Oracle® Retail Integration Bus

Installation Guide

Release 13.2.4

E28080-04

January 2014

ORACLE®

Oracle Retail Integration Bus Installation Guide, Release 13.2.4

Primary Author: Kris Lange

VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

4 Run the RIB Application Installer

5 Post-Installation Tasks

6 Integration Gateway Services Installation Tasks

7 RIB Security

A Appendix: RIB Application Installer Screens

B Appendix: RIB Installer Common Errors

C Appendix: RIB Installation Checklists

D Appendix: Changing the RIB Admin GUI Password

Send Us Your Comments

Oracle Retail Integration Bus Installation Guide, Release 13.2.4

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

Related Documents

For more information, see the following documents in the Oracle Retail Integration Bus 13.2.4 documentation set:

- *Oracle Retail Integration Bus Data Model*
- *Oracle Retail Integration Bus Implementation Guide*
- *Oracle Retail Integration Bus Operations Guide*
- *Oracle Retail Integration Bus Release Notes*
- *Oracle Retail Integration Bus Hospital Administration Guide*
- *Oracle Retail Functional Artifacts Guide*
- *Oracle Retail Functional Artifact Generator Guide*
- *Oracle Retail Service-Oriented Architecture Enabler Tool Guide*

Customer Support

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

`https://support.oracle.com`

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create

- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.1) or a later patch release (for example, 13.1.2). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:

http://www.oracle.com/technology/documentation/oracle_retail.html

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

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 RIB's Java EE applications (rib-<app>.ear) on Java EE 5 compliant application server.
- 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 13.2 application cannot be installed over an existing version, such as 13.1.1.

RIB Installation Master Checklist

This list covers all of the sequential steps required to perform a full installation of the RIB, using either the GUI RIB Installer (strongly recommended) or a command line installation.

Task	Notes
Prepare the Oracle WebLogic Servers for installation of the RIB Components.	Prerequisite

Task	Notes
WLS 10.3.4 Patches for RIB	Download the following from My Oracle Support. ■ Patch 11818904: SU Patch [G1X4]: ENHANCEMENT REQUEST TO CONTROL INITIAL STATE(START/STOP) OF A MDB ■ Patch 13415672: SU Patch [LV78]: SUSPEND OPERATION ON MDB IS TAKING VERY LONG TIME
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.	
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 one of these methods: Installation using the RIB Installer GUI or Installation using the RIB App Builder Command Line Tools.	It is strongly recommended that the RIB Installer GUI method be used.
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 C, "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.	

Task	Notes
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 C, "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.

Check Server Requirements

Supported On	Versions Supported
Database Server OS	OS certified with Oracle Database 11gR2 Enterprise Edition. Options are: ■ Oracle Enterprise Linux 5 Update 5 for x86-64 (actual hardware or Oracle virtual machine) ■ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (actual hardware or Oracle virtual machine) ■ IBM AIX 6.1 (actual hardware or LPARs) ■ Solaris 10 Sparc (actual hardware or logical domains) ■ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.2) with the following specifications: **Components:** ■ Oracle Partitioning ■ Examples CD (formerly the companion CD) **One-off Patches:** ■ 10170431—CTWR consumes a lot of CPU cycles. If ASM is used, apply the following patch to Oracle Database home. ■ 11808931—Merge request on top of 11.2.0.2.0 for defects 10410054 and 1042216. **Other Components:** ■ Perl Compiler 5.0 or later ■ X-Windows interface
AQ JMS Server	Oracle Database 11g R2

Supported On	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.4). Options are: ■ Oracle Linux 5 Update 5 for x86-64 (actual hardware or Oracle virtual machine) ■ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (actual hardware or Oracle virtual machine) ■ IBM AIX 6.1 (actual hardware or LPARs) ■ Solaris 10 Sparc (actual hardware or logical domains) ■ HP-UX 11.31 Integrity (actual hardware or HPVM)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.4) **Components:** ■ Oracle WebLogic Server 11g Release 1 (10.3.4)
Minimum required JAVA version for Solaris OS	■ JDK 1.6.0_18+ 64 bit ■ Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only.
Minimum required JAVA version for all other operating systems	■ JDK 1.6.0+ 64 bit ■ Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only.

Important: If there is an existing WebLogic 10.3.3 installation on the server, you must upgrade WebLogic 10.3.3 to WebLogic 10.3.4. All middleware components associated with WebLogic server 10.3.3 should be upgraded to 11.1.1.4.

Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.

For information about how to complete the upgrade to WebLogic 10.3.4, see the My Oracle Support document, "How to Upgrade from WebLogic11g 10.3.3 to WebLogic11g 10.3.4" (ID 1432575.1).

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 11.1.1.4 or higher is available in the WebLogic Application Server (10.3.4) and applied to the domain where RIHA will be installed.

Other Resources

For information about WebLogic Application Server 11g, see the Oracle WebLogic Server Documentation Library:

- WebLogic Application Server 11g - Index
  http://download.oracle.com/docs/cd/E15523_01/index.htm
- WebLogic Application Server 11g - Documents
  http://download.oracle.com/docs/cd/E15523_01/wls.htm.

Note: See also the Oracle Database Administrator's Guide 11g Release 2 (11.2) and the Oracle WebLogic Application Server 11g (10.3.4) documentation.

RIB Integration Gateway Services (IGS) Supported Operating Systems

Supported On	Version Supported
Oracle WebLogic Server OS	OS certified with OracleWebLogic Server 11g 10.3.4. Options are: ■ Oracle Enterprise Linux 5 Update 5 for x86-64 (Actual hardware or Oracle virtual machine) ■ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.3) for x86-64 (Actual hardware or Oracle virtual machine) ■ IBM AIX 6.1 (Actual hardware or LPARs) ■ Solaris 10 Sparc (Actual hardware or logical domains) ■ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Oracle WebLogic Server	Oracle WebLogic Server 11g (10.3.4)

Supported Oracle Retail Products

Supported On	Version Supported
RWMS 13.2.4	RIB 13.2.4
RMS 13.2.4	RIB 13.2.4
RPM 13.2.4	RIB 13.2.4
SIM 13.2.4	RIB 13.2.4
AIP 13.2.4	RIB 13.2.4

Supported Oracle Applications

Supported On	Version Supported
Oracle E-Business Suite (General Ledger and Accounts Payable)	Oracle Application Integration Architecture (AIA) Media Pack 2.5 Oracle E-Business Suite 12.1.1 and 12.1.3 integration is supported using the Oracle Financial Operations Control Integration Pack for Oracle Retail Merchandising Suite and Oracle E-Business Suite Financials. See the *Oracle® Application Integration Architecture 2.5: Installation and Upgrade Guide* for specific version information.

Supported On	Version Supported
PeopleSoft Enterprise Financials	Oracle Application Integration Architecture (AIA) Media Pack 2.5 PeopleSoft Enterprise Financials integration is supported using the Oracle Retail Merchandising Integration Pack for PeopleSoft Enterprise Financials: Financial Operations Control. See the Oracle® Application Integration Architecture 2.5: Installation and Upgrade Guide for specific version information.

The 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, the RIB can take advantage of database RAC capability for its JMS provider. By using RAC AQ as the 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 a Oracle RAC database, the only configuration change required in rib-<app> is the RAC JDBC connection URL.

Note: The RIB supports only the use of the Oracle Type 4 Thin Java Database Connectivity (JDBC) driver (ojdbc6.jar) for all JDBC connections, including RAC.

The RIB and Oracle WebLogic Server Cluster

The RIB uses JMS server for message transportation between the integrating retail applications. Because the 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.

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.

Check for the Current Version of the Installation Guide

Corrected versions of Oracle Retail installation guides may be published whenever critical corrections are required. For critical corrections, the rerelease of an installation guide may not be attached to a release; the document will simply be replaced on the Oracle Technology Network Web site.

Before you begin installation, check to be sure that you have the most recent version of this installation guide. Oracle Retail installation guides are available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technology/documentation/oracle_retail.html

If a more recent version of this installation guide is available, that version supersedes all previous versions. Only use the newest version for your installation.

An updated version of an installation guide is indicated by 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 an installation guide with part number E123456-01.

Determine the UNIX User Account to Install the Software

The user account that installs the 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>-wls-instance. 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 optional to create a new domain or to use the base/default domain of WLS.

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 11g Release 1 (10.3.4).

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

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

- rib-func-artifact-wls-instance. (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-wls-instance. (It is recommended, but not required, that this naming convention be followed).

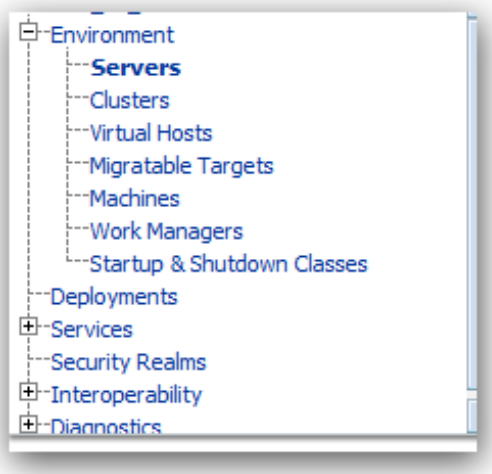
The following is a list of optional application instances, depending on deployment choices. It is recommended, but not required, that you use the following naming convention:

- rib-aip-wls-instance
- rib-rfm-wls-instance
- rib-rms-wls-instance
- rib-rpm-wls-instance
- rib-rwms-wls-instance
- rib-sim-wls-instance

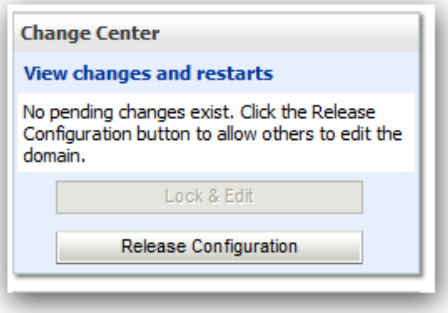
Note: See Oracle WebLogic Server 11g Release 3 (10.3.4) documentation for more details on How to Create managed servers.

To create the rib<app>wls instance, 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.



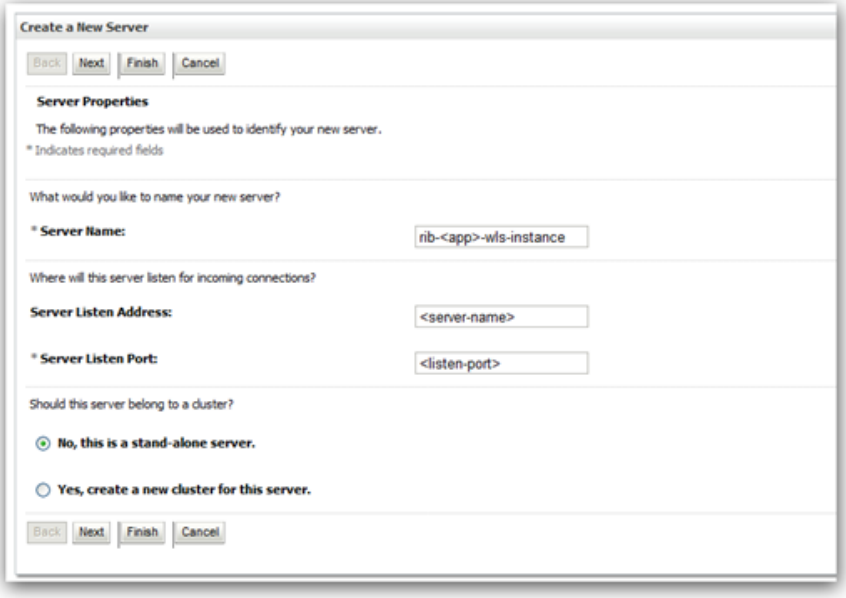
3. Click **Lock & Edit**.



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

   For example:

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

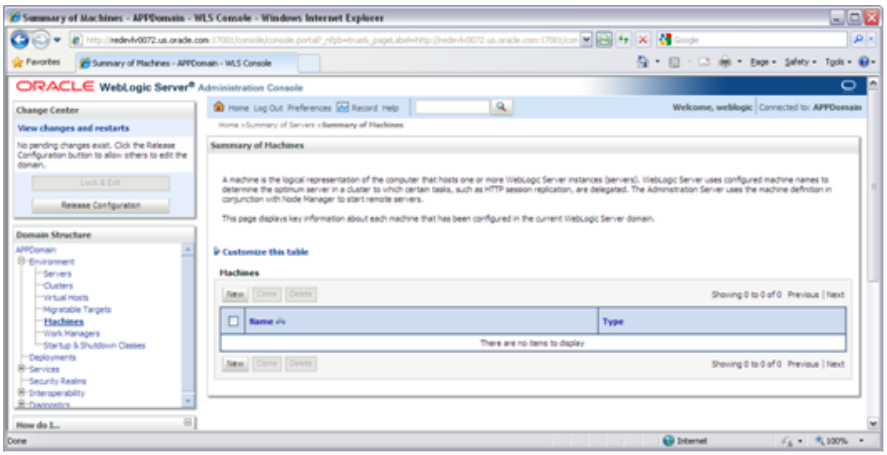


6. Click **Next**. Click **Finish**. Make sure you see this instance listed under Servers

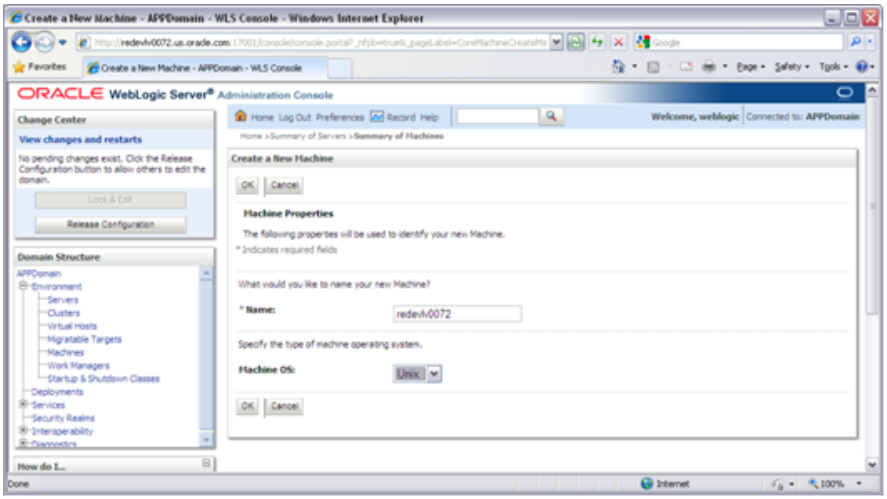
Install NodeManager

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.

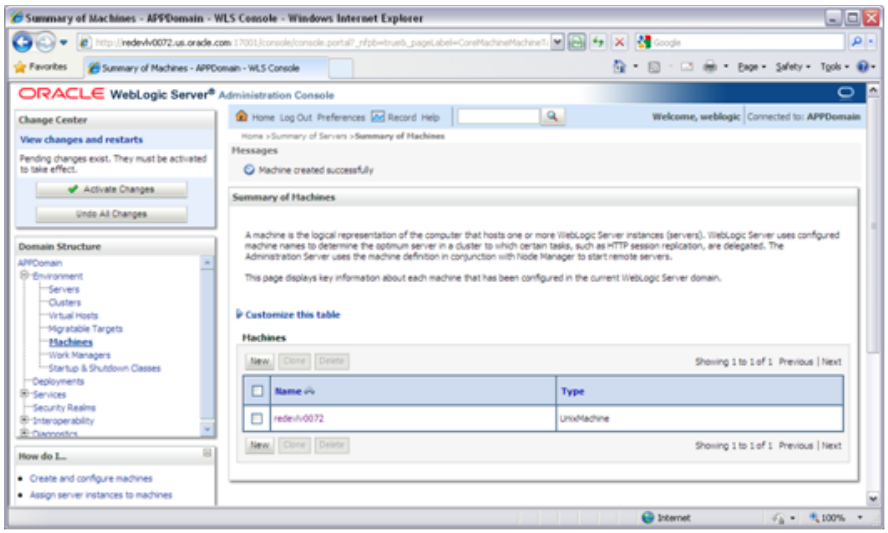
1. Log in to the administration console.
2. Click **Lock & Edit** and navigate to Environments> Machines.



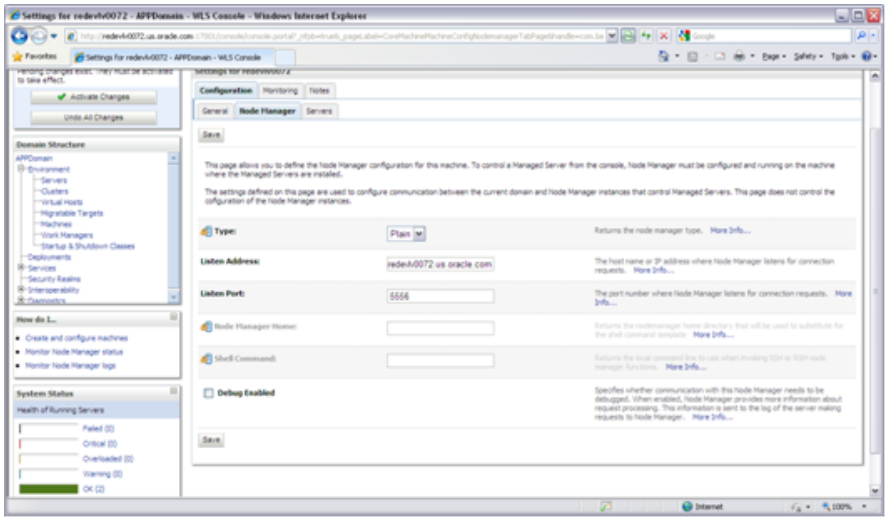
3. Click **New**.



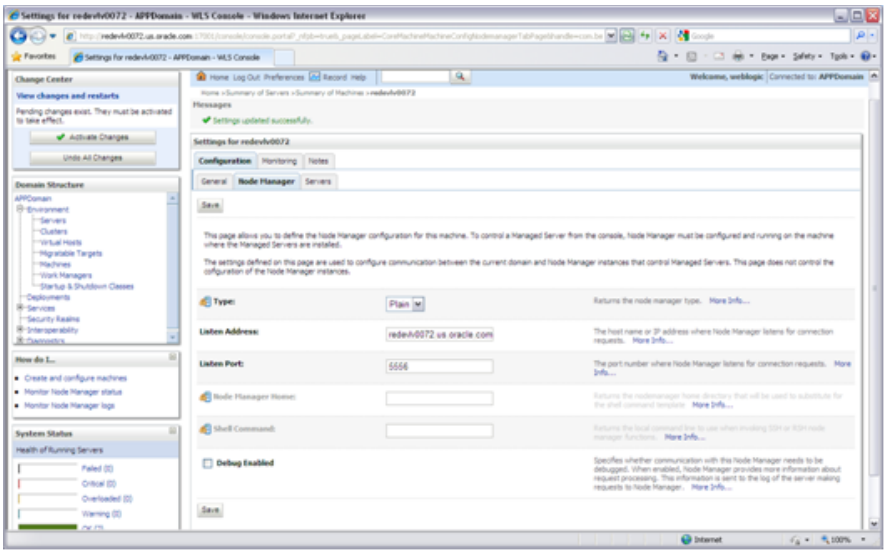
4. Set the following variables:
   - Name: Logical machine name
   - Machine OS: UNIX
5. Click **OK**.
6. Click on the machine created



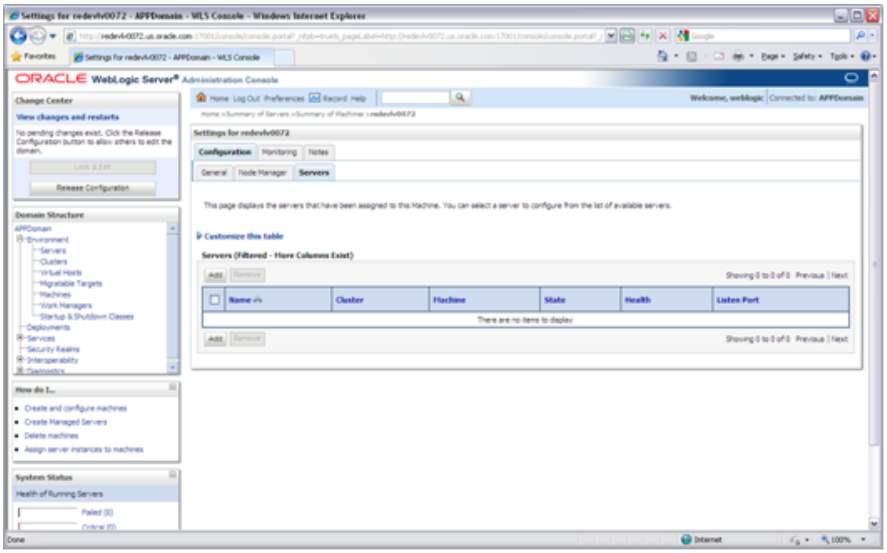
7. Click the NodeManager tab and update the details below.
   - **Type:** Plain
   - **Listen Address**: For example, redevlv0072.us.oracle.com
   - **Listen Port**: Default port (for example, 5556) or any available port



8. Click **Save**.



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



12. Add the managed servers that need to be configured with NodeManager. Save the changes.
13. Click **Add** to repeat for additional servers.
14. Click **Activate Changes**.
15. Start NodeManager from the server using the startNodeManager.sh at $WL_HOME/wlserver_10.3/server/bin.

Note: To activate changes the server must be stopped: $WL_HOME/user_projects/domains/<RIB_Domain>/bin/stopManagedWebLogic.sh <rpm>-server ${server_name}:${server_port}

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

    $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties

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

17. 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.

18. After NodeManager is restarted, log in to the Administration Console and select environments servers. For each managed server, do the following.
    1. Select Environments.
    2. From Servers, select the managed server.
    3. Select the Configuration tab.
    4. Select the Server Start tab.
    5. In the text area for each managed server, enter -Xms1024m -Xmx2048m -XX:MaxPermSize=512m.

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 (RibKernel13.2.4ForAll13.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 RibKernel13.2.4ForAll13.x.xApps_eng_ga.jar.
4. Change directories to Rib13.2.4ForAll13xxApps/rib-home. This location will be referred to as <RIB_HOME> for the remainder of this chapter.

Configure the rib-<app>-wls-instance

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

1. Configure the startup script

1. Take a backup of the script $DOMAIN_HOME/base_domain/bin/ startWebLogic.sh
2. Edit the script $DOMAIN_HOME/base_domain/bin/ startWebLogic.sh to add the following attributes.

