BEA WebLogic Commerce Server
BEA WebLogic Personalization Server

Deployment Guide

WebLogic Commerce Server 3.2
WebLogic Personalization Server 3.2
Document Edition 3.2.1
January 2001

Copyright

Restricted Rights Legend

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

Deployment Guide

Document Edition	Date	Software Version
3.2	December 2000	WebLogic Commerce Server and Personalization Server 3.2
3.2.1	January 2001	WebLogic Commerce Server and Personalization Server 3.2

Contents

About This Document

1. Setting Up Oracle for New Installations

2. Exporting Oracle Data Objects

3. Review Server Defaults

4. Deploying Web Applications

5. Starting the Server

Index

About This Document

When you install WebLogic Commerce Server and WebLogic Personalization Server, it is configured to use a Cloudscape database and the BulkLoader content-management system. When you are ready to configure WebLogic Commerce Server and WebLogic Personalization Server for developing, testing, and publishing your own e-commerce Web site and portals, follow the guidelines and procedures in this document.

This document describes the following tasks:

- Chapter 1, “Setting Up Oracle for New Installations,” which describes deploying database schemas for Oracle databases.
- Chapter 2, “Exporting Oracle Data Objects,” which describes exporting data from one environment to the next (for example, from a development environment to a staging environment).
- Chapter 3, “Review Server Defaults,” which describes reviewing default settings used to deploy WebLogic Commerce Server and WebLogic Personalization Server.
- Chapter 4, “Deploying Web Applications,” which describes deploying the WebLogic Commerce Server and WebLogic Personalization Server Web applications.
- Chapter 5, “Starting the Server,” which describes configuration and startup files for WebLogic Commerce Server and WebLogic Personalization Server.

What You Need to Know

This document is intended mainly for database administrators and Web site administrators who configure properties for WebLogic Server™ and WebLogic Commerce Server and WebLogic Personalization Server. It assumes a familiarity with the database management system that your organization uses with WebLogic Commerce Server and WebLogic Personalization Server as well as the WebLogic Server platform and J2EE specifications.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at http://e-docs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the WebLogic Commerce Server and WebLogic Personalization Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Commerce Server and WebLogic Personalization Server documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

The following BEA documents describe configuring WebLogic Server and WebLogic Commerce Server and WebLogic Personalization Server for optimal performance in a production environment:

- *BEA WebLogic Server Performance Tuning Guide*
- *BEA WebLogic Commerce Server and Personalization Server Performance Tuning Guide*

Contact Us!

Your feedback on the BEA WebLogic Commerce Server and WebLogic Personalization Server documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Commerce Server and WebLogic Personalization Server documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Commerce Server and WebLogic Personalization Server 3.2 release.

If you have any questions about this version of BEA WebLogic Commerce Server and WebLogic Personalization Server, or if you have problems installing and running BEA WebLogic Commerce Server and WebLogic Personalization Server, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes

- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*

Convention	Item
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Setting Up Oracle for New Installations

To use an Oracle database with a WebLogic Commerce Server and WebLogic Personalization Server installation, complete the following tasks:

Note: In a clustered environment, you must complete all of the following tasks for each node in the cluster.

Step 1: Add Statements to the SQL Scripts

Step 2: Create Tablespaces for Oracle

Step 3: Install the Oracle Client Software

Step 4: Create Oracle User Accounts

Step 5: Create the Schema for Oracle

Step 6: Rebuild Indexes

Step 7: Configure Properties Files and Environment Variables for Oracle

Step 8: Set Up the Example Portal for Oracle (optional)

Step 1: Add Statements to the SQL Scripts

WebLogic Commerce Server and WebLogic Personalization Server provides SQL scripts to create tablespaces. The following scripts are missing SQL statements:

- `WL_COMMERCE_HOME/db/oracle/create-tablespaces.sql` (for Oracle 8.1.5 databases, currently supported on Solaris systems)
- `WL_COMMERCE_HOME/db/oracle816/create-tablespaces.sql` (for Oracle 8.1.6 databases, currently supported in Windows systems)

In each of the scripts, there are two sections: Variable definition and command execution. Currently in the Variable definition portion of each file, you see something similar to the following:

```
UNDEFINE DATA_PATHNAME
UNDEFINE INDEX_PATHNAME

DEFINE DATA_PATHNAME=C:\ORACLE\ORADATA\PROD\
DEFINE INDEX_PATHNAME=D:\ORACLE\ORADATA\PROD\
REM DEFINE DATA_PATHNAME=/u01/oradata/PROD/
REM DEFINE INDEX_PATHNAME=/u11/ordata/PROD/
```

To Add Statements to the SQL Scripts

Do the following:

1. In the Variable definition of the SQL file for your database version, add the following statements:

```
UNDEFINE DATA_FILENAME
UNDEFINE INDEX_FILENAME
DEFINE DATA_FILENAME=WLCS_DATA01.DBF
DEFINE INDEX_FILENAME=WLCS_INDEX01.DBF
```

2. In the Command execution portion of the SQL file for your database version, replace the `"CREATE TABLESPACE"` command statements with the following:

```
PROMPT
PROMPT
PROMPT CREATING TABLESPACE WLCS_DATA
CREATE TABLESPACE WLCS_DATA DATAFILE
'&&DATA_PATHNAME&&DATA_FILENAME' SIZE 100M
;
PROMPT
PROMPT
PROMPT CREATING TABLESPACE WLCS_INDEX
CREATE TABLESPACE WLCS_INDEX DATAFILE
'&&INDEX_PATHNAME&&INDEX_FILENAME' SIZE 100M
;
```

Step 2: Create Tablespaces for Oracle

To separate its data from the Oracle data dictionary and from data that other applications use, WebLogic Commerce Server and WebLogic Personalization Server uses the following tablespaces:

- WLCS_DATA, which contains tables for WebLogic Commerce Server and WebLogic Personalization Server
- WLCS_INDEX, which contains indexes for WebLogic Commerce Server and WebLogic Personalization Server

For optimal performance, we recommend that you reduce I/O contentions by placing each tablespace on its own physical disk drive.

Creating WLCS_DATA and WLCS_INDEX

Note: In this document, `$WL_COMMERCE_HOME` refers to the directory into which you installed WebLogic Commerce Server and/or WebLogic Personalization Server and *`database-type`* refers to the type and version of RDBMS that you installed.

1. On the server that hosts the Oracle database, start SQL*Plus and log in as SYSTEM (or another user with create-tablespace privileges). You must complete this procedure on the local host; you cannot successfully create tablespaces from a remote session.
2. Make a backup copy of `$WL_COMMERCE_HOME/db/`*`database-type`*`/create-tablespaces.sql`

3. In a text editor, open `create-tablespaces.sql` and modify the pathnames for the `DATA_PATHNAME` and `INDEX_PATHNAME` variables to match your own directory path structures. For example, on a UNIX workstation two disks are mounted as `/usr1` and `/usr2` and the Oracle SID is `PROD`. We recommend the following pathnames:

   ```
   DEFINE DATA_PATHNAME=/usr1/oracle/oradata/PROD
   DEFINE INDEX_PATHNAME=/usr2/oracle/oradata/PROD
   ```

4. In the SQL*Plus session you started in step 1, enter the following command:

   ```
   @ $WL_COMMERCE_HOME/db/database-type/create-tablespaces.sql
   ```

Step 3: Install the Oracle Client Software

On the WebLogic Commerce Server and WebLogic Personalization Server host, use the Oracle installation program to install Oracle client software.

For information on the Oracle release that WebLogic Commerce Server and WebLogic Personalization Server supports for your platform type, refer to “Supported Platforms” in the *Release Notes*.

During the installation process, use the Oracle Net8 Assistant application to set up a new TNS service. The Net8 Assistant prompts you for the name or IP address of the computer that hosts the Oracle server software, the port number Oracle is listening on (usually 1521) and the SID name for the database. Test the connection with the database username and password that your database administrator assigned to you.

If you are using the WebLogic jDriver to connect to the Oracle database, refer to “Installing WebLogic jDriver for Oracle” on the WebLogic documentation site for information on setting up the appropriate jDriver.

Step 4: Create Oracle User Accounts

To simplify WebLogic Commerce Server and WebLogic Personalization Server administration, we recommend that you create a user account exclusively for the WebLogic Commerce Server and WebLogic Personalization Server schema. WebLogic Commerce Server and WebLogic Personalization Server include a script that creates a user account named WEBLOGIC and specifies a default tablespace for the account. Although you can modify the script to change the account name, we recommend that you do not; several other scripts assume that the account is named WEBLOGIC. If you change the account name, you must review each script for the use of the WEBLOGIC account name and modify it.

To create a user account:

1. From the host on which you installed the Oracle client software, start SQL*Plus and log in as SYSTEM (or any user with create-user privileges).
2. Enter the following command:

   ```
   @ $WL_COMMERCE_HOME/db/database-type/create-users.sql
   ```

Step 5: Create the Schema for Oracle

WebLogic Commerce Server and WebLogic Personalization Server includes a script, `create-all-oracle.sql`, that calls a series of other scripts to create the WebLogic Commerce Server and WebLogic Personalization Server schema and install sample data.

