

Oracle® Retail Returns Management

Installation Guide

Release 1.0

November 2006

Oracle Retail Returns Management Installation Guide, Release 1.0

Copyright © 2006, Oracle. All rights reserved.

Primary Author: Bernadette Goodman

Contributor: Robin Kar

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	ix
Audience.....	ix
Related Documents	ix
Customer Support	ix
 1 Application Overview	
Application Components.....	1-1
 2 Preparing to Deploy	
Deployment Overview	2-1
Hardware Requirements	2-1
Software Requirements.....	2-2
Oracle Retail Returns Management Installation	2-3
Database Setup	2-3
JMS Configuration	2-3
Application Server Configuration	2-3
Oracle Retail Returns Management Deployment.....	2-4
Unpacking Oracle Retail Returns Management from the Oracle Web Site	2-4
Loading the Oracle Retail Returns Management Database.....	2-4
 3 Configuring WebSphere MQ	
Installing WebSphere MQ, WebSphere Application Server, and Fix Packs	3-1
Configuring WebSphere MQ.....	3-1
Creating the MQ Queues with an Input File.....	3-2
createq.dat Input File.....	3-2
Creating the MQ Queues with a Command Line.....	3-3
Useful WebSphere MQ Commands	3-3
 4 Configuring WebSphere Application Server	
Required Resources	4-1
Configuring WebSphere Application Server.....	4-1
Configuring WebSphere Using JACL Scripts	4-2
Editing the was.properties File	4-2

Executing the ws_ant Targets	4-3
Completing the Configuration.....	4-3
Configuring WebSphere Using the Administrative Console	4-4
Creating the Custom Security Jar File	4-4
Installing and Editing Necessary Files.....	4-4
Setting up the WebSphere MQ JMS Provider.....	4-5
Setting up the JDBC Providers.....	4-8
Setting up the Message Listener Service	4-13
Configuring the Mail Provider	4-14
Configuring Security	4-14
Configuring a Headless Environment and Other JVM Options.....	4-15

5 Deploying Oracle Retail Returns Management in WebSphere

Installing and Starting Oracle Retail Returns Management	5-1
Importing Initial Parameters.....	5-2
Importing Parameters Through the User Interface.....	5-2
Importing Parameters By Using an Ant Target.....	5-2

A WebSphere MQ and Application Server Installation

Installing WebSphere MQ Version 5.3 and Service Pack 11	A-1
Testing the WebSphere MQ Installation.....	A-2
Installing the WebSphere Application Server Version 5.1.....	A-3
Installing WebSphere Application Server Fix Packs.....	A-8

B Returns Data Loader

Using the Returns Data Loader	B-1
-------------------------------------	-----

List of Examples

2-1	Properties for Pre-loading Returns Management Database	2-5
3-1	createq.dat Input File.....	3-2
3-2	Output Displayed When Creating a Queue.....	3-3
4-1	was.properties File.....	4-2

List of Figures

2-1	Directory Tree Created for Database Load	2-5
4-1	Connection Pool Configuration	4-7
4-2	Custom Properties for JDBC XA provider Data Source	4-10
A-1	Output of Queue Manager Creation	A-2
A-2	WebSphere Application Server Launchpad Screen	A-3
A-3	Operating System Level Check Screen	A-4
A-4	Setup Type Selection Screen	A-4
A-5	Feature Selection Screen	A-5
A-6	Installation Directories Selection Screen	A-6
A-7	Node Name and Host Name Screen	A-6
A-8	Selected Features List Screen	A-7
A-9	Product Registration Screen	A-7
A-10	Installation Finish Screen	A-7
A-11	Product Selection Screen	A-9
A-12	Fix Pack and Fix Installation and Uninstallation Screen	A-9
A-13	Fix Pack Directory Screen	A-10
A-14	Fix Pack Selection Screen	A-10
A-15	Select External Components Screen	A-11
A-16	Fix Pack Review Screen	A-11
A-17	Fix Pack Finish Screen	A-12

List of Tables

2-1	Supported Hardware.....	2-2
2-2	Supported Software	2-2
3-1	Useful MQ Commands	3-3
3-2	Useful Queue Manager Shell (runmqsc) Commands.....	3-4
4-1	Required Resources	4-1
4-2	Sub-targets for ws_ant configure.....	4-3
4-3	Sample J2C Authentication Data	4-5
4-4	Sample Queue Connection Factory Data.....	4-6
4-5	Sample MQ Queue Destination Data for the EJournalImport Queue	4-7
4-6	Required Queue Destinations	4-8
4-7	Sample J2C Authentication Data for the XA Data Source	4-9
4-8	Sample J2C Authentication Data for the Non-XA Data Source	4-9
4-9	Sample JDBC XA Provider Data	4-9
4-10	Sample JDBC XA Provider Data Source Data.....	4-10
4-11	Sample JDBC Non-XA Provider Data.....	4-11
4-12	Sample JDBC Non-XA Provider Data Source Data	4-11
4-13	Sample JDBC XA Provider CrystalClear Data Source Data	4-12
4-14	Sample Message Listener Service Data for the EJournalImportPort.....	4-13
4-15	Required Listener Ports.....	4-14
4-16	Sample Mail Provider Data	4-14
4-17	Sample Custom Registry Data	4-15
4-18	Custom Properties Data	4-15
4-19	Sample Global Security Data.....	4-15
4-20	Sample Java Virtual Machine Data.....	4-16

Preface

The purpose of this document is to guide administrators and developers through the deployment process for Oracle Retail Returns Management. It includes hardware and software prerequisites and the steps to deploy the application in a WebSphere application server.

Audience

This document is intended for administrators and developers who install Oracle Retail Returns Management.

Related Documents

For more information, see the following documents in the Oracle Retail Returns Management Release 1.0 documentation set:

- *Oracle Retail Returns Management Release Notes*
- *Oracle Retail Returns Management Operations Guide*
- *Oracle Retail Returns Management User Guide*

Customer Support

- <https://metalink.oracle.com>

When contacting Customer Support, please provide:

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

Application Overview

Customers try to return many types of products, for a wide variety of reasons. The procedure for managing this process may vary greatly from store to store and manager to manager, and perhaps even from cashier to cashier.

Oracle Retail Returns Management is a proactive, centralized, multi-channel solution that provides the ability to reduce overall return rates, prevent and catch return fraud, and improve customer service. It provides the following benefits:

- Definition and enforcement of consistent policies about returns and other types of transactions that may result in a customer refund.
- The customer, cashier, or manager can be informed of the reason why a return received a particular response. This can be done through online or printed messages.
- Patterns of customer return behavior can be studied to determine if a return merits return denial, or at least a prompt for manager approval at the point-of-return.
- Online research of customer and cashier return activity.
- Customer service overrides for denied returns can be granted, if necessary.

Application Components

The application is comprised of the following components:

- Multiple return policies and exception policies allow the retailer to centrally manage returns, using configurable rules that can be evaluated at the merchandise, customer, or location level to determine authorization. This provides quick adaptation to changes in business and seasonal conditions.
- Defined XML messages are sent from the point-of-return or other channel to trigger action in Oracle Retail Returns Management and an appropriate response message. This allows Oracle Retail Returns Management to be rapidly integrated to work with existing store and channel systems.
- An Analytic Engine checks Return Policies to evaluate the returnability of an item. Item returnability is determined by examining the item, customer, associate, and store in question. A real-time message is then sent advising the point-of-return on the most appropriate path to take.
- The Exception File is a constantly evolving knowledge base for preventing fraudulent returns by tracking which shoppers and cashiers have made exceptions to the defined return policy and therefore are at a higher risk of fraud.

- Return Tickets act as a Customer Service Record (CSR) that allows inquiry and audit of the trail of return activity for a given customer, associate, item, or store. This allows the retailer to handle customer service inquiries into which steps may have been taken to prevent return fraud.
- A web-based user interface for Loss Prevention, IT, Store Operations, Call Center, and field personnel to configure and manage the returns process.

Preparing to Deploy

This chapter gives a general summary of the deployment process and describes the prerequisites you must satisfy prior to deploying Oracle Retail Returns Management. Actual deployment steps vary for different application servers and databases; therefore, this book is intended to be used with the product documentation provided with your database, application server, and other software. The following chapter describes the configuration for the WebSphere application server in greater detail.

Deployment Overview

An Oracle Retail Returns Management deployment requires that you complete the following tasks:

- Verify that all hardware requirements are met.
- Verify that all software prerequisites are installed and configured.
- Configure your JMS queue server software.
- Install required `.jar` and `.properties` files into the application server.
- Configure your application server with data sources, JMS Queues, and other resources.
- Load your database with initial data.
- Deploy Oracle Retail Returns Management into the application server.
- Configure Oracle Retail Returns Management for your specific business.

Hardware Requirements

Specific hardware requirements for running Oracle Retail Returns Management depend on several variables:

- Transaction volume
- Returns data retention periods
- Expected number of users
- Number of other applications running on the same application server

Work with Oracle Retail Professional Services to determine the specific requirements for your situation.

[Table 2–1](#) lists hardware in the required categories which have been tested for this release in at least one configuration.

Table 2–1 Supported Hardware

Item	Supported Products
Hardware	Dell 2650
CPU	Intel 2.8 GHz
Memory	1 GB

Software Requirements

Oracle Retail Returns Management is packaged as an Enterprise Archive (.ear) file and must be deployed to a J2EE 1.3-compliant application server. Oracle Retail software is designed to open standards and is compatible with many operating systems, application servers, databases, and other third-party software.