Note: If using jrockit jdk, add the following:
USER_MEM_ARGS="-Xms1024m -Xmx2048m -XnoOpt"

```
CLASSPATH=$DOMAIN_HOME/servers/$SERVER_NAME:$CLASSPATH
JAVA_OPTIONS="-Dweblogic.ejb.container.MDBMessageWaitTime=2 ${JAVA_
OPTIONS}"
JAVA_VM="-server"
USER_MEM_ARGS="-Xms1024m -Xmx2048m -XX:MaxPermSize=512m"
```

The following is a portion of a startWebLogic.sh sample.

```
echo "."

echo "."

echo "JAVA Memory arguments: ${MEM_ARGS}"

echo "."

echo "WLS Start Mode=${WLS_DISPLAY_MODE}"

echo "."

CLASSPATH=$DOMAIN_HOME/servers/$SERVER_NAME:$CLASSPATH
JAVA_OPTIONS="-Dweblogic.ejb.container.MDBMessageWaitTime=2 ${JAVA_
OPTIONS}"
JAVA_VM="-server"
USER_MEM_ARGS="-Xms1024m -Xmx2048m -XX:MaxPermSize=512m"

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 "*************************************************"

# CLASS CACHING

if [ "${CLASS_CACHE}" = "true" ] ; then
echo "Class caching enabled..."
```

```
JAVA_OPTIONS="${JAVA_OPTIONS} -Dlaunch.main.class=${SERVER_CLASS}
-Dlaunch.class.path="${CLASSPATH}"
-Dlaunch.complete=weblogic.store.internal.LockManagerImpl -cp ${WL_
HOME}/server/lib/pcl2.jar"
export JAVA_OPTIONS
SERVER_CLASS="com.oracle.classloader.launch.Launcher"
fi

# START WEBLOGIC

echo "starting weblogic with Java version:"

${JAVA_HOME}/bin/java ${JAVA_VM} -version

if [ "${WLS_REDIRECT_LOG}" = "" ] ; then
echo "Starting WLS with line:"
echo "${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}"
${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 ${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

stopAll
```

Note: In the startWebLogic script, the above statements must be added before the call is made to start the server.

2. Update $WL_HOME/<wlserver_10.3>/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/ 10.3.X_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";
};
```

3. Move the RIB packaged jars to the server library.

   Copy aqapi.jar and ojdbc6.jar from rib-home/integration-lib/third-party/oracle/db/11.2.0.1.0 to $WL_HOME/wlserver_10.3/server/lib.

   Example:

   [11.2.0.1.0]$cp aqapi.jar $WL_HOME/wlserver_10.3/server/lib

   [11.2.0.1.0]$cp ojdbc6.jar $WL_HOME/wlserver_10.3/server/lib

4. Start rib-<app>managed server.

   WebLogic managed servers where rib-<app> is deployed can be started in 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 the DOMAIN_HOME/bin

      For example:

      $cd product/10.3.X_RIB/WLS/user_projects/domains/RIBDomain/bin

   3. Run the startManagedWebLogic script.

      For example:

      sh startManagedWebLogic.sh rib-rms-wls-instance http:localhost7001

   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 installer GUI or on 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 the 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 Multibyte Deployments

If the RIB is deployed into an environment where multibyte 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 Multibyte 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. In previous releases, rib-<app> used the respective retail application database schema for its location of the RIB Hospital tables. In this release, 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 C, "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 rib-private-kernel-database-library.zip file. The rib-private-kernel-database-library.zip can be found in the rib-home directory structure.

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 11gR2. Use the grants shown below to enable XA transactions for the RIB database user.

- grant select on v$xatrans$ to public
- grant select on pending_trans$ to public

- grant select on dba_2pc_pending to public
- grant select on dba_pending_transactions to public
- grant execute on dbms_system to public

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, 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 uses both mechanism, whereas RWMS uses only Oracle Objects to communicate with RIB.

1. Verify that the RMS and RWMS database schema has 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's database schema already have 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*.

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.
4. Verify that the RMS retail application's database schema has all the database objects defined equivalent to the ones defined in the RIB delivered script called 1_CLOB_CREATE_OBJECTS.SQL.

   Note: See the *Oracle Retail Integration Bus Operations Guide*.

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 13.x, 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 "TAFRHOSP"

IDENTIFIED BY "TAFRHOSP"

DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP";

GRANT "CONNECT" TO " TAFRHOSP ";

GRANT "RESOURCE" TO " TAFRHOSP ";

ALTER USER "TAFRHOSP"

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 rib-private-kernel-database-library.zip file. The rib-private-kernel-database-library.zip can be found in the rib-home/integration-lib/internal-build/rib/ directory structure.

Prepare Oracle AQ JMS Provider

Oracle Streams AQ is the JMS provider that RIB uses for a synchronous 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 "RIBAQ" IDENTIFIED BY "RIBAQ"
DEFAULT TABLESPACE "RETEK_DATA"
TEMPORARY TABLESPACE "TEMP";
GRANT "CONNECT" TO "RIBAQ";
GRANT "RESOURCE" TO "RIBAQ";
GRANT CREATE SESSION TO "RIBAQ";
GRANT EXECUTE ON "SYS"."DBMS_AQ" TO "RIBAQ";
GRANT EXECUTE ON "SYS"."DBMS_AQADM" TO "RIBAQ";
GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO "RIBAQ";
GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO "RIBAQ";
GRANT "AQ_ADMINISTRATOR_ROLE" TO "RIBAQ";
ALTER USER "RIBAQ"
QUOTA UNLIMITED ON RETEK_DATA;
```

Note: See also:

Oracle® Database Administrator's Guide 11g Release 2 (11.2)

Oracle® Streams Advance Queuing User's Guide and Reference 11g Release 2 (11.2)

4

Run the RIB Application Installer

This chapter provides instructions for running the RIB Application Installer.

> **Note:** If there is an existing WebLogic 10.3.3 installation on the server, you must upgrade WebLogic 10.3.3 to WebLogic 10.3.4. All middleware components associated with WebLogic server 10.3.3 should be upgraded to 11.1.1.4.
>
> Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.
>
> For information about how to complete the upgrade to WebLogic 10.3.4, see the My Oracle Support document, "How to Upgrade from WebLogic11g 10.3.3 to WebLogic11g 10.3.4" (ID 1432575.1).

RIB Application Installer Tasks

The RIB application installer 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.
- Generate the rib-deployment-env-info.xml file, which configures the RIB installation.
- 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.

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

Oracle Configuration Manager

The Oracle Retail OCM Installer packaged with this release installs the latest version of OCM.

The document, "Oracle Configuration Manager Installer Guide" (ID 1071030.1), is available through My Oracle Support:

https://support.oracle.com

This document describes the procedures and interfaces of the Oracle Retail Oracle Configuration Manager Installer that a retailer runs at the beginning of the installation process.

OCM Documentation Link

http://www.oracle.com/technology/documentation/ocm.html

How to Run the RIB Application Installer

To run the RIB application installer, do the following.

1. Undeploy all old rib-apps and completely remove them from the AdminServer upload directory as follows.
    1. Shut down all the rib-* servers.
    2. 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 deploymets will break.:

    3. Remove them from the upload directory:

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

    4. Start the rib-servers up again before starting the installer.
2. Expand the RIB Kernel distribution as described above.
3. Download the RIB Functional Artifacts distribution (RibFuncArtifact13.2.3ForAll13.2.3Apps_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 (RibPak13.2.4For<app>13.2.4_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 (Rdmt13.2.4ForAll13.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. This allows the installer to run the <RIB_HOME>/tools-home/rdmt/configbuilder.sh script as part of 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 1.6.0+ 64 bit or Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only.
8. Be sure there are no pre-existing instances of ANT in your path:

$ unset ANT_HOME

$ unset ANT_CONTRIB

$ unset CLASSPATH

The following command should not find the ant executable:

$ which ant

9. If you are using an X server, such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
10. Make sure that all WebLogic instances that you intend to deploy to are currently running.
11. Give execute permissions to rib-home:

    For example, chmod -R 700 rib-home.
12. Change directories to the <RIB_HOME> directory.
13. Run the rib-installer.sh script. The RIB installer appears.

> **Note:** See Appendix A, "Appendix: RIB Application Installer Screens" for details about every screen and input field in the installer.

14. Restart the rib-<app>-wls-instances. 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.
15. If the installer finds the configbuilder.sh script, it will attempt to run it. However, if the installer is unable to run the RDMT script or if the RDMT setup failed for some reason, 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 installer with the existing <RIB_HOME>/deployment-home/conf/rib-deployment-env-info.xml file. Running the installer 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 installer.

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. Examine the rib-integration-flows.xml to identify all the RIB applications in the full integration flow.
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.

Note: If this is a first-time installation (or if you are using the installer to rewrite the rib-deployment-env-info.xml) you do not need to complete Step 6.

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, the installer tool does the following to complete preinstallation activities:

Note: This step is not required if this is a first time installation or when the installer is used for installation.

   - 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>).

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.

Backups and Logs Created by the Installer

The RIB application installer creates the following backup and log files:

- Each time the installer is used to generate a new rib-deployment-env-info.xml, a backup of the existing file will be created in: <RIB_HOME>/deployment-home/conf/archive/rib-deployment-env-info.xml.<timestamp>
- Each time the installer is run, the output of the installer script will be written to a log file. The installer's log file will be located in: <RIB_HOME/.retail-installer/rib/log/rib-install-app.<timestamp>.log.
- Each time the installer is run, the user inputs will be recorded in: <RIB_HOME/.retail-installer/rib/log/ant.install.properties.<timestamp>. This file should only be used during troubleshooting to verify the exact inputs that were given to the installer. Modifying the file is not recommended, as it is a record of the inputs at the time the installer was run.

Resolving Installation Errors

If an error is encountered while running the installer, the cause of the error must be corrected before making another attempt to run the installer. The installer's log file may contain helpful information for determining the cause of the error. After you have examined the log files, see "Appendix: RIB Installer Common Errors" for a list of commonly encountered errors.

When you are ready to attempt another installation, keep in mind that you may be able to avoid re-entering all your inputs if the previous installation process was far enough along to configure the rib-deployment-env-info.xml. If the installer has already generated the rib-deployment-env-info.xml file, or if you have manually edited the rib-deployment-env-info.xml file, then it is not necessary to re-enter all the inputs in the installer. Verify that the rib-deployment-env-info.xml contains the correct settings, and run the installer with the **Use existing rib-deployment-env-info.xml** option.

5

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

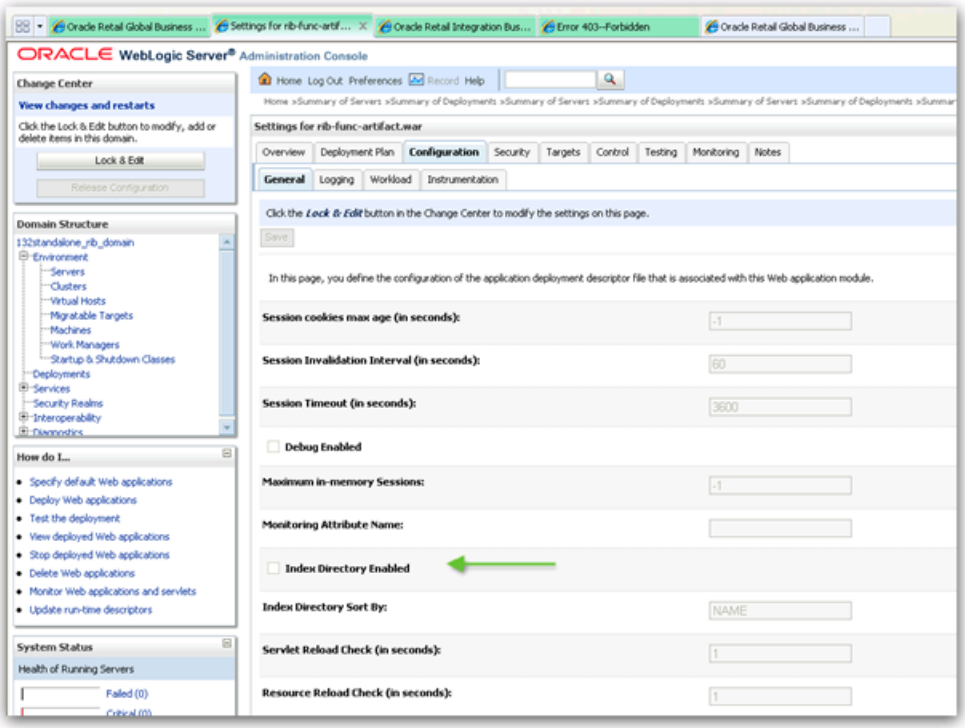
Verify that correct URL's to the RIB Functional Artifacts are configured in the Java EE Applications.

- Functional Artifact URL
- JNDI URL

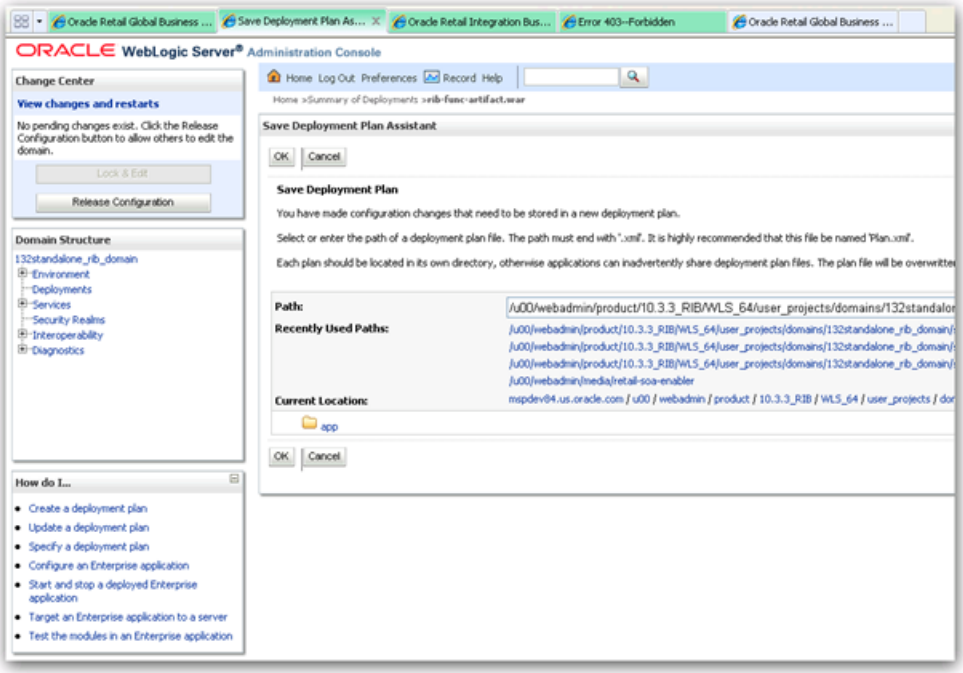
RIB-FUNC Post Deployment

If an Error 403-Forbidden screen is received upon launching the rib-func-artifact URL, complete the following steps to enable indexing & correct the error.

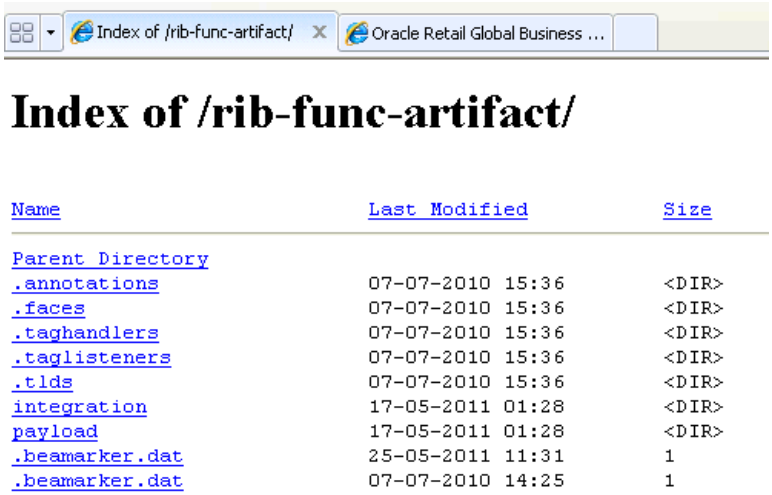
1. Go to the configuration tab of the rib-func-artifact.war deployment in the WLS console. In the Change Center, click **Lock & Edit**. Check the Index Directory Enabled box, as shown in the following illustration.



2. Click **Save**. Click **OK** to save deployment plan.



3. In the Change Center, click **Activate Changes** and relaunch the rib-func 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 5.0. 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 (Rdmt13.2.4ForAll13.x.xApps_eng_ga.tar) and extract it. >tar svf Rdmt13.2.4ForAll13.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	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt/
RDMTLOGS Directory	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt /RDMTLOGS
Temp Files Directory	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt /RDMTLOGS/tmp
RIB App Builder rib-home Directory	/u00/Rib1324ForAll13xxApps/rib-home

The following are parameters for the JMS provider.

Parameter	Setting
AQ JMS User ID	ribaq
AQ JMS Password	retail
JMS Connection URL	jdbc:oracle:thin:@host-name:port:sid

The following are WLS parameters for JMX functions:

Parameter	Setting
WLS/JMX Host	mspdev72
WLS Admin Port	8001
WLS Instance Name	rib-rms-wls-instance
WLS Instance Port	8002
WLS App Name	rib-rms
WLS User Name	weblogic
WLS Password	welcome1

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

Parameter	Setting
User Name	rms
Password	retail
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 10.3.4.

6

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 11g Release 3 (10.3.4.0) and Java 6.

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 6.

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. Every .ear file or ejb-jar file containing the services should be deployed on its own WebLogic 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, wls_instance.

   For example, if the .ear file name is igs-service.ear, the instance name would be igs-service_wls_instance.

Prepare Integration Gateway Services (IGS)

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

Option 1: Running IGS under $RIB_HOME

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

1. Download the IntegrationGatewayService13.2.4ForAll13.2.4Apps.tar and untar it under rib-home/tools-home.

   ```
   cd rib-home/tools-home/
   tar -xvf IntegrationGatewayService13.2.4ForAll13.2.4Apps_eng_ga.tar
   ```

2. Copy the ojdbc6.jar and aqapi.jar from $RIB_HOME/integration-lib/third-party/oracle/db/11.2.0.2.0 to WL_HOME/server/lib.

3. 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).

4. Go to rib-home/tools-home/integration-bus-gateway-services/bin. Run the igs-admin.sh. Running the script does the following:

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

     Note: The WebLogic user name used here should be set up with the admin 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 JSMS server information listed in the rib-deployment-env-info.xml.
   - Deploys the igs-service.ear to the WebLogic server.

   All items in Step 5 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. This action verifies whether the attempted IGS installation is from within the rib-home or in standalone mode; the preconfiguration cleanup is performed 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 rib-home/tools-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 the number of configured AQ JMS servers.

   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. If the igs-service.ear 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 here: $RIB_HOME/tools-home/integration-bus-gateway-services/log

---

Option 2: Running Standalone IGS

To run standalone IGS, complete the following steps.

1. Download the IntegrationGatewayService13.2.4ForAll13.2.4Apps.tar. Untar it in the directory from which you want to run IGS. For convenience, the directory in which igs is untarred is referred to as $IGS_HOME

2. Extract the contents of the .tar file.

   tar -xvf IntegrationGatewayService13.2.4ForAll13.2.4Apps_eng_ga.tar

3. Copy the ojdbc6.jar and aqapi.jar from $IGS_HOME/integration-lib/third-party/oracle/db/11.2.0.1.0 to WL_HOME/server/lib.

4. Under a "pre-configured" $RIB_HOME/deployment-home, copy the "conf" folder to IGS_HOME/deployment-home.

5. From $RIB_HOME/application-assembly-home/rib-<app>, copy all rib-<app>-adapters.xml to IGS_HOME/ application-assembly-home/rib-<app>.

6. Go to $IGS_HOME/integration-bus-gateway-services/conf. Edit the IgsConfig.properties as follows.

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

7. Go to rib-home/tools-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.

All of the items in Step 7 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. If the igs-service.ear must be undeployed, run therib-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.

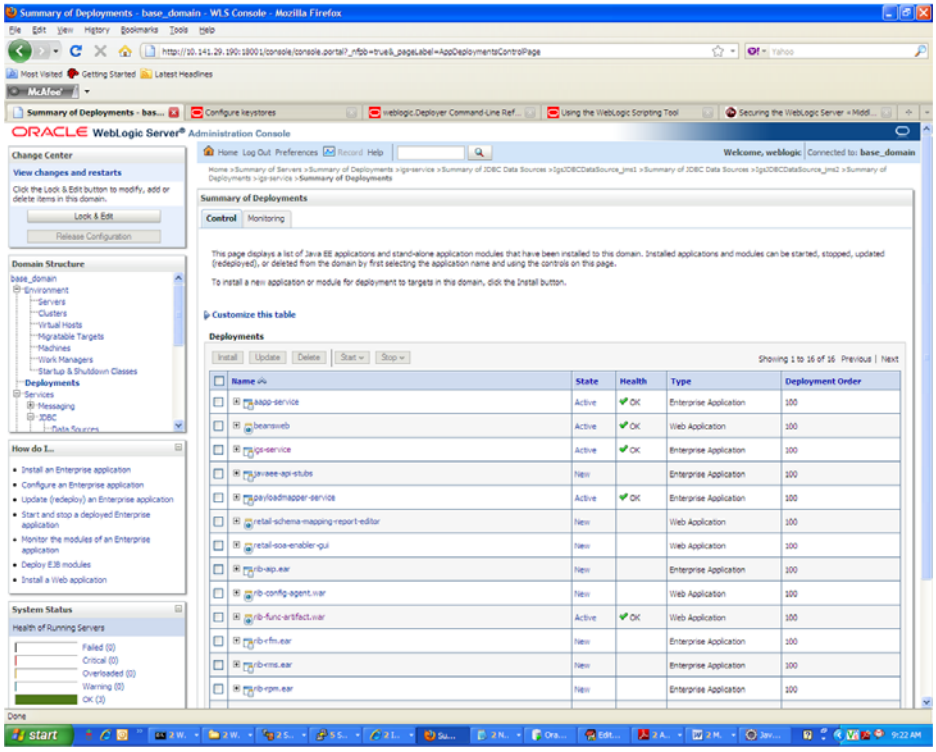
Server-side Setup for User Name and Password Authentication

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.