To run this script, do the following:

1. On the Oracle client host, log in to the WEBLOGIC user account by entering the following command from SQL*Plus:
   `connect WEBLOGIC/WEBLOGIC`@ORACLE_SID

2. Enter the following command:
   `@ $WL_COMMERCE_HOME/db/`*`database-type`*`/create-all-oracle.sql`

For a description of the tables, indexes, and constraints, see the following topics:

- "The Personalization Server Schema" in the *WebLogic Personalization Server Developer's Guide*
- "Product Catalog Database Schema" in the *Product Catalog Management*
- "Order Processing Database Schema" in the *Order Processing Package*

Step 6: Rebuild Indexes

The `create-all-oracle.sql` places the WebLogic Commerce Server and WebLogic Personalization Server index in the default tablespace (`WLCS_DATA`). To locate and rebuild the index; in the `WLCS_INDEX` tablespace, do the following:

1. From the Oracle client host, log in to the WEBLOGIC user account by entering the following command from SQL*Plus:
   `connect WEBLOGIC/WEBLOGIC`@ORACLE_SID
2. Enter the following command:
   `@ $WL_COMMERCE_HOME/db/`*`database-type`*`/rebuild-wlcs-indexes.sql`

Step 7: Configure Properties Files and Environment Variables for Oracle

To configure properties files and environment variables for Oracle, complete the following tasks:

- Edit the weblogic.properties File
- Edit the weblogiccommerce.properties File
- Update Environment Variables for the Server

Edit the weblogic.properties File

To edit this file, do the following:

1. In a text editor, open `$WL_COMMERCE_HOME/weblogic.properties`.
2. Deactivate the statement that begins with
   `weblogic.jdbc.connectionPool.commercePool=\`
   `url=jdbc:cloudscape:Commerce;`
   by prepending all of its lines (approximately 17) with a pound (#) character.
3. Activate the statement that loads an Oracle driver by removing the pound (#) character from the beginning of each line. This statement contains the following text:

```
#weblogic.jdbc.connectionPool.commercePool=\
#                    url=jdbc:weblogic:oracle,\
#       driver=weblogic.jdbc.oci.Driver,\
#       loginDelaySecs=0,\
#       initialCapacity=10,\
#       maxCapacity=20,\
#       capacityIncrement=1,\
#       allowShrinking=true,\
#       shrinkPeriodMins=15,\
#       testConnsOnReserve=true,\
#       testTable=dual,\
#       refreshMinutes=5,\
#

props=user=@ORACLE_USER@;password=@ORACLE_PASSWORD@;server=@ORACL
E_NET_SERVICE_NAME@,\
#             weblogic.t3.waitForConnection=true;\
#             weblogic.t3.waitSecondsForConnection=999999999999,\
#
weblogic.jts.waitSecondsForConnectionSecs=999999999999,\
#       verbose=false
```

4. In the statement that loads the Oracle driver (illustrated in the previous step), replace the variables that are enclosed in the commercial at (@) characters (such as `@ORACLE_USER@`) with the values that describe your environment. For example, replace `@ORACLE_USER@` with the user account name you create in “Step 4: Create Oracle User Accounts.”
5. Save your modifications to `$WL_COMMERCE_HOME/weblogic.properties`.

Edit the weblogiccommerce.properties File

If you are using the RDBMS security realm, you must change the `RDBMSRealm` settings in the `weblogiccommerce.properties` file to match the database type that stores the user information. If you are using LDAP or some other security realm, you can ignore these settings.

For information on the Oracle release that WebLogic Commerce Server and WebLogic Personalization Server supports for your platform type, refer to "Supported Platforms" in the *Release Notes*.

For example, Listing 1-1 shows a `weblogiccommerce.properties` file that specifies the WebLogic jDriver.

To change these settings, you must do **all** of the following:

1. Remove the comment tags from the lines that specify the database type you are using.
2. Change any placeholders in the lines. For example, you must change the string `@ORACLE_SERVER@` to the name of the Oracle server that hosts the WebLogic Commerce Server and WebLogic Personalization Server database.
3. Deactivate any other lines in this section by adding comment tags the beginning of each line.
4. Make sure that the value for `commerce.usermgmt.RDBMSRealm.dbUrl` matches the value for the `commerce.jdbc.pool.url` property (also located in `weblogiccommerce.properties`), which is used to establish JDBC connection pools.

   Also verify that the connection pool properties in `weblogic.properties` specify the same database type as these settings for the RDBMS realm.

Listing 1-1 Settings for the RDBMS Realm

```
# RDBMS Realm

#------Oracle thin Driver---------#

#commerce.usermgmt.RDBMSRealm.driver=oracle.jdbc.driver.OracleDriver
#commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:oracle:thin:ENTERPRISE:1205:COMMERCE
#commerce.usermgmt.RDBMSRealm.dbUser=@ORACLE_USER@
#commerce.usermgmt.RDBMSRealm.dbPassword=@ORACLE_PASSWORD@

#------WebLogic jDriver for Oracle 8.1.5---------#

commerce.usermgmt.RDBMSRealm.driver=weblogic.jdbc.oci.Driver
commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:weblogic:oracle
commerce.usermgmt.RDBMSRealm.dbServer=bread
commerce.usermgmt.RDBMSRealm.dbUser=WEBLOGIC
commerce.usermgmt.RDBMSRealm.dbPassword=WEBLOGIC

#-------------Cloudscape-----------------#

#commerce.usermgmt.RDBMSRealm.driver=COM.cloudscape.core.JDBCDriver
#commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:cloudscape:Commerce;create=true;autoco
mmit=false
#commerce.usermgmt.RDBMSRealm.dbUser=none
#commerce.usermgmt.RDBMSRealm.dbPassword=none
```

Update Environment Variables for the Server

To update environment variables, do the following:

1. In a text editor, open `%WL_COMMERCE_HOME%\bin\win32\set-environment.bat` (Windows) or `$WL_COMMERCE_HOME/bin/unix/set-environment.sh` (UNIX).

A subsequent section describes in detail each of the remaining steps.

2. Set `@ORACLE_HOME@`.

3. Specify the database.

4. Set variables for Oracle drivers.

Set @ORACLE_HOME@

After opening `set-environment`, find the line `WLCS_ORACLE_HOME=@ORACLE_HOME@` and substitute `@ORACLE_HOME@` with the absolute pathnames of the directory in which you installed the Oracle client software.

For example, Listing 1-2 shows a Windows environment in which the Oracle client software is installed in `D:\oracle`.

Listing 1-2 Set @ORACLE_HOME@

```
@ECHO OFF
REM
##############################################################################

REM #      (c) 2000 BEA SYSTEMS INC. All rights reserved
REM #
REM #      BEA Commerce Server variable setup script.
REM #
REM
##############################################################################


REM ----------- Set the following variables appropriately -----------
REM ----------- or define them as environment variables  -----------
REM ----------- in your system properties and comment out -----------
REM ----------- the next 4 lines.                         -----------

SET WL_COMMERCE_HOME=D:\WebLogicCommerceServer3.2
SET JDK_HOME=D:\jdk1.3

SET WLCS_ORACLE_HOME=D:\oracle

SET WEBLOGIC_HOME=D:\weblogic

REM -- Add WebLogic, CyberCash, and Taxware bin directories to the path --

SET PATH=%PATH%;%WEBLOGIC_HOME%\bin;%WL_COMMERCE_HOME%\eval\win32\CyberCash\bin;
%WL_COMMERCE_HOME%\eval\win32\Taxware\bin
```

The `WLCS_ORACLE_HOME` variable adds Oracle-client Java classes and libraries to the Java classpath and to the system path.

Specify the Database

Under the section titled
`-------- Specify which database to use --------,`
activate the line that sets the `DATABASE` variable to the database type that you are using and deactivate any other line. For example, Listing 1-3 shows a Windows environment that uses the WebLogic jDriver.

Listing 1-3 Specify the Database

```
REM ----------- Specify which database to use -----------

REM SET DATABASE=CLOUDSCAPE

REM DATABASE=ORACLE

SET DATABASE=ORACLE_OCI_815


if %DATABASE% EQU CLOUDSCAPE GOTO CLOUDSCAPE
if %DATABASE% EQU ORACLE GOTO ORACLE
if %DATABASE% EQU ORACLE_OCI_815 GOTO ORACLE_OCI_815

:CLOUDSCAPE

REM ----------- CLOUDSCAPE classes -----------
SET
DB_CLASSPATH=%WEBLOGIC_HOME%\eval\cloudscape\lib\cloudscape.jar;%WEBLOGIC_HOME%
\eval\cloudscape\lib\tools.jar;%WEBLOGIC_HOME%\eval\cloudscape\lib\client.jar

GOTO continue
```

Set Variables for Oracle Drivers

Make sure that the `:ORACLE_OCI_815` section sets the `DB_CLASSPATH` variable to null by leaving the value empty.

Then verify that the second line correctly identifies the location of the WebLogic jDriver and Oracle client libraries.

The following example from a Windows installation sets the `DB_CLASSPATH` to null and adds the WebLogic jDriver and Oracle client libraries to the system path:

```
:ORACLE_OCI_815
REM ----------- ORACLE classes -----------
SET DB_CLASSPATH=
SET PATH=%PATH%;%WEBLOGIC_HOME%\bin\oci815_8;%WLCS_ORACLE_HOME%\bin
```

Step 8: Set Up the Example Portal for Oracle

If you want to use the WebLogic Personalization Server Example Portal with your Oracle database, do the following:

1. From a command shell on the WebLogic Commerce Server and WebLogic Personalization Server host, enter the following command:
   `%WL_COMMERCE_HOME%\StartCommerce.bat` (Windows)
   `$WL_COMMERCE_HOME/StartCommerce.sh` (UNIX)

2. Open another command shell and enter the following command:
   `%WL_COMMERCE_HOME%\bin\loadrules.bat` (Windows)
   `$WL_COMMERCE_HOME/bin/loadrules.sh` (UNIX)

CHAPTER

2 Exporting Oracle Data Objects

WebLogic Commerce Server and WebLogic Personalization Server includes a set of SQL scripts that create Oracle tables and other data objects in your testing, staging, or production environment. After creating the data objects, you can use Oracle commands to export data from the source environment and import it into a destination environment.

