running.
3. Download and extract the RibKer-nel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar.

   Copy the latest version to the rib-app-builder and then extract it to build your "rib-home." This "rib-home" will be the directory from where you will perform "all" the rib-<app> related tasks from now on.



```
bash-4.2$ cd /scratch/ssettisa/
bash-4.2$ tar -xvf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar
```

   directory structure of "rib-home":



```
    4096 May  4 20:47 .
    4096 May  4 20:34 ..
    4096 May  5 01:08 application-assembly-home
     175 May  3 22:05 commons-logging.properties
    4096 May  3 22:05 deployment-home
    4096 May  3 22:05 download-home
   12288 May  5 03:16 integration-lib
    4523 May  3 22:05 log4j2.xml
    4096 May  3 22:05 maintenance-home
    4096 May  3 22:05 operation-home
 8608123 Jun  9 04:21 rib-app-builder.global.log
     475 May  3 22:05 rib-app-builder-paths.properties
    4096 May  3 22:05 tools-home
```

4. Download the RibFuncArti-fact<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar and put it in rib-home/download-home/rib-func-artifacts directory.

Note: Do not extract the tar file. This will be done by the check-version-and-unpack tool.

Example: Directory structure of rib-func-artifacts directory:

```
   4096 May  4 20:47 .
   4096 May  3 22:05 ..
   4096 May  4 20:47 archive
     69 May  3 22:05 README.txt
7936000 May  4 20:42 RibFuncArtifact19.1.000ForAll19.1.000Apps_eng_ga.tar
```

5. Download the RibPak<RIB_MAJOR_VERSION>For<RETAIL_APP_NAME><RETAIL_APP_VERSION>_eng_ga.tar and put it in rib-home/download-home/all-rib-apps directory.

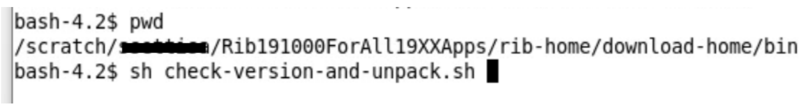
   Note: Do not extract the tar file. This will be done by the check-version-and-unpack tool.

   Example: Directory structure of rib-func-artifacts directory:

```
 4096 May  5 01:07 .
 4096 May  3 22:05 ..
 4096 May  5 01:08 archive
  205 May  3 22:05 README.txt
51200 May  5 01:05 RibPak19.1.000ForExt19.1.000_eng_ga.tar
10240 May  3 21:56 RibPak19.1.000ForRce19.1.000_eng_ga.tar
81920 May  4 20:41 RibPak19.1.000ForRms19.1.000_eng_ga.tar
10240 May  5 01:06 RibPak19.1.000ForRsys19.1.000_eng_ga.tar
```

6. Run the rib-home/download-home/bin/check-version-and-unpack.sh script from rib-home/download-home/bin directory.

   This script verifies the version compatibility between the packs and extract the files if they are compatible.



```
bash-4.2$ pwd
/scratch/[redacted]/Rib191000ForAll19XXApps/rib-home/download-home/bin
bash-4.2$ sh check-version-and-unpack.sh
```

7. Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml file to specify the deployment environment information.

   See the "Information to Gather for Installation in Remote Server" section before starting the edit. This file (rib-deployment-env-info.xml) is the only file that the user has to edit. See the "Rib-app-builder documentation" for details and examples.

   The XML file has four major sections.

   - app-in-scope-for-integration section:

     In this section you define what applications are in scope for this environment.

   - rib-jms-server section:

     In this section you define the JMS server information.

     Note: See also"Preinstallation Steps for Multiple JMS Server Setup" in Chapter 4 of this guide.

   - rib-javaee-containers section:

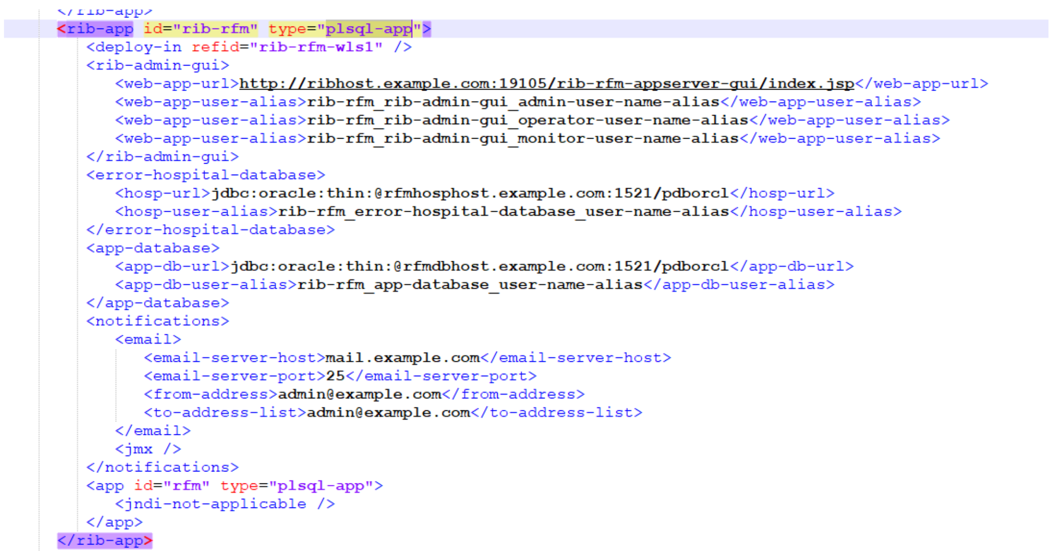
In this section you define the "Java EE container information" for each of the rib-<app> that are in scope.

- rib-applications section:

  In this section you define the rib-<app> specific information for each of the rib-<app> that are in scope.

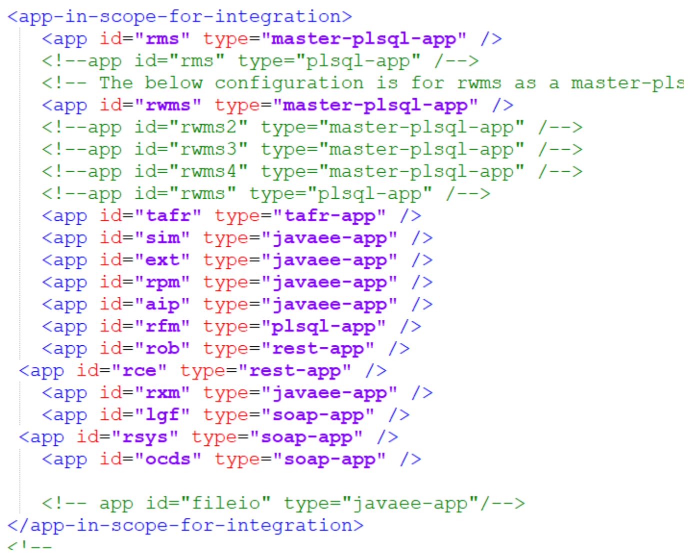
For PL/SQL applications you must define RIB Hospital connection and email notification information.

For Java EE applications you will need to define RIB Hospital connection, email notification information and the connecting retail application's (for example, <app>) JNDI information.



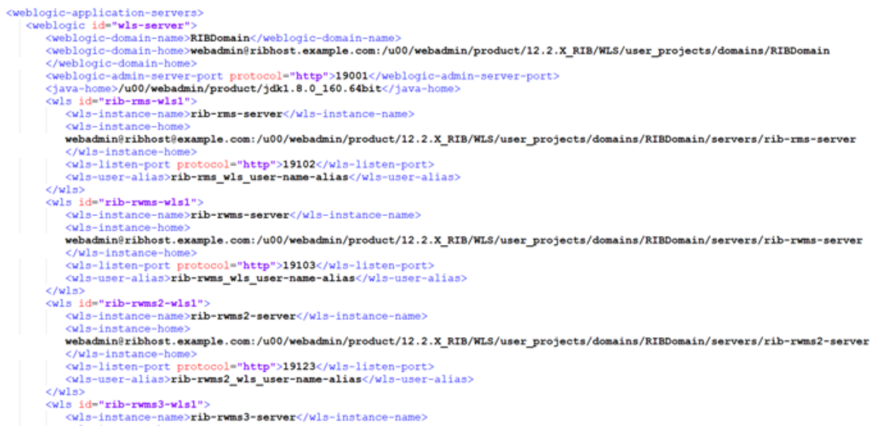
```
</rib-app>
<rib-app id="rib-rfm" type="plsql-app">
   <deploy-in refid="rib-rfm-wls1" />
   <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:19105/rib-rfm-appserver-gui/index.jsp</web-app-url>
      <web-app-user-alias>rib-rfm_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rfm_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rfm_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
   <error-hospital-database>
      <hosp-url>jdbc:oracle:thin:@rfmhosphost.example.com:1521/pdborcl</hosp-url>
      <hosp-user-alias>rib-rfm_error-hospital-database_user-name-alias</hosp-user-alias>
   </error-hospital-database>
   <app-database>
      <app-db-url>jdbc:oracle:thin:@rfmdbhost.example.com:1521/pdborcl</app-db-url>
      <app-db-user-alias>rib-rfm_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="rfm" type="plsql-app">
      <jndi-not-applicable />
   </app>
</rib-app>
```

Edit the app-in-scope-for-integration section to match the desired deployment. Define what applications are in scope for this environment.



```
<app-in-scope-for-integration>
   <app id="rms" type="master-plsql-app" />
   <!--app id="rms" type="plsql-app" /-->
   <!-- The below configuration is for rwms as a master-pls
   <app id="rwms" type="master-plsql-app" />
   <!--app id="rwms2" type="master-plsql-app" /-->
   <!--app id="rwms3" type="master-plsql-app" /-->
   <!--app id="rwms4" type="master-plsql-app" /-->
   <!--app id="rwms" type="plsql-app" /-->
   <app id="tafr" type="tafr-app" />
   <app id="sim" type="javaee-app" />
   <app id="ext" type="javaee-app" />
   <app id="rpm" type="javaee-app" />
   <app id="aip" type="javaee-app" />
   <app id="rfm" type="plsql-app" />
   <app id="rob" type="rest-app" />
 <app id="rce" type="rest-app" />
   <app id="rxm" type="javaee-app" />
   <app id="lgf" type="soap-app" />
 <app id="rsys" type="soap-app" />
   <app id="ocds" type="soap-app" />

   <!-- app id="fileio" type="javaee-app"/-->
</app-in-scope-for-integration>
<!--
```

Edit the application server section.