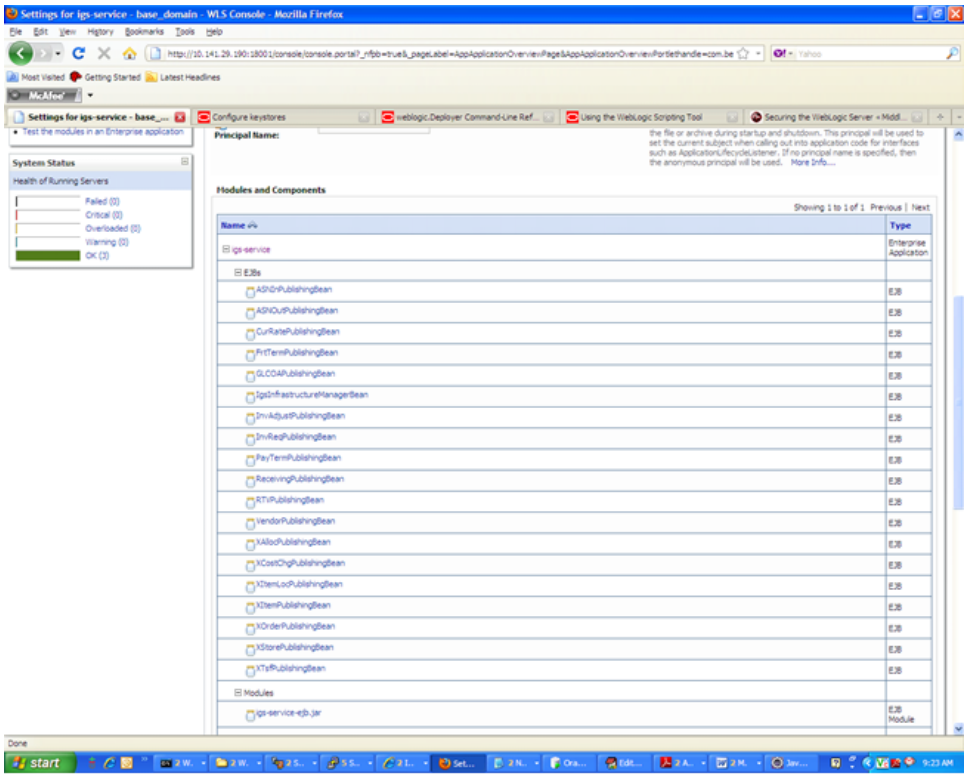
Attach Policy File to the Web Service

The usernametoke.xml contains the policy used by the WebsService 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.

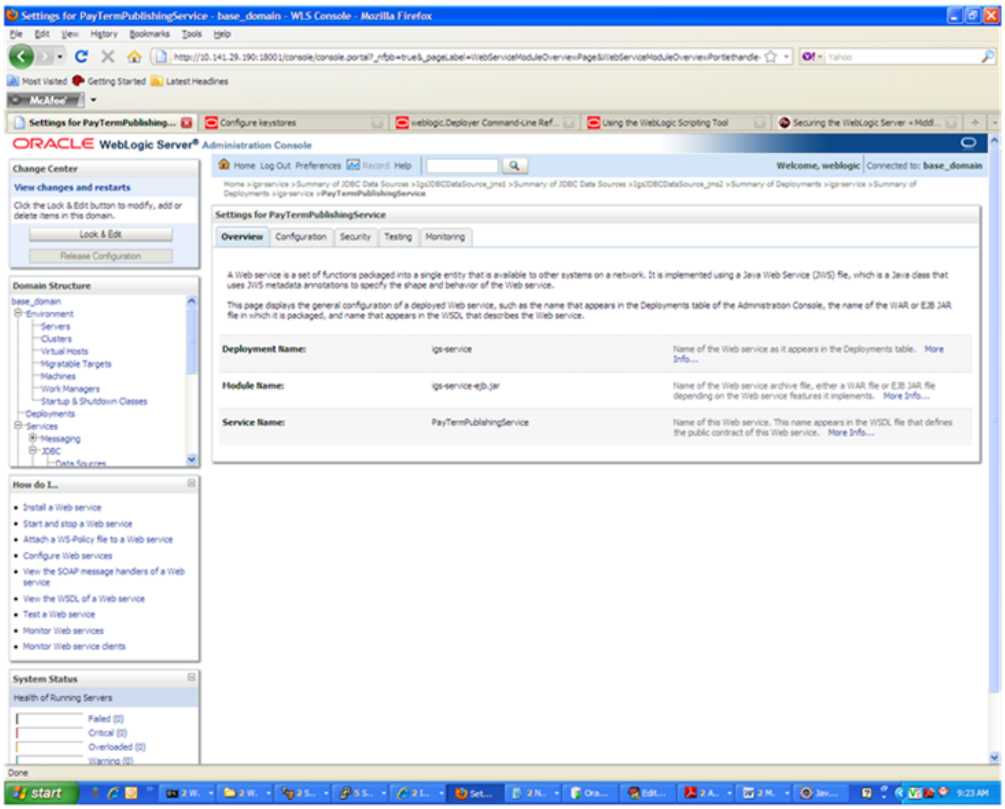
1. In the Summary of Deployments screen, click the application. In the illustration below, the application is igs-service.



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



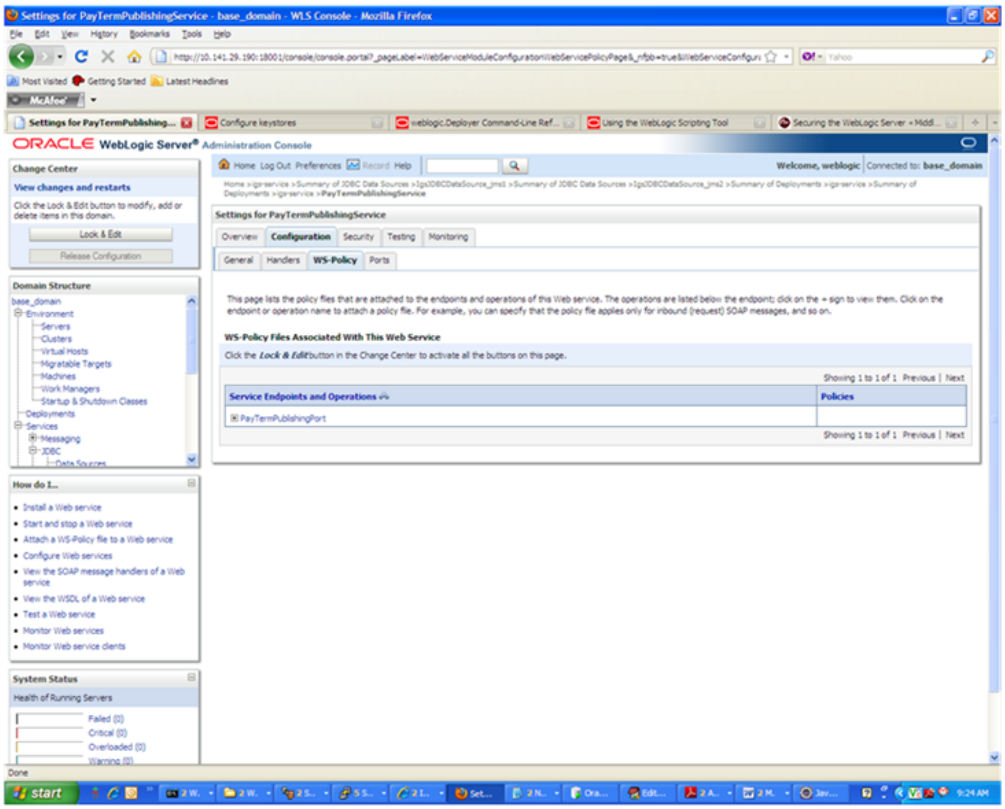
3. In the Web service list, click the service for which you want to enable security. The following screen is displayed to provide an overview of the Web service.



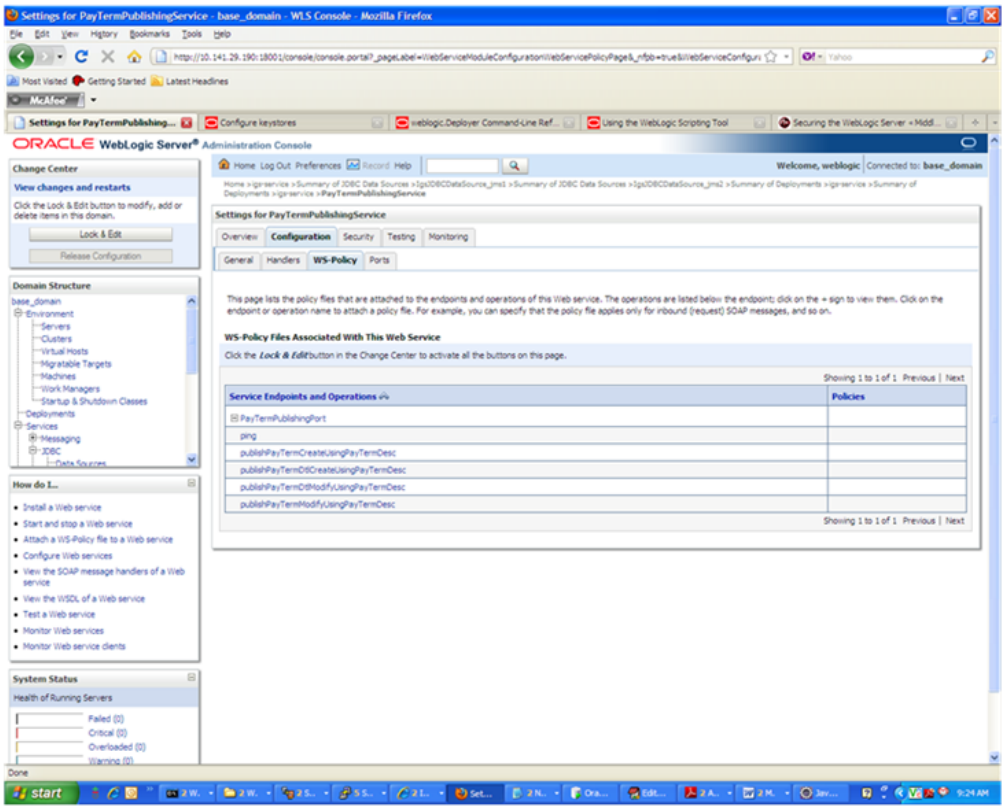
4. On this overview screen, click the Configuration tab. Click the WS-Policy tab. The Web service port is shown under Service Endpoints and Operations.

Note: Clicking the WS-Policy tab may result in the following WLS exception message: "WS-Policy files associated with this web service."

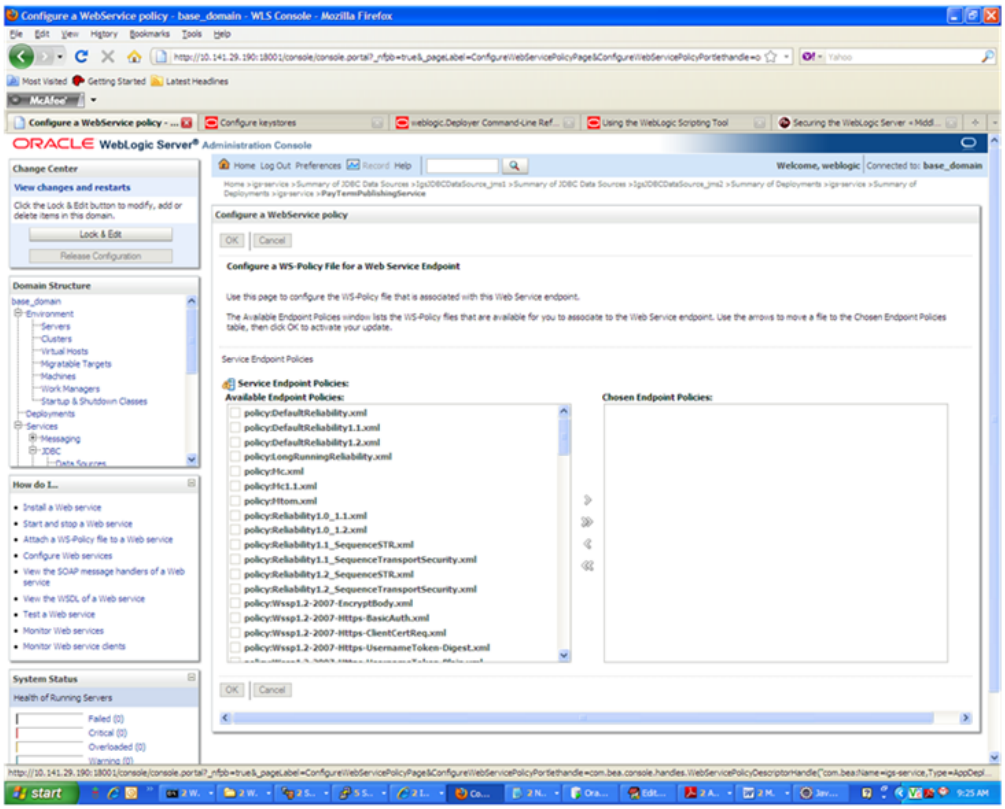
When WLS issues this exception, there is no way to secure IGS unless WLS is reinstalled. If this exception is encountered, you must reinstall WLS.



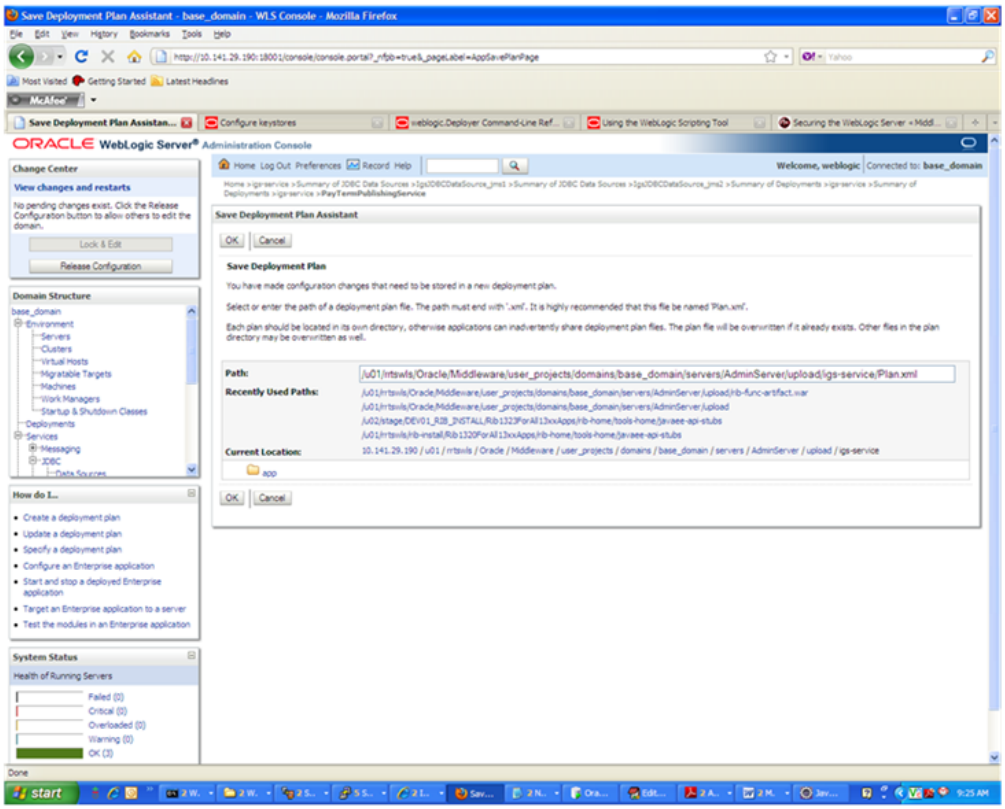
5. Click the plus sign next to the port name. The Web service operations are displayed.



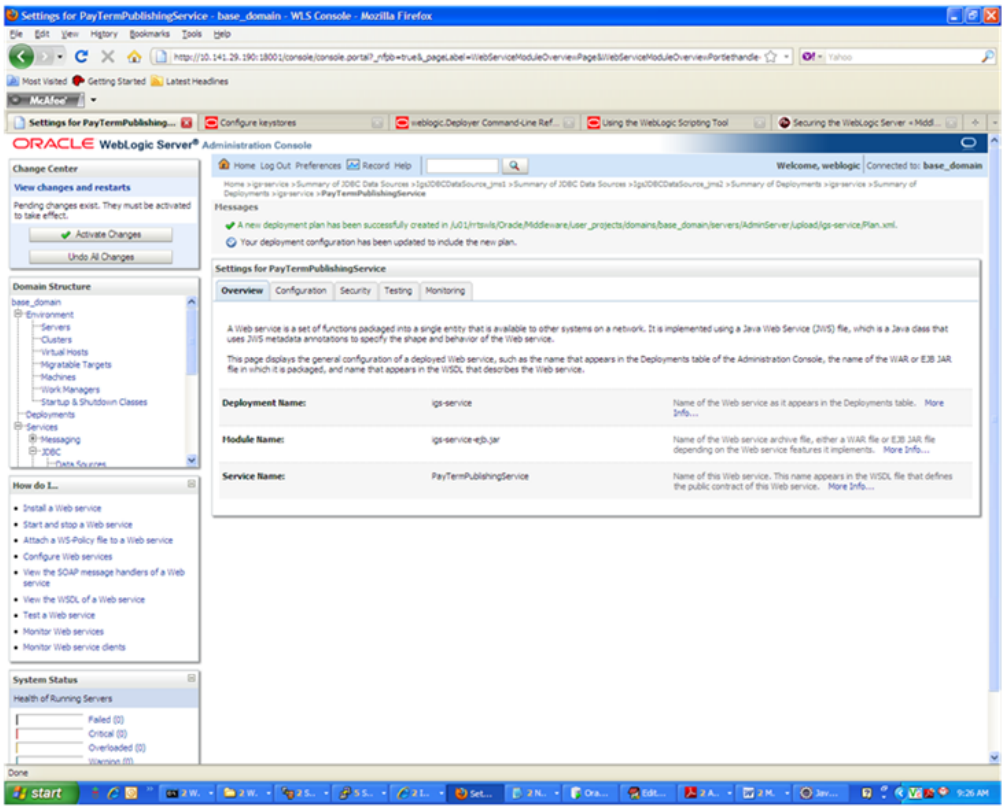
6. You can secure all the Web service operations at once or select only the operations you want to secure. Click the name of the port. On the Configure a Web Service policy screen, you can attach the policy file to the Web service.



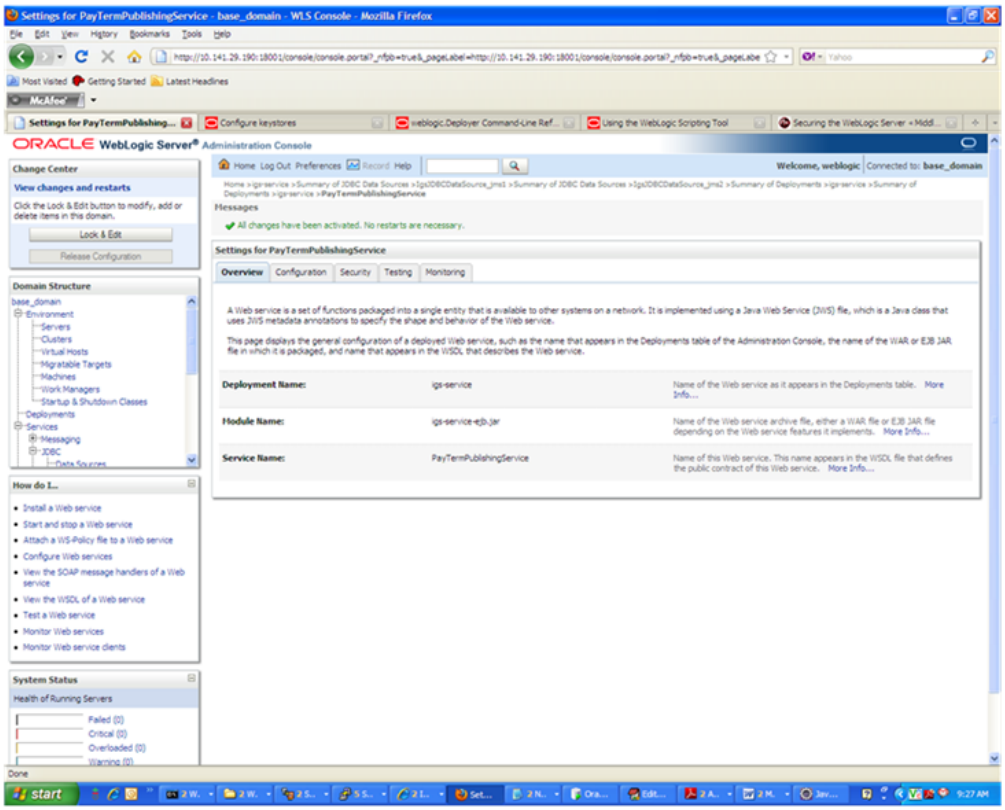
7. From the Available Endpoint Policies list, select policy:usernametoken.xml. Click the right arrow to move it to the drop down list below Chosen Endpoint Policies. Click **OK**. The Save Deployment Plan Assistant screen is displayed.



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



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



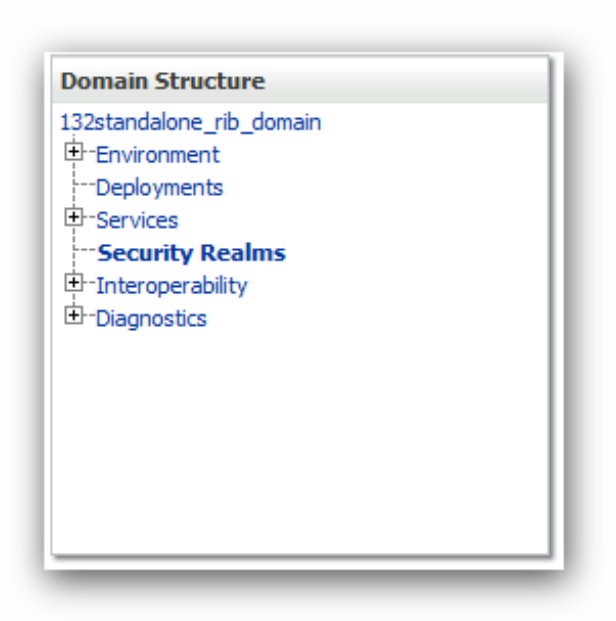
10. Under the Testing tab, on the Web Service page, click the WSDL to view the details of the policy just added to the Web service. The WSDL contains information similar to the following.

```
<?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>
```

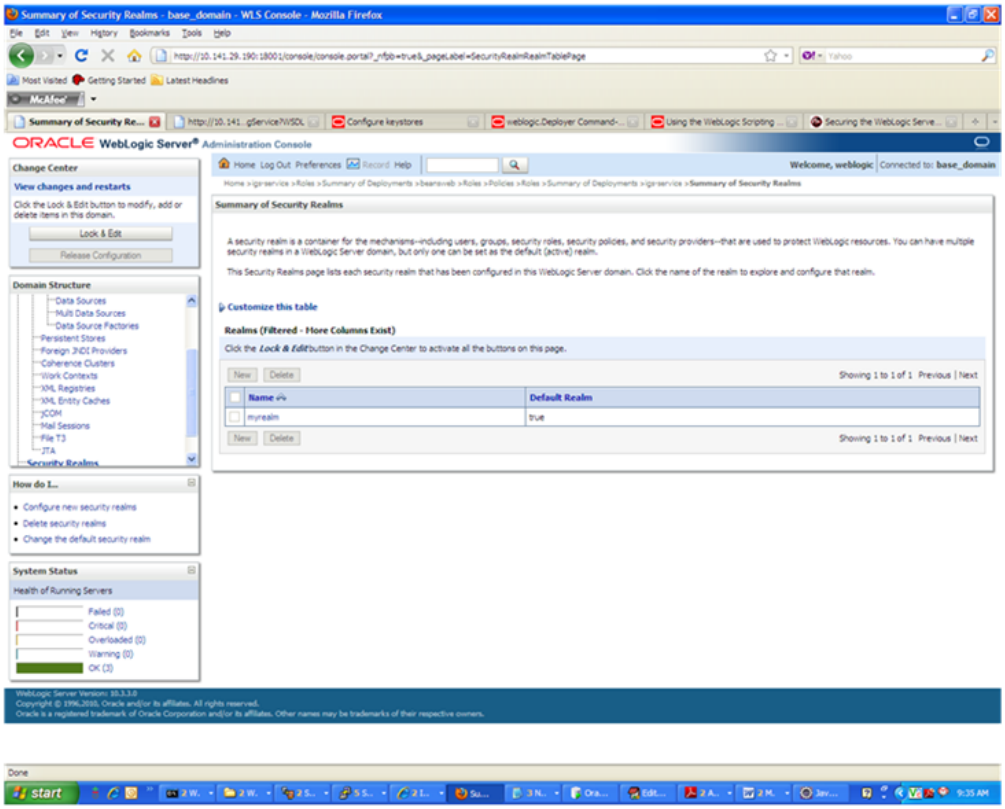
Create Roles and Users

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.