The Oracle commands export and import all WebLogic Commerce Server and WebLogic Personalization Server data including any document metadata that is part of the default content management implementation. The commands do not move the following:

- JSP files or other configuration files.
- Documents and document metadata that is maintained by a third-party content-management tool, such as Documentum or Interwoven.

To export Oracle data objects to a destination environment, complete the following tasks:

Step 1: Create the Destination Environment

Step 2: Review Parameter Files

Step 4: Delete Sample Data

Step 5: Remove Orphaned Records

Step 6: Export the Data

Step 7: Stop the Server in the Destination Environment

Step 8: Import the Data

Step 9: Start the Server in the Destination Environment

Step 1: Create the Destination Environment

To create the destination environment, follow the procedures in Chapter 1, "Setting Up Oracle for New Installations."

For more information about WebLogic Commerce Server and WebLogic Personalization Server database schemas, refer to the following documents:

- "The Personalization Server Schema" in the *WebLogic Personalization Server Developer's Guide*
- "Product Catalog Database Schema" in the *Product Catalog Management*
- "Order Processing Database Schema" in the *Order Processing Package*

Step 2: Review Parameter Files

Complete this step only if you use a third-party solution (such as Interwoven or Documentum) for content management.

If your Oracle user account contains data objects that WebLogic Commerce Server and WebLogic Personalization Server does not create and maintain (such as Interwoven and Documentum objects), you must use parameter files to establish user identity and determine the tables for which you want to export and import data.

Reviewing the parameter files involves the following tasks:

- Review the List of Tables
- Add FROMUSER to the Import Parameter File

For more information about the Oracle export and import commands and about the function of the parameters in the WebLogic Commerce Server and WebLogic Personalization Server parameter files, see the following Oracle document: http://oradoc.photo.net/ora81/DOC/server.815/a67792/toc.htm.

Review the List of Tables

Complete this procedure only if you use a third-party solution (such as Interwoven or Documentum) for content management.

Note: In this document, `$WL_COMMERCE_HOME` refers to the directory into which you installed WebLogic Commerce Server and/or WebLogic Personalization Server and *`database-type`* refers to the type and version of RDBMS that you installed.

Review the list of tables in the parameter files and, if necessary, modify them to match your environment:

1. Save backup copies of the following files:
   `$WL_COMMERCE_HOME/db/`*`database-type`*`/migration/staging/wlcs_exp.par`

   `$WL_COMMERCE_HOME/db/`*`database-type`*`/migration/staging/wlcs_imp.par`
2. Open both files in a text editor.
3. If you are using Interwoven or Documentum for content management, prevent the WebLogic Commerce Server and WebLogic Personalization Server scripts from exporting and importing data into content management tables by removing the following table names:
   - WLCS_DOCUMENT
   - WLCS_DOCUMENT_METADATA

Note: If you are using the reference implementation (BulkLoader) for content management, you must leave these table names in the files.

4. If you added or removed tables from the default database schemas, add or remove the corresponding table names in the parameter files.
5. Save your modifications to the parameter files.

Add FROMUSER to the Import Parameter File

Complete this procedure only if you use a third-party solution (such as Interwoven or Documentum) for content management.

To specify the user account that owns the schema you want to import, do the following:

1. Make a backup copy of the following file:
   `$WL_COMMERCE_HOME/db/`*`database-type`*`/migration/staging/wlcs_imp.par`
2. Open `wlcs_imp.par` in a text editor.
3. For the `FROMUSER` entry, specify the name of the user who owned the schema in the source environment. For example, in your development environment, the owner of the Oracle schema is `WEBLOGIC`. In `wlcs_imp.par`, you modify the first line as follows:
   `FROMUSER=WEBLOGIC`

Step 4: Delete Sample Data

Caution: Before you delete sample data, **back up your database**.

If you do not plan to use sample data in the destination environment, we recommend that you delete it before you export data. **Do not delete the administrator users or administrator group**.

Your database might contain samples for the following types of data:

- Product items, categories, and keywords
- Orders
- Users and groups
- Portals
- Rules
- Documents (if you use the reference implementation for content management instead of an integration with a third party)

You can use the WebLogic Commerce Server and WebLogic Personalization Server Administration tool to delete sample data. To access the Administration tool, start WebLogic Commerce Server and WebLogic Personalization Server and enter the following URL in a Web browser:
`http://localhost:7501/tools/index.jsp`

For example, to delete sample users from the Administration tool:

1. Back up your database.
2. From the WebLogic Commerce Server and WebLogic Personalization Server Administration tool click User Management.
3. From the User Management page, click Users.
4. From the Users Page, under Search for Users, enter `*` (asterisk) in the Username box. Then click Search.

   The search mechanism returns a list of all users.
5. To delete a user, click the red X next to the username. **Do not delete the administrator users**.
6. After deleting all sample users, click Finished from the Users Page.
7. From the Users Page, click Groups.
8. From the Groups page, click the red X next to sample group names. **Do not delete the administrator group**.
9. Click Finished.

Step 5: Remove Orphaned Records

Note: In this document, `$WL_COMMERCE_HOME` refers to the directory into which you installed WebLogic Commerce Server and/or WebLogic Personalization Server and *`database-type`* refers to the type and version of RDBMS that you installed.

To verify that there are no orphaned records in your database, do the following:

1. In SQL*Plus, log in as SYSTEM (or another user with DBA privileges).
2. In the SQL*Plus session, enter the following command:

   ```
   @
   $WL_COMMERCE_HOME/db/database-type/migration/staging/del-orphan
   ed-recs.sql
   ```

   The `del-orphaned-recs.sql` removes any orphaned records.

Step 6: Export the Data

To export data, complete **one** of the following tasks:

- If you are using the reference implementation for content management, Export All Tables in the User Account.
- If you are using a third party tool for content management, Export Specific Tables from the User Account.

Export All Tables in the User Account

If you are using the reference implementation for content management and the Oracle user account in the destination environment contains only WebLogic Commerce Server and WebLogic Personalization Server data objects, do the following to export data:

1. From a command line, enter the following command:

```
exp username/password@ORACLE_SID file=wlcs.dmp owner=WEBLOGIC
log=wlcs_exp.log
```

Note: You can use a `username` and `password` for any account with DBA privileges. `ORACLE_SID` must specify the database in which the WebLogic Commerce Server and WebLogic Personalization Server schema resides. If you changed the `WEBLOGIC` user account name, change the value for `owner`.

When the `exp` command finishes exporting data, it prints the following message:

```
Export terminated successfully without warnings
```

2. To review the results of your export, open the log file that you specified in step 1.

For more information about the Oracle export and import commands, see the following Oracle document: http://oradoc.photo.net/ora81/DOC/server.815/a67792/toc.htm.

Export Specific Tables from the User Account

Complete this procedure only if you use a third-party solution (such as Interwoven or Documentum) for content management.

If your Oracle accounts contain data objects that WebLogic Commerce Server and WebLogic Personalization Server does not create and maintain (such as Interwoven and Documentum objects), do the following to export data:

1. In SQL*Plus, log in as SYSTEM (or another user with DBA privileges).

2. In the SQL*Plus session, enter the following command:

```
exp username/password@ORACLE_SID file=wlcs.dmp owner=WEBLOGIC
parfile=$WL_COMMERCE_HOME/db/database-type/migration/staging/wlcs_exp.par
log=wlcs_exp.log
```

Note: You can use a `username` and `password` for any account with DBA privileges. `ORACLE_SID` must specify the database in which the WebLogic Commerce Server and WebLogic Personalization Server schema resides. If you changed the `WEBLOGIC` user account name, change the value for `owner`.

When the `exp` command finishes exporting data, it prints the following message: `Export terminated successfully without warnings`

3. To review the results of your export, open the log file that you specified in step 2.

   If you installed WebLogic Personalization Server without WebLogic Commerce Server, the log file may include `TABLE OR VIEW DOES NOT EXIST` statements for each WebLogic Commerce Server table. You can ignore these messages.

For more information about the Oracle export and import commands, see the following Oracle document: http://oradoc.photo.net/ora81/DOC/server.815/a67792/toc.htm.

Step 7: Stop the Server in the Destination Environment

Before importing data, you must stop WebLogic Commerce Server and WebLogic Personalization Server by entering `Ctrl+C` in the shell that is running the server.

Step 8: Import the Data

Caution: Before you import data, **back up your database** in the destination environment.

To import data, complete **one** of the following tasks:

- If you are using the reference implementation for content management, Import to All Tables in the User Account.
- If you are using a third party tool for content management, Import to Specific Tables in the User Account.

Import to All Tables in the User Account

If the Oracle user account in the destination environment contains only WebLogic Commerce Server and WebLogic Personalization Server data objects, do the following to import data:

1. In the destination environment, back up your database.
2. Start SQL*Plus and log in as SYSTEM (or another user with DBA privileges).
3. Enter the following command:

   ```
   imp username/password@ORACLE_SID file=wlcs.dmp
   log=wlcs_imp.log fromuser=WEBLOGIC touser=WEBLOGIC ignore=y
   commit=y
   ```

Note: You can use a `username` and `password` for any account with DBA privileges. `ORACLE_SID` must specify the database in which the WebLogic Commerce Server and WebLogic Personalization Server schema resides. If you changed the `WEBLOGIC` user account name, change the value for `fromuser` and `touser`.