The Oracle Retail Returns Management application requires that the following be installed and configured on one or more host server machines:

- Java Development Kit (JDK) v. 1.4.2 or later
- A J2EE 1.3-compliant application server
- JMS messaging software
- A database server

If you plan to use scripts to configure and deploy your application server, you must also install appropriate software for the scripts. For example, if you use Ant scripts developed by Oracle Retail, you should install Apache Ant 1.6.2 or later, or use the version of Apache Ant bundled with WebSphere.

[Table 2–2](#) lists software products in the required categories which have been tested for this release in at least one configuration.

Table 2–2 Supported Software

Required Item	Supported Products
Client Browser	Internet Explorer 6
Operating System	IRES V2 (SUSE Linux Enterprise Server 9)
Java environment	Java 2 Platform, Enterprise Edition (J2EE), version 1.3.x
Java Development Kit (JDK)	Version 1.4.2 or higher
Application Server	IBM WebSphere Application Server 5.1.1.3
Messaging Software	IBM WebSphere MQ Series 5.3, Fix Pack 11 or higher
Database	IBM DB2 v8.2, Fix Pack 1 or higher

Note: WebSphere Application Server, WebSphere MQ, and DB2 are separate IBM products which can be installed separately.

Oracle Retail Returns Management Installation

Oracle Retail recommends that configuration and setup specific to Oracle Retail Returns Management be done in the following order:

1. Database setup
2. JMS configuration
3. Application server configuration
4. Oracle Retail Returns Management deployment

Database Setup

Install and configure the database server before you deploy Oracle Retail Returns Management. See [Table 2-2](#) for compatible database systems.

Note: Oracle Retail recommends that you have a qualified DBA install and configure your database prior to installing Oracle Retail Returns Management database schemas. Oracle Retail Returns Management requires a user temporary tablespace. Your DBA needs to configure the user temporary tablespace so that the database user for Oracle Retail Returns Management has access to it.

After the database server is installed and a database is created, run the SQL scripts provided with the product to create the database tables and insert minimum configuration data. The scripts create the base Oracle Retail schema and Oracle Retail Returns Management schema.

After the database tables exist, you can run custom scripts to insert enterprise specific data, such as initial users and a set of stores. Typically, your developers and Oracle Retail professional services work together to develop these scripts. For information on loading the database, see "[Loading the Oracle Retail Returns Management Database](#)".

JMS Configuration

Your Oracle Retail Returns Management deployment requires a set of configured JMS queues on the JMS server. See [Table 4-6](#) for the list of JMS queues. The actual list of queues required for your deployment can vary depending on how you configure Oracle Retail Returns Management. Determine the required configuration for your deployment with your development team and configure the queues as appropriate for your application server. For more information, see [Chapter 5](#).

Application Server Configuration

Before deploying the Oracle Retail Returns Management application, the J2EE application server must be configured with the following appropriate artifacts:

- Data sources
- Connections to the JMS queues
- Security for user authentication
- Other artifacts as needed

For more information, see [Chapter 5](#).

Oracle Retail Returns Management Deployment

After completing the setup and configuration for the database, JMS server, and application server, the Oracle Retail Returns Management can be deployed as a standard J2EE application.

Unpacking Oracle Retail Returns Management from the Oracle Web Site

To unpack the application software:

1. From the Oracle web site, download the zip file containing the Oracle Retail Returns Management application into the installation directory of your choice.

Note: `<ORRM_INSTALL>` is used as the name of the installation directory you chose, in the rest of this installation guide.

2. Change to the installation directory.

```
cd <ORRM_INSTALL>
```

3. Unzip the downloaded files into the current directory.

4. List the contents of the directory. You will see the following two files:

- `returnsmgmtInstall.jar`
- `returnsmgmtDocs.zip`

5. Unzip the documents files under the current directory or any directory of your choice.

6. Extract the files in the `.jar` file.

```
jar -xvf returnsmgmtInstall.jar
```

7. List the contents of the directory. The list includes the following:

- `returnsmgmt.ear`
J2EE archive to be deployed in the application server
- `returnsmgmtDBInstall.jar`
Scripts for loading the Returns Management database
- `appservers`
Directory that contains the configuration files for different application servers. Only WebSphere Application Server is fully tested and supported in this release.

Loading the Oracle Retail Returns Management Database

To load the database:

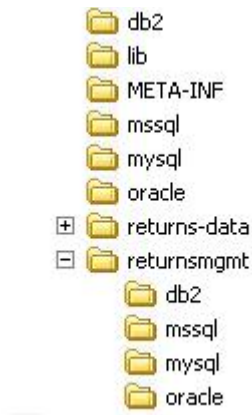
1. Create a directory for the database scripts. `<ORRM_DBINSTALL>` is used as the name of that directory in the rest of this installation guide.
2. Copy and extract `returnsmgmtDBInstall.jar` to the directory `<ORRM_DBINSTALL>`.

```
cd <ORRM_DBINSTALL>
```

```
jar -xvf returnsmgmtDBInstall.jar
```


This produces a directory tree under `<ORRM_DBINSTALL>` as shown in [Figure 2-1](#).

Figure 2-1 Directory Tree Created for Database Load



The sub-directories contain SQL script files that are oriented towards specific databases like DB2, Oracle, MySQL, and Microsoft SQL Server. The files named FinalXXX.sql are the ones executed. DB2 is supported for this release. The other databases have not been tested and are not supported for this release.

3. Create and initially-load the Oracle Retail Returns Management database. The easiest way to do this is by running an (Apache) Ant script in the build.xml file, found in the top-level directory above.

Open a command shell, and change to the `<ORRM_DBINSTALL>` directory.

Edit the `db.properties` file in this directory, to point to the database where Returns Management data is to be loaded. [Example 2-1](#) shows what your `db.properties` file might look like for a DB2 database.

Example 2-1 Properties for Pre-loading Returns Management Database

```

# Replace values below with the user ID and password for database access.
db.user=MyDBUserID
db.password=MyDBPassword
db.autocommit=true

# Replace DB_HOST_NAME and DB_NAME and uncomment the entire block
# starting with db.product and ending with db.jdbc-url
# for the db that you want to configure.

# Comment out if you DO NOT want to load any third party sql (e.g for Quartz or
crystal)
# true/false doesn't matter to ANT; either the property is defined or not.
db.product.3rdparty=true

#####
# Database-specific properties.
# Uncomment entire block, and replace DB_HOST_NAME, DB_NAME
# for the database to be used.
#####

# Oracle
#db.product=oracle

```

```
#db.version=10g
#db.jdbc-driver=oracle.jdbc.driver.OracleDriver
#db.jdbc-jar=lib/ojdbc14.jar
#db.jdbc-url=jdbc:oracle:thin:@DB_HOST_NAME:1521:DB_NAME

# DB2
db.product=db2
db.version=8.1.0
db.jdbc-driver=com.ibm.db2.jcc.DB2Driver
db.jdbc-jar=lib/db2jcc.jar
db.jdbc-jar2=lib/db2jcc_license_cisuz.jar
db.jdbc-jar3=lib/db2jcc_license_cu.jar
db.jdbc-url=jdbc:db2://DB_HOST_NAME:50000/DB_NAME

# MS-SqlServer
#db.product=mssql
#db.jdbc-driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
#db.jdbc-jar=lib/mssqlserver.jar
#db.jdbc-jar2=lib/msbase.jar
#db.jdbc-jar3=lib/msutil.jar
:
:
```

4. Make sure your database server is running and an empty database (*DB_NAME* in [Example 2-1](#)) has already been created. Then run the following command:

```
ant load_sql
```

This creates necessary database tables, and loads them with initial (seed) data.

Configuring WebSphere MQ

Oracle Retail Returns Management is packaged as an enterprise application archive (`ear`) file. A successful deployment requires an application server that enables Oracle Retail Returns Management to access a variety of external resources. You must configure the application server appropriately for these resources.

Perform the following steps before deploying Oracle Retail Returns Management in the WebSphere Application Server:

- Install WebSphere MQ, WebSphere Application Server (WAS), and fix packs
- Create a queue manager and queues
- Create the custom security jar file
- Install and edit the necessary properties and jar files
- Configure WebSphere Application Server, using either JACL scripts or the Administrative Console

The first two steps are described in this chapter. After performing these steps, see [Chapter 4](#) for information on completing the remaining steps.

Installing WebSphere MQ, WebSphere Application Server, and Fix Packs

Oracle Retail recommends that you use a standalone messaging service (WebSphere MQ) rather than the messaging service integrated with the WebSphere Application Server. Oracle Retail also recommends that you install the messaging service first so that the application server can be configured to connect to the external messaging service.

For these installations, consult your WebSphere documentation. For an example of the steps to use to do these installations, see [Appendix A](#).

The following instructions use the `mqm` user which should be created as part of the WebSphere installation.

Configuring WebSphere MQ

The configuration of WebSphere MQ requires two procedures:

- Defining a queue manager. In a UNIX environment, this is done using the command line.
- Defining the necessary queues. In a UNIX environment, this can be done using the command line or can be done using a script.

Note: It is assumed that no other MQ queue managers have been created on this host. Therefore, the queue manager created in the following steps is the default queue manager and can use the default TCP port 1414.