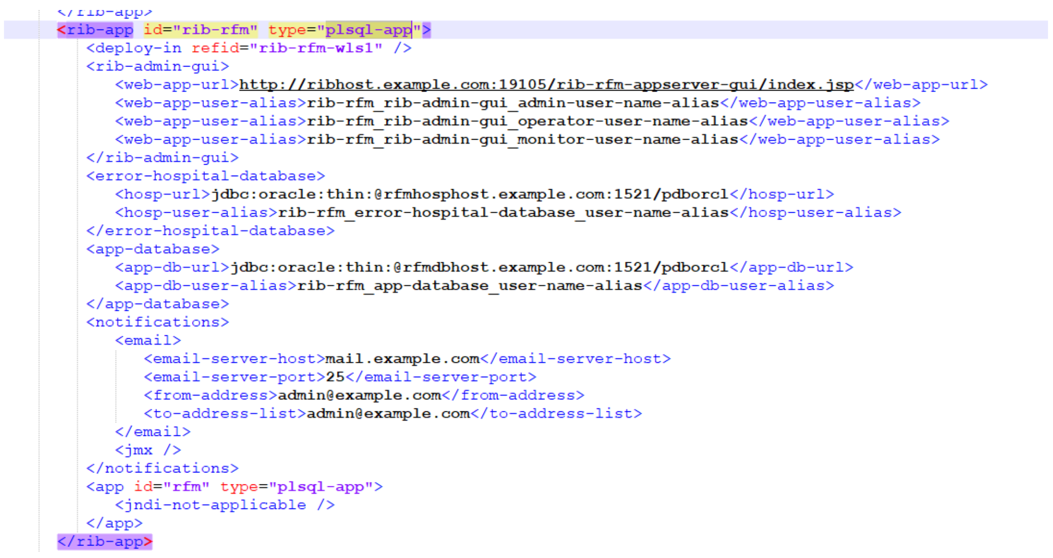


```
<weblogic-application-servers>
  <weblogic id="wls-server">
    <weblogic-domain-name>RIBDomain</weblogic-domain-name>
    <weblogic-domain-home>webadmin@ribhost.example.com:/u00/webadmin/product/12.2.X_RIB/WLS/user_projects/domains/RIBDomain
    </weblogic-domain-home>
    <weblogic-admin-server-port protocol="http">19001</weblogic-admin-server-port>
    <java-home>/u00/webadmin/product/jdk1.8.0_160.64bit</java-home>
    <wls id="rib-rms-wls1">
      <wls-instance-name>rib-rms-server</wls-instance-name>
      <wls-instance-home>
      webadmin@ribhost@example.com:/u00/webadmin/product/12.2.X_RIB/WLS/user_projects/domains/RIBDomain/servers/rib-rms-server
      </wls-instance-home>
      <wls-listen-port protocol="http">19102</wls-listen-port>
      <wls-user-alias>rib-rms_wls_user-name-alias</wls-user-alias>
    </wls>
    <wls id="rib-rwms-wls1">
      <wls-instance-name>rib-rwms-server</wls-instance-name>
      <wls-instance-home>
      webadmin@ribhost.example.com:/u00/webadmin/product/12.2.X_RIB/WLS/user_projects/domains/RIBDomain/servers/rib-rwms-server
      </wls-instance-home>
      <wls-listen-port protocol="http">19103</wls-listen-port>
      <wls-user-alias>rib-rwms_wls_user-name-alias</wls-user-alias>
    </wls>
    <wls id="rib-rwms2-wls1">
      <wls-instance-name>rib-rwms2-server</wls-instance-name>
      <wls-instance-home>
      webadmin@ribhost.example.com:/u00/webadmin/product/12.2.X_RIB/WLS/user_projects/domains/RIBDomain/servers/rib-rwms2-server
      </wls-instance-home>
      <wls-listen-port protocol="http">19123</wls-listen-port>
      <wls-user-alias>rib-rwms2_wls_user-name-alias</wls-user-alias>
    </wls>
    <wls id="rib-rwms3-wls1">
      <wls-instance-name>rib-rwms3-server</wls-instance-name>
```

Configure the rib-applications section:

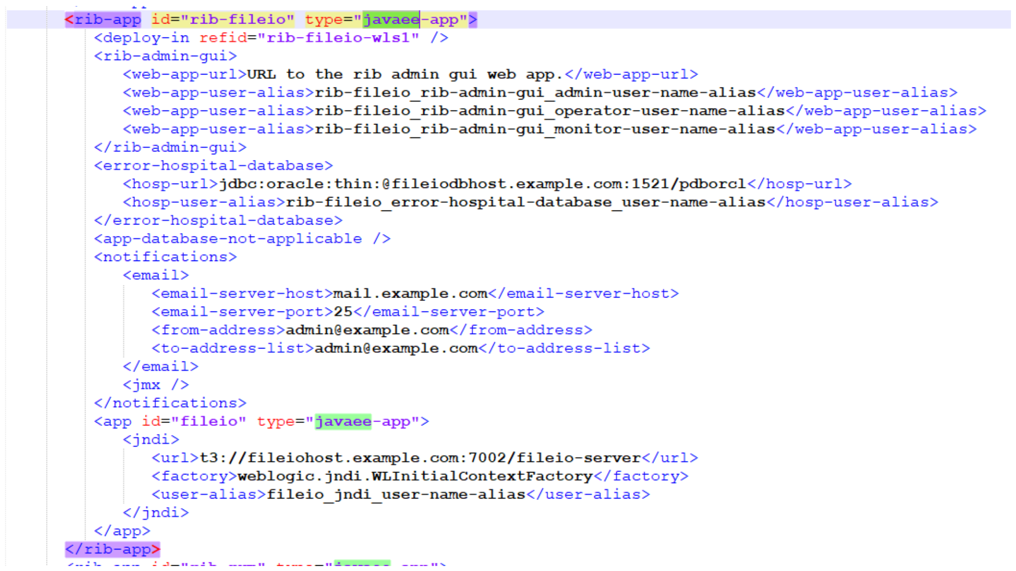
In this section you define the rib-<app> specific information for each rib-<app> that in scope.

For PL/SQL applications you must define the RIB Hospital connection, application database connections, and email notification information.



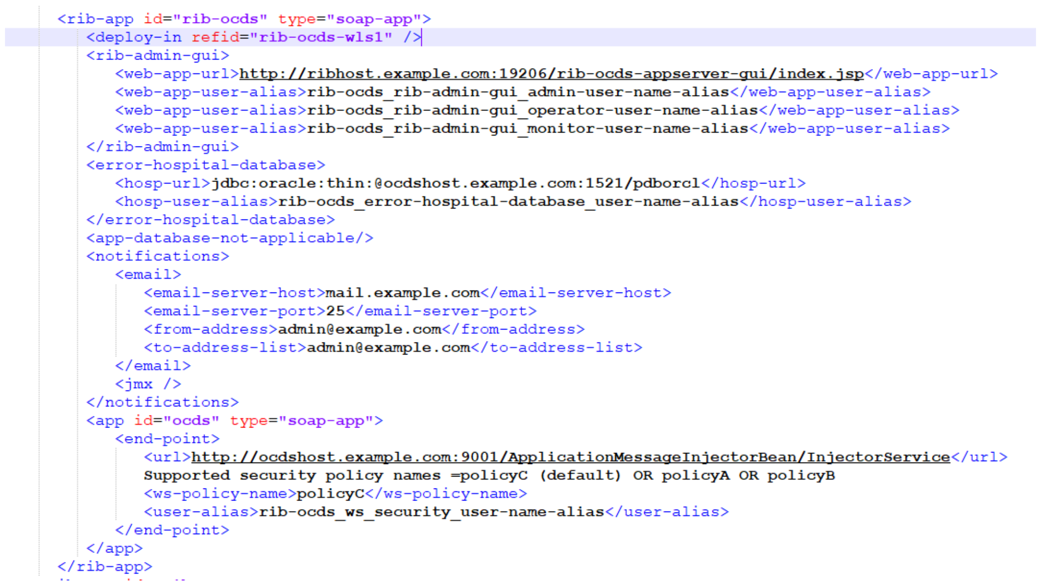
```
</rib-app>
<rib-app id="rib-rfm" type="plsql-app">
   <deploy-in refid="rib-rfm-wls1" />
   <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:19105/rib-rfm-appserver-gui/index.jsp</web-app-url>
      <web-app-user-alias>rib-rfm_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rfm_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-rfm_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
   <error-hospital-database>
      <hosp-url>jdbc:oracle:thin:@rfmhosphost.example.com:1521/pdborcl</hosp-url>
      <hosp-user-alias>rib-rfm_error-hospital-database_user-name-alias</hosp-user-alias>
   </error-hospital-database>
   <app-database>
      <app-db-url>jdbc:oracle:thin:@rfmdbhost.example.com:1521/pdborcl</app-db-url>
      <app-db-user-alias>rib-rfm_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="rfm" type="plsql-app">
      <jndi-not-applicable />
   </app>
</rib-app>
```

For Java EE applications, you must define RIB admin GUI information, RIB Hospital connection, email notification information, and the connecting retail application's (<app>) JNDI information.



```
<rib-app id="rib-fileio" type="javaee-app">
   <deploy-in refid="rib-fileio-wls1" />
   <rib-admin-gui>
      <web-app-url>URL to the rib admin gui web app.</web-app-url>
      <web-app-user-alias>rib-fileio_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-fileio_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-fileio_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
   <error-hospital-database>
      <hosp-url>jdbc:oracle:thin:@fileiodbhost.example.com:1521/pdborcl</hosp-url>
      <hosp-user-alias>rib-fileio_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="fileio" type="javaee-app">
      <jndi>
         <url>t3://fileiohost.example.com:7002/fileio-server</url>
         <factory>weblogic.jndi.WLInitialContextFactory</factory>
         <user-alias>fileio_jndi_user-name-alias</user-alias>
      </jndi>
   </app>
</rib-app>
```

For SOAP application, you must define the RIB Admin GUI information, RIB Hospital connection, email notification information, and the end point application url information.



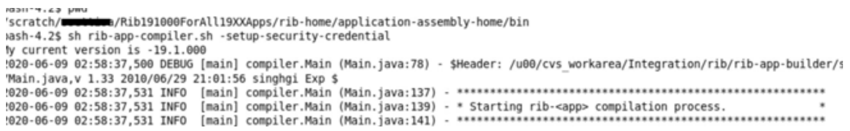
```
<rib-app id="rib-ocds" type="soap-app">
   <deploy-in refid="rib-ocds-wls1" />
   <rib-admin-gui>
      <web-app-url>http://ribhost.example.com:19206/rib-ocds-appserver-gui/index.jsp</web-app-url>
      <web-app-user-alias>rib-ocds_rib-admin-gui_admin-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-ocds_rib-admin-gui_operator-user-name-alias</web-app-user-alias>
      <web-app-user-alias>rib-ocds_rib-admin-gui_monitor-user-name-alias</web-app-user-alias>
   </rib-admin-gui>
   <error-hospital-database>
      <hosp-url>jdbc:oracle:thin:@ocdshost.example.com:1521/pdborcl</hosp-url>
      <hosp-user-alias>rib-ocds_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="ocds" type="soap-app">
      <end-point>
         <url>http://ocdshost.example.com:9001/ApplicationMessageInjectorBean/InjectorService</url>
         Supported security policy names =policyC (default) OR policyA OR policyB
         <ws-policy-name>policyC</ws-policy-name>
         <user-alias>rib-ocds_ws_security_user-name-alias</user-alias>
      </end-point>
   </app>
</rib-app>
```

For master-plsql-app type applications, you must define the RIB Admin GUI information, RIB Hospital connection, email notification information, and slave end point application url information.

Note: For additional information, follow Chapter 7, "RIB-RWMS Hybrid Cloud Installation Instructions".

8. Run the rib-home/application-assembly-home/bin/rib-app-compiler.sh script with set-up-security-credential from rib-home/application-assembly-home/bin directory.

   Example: ./rib-app-compiler.sh -setup-security-credential



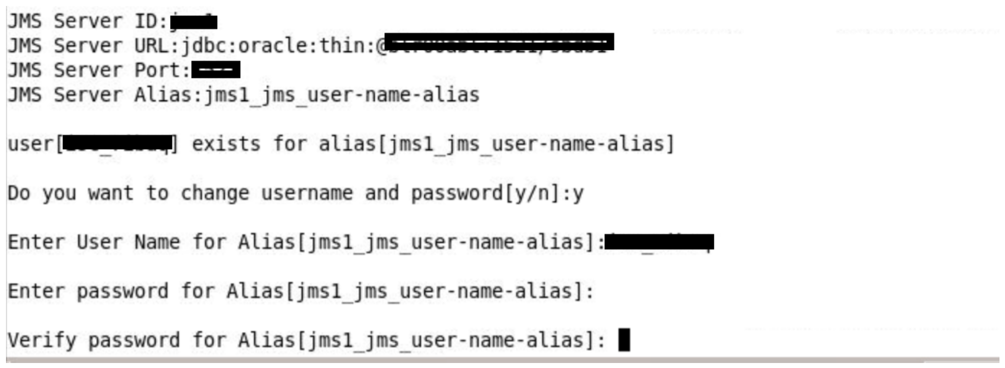
```
'scratch/[redacted]/Rib191000ForAll19XXApps/rib-home/application-assembly-home/bin
ash-4.2$ sh rib-app-compiler.sh -setup-security-credential
ly current version is -19.1.000
!020-06-09 02:58:37,500 DEBUG [main] compiler.Main (Main.java:78) - $Header: /u00/cvs_workarea/Integration/rib/rib-app-builder/s
'Main.java,v 1.33 2010/06/29 21:01:56 singhgi Exp $
!020-06-09 02:58:37,531 INFO  [main] compiler.Main (Main.java:137) - ********************************************************
!020-06-09 02:58:37,531 INFO  [main] compiler.Main (Main.java:139) - * Starting rib-<app> compilation process.              *
!020-06-09 02:58:37,531 INFO  [main] compiler.Main (Main.java:141) - ********************************************************
```

This will ask for user name and password information for aliases provided in the rib-deployment-env-info.xml file.