For more information about the Oracle export and import commands, see the following Oracle document: http://oradoc.photo.net/ora81/DOC/server.815/a67792/toc.htm.

Import to Specific Tables in the User Account

Complete this procedure only if you use a third-party solution (such as Interwoven or Documentum) for content management.

If your Oracle accounts contain data objects that WebLogic Commerce Server and WebLogic Personalization Server does not create and maintain (such as Interwoven and Documentum objects), do the following to import data:

1. In the destination environment, back up your database.
2. Start SQL*Plus and log in as SYSTEM (or another user with DBA privileges).
3. Enter the following command:

```
imp username/password@ORACLE_SID file=wlcs.dmp log=wlcs_imp.log
parfile=$WL_COMMERCE_HOME/db/database-type/migration/staging/wlcs_imp.par
```

Note: You can use a `username` and `password` for any account with DBA privileges. `ORACLE_SID` must specify the database in which the WebLogic Commerce Server and WebLogic Personalization Server schema resides.

For more information about the Oracle export and import commands, see the following Oracle document: http://oradoc.photo.net/ora81/DOC/server.815/a67792/toc.htm.

Step 9: Start the Server in the Destination Environment

To start WebLogic Commerce Server and/or WebLogic Personalization Server on UNIX, enter the following command from a WebLogic Commerce Server and WebLogic Personalization Server host:

```
$WL_COMMERCE_HOME/StartCommerce.sh
```

To start WebLogic Commerce Server and/or WebLogic Personalization Server on Windows, on a WebLogic Commerce Server and WebLogic Personalization Server host do one of the following:

- Click Start → Programs → WebLogic Commerce Server 3.1 → Start Commerce Server
- From a command prompt, enter the following command: `%WL_COMMERCE_HOME%\StartCommerce.bat`

For information on starting the server, refer to Chapter 5, “Starting the Server.”

CHAPTER

3 Review Server Defaults

Before you start WebLogic Commerce Server and WebLogic Personalization Server, we recommend that you review default properties in the following configuration files:

- Review Defaults in weblogic.properties
- Review Defaults in weblogiccommerce.properties

These default settings assume that you are deploying WebLogic Commerce Server and WebLogic Personalization Server in a development environment. For information on tuning performance for a production environment, refer to the *Performance Tuning Guide*.

Review Defaults in weblogic.properties

When you install WebLogic Commerce Server and WebLogic Personalization Server, it includes a `weblogic.properties` file that configures WebLogic Server to use the default services and features of WebLogic Commerce Server and WebLogic Personalization Server. To modify these settings, or to add other services, open `WL_COMMERCE_SERVER/weblogic.properties` in a text editor. For most production environments, you need to make the following modifications:

- Set the Listen Ports
- Change the Document Root
- Deploy Additional EJBs

Set the Listen Ports

The `weblogic.properties` file specifies two separate listen ports:

- A port for standard transfers via HTTP, illustrated in Listing 3-1.
- A port for encrypted transfers with SSL via HTTPS, illustrated in Listing 3-2. For information on setting up SSL for WebLogic Server, refer to "Using WebLogic SSL," which is part of the WebLogic Server documentation set.

If you modify the default values, **you must also modify the port numbers for each Web application** that you deploy to ensure that the Web applications create URLs that specify the proper port number. For information on modifying port numbers for Web applications, refer to "Port Numbers" on page 4-10.

Listing 3-1 Listen Port for HTTP

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# CORE SYSTEM PROPERTIES
# -------------------------------------------------

# TCP/IP port number at which the WebLogic Server listens for connections

weblogic.system.listenPort=7501
```

Listing 3-2 Listen Port for HTTPS

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# CORE SECURITY-RELATED PROPERTIES FOR SSL
# -------------------------------------------------
# Read important information about SSL at:
#
#   http://www.weblogic.com/docs51/classdocs/API_secure.html
#

# Enable SSL
# (default if property not defined is false)

weblogic.security.ssl.enable=true

#
# SSL listen port

weblogic.system.SSLListenPort=7502
```

Change the Document Root

By default, `weblogic.properties` sets the document root to the `public_html` directory below the directory in which you installed WebLogic Commerce Server and WebLogic Personalization Server. If you want to change the location, modify the value for `weblogic.httpd.documentRoot`. For example, Listing 3-3 shows a UNIX environment that assigns the document root to a file system that is mounted as **/usr1**.

The document root establishes the top directory from which WebLogic Commerce Server and WebLogic Personalization Server serve Web resources that are not deployed as Web applications.

Listing 3-3 Modifying the DocumentRoot Configuration

```
# DocumentRoot configuration

# -------------------------------------------------

# The documentRoot is by default set to this directory.

weblogic.httpd.documentRoot=/usr1/publish/public_html/
```

Deploy Additional EJBs

The `weblogic.ejb.deploy` property value specifies a pathname for all EJBs that WebLogic Server deploys. (See Listing 3-4.) If you create your own EJBs that you want to deploy, or if you change the location of the EJBs, then modify the value for the `weblogic.ejb.deploy` property.

Note: You can use relative or absolute pathnames.

Listing 3-4 List of Deployed EJBs

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# BEA WEBLOGIC COMMERCE SERVER EJB PROPERTIES
# ------------------------------------------------
# CLUSTER USERS: Note that ALL EJB deployment should be done in the
# *per-cluster* properties file ONLY.
#
weblogic.ejb.deploy=\
        D:/WebLogicCommerceServer3.1/lib/foundation.jar,\
        D:/WebLogicCommerceServer3.1/lib/axiom.jar,\
        D:/WebLogicCommerceServer3.1/lib/ebusiness.jar,\
        D:/WebLogicCommerceServer3.1/lib/advisor.jar,\
        D:/WebLogicCommerceServer3.1/lib/bridge.jar,\
        D:/WebLogicCommerceServer3.1/lib/document.jar,\
        D:/WebLogicCommerceServer3.1/lib/p13nadvisor.jar,\
        D:/WebLogicCommerceServer3.1/lib/portal.jar,\
        D:/WebLogicCommerceServer3.1/lib/ruleeditorbeans.jar,\
        D:/WebLogicCommerceServer3.1/lib/rulesservice.jar,\
        D:/WebLogicCommerceServer3.1/lib/servicemgr.jar
```

Review Defaults in weblogiccommerce.properties

The `$WL_COMMERCE_SERVER/weblogiccommerce.properties` file sets values for WebLogic Commerce Server and WebLogic Personalization Server. Before you deploy the server and your Web applications, we recommend that you review the file, which is divided into the following sections:

- JDBC Pool and JNDI Helpers
- Location of GIF Files
- Catalog Cache
- EJB Mappings
- Default Values for Portals
- User Management
- Name and Location of Documentation
- Database Timeout
- Message Log
- Component Properties
- Payment Service
- Credit Card Security
- TAXWARE
- Debug Mode for PipelineSession
- Webflow and Pipeline Hot Deploy
- Property Cache Settings

JDBC Pool and JNDI Helpers

The properties in Listing 3-5 enable WebLogic Commerce Server and WebLogic Personalization Server to use JDBC pools to connect to the database. Do not change these properties.

Listing 3-5 also shows properties that define the JNDI context that WebLogic Commerce Server and WebLogic Personalization Server uses to locate EJBs.

If you place JSPs and EJBs on separate servers, set the `commerce.configuration.splitEjbJspServer` property to `true` and enable the default context properties by removing the comment tag (#).

Listing 3-5 JDBC Pool and JNDI Helpers

```
# JDBC Pool entries

commerce.jdbc.pool.name=commercePool

commerce.jts.pool.name=commercePool

PersistenceURL=jdbc:weblogic:jts:commercePool

commerce.jdbc.pool.url=weblogic.jdbc.jts.commercePool


# JNDI helper default context setup properties. Typically you wouldn't change these.

# Only change these if your JNDI tree is located on another server, and it is not in a cluster.

#default.context.initial.context.factory=weblogic.jndi.WLInitialContextFactory

#default.context.provider.url=t3://localhost:7501

#default.context.security.authentication=simple

#default.context.security.principal=admin

#default.context.security.credentials=adminpassword

#default.context.security.protocol=ssl

#default.context.authoritative=
              #default.context.batchsize=
              #default.context.dns.url=dns://somehost/wiz.com
              #default.context.language=
              #default.context.object.factories=
              #default.context.referral=
              #default.context.url.pkg.prefixes=
###############################################
# Server configuration
###############################################
# Set to true if the EJB server is not on the same cluster node as the
# JSP server.
commerce.configuration.splitEjbJspServer=false
```

Location of GIF Files

The WebLogic Commerce Server JSP templates uses the Webflow helper to generate references to images. By default, the Webflow helper references are relative to the root of the current Web application.

If you locate your images outside the Web application directory tree, activate the property in Listing 3-6 and provide a base URL for image pathnames.

Listing 3-6 Location of GIF Files

```
# Location of where gif's are stored.
# If not set then they are taken relative to the web app.
# commerce.gif.url.base=http://<hostname>:<port>/<path>
```

Catalog Cache

The properties in Listing 3-7 define caches for the product catalog. For more information about this section, refer to “Improving Catalog Performance by Optimizing the Catalog Cache” under “Catalog Administration Tasks” in *Product Catalog Management*.

Listing 3-7 Catalog Cache

```
# Cache entries

ProductItemCache.ttl=21600000

ProductItemCache.capacity=10000

ProductItemCache.enabled=true


CategoryCache.ttl=86400000

CategoryCache.capacity=1000

CategoryCache.enabled=true
```

EJB Mappings

WebLogic Commerce Server and WebLogic Personalization Server uses the properties in Listing 3-8 to locate EJB homes. Do not modify these properties.

Listing 3-8 EJB Mappings