To create the MQ queue manager:

1. Log in as the mqm user.
2. Change to the MQ installation bin directory, for example `/opt/mqm/bin`, if this directory is not in the `$PATH`:

```
<shell prompt> crtmqm -q rm.queue.manager
```

3. Start the queue manager and queue listener using the following commands:

```
<shell prompt> strmqm rm.queue.manager
```

```
<shell prompt> runmqclsr -m rm.queue.manager -p 1414 -t tcp &
```

Creating the MQ Queues with an Input File

To create the MQ queues using a script, run `runmqsc` with an input file of commands:

```
<shell prompt> runmqsc rm.queue.manager <
<ORRM_INSTALL>/appservers/was/createq.dat
```

createq.dat Input File

[Example 3–1](#) shows the input file used to create the necessary MQ queues:

Example 3–1 createq.dat Input File

```
DEFINE QLOCAL (EJ.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (EJOURNALIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(EJ.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (TASKEXECUTION.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (TASKEXECUTION) USAGE (NORMAL) DEFPSIST(YES)
BOQNAME(TASKEXECUTION.ERROR) BOTHRESH(1)
DEFINE QLOCAL (FINALRESULT) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (JOURNALINGMESSAGE) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (SIGCAP.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (SIGNATURECAPTURE) USAGE (NORMAL) DEFPSIST(YES)
BOQNAME(SIGCAP.ERROR) BOTHRESH(1)
DEFINE QLOCAL (POSLOG.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (POSLOG) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(POSLOG.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (STORE_<XXXXX>) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (PARAMETERIMPORT) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (ACKNOWLEDGEMENT) USAGE (NORMAL) DEFPSIST(YES)
```

Note: The line

```
DEFINE QLOCAL (STORE_<XXXXX>) USAGE (NORMAL) DEFPSIST(YES)
```

is repeated several times in the above file. `<XXXXX>` stands for store numbers initially loaded into the database for test purposes.

Creating the MQ Queues with a Command Line

The following steps are an example of creating the MQ queues using a command line.
To create individual MQ queues using the command line:

1. Create the MQ command line shell using the following command:

```
<shell prompt> runmqsc rm.queue.manager
```

2. Begin by creating an error queue (backout queue with a backout threshold) for each queue that is required. Use the following command to create an error queue:

```
DEFINE QLOCAL (<name of queue>) USAGE (NORMAL) DEFPSIST (YES)
      BOQNAME (<name of backout queue>) BOTHRESH (<backout
      threshold>)
```

Note: Use all uppercase letters for a queue name.

Value	Description
<name of queue>	Name of the queue, for example, EJOURNALIMPORT
<name of backout queue>	Name of the error queue, for example, EJ.ERROR
<backout threshold>	Number of times MQ attempts to send a message before moving it to the backout queue

3. Repeat step 2 for each required error queue.

4. Create a queue using the following command:

```
DEFINE QLOCAL (<name of queue>) USAGE (NORMAL) DEFPSIST (YES)
```

[Example 3–2](#) shows an example of the output displayed by the command:

Example 3–2 Output Displayed When Creating a Queue

```
<shell prompt> runmqsc rm.queue.manager
5724-B41 (C) Copyright IBM Corp. 1994, 2002. ALL RIGHTS
RESERVED.
Starting MQSC for queue manager rm.queue.manager

DEFINE QLOCAL(<name of queue>) USAGE (NORMAL) DEFPSIST(YES)
      1: DEFINE QLOCAL(EJOURNALIMPORT) USAGE (NORMAL)
DEFPSIST(YES)
AMQ8006: WebSphere MQ queue created
```

5. Repeat step 4 for each queue.

Useful WebSphere MQ Commands

[Table 3–1](#) lists the commands used to manage the queue manager.

Table 3–1 Useful MQ Commands

Purpose	Command
To end the queue manager	endmqm <QUEUE_MANAGER> Use the -i flag to end the queue manager immediately.

Table 3–1 (Cont.) Useful MQ Commands

Purpose	Command
To access the queue manager command shell	<code>runmqsc <QUEUE_MANAGER></code>

Table 3–2 lists the commands used to manage queues.

Table 3–2 Useful Queue Manager Shell (runmqsc) Commands

Purpose	Command
To create a local queue	<code>define qlocal (<QUEUE_NAME>) usage(NORMAL)</code>
To delete a local queue	<code>delete qlocal (<QUEUE_NAME>)</code>
To list all local queues	<code>display qlocal (*)</code>
To list all channels	<code>display channel (*)</code>
To exit the shell	<code>end</code>
To display information about a queue	<code>DISPLAY QLOCAL (<QUEUE_NAME>)</code>
To clear messages from a queue	<code>CLEAR QLOCAL (<QUEUE_NAME>)</code>

Configuring WebSphere Application Server

This chapter covers the steps required to configure the WebSphere Application Server. Before performing the steps in this chapter, you must complete the configuration of WebSphere MQ as described in [Chapter 3](#).

Required Resources

[Table 4–1](#) summarizes the resources required for Oracle Retail Returns Management.

Table 4–1 Required Resources

Type	JNDI Name	Description
WebSphere MQ Queue Connection Factory	jms/QueueConnectionFactory	The Queue Connection Factory
WebSphere MQ Queue Destinations	For a list of the required queue destinations, see Table 4–6 .	
JDBC Providers	Not applicable	XA provider
	Not applicable	Non-XA provider
JDBC Data Sources	jdbc/DataSource	Data source for the XA provider
	jdbc/CrystalClearDS	IBM WebSphere version 4.0 - compatible XA data source
	jdbc/Other	Non-transactional data source for the non-XA JDBC provider
Listener Ports	For a list of required listener ports, see Table 4–15 .	
Mail Session	mail/Mail	For the application to send e-mail messages about job completion

Configuring WebSphere Application Server

An application server makes necessary external resources available to an application. Each of these external resources must be defined. You can use a Java command line (JACL) script or do this manually using the WebSphere Application Server Administrative Console:

- To use a JACL script, see "[Configuring WebSphere Using JACL Scripts](#)".
- To do the configuration manually, see "[Configuring WebSphere Using the Administrative Console](#)".

Configuring WebSphere Using JACL Scripts

Oracle Retail provides a `build.xml` file to automate some of the configuration of resources in WebSphere. This file specifies ant targets for Oracle Retail Returns Management deployment; most of these targets execute Java Command Language (JACL) scripts. The primary ant target is `configure`.

The JACL scripts reconfigure many settings you have already established. For example, if your host is running headless and you have already configured `xvfb`, make sure your prior settings are still present after running the JACL scripts or make necessary adjustments to reconfigure `xvfb`.

Editing the `was.properties` File

The `<ORRM_INSTALL>/appservers/was/was.properties` file contains the settings for the ant scripts which configure WAS and deploy Oracle Retail Returns Management. Edit it to make certain that the property values are correct. [Example 4-1](#) shows the settings for the WAS properties.

Example 4-1 `was.properties` File

```
# Used by the ws_ant (WebSphere ant) build task, to configure deployment of
# Returns Management, in WebSphere App Server (WAS) 5.x.

jars.dir=<RM_DBINSTALL>/lib
temp.dir=/tmp

# WAS - database connection settings.
# Modify the DB_XXX values as appropriate
db.user=DB_USER
db.password=DB_PASSWORD
db.jdbc-url=jdbc:db2://DB_HOST:50000/DB_NAME
db.jdbc-driver=com.ibm.db2.jcc.DB2Driver

# WAS - WebSphere MQ connection settings
mq.install.path=/opt/mqm
mq.user=mqm
mq.passwd=mqm
mq.queue.manager=rm.queue.manager

# WAS installation root
was.home=/opt/WebSphere/AppServer
# Location of WAS' Java libraries (.jars)
was.libs=${was.home}/lib
task.lib=${was.libs}/wsanttasks.jar
app.name>ReturnsManagement
# Full path of .ear to be deployed
ear.file=<ORRM_INSTALL>/returnsmgmt.ear
node.name=<your WebSphere node name>
server.name=server1
server.user=pos
server.passwd=pos
# Protocol by which wsadmin communicates with WAS
server.connection=SOAP
# Classname of "Custom User Registry", for WAS security
registry.class=com._360commerce.commerceservices.security.websphere.COUserRegistry
jvmArgs=-Duser.timezone=America/Chicago
# Location of .jar containing database driver
db.install.path=${was.libs}
data.source=jdbc/DataSource
```



```
# LDAP Settings for Sun One Directory Server
```

```
:
:
```

Executing the ws_ant Targets

To execute the ws_ant targets:

1. Start the WebSphere Application Server.
2. Change to the `<ORRM_INSTALL>/appservers/was` directory.
3. Execute the ant target.

```
ws_ant configure
```

[Table 4-2](#) describes the series of sub-targets executed by `ws_ant configure`.

4. Stop and restart WebSphere Application Server.

Table 4-2 Sub-targets for ws_ant configure

Sub-target	Description
create_jms	This creates the JMS queue destinations and message listener ports.
create_datasource	This creates the JDBC providers and data sources to connect to the database defined in the <code>was.properties</code> file. Test each data source to verify its connection to the database. Perform the same test for the XA and non-XA provider data sources.
create_mail_session	This sets up the mail provider, which sends automated emails.
create_jvm_args	This sets up the command-line arguments for the Java Virtual Machine (JVM) that executes to start the WebSphere Application Server instance.
create_url	This creates the URL providers and URLs for the application.
copy_jars copy_props	These copy the required <code>jar</code> and <code>properties</code> files into the WebSphere Application Server installation.
enable_security	This sets up a custom registry class to be used for security and enables WebSphere Application Server global security.
disable_internal_jms	This disables the internal JMS server.
create_environment_variables	This creates the WebSphere Application Server environment variables on the server referenced by other JACL scripts.