The user names and passwords are stored in a wallet file inside rib-home/deployment-home/conf/security directory.

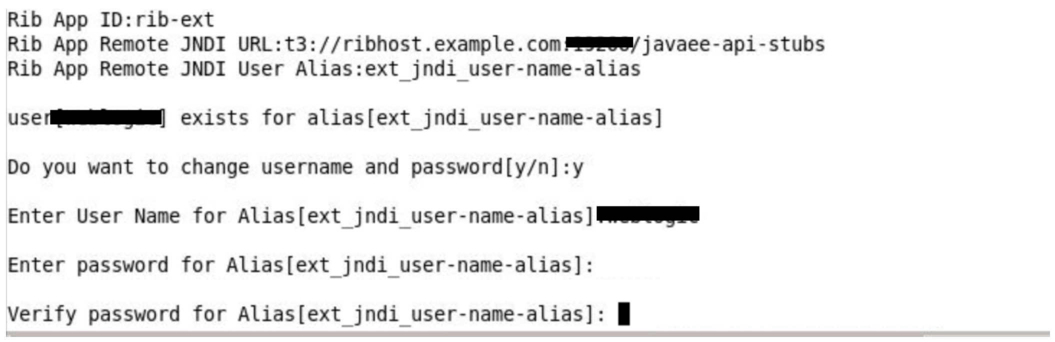
Below are few screens which appears while executing rib-app-compiler.sh

JMS user name alias and password:

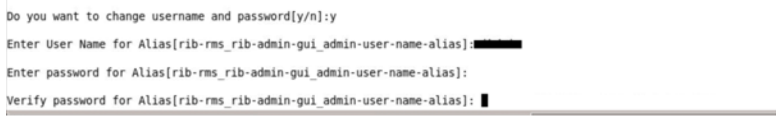


```
JMS Server ID:[redacted]
JMS Server URL:jdbc:oracle:thin:@[redacted]
JMS Server Port:[redacted]
JMS Server Alias:jms1_jms_user-name-alias

user[[redacted]] exists for alias[jms1_jms_user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[jms1_jms_user-name-alias]:[redacted]

Enter password for Alias[jms1_jms_user-name-alias]:

Verify password for Alias[jms1_jms_user-name-alias]: █
```

JNDI user name alias and password:

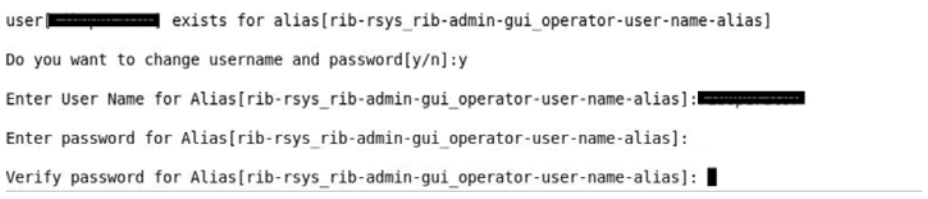


```
Rib App ID:rib-ext
Rib App Remote JNDI URL:t3://ribhost.example.com[redacted]/javaee-api-stubs
Rib App Remote JNDI User Alias:ext_jndi_user-name-alias

user[redacted]] exists for alias[ext_jndi_user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[ext_jndi_user-name-alias][redacted]

Enter password for Alias[ext_jndi_user-name-alias]:

Verify password for Alias[ext_jndi_user-name-alias]: █
```



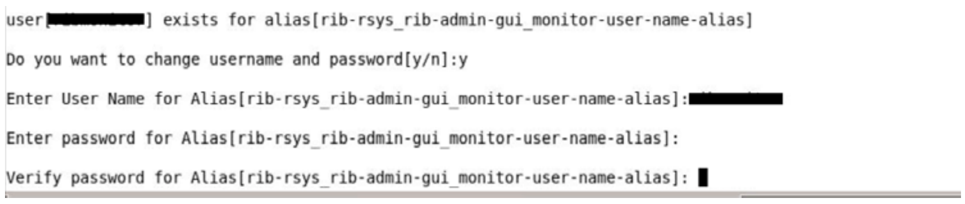
```
Do you want to change username and password[y/n]:y

Enter User Name for Alias[rib-rms_rib-admin-gui_admin-user-name-alias]:[redacted]

Enter password for Alias[rib-rms_rib-admin-gui_admin-user-name-alias]:

Verify password for Alias[rib-rms_rib-admin-gui_admin-user-name-alias]: █
```

Rib-admin-gui username alias and password for admin role:



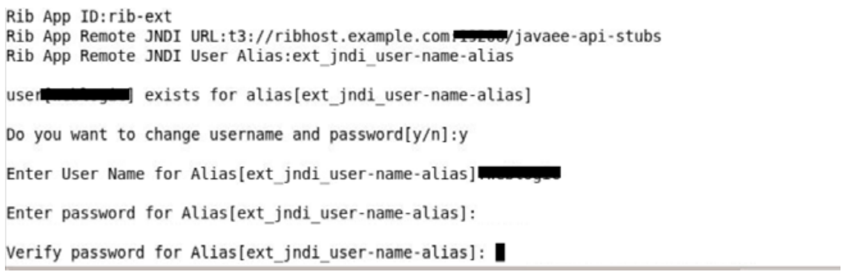
```
user[redacted] exists for alias[rib-rsys_rib-admin-gui_operator-user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[rib-rsys_rib-admin-gui_operator-user-name-alias]:[redacted]

Enter password for Alias[rib-rsys_rib-admin-gui_operator-user-name-alias]:

Verify password for Alias[rib-rsys_rib-admin-gui_operator-user-name-alias]: █
```

Rib-admin-gui username alias and password for monitor role:



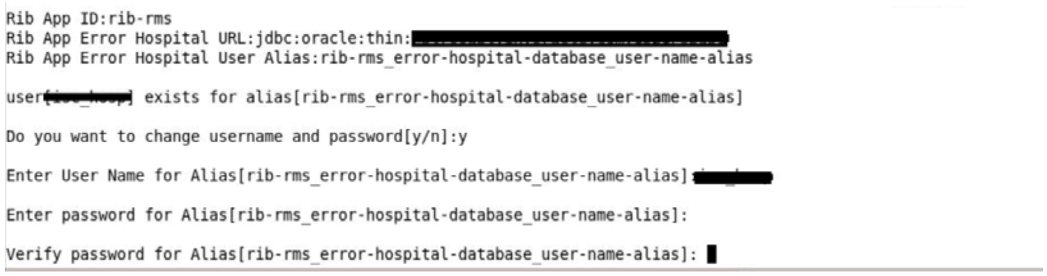
```
user[redacted]] exists for alias[rib-rsys_rib-admin-gui_monitor-user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[rib-rsys_rib-admin-gui_monitor-user-name-alias]:[redacted]

Enter password for Alias[rib-rsys_rib-admin-gui_monitor-user-name-alias]:

Verify password for Alias[rib-rsys_rib-admin-gui_monitor-user-name-alias]:
```

Rib-admin-gui username alias and password for operator role:



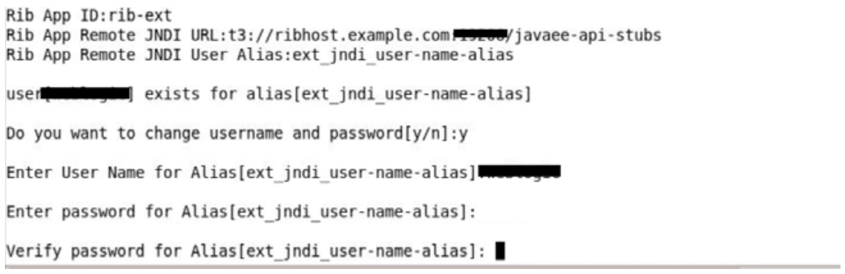
```
Rib App ID:rib-ext
Rib App Remote JNDI URL:t3://ribhost.example.com[redacted]/javaee-api-stubs
Rib App Remote JNDI User Alias:ext_jndi_user-name-alias

user[redacted] exists for alias[ext_jndi_user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[ext_jndi_user-name-alias][redacted]

Enter password for Alias[ext_jndi_user-name-alias]:

Verify password for Alias[ext_jndi_user-name-alias]:
```

Error hospital details:



```
Rib App ID:rib-rms
Rib App Error Hospital URL:jdbc:oracle:thin:[redacted]
Rib App Error Hospital User Alias:rib-rms_error-hospital-database_user-name-alias

user[redacted] exists for alias[rib-rms_error-hospital-database_user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[rib-rms_error-hospital-database_user-name-alias][redacted]

Enter password for Alias[rib-rms_error-hospital-database_user-name-alias]:

Verify password for Alias[rib-rms_error-hospital-database_user-name-alias]:
```

WebLogic credentials:



```
Rib App ID:rib-ext
Rib App Remote JNDI URL:t3://ribhost.example.com[redacted]/javaee-api-stubs
Rib App Remote JNDI User Alias:ext_jndi_user-name-alias

user[redacted] exists for alias[ext_jndi_user-name-alias]

Do you want to change username and password[y/n]:y

Enter User Name for Alias[ext_jndi_user-name-alias][redacted]

Enter password for Alias[ext_jndi_user-name-alias]:

Verify password for Alias[ext_jndi_user-name-alias]:
```

After that this will generate/assemble a rib-<app> and make it ready for deployment

The RIB apps are now ready to deploy.

9. Log level change in "Log manager" page for any adapters can be restricted (read-only) by adding names of adapters as comma (",") separated value against key "disableLogLevelUpdatesForAdapter" in rib-<appName>.properties file located in rib-home/application-assembly-home/rib-<app>.

   Example: disableLogLevelUpdatesForAdapter = Customer_pub

   Below are list of properties file that are updated for current release

   - rib-ext.properties

10. Execute the rib-home/deployment-home/bin/rib-app-deployer.sh script with the appropriate com-mand line parameter.

This script is located in the rib-home/deployment-home/bin directory.

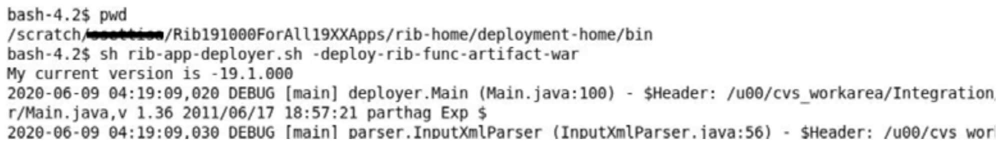
- rib-app-deployer.sh -prepare-jms

This creates a new JMS server with all RIB configured topics.

`>rib-app-deployer.sh -verify-error-hospital rib-<app>`
This verifies the error-hospital database configurations by testing the connection to the database if the error-hospital tables are created in the schema.

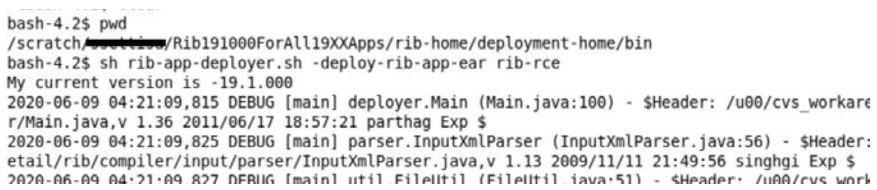
Note: The database must be already running.