```
# Entries for mapping schema group names to the appropriate home name

commerce.schemaGroupHomeName.USER=com.beasys.commerce.foundation.property.Schem
a

commerce.schemaGroupHomeName.REQUEST=com.beasys.commerce.foundation.property.Sc
hema
```

```
commerce.schemaGroupHomeName.SESSION=com.beasys.commerce.foundation.property.Sc
hema

commerce.schemaGroupHomeName.CONTENT=com.beasys.commerce.axiom.document.Documen
tSchema

commerce.schemaGroupHomeName.APPLICATION_INIT=com.beasys.commerce.foundation.pr
operty.Schema

commerce.schemaGroupHomeName.CATALOG=com.beasys.commerce.foundation.property.Sc
hema


# EJB Home Name Associations

commerce.home.p13n.AdvisorHome=\

    com.beasys.commerce.axiom.p13n.advisor.PersonalizationAdvisorHome

commerce.home.rules.RulesServiceHome=\

    com.beasys.commerce.axiom.reasoning.rules.service.RulesServiceHome

commerce.home.property.SchemaManagerHome=\

    com.beasys.commerce.foundation.property.SchemaManager

 commerce.home.AdvisorHome=\

     com.beasys.commerce.axiom.advisor.AdvisorHome
```

Default Values for Portals

The properties in Listing 3-9 establish defaults for new portals. A developer might want to adjust these values based on the nature of the portal being built. The defaults can be overridden using the portal administration tools.

Listing 3-9 Default Values for Portals

```
# Entries taken from PortalProperties
commerce.default.portal.schemaName=_DEFAULT_PORTAL_SCHEMA
commerce.default.portal.headerURL=header.jsp
commerce.default.portal.contentURL=portalcontent.jsp
commerce.default.portal.footerURL=footer.jsp
commerce.default.portal.cols=3
commerce.default.portal.suspendedURL=suspended.jsp
commerce.default.portlet.dir=portlets/
commerce.default.portlet.image.dir=portlets/images/
commerce.default.portlet.titlebarURL=titlebar.jsp
commerce.default.banner.color=#666666
commerce.default.titlebar.bgcolor=#333399
commerce.default.show.borders=false
commerce.default.content.bgcolor=#CCCCCC
commerce.default.titlebar.font.color=#FFFFFF
commerce.default.body.bgcolor=#FFFFFF
commerce.default.profile.group.property=portal.profile.group
commerce.default.portal.password=guest
commerce.application.minimum.build=83914
commerce.default.category.name=Home
commerce.portalservicemanager.fast.portal.get=true
```

For example, the following property tells the portal component which property set to use as the default for portal users. A developer may want to change this value based on property sets created using the Property Set Management administration tool. Example:

```
commerce.default.portal.schemaName=_DEFAULT_PORTAL_SCHEMA
```

For example, the following portal properties are for internal use and should not be altered by a developer. Example:

```
commerce.application.minimum.build=83914
commerce.default.category.name=Home
```

User Management

Listing 3-10 illustrates the section of `weblogiccommerce.properties` that specifies properties for user management. The first set of properties map table names. Do not modify these properties.

The set of properties under `# RDBMS Realm` determine which database type the reference implementation uses for storing user information. Modify this section only if you use the reference implementation for user management. If you use an LDAP server for user management, you do not need to modify this section. For more information, see “Using WebLogic Realms” under “Creating and Managing Users” in *Using WebLogic Personalization Server*.

Listing 3-10 User Management

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

# User Management

###############################################

commerce.usermgmt.UserTable=WLCS_USER

commerce.usermgmt.GroupTable=WLCS_GROUP

commerce.usermgmt.UserGroupHierarchyTable=WLCS_USER_GROUP_HIERARCHY

commerce.usermgmt.GroupHierarchyTable=WLCS_GROUP_HIERARCHY

commerce.usermgmt.EntityIdTable=WLCS_ENTITY_ID

commerce.usermgmt.ProfileTypeTable=WLCS_UNIFIED_PROFILE_TYPE


# RDBMS Realm

#------Oracle thin Driver---------#

#commerce.usermgmt.RDBMSRealm.driver=oracle.jdbc.driver.OracleDriver

#commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:oracle:thin:@@ORACLE_SERVER@:@ORACLE_P
ORT@:@ORACLE_SID@

#commerce.usermgmt.RDBMSRealm.dbUser=@ORACLE_USER@

#commerce.usermgmt.RDBMSRealm.dbPassword=@ORACLE_PASSWORD@

#------WebLogic jDriver for Oracle 8.1.5---------#

#commerce.usermgmt.RDBMSRealm.driver=weblogic.jdbc.oci.Driver

#commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:weblogic:oracle

#commerce.usermgmt.RDBMSRealm.dbServer=@ORACLE_NET_SERVICE_NAME@

#commerce.usermgmt.RDBMSRealm.dbUser=@ORACLE_USER@

#commerce.usermgmt.RDBMSRealm.dbPassword=@ORACLE_PASSWORD@
```

```
#--------------Cloudscape------------------#

commerce.usermgmt.RDBMSRealm.driver=COM.cloudscape.core.JDBCDriver

commerce.usermgmt.RDBMSRealm.dbUrl=jdbc:cloudscape:Commerce;create=true;autocom
mit=false

commerce.usermgmt.RDBMSRealm.dbUser=none

commerce.usermgmt.RDBMSRealm.dbPassword=none
```

Name and Location of Documentation

Listing 3-11 includes the following properties that configure the WebLogic Personalization Server Content Management service:

- `commerce.home.content.*` and `commerce.home.document.*` specify JNDI home names for EJBs and provide constants for the `com.beasys.commerce.content.ContentHelper` class.

  `commerce.home.content.ContentManagerHome` provides the default value for the `contentHome` parameter in the `<cm:select>` and `<cm:selectById>` JSP tags. For information about Content Management JSP tags, refer to "JSP Tag Library Reference" in the *WebLogic Personalization Server Developer's Guide*.

- `commerce.content.defaultReadOnly` controls the default read/write state of objects that Content Management JSP tags return. Do not modify this property.

- `commerce.content.cache.useSoftHashMap` controls whether the `ContentCache` object uses `SoftReferences` to maintain the cache. `SoftReferences` work correctly under the non-HotSpot Win32 1.2.2 Java VM. However, they do not work correctly under the Solaris production 1.2.1 Java VM. Modify this property only to resolve memory issues.

- `commerce.xml.entity.basePath` provides the `com.beasys.commerce.axiom.util.P13NEntityResolver` object with the location of DTDs. Do not modify this property. For more information, refer to the Javadoc for the `com.beasys.commerce.axiom.util.P13NEntityResolver` class.

Listing 3-11 Name and Location of Documentation

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

# Documentation name and location properties

###############################################

commerce.p13n.document.dev.name=index.html

commerce.p13n.document.dev.version=32

commerce.p13n.document.dev.docRootDir=docs

commerce.home.content.ContentHome=com.beasys.commerce.axiom.document.Document

commerce.home.content.ContentManagerHome=com.beasys.commerce.axiom.document.Doc
umentManager

commerce.home.content.ContentSchemaHome=com.beasys.commerce.axiom.document.Docu
mentSchema

commerce.home.document.DocumentHome=com.beasys.commerce.axiom.document.Document

commerce.home.document.DocumentManagerHome=com.beasys.commerce.axiom.document.D
ocumentManager

commerce.home.document.DocumentSchemaHome=com.beasys.commerce.axiom.document.Do
cumentSchema

commerce.content.defaultReadOnly=true

commerce.content.cache.useSoftHashMap=false


commerce.xml.entity.basePath=D:/WebLogicCommerceServer3.2/lib/dtd
```

Database Timeout

The properties in Listing 3-12 specify the amount of time that WebLogic Commerce Server and WebLogic Personalization Server waits to obtain a connection to the database pool before timing out.

Listing 3-12 Database Timeout

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

# Time out properties for weblogic database connection

##############################################

connection.TimeOutPeriod=999999999

waitForConnection=true
```

Message Log

The `# Logger class` properties in Listing 3-13 specify the class that logs messages. Do not modify these properties.

The `commerce.log.minimum.level` in Listing 3-13 determines the types of messages that WebLogic Commerce Server and WebLogic Personalization Server outputs to the console. If you are running WebLogic Commerce Server and WebLogic Personalization Server in debug mode, you must set this value to `200` or `199`.