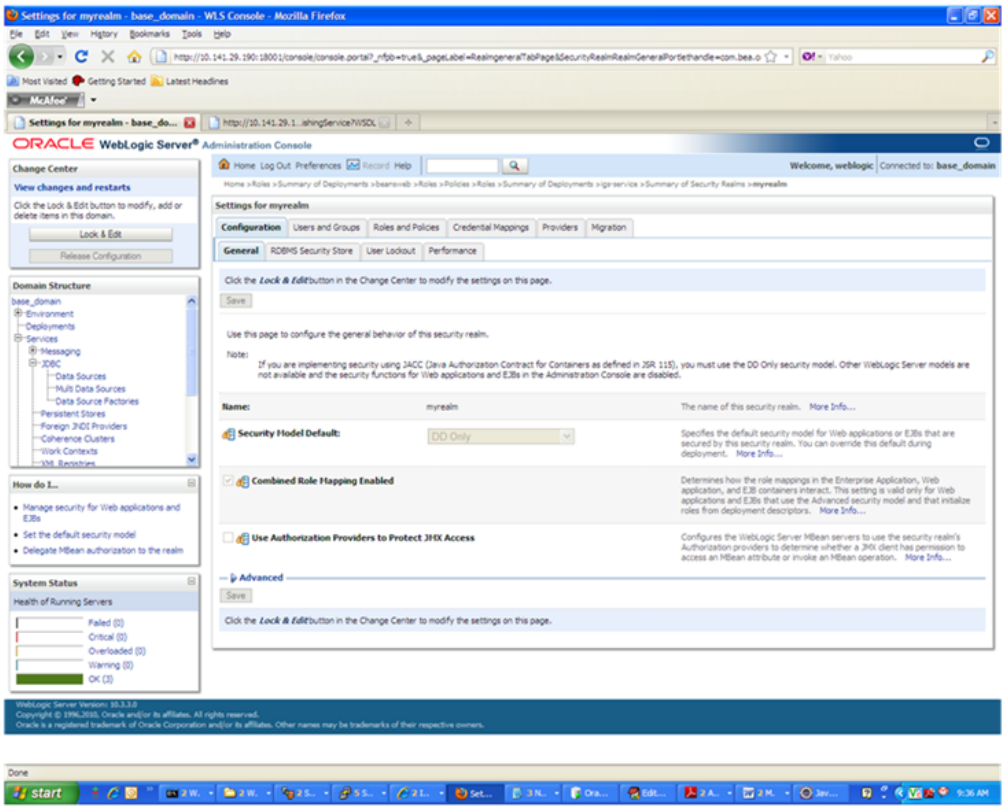
1. In the Domain Structure window of the Oracle WebLogic Services Administration Console, click the Security Realms link.



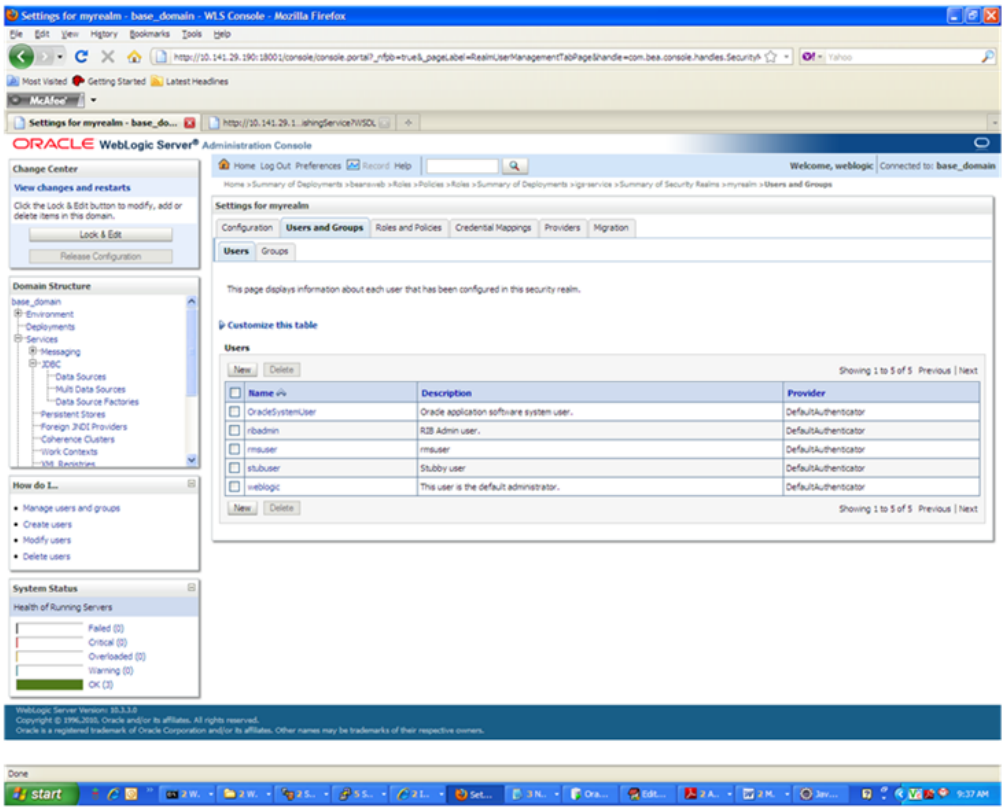
2. The Summary of Security Realms screen is displayed, including the name of the default realm.



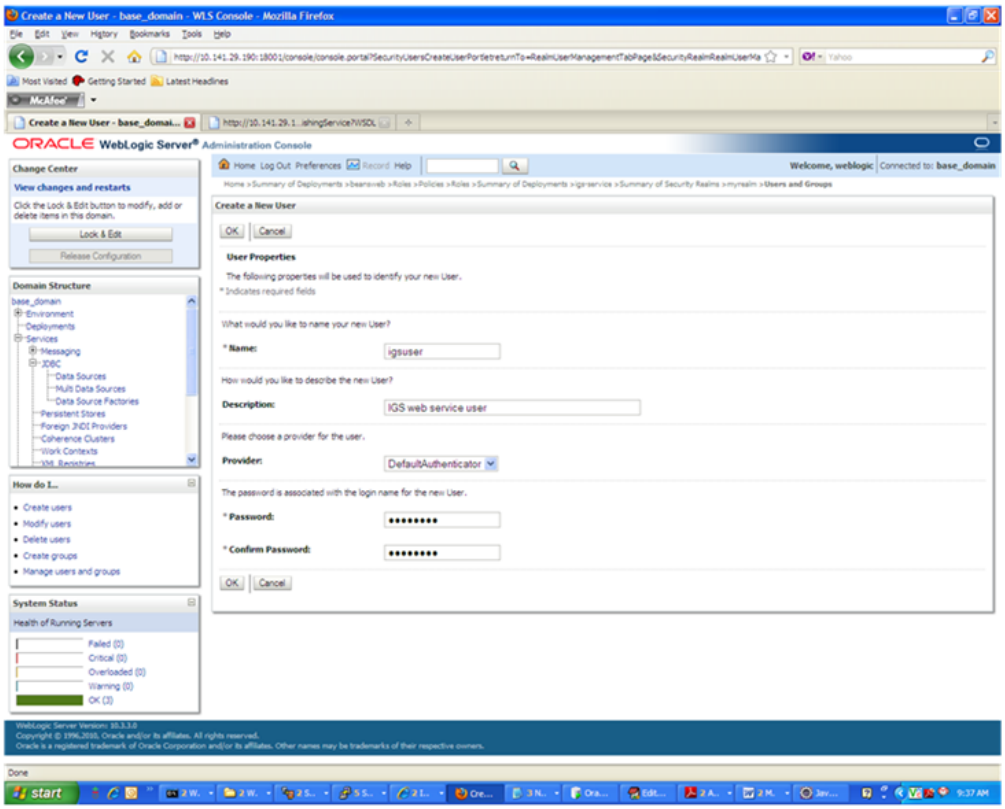
3. Click the name of the default realm. The settings for the realm are displayed.



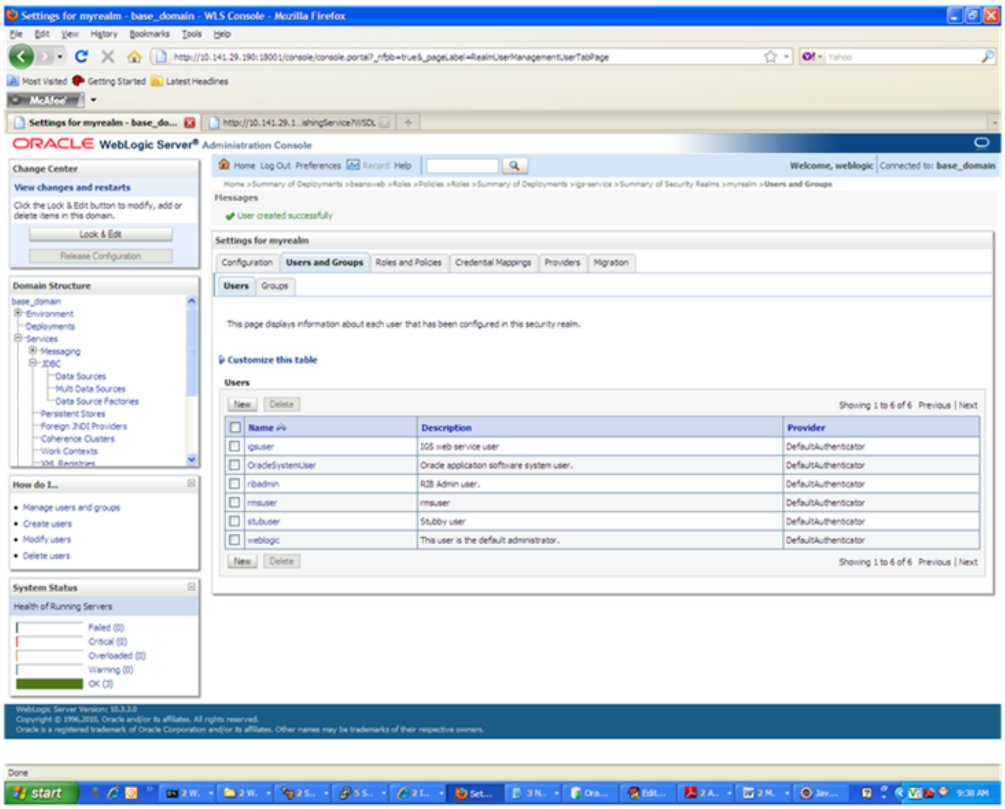
4. On the Settings screen, click the Users and Groups tab.



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.

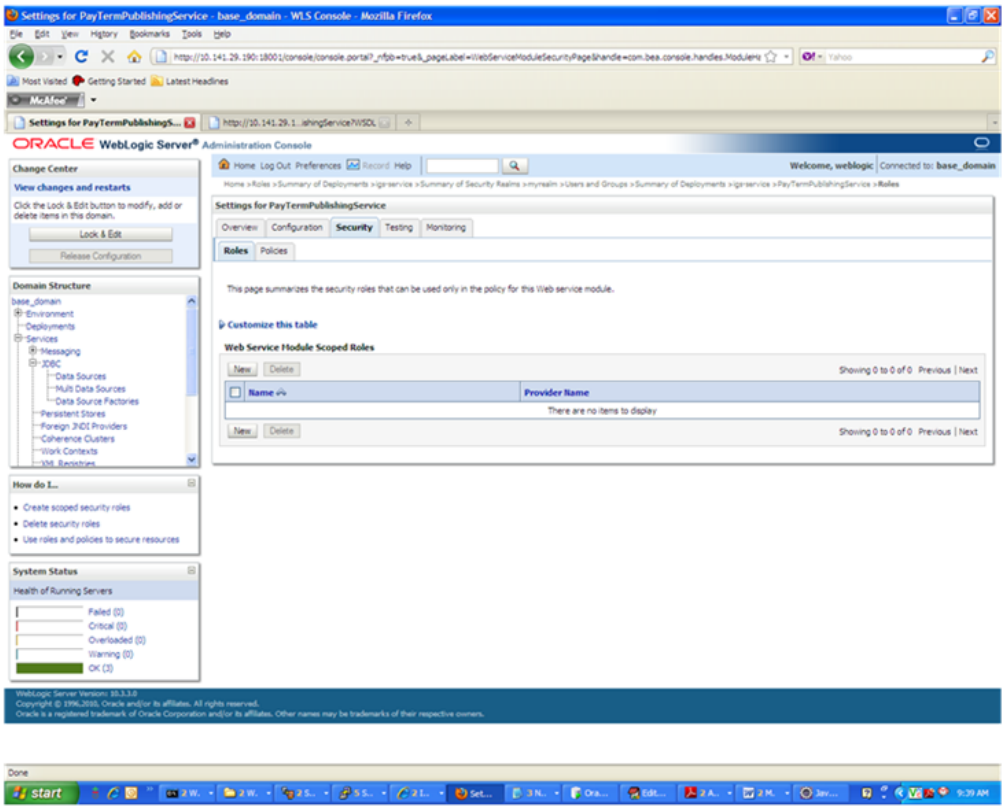


6. In the Create a New User screen, enter a user name and password. Leave the default value for Provider. Click **OK** to save the information. The new user is added to the list of users.

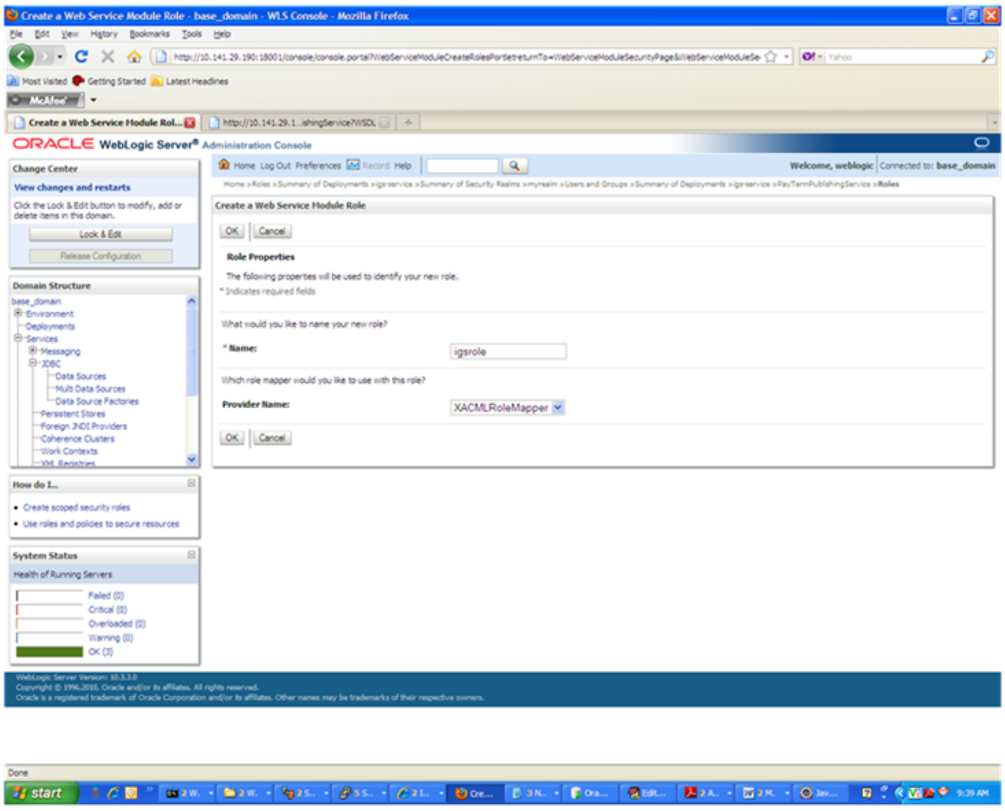


Note: You can add roles from the Roles and Policies tab of the security realm or through the Security tab of the Web service. The following instructions are for creating a role through the Security tab of the Web service.

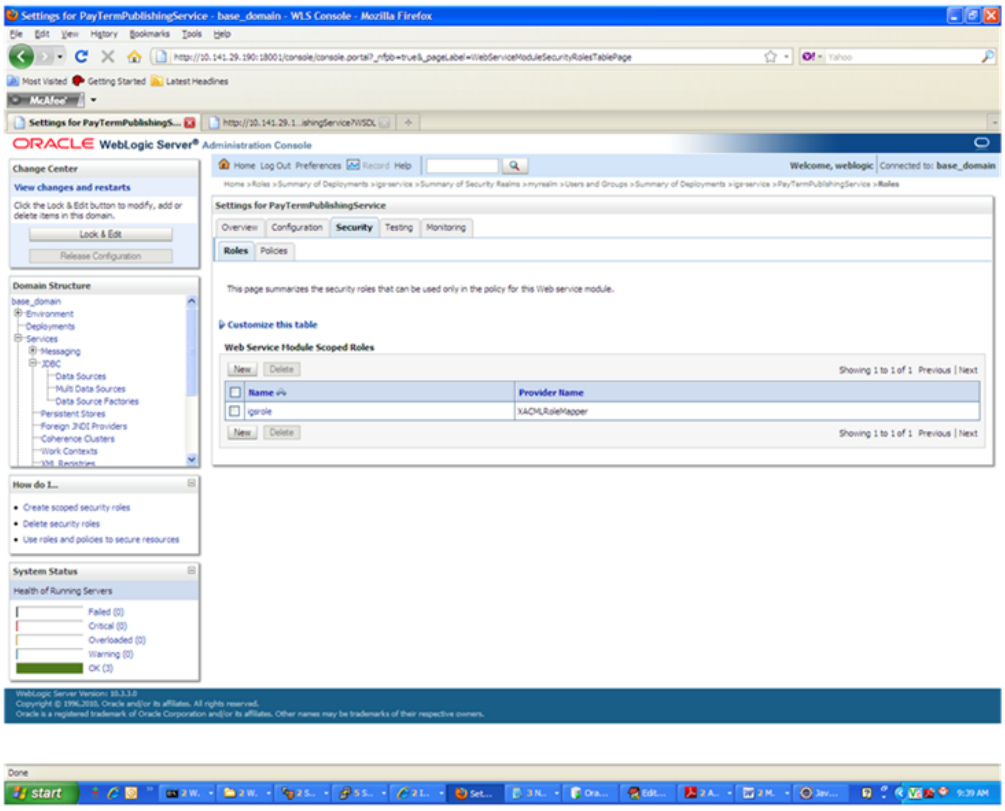
7. Navigate to the Security tab of the Web service. Click the Roles tab.



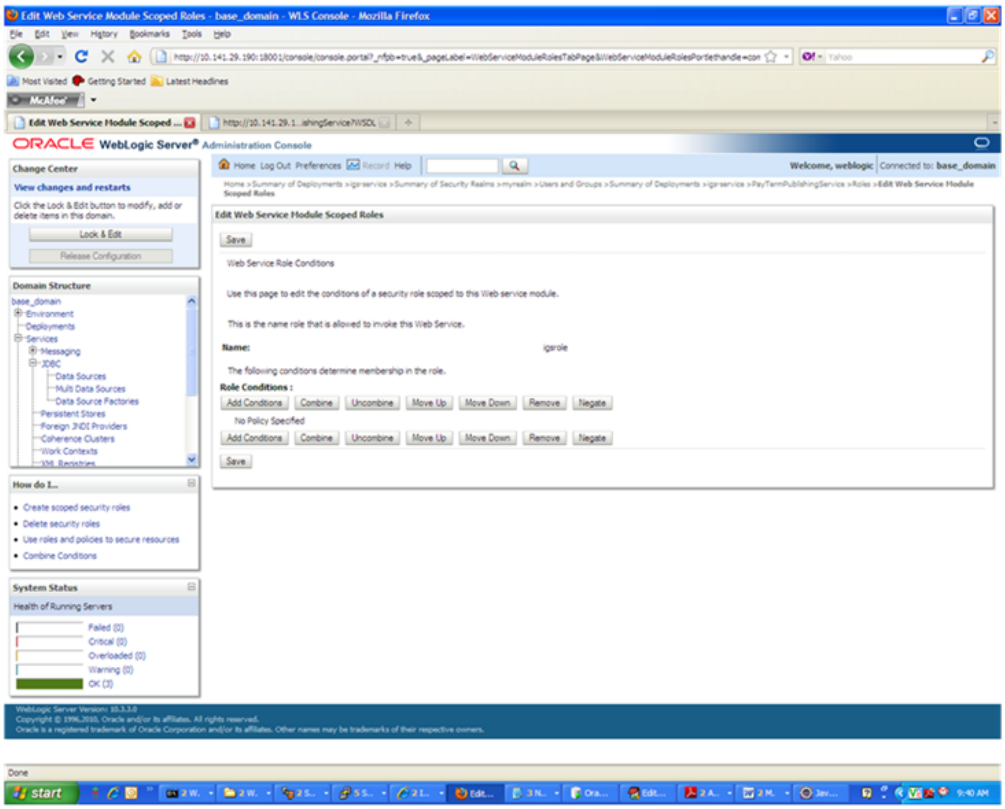
8. In the Roles tab, click **New**. The Create a Web Serivce Module role screen is displayed.