`rib-app-deployer.sh -deploy-rib-func-artifact-war`



```
bash-4.2$ pwd
/scratch/[redacted]/Rib191000ForAll19XXApps/rib-home/deployment-home/bin
bash-4.2$ sh rib-app-deployer.sh -deploy-rib-func-artifact-war
My current version is -19.1.000
2020-06-09 04:19:09,020 DEBUG [main] deployer.Main (Main.java:100) - $Header: /u00/cvs_workarea/Integration
r/Main.java,v 1.36 2011/06/17 18:57:21 parthag Exp $
2020-06-09 04:19:09.030 DEBUG [main] parser.InputXmlParser (InputXmlParser.java:56) - $Header: /u00/cvs wor
```

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



```
bash-4.2$ pwd
/scratch/[redacted]/Rib191000ForAll19XXApps/rib-home/deployment-home/bin
bash-4.2$ sh rib-app-deployer.sh -deploy-rib-app-ear rib-rce
My current version is -19.1.000
2020-06-09 04:21:09,815 DEBUG [main] deployer.Main (Main.java:100) - $Header: /u00/cvs_workare
r/Main.java,v 1.36 2011/06/17 18:57:21 parthag Exp $
2020-06-09 04:21:09,825 DEBUG [main] parser.InputXmlParser (InputXmlParser.java:56) - $Header:
etail/rib/compiler/input/parser/InputXmlParser.java,v 1.13 2009/11/11 21:49:56 singhgi Exp $
2020-06-09 04:21:09.827 DEBUG [main] util.FileUtil (FileUtil.java:51) - $Header: /u00/cvs work
```

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

Note: <app> must be rms, rwms, tafr, sim, rpm, rsys, rce or aip.

During the installation a shared library is created that contains the JDBC Driver update. It is necessary to bounce the wls instance.

B

Appendix: RIB Installation Checklists

This appendix is intended as an aid in the installation of RIB. It is not intended to replace the detailed description of each of the process steps and prerequisites, but to act as a companion to those steps. For a successful installation, a methodical reading and understanding of each step of the *Oracle Retail Integration Bus Installation Guide* is strongly recommended.

RIB Installation Master Checklist

This checklist covers all of the sequential steps required to perform a full installation of the RIB, using a command line installation.

Task	Notes
Prepare the WebLogic servers for installation of the RIB Components	Prerequisite
Prepare the Oracle Database schemas that the RIB will use.	Prerequisite
Prepare the Oracle AQ JMS	Prerequisite
Verify the applications the RIB will be integrating to are configured appropriately.	In the documentation for each Oracle application, see the sections on integration with the RIB.
Information to gather for the installation	During the prerequisites steps there is information that should be note that will be used to configure the RIB during the installation process.
Install the RIB using the RIB App Builder Command Line Tools.	
Verify Application URL settings match RIB installation.	RIB Functional Artifact URL JNDI URL
Complete the setup of RDMT using the same information to gather for the installation.	During either of the installation methods, one of the manual steps will have extracted the rdmt tools to the appropriate directory.
Verify the RIB installation using the RDMT tools.	
Install RIHA	RIB Hospital maintenance tool

Task	Notes
Install IGS	This step is optional and should be performed only if there is a requirement to do so. See "Integration Gateway Services" in the *Oracle Retail Integration Bus Implementation Guide*.

Prerequisite - Prepare WebLogic Server for RIB Components

Task	Notes
Install WebLogic Server 12.2.1.4	See the *Oracle Retail Integration Bus Release Notes* for the certifications. See the *Oracle Retail Integration Bus Implementation Guide* for deployment architectures.
Create the RIB WebLogic managed server instances. Warning: Each rib-<app> application requires a separate WebLogic managed server instance that is not shared with any other application. Create the rib-<app>-wls-instance using WebLogic admin console GUI Log in to the WebLogic admin console GUI (http://<host>:<port>/console) as administrator On the left side menu, navigate to Environment > Servers Click **New**. Fill in the Name, Port, Listen address of the managed server instance to be create. **Example**: Server Name: rib-<app>-wls-instance Server Listen Address: ribhost Server Listen Port:19107	For information about creating and managing WebLogic managed server instances, see Oracle® Fusion Middleware Administrator's Guide 12.2.1.4 Replace <app> with the actual value of the RIB application for the associated retail application. Acceptable values for <app> are "rms", "lgf","ocds", "rwms", "tafr", "sim", "rpm", "rfm", "aip", "ext", and "rob." Port number must be a unique port. There are two RIB specific WebLogic instances that must be created regardless of the other application deployment choices. ■ rib-func-artifact-wls-instance. (It is recommended, but not required, that this naming convention be followed.) These are the optional application instances depending on the deployment choices. It is recommended, but not required that this naming convention be followed: ■ rib-rms-server ■ rib-tafr-server ■ rib-rpm-server ■ rib-sim-server ■ rib-rwms-server ■ rib-rfm-server ■ rib-lgf-server ■ rib-ocds-server ■ rib-aip-server ■ rib-rob-server ■ rib-ext-server
Click **Next**. Click **Finish**. Make sure you see this instance listed under Servers.	
Go to the configurations page of the server and select the host name in the Machine field. Click **Save**. Managed server instance creation is complete.	

Task	Notes
Edit the script $DOMAIN_ HOME/base_ domain/bin/startWebLogic.sh to add the following attributes after DOMAIN_ HOME. CLASSPATH=$DOMAIN_ HOME/servers/$SERVER_ NAME:$CLASSPATH JAVA_ OPTIONS="-Dweblogic.ejb.container.MDBMessageWaitTime=2 ${JAVA_OPTIONS}"	Sample from startWebLogic: echo "JAVA Memory arguments: ${MEM_ARGS}" echo "." echo "WLS Start Mode=${WLS_DISPLAY_MODE}" echo "." JAVA_ OPTIONS="-Dweblogic.ejb.container.MDBMessageWaitTim e=2 ${JAVA_OPTIONS}" CLASSPATH=$DOMAIN_HOME/servers/$SERVER_ NAME:$CLASSPATH echo "CLASSPATH=${CLASSPATH}" echo "." echo "PATH=${PATH}" echo "." echo "** *" echo "* To start WebLogic Server, use a username and *" echo "* password assigned to an admin-level user. For *" echo "* server administration, use the WebLogic Server *" echo "* console at http://hostname:port/console *" echo "****************************** *********************"

Task	Notes
	Note: In the startWebLogic script, the above statements must be added before the WebLogic server is started. In other words, the statements must be before these lines:

```
if [ "${WLS_REDIRECT_LOG}" = "" ] ; then
       echo "Starting WLS with line:"
       echo "${JAVA_HOME}/bin/java -d64 ${JAVA_VM}
${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_
HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}"
       #echo "${JAVA_HOME}/bin/java -d64 ${JAVA_VM}
${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_
HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}"
       ${JAVA_HOME}/bin/java
${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_
NAME} -Djava.security.policy=${WL_
HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}
else
       echo "Redirecting output from WLS window to
${WLS_REDIRECT_LOG}"
       #${JAVA_HOME}/bin/java -d64 ${JAVA_VM}
${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_
HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}  >"${WLS_
REDIRECT_LOG}" 2>&1
       ${JAVA_HOME}/bin/java  ${JAVA_VM} ${MEM_
ARGS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_
HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}  >"${WLS_
REDIRECT_LOG}" 2>&1
fi
```

Task	Notes

Update $WL_ HOME/<wlserver>/server/lib/weblogic.policy file with the information below.	**Note**: If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.
	
	Note: <WEBLOGIC_DOMAIN_HOME> in the below example is the full path of the Weblogic Domain, <managed_server> is the RIB managed server created and <context_root> correlates to the rib-app ears for all managed servers hosting rib-apps, except for rib-func-artifact-instance. See the example below.
	
	Note: The path tmp/_WL_user/rib-<app>.ear will not be available before the deployment.
	
	Example:
	
	grant codeBase "file:
	
	<WEBLOGIC_DOMAIN_HOME>/servers/<managed_ server>/tmp/_WL_user/<context_root>/-" {
	
	permission java.security.AllPermission;
	
	permission oracle.security.jps.service.credstore.CredentialAccessPermission "credstoressp.credstore", "read,write,update,delete";
	
	permission oracle.security.jps.service.credstore.CredentialAccessPermission "credstoressp.credstore.*", "read,write,update,delete";
	
	};
	
	};
	
	Note: Add the path to the patch jars. If any patches are installed into WLS (now or in the future) and this line is not included it could cause the RIB to fail. WLS_HOME refers to the location where Weblogic 12.2.1 has been installed.
	
	`grant codeBase "file:<<WLS_HOME>/patch_wls/patch_ jars/-" {`
	`permission java.security.AllPermission;`
	`permission oracle.security.jps.service.credstore.CredentialAccessPermission "credstoressp.credstore", "read,write,update,delete";`
	`permission oracle.security.jps.service.credstore.CredentialAccessPermission "credstoressp.credstore.*", "read,write,update,delete";`
	`};`
	
	`};`

Task	Notes

Start WebLogic managed server.
 	
Note: This procedure can be done through the command line or through the admin console. Both methods are included below.
 	
Start WebLogic managed server through the command line:
 	
Log in to the machine where WLS was installed with the operating system user that was used to install the WebLogic Application Server (WLS).
 	
Navigate to the DOMAIN_ HOME/bin
 	
Example: $cd product/ 19.1.000_RIB/WLS/user_ projects/domains/base_ domain/bin
 	
run startManagedWebLogic script with instance name as a parameter
 	
Example: sh startManagedWebLogic.sh rib-rms-wls-instance
 	
Starting WebLogic managed server through admin console.
 	
To be able to properly start RIB managed server instance, the properties below need to be modified in $WL_ HOME/wlserver/common/n ode manager/nodemanager. properties file:
 	
Change the value of StartScriptEnabled property to true.
 	
Change the value of StartScriptName property to startWebLogic.sh
 	
Restart the NodeManager after making changes.
	Sample nodemanager.properties file:
	
	SecureListener=false
	
	`StartScriptName=startWebLogic.sh`
	`StartScriptEnabled=true`

Creating the WebLogic instances is complete.	

Prerequisite - Oracle Database Schemas

Task	Notes
Each Oracle Retail Application has an associated set of RIB Artifacts that must be installed as part of the RIB integration. For example, the RIB Hospital Tables, CLOB API libraries, and Oracle Objects. ■ Ensure that these have been installed appropriately per the individual applications. ■ Ensure that the TAFR Hospital user and objects exist. ■ Ensure that the RIB user has appropriate access and permissions.	Each Application packages the RIB artifact creation scripts and they are installed at the time of the application's installation. It is critical to ensure that they have been installed and are the correct version. The TAFR Hospital is independent of any of the applications and should have a separate user/schema created for it. It is recommended that all applications have a separate Hospital and that they be logically and operationally associated with that application.
Ensure that each PL/SQL application schema has run the RIB supplied scripts to create the RIB Artifacts: ■ 1_KERNEL_CREATE_ OBJECTS.SQL script. ■ InstallAndCompileAllRib OracleObjects.sql ■ 1_CLOB_CREATE_ OBJECTS.SQL (RMS only)	Verify the XML Developer's Kit for PL/SQL is installed.
RMS Application: Verify that the row in the RIB_ OPTIONS table has correct values to match the RIB deployment environment.	XML_SCHEMA_BASE_URL= http://<hostname>:<port>/rib-func-artifact;
Ensure that each Java EE application schema has run the RIB supplied scripts to create RIB artifacts: ■ 1_KERNEL_CREATE_ OBJECTS.SQL script.	

Task	Notes
RIB TAFR RIB Hospital Ensure that the schema exists and has run the RIB supplied script to create the RIB Hospital. ■ 1_KERNEL_CREATE_OBJECTS.SQL script.	In RIB 19.x, there is a separate hospital for all RIB TAFRs. Ensure that there is a user created for the RIB components and the scripts that create the hospital objects have been run. The TAFR Hospital user requires no special permissions. CREATE USER <*tafr hosp user*> IDENTIFIED BY <*tafr hosp password*> DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP"; GRANT "CONNECT" TO <*tafr hosp user*>; GRANT "RESOURCE" TO <*tafr hosp user*>; ALTER USER <*tafr hosp user*> QUOTA UNLIMITED ON USERS;

Prerequisite - Prepare Oracle AQ JMS Provider

Task	Notes
Create the Oracle Database instance that will be the JMS Provider.	Oracle Streams AQ is provided by the Oracle Database Enterprise Edition installation. **Note:** It is strongly recommended that the Oracle Database instance that is configured to be the JMS provider is not shared with any other applications and not be on the same host (physical or logical) with any other applications. See "Deployment Architecture" in the *Oracle Retail Integration Bus Implementation Guide*.