Listing 3-13 Message Log

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

# Logger class

##############################################

commerce.log.class=com.beasys.commerce.axiom.util.weblogic.Log

commerce.log.display.deprecated=true
```

```
###########################################################################
# Set Message display level codes
#
# LOG_DEPRECATED=199    : Set this to get deprecated warning output
# LOG_DEBUG=200         : Set this to get debug messages
# LOG_INFO=201          : Set this to get informational messages
# LOG_WARNING=202       : Set this to get warning messages
# LOG_ERROR=203         : Set this to get error messages
# LOG_FATAL=204         : Set this to get Fatal error messages
# LOG_SECURITY=205      : Set this to get Security related messages
#
# Note: Whenever you set the level to any of the above, by default all the
# higher level messages are also enabled.
# For ex: if the level is set to 200 all levels >= 200
# are enabled.
###########################################################################

commerce.log.minimum.level=203
```

Component Properties

The properties in Listing 3-14 are used by the Bean-Managed Persistence (BMP) layer for data persistence. Do not modify these properties.

Listing 3-14 Component Properties

```
#############################################
# Properties required for components
############################################

# Use the BMP plugin by default
DefaultPersistenceHelperPlugin=com.beasys.commerce.foundation.plugin.bmp.BMPPer
sistenceHelperPlugin
```

Payment Service

The properties in Listing 3-15 configure the payment service, including the CyberCash credit card validation service.

For more information about this section, refer to “Configuration Activities for Using CyberCash” under “Payment Services” in *Order Processing Package*.

Listing 3-15 Payment Service

```
###################################################
# Properties required for the payment component
###################################################
#
# This property defers payment authorization to the administration tools.
# If set to true, all payment service authorization calls are disabled

# and payment transactions are persisted in a RETRY state. Payments must

# then be reauthorized through the payment administration tool.

#

commerce.payment.defer.authorization=true



#

# CyberCash configuration files contain CyberCash-specific data, such as a

# merchant-id and merchant hash secret. The specific properties in the
configuration

# files depend upon the payment model assigned to a merchant by his/her financial

# institution. The two files declared below are example files and are provided

# for demonstration purposes ONLY. MERCHANTS MUST AQUIRE A CYBERCASH CONFIGURATION

# FILE FROM CYBERCASH. These will be furnished by CyberCash as part of the
# merchant agreement. Once a merchant has a CyberCash configuration file, the
# property below must be replaced with the location of the configuration file.

#

# Example:
CyberCashConfigFile=c:/merchant/config/file/location/merchant_conf-terminal

# This file may be used for testing terminal based payment models.

CyberCashConfigFile=D:/WebLogicCommerceServer3.2/eval/common/CyberCash/conf/mer
chant_conf-terminal

# This file may be used for testing host based payment models.

#CyberCashConfigFile=D:/WebLogicCommerceServer3.2/eval/common/CyberCash/conf/me
rchant_conf-host

#

# Properties below represent the different payment models provided by CyberCash.

#

# Terminal based models.

PaymentModel=AUTO_MARK_AUTO_SETTLE
```

```
#PaymentModel=AUTO_MARK_AUTO_SETTLE_AVS

#PaymentModel=AUTO_MARK_MANUAL_SETTLE

#PaymentModel=AUTO_MARK_MANUAL_SETTLE_AVS

#PaymentModel=MANUAL_MARK_AUTO_SETTLE

#PaymentModel=MANUAL_MARK_AUTO_SETTLE_AVS

#PaymentModel=MANUAL_MARK_MANUAL_SETTLE

#PaymentModel=MANUAL_MARK_MANUAL_SETTLE_AVS

# Host based models.

#PaymentModel=HOST_AUTH_CAPTURE

#PaymentModel=HOST_AUTH_CAPTURE_AVS

#PaymentModel=HOST_POST_AUTH_CAPTURE

#PaymentModel=HOST_POST_AUTH_CAPTURE_AVS
```

Credit Card Security

The properties in Listing 3-16 configures security services for credit card transactions. For information about this section, refer to “Credit Card Security Service” under “Payment Services” in *Order Processing Package*.

Listing 3-16 Credit Card Security

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

# Properties required for Security Services

####################################################

#

# Credit card encryption services are turned on by setting this property to true.

# Commenting out this property or setting it to false will disable credit card

# encryption.
```

```
#
is.encryption.enabled=true

#
# The name of the security table and column names for public and private encryption
# keys can be specified using the properties below.
#
security.table.name=WLCS_SECURITY
security.backup.table=WLCS_SECURITY_BACKUP
public.key.column.name=PUBLIC_KEY
private.key.column.name=PRIVATE_KEY

#
# The key bit size desired.  Key bit length and length of data that can be encrypted
# are related as follows:
#    KEY BIT LENGTH(bits)           DATA LENGTH (bytes)
#        512                             53
#       1024                            117
#       2048 (MAX LENGTH)               245
key.bit.size=1024

#
# This optional parameter specifies the private key password used to decrypt the
# encrypted credit card encryption private key.  WARNING: Setting this property
# will start up the server without prompting for a password.
private.key.password=WLCS
```

TAXWARE

The properties in Listing 3-17 configure the TAXWARE tax calculation service. For information about this section, refer to “TAXWARE Configuration and Deployment” under “Taxation Services” in *Order Processing Package*.

Listing 3-17 TAXWARE

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

# TAXWARE TAX SERVICE  PROPERTIES

# -------------------------------------------------

# For instructions on individual properties in this

# section, Pleae refer to the Taxware SalesUse and Verazip manuals.

#

#################################################################


#################################################################

# JNDI name of the TaxCalculator Session Bean

# ----------------------------------------------------------------

tax.calculator.jndi.name =
com.beasys.commerce.ebusiness.tax.taxware.TaxwareTaxCalculator


#################################################################

# Currency for Tax Calculation (Taxware only supports USD)

# ----------------------------------------------------------------

tax.currency = USD


#################################################################

# ShipFrom Address
# ----------------------------------------------------------------
# ShipFrom Address is address from where goods are shipped
# Please review Taxware documentation when setting these properties
#
shipfrom.countycode=000
shipfrom.state=MA
shipfrom.city=SALEM
shipfrom.zip=01970
shipfrom.geocode=00
shipfrom.country=USA

################################################################
# Order Acceptance Address
#----------------------------------------------------------------
# OrderAcceptance is the address where orders are accepted
# Please review Taxware documentation when setting these properties
#
orderacceptance.countycode=000
orderacceptance.state=MA
orderacceptance.city=SALEM
orderacceptance.zip=01970
orderacceptance.geocode=00
orderacceptance.country=USA

#################################################################
# Order Origin Address
#---------------------------------------------------------------
```

```
# Order Origin is the address where orders are Originated
# Please review Taxware documentation when setting these properties
#
orderorigin.countycode=000
orderorigin.state=MA
orderorigin.city=SALEM
orderorigin.zip=01970
orderorigin.geocode=00
orderorigin.country=USA

#################################################################
# Point of title passage
# ---------------------------------------------------------------
# Location at which legal title has transferred to purchaser

#titlepassage=shipto
titlepassage=shipfrom

#################################################################
# Company Identification
#---------------------------------------------------------------
# User Defined company identification to access information
# for tax calculating and reporting

companyId=companyId

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

```
# TaxType
#----------------------------------------------------------------
# Type of tax to be calculated

#taxtype=use
#taxtype=rental
#taxtype=consumeruse
#taxtype=services
taxtype=sales

################################################################
# TaxSelParm
#---------------------------------------------------------------
# Taxselparm to decide jurisdiction while calculating
# if value is 2 Calculate tax only
# if value is 3 Determine jurisdiction and calculate taxes
#taxselparm=2
taxselparm=3
```

Debug Mode for PipelineSession

The properties in Listing 3-18 establish a debug mode, in which you can determine whether you have assigned the correct scope to an attribute. To see the debug messages, you must set the message log to display debug messages. For more information, see “Message Log” on page 3-14.

For more information about attribute scopes, refer to “Attribute Scoping” under “Extending Webflow and Pipelines” in *Webflow and Pipeline Management*.

Listing 3-18 Debug Mode for PipelineSession

```
################################################################
# Enable Debug Mode for PipelineSession
#---------------------------------------------------------------
# This checks if the attribute that the user is trying to set to the
# PipelineSession is serializable or not
# If the attribute is not serializable then a message and exception
# stack trace is displayed

pipelineSession.debug=false
```

Webflow and Pipeline Hot Deploy

The properties in Listing 3-19 determine whether you can change your Web application’s Webflow and Pipeline without restarting WebLogic Commerce Server and WebLogic Personalization Server.

For more information about changing Webflow and Pipeline, see “Customizing Webflow and Pipelines” in *Webflow and Pipeline Management*.

Listing 3-19 Webflow and Pipeline Hot Deploy

```
#############################################
# Enable webflow and pipeline hot deploy
#--------------------------------------------
webflow.hotdeploy.enable=false
pipeline.hotdeploy.enable=false
```

Property Cache Settings

The properties in Listing 3-20 configure the property cache, which decreases the amount of time needed to access user, group, and other properties. For information on the property cache, refer to "Enable Property Caching" in *Performance Tuning Guide*.

Listing 3-20 Property Cache Settings

```
#############################################
# Cache settings
# cache of unified profile types, 1 hour, 100 entries
unifiedProfileTypeCache.ttl=3600000
unifiedProfileTypeCache.capacity=100
unifiedProfileTypeCache.enabled=true

# cache of entity properties, 10 minutes, 500 entries
entityPropertyCache.ttl=600000
entityPropertyCache.capacity=500
entityPropertyCache.enabled=true

# cache of property default values, 1 hour, 100 entries
propertyDefaultCache.ttl=3600000
propertyDefaultCache.capcity=100
propertyDefaultCache.enabled=true

# cache of entity ids, 1 hour, 500 entries
entityIdCache.ttl=3600000
entityIdCache.capacity=500
entityIdCache.enabled=true
```

```
# cache of explicit properties, 10 minutes, 100 entries
directPropertyManager.ttl=600000
directPropertyManager.capacity=100
directPropertyManager.enabled=true

# cache of ConfigurableEntity methods, 1 hour, 100 entries
ConfigurableEntityMethodCache.ttl=3600000
ConfigurableEntityMethodCache.capacity=100
ConfigurableEntityMethodCache.enabled=true

# cache of ldap usernames, 1 hour, 100 entries
ldapUserCache.ttl=3600000
ldapUserCache.capacity=100
ldapUserCache.enabled=true

# cache of ldap group names, 1 hour, 10 entries
# (raise this if you have a lot of groups in ldap)
ldapGroupCache.ttl=3600000
ldapGroupCache.capacity=10
ldapGroupCache.enabled=true

#cache of session values
_sessionCache.ttl=900000
_sessionCache.capacity=10000
_sessionCache.enabled=true
```

```
#cache of values across multiple users

_globalCache.ttl=600000

_globalCache.capacity=1000

_globalCache.enabled=true
```

CHAPTER

4 Deploying Web Applications

You can use WebLogic Commerce Server and WebLogic Personalization Server services and resources to deploy your e-commerce Web site as a Web application.