Completing the Configuration

The JACL scripts do about 95% of the WAS configuration. To manually complete the configuration process:

1. Open the WAS Administration console in a Web browser. The URL is `http://<WAS host>:9090/admin`
Because WAS security is now enabled, you are prompted for a user ID and password. Enter `pos` for the user ID and `pos` for the password.
2. From the console, select **Environment**, and then select **Manage WebSphere Variables**.

3. Verify the values for the node-scope and server-scope variables.
If needed, edit the values set by the JACL scripts. In particular, ensure that DB_INSTALL_ROOT and MQ_INSTALL_ROOT are accurate for your installation.
4. Select **Resources** and then select **JDBC Providers**.
5. Select the Data sources and use the test connection feature to ensure that there is connectivity between WAS and your database. If you see an error message, fix the problem and test the connection again. Repeat this step for the data source of the non-XA provider.
6. Log out of the administration console. Restart WAS.

Configuring WebSphere Using the Administrative Console

Note: Perform the following procedures only if you have not already configured WebSphere using the JACL scripts described in the previous section.

This section describes the process of configuring the external resources using the WebSphere Application Server Administrative Console.

Creating the Custom Security Jar File

Build the custom security jar file (security-360.jar) from the command line of the operating system shell. If your installation uses LDAP or some other kind of authentication security or you have a pre-built security-360.jar, you do not need to perform this procedure.

To build the customer security jar file:

1. Change to the directory `<ORRM_INSTALL>/appservers/was`.
2. Enter the following command:

```
ws_ant build_security_jar
```

Note: ws_ant is a version of the ant utility supplied with WebSphere. It is found in the WAS bin directory, for example, `/opt/WebSphere/AppServer/bin`.

Installing and Editing Necessary Files

Oracle Retail Returns Management requires a series of properties and jar files to be included in the WebSphere Application Server installation.

To install and configure the necessary files:

1. Copy the log4jxxx.jar file to `/opt/WebSphere/AppServer/lib`.
2. Copy the `<ORRM_INSTALL>/appservers/was/log4j.properties` file to `/opt/WebSphere/AppServer/properties`.
3. Edit log4j.properties to set the level of log messaging that you want.
4. Copy security-360.jar to `/opt/WebSphere/AppServer/lib`.
5. Copy `<ORRM_INSTALL>/appservers/was/quartz.properties` to `/opt/WebSphere/AppServer/properties`.

6. Edit `quartz.properties` to set the `org.quartz.jobStore.driverDelegateClass` property correctly for your database type.
7. Copy the WebSphere MQ authorization file `<ORRM_INSTALL>/appservers/was/integral-jms-authorizations.xml` to `/opt/WebSphere/AppServer/config/cells/<node name>`.
8. Copy the database driver and license files (`db2jcc.jar`, `db2jcc_license_cisuz.jar`, and `db2jcc_license_cu.jar`) from your DB2 version 8.2 installation to `/opt/WebSphere/AppServer/lib`.
9. Copy `<ORRM_INSTALL>/appservers/was/FixedIsolationDataStoreHelper.jar` to `/opt/WebSphere/AppServers/lib`.

Setting up the WebSphere MQ JMS Provider

This process includes setting up the J2C authentication data, creating the queue connection factory, and defining the queue destinations.

To configure the Java 2 Connector Architecture (J2C) aliases:

1. In the left pane of the WebSphere Administrative Console, select Security.
2. Select JAAS Configuration.
3. Select J2C Authentication Data.
4. In center pane, click **New**.
5. Enter data in the available fields. [Table 4-3](#) shows sample data.
6. Click **OK**.

Table 4-3 Sample J2C Authentication Data

Field	Value
Alias	AuthDataAliasJMS_QCF
User ID	pos
Password	pos
Description	(optional)

Add a WebSphere Environment variable for the MQ installation root.

To add the WebSphere Environment variable:

1. In the left pane of the WebSphere Administrative Console, select Environment.
2. Select Manage WebSphere Variables.
3. In the right pane, add or update the variable **MQ_INSTALL_ROOT**. Set its value to the root directory of the MQ software installation. For example, `/opt/mqm`.

Define the queue connection factory with the JNDI name of `jms/QueueConnectionFactory`.

To configure the WebSphere MQ JMS Provider Queue Connection Factory:

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select WebSphere MQ JMS Provider.
3. In the center pane, scroll to the Additional Properties section.
4. Select WebSphere MQ Queue Connection Factories.
5. Select **New**.
6. Enter and select data for the available fields. [Table 4–4](#) shows sample data.

Table 4–4 Sample Queue Connection Factory Data

Field	Value
Name	QueueConnectionFactory
JNDI Name	jms/QueueConnectionFactory
Description	(optional)
Category	Returns Management
Component-managed authentication Alias	(none)
container-managed Authentication Alias	(none)
Mapping-Configuration Alias	(none)
Queue Manager	rm.queue.manager
Host	<your MQ host>
Port	1414
Channel	SYSTEM.DEF.SVRCONN
Transport Type	CLIENT
Model Queue Definition	(optional)
Client ID	(optional)
CCSID	(optional)
Message Retention	Checked
XA Enabled	Checked

7. Click **OK**. The Queue Connection Factories page opens.
8. Select QueueConnectionFactory.
9. In the right pane, scroll to the Additional Properties section.
10. Enter and select data for the available fields. For sample data, see [Figure 4–1](#).
11. Click **OK**.

Figure 4–1 Connection Pool Configuration

Configuration		
General Properties		
Scope	cells:<serverName>.nodes:<serverName>.servers:server1	The scope of the configured resource. This value indicates the configuration location for the configuration file.
Connection Timeout	360 seconds	Interval, in seconds, after which a connection request times out and a <code>ConnectionWaitTimeoutException</code> is thrown.
Max Connections	30 connections	The maximum number of <code>ManagedConnections</code> that can be created in this pool.
Min Connections	1 connections	The minimum number of <code>ManagedConnections</code> that should be maintained.
Reap Time	60 seconds	Interval, in seconds, between runs of the pool maintenance thread.
Unused Timeout	1800 seconds	Interval, in seconds, after which an unused connection is discarded by the pool maintenance thread.
Aged Timeout	1800 seconds	Interval, in seconds, after which an unused, aged connection is discarded (regardless of recent usage activity) by the pool maintenance thread.
Purge Policy	EntirePool	Specifies how to purge connections when a "stale connection" or "fatal connection error" is detected.
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>		

To define MQ queue destinations:

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select WebSphere MQ JMS Provider.
3. Select WebSphere MQ Queue Destinations.
4. Select **New**.
5. Enter and select data for the available fields. [Table 4–5](#) shows sample data for the `EJournalImport` queue. Use all capital letters for the base queue name (the name in MQ). Ensure that each store has a queue defined. The default queue mask for a store is `store_<1>` where `<1>` is the store number.
6. Click **OK**.
7. Repeat for each queue. See [Table 4–6](#) for a list of required queues.

Table 4–5 Sample MQ Queue Destination Data for the `EJournalImport` Queue

Field	Value
Name	EJournalImport
JNDI Name	jms/EJournalImport
Description	(optional)
Category	(optional)
Persistence	QUEUE DEFINED
Priority	QUEUE DEFINED
Specified Priority	0
Expiry	SPECIFIED

Table 4–5 (Cont.) Sample MQ Queue Destination Data for the EJournalImport Queue

Field	Value
Specified Expiry	60000
Base Queue Name	EJOURNALIMPORT
Base Queue Manager	rm.queue.manager
CCSID	(optional)
Native Encoding	Unchecked
Integer Encoding	Normal
Decimal Encoding	Normal
Floating Point Encoding	IEEENormal
Target Client	JMS
Queue Manager Host	<your MQ host>
Queue Manager Port	1414
Server Connection Channel Name	SYSTEM.DEF.SRVCONN
UserID	Windows: {leave blank} Unix: MQ userid (usually mqm)
Password	Windows: {leave blank} Unix: MQ password

Table 4–6 lists the queues that are required for Oracle Retail Returns Management.

Table 4–6 Required Queue Destinations

Name	JNDI Name	Queue Name (in WebSphere MQ)
EJournalImport	jms/EJournalImport	EJOURNALIMPORT
POSLog	jms/POSLog	POSLOG
FinalResult	jms/FinalResult	FINALRESULT
TaskExecution	jms/TaskExecution	TASKEXECUTION
SignatureCapture	jms/SignatureCapture	SIGNATURECAPTURE
Acknowledgement	jms/Acknowledgement	ACKNOWLEDGEMENT
ParameterImport	jms/ParameterImport	PARAMETERIMPORT
Store_XXXXX*	jms/store_XXXXX	STORE_XXXXX
JournalingMessage	jms/JournalingMessage	JOURNALINGMESSAGE
*XXXXXX represents the store number. Use the format shown to define the store queues.		

Setting up the JDBC Providers

Setting up the JDBC providers includes setting up the J2C authentication data aliases and then defining the three providers: the XA provider, the non-XA provider, and the CrystalClear data source provider.