Create the AQ JMS user with the appropriate access and permissions to the Oracle Streams AQ packages. This user must have at least the following database permissions.
■ CONNECT
■ RESOURCE
■ CREATE SESSION
■ EXECUTE ON SYS.DBMS_AQ
■ EXECUTE ON SYS.DBMS_AQADM
■ EXECUTE ON SYS.DBMS_AQIN
■ EXECUTE ON SYS.DBMS_AQJMS
	Sample script:
	`CREATE USER <rib aq user> IDENTIFIED BY <rib aq password>`
	`DEFAULT TABLESPACE "RETAIL_DATA"`
	`TEMPORARY TABLESPACE "TEMP";`
	`GRANT "CONNECT" TO <rib aq user>;`
	`GRANT "RESOURCE" TO <rib aq user>;`
	`GRANT CREATE SESSION TO <rib aq user>;`
	`GRANT EXECUTE ON "SYS"."DBMS_AQ" TO <rib aq user>;`
	`GRANT EXECUTE ON "SYS"."DBMS_AQADM" TO <rib aq user>;`
	`GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO <rib aq user>;`
	`GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO <rib aq user>;`
	`GRANT "AQ_ADMINISTRATOR_ROLE" TO <rib aq user>;`
	`ALTER USER <rib aq user>`
	`QUOTA UNLIMITED ON RETAIL_DATA;`

Information	Notes
jms-server-home jms-url jms-port jms-user jms-password	JMS Provider for RIB Release 19.1.000 is AQ. ▪ jms-server-home: The server home must be in the format OsUser@AqHostName:/AqHomeDirectory For example, ribaq@ribaq-lnx-host:/u00/db "jms-url : AQ thin JDBC connection URL. For example, jdbc:oracle:thin:@ribaq-lnx-host:1521:orcl On AQ on RAC database use the long JDBC URL (for example, jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = ribaq-lnx-virtual-host-1)(PORT = 1521))(ADDRESS = (PROTOCOL = TCP)(HOST = ribaq-lnx-virtual-host-2)(PORT = 1521))(LOAD_ BALANCE = yes))(CONNECT_DATA =(SERVICE_ NAME = orcl))) ▪ jms-port: AQ JMS server listener port. This is same as the AQ JDBC listener port (for example, 1521). ▪ jms-user: AQ JMS user. This is the database user that can connect to jms-url (see above). ▪ jms-password: AQ JMS password. This is the database password that can connect to jms-url.
weblogic-domain-name weblogic-domain-home weblogic-domain-server -port java-home	For each of the WebLogic Servers to which the RIB components will be deployed. ▪ weblogic-domain-name: Your weblogic domain name (for example, base_domain). ▪ weblogic-domain-home: The format of the home must follow the format OsUser@WeblogicHostName:/WeblogicDomainPath. For example: ribapp@ribapp-lnx-host:/home/Oracle/Middleware/us er_projects/domains/base_domain ▪ weblogic-admin-server-port: The port where weblogic admin server is listening (for example, 7001) ▪ java-home : Java Home directory of the remote Weblogic server (for example, /usr/java/jdk1.8.0_64)
wls-instance-name wls-instance-home wls-listen-port wls-user-alias	The WebLogic instances for each of your rib-<app> applications that are in-scope. ▪ wls-instance-name: The WebLogic managed server instance name. For example, rib-rms will be deployed in rib-rms-server. ▪ wls-instance-home: The WebLogic instance server home information. For example, ribapp@ribapp-lnx-host:/home/ Oracle/Middleware/user_projects/domains/base_ domain /servers/rib-rms-server ▪ wls-listen-port: The WebLogic managed server listen port. For example, 7003. ▪ wls-user-alias: User alias for username/password to connect to the WebLogic managed server. The username/password are stored in a wallet file in rib-home/deployment-home/conf/security folder and the rib-deployment-env-info.xml contains the alias name for that. The user name/password to connect to the managed server will be same as the user who starts the WebLogic server.

Information	Notes
To configure each rib-<app>, this information is needed for each.	■ The application server where it will be deployed. ■ RIB Hospital database information. ■ PL/SQL application database information. ■ E-mail notification information. ■ jndi information for Java EE applications.
For RIB Hospital database: database/url *<database user>* *<database password>*	■ database/url: rib-<app> error hospital thin JDBC connection URL. For example, jdbc:oracle:thin:@ribapp-lnx-host:1521:orcl If RIB Hospital tables are running on RAC database use the long JDBC url format. For example, jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = ribapp-lnx-virtual-host-1)(PORT = 1521))(ADDRESS = (PROTOCOL = TCP)(HOST = ribapp-lnx-virtual-host-2)(PORT = 1521))(LOAD_ BALANCE = yes))(CONNECT_DATA =(SERVICE_ NAME = orcl))) ■ *<database user>*: This is the database user that will be used to connect to rib-<app> error hospital tables. ■ <database password>: This is the database password that will be used to connect to rib-<app> error hospital tables.
For PL/SQL application database: database/url *<database user>* *<database password>*	See samples in the row above for RIB Hospital Database.
For email notifications: email-server-host email-server-port from-address to-address-list	■ email/email-server-host: The SMPT mail server (for example, mail.yourcompany.com) ■ email/email-server-port: The SMTP mail server port. (for example, 25) ■ email/from-address: The email address from which the RIB notifications will originate (for example, ribadmin@example.com) ■ email/to-address-list: Comma separated list of destination email address to which RIB notifications will be sent (for example,ribappsadmin1@example.com, ribappsadmin2@example.com)
joined information for Java EE applications: jndi/url jndi/factory jndi/user jndi/password	jndi/url: The JNDI url for the retail <app> that this rib-<app> is connecting to. The URLs must use the following format. **WLS URL format:**t3://host:port/applicationName For example, t3://simhost.example.com:18022/rib-sim. **jndi/factory**: The JNDI provider factory class name. The factory must be one of the following. ■ WLS factory: weblogic.jndi.WLInitialContextFactory ■ jndi/user: The retail <app> JNDI user name. This is the same as the retail <app> WLS server instance user name. ■ jndi/password: The retail <app> JNDI password. This is same as the retail <app> WLS server instance password. For example, weblogic (must be encrypted)

Install Using the RIB App Builder Command Line Tools

Task	Notes
Make sure that the JAVA_HOME environment variable is set for the user that will be performing these tasks. > echo $JAVA_HOME /usr/bin/java/jdk1.8.0_64	Example: export JAVA_HOME=/usr/bin/java/jdk1.8.0_64
Make sure that all RIB WebLogic instances that are to be deployed to are running.	
Determine the host and file system to create the rib-app-builder home directory on. > mkdir rib-app-builder	See the *Oracle Retail Integration Bus Implementation Guide* for guidelines and deployment approaches. This is an important strategic decision, because all RIB configurations and management for a given deployment will be from this single, central location.
Download and extract the RibKernel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar. > tar xvf RibKernel19.1.000ForAll19.x.xApps_eng_ga.jar	Copy the latest version to the rib-app-builder and then extract it to build your "rib-home." This "rib-home" will be the directory from where you will perform "all" the rib-<app> related tasks from now on.
Download the RibFuncArtifact<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar and put it in rib-home/download-home/rib-func-artifacts directory.	Do not extract the tar file. This will be done by the check-version-and-unpack tool.
Download the RibPak<RIB_MAJOR_VERSION>For<RETAIL_APP_NAME><RETAIL_APP_VERSION>_eng_ga.tar and put it in rib-home/download-home/all-rib-apps directory.	Do not extract the tar file. This will be done by the check-version-and-unpack tool.
Run the rib-home/download-home/bin/check-version-and-unpack.sh script from rib-home/download-home/bin directory.	This script verifies the version compatibility between the paks and extract the files if they are compatible.

Task	Notes
Edit rib-home/deployment-home/conf/rib-deployment-env-info.xml file to specify the deployment environment information. See the "Information to Gather for Installation in Remote Server" section before starting the edit.	This file (rib-deployment-env-info.xml) is the only file that the user has to edit. See the "Rib-app-builder documentation" for details and examples. The XML file has four major sections. **1.** app-in-scope-for-integration section: In this section you define what applications are in scope for this environment. **2.** rib-jms-server section: In this section you define the JMS server information. **Note**: See also"Preinstallation Steps for Multiple JMS Server Setup" in Chapter 4 of this guide. **3.** rib-javaee-containers section: In this section you define the "Java EE container information" for each of the rib-<app> that are in scope. **4.** rib-applications section: In this section you define the rib-<app> specific information for each of the rib-<app> that are in scope. For PL/SQL applications you must define RIB Hospital connection and email notification information. For Java EE applications you will need to define RIB Hospital connection, email notification information and the connecting retail application's (for example, <app>) JNDI information.

Edit the app-in-scope-for-integration section to match the desired deployment.	Define what applications are in scope for this environment.
	
	`<app-in-scope-for-integration>`
	`  <app id="rms" type="plsql-app"/>`
	`  <app id="tafr" type="tafr-app"/>`
	`  <app id="sim" type="javaee-app"/>`
	`  <app id="rwms" type="plsql-app"/>`
	`  <app id="rpm" type="javaee-app"/>`
	`  <app id="lgf" type="soap-app"/>`
	`  <app id="ocds" type="soap-app"/>`
	`  <app id="rfm" type="plsql-app"/>`
	`  <app id="aip" type="javaee-app"/>`
	`  <app id="rob" type="soap-app"/>`
	`  <app id="ext" type="javaee-app"/>`
	`</app-in-scope-for-integration>`
	`<app id="ocds" type="soap-app"/>`
	
	The below configuration is for RWMS as a master-plsql-app. Use this when RIB is on cloud and RWMS is on-premises.
	
	`<app id="rwms" type="master-plsql-app" />`
	
	Note: For additional information on rib-rwms hybrid cloud installation, see Chapter 7, "RIB-RWMS Hybrid Cloud Installation Instructions".

Task	Notes

Edit the rib-jms-server section.
 	
See "Preinstallation Steps for Multiple JMS Server Setup"in Chapter 4 of this guide.
	For AQ:
	
	`<jms-server-home>linux1@linux1:/home/oracle/oracle/product/12.1.0.2/db_1</jms-server-home>`
	
	`<jms-url>jdbc:oracle:thin:@linux1:1521:ora12c</jms-url>`
	`<jms-port>1521</jms-port>`
	`<jms-user-alias>aq1-alias</jms-user-alias>`

Edit the application server section	For example: <weblogic-domain-name>base_domain</weblogic-domain-name> <weblogic-domain-home>user1@linux1:/home/user1/Oracle/Middleware/user_projects/domains/base_domain</weblogic-domain-home> <weblogic-admin-server-port>7001</weblogic-admin-server-port> <java-home>/usr/java/jdk1.8</java-home>
Configure the WebLogic instances for each of your rib-<app> applications that are in scope.	<wls id="rib-rms-server"> <wls-instance-name>rib-rms-server</wls-instance-name> <wls-instance-home>soa1@linux1:/home/soa1/Oracle/Middleware/user_projects/domains/base_domain/servers/rib-rms-server</wls-instance-home> <wls-listen-port protocol="http">7003</wls-listen-port> <wls-user-alias>rib-rms-wls-user-alias</wls-user-alias> </wls>

Task	Notes

Configure the rib-applications section:
 	
In this section you define the rib-<app> specific information for each rib-<app> that in scope.
	For PL/SQL applications you must define the RIB Hospital connection, application database connections, and email notification information.
	