Note: Before you read this section, we recommend that you read the BEA WebLogic Server document *Writing a Web Application*. It contains background information you need, even if you used the WebLogic Commerce Server sample JSP templates and the WebLogic Personalization Server example portal to develop your Web application.

This section describes the process of deploying Web applications that use WebLogic Commerce Server and WebLogic Personalization Server services and resources. It includes the following topics:

- About Web Applications
- Recommendation for Deploying Expanded Directory Hierarchies or WAR Files
- Registering a Web Application in weblogic.properties
- Setting Up a Deployment Descriptor
- Restarting the Server

About Web Applications

A Web application, as defined by the *Java Servlet Specification v2.2*, is a set of server-side resources that make up an online application. These resources include:

- Servlets
- JavaServer Pages (JSP)
- Static HTML pages
- Server side classes
- Enterprise JavaBeans (EJB)
- Other resources specific to the Web application

WebLogic Commerce Server and WebLogic Personalization Server provide EJBs, classes, and other infrastructure that you need to create Web applications. They also provide sample JSPs and other sample objects that you can use as a template for developing your own Web applications.

Sample Web Applications

The WebLogic Commerce Server and WebLogic Personalization Server software includes three Web applications that you can use as a template for your own Web applications:

- The WebLogic Commerce Server e-commerce Webapp, which is packaged as an expanded directory hierarchy in `$WL_COMMERCE_HOME/server/webapps/wlcs`.
- The WebLogic Personalization Server example portal Web application, which is packaged as a `.war` file in `$WL_COMMERCE_HOME/server/webapps/examples/portal/portal.war`

Note: A WebLogic Personalization Server example portal is also installed in `$WL_COMMERCE_HOME/server/public_html/portals`. This example portal is not a Web application; it does not include a deployment descriptor and it does not use other features that the J2EE specification makes available to Web applications. For more information, refer to the following section, Comparison of Deploying the Example Portals.

- The administration tools Web application, which is packaged as an expanded directory hierarchy in `$WL_COMMERCE_HOME/server/webapps/admin`.

Comparison of Deploying the Example Portals

When you install WebLogic Personalization Server, it includes two example portals that differ only in the manner in which they are deployed:

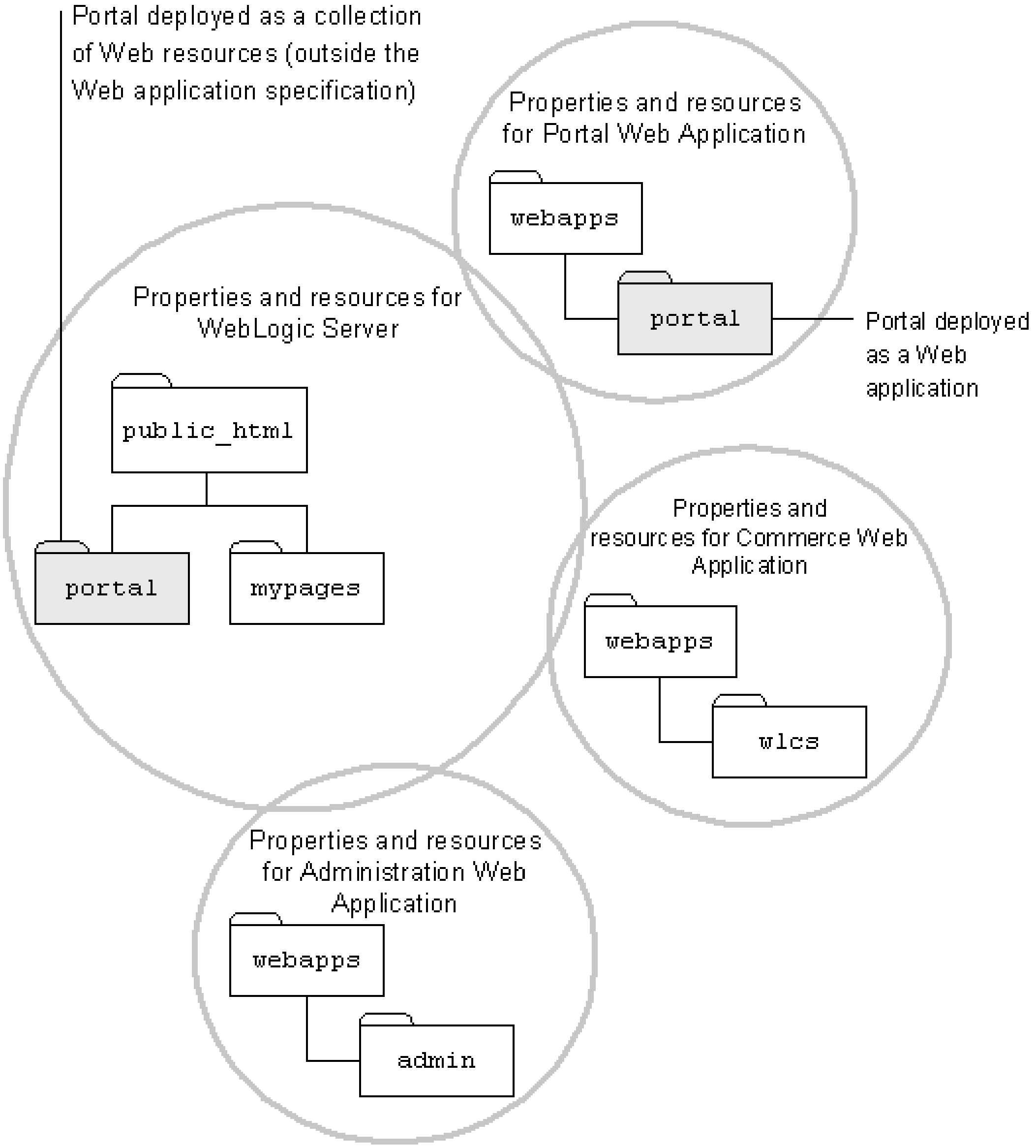
- `$WL_COMMERCE_HOME/server/webapps/examples/portal/portal.war` is deployed as a Web application. Because it is a Web application, its resources and properties are self contained within the WebLogic Server EJB container. Other Web applications and other services that are deployed in the WebLogic Server container cannot access the example portal resources. For example, it defines the rate at which the server checks for modifications to JSPs within the Web application. You can set different page-check intervals for other Web applications or you can use the default page-check interval property.

  In addition, because it is a Web application, it uses features such as declarative security that the J2EE specification defines for Web applications.

- `$WL_COMMERCE_HOME/server/public_html/portals` is deployed as a collection of servlets, JSPs, and EJBs that share resources and properties with other collections. It cannot override settings in `weblogic.properties` and it cannot use declarative security.

Figure 4-1 illustrates the relationships between Web application properties and resources and WebLogic Server properties and resources. Note that each Web application shares some common properties and resources but contains another set that are not shared. Files deployed under `public_html` (which is the document root that `weblogic.properties` establishes) use only the WebLogic Server properties and resources.

Figure 4-1 Web Application Properties and WebLogic Server Properties


Portal deployed as a collection of Web resources (outside the Web application specification)
Properties and resources for Portal Web Application
webapps
portal
Portal deployed as a Web application
Properties and resources for WebLogic Server
public_html
portal
mypages
Properties and resources for Commerce Web Application
webapps
wlcs
Properties and resources for Administration Web Application
webapps
admin

Recommendation for Deploying Expanded Directory Hierarchies or WAR Files

Within the Web application specification, you can deploy a Web application as an expanded directory hierarchy or in an archived format such as a `.war` file.

We recommend that you deploy a Web application as an expanded directory hierarchy when any of the following conditions is true:

- **While developing and testing your Web application**. During your development cycle you may frequently modify files in the Web application and want to see the results of your modifications without needing to rejar the Web application.
- **On a production site that frequently needs to update files within the Web application**. For example, a site that frequently adds or updates JSPs should be deployed as an expanded directory. When the server checks for modifications to JSPs, it will recompile any modified JSPs that you have deployed. For information about setting the page-check interval, refer to the *Performance Tuning Guide*. (Because the database and any third-party content management system do not reside within Web application hierarchy, updates to these objects do not modify the Web application.)

  If you deploy your Web application as an expanded directory hierarchy, you can also hot deploy modifications to EJBs and some servlets. For more information about hot deployment, see “Using WebLogic Server Hot Deploy” on the WebLogic Server documentation site.

We recommend that you deploy your Web application as a `.war` file on production sites that do not frequently update files within the Web application. Deploying a single file is less error prone than deploying an expanded directory hierarchy, but you cannot hot deploy a modified `.war` file.

Registering a Web Application in weblogic.properties

To notify WebLogic Commerce Server and WebLogic Personalization Server that you have added a Web application, you must first shut down the server. Then add the following property to the `$WL_COMMERCE_HOME/weblogic.properties` file:

`weblogic.httpd.webApp.`*`myapp=location`*

Where:

`myapp`

Is the context name given to the Web application. The context name is included in the initial part of any URL request to the Web application.