To set up the J2C authentication data aliases:

1. In the left pane of the WebSphere Administrative Console, select Security.
2. Select JAAS Configuration.

3. Select J2C Authentication Data.
4. Select **New**.
5. Enter appropriate values for the XA data source authentication data. [Table 4-7](#) shows sample data.

Table 4-7 Sample J2C Authentication Data for the XA Data Source

Field	Value
Alias	XAAuthAlias
UserID	<your ID for the XA resource>
Password	<your password for the XA resource>
Description	(optional)

6. Click **OK**.
7. Select **New**.
8. Enter the appropriate values for the non-XA data source authentication data. [Table 4-8](#) shows sample data.
9. Click **OK**.

Table 4-8 Sample J2C Authentication Data for the Non-XA Data Source

Field	Value
Alias	AuthAlias
UserID	<your ID for the non-XA resource>
Password	<your password for the non-XA resource>
Description	(optional)

To define the JDBC XA provider:

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select JDBC Providers.
3. Select **New**.
4. From the menu, select the type of JDBC provider you are configuring.
5. Enter appropriate data in the fields. [Table 4-9](#) shows sample data.

Table 4-9 Sample JDBC XA Provider Data

Field	Value
Name	DB2 JDBC Driver (XA)
Description	DB2 JDBC Driver (XA)
Classpath	<Full path to DB2 driver>/db2jcc.jar
Native Library Path	Leave blank since we are using Type 4 JDBC Driver
Implementation Classpath	com.ibm.db2.jcc.DB2XADataSource

6. Click **OK**. The JDBC Providers page opens.

7. Select the JDBC provider you just created.
8. In the right pane, scroll to the Additional Properties section.
9. Select Data Sources.
10. Select **New**. The Data Sources New page opens.
11. Enter appropriate data. [Table 4–10](#) shows sample data.

Table 4–10 Sample JDBC XA Provider Data Source Data

Field	Value
Name	DB2 DataSource(XA)
JNDI Name	jdbc/DataSource
Container-managed persistence	Unchecked
Description	(optional)
Category	(optional)
Statement Cache Size	30
Datasource Helper Classname	com.ibm.swservices.database.helpers.FixedIsolationDB2DataStoreHelper
Component-managed Authentication Alias	<node name>/XAAuthAlias
Container-managed Authentication Alias	<node name>/XAAuthAlias
Mapping-Configuration Alias	DefaultPrincipalMapping

12. Click **OK**.
13. Select the data source you just configured.
14. In the right pane, scroll to the Additional Properties section.
15. Select Custom Properties.

Figure 4–2 Custom Properties for JDBC XA provider Data Source

[JDBC Providers](#) > [JdbcDriverDB2XA](#) > [Data Sources](#) > [DataSourceDB2XA](#) >

Custom Properties

Custom properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database. [i](#)

Total: 4
☐ Filter
☐ Preferences

<input type="checkbox"/>	Name ^	Value ^	Description ^	Required
<input type="checkbox"/>	databaseName	CODE	database	false
<input type="checkbox"/>	driverType	4	drivertype	false
<input type="checkbox"/>	serverName	yourserverName	db server	false
<input type="checkbox"/>	portNumber	50000	port	false

16. Update `databaseName` and `serverName` to the appropriate values for your installation. The `portNumber` is usually **50000** for DB2. The `driverType` should be **4**.
17. Click **OK**.
18. Return to the data sources page.
19. Click **Test Connection** to verify your settings.

To define the JDBC non-XA provider:

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select JDBC Providers.
3. Select **New**.
4. From the menu, select the type of JDBC provider you are configuring.
5. Enter appropriate data in the fields. [Table 4-11](#) shows sample data.

Table 4-11 Sample JDBC Non-XA Provider Data

Field	Value
Name	DB2 JDBC Driver
Description	DB2 JDBC Driver
Classpath	<Full path to DB2 driver>/db2jcc.jar
Native Library Path	Leave blank when using Type 4 JDBC Driver
Implementation Classname	com.ibm.db2.jcc.DB2ConnectionPoolDataSource

6. Click **OK**. The JDBC Providers page opens.
7. Select the JDBC provider you just created.
8. In the right pane, scroll to the Additional Properties section.
9. Choose Data Sources. The Data Sources page opens.
10. Click **New**. The new page appears.
11. Enter appropriate data. [Table 4-12](#) shows sample data.

Table 4-12 Sample JDBC Non-XA Provider Data Source Data

Field	Value
Name	DB2 DataSource
JNDI Name	jdbc/Other
Container-managed persistence	Unchecked
Description	(optional)
Category	(optional)
Statement Cache Size	30
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2DataStoreHelper

Table 4–12 (Cont.) Sample JDBC Non-XA Provider Data Source Data

Field	Value
Component-managed Authentication Alias	<host name> / AuthAlias
Container-managed Authentication Alias	<host name> / AuthAlias
Mapping-Configuration Alias	DefaultPrincipalMapping

12. Click **OK**. The Data Sources page opens.
13. Select the data source you just entered.
14. In the right pane, scroll to the Additional Properties section.
15. Select Custom Properties. See [Figure 4–2](#).
16. Update `databaseName` and `serverName` to the appropriate values for your installation. The `portNumber` is usually **50000** for DB2. The `driverType` should be **4**.
17. Click **OK**.
18. Return to the Data Sources page.
19. Click **Test Connection** to verify your configuration.

To set up the CrystalClear data source:

Note: This data source is not directly used by Oracle Retail Returns Management but is needed during system initialization.

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select JDBC Providers.
3. In the right pane, select **DB2 JDBC Driver (XA)**.
4. Choose Data Sources (version 4).
5. To create a data source for CrystalClear, click **New**.
6. Enter appropriate data. [Table 4–13](#) shows sample data.

Table 4–13 Sample JDBC XA Provider CrystalClear Data Source Data

Field	Value
Name	CrystalClearDS
JNDI Name	jdbc/CrystalClearDS
Container-managed persistence	Unchecked
Description	(optional)
Category	(optional)
Database Name	<database name>
Default User ID	<database user ID>

Table 4–13 (Cont.) Sample JDBC XA Provider CrystalClear Data Source Data

Field	Value
Default Password	<database password>

7. Click **OK**. The Data Sources page opens.
8. Choose the data source you just entered.
9. In the right pane, scroll to the Additional Properties section.
10. Choose Custom Properties. See [Figure 4–2](#).
11. Update `databaseName` and `serverName` to the appropriate values for your installation. The `portNumber` is usually **50000** for DB2. The `driverType` should be **4**.
12. Click **OK**.
13. Return to the Data Sources page.
14. Click **Test Connection** to verify your settings.

Setting up the Message Listener Service

This process involves setting up listener ports for each JMS queue.

To set up the message listener services:

1. In the left pane of the WebSphere Administrative Console, select Servers.
2. Select Applications Servers.
3. Select your server.
4. In the right pane, scroll to the Additional Properties section.
5. Select Message Listener Service.
6. Select Listener Ports.
7. Click **New**.
8. Enter appropriate values for the fields. [Table 4–14](#) shows sample data for one listener port.

Table 4–14 Sample Message Listener Service Data for the EJournalImportPort

Field	Value
Name	EJournalImportPort
Initial State	Started
Description	(optional)
Connection Factory JNDI Name	jms/QueueConnectionFactory
Destination JNDI Name	jms/EJournalImport
Maximum Sessions	1
Maximum Retries	2
Maximum Messages	1

9. Click **OK**.

10. Repeat for each required listener port. [Table 4–15](#) lists the required listener ports.

Table 4–15 Required Listener Ports

Name	Connection Factory JNDI Name	Destination JNDI Name
EJournalImportPort	jms/QueueConnectionFactory	jms/EJournalImport
POSLogPort	jms/QueueConnectionFactory	jms/POSLog
FinalResultPort	jms/QueueConnectionFactory	jms/FinalResult
SignatureCapturePort	jms/QueueConnectionFactory	jms/SignatureCapture
TaskExecutionPort	jms/QueueConnectionFactory	jms/TaskExecution
AcknowledgementPort	jms/QueueConnectionFactory	jms/Acknowledgement
ParameterImportPort	jms/QueueConnectionFactory	jms/ParameterImport
JournalingMessagePort	jms/QueueConnectionFactory	jms/JournalingMessage
TestStore44444MDBPort	jms/QueueConnectionFactory	jms/store_44444
TestStore66666MDBPort	jms/QueueConnectionFactory	jms/store_66666
TestStore77777MDBPort	jms/QueueConnectionFactory	jms/store_77777

Configuring the Mail Provider

To configure the mail provider:

1. In the left pane of the WebSphere Administrative Console, select Resources.
2. Select Mail Providers.
3. Select Built-in Mail Provider.
4. In the center pane, scroll to the Additional Properties section.
5. Select Mail Sessions.
6. Click **New**.
7. Enter appropriate data. [Table 4–16](#) shows sample data.

Table 4–16 Sample Mail Provider Data

Field	Value
Name	Returns Management Mail Session
JNDI Name	mail/Mail
Mail Transport Host	<server for connection when emails are sent>

Configuring Security

Configure the Custom User Registry for the application's built-in security.

To configure security:

1. In the left pane of the Administrative Console, select Security.
2. Select User Registries.
3. Select Custom.
4. Enter appropriate data. [Table 4–17](#) shows sample data.