	`<rib-app id="rib-rms" type="plsql-app">`
	`<deploy-in refid="rib-rms-wls-instance" />`
	`  <error-hospital-database>`
	
	`<hospurl>jdbc:oracle:thin:@rmsdbhost.example.com:`
	`      1521:orcl`
	`     </hosp-url>`
	`     <hosp-user-alias>rib-<app>_`
	`error-hospital-database_`
	`user-name-alias</hosp-user-alias>`
	`  </error-hospital-database>`
	`  <app-database>`
	`   <app-db-url>jdbc:oracle:thin:@rmsdbhost.example`
	`.com:`
	`     1521:orcl`
	`    </app-db-url>`
	`     <app-db-user-alias>rib-<app>_app-database_`
	`user-name-alias</app-db-user-alias>`
	`  </app-database>`
	`  <notifications>`
	`   <email>`
	
	`<email-server-host>mail.oracle.com</email-server-h`
	`ost>`
	`<email-server-port>25</email-server-port>`
	`<from-address>admin@example.com</from-address>`
	`<to-address-list>admin@example.com</to-address-lis`
	`t>`
	`    </email>`
	` <jmx/>`
	`</notifications>`

Task	Notes

	`<rib-app id="rib-rms" type="plsql-app">`
	`<deploy-in refid="rib-rms-server"/>`
	`<rib-admin-gui>`
	`<web-app-url>http://linux1:7003/rib-rms-admin-gui<`
	`/web-app-url>`
	`<web-app-user-alias>rib-rms_rib-admin-gui_`
	`web-app-user-alias</web-app-user-alias>`
	`</rib-admin-gui>`
	`<app-database>`
	`<app-db-url>jdbc:oracle:thin:@rmsdbhost.example.co`
	`m:1521:orribdb`
	`</app-db-url>`
	`<app-db-user-alias>rib-rms_app-database_`
	`user-name-alias</app-db-user-alias>`
	`</app-database>`
	`<error-hospital-database>`
	`<hosp-url>jdbc:oracle:thin:@rmsdbhost.example.com:`
	`1521:orribdb</hosp-url>`
	`<hosp-user-alias>rib-rms_error-hospital-database_`
	`user-name-alias</hosp-user-alias>`
	`</error-hospital-database>`
	`<notifications>`
	`<email>`
	
	`<email-server-host>mail.oracle.com</email-server-h`
	`ost>`
	`<email-server-port>25</email-server-port>`
	`<from-address>admin@example.com</from-address>`
	`<to-address-list>admin@example.com</to-address-lis`
	`t>`
	`</email>`
	`<jmx/>`
	`</notifications>`
	`<app id="rms" type="plsql-app">`
	`<jndi-not-applicable/>`
	`</app>`
	`</rib-app>`

	For Java EE applications, you must define RIB admin GUI information, RIB Hospital connection, email notification information, and the connecting retail application's (<app>) JNDI information.

Task	Notes

	For SOAP application, you must define the RIB Admin GUI information, RIB Hospital connection, email notification information, and the end point application url information.
	`<app id="rob" type="soap-app">`
	`<end-point>`
	`<url>http://host:port/rib-injector-services-web/resources/injector/inject</url>`
	`<!--  Do we want this location or derive by default? -->`
	`<!--  TODO : We need pick the ws policy from this location and push the policy to server. -->`
	`<ws-policy-name>policyB</ws-policy-name>`
	`<user-alias>rib-rob_ws_security_user-name-alias</user-alias>`
	`</end-point>`
	`</app>`

Configure the rib-applications section: In this section you define the rib-<app> specific information for each rib-<app> that in scope	For master-plsql-app type applications, you must define the RIB Admin GUI information, RIB Hospital connection, email notification information, and slave end point application url information. **Note:** For additional information, follow Chapter 7, "RIB-RWMS Hybrid Cloud Installation Instructions".
Run the rib-home/application-assembly-home/bin/rib-app-compiler.sh script with setup-security-credential from rib-home/application-assembly-home/bin directory. Example: ./rib-app-compiler.sh -setup-security-credential	This will ask for user name and password information for aliases provided in the rib-deployment-env-info.xml file. The user names and passwords are stored in a wallet file inside rib-home/deployment-home/conf/security directory. After that this will generate/assemble a rib-<app> and make it ready for deployment
The RIB apps are now ready to deploy. Execute the rib-home/deployment-home/bin/rib-app-deployer.sh script with the appropriate command line parameter.	This script is located in the rib-home/deployment-home/bin directory.
> rib-app-deployer.sh -prepare-jms	This creates a new JMS server with all RIB configured topics.
>rib-app-deployer.sh -verify-error-hospital rib-<app>	This verifies: **1.** Error-hospital database configurations by testing the connection to the database. **2.** If the error-hospital tables are created in the schema. Note: Database must be already running.
> rib-app-deployer.sh -deploy-rib-func-artifact-war	This deploys the rib-func-artifact.war to the Java EE container.

Task	Notes
> rib-app-deployer.sh -deploy-rib-app-ear rib-<app>	This deploys the rib-<app> to the Java EE container. Repeat this step for all rib-<app> that is in scope for this integration environment. **Note:** <app> must be rms, rwms, tafr, sim, rpm, or aip.
Bounce all of the rib-<app>-wls-instances.	During the installation a shared library is created that contains the JDBC Driver update. It is necessary to bounce the wls instance.
Verify Application URL settings match RIB installation.	RIB Functional Artifact URL JNDI URL
Verify the installation using RDMT	

RDMT - Information to Gather

The following are necessary directory parameters.

RDMT Home Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt/
RDMTLOGS Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt /RDMTLOGS
Temp Files Directory	Rib191000ForAll19xxApps/rib-home/tools-home/rdmt /RDMTLOGS/tmp
RIB App Builder rib-home Directory	/u00/Rib191000ForAll19xxApps/rib-home

The following are parameters for JMS Provider.

AQ JMS User ID	*<rib aq user>*
AQ JMS Password	*<rib aq password>*
AQ JMS Database Name	soa1
JMS HOST	jmshost.example.com
JMS PORT	1521

The following are WebLogic parameters for JMX functions.

WebLogic/JMX Host	wlshost.example.com
JMX Req Port	7003
WebLogic Instance name	rib-rms-wls-instance
WebLogic App Name	rib-rms
WebLogic User Name	*<weblogic user>*
WebLogic Password	*<weblogic password>*

The following are parameters for each hospital (RMS, RWMS, SIM, and others).

User Name	*<rms user>*

Password	*<rms password>*
Database (SID)	orcl
Database Host	dbhost.example.com
Listener Port	1521

RDMT - Installation

The following are the steps required to complete RDMT installation.

Task	Notes
Make sure that the Java path is set Java 8.0. > java -version	The RDMT Java support classes require Java 8.0, and installation will perform a check and fail if the path is not correct. Prior to the installation, verify that your Java version is correct.
Download the Rdmt19.1.000ForAll19.x.x Apps_eng_ga.tar.	The recommended location is in rib-home/tools-home directory. There is an empty rdmt subdirectory already there. This is only a placeholder. RDMT can be installed under any user in any directory.
Extract the tar file. > tar xvf Rdmt19.1.000ForAll19.x.x Apps_eng_ga.tar	Extract the tar file. It will create or over-write a directory call rdmt.
Execute the configbuilder.sh script. > ./configbuilder.sh	cd to the rdmt directory and execute the configbuilder.sh script supplied with the toolkit.
If rdmt is extracted under rib-home, it updates the necessary rdmt configuration files if installed under rib-home/tools-home/rdmt directory.	The configbuilder.sh script checks if rdmt is installed under rib-home. If so, it fetches and updates all the necessary configuration information from rib-deployment-env-info.xml present under rib-home/deployment-home/conf directory. Also, it configures for all the rib-<app>s depending upon the applications in scope as defined in rib-deployment-env-info.xml.
If rdmt is extracted in some other directory outside rib-home, it updates the necessary rdmt configuration files if installed in some other directory with rib-home present on same server.	Once prompted for rib-home path, provide the same and it fetches and updates all the necessary configuration information from rib-deployment-env-info.xml present under specified rib-home/deployment-home/conf directory. Also, it configures for all the rib-<app>s depending upon the applications in scope as defined in rib-deployment-env-info.xml.
If rdmt is extracted in a remote server with no rib-home present, answer prompts for RIB configuration values during Setup if installed in a remote server with no rib-home present on that server.	The installation script will prompt for the configuration settings need to run the tools in the toolkit (See the section, "Information to Gather for Installation in Remote Server", in this manual.) **Note:** After the installation, these configurations can be changed at any time via any text editor in the appropriate configuration file.

Task	Notes
Answer prompts for the additional JMX configurations. Answer yes to configure additional rib-apps in case of remote installation.	After prompting for the necessary configuration parameters, the setup script updates the various configuration files and then prompts the user for additional JMX configurations that the user will be interested in. It is recommended that you configure all the rib-apps that have been installed in the RIB Installation process and then run the RibConfigReport. This report will run a battery of tests that will validate the RIB components installed.
The configbuilder.sh script will set the permissions to 700 (-rwx------) on all tools and files within the rdmt directory structure.	There are configurations that contain passwords.
Run Configuration Report	This report will execute using all of the configuration parameter that have been supplied and will verify them against the RIB installation
Installation is complete.	

RIB Hospital Administration (RIHA) - Installation

The following is a checklist for Oracle Retail RIHA installation.

Task	Notes
Verify the JRE Installed on server/PC where RIHA will be installed.	The minimum and preferred Java Runtime Engine (JRE) version to use with RIHA is 1.7.
The RIB XSDs must be made network-accessible for RIHA to properly display RIB messages.	The RIB Functional Artifact URL (for example, http://ribhost.example.com:7777/rib-func-artifact/payload/xsd/) should be accessible to all RIHA users.
Verify RIHA version is compatible with RIB version.	Due to changes in the underlying RIB architecture RIHARelease 19.1.000 is compatible only with RIB19.1.X and higher.
Verify ADF runtime 12.2.1 is available in the WebLogic 12.2.1 domain where RIHA will be installed.	RIHA model and view components needs ADF runtime to function properly.
Ensure the Firefox browser version is 3.5 or higher.	RIHA GUI works better in Firefox version 3.5 or higher.
Deploy EAR	

Task	Notes
RIHA app can be deployed from rib-home	Steps to deploy riha app from rib-home: 1. Download the RibHospitalAdministration-Web-19.1.000ForAll19.x.xApps_eng_ga.tar and extract it to RIB_INSTALL/rib-home/tools-home 2. Navigate to the location rib-home/tools-home/riha/conf 3. Edit the riha-deployment-env-info.properties with riha-admin-server-connection-url value 4. Compiling the riha-app by executing tools-home/riha/bin: ./riha-app-compiler.sh -setup-security-credential 5. Prepare weblogic for riha deployment by executing (Creating Datasource for no of apps in scope of rib-deployment info xml) tools-home/riha/bin: ./riha-app-deployer.sh -prepare-wls 6. Deployment by executing tools-home/riha/bin: ./riha-app-deployer.sh -deploy-riha-app 7. Undeploying by executing tools-home/riha/bin: ./riha-app-deployer.sh -undeploy-riha-app **Note:** 1. The target server name where RIHA app should be deployed in riha-deployment properties For example: **`riha-wls-target-name=AdminServer - Means riha app will deploys to 'AdminServer' riha-wls-target-name=m1 - Means riha app will deploys to Managed Server 'm1'`** 2. The cluster name where RIHA app should be deployed For example: **`riha-wls-target-name=no_cluster - Means riha app will deploys to target name given for the property riha-wls-target-name riha-wls-target-name=New_Cluster_1 - Means riha app will deploys to cluster named New_Cluster_1`**
Log in to the WebLogic Console	Log in to the WebLogic server console where RIHA will be installed.
Post Deployment Configuration	
Test Deployment	1. In the left pane, select Deployments > Applications. 2. Select the installed RIHA application. 3. Select Context > Test.

Integration Gateway Services (IGS) Installation - Information to Gather

The following are the details for the RIB AQ JMS.

Field Name	Example	Comment
Database Name	ora12c	AQ Database instance name
Host Name	linux1.us.oracle.com	Database system
Port	1521	Database listener port
Database User Name	*<rib aq user>*	AQ user
Password	*<rib aq password>*	AQ user password

IGS - Installation (Optional)

Task	Notes
Install IGS component.	This component is optional and should be installed only if there is a requirement to do so. See "Integration Gateway Services" in the *Oracle Retail Integration Bus Implementation Guide*.
Prepare Oracle WebLogic Server	Prerequisite. Work with System and Application administrators on appropriate deployment. See "Integration Gateway Services" in the *Oracle Retail Integration Bus Implementation Guide*.
Create IGS WebLogic Server	The igs-service.ear file should be deployed on its own WebLogic server. When naming the WebLog instance, it is recommended (but not required) that the .ear file name is used (without the extension), along with underscore, wls_instance. For example, if the .ear file name is igs-service.ear, the instance name would be igs-service_wls_instance.
Prepare to deploy the IGS application:	The recommended location is rib-home/tools-home directory. There is an empty integration-bus-gateway-services subdirectory already there. This is only a placeholder.
Download the IntegrationGatewayService19.1.000ForAll19.1.000Apps_eng_ga.tar	
Extract the tar file	>tar -xvf IntegrationGatewayService19.1.000ForAll19.1.000Apps_eng_ga.tar
Modify the IgsConfig.properties file	Update the WlsUrl property in this file to the WebLogic URL where IGS is going to be deployed. For example, t3://igshost.example.com:18001 Update the WlsTarget property in this file tothe name of the WebLogic instance to which it will be deployed. For example, igs-service-wls-instance
Install IGS	Run the igs-install.sh located under rib-home/tools-home/integration-bus-gateway-services/bin The script will prompt for the WebLogic user name and password. The script will configure the server and install IGS.