`location`

Is the root directory of the Web application or the location of the `.war` file that contains the Web application archive.

For example, the administration Web application places the following lines in `weblogic.properties` file:

```
# Defines the admin tools Web Application.
weblogic.httpd.webApp.tools=/WLC3/server/webapps/admin/
```

When deploying in your production environment, consider removing or deactivating (by placing a # character at the beginning of the line) the properties for the example portal and sample JSP templates:

```
# Defines the example portal WAR deployment.
#weblogic.httpd.webApp.portal=/WLC3/server/webapps/examples/portal/portal.war

# Defines the wlcs Web Application.
#weblogic.httpd.webApp.wlcs=/WLC3/server/webapps/wlcs/
```

Defining a Default Web Application

You can specify a default Web application so that WebLogic Server interprets any requests for the document root (as defined in `weblogic.properties`) as a request for the default Web application.

If you are deploying only one Web application, consider making it the default. If you are deploying multiple Web applications, you can create a default Web application that acts as a portal for the other Web applications.

To define a default Web application, add the following property to `$WL_COMMERCE_HOME/weblogic.properties`:

```
weblogic.httpd.defaultWebApp=myapp
```

For more information about default Web applications, refer to “Default Web Application,” under “Deploying on WebLogic Server” in the WebLogic Server document *Writing a Web Application*.

Setting Up a Deployment Descriptor

Each Web application uses a deployment descriptor, formatted in XML, to specify the following information:

- Servlet/JSP definitions and mappings
- Session configuration
- Initialization parameters
- MIME type mappings
- Welcome file list
- Error pages
- Security declarations

If you are using the WebLogic Commerce Server and WebLogic Personalization Server deployment descriptors as a starting point for you own Web application, open `$WL_COMMERCE_HOME/server/webapps/wlcs/WEB-INF/web.xml` in a text editor and verify or modify the following parameters:

- Port Numbers
- Main Page
- Session Timeout
- Security Declarations

In most environments, you can deploy your Web application without changing the other parameters in the file. For information on modifying parameters to enhance performance in the production environment, refer to the *Performance Tuning Guide*.

For information on creating your own deployment descriptor, see “Creating the Deployment Descriptor” in the WebLogic Server document *Writing a Web Application*.

Port Numbers

When generating a URL for a Web application, the `createWebflowURL()` method refers to the `HTTP_PORT` and `HTTPS_PORT` parameters in the Web application's deployment descriptor to determine the port numbers to encode (Listing 4-1).

Listing 4-1 Parameters for Port Numbers

```
<context-param>
 <param-name>HTTP_PORT</param-name>
 <param-value>port-number</param-value>
</context-param>

<context-param>
   <param-name>HTTPS_PORT</param-name>
   <param-value>port-number</param-value>
</context-param>
```

The WebLogic Commerce Server `web.xml` already defines these parameters. If you are modifying this file for your own Web application, open `$WL_COMMERCE_HOME/server/webapps/wlcs/web.xml` and do one of the following:

- If you want the Web application to use the same port numbers as WebLogic Server, either deactivate the parameters by surrounding them in comment tags, `<!-- -->`, or change the values to match the values in `$WL_COMMERCE_HOME/weblogic.properties`.
- If you want the Web application to use different port numbers (for example, if you want to use the port numbers of a proxy server), change the values.

Main Page

To specify a main page for your Web application, use the `<welcome-file>` parameter (Listing 4-2). If you have named your main page something other than `index.jsp`, modify the value in your `web.xml` file.

By specifying a main page, you can access the Web application by specifying the context name in the URL: `http://host:port/context-name`. You determine the Web application context name when you register it in `weblogic.properties`. For more information about context names, see "Registering a Web Application in weblogic.properties."

Listing 4-2 Parameter for the Main Page

```
<welcome-file-list>
     <!-- This is the entry point to a WLCS site.
          Change this appropriately -->
 <welcome-file>/index.jsp</welcome-file>
</welcome-file-list>
```

Session Timeout

The `<session-timeout>` parameter determines how many minutes of inactivity the server will tolerate before ending the session (Listing 4-3). The `$WL_COMMERCE_HOME/server/webapps/wlcs/web.xml` file sets the parameter to 15 minutes. You can modify this setting depending on your security needs.

Listing 4-3 Parameter for Session Timeout

```
<!-- Commerce Server Session Configuration -->

<session-config>

<!-- The default session timeout interval is 15 minutes. This is
measured based on session inactivity - that is, the session will be
invalidated 15 minutes after a client last accesses WLCS.

Configure this to a suitable interval -->

   <session-timeout>15</session-timeout>

</session-config>
```

Security Declarations

You set up security declarations for your Web application in the deployment descriptor. The security declarations do the following:

- Establish Roles
- Establish Security Constraints
- Determine Which Links Use HTTPS

Establish Roles

The WebLogic Commerce Server web.xml file includes the following parameters to establish one role, called CustomerRole (Listing 4-4).

Listing 4-4 Parameters for Establishing Roles

```
<security-role>

  <description>Registered customers with role
  "CustomerRole"</description>

   <role-name>CustomerRole</role-name>

</security-role>
```

When users create a user account through WebLogic Commerce Server, Pipeline components add the username to the CustomerRole.

If you set up Pipeline components to add users to other roles, you must declare each role in the Web application's deployment descriptor by adding parameters that are illustrated in Listing 4-4, substituting the <role-name> value with the name of the role you created.

Establish Security Constraints

After defining roles, you can specify a collection of resources (JSPs) that only specific roles can access.

The security constraint in Listing 4-5 is an excerpt from $WL_COMMERCE_HOME/server/webapps/wlcs/web.xml. If you are using this file as a template for your own deployment descriptor, make sure that the <url-pattern> parameters match any JSPs that you want to be accessible only to users in the CustomerRole. You can add <url-pattern> parameters to specify new directories or to specify specific files. Note that a pattern or filename must start with a / character (forward slash). For more information on specifying URL patterns, refer to the *Java Servlet Specification v2.2*.

If you add any URL patterns to the <security-constraint> element, you must also add those patterns to the patterns defined for <param-name>HTTPS_URL_PATTERNS</param-name>. For more information, see “Determine Which Links Use HTTPS” on page 4-15.

To give another role access to the resource collection, add <role-name> elements to the <auth-constraint> element.

Listing 4-5 Security Constraint for A Collection of Resources

```
<security-constraint>

<!-- Define a resource collection -->

 <web-resource-collection>

    <web-resource-name>Customer Profile - Self Administration
    Pages</web-resource-name>

     <description>Customer Profile - Self Administration Pages</description>

     <!-- URL pattern for the resource collection -->

     <url-pattern>/commerce/user/*</url-pattern>

     <url-pattern>/commerce/order/*</url-pattern>

     <http-method>GET</http-method>

     <http-method>POST</http-method>

    </web-resource-collection>


    <!-- This constraint applies to users with role "CustomerRole" -->

    <auth-constraint>

       <description>Users with role "CustomerRole"</description>

       <role-name>CustomerRole</role-name>

    </auth-constraint>


    <!-- For enabling SSL, specify CONFIDENTIAL or INTEGRAL. -->

    <user-data-constraint>
```

```
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Determine Which Links Use HTTPS

The security constraints described in the previous topic identify files that only specific roles can access. In addition to declaring a set of files secured by role, you also declare a set of files, pipelines, and input processors that need to be accessed via HTTPS. When the `createWebflowURL()` method encounters one of the resources you declare, it generates a URL that uses the HTTPS protocol.

Listing 4-6 is an excerpt from `$WL_COMMERCE_HOME/server/webapps/wlcs/web.xml`. If you are using this file as a template for your own deployment descriptor, make sure that the `<param-value>` values match any JSPs, pipelines, and input processors that you want to be accessed via HTTPS.

You can add `<param-value>` values to name specific resources or to name new patterns. Name specific resources by adding their name to the list. When naming specific pipelines and input processors, do not use the `.inputprocessor` or `.pipeline` extension.

Specify patterns for pipelines and input processors in the form of **`pattern`**`_*`. For example, to enable SSL for all requests to input processors whose names start with `profileeditcc_`, use the pattern `profileeditcc_*`.

If you add any target names or patterns to the <param-value> element, you must also add them to the <security-constraint> element, which is described in "Establish Security Constraints" on page 4-13.

Listing 4-6 Parameters for Generating Links to Secured Resources

```
<context-param>
   <param-name>HTTPS_URL_PATTERNS</param-name>
   <param-value>
      /commerce/user/*,/commerce/order/*,/commerce/register/*,
      newuser_*,profilenewcc_*,paymentnewcc_*,profileeditcc_*,
      paymenteditcc_*,viewprofile_*,editprofile_*,profilenewaddress_*,
      profileeditaddress_*,changepassword_*,EnterShippingInfo_*,
      SelectShippingAddress_*,AddNewShippingAddress_*,orderhistory_*,
      RefreshOrderHistory,AuthorizePayment,CommitOrder,
      RefreshPaymentHistory,DeleteCreditCard,TaxVerifyShippingAddress_*,
      shoppingcart_InitShippingMethodList
   </param-value>
</context-param>
```

Restarting the Server

After modifying properties files for the Web application, you can start WebLogic Commerce Server and WebLogic Personalization Server. For information on starting the server, refer to Chapter 5, "Starting the Server."

For a description of the URLs you use to access Web applications, see the WebLogic Server topic "URLs and Web Applications" in *Writing a Web Application*.

CHAPTER

5 Starting the Server

To start WebLogic Commerce Server and WebLogic Personalization