Table 4–17 Sample Custom Registry Data

Field	Value
Server User ID	pos
Server User Password	pos
Custom Registry Classname	com._360commerce.commerceservices.security.websphere.COUserRegistry
Ignore Case	Checked

5. Click **Custom Properties**. [Table 4–18](#) shows sample data.

Table 4–18 Custom Properties Data

Field	Value
jdbc Driver	com.ibm.db2.jcc.DB2Driver
jdbcURL	jdbc:db2://<db host name>:<db port>/<db name>
jdbcUser	<database user ID>
jdbcPass	<database password>

6. In the left pane of the Administrative Console, select Security.
7. Select Global Security.
8. Enter appropriate data. [Table 4–19](#) shows sample data.

Note: You need to restart WebSphere Application Server to enable global security.

Table 4–19 Sample Global Security Data

Field	Data
Enabled	Checked
Enforce Java 2 Security	Unchecked
Use Domain Qualified User IDs	Unchecked
Active User Registry	Custom

Configuring a Headless Environment and Other JVM Options

Unless you have a graphical driver, you need to configure WebSphere to run in a headless environment.

You may also want to set a different time zone and increase the heap size.

To configure a headless environment:

1. Logon as root and run the xvfb (X11 Virtual Frame Buffer) process on your host machine.
2. In the left pane of the WebSphere Administrative Console, select Servers.
3. Select Application Servers.
4. Select your server.

5. Select Process Definition.
6. In the center pane, scroll to the Additional Properties section.
7. Select Java Virtual Machine.
8. Enter appropriate data. [Table 4–20](#) shows sample data.
9. Click **OK**.

Table 4–20 *Sample Java Virtual Machine Data*

Field	Value
Classpath	(optional)
Boot Classpath	(optional)
Verbose Class Loading	Unchecked
Verbose Garbage Collection	Unchecked
Verbose JNI	Unchecked
Initial Heap Size	0
Maximum Heap Size	256
Run HProf	Unchecked
HProf Arguments	(optional)
Debug Mode	Unchecked
Debug Arguments	(Default value)
Generic JVM Arguments	-Duser.timezone=<time zone> -Djava.awt.headless=true
Executable Jar file name	(optional)
Disable JIT	Unchecked
Operating System Name	(optional)

Deploying Oracle Retail Returns Management in WebSphere

Perform the following steps to deploy Oracle Retail Returns Management in the WebSphere Application Server:

- Install Oracle Retail Returns Management into WAS
- Start the application
- Import parameters

The Oracle Retail Returns Management application provides ant scripts to do the above tasks. The ant scripts, however, read information from the `<ORRM_INSTALL>/appservers/was/was.properties` file.

See ["Editing the was.properties File"](#) in [Chapter 4](#). Ensure that all values in this file are accurate for your installation.

Installing and Starting Oracle Retail Returns Management

To install the Oracle Retail Returns Management .ear file into WAS and start the application, execute the following command in the

`<ORRM_INSTALL>/appservers/was` directory:

```
ws_ant installApp startApp
```

Note: You do not always need to restart the WAS server after installing Oracle Retail Returns Management, but it often helps fix configuration issues, especially with regard to JMS queues.

If you generate or receive a new .ear file and you want to reinstall Oracle Retail Returns Management into WAS, execute the following command:

```
ws_ant redeploy
```

This command executes the ant targets `stopApp`, `uninstallApp`, `installApp`, and `startApp`.

Importing Initial Parameters

Note: You must import an initial set of parameters before you can use Oracle Retail Returns Management.

This section provides an overview of the procedures for importing an initial set of parameters. You can import the parameters through the Oracle Retail Returns Management user interface or by using an ant target. The procedure for importing parameters through the application user interface is described in more detail in the Oracle Retail Returns Management User Guide.

Importing Parameters Through the User Interface

To import the initial parameters through the user interface:

1. Open the Oracle Retail Returns Management application in a web browser. The address is `http://<your host name>:9080/returnsmanagement`
2. Log in to the application as user ID **pos** and password **pos**, or any other user ID that has full administrative rights.
3. Click the **Data Management** tab. The Available Imports screen is displayed.
4. To import the master parameter set, click the **File** link in the Import Parameters row. Follow the instructions to import `parameterset.xml` from the `<RM_DBINSTALL>` folder.
5. To import the initial set of Oracle Retail Returns Management application parameters, click the **File** link in the Import Application Parameters row. Follow the instructions to import `returnsmgmt.xml` from the `<ORRM_DBINSTALL>` folder.

Importing Parameters By Using an Ant Target

To import parameters using an ant target:

1. Change to the `<ORRM_DBINSTALL>` directory.
2. Edit the `db.properties` file. Ensure that the properties that affect parameter loading are properly set.
3. Execute the following command:

```
ant load_parameters
```

WebSphere MQ and Application Server Installation

This appendix provides examples of the steps used to install WebSphere MQ, WebSphere Application Server (WAS), and a fix pack. The file names and values used in this appendix may not match the data used for your installation.

Installing WebSphere MQ Version 5.3 and Service Pack 11

Oracle Retail recommends that you use a standalone messaging service (WebSphere MQ) rather than the messaging service integrated with the WebSphere Application Server. Oracle Retail also recommends that you install the messaging service first so that the application server can be configured to connect to the external messaging service.

To install WebSphere MQ:

1. On the WebSphere MQ installation CD, locate the installation directory. Change to that directory:

```
cd /media/cdrom/MQ.V5301
```

2. Execute the license using the following command:

```
./mqlicense_lnx.sh -text_only
```

3. Accept the license agreement.

4. Install MQ v5.3 by issuing commands in the following order:

a. Runtime `rpm -i MQSeriesRuntime-5.3.0-2.i386.rpm`

b. SDK `rpm -i MQSeriesSDK-5.3.0-2.i386.rpm`

c. Server `rpm -i MQSeriesServer-5.3.0-2.i386.rpm`

d. Java `rpm -i MQSeriesJava-5.3.0-2.i386.rpm`

5. Install the CSD11¹ fixpack by issuing commands in the following order:

a. Runtime `rpm -i MQSeriesRuntime-U802143-5.3.0-11.i386.rpm`

b. SDK `rpm -i MQSeriesSDK-U802143-5.3.0-11.i386.rpm`

c. Server `rpm -i MQSeriesServer-U802143-5.3.0-11.i386.rpm`

d. Java `rpm -i MQSeriesJava-U802143-5.3.0-11.i386.rpm`

¹ Cumulative Service Distribution 11, or CSD11, is also called Service Pack 11.

- Set the correct number of processors for your computer. See WebSphere MQ documentation for more information. If there are only one or two processors, use the following command:

```
setmqcap 4
```

Testing the WebSphere MQ Installation

An easy way to test whether the installation of WebSphere MQ was successful is to create and start a simple queue manager. You log in as user mqm or as a member of the mqm group. TESTQM is used in the following steps as the name of the test queue manager.

To create a test queue manager:

- Open the shell window and log in as user mqm.
- Set the following environment variable so that MQ commands are executed with backward compatible shared Linux libraries.

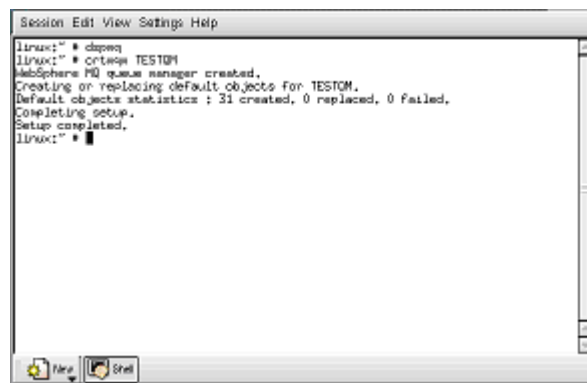
```
export LD_ASSUME_KERNEL=2.4.19
```

- Create a queue manager using the create queue manager command.

```
crtmqm <TESTQM>
```

Figure A-1 illustrates the expected output when a queue manager is created.

Figure A-1 Output of Queue Manager Creation



- Start the queue manager using the start command.

```
strmqm <TESTQM>
```

- Display the status of the queue manager using the display status command.

```
dspmq
```

The window should display the queue name with a status of running.

- Add JMS publish and subscribe queues to the queue manager.

```
runmqsc <TESTQM> </opt/mqm/java/bin/MQJMS_PSQ.mqsc>
```

- Start the queue manager listener.

```
runmqrlsr -t TCP -p 1414 -m <TESTQM> &
```

To delete this test queue manager:

1. Stop the queue manager.

```
endmqm <TESTQM>
```

This will also stop the queue manager listener. This make take a minute or two.

2. To verify the status of the queue manager, use the following command:

```
dspmqr <TESTQM>
```

Verify that the status of the queue manager is *Ended normally*.