IGS - Verify Installation

Task	Notes
Verify the IGS Application installation using the Administration Console.	For the Test Client link to be visible the server must be in Development mode. For more detailed verification testing, see Chapter 4, "Integration Gateway Service (IGS) Testing," in the *Oracle Retail Integration Guide Operations Guide.*
Navigate to Deployments page.	Navigate to the Deployments page. On the Summary of Deployments page, locate the igs-service on. Click the plus sign next to the ig-service to expand the tree. Locate the Web Services section.
Click any Web service to move to **Settings for ASNInPublishingService** page.	For example, ASNInPublishingService.
Select the Testing tab.	Click the **+** next to the service name to expand the tree. Locate the **Test Client** link. Select to move to the **WebLogic Test Client** page.
Select **Ping** operations.	Select the **Ping** operation. Fill in test data in the string arg0: text box. Click **Ping**. The test page will show the request message and the response message.

C

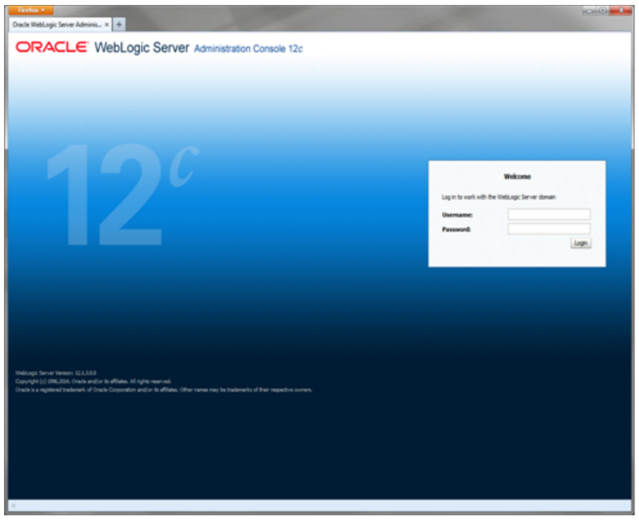
Appendix: Changing the RIB Admin GUI Password

This appendix describes the steps required to change the RIB Admin GUI password.

Procedure

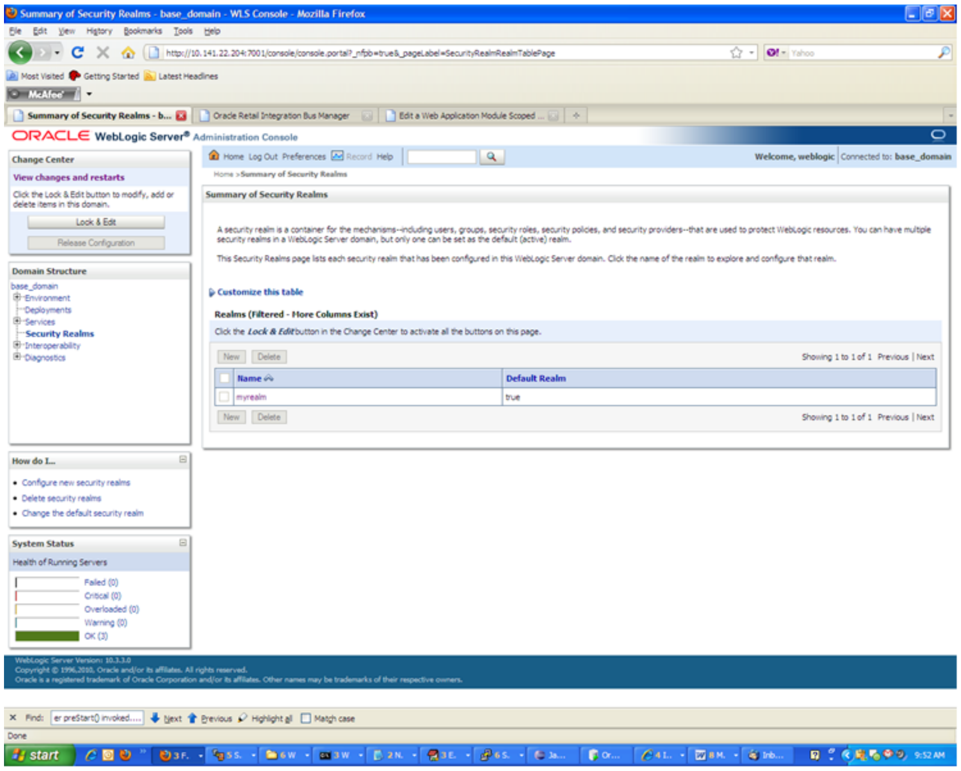
To change the RIB Admin GUI password, complete the following steps.

1. Log in to the WebLogic Server Administration Console.


ORACLE WebLogic Server Administration Console 12c
12c
Welcome
Log in to work with the WebLogic Server domain
Username:
Password:
Login

WebLogic Server Version: 12.1.3.0.0
Copyright (c) 1996,2014, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

2. In the left panel, click the **Security Realms** link.



Summary of Security Realms

A security realm is a container for the mechanisms--including users, groups, security roles, security policies, and security providers--that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

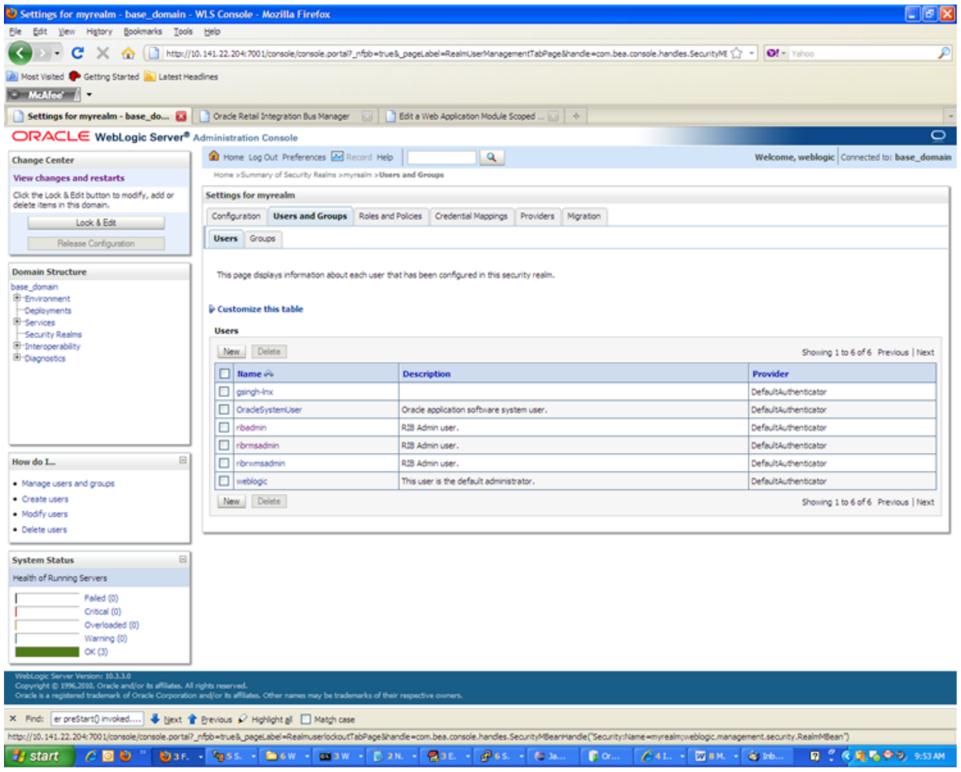
Realms (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

Showing 1 to 1 of 1 Previous | Next

Name	Default Realm
myrealm	true

3. Click the realm name. Go to the Users and Groups tab.



Settings for myrealm

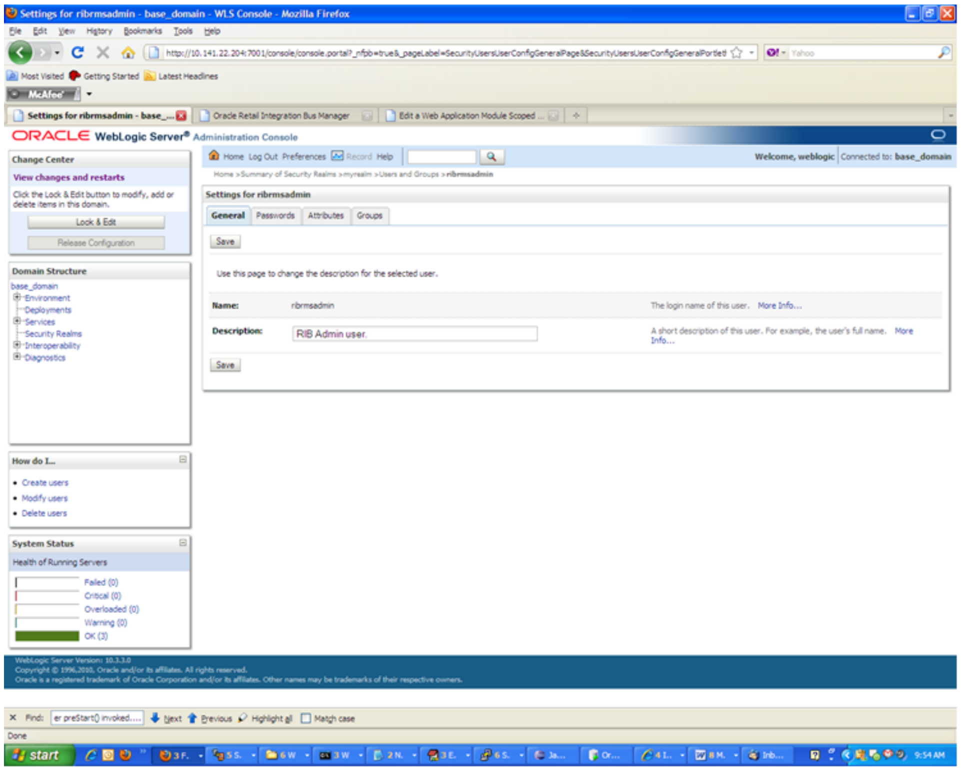
This page displays information about each user that has been configured in this security realm.

Users

Showing 1 to 6 of 6 Previous | Next

Name	Description	Provider
gsingh-lnx		DefaultAuthenticator
OracleSystemUser	Oracle application software system user.	DefaultAuthenticator
ribadmin	RIB Admin user.	DefaultAuthenticator
ribrmsadmin	RIB Admin user.	DefaultAuthenticator
ribrwmsadmin	RIB Admin user.	DefaultAuthenticator
weblogic	This user is the default administrator.	DefaultAuthenticator

4. Click the user name for which you want to change the password.



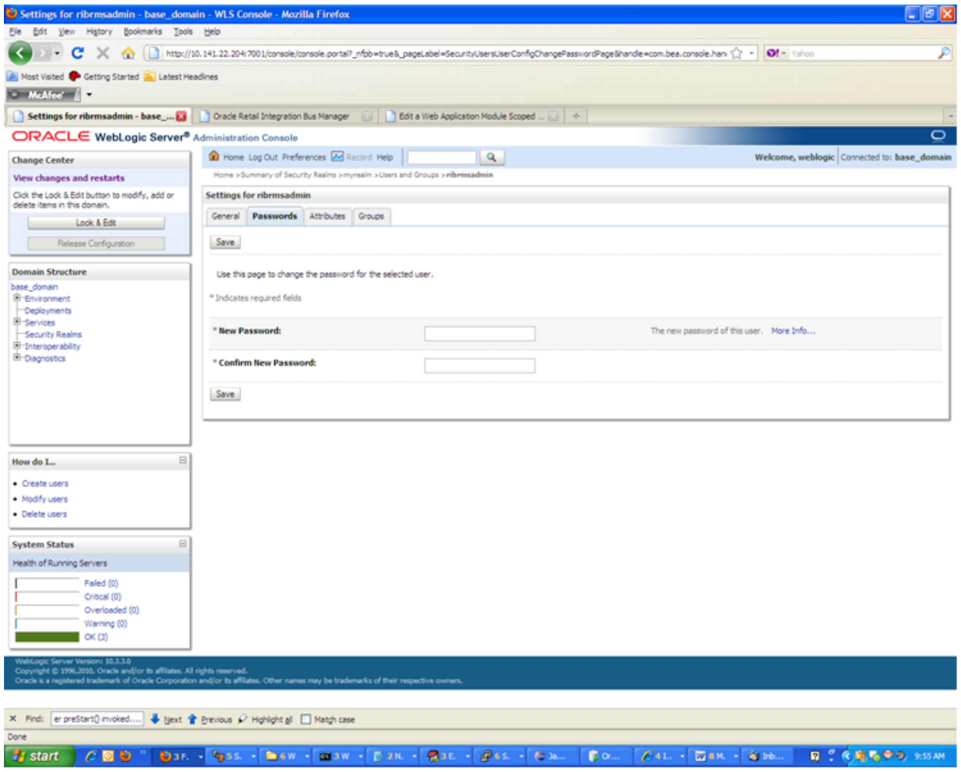
Settings for ribrmsadmin

General | Passwords | Attributes | Groups

Use this page to change the description for the selected user.