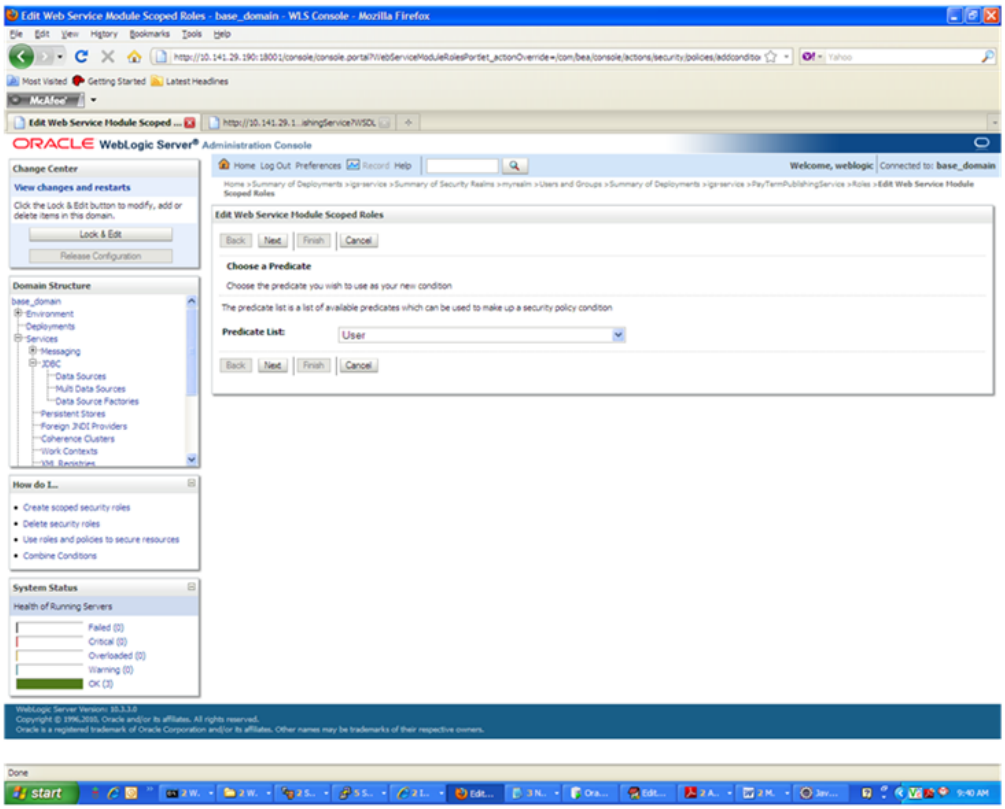
9. In the Create a Web Service Module Role screen, enter the role name in the Name field (for example, rmsrole). Leave the default value in the Provider Name field. Click **OK**. The new role is displayed in the Roles tab of the Web service.



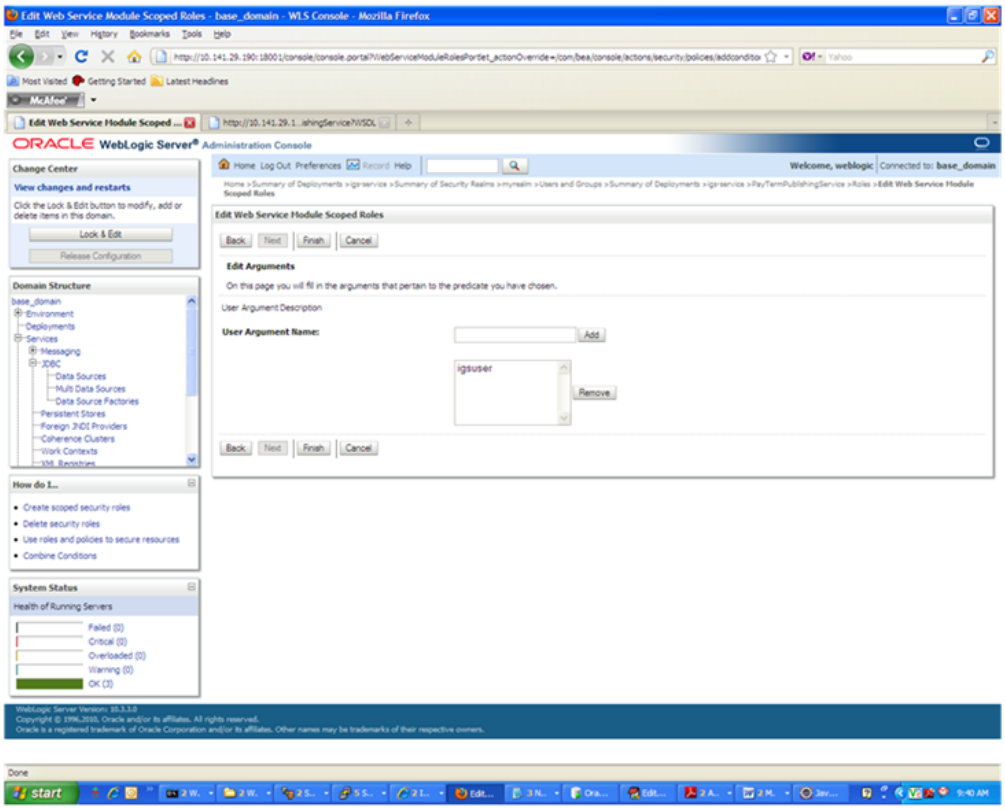
10. To add the user to the role, click the name of the new role in the Roles tab. The Edit Web Service Module Scoped Roles screen is displayed.



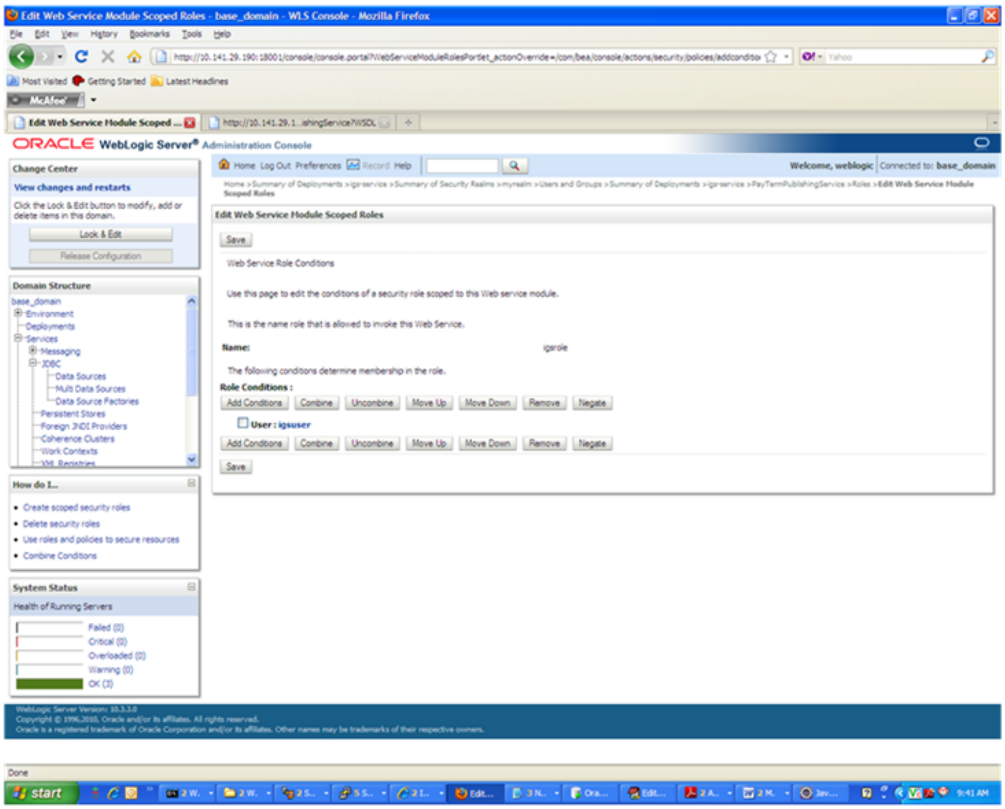
11. In the Edit Web Service Module Scoped Roles screen, click **Add Conditions**. The "Choose a Predicate" option is displayed.



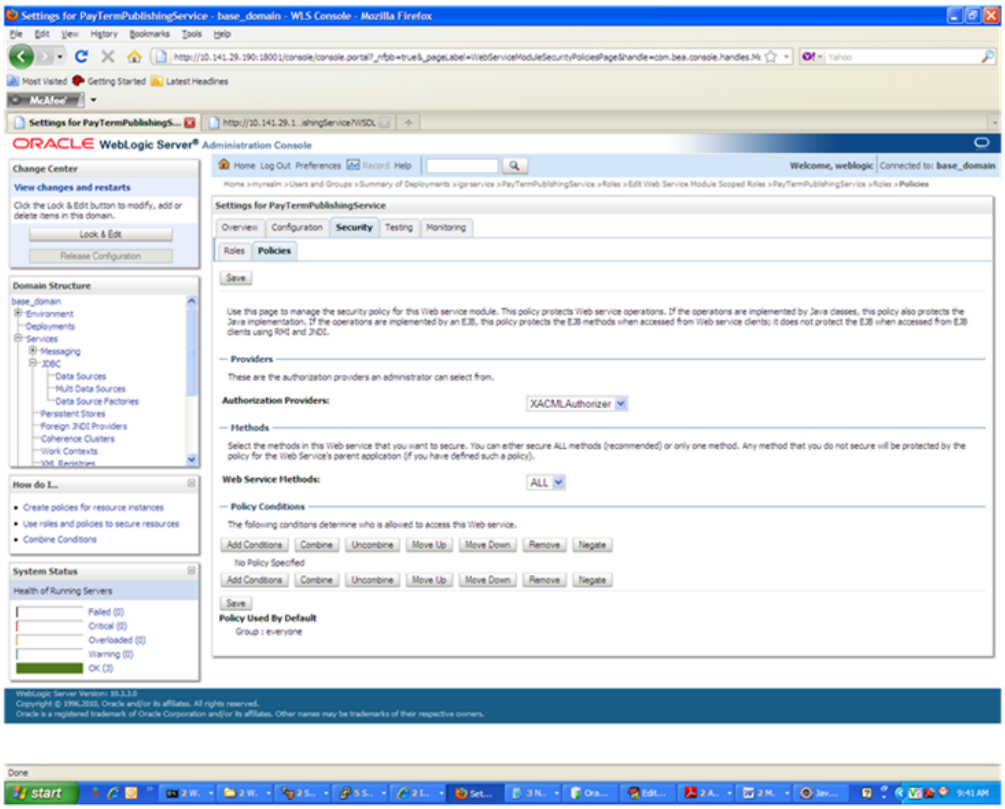
12. From the Predicate List, select User. Click **Next**. The Edit Arguments argument is displayed.



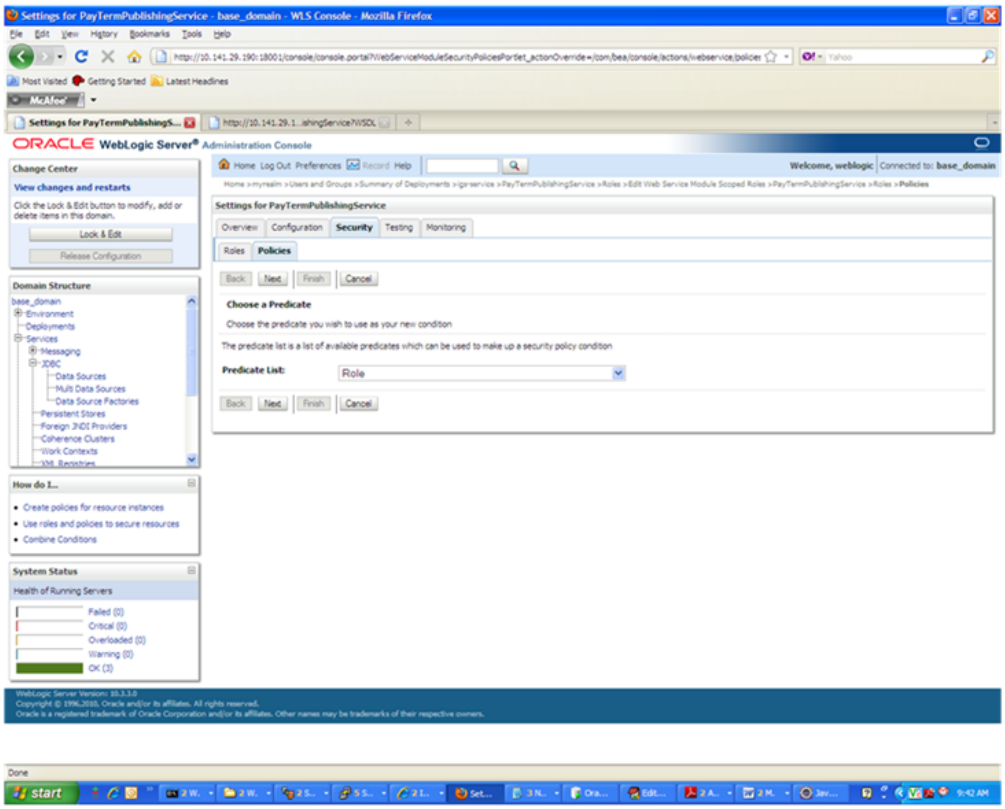
13. In the User Argument Name field, enter the user name created in the security realm. Click **Add**. The name will move down to the box below the Add button. Click **Finish**. The following screen is displayed.



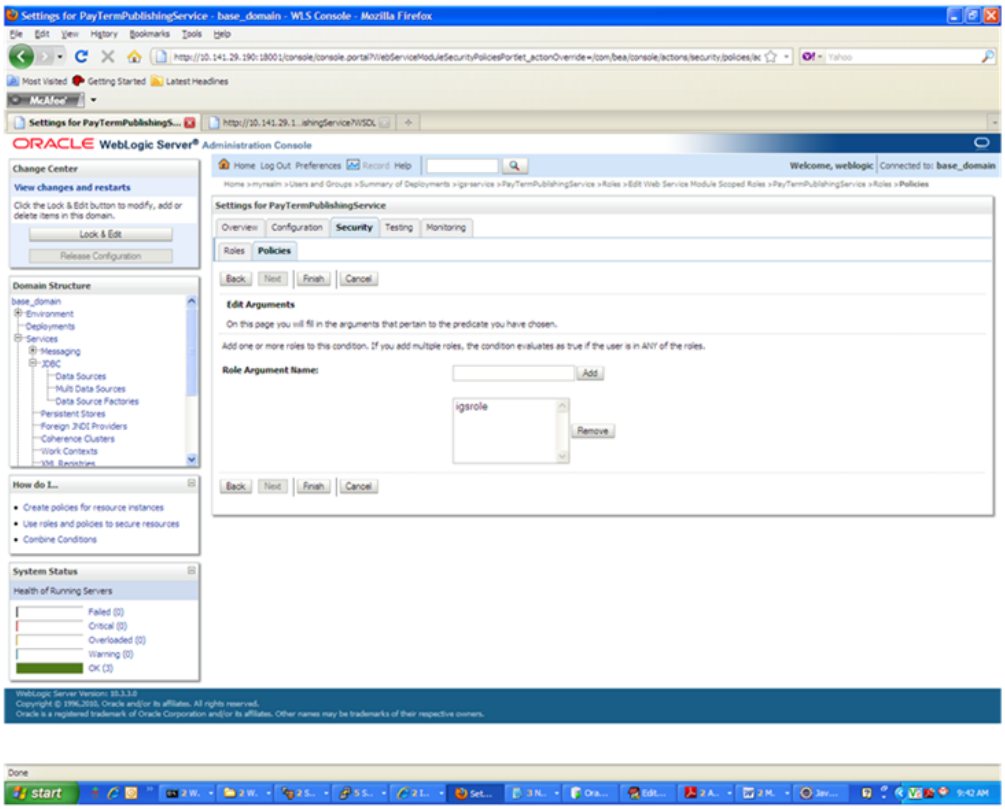
14. Click **Save**. The same screen is displayed with this message near the top: "Changes saved successfully."
15. Return the Security tab of the Web service and click the Policies tab.



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

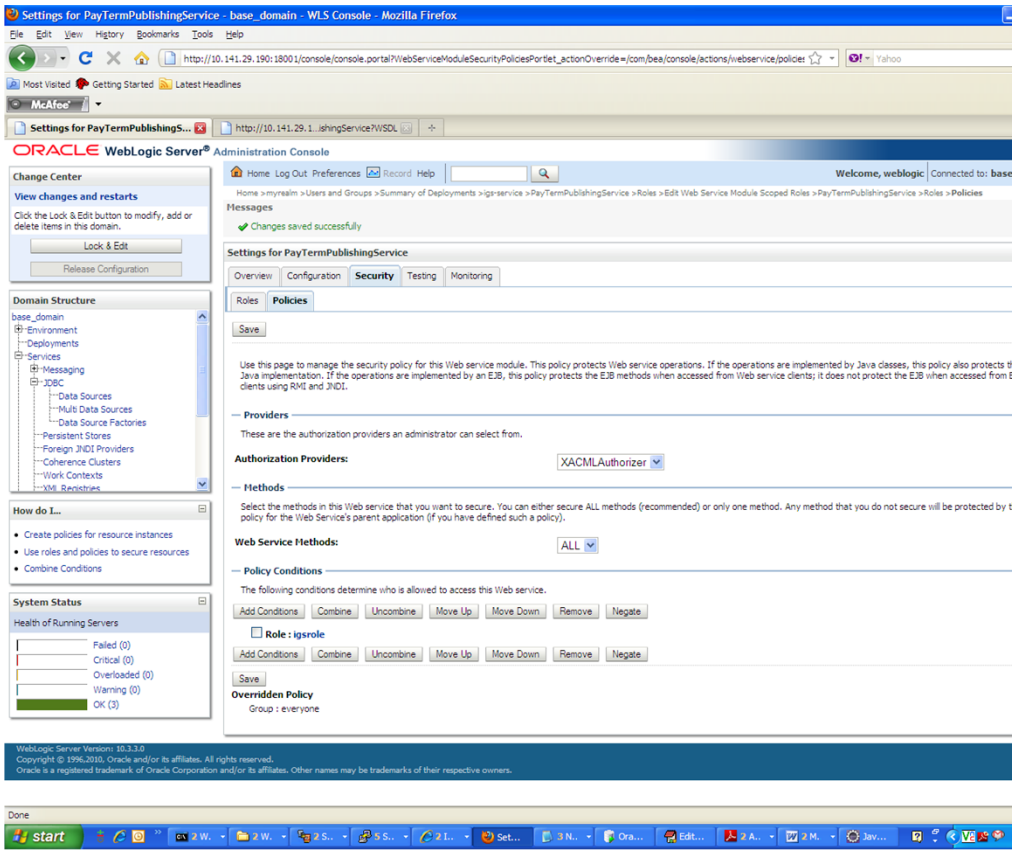


17. From the Predicate List, select Role. Click **Next**. The Edit Arguments option is displayed.



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."



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://10.141.29.190:18030/PayTermPublishingBean/PayTermPublishingService?WSD
L");

                    //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("rmsuser","rmsuser1");
                    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 $WL_HOME/Middleware/wlserver_10.3/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 $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
   utils.CertGen -certfile ClientCert -keyfile ClientKey -keyfilepass ClientKey
   -cn rmsuser
   ```

   The above command generates the following files.

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

   The user name is rmsuser. 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 $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
   utils.CertGen -certfile ServerCert -keyfile ServerKey -keyfilepass ServerKey
   -cn rmsuser
   ```

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

The user name rmsuser. 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 $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile ClientCert.der -keyfile ClientKey.der
-keyfilepass ClientKey -keystore ClientIdentity.jks -storepass ClientKey -alias
identity - keypass ClientKey

java -classpath $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile ServerCert.der -keyfile ServerKey.der
-keyfilepass ServerKey -keystore ServerIdentity.jks -storepass ServerKey -alias
identity - keypass ServerKey
```

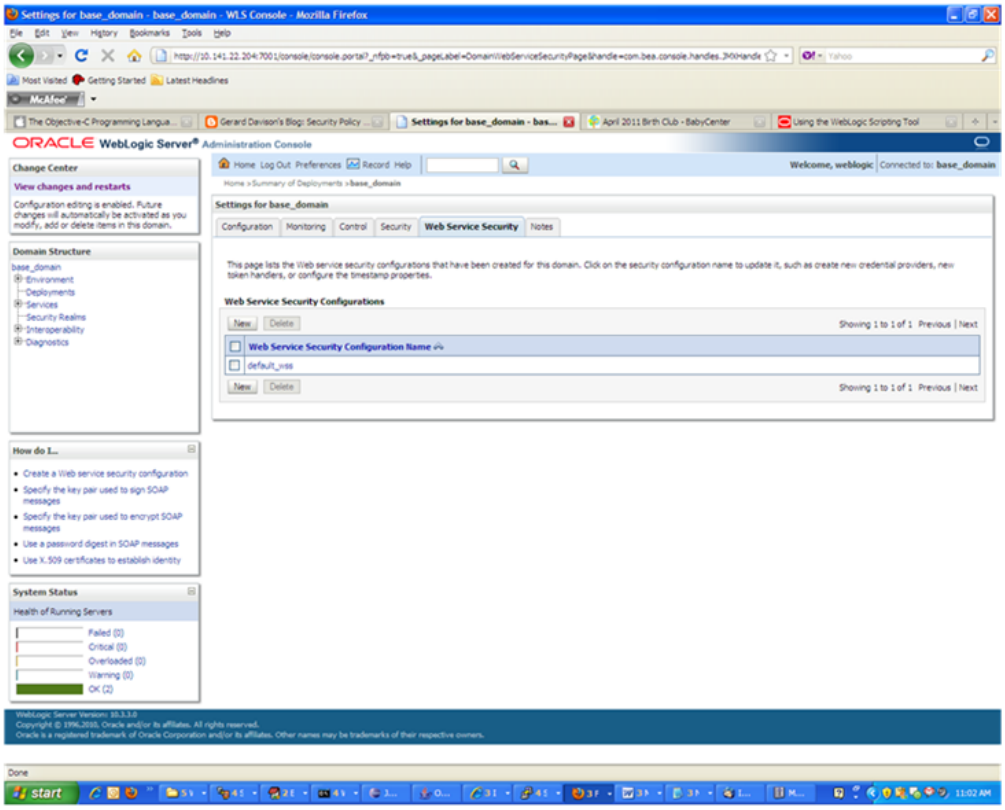
7. Using the script in Appendix: configWss.py, configure the WebLogic server to use the key. Copy the script and save it in the location from which it will run.

```
Java -classpath $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
weblogic.WLST configWss.py <weblogicuser> <weblogicpassword> <weblogichost>
<weblogic admin port> ServerIdentity.jks ServerKey identity ServerKey
```

For example:

```
Java -classpath $WL_HOME/Middleware/wlserver_10.3/server/lib/weblogic.jar
weblogic.WLST configWss.py weblogic weblogic1 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.