3. Delete the queue manager.

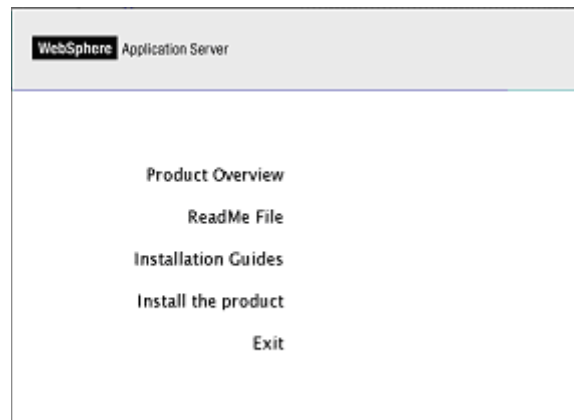
```
dlrmqm <TESTQM>
```

Installing the WebSphere Application Server Version 5.1

To install the WebSphere Application Server (WAS):

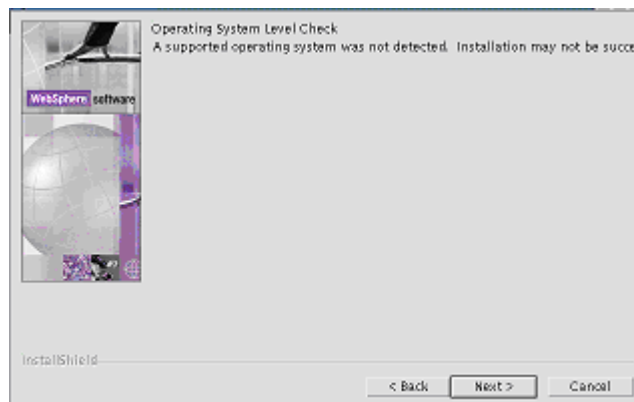
1. Mount the installation CD.
2. Locate the installation directory.
3. Execute the `launchPad.sh` script. The launchpad opens.

Figure A–2 WebSphere Application Server Launchpad Screen



4. Select **Install the product** to begin the installation process. The Installation Wizard opens.
5. If prompted for the language for the installation, select a language from the menu. Click **OK**.
6. Click **Next** to continue. The Software License Agreement screen opens.
7. To proceed with the installation, accept the software agreement.
8. Click **Next**. The Operating System Level Check screen opens.

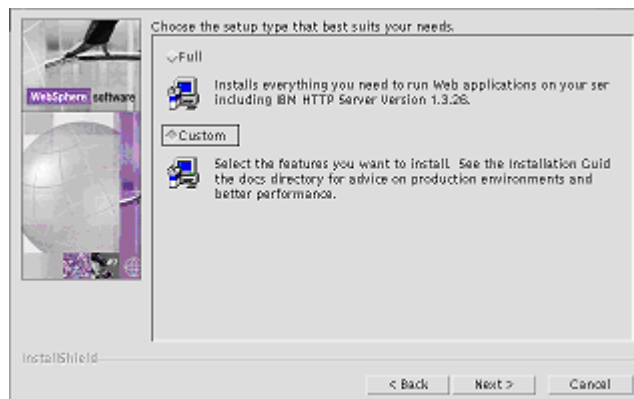
Figure A–3 Operating System Level Check Screen



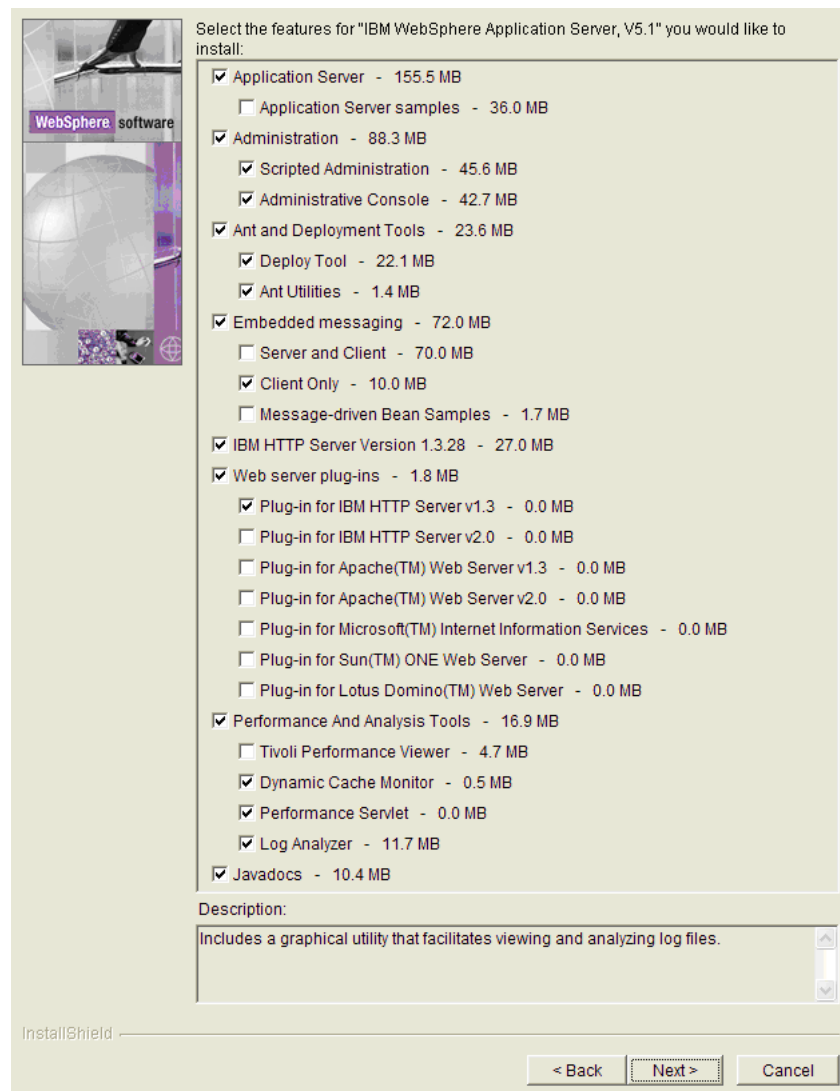
If an error occurs, as shown in this example, ignore the error.

9. Click **Next**. A screen that enables you to select the setup type opens.

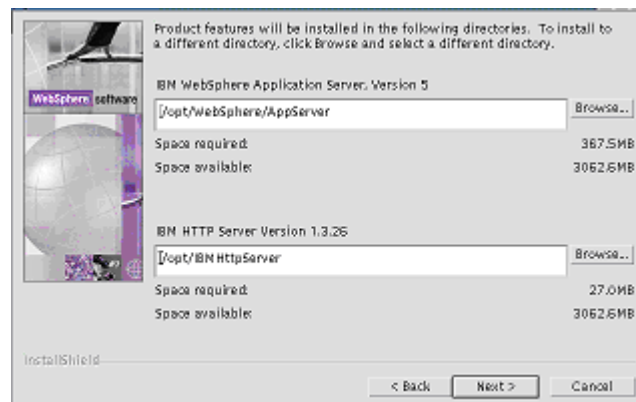
Figure A–4 Setup Type Selection Screen



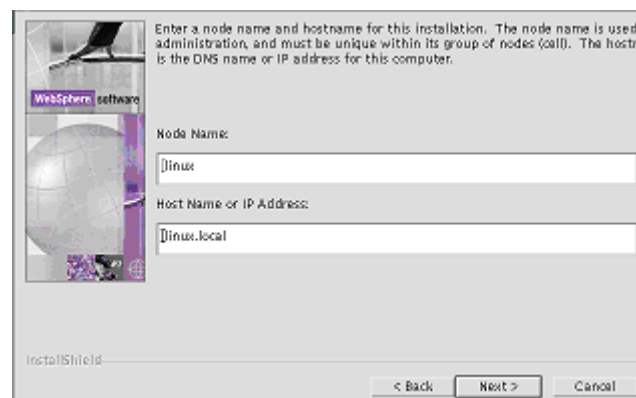
10. Check the box for **Custom**.
11. Click **Next**. A screen that enables you to select features opens.

Figure A-5 Feature Selection Screen

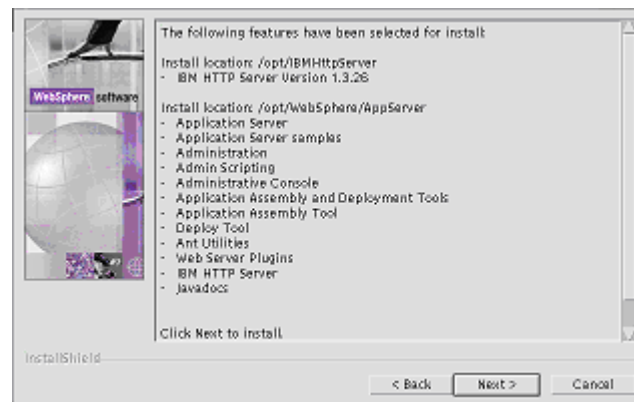
12. Uncheck the boxes with choices that relate to samples and performance analysis. Make sure to change the Embedded messaging option from *Server and Client* to *Client Only* because WebSphere MQ is already installed. The installer will display a validation error if this change is not made.
13. Click **Next**. A screen that enables you to select the installation directories opens.

Figure A–6 Installation Directories Selection Screen

14. Default directories are displayed in the text fields. To change them, follow the instructions at the top of the screen.
15. Click **Next**. A screen that enables you to select the node name and host name opens.

Figure A–7 Node Name and Host Name Screen

16. Enter a node name. This must be unique within its cell or group of nodes.
17. Enter a host name. This is the domain name system (DNS) name or IP address.
18. Click **Next**. A screen that lists the selected features opens.

Figure A–8 Selected Features List Screen

19. Review the list to make certain it is accurate and click **Next**. The installation begins and may take several minutes to complete. When completed, a screen that enables you to register the product opens.