Name:	ribrmsadmin	The login name of this user. More Info...
Description:	RIB Admin user.	A short description of this user. For example, the user's full name. More Info...

5. Click the **Passwords** tab.



Settings for ribrmsadmin

General | Passwords | Attributes | Groups

Use this page to change the password for the selected user.

* Indicates required fields

* New Password:		The new password of this user. More Info...
* Confirm New Password:		

6. Enter the new password. Click **Save.**

D

Appendix: configWss.py

This appendix includes a code sample for configWss.py.

```
userName = sys.argv[1]
passWord = sys.argv[2]
url="t3://"+sys.argv[3]+":"+sys.argv[4]
print "Connect to the running adminSever"
connect(userName, passWord, url)
edit()
startEdit()
#Enable assert x509 in SecurityConfiguration
rlm = cmo.getSecurityConfiguration().getDefaultRealm()
ia = rlm.lookupAuthenticationProvider("DefaultIdentityAsserter")
activeTypesValue = list(ia.getActiveTypes())
existed = "X.509" in activeTypesValue
if existed == 1:
  print 'assert x509 is aleady enabled'
else:
  activeTypesValue.append("X.509")
ia.setActiveTypes(array(activeTypesValue,java.lang.String))
ia.setDefaultUserNameMapperAttributeType('CN');
ia.setUseDefaultUserNameMapper(Boolean('true'));

#Create default WebServcieSecurity

securityName='default_wss'
defaultWss=cmo.lookupWebserviceSecurity(securityName)
if defaultWss == None:
  print 'creating new webservice security bean for: ' + securityName
  defaultWss = cmo.createWebserviceSecurity(securityName)
else:
  print 'found exsiting bean for: ' + securityName

#Create credential provider for DK

cpName='default_dk_cp'
wtm=defaultWss.lookupWebserviceCredentialProvider(cpName)
if wtm == None:
wtm = defaultWss.createWebserviceCredentialProvider(cpName)
wtm.setClassName('weblogic.wsee.security.wssc.v200502.dk.DKCredentialProvider')
wtm.setTokenType('dk')
cpm = wtm.createConfigurationProperty('Label')
cpm.setValue('WS-SecureConversationWS-SecureConversation')
cpm = wtm.createConfigurationProperty('Length')
cpm.setValue('16')

else:
  print 'found exsiting bean for: DK ' + cpName

#Create credential provider for x.509

cpName='default_x509_cp'
wtm=defaultWss.lookupWebserviceCredentialProvider(cpName)
if wtm == None:
wtm = defaultWss.createWebserviceCredentialProvider(cpName)
wtm.setClassName('weblogic.wsee.security.bst.ServerBSTCredentialProvider')
wtm.setTokenType('x509')
else:
  print 'found exsiting bean for: x.509 ' + cpName


#Custom keystore for xml encryption

cpName='ConfidentialityKeyStore'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
keyStoreName=sys.argv[5]
cpm.setValue(keyStoreName)
cpName='ConfidentialityKeyStorePassword'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
cpm.setEncryptValueRequired(Boolean('true'))
KeyStorePasswd=sys.argv[6]
cpm.setEncryptedValue(KeyStorePasswd)
cpName='ConfidentialityKeyAlias'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
keyAlias=sys.argv[7]
cpm.setValue(keyAlias)
cpName='ConfidentialityKeyPassword'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty('ConfidentialityKeyPassword')
cpm.setEncryptValueRequired(Boolean('true'))
keyPass=sys.argv[8]
cpm.setEncryptedValue(keyPass)

#Custom keystore for xml digital signature

cpName='IntegrityKeyStore'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
keyStoreName=sys.argv[5]
cpm.setValue(keyStoreName)
cpName='IntegrityKeyStorePassword'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
cpm.setEncryptValueRequired(Boolean('true'))
KeyStorePasswd=sys.argv[6]
cpm.setEncryptedValue(KeyStorePasswd)
cpName='IntegrityKeyAlias'
cpm=wtm.lookupConfigurationProperty(cpName)
```

```
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
keyAlias=sys.argv[7]
cpm.setValue(keyAlias)
cpName='IntegrityKeyPassword'
cpm=wtm.lookupConfigurationProperty(cpName)
if cpm == None:
cpm = wtm.createConfigurationProperty(cpName)
cpm.setEncryptValueRequired(Boolean('true'))
keyPass=sys.argv[8]
cpm.setEncryptedValue(keyPass)

#Create token handler for x509 token

#cpName='default_x509_handler'
th=defaultWss.lookupWebserviceTokenHandler(cpName)
if th == None:
th = defaultWss.createWebserviceTokenHandler(cpName)
th.setClassName('weblogic.xml.crypto.wss.BinarySecurityTokenHandler')
th.setTokenType('x509')
cpm = th.createConfigurationProperty('UseX509ForIdentity')
cpm.setValue('true')
save()
activate(block="true")
disconnect()
exit()
```

E

Appendix: RIB Domain Configuration for Policy A

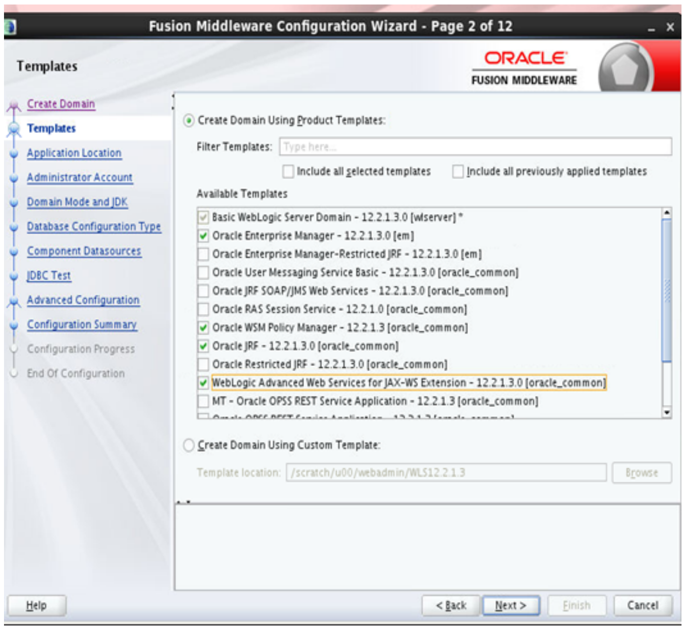
The RIB Domain Configuration for Policy A appendix provides instructions to deploy wsm-pm and EM apps, as well to configure ssl port for wsm-pm.

RIB Domain Creation Template Options

Perform the following procedure to configure the RIB domain:

1. Select the following RIB domain creation template options:

Figure E–1 Domain Creation Template Options



Fusion Middleware Configuration Wizard - Page 2 of 12

Templates

(●) Create Domain Using Product Templates:

Filter Templates: Type here...

☐ Include all selected templates ☐ Include all previously applied templates

Available Templates

- ✓ Basic WebLogic Server Domain - 12.2.1.3.0 [wlserver] *
- ✓ Oracle Enterprise Manager - 12.2.1.3.0 [em]
- ☐ Oracle Enterprise Manager-Restricted JRF - 12.2.1.3.0 [em]
- ☐ Oracle User Messaging Service Basic - 12.2.1.3.0 [oracle_common]
- ☐ Oracle JRF SOAP/JMS Web Services - 12.2.1.3.0 [oracle_common]
- ☐ Oracle RAS Session Service - 12.2.1.0 [oracle_common]
- ✓ Oracle WSM Policy Manager - 12.2.1.3 [oracle_common]
- ✓ Oracle JRF - 12.2.1.3.0 [oracle_common]
- ☐ Oracle Restricted JRF - 12.2.1.3.0 [oracle_common]
- ✓ WebLogic Advanced Web Services for JAX-WS Extension - 12.2.1.3.0 [oracle_common]
- ☐ MT - Oracle OPSS REST Service Application - 12.2.1.3 [oracle_common]

( ) Create Domain Using Custom Template:

Template location: /scratch/u00/webadmin/WLS12.2.1.3

2. On the Deployments and Services page, target the wsm-pm application on the Admin server.

> **Note:** By default, wsm-pm is not targeted on any server. If not targeted, domain creation will fail.

3. Click **Next** and verify the domain created successfully.

Resolving wsm-pm Issues

In case issues are encountered for wsm-pm, perform the following steps to point wsm-pm to the correct port:

1. Access the enterprise Manager URL of WebLogic.

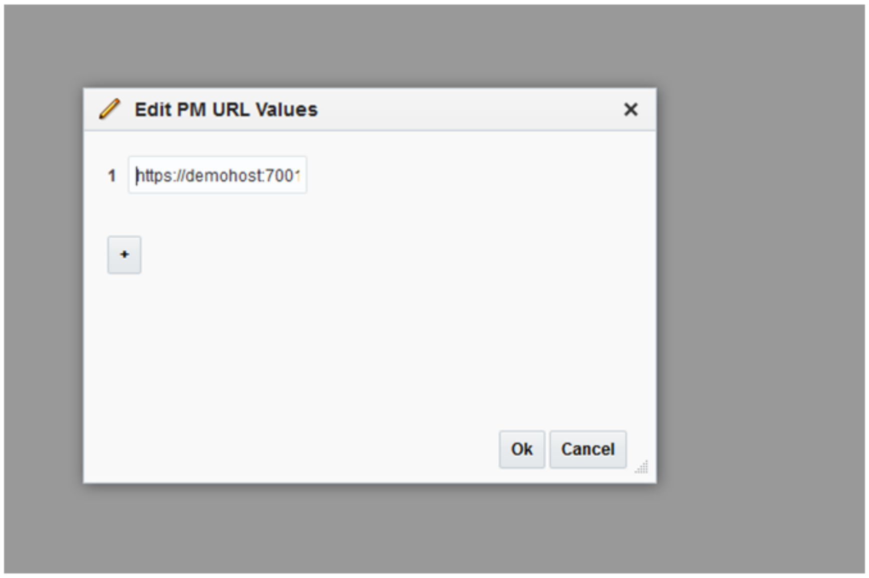
   For example: https://<host>:<port>/em

2. From the navigation pane, expand WebLogic Domain and select the domain to be configured.
3. From the WebLogic Domain menu, select Web Services, then WSM Domain Configuration.
4. Select the Policy Access tab.
5. In the Policy Manager section of the page, clear the Auto Discover check box. The PM URL Edit button is enabled.
6. Click the PM URL Edit button.
7. In the Edit PM URL Values page, click the sign and enter the URL for the Administration Server, such as https://host:*admin_port*/wsm-pm.

   For example, https://localhost:9002/wsm-pm.

8. Click **OK** to close the window.

Figure E–2   Edit the Policy Manager URL



9. Click **Apply** on the Policy Access page.