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 aWweb 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://10.141.29.190:18030/PayTermPublishingBean/PayTermPublishingService?WSD
L");

//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 ClientUNTCredentialProvider("rmsuser",
"rmsuser1");
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();
}
}
}
```

7

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. The RIB Installer gathers the appropriate values from the user, constructs the file, and invokes the appropriate tools.

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/10.2.0/db_
1</jms-server-home>
<jms-url>jdbc:oracle:thin:@linux1:1521:ora10g</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/deployement-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 in RIB Deployment Configuration File Editor

The RIB Deployment Configuration File Editor is an application used to configure the rib-deployment-env-info.xml file, following installation. It provides a user interface for adding, removing, and rearranging the elements of the RIB configuration.

This tool has fields for entering user names and passwords required for connecting to application server and databases. Values entered in the password field in the tool are displayed as a series of asterisks (one for each character). The values entered in this field are stored in the secured wallet file in the rib-home/deployement-home/conf/security/ directory.

For information about the RIB Deployment Configuration File Editor, see the section, "RIB Deployment Configuration File Editor," in the "Application Builder" chapter in the *Oracle Retail Integration Bus Operations Guide*.

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. This realm defines a group called ribadminrole. By default, RIB creates a user that belongs to the ribadminrole 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 ribadminrole 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 aunthentication. The user name must be the same as the one created by RIB in ribadminrole. When the credentials are verified, the RIB administration GUI home page is displayed.

RIB System Administrators Security Domain

The RIB System Administrators primarily focus on managing access the 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.

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. After deployment, the system administrator can create roles and users in WebLogic with access to the tool.

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 montoring 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 the 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 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 11g server:

http://download.oracle.com/docs/cd/E15523_01/web.1111/e13707/ssl.htm#SECMG384

Deployment of RIB applications over SSL protocol currently is not supported. However, after the applications are deployed they can run on the SSL protocol. Therefore, administrators can disable the non-SSL ports after RIB applications are deployed, but they must keep them open during the deployment process. The configuration file rib-deployment-env-info.xml file must contain non-SSL port numbers of WebLogic instances where RIB applications will be deployed.

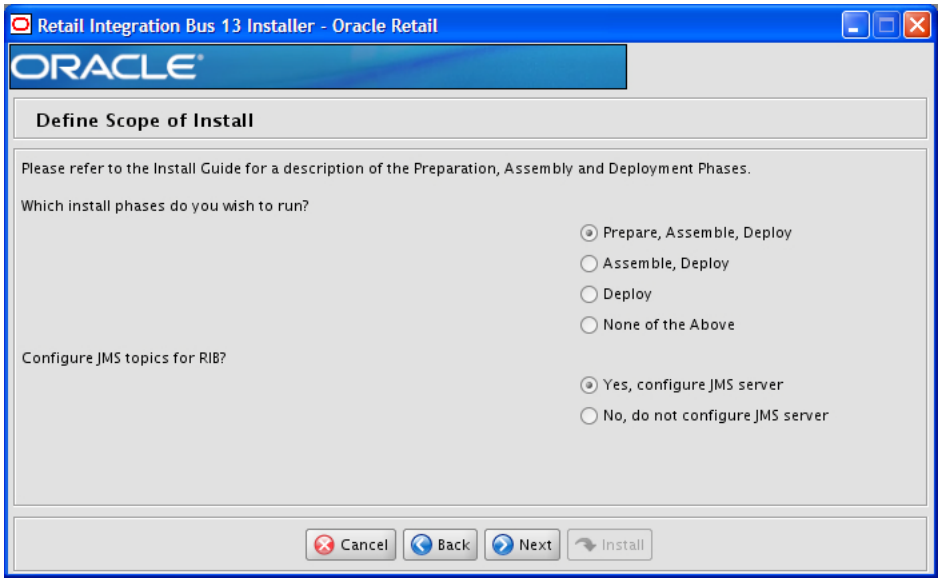
Below are the steps for running RIB in SSL environment.

1. Configure SSL in the WebLogic server. (See WebLogic documentation for detailed steps.)
2. Keep the non-SSL ports of the WebLogic server instances open for RIB deployment.. Verify that the non-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.
3. Make sure that the rib-deployment-env-info.xml file has protocol specified as http and port numbers are http port numbers for WebLogic server instances.
4. Deploy the RIB applications.
5. 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.

A

Appendix: RIB Application Installer Screens

You will need the following details about your environment for the installer to successfully deploy the RIB applications. Depending on the options you select, you may not see some screens.



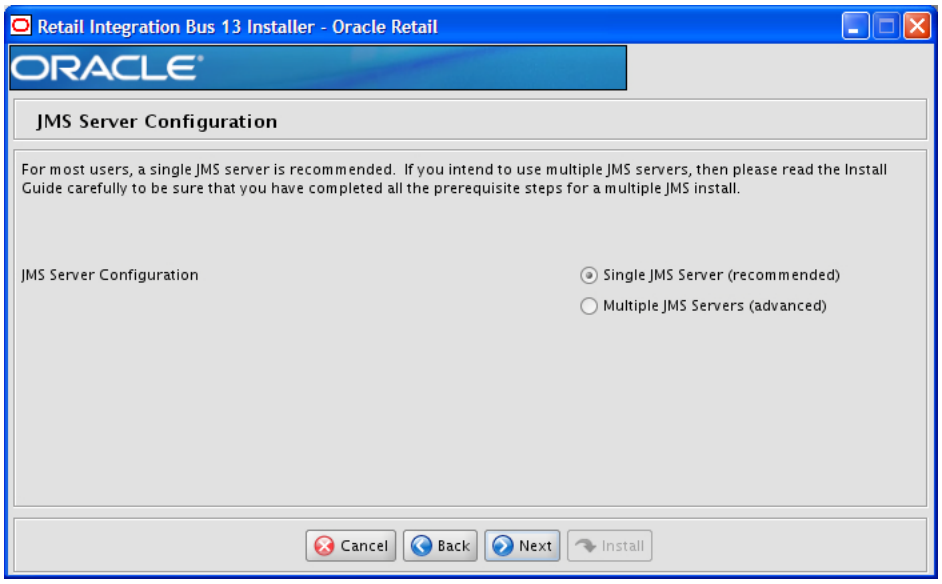
Field Title	Which installation phases do you wish to run?
Field Description	Used by the installer's build.xml to determine which phases to run during the installation. Each installation phase will run a different command-line tool. Preparation Phase: check-version-and-unpack.sh Assembly Phase: rib-app-compiler.sh Deployment Phase: rib-app-deployer.sh -deploy-rib-func-artifact-war and/or rib-app-deployer.sh -deploy-rib-app-ear rib-<app>

Field Title	Configure JMS topics for RIB?
Field Description	Used by the installer's build.xml to determine whether to configure the JMS topics. Will run the command-line tool: rib-app-deployer.sh -prepare-jms

Screen: Provide Inputs to Installer?

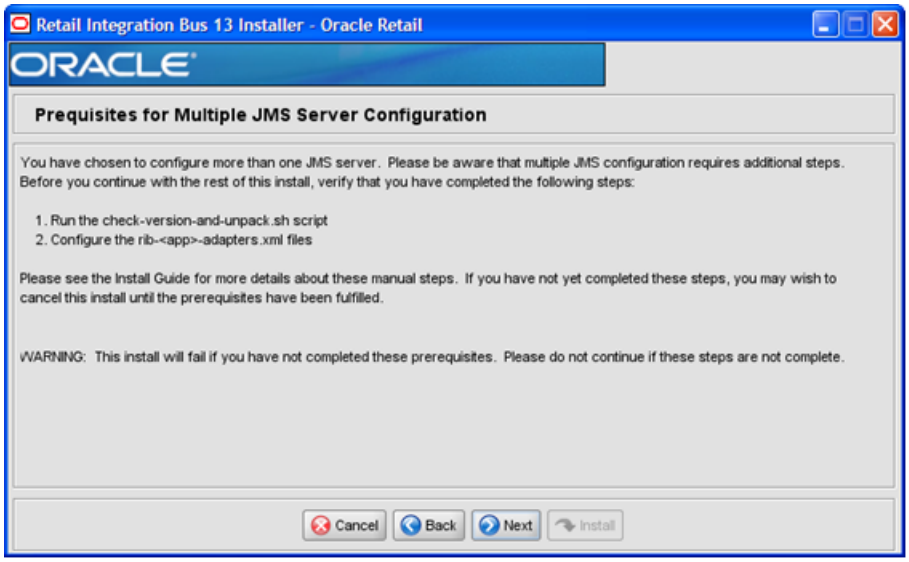
Field Title	Generate new config file?
Field Description	Used by the installer to determine whether to prompt user for inputs needed to generate the rib-deployment-env-info.xml file. Also used by the installer's build.xml to determine whether or not to actually generate the new file.

Screen: JMS Server Configuration

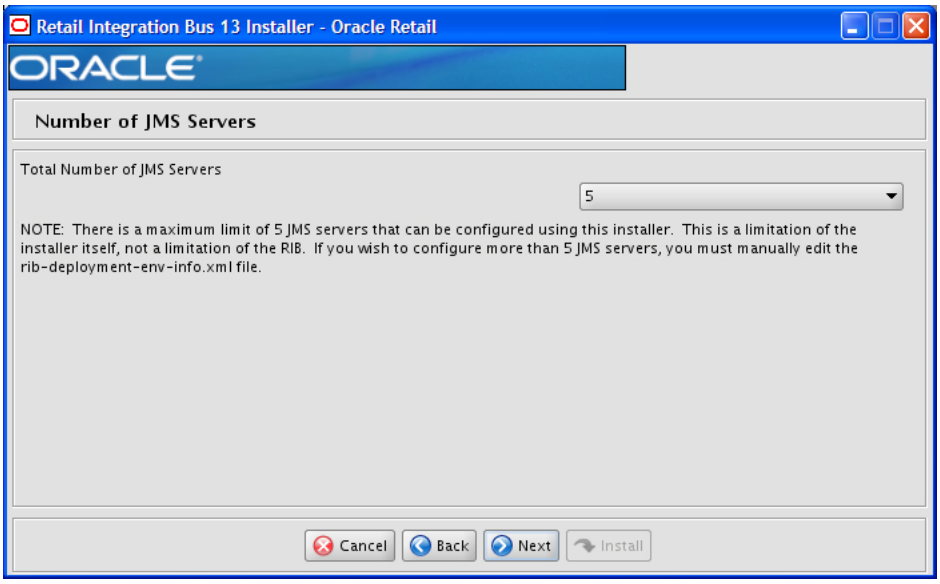


Field Title	JMS Server Configuration
Field Description	Used by the installer to determine how many sets of JMS server inputs should be collected from the user.

Screen: Prerequisites for Multiple JMS Server Configuration

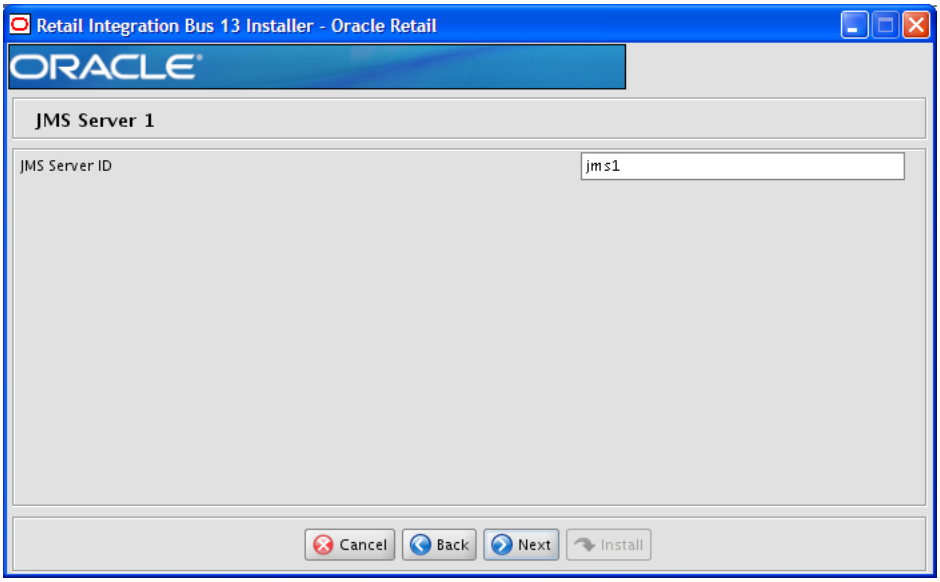


Screen: Number of JMS Servers



Field Title	Total Number of JMS Servers
Field Description	Used by the installer to determine how many sets of JMS server inputs should be collected from the user.
Example	5

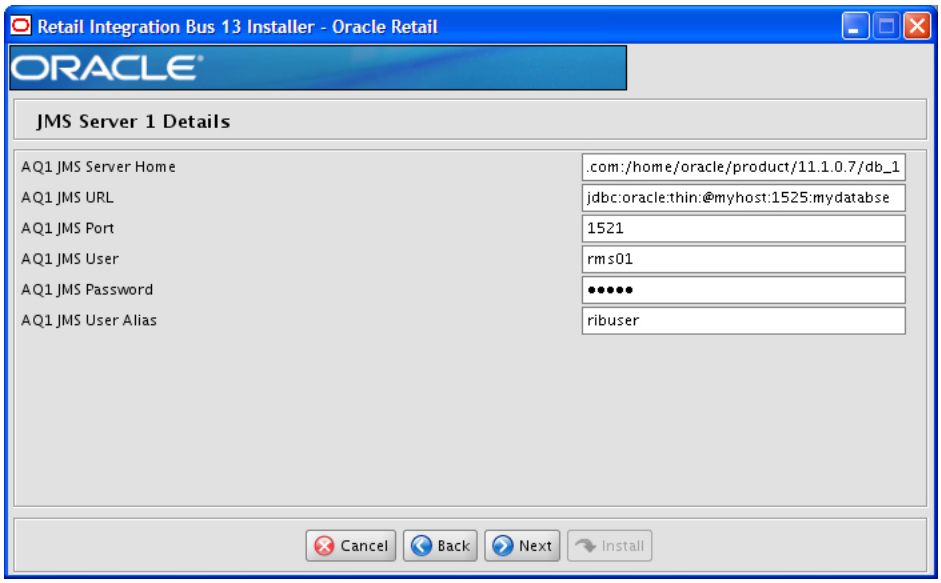
Screen: JMS Server 1



Note: The installer will request input for as many JMS servers as were chosen on the previous screen. There will be one input screen for each JMS server.

Field Title	JMS Server ID
Field Description	The name of the JMS server.
Destination	rib-deployment-env-info.xml
Example	jms1
Notes	

Screen: JMS Server 1 Details



Note: The installer will request input for as many JMS Servers as were chosen on the previous screen. There will be one input screen for each JMS server.

Field Title	AQ1 JMS Server Home
Field Description	The AQ1 JMS server home
Destination	rib-deployment-env-info.xml
Example	oracle@myhost:/u01/oradata

Field Title	AQ1 JMS URL
Field Description	The AQ1 JMS URL
Destination	rib-deployment-env-info.xml
Example	single instance thin client: jdbc:oracle:thin:@myhost:1521:mydb

Field Title	AQ1 JMS Port
Field Description	The AQ1 JMS port
Destination	rib-deployment-env-info.xml

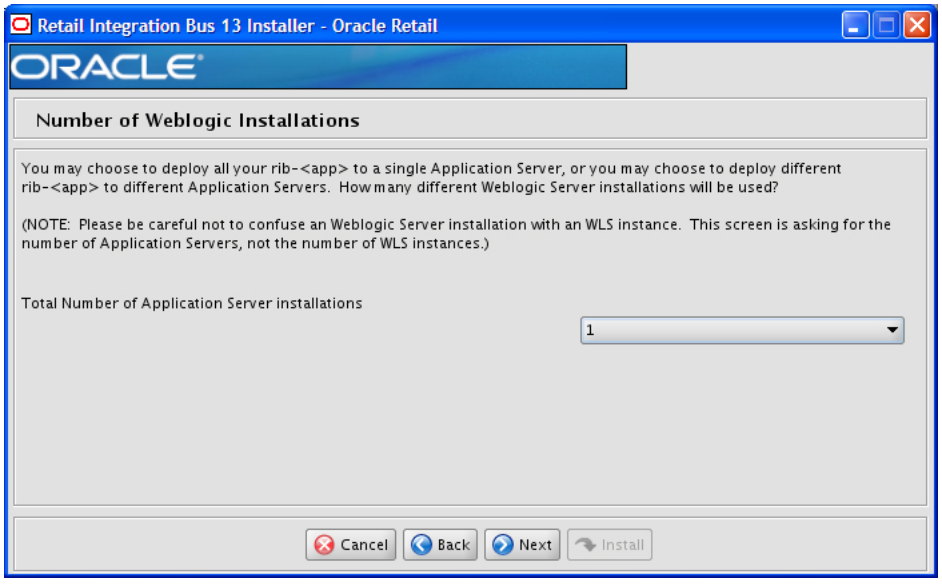
Field Title	AQ1 JMS Port
Example	1521

Field Title	AQ1 JMS User
Field Description	The AQ1 JMS user
Destination	rib-deployment-env-info.xml
Example	RIB_AQ

Field Title	AQ1 JMS Password
Field Description	The AQ1 JMS password.
Destination	rib-deployment-env-info.xml

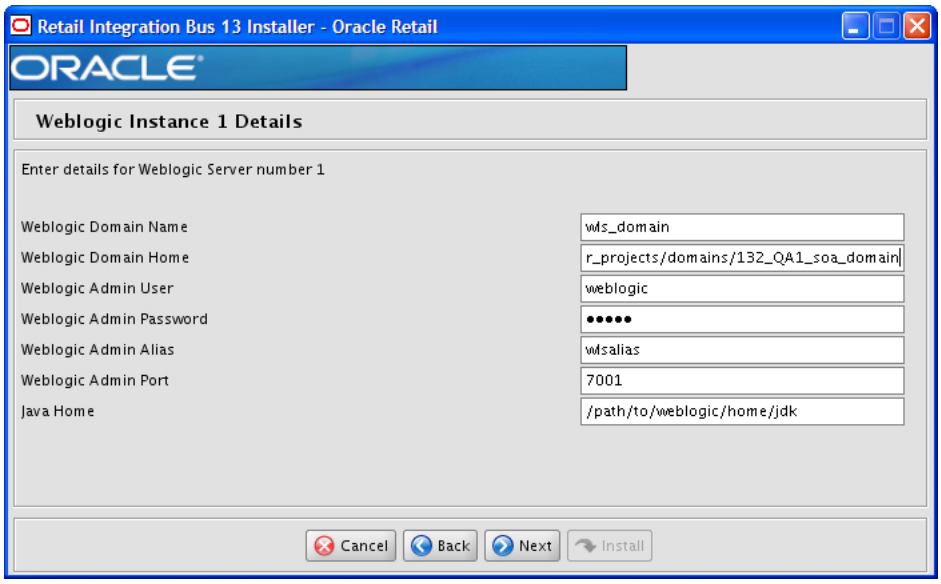
Field Title	AQ1 JMS Alias
Field Description	The alias is used by the application to access user names and passwords in the wallet file cwallet.sso.
Destination	rib-deployment-env-info.xml
Example	RIB_AQ_ALIAS
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

Screen: Number of WebLogic Installations



Field Title	Total Number of Oracle WebLogic Server Installations
Field Description	The number of different WebLogic Servers to which your rib-<app> applications will be installed. The installer uses this information to determine how many Oracle WebLogic servers it must request input for.
Example	1

Screen: WebLogic Instance <X> Details



Note: The installer will request input for as many Oracle WebLogic servers as were chosen on the previous screen. There will be one input screen for each Oracle WebLogic server.

Field Title	WebLogic Domain Name
Field Description	Your App Server's domain name.
Destination	rib-deployment-env-info.xml
Example	rib_domain

Field Title	WebLogic Domain Home
Field Description	The format should be: <user>@<host>:<WEBLOGIC_DOMAIN_HOME> where <user> is the user who owns the files in the ORACLE_HOME <host> is the name or IP address of the server where the App Server is installed <WEBLOGIC_DOMAIN_HOME> is the filesystem path to the installed domain.
Destination	rib-deployment-env-info.xml

Field Title	WebLogic Domain Home
Example	myuser@myhost:/path/to/weblogic/domain/home

Field Title	WebLogic Admin User
Field Description	The WebLogic admin user for this WebLogic instance.
Destination	rib-deployment-env-info.xml
Example	weblogic

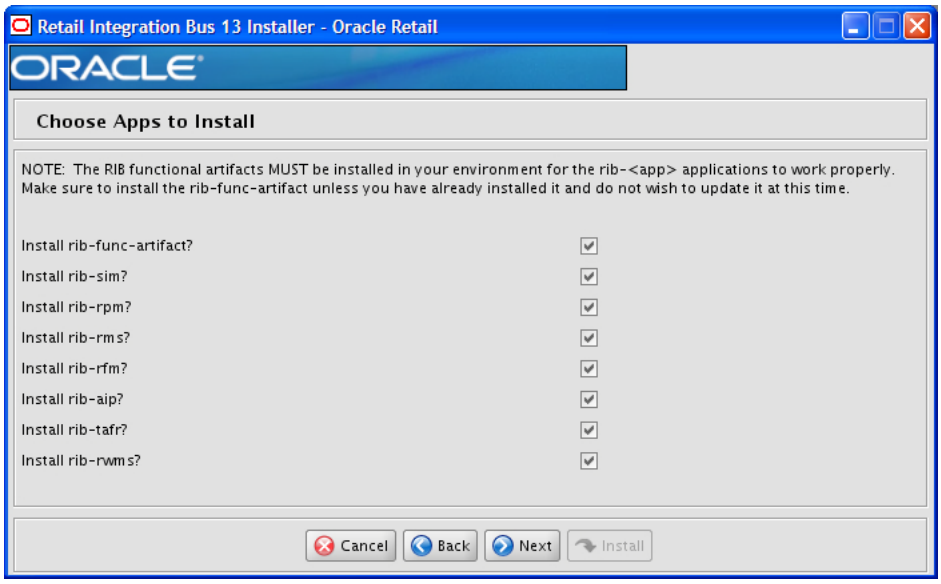
Field Title	WebLogic Admin Password
Field Description	The WebLogic admin password for this WebLogic instance.
Destination	rib-deployment-env-info.xml

Field Title	WebLogic Admin Alias
Field Description	The alias is used by the application to access user names and passwords in the wallet file cwallet.sso.
Destination	rib-deployment-env-info.xml
Example	weblogic-alias
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

Field Title	WebLogic Admin Port
Field Description	The port used to access the AdminServer for this domain. It is found in $WEBLOGIC_DOMAIN_HOME/config/config.xml.
Destination	rib-deployment-env-info.xml
Example	7001

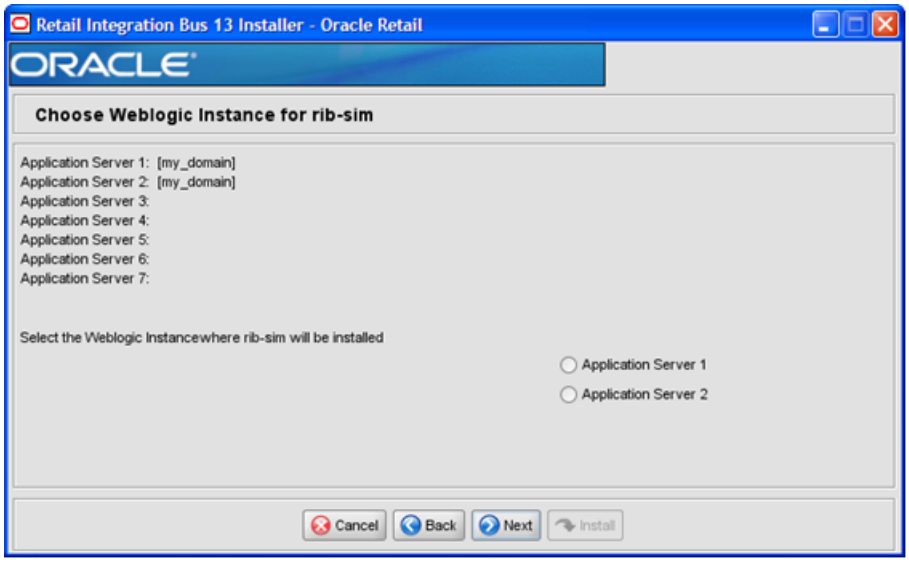
Field Title	Java Home
Field Description	The JDK in the ORACLE_HOME.
Destination	rib-deployment-env-info.xml
Example	/path/to/jdk

Screen: Choose Apps to Install



Field Title	Install rib-<app>
Field Description	Used by the installer's build.xml to determine which applications to deploy during the Deployment Phase. This screen may also be shown if you have chosen not to run the Deployment Phase, but have chosen to generate a new rib-deployment-env-info.xml file. In this case, it is used by the installer to determine the input to request from the user to create the rib-deployment-env-info.xml file.
Destination	rib-deployment-env-info.xml

Screen: Choose App Server for rib-<app>



Note: The screen capture above shows rib-sim. There are similar screens for rib-func-artifacts and for each other rib-<app> that you have chosen to install.

Field Title	Select the App Server where rib-<app> will be installed
Field Description	Used by the installer's build.xml to determine the application server with which to associate the rib-<app>'s WLS instance. **Note:** The installer requests this informaiton for rib-func-artifact, even if you have chosen to install rib-func-artifact at this time. The reason is that the rib-func-artifact inputs must exist in the rib-deployment-env-info.xml file to deploy any rib-<app:>.
Destination	rib-deployment-env-info.xml

Screen: Set Email Defaults

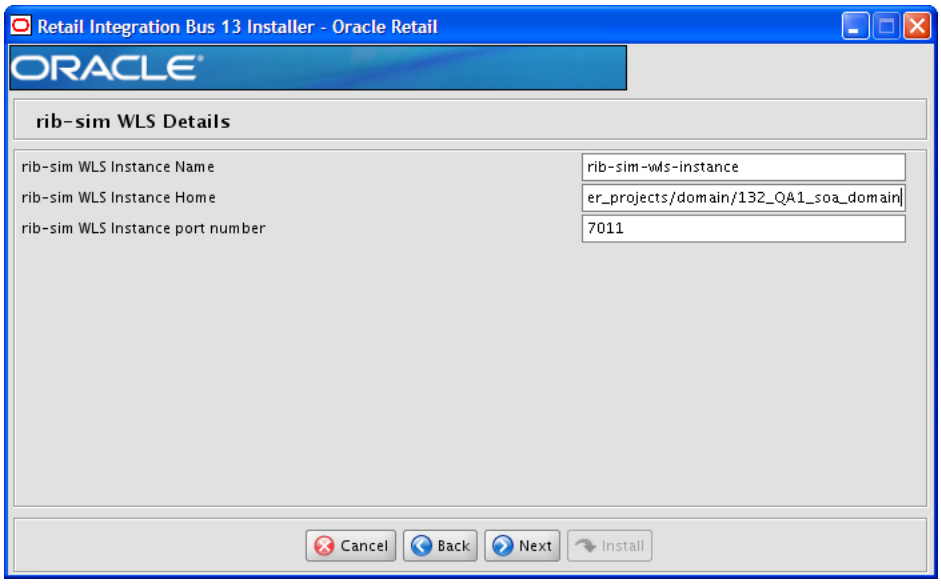
Field Title	Email server host (SMTP)
Field Description	If you are going to use the same email host for multiple rib <app> applications, you can enter it here.
Example	smtp.mycompany.com

Field Title	Email server port
Field Description	If you are going to use the same email port for multiple rib-<app> applications you can enter it here.
Example	25

Field Title	From email address
Field Description	If you are going to use the same email originator address for multiple rib-<app> applications you can enter it here.
Example	rib@mycompany.com

Field Title	To email addresses
Field Description	If you are going to use the same email recipients list for multiple rib-<app> applications you can enter it here.
Example	name1@mycompany.com, name2@mycompany.com

Screen: rib-<app> WLS Details



Note: The screen capture above shows the WLS details input screen for rib-sim. Depending on which rib-<app> applications you are installing, the installer may display one or more input screens for each rib-<app>.

Field Title	rib-<app> WLS Instance Name
Field Description	The name of the WebLogic managed server instance where the rib-<app> will be deployed.
Destination	rib-deployment-env-info.xml
Example	rib-sim-wls-instance

Field Title	rib-<app> WLS Instance Home
Field Description	The format should be as follows: <user>@<host>:<WEBLOGIC_DOMAIN_HOME>/servers /<wls-instance> where: <user> is the user who owns the files in the ORACLE_HOME <host> is the name or IP address of the server where the App Server is installed <WEBLOGIC_DOMAIN_HOME> is the filesystem path to the WEBLOGIC_DOMAIN_HOME <wls-instance> is the WebLogic managed server instance name
Destination	rib-deployment-env-info.xml
Example	myuser@myhost://<WEBLOGIC_DOMAIN_ HOME>/servers/rib-sim-wls-instance

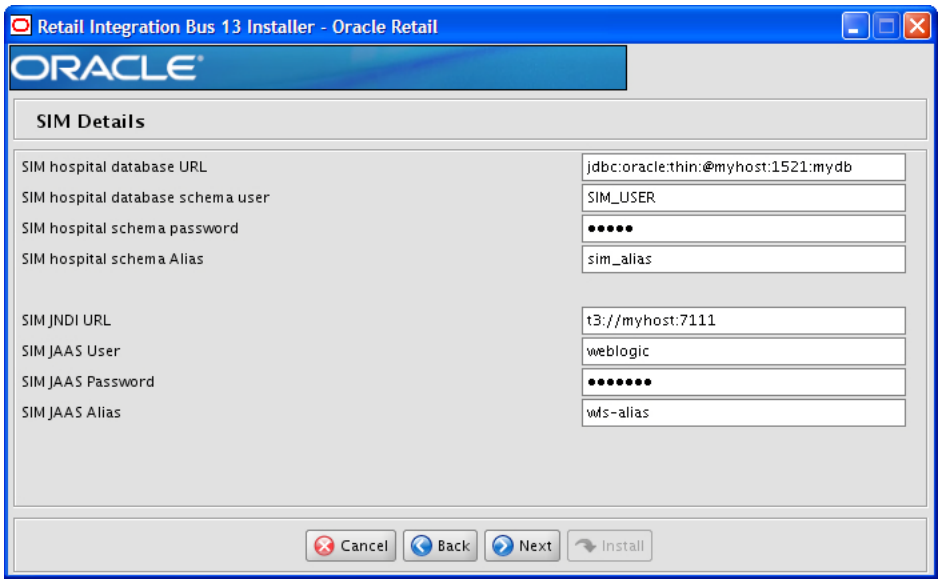
Field Title	rib-<app> WLS Instance port number
Field Description	The port number used by this WebLogic managed server instance.
Destination	rib-deployment-env-info.xml
Example	7011

Field Title	rib-<app> WLS User
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso
Example	weblogic

Field Title	rib-<app> WLS Password
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso

Field Title	rib-<app> WLS Alias
Field Description	The alias is used by the application to access user names and passwords in the wallet file cwallet.sso.
Destination	rib-deployment-env-info.xml
Example	myalias
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

Screen: <app> Details



Note: The screen capture above shows the Details screen for SIM. Depending on which rib-<app> applications you are installing, you will see different details input screens. For some of the Oracle Retail applications, these inputs may appear on separate installer screens rather than all on one screen.

Field Title	**<app> database URL**
Field Description	JDBC URL for the database
Destination	rib-deployment-env-info.xml
Example	single instance thin client: jdbc:oracle:thin:@myhost:1521:mydb

Field Title	**<app> database schema User**
Field Description	Database user where the <app> database schema was installed
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso
Example	SIM_USER

Field Title	**<app> database schema password**
Field Description	Password for the <app> database schema user
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso

Field Title	**<app> database schema alias**
Field Description	Alias for the <app> database schema user stored in the wallet file cwallet.sso.
Destination	rib-deployment-env-info.xml

Field Title	<app> database schema alias
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

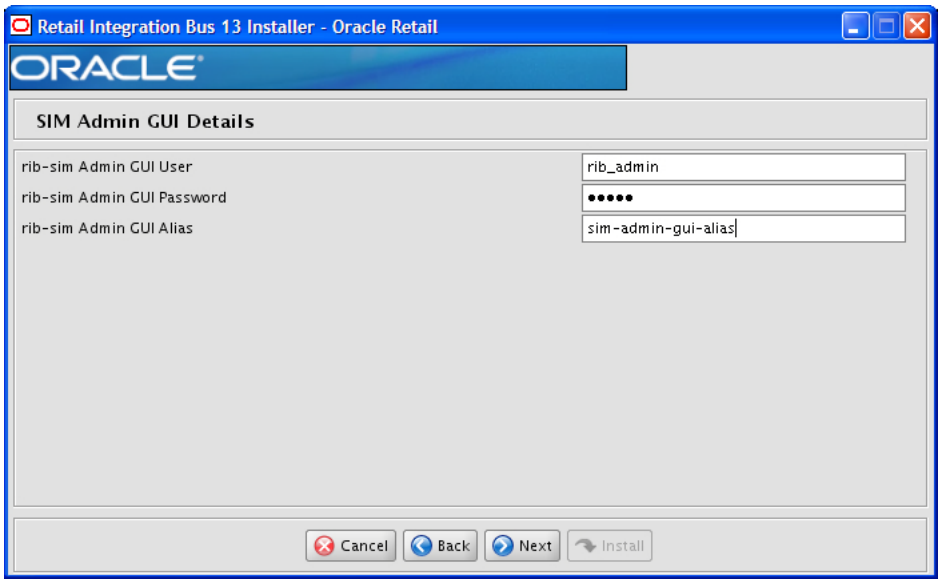
Field Title	<app> JNDI URL
Field Description	URL used by rib-<app> to connect to the <app> application. For the port, use the port number where the SIM Instance is deployed in the Web logic and not rib-sim.
Destination	rib-deployment-env-info.xml
Example	t3://myhost:7111

Field Title	<app> JAAS User
Field Description	When rib-<app> authenticates to the <app> JNDI naming service through the URL in the previous field, it will provide this user name.
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso
Example	weblogic

Field Title	<app> JAAS Password
Field Description	The password for the <app> JAAS user
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso

Field Title	<app> JAAS Alias
Field Description	The alias for the <app> JAAS user stored in the wallet file cwallet.sso.
Destination	<rib-deployment-env-info.xml
Example	sim-jaas-alias
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

Screen: rib-<app> Admin GUI Details



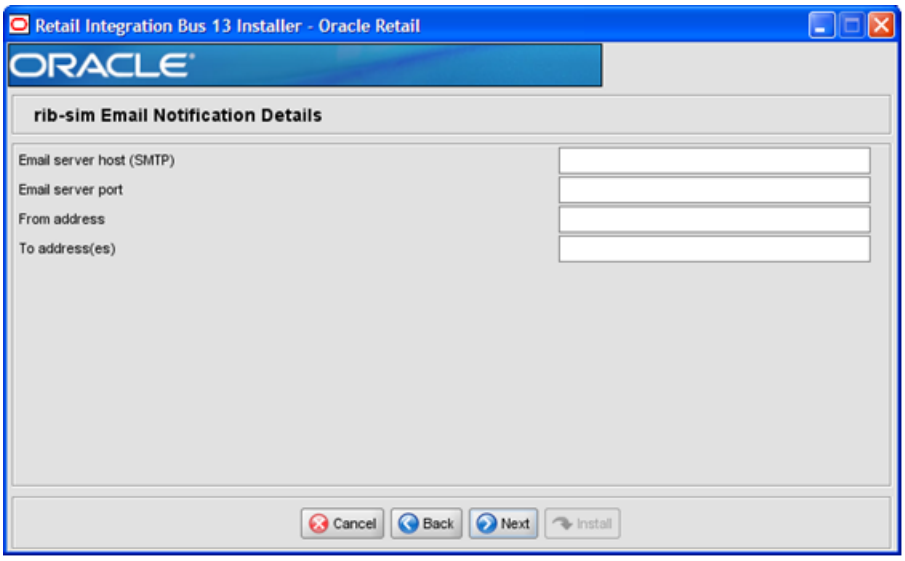
Note: The screen capture above shows the Admin GUI Details screen for rib-sim. The installer may show similar screens for other rib-<app>, depending on which rib-<app> applications you are currently installing.

Field Title	<app> Admin GUI User
Field Description	When logging in to the admin GUI for rib-<app>, use this user name.
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso
Example	weblogic

Field Title	<app> Admin GUI Password
Field Description	The password for the <app> Admin GUI user.
Destination	<RIB_HOME>/deployment-home/conf/security/cwallet.sso
Notes	For WLS 10.3.x, passwords must include at least one numeral.

Field Title	<app> Admin GUI Alias
Field Description	The alias for the <app> Admin GUI user stored in the wallet file cwallet.sso.
Destination	rib-deployment-env-info.xml
Example	sim-admin-gui-alias
Notes	This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application.

Screen: rib-<app> Email Notification Details



Note: The screen capture above shows the Email Notification Details screen for rib-sim. The installer may show similar screens for other rib-<app>, depending on which rib-<app> applications you are currently installing.

Field Title	Email server host (SMTP)
Field Description	The SMTP server that will be used to send notification emails from RIB.
Destination	rib-deployment-env-info.xml
Example	smtp.mycompany.com

Field Title	Email server port
Field Description	The port for outgoing emails
Destination	rib-deployment-env-info.xml
Example	25

Field Title	From address
Field Description	The email address from which the rib-<app>email notifications will originate.
Destination	rib-deployment-env-info.xml
Example	rib@mycompany.com

Field Title	To addresses
Field Description	List of recipients for rib-<app> email notifications.
Destination	rib-deployment-env-info.xml

Field Title	To addresses
Example	name1@mycompany.com, name2@mycompany.com
Notes	

B

Appendix: RIB Installer Common Errors

This appendix provides some common errors encountered during installation to aid in troubleshooting.

Unreadable Buttons in the Installer

If you are unable to read the text within the installer buttons, it could mean that your JAVA_HOME is pointed to an older version of the JDK than is supported by the installer. Set JAVA_HOME to a JDK 1.6.0+ 64 bit or Jrockit 1.6 R28 build or later (64 bit for Linux and Solaris OS only) and run the installer again.

Warning: Could not Create System Preferences Directory

Symptom:

The following text appears in the installer Errors tab:

```
May 22, 2010 11:16:39 AM java.util.prefs.FileSystemPreferences$3 run
WARNING: Could not create system preferences directory. System preferences are
unusable.
May 22, 2010 11:17:09 AM java.util.prefs.FileSystemPreferences
checkLockFile0ErrorCode
WARNING: Could not lock System prefs. Unix error code -264946424.
```

Solution:

This is related to Java bug 4838770. The /etc/.java/.systemPrefs directory may not have been created on your system. See http://bugs.sun.com for details.

This is an issue with your installation of Java and does not affect the Oracle Retail product installation.

ConcurrentModificationException in Installer GUI

Symptom:

In GUI mode, the Errors tab shows the following error:

```
java.util.ConcurrentModificationException
........at
 java.util.AbstractList$Itr.checkForComodification(AbstractList.java:448)
........at java.util.AbstractList$Itr.next(AbstractList.java:419)
........etc.
```

Solution:

You can ignore this error. It is related to third-party Java Swing code for rendering of the installer GUI and does not affect the retail product installation.

Warning: Could Not Find X Input Context

Symptom:

The following text appears in the console window during execution of the installer in GUI mode:

```
Couldn't find X Input Context
```

Solution:

This message is harmless and can be ignored.

Message: Problem Occurred during Parsing Input XML Files

Symptom:

The following text appears in the console window during execution of the installer:

```
ERROR oracle.retail.rib.compiler.Main - Problem occurred during parsing input xml
files. Please check the log file(../../../rib-home/application-assembly-home/log)
for more details.
.....
Caused by: ValidationException:
.....
```

Solution:

The rib-deployment-env-info.xml file is validated during the Assembly Phase using stricter criteria than is enforced by the installer input screens. If the validation fails, the installer will print an error message to help you determine the cause of the validation failure. It is recommended that you fix the rib-deployment-env-info.xml file manually, and then re-run the installer with the "Use existing rib-deployment-env-info.xml" option.

rib-app-builder Hangs if a User is Logged in to the Administration Console during Deployment

If the WebLogic server is installed in development mode, the rib-app-builder may "hang" if a user is already logged into the admin console. Click **Activate Changes** in the admin console to continue with rib-app-builder process every time it happens. To prevent this from occuring, use the option in the WebLogic admin console.

1. Click the **Preferences** link on the admin console.
2. Uncheck the checkbox, "Automatically Acquire Lock and Activate Changes."
3. Click **Release Configuration.**

The above process will enable only one user to make admin changes to the server. For making admin changes to the console, the user must explicitly acquire the lock, make changes, and release configuration.

Make sure the Release Configuration button in the admin console is disabled before starting to deploy rib apps. So even if you navigate to other screens while deploying RIB apps, the rib deployer will not "hang."

Note: If the WebLogic server is installed in production mode, by default it acquires the lock before making any changes to the server. Therefore, the above steps need not be followed when deploying rib-apps to a WebLogic server running in production mode.

C

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 either the GUI RIB Installer (strongly recommended) or 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 one of these methods: ■ Installation using the RIB Installer GUI ■ Installation using the RIB App Builder Command Line Tools.	It is strongly recommended that the Installation using the RIB Installer GUI method be used.
Verify Application URL settings match RIB installation.	RIB Functional Artifact URL JNDI URL

Task	Notes
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
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 10.3.4	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: blrvmo28 Server Listen Port:19007	> with the actual value of the RIB application for the associated retail application. 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-wls-instance ■ rib-tafr-wls-instance ■ rib-rpm-wls-instance ■ rib-sim-wls-instance ■ rib-rwms-wls-instance ■ rib-rfm-wls-instance ■ rib-aip-wls-instance
Click **Next**. Click **Finish**. Make sure you see this instance listed under Servers.	

Task	Notes
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.	
> with the actual value of the RIB application for the associated retail application. Acceptable values for <app> are "rms", "rwms", "tafr", "sim", "rpm", "rfm" and "aip." Port number must be a unique port. **Note**: For information about creating and managing WebLogic managed server instances, see Oracle® Fusion Middleware Administrator's Guide 11g Release 1 (11.1.1)	

Task	Notes

Edit the script $DOMAIN_HOME/base_domain/bin/startWebLogic.sh to add the following attributes.
CLASSPATH=$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.MDBMessageWaitTime=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_10.3>/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";
	
	};
	
	};

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/10.3.X_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_10.3/common/node 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. ■ InstallAndCompileAllRibOracleObjects.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 13.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 "TAFRHOSP" IDENTIFIED BY "TAFRHOSP" DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP"; GRANT "CONNECT" TO " TAFRHOSP "; GRANT "RESOURCE" TO " TAFRHOSP "; ALTER USER "TAFRHOSP" QUOTA UNLIMITED ON USERS;
Ensure that the XA grants are made appropriately. Note: For details, see the Oracle® Database Administrator Guide 11g Release 2.	grant select on v$xatrans$ to public; grant select on pending_trans$ to public; Verify that the XA scripts have been run on the database.grant select on dba_2pc_pending to public; grant select on dba_pending_transactions to public; grant execute on dbms_system to public;

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 "RIBAQ" IDENTIFIED BY "RIBAQ"`
	`DEFAULT TABLESPACE "RETEK_DATA"`
	`TEMPORARY TABLESPACE "TEMP";`
	`GRANT "CONNECT" TO "RIBAQ";`
	`GRANT "RESOURCE" TO "RIBAQ";`
	`GRANT CREATE SESSION TO "RIBAQ";`
	`GRANT EXECUTE ON "SYS"."DBMS_AQ" TO "RIBAQ";`
	`GRANT EXECUTE ON "SYS"."DBMS_AQADM" TO "RIBAQ";`
	`GRANT EXECUTE ON "SYS"."DBMS_AQIN" TO "RIBAQ";`
	`GRANT EXECUTE ON "SYS"."DBMS_AQJMS" TO "RIBAQ";`
	`GRANT "AQ_ADMINISTRATOR_ROLE" TO "RIBAQ";`
	`ALTER USER "RIBAQ"`
	`QUOTA UNLIMITED ON RETEK_DATA;`

Information	Notes
jms-server-home jms-url jms-port jms-user jms-password	JMS Provider for RIB 13.2.3 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/user_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/jdk1.6.0_10)
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-wls-instance. ■ 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-wls-instance ■ nwls-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 (for example, rms13user). ■ 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@yourcompany.com) ■ email/to-address-list: Comma separated list of destination email address to which RIB notifications will be sent (for example,ribappsadmin1@yourcompany.com, ribappsadmin2@yourcompany2.com)

Information	Notes
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://mspdv170:18022/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. For example, weblogic ■ jndi/password: The retail <app> JNDI password. This is same as the retail <app> WLS server instance password.

Install Using the RIB Installer GUI

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/1.6.0+ 64 bitor Jrockit 1.6 R28 build or later, within 1.6 code line. 64 bit. For Linux and Solaris OS only.	Example: export JAVA_HOME=/usr/bin/java/1.6.0+ 64 bit or Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only.
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 RibKernel13.2.3ForAll13.x.xApps_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 rib-<app> related tasks from now on.

Task	Notes
Download the RibFuncArtifact<RIB_{MAJOR|MINOR}_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 all the RibPak<RIB_{MAJOR|MINOR}_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.
Return to the root rib-home directory.	
Execute rib-installer.sh >./rib-installer.sh This will start the x-term GUI.	For installations using a remote client (x-term) set the DISPLAY variable appropriately first. > export DISPLAY=141.144.112.189:0.0 Make sure that your local machine has an X-server (such as Exceed) running.
Verify Application URL settings match RIB installation.	RIB Functional Artifact URL JNDI URL
Bounce all rib-<app>-wls server instances.	During the installation a shared library is created that contains the JDBC Driver update. It is necessary to bounce the instance.
Verify the installation using RDMT.	

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/jdk1.6.0_18	Example: export JAVA_HOME=/usr/bin/java/jdk1.6.0_18
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.

Task	Notes
Download and extract the RibKernel<RIB_MAJOR_VERSION>ForAll<RETAIL_APP_VERSION>Apps_eng_ga.tar. > tar xvf RibKernel13.2.3.ForAll13.x.x 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.
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 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 application 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="rfm" type="plsql-app"/>`
	`<app id="aip" type="javaee-app"/>`
	`</app-in-scope-for-integration>`

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/10.2.0/db_1</jms-server-home>`
	
	`<jms-url>jdbc:oracle:thin:@linux1:1521:ora11g</jms-url>`
	`<jms-port>1521</jms-port>`
	`<jms-user>ribaq</jms-user>`
	`<jms-password>password</jms-password>`

Task	Notes
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/jdk1.6</java-home>
Configure the WebLogic instances for each of your rib-<app> applications that are in scope.	<wls id="rib-rms-wls-instance"> <wls-instance-name>rib-rms-wls-instance</wls-instance-name> <wls-instance-home>soa1@linux1:/home/soa1/Oracle/Middleware/user_projects/domains/base_ domain/servers/rib-rms-wls-instance</wls-instance-home> <wls-listen-port protocol="http">7003</wls-listen-port> <wls-user-alias>rib-rms-wls-user-alias</wls-user-alias> </wls>
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:@10.141.27.136: 1521:orcl </hosp-url> <hosp-user>hospuser</hosp-user> <hosp-password>password</hosp-password> </error-hospital-database> <app-database> <app-db-url>jdbc:oracle:thin:@10.141.27.136: 1521:orcl </app-db-url> <app-db-user>rmsuser</app-db-user> <app-db-password>password</app-db-password> </app-database> <notifications> <email> <email-server-host>mail.oracle.com</email-server-host> <email-server-port>25</email-server-port> <from-address>david.burch@oracle.com</from-address> <to-address-list>david.burch@oracle.com</to-address-list> </email> <jmx/> </notifications>

Task	Notes
	<rib-app id="rib-rms" type="plsql-app"> <deploy-in refid="rib-rms-wls-instance"/> <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:@10.141.20.184:1521:ora11gr1 </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:@10.141.20.184:1521:ora11gr1</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-host> <email-server-port>25</email-server-port> <from-address>prem.polavarapu.bad@oracle.com</from-address> <to-address-list>prem.polavarapu.bad@oracle.com</to-address-list> </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.
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

Task	Notes
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.
> 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	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt/
RDMTLOGS Directory	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt /RDMTLOGS
Temp Files Directory	Rib1324ForAll13xxApps/rib-home/tools-home/rdmt /RDMTLOGS/tmp
RIB App Builder rib-home Directory	/u00/Rib1324ForAll13xxApps/rib-home

The following are parameters for JMS Provider.

AQ JMS User ID	ribaq
AQ JMS Password	#Password#
AQ JMS Database Name	soa1
JMS HOST	mspdev38

JMS PORT	1521

The following are WebLogic parameters for JMX functions.

WebLogic/JMX Host	mspdev72
JMX Req Port	7003
WebLogic Instance name	rib-rms-wls-instance
WebLogic App Name	rib-rms
WebLogic User Name	weblogic

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

User Name	rms
Password	#Password#
Database (SID)	orcl
Database Host	mspdev68
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 5.0. > java -version	The RDMT Java support classes require Java 5.0, 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 Rdmt13.2.3ForAll13.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 Rdmt13.2.3ForAll13.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.

Task	Notes
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.
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 1.6.
The RIB XSDs must be made network-accessible for RIHA to properly display RIB messages.	The RIB Functional Artifact URL (for example, http://mspdev85:7777/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 RIHA13.2.3 is compatible only with RIB13.0.X and higher.

Task	Notes
Verify ADF runtime 11.1.1.0 or higher is available in WebLogic Application Server 10.3.4 is applied to the 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	
Log in to the WebLogic Console	Log in to the WebLogic server console where RIHA will be installed.
Setup Data Sources	1. In the left pane, click Services > JDBC > DataSources > New. (Make sure the Lock and Edit option is selected to enable New.) 2. On the JDBC Data Source Properties page, Enter the following details: Name: riha-<app>DS JNDI: jdbc/riha-<app>DS (follow this naming convention) 3. Click **Next**. 4. Add database driver: Oracle's driver (Thin XA) for instance connections, versions 9.0.1, 9.2.0, 10, 11 5. Click **Next**. 6. On Transaction Options page, click **Next**. 7. On Connection Properties page, enter the following details: Database Name Host Name Port Database User Name: Password 8. Click **Next**. 9. On the Test Database Connection page, click **Test Configuration** and make sure the test is successful. 10. Click **Finish**. **Note**: The jdbc resource must be targeted to the same managed server.

Task	Notes
Deploy EAR	1. Download and untar the RibHospitalAdministration-web-13.2.4ForAll13.x.xApps_eng_ga. tar. 2. From the server console, select Deployment in Domain Structure from the left pane. Click **Install**. (Make sure the Lock and Edit option is selected if the **Install** button is inactive.) 3. Select Path of EAR. Select EAR file. Click **Next**. 4. In Choose Target Style, select Install this Deployment as an Application. Click **Next**. 5. Select deployment targets from one of the available servers. Click **Next**. 6. In Optional Settings: Check **Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console** radio button in Security Option. Check **I will make the deployment accessible from the following location** option under Source Accessibility Option. Click **Next**. 7. Click **Finish**. Check logs for errors. **Note:** If the admin server console issues a message to restart the server at this point, restart the server.
Post Deployment Configuration	
Security and Roles	1. From the post deployment screen, select the Security tab. 2. Add role: RihaUserRole. 3. Click **Finish**. 4. Select UserRole -> Conditions. 5. Select User from the select box for Predicate List. Click **Next**. 6. Add user argument: <desired user name>. 7. Click **Finish**. Click **Save**. **Note:** If the admin server console issues a message to restart the server at this point, restart the server.
Policies	1. From the select box for Predicate List, select the role created in the "Security and Roles" steps above. 2. Click **Next**. 3. Click **Finish.**
Security Realms	1. In the left pane, Select Security Realms > myRealms 2. In the Users and Group Tab: Add new user and enter user name/password.
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	ora11g	AQ Database instance name
Host Name	linux1.us.oracle.com	Database system
Port	1521	Database listener port
Database User Name	RIBAQ	AQ user
Password	#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 IntegrationGatewayService 13.2.4ForAll13.2.4Apps.tar	
Extract the tar file	>tar -xvf IntegrationGatewayService13.2.4forAll13.2.4Apps_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://mspdv170:18001 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.

Task	Notes
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.

D

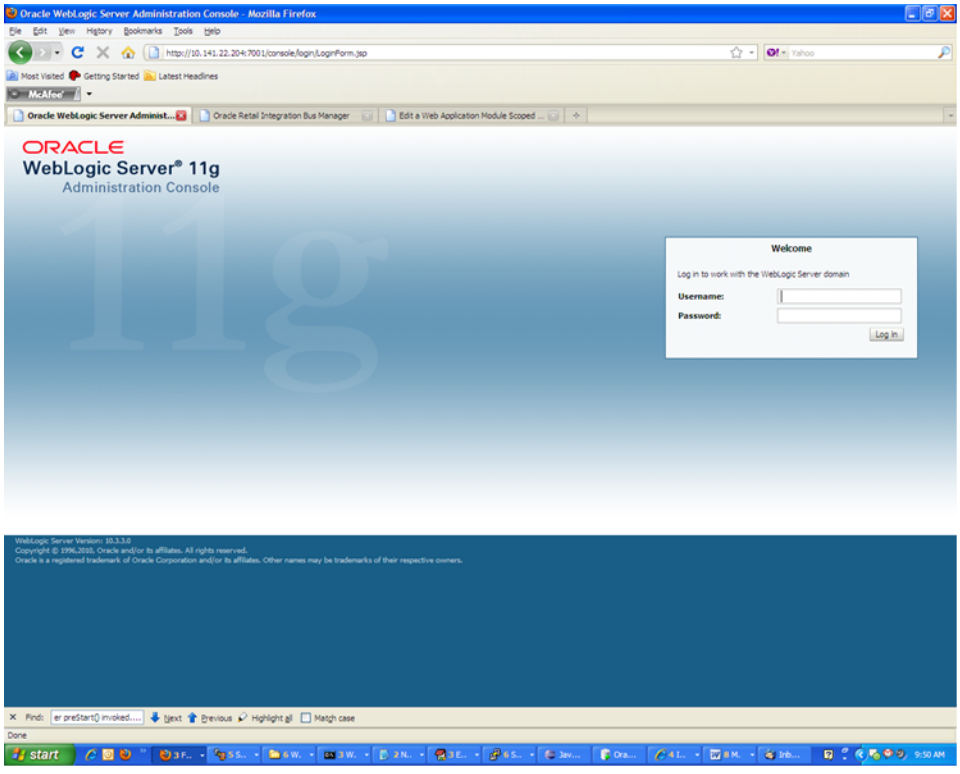
Appendix: Changing the RIB Admin GUI Password

This appendix describes the steps required to change the RIB Admin GUI password.

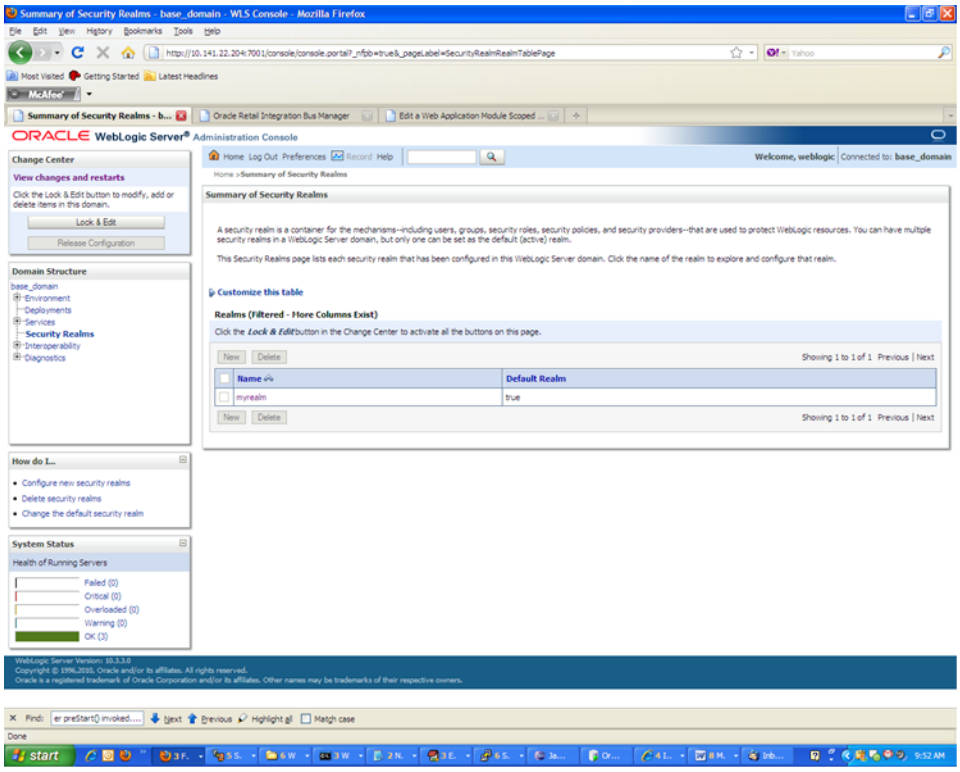
Procedure

To change the RIB Admin GUI password, complete the following steps.

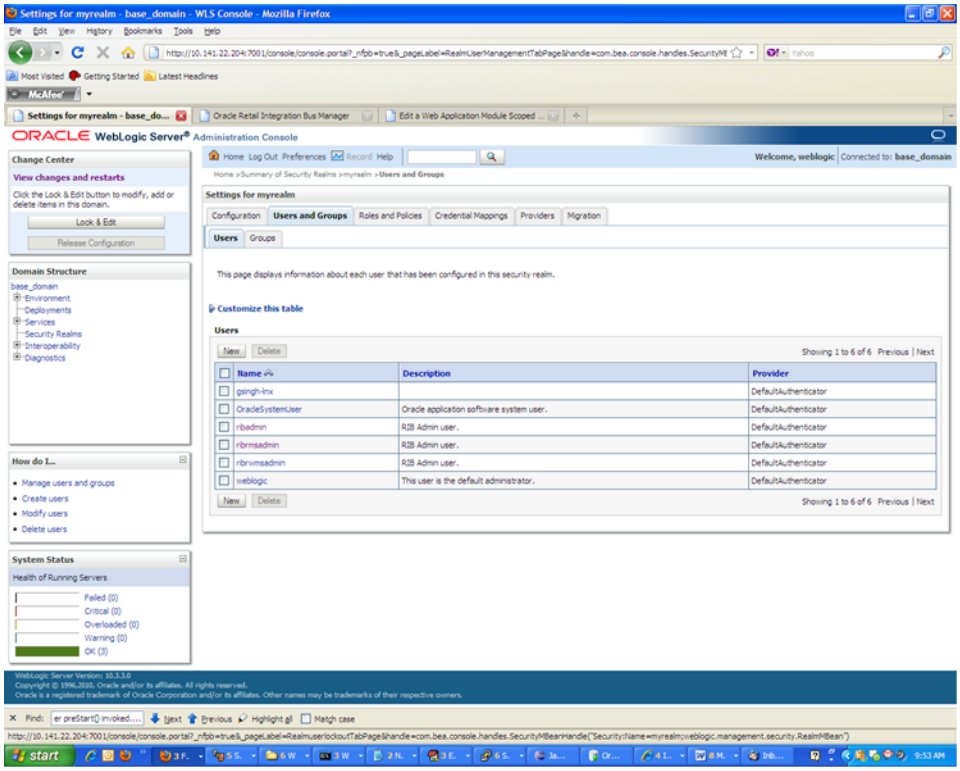
1. Log in to the WebLogic Server Administration Console.



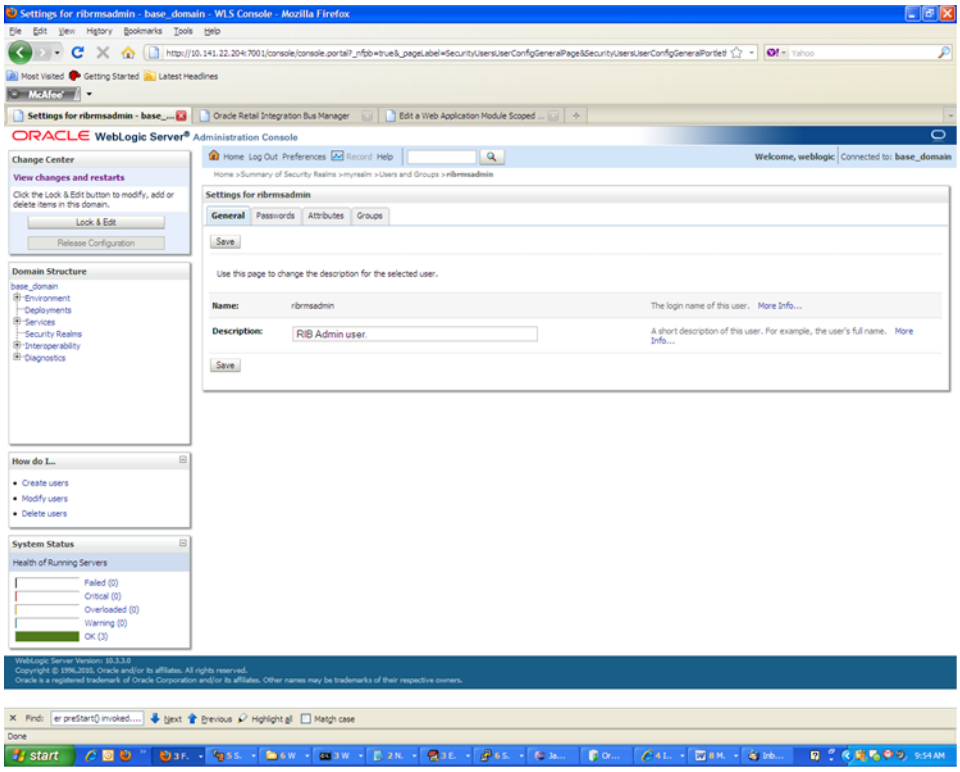
2. In the left panel, click the **Security Realms** link.



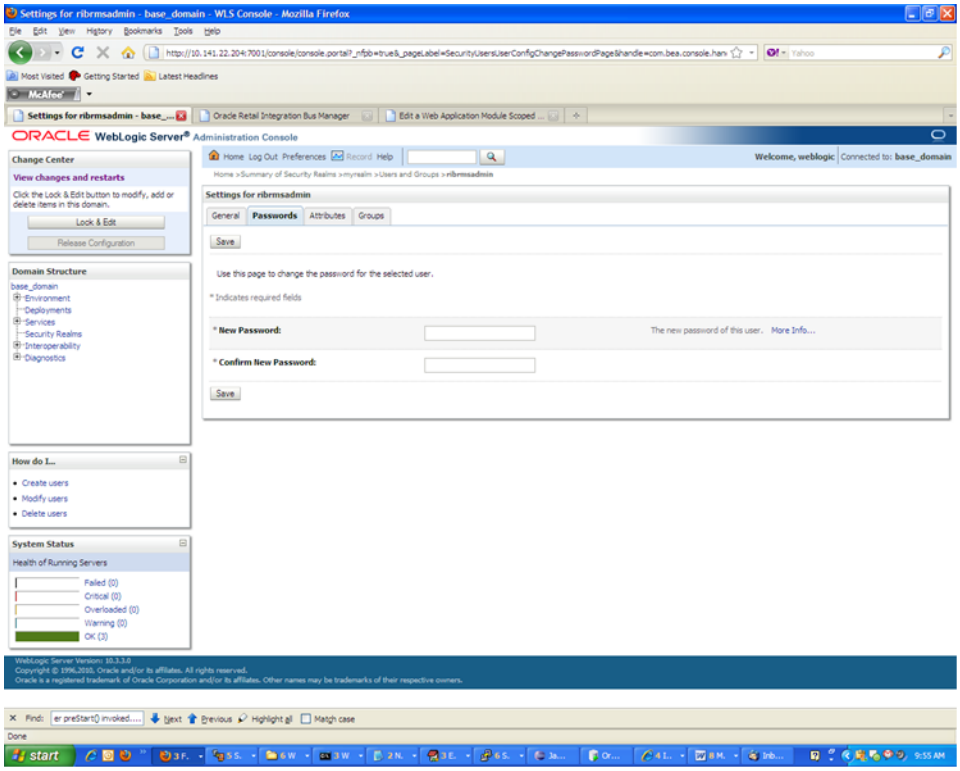
3. Click the realm name. Go to the Users and Groups tab.



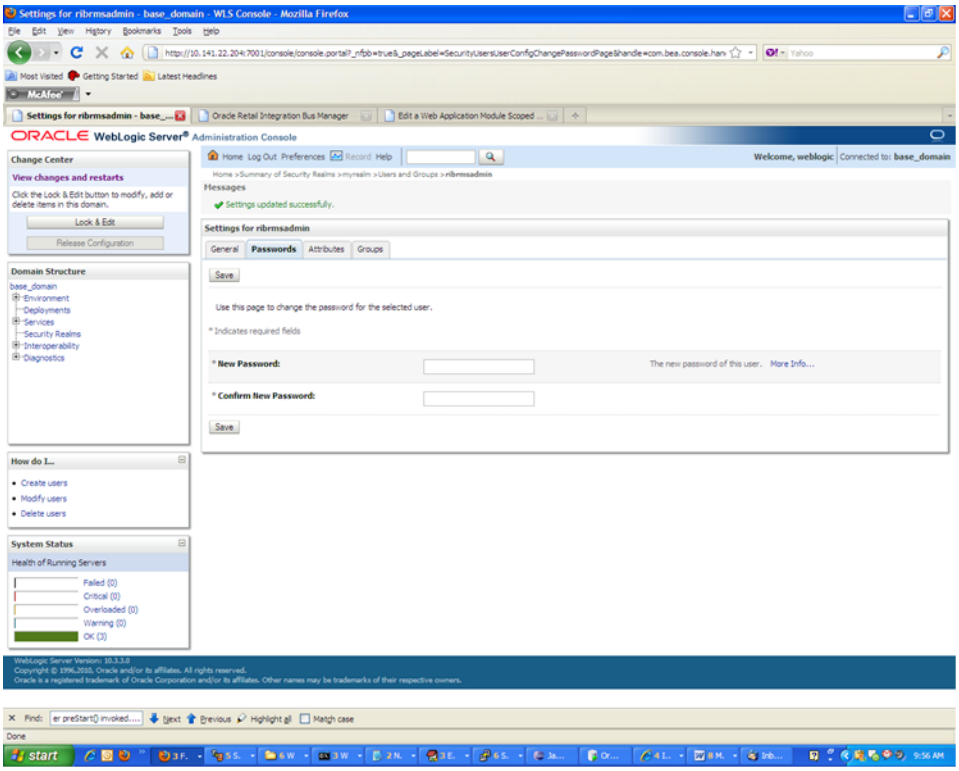
4. Click the user name for which you want to change the password.



5. Click the **Passwords** tab.



6. Enter the new password. Click **Save.**


Settings for ribrmsadmin - base_domain - WLS Console - Mozilla Firefox
ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
base_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create users
Modify users
Delete users
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Messages
Settings updated successfully.
Settings for ribrmsadmin
General Passwords Attributes Groups
Save
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:
Save
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.
Done
start

E

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()
```

F

Appendix: Installation Order

This appendix provides a guideline for the order in which the Oracle Retail applications should be installed. If a retailer has chosen to use only some of the applications, the order is still valid, less the applications not being installed.

> **Note:** The installation order is not meant to imply integration between products.

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM), Oracle Retail Sales Audit (ReSA), Optional: Oracle Retail Fiscal Management (ORFM).

   > **Note:** ORFM is an optional application for RMS if you are implementing Brazil localization.

2. Oracle Retail Service Layer (RSL)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Price Management (RPM)
7. Oracle Retail Invoice Matching (ReIM)
8. Oracle Retail Allocation

   > **Note:** During installation of RPM, you are asked for the RIBforRPM provider URL. Because RIB is installed after RPM, make a note of the URL you enter. If you need to change the RIBforRPM provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

9. Oracle Retail Central Office (ORCO)
10. Oracle Retail Returns Management (ORRM)
11. Oracle Retail Back Office (ORBO) or Back Office with Labels and Tags (ORLAT)
12. Oracle Retail Store Inventory Management (SIM)

Note: During installation of SIM, you are asked for the RIB provider URL. Because RIB is installed after SIM, make a note of the URL you enter. If you need to change the RIB provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (CM)
16. Oracle Retail Replenishment Optimization (RO)
17. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
18. Oracle Retail Regular Price Optimzation (RPO)
19. Oracle Retail Merchandise Financial Planning (MFP)
20. Oracle Retail Size Profile Optimization (SPO)
21. Oracle Retail Assortment Planning (AP)
22. Oracle Retail Item Planning (IP)
23. Oracle Retail Item Planning Configured for COE (IP COE)
24. Oracle Retail Advanced Inventory Planning (AIP)
25. Oracle Retail Integration Bus (RIB)
26. Oracle Retail Point-of-Service (ORPOS)
27. Oracle Retail Markdown Optimization (MDO)
28. Oracle Retail Clearance Optimization Engine (COE)
29. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
30. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
31. Oracle Retail Promotion Intelligence and Promotion Planning and Optimization (PI-PPO)
32. Oracle Retail Analytics
33. Oracle Retail Workspace (ORW)