Figure A–9 Product Registration Screen

20. Uncheck the **Register this product now** box.

21. Click **Next**. A screen that announces the successful installation opens.

Figure A–10 Installation Finish Screen

22. Click Finish.

Note: Oracle Retail recommends that you run the `backupConfig` script. If you run this with the server running, use the `-nostop` option. See the WebSphere documentation for further explanation.

Installing WebSphere Application Server Fix Packs

Before installing a fix pack, make sure the `JAVA_HOME` environment variable is either not set or set to point to the IBM JRE that is installed with the WebSphere Application Server. The latest fix pack, with download instructions, can be found on the IBM web site (www.ibm.com).

Before installing a fix pack, turn off the internal JMS server.

To turn off the JMS server:

1. Start the WebSphere Application Server:

```
/opt/WebSphere/AppServer/bin/startServer.sh Server1
```

2. Access the Administrative Console using the following URL:

```
http://<WAS hostname>:9090/admin
```

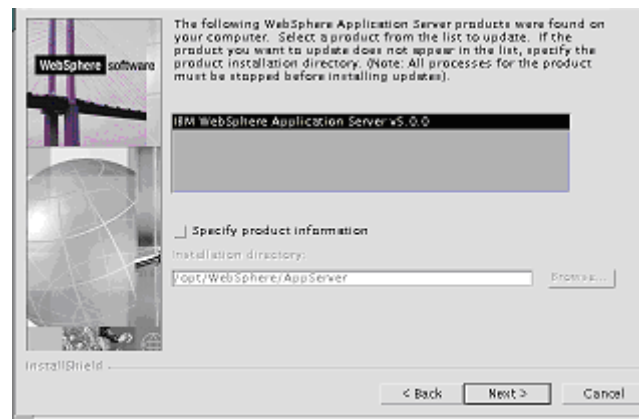
3. Select Servers by using the left navigation bar.
4. Select App Servers by using the left navigation bar.
5. Select server1.
6. Select Server Components under Additional Properties.
7. Select JMS Servers.
8. In the Internal JMS Server window, change the value of Initial State from *Started* to *Stopped*. Click **Apply**.
9. Save the configuration change.
10. Log out from the Administrative Console.
11. Stop the WebSphere Application Server:

```
/opt/WebSphere/AppServer/bin/stopServer.sh Server1
```

The following example shows how to install the WAS 5.1 Fixpack 1, which upgrades WAS from version 5.1.0 to 5.1.1. A very similar procedure must be used to upgrade from version 5.1.1 to version 5.1.1.3 or higher.

To install the fix pack:

1. Mount the installation CD.
2. Locate the installation directory.
3. Execute the `updateWizard` script. The Installation Wizard opens.
4. If prompted for the language, select a language from the menu.
5. Click **OK**. A screen opens to welcome you.
6. Click **Next**. A screen that enables you to select the product to be updated opens.

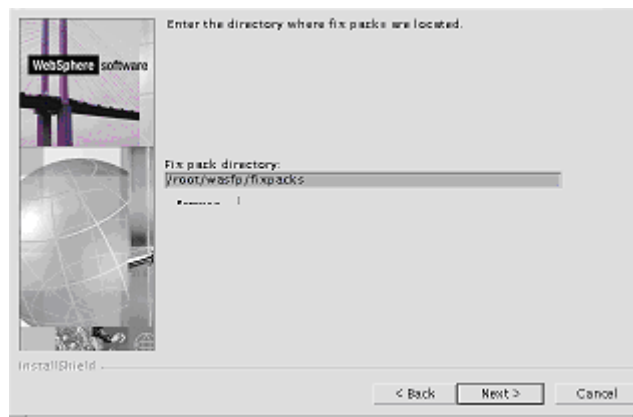
Figure A-11 Product Selection Screen

7. From the list, select the product to be updated. If it is not listed, locate it using the directions provided at the top of the screen. A screen that enables you to select installing or uninstalling fixes and fix packs opens.

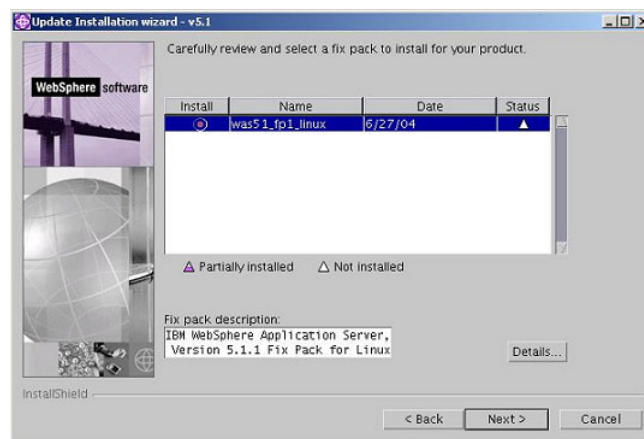
Note: As the information in the screen indicates, stop all processes for the selected product before you begin the installation of the fix pack.

Figure A-12 Fix Pack and Fix Installation and Uninstallation Screen

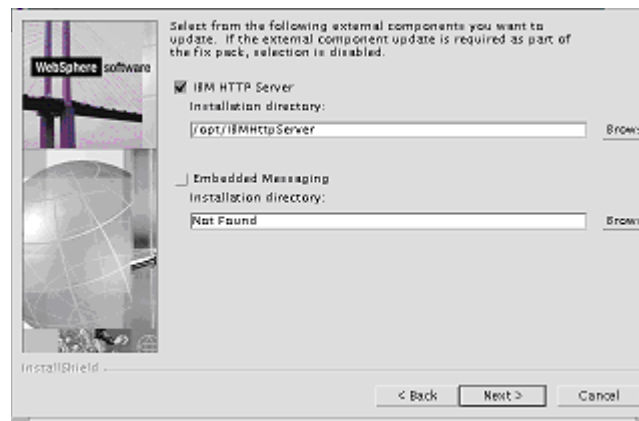
8. Check the appropriate button.
9. Click **Next**. A screen that enables you to enter the location of the fix packs opens.

Figure A-13 Fix Pack Directory Screen

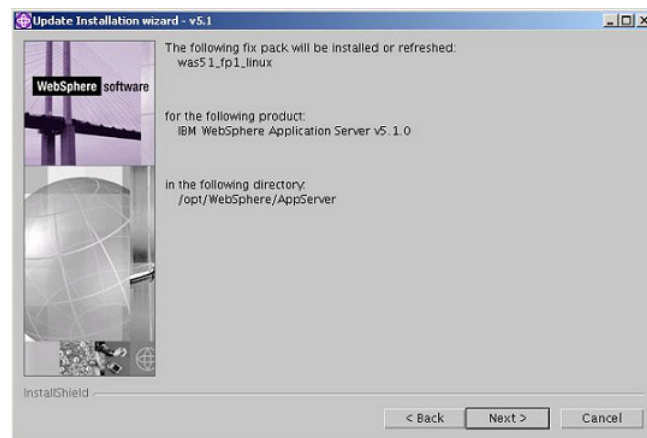
10. Enter the location of the fix pack.
11. Click **Next**. A screen that enables you to select the fix pack opens.

Figure A-14 Fix Pack Selection Screen

12. From the list, select the fix pack you want to install. The screen provides you with a description of the fix pack, the opportunity to access additional information by clicking **Details**, and an indication of the status of the fix pack (partially installed or not installed).
13. Click **Next**. A screen that enables you to select external components for updating opens.

Figure A-15 Select External Components Screen

14. Uncheck the boxes for IBM HTTP Server and embedded messaging.
15. Click **Next**. A screen that enables you to review your selections opens.

Figure A-16 Fix Pack Review Screen

16. Review the information on the screen and click **Next** to initiate the installation. A screen that indicates that the installation was successful opens.

Figure A-17 *Fix Pack Finish Screen*



17. Click **Finish**.

Returns Data Loader

The Oracle Retail Returns Management installation includes return ticket data, in XML format, which you can optionally load into the Returns Management database. There are several reasons why you would want to load this data:

- Once return tickets are loaded into the database, you can use the data to get familiar with those parts of the user interface that deal with return tickets, such as, searching for return tickets.
- Loading the return tickets acts as an end-to-end test of the Oracle Retail Returns Management software installation, from the web services interface up to the back-end database.
- The return ticket data is good sample data that can be used as a starting point for customization and experimentation with data relevant to your organization.

Using the Returns Data Loader

To use the returns data loader:

1. Change to the `<RM_DBINSTALL>` directory.
2. Edit the part of the `db.properties` file that deals with the returns data loader.

Set the values of the properties as needed. Replace the host name `My_RM_Server` shown in the following example.

```
#####
# Properties for Returns Seed Data Loading
#####

# the host name where the seed data should be loaded
dataLoader.host=My_RM_Server

# the port number where the seed data should be loaded
# WebSphere normally uses 9080
dataLoader.port=9080

# The URL shouldn't need to be modified unless the deployment location moves
dataLoader.url=http://${dataLoader.host}:${dataLoader.port}/retwebsvc/services/
ReturnsManager
```

3. Execute the following command:

```
ant load_returns_data
```

About 100 sample return requests and final result messages are sent to the Returns Management server. This step may take several minutes to complete.

This command sends some output to `DataTools.log` in the current directory. Ignore the warning message about attachment support, as the `DataLoader` does not need it to operate properly.

You can view the contents of the submitted XML messages in the `returns-data\tickets` directory. You can also modify the messages and resubmit them by repeating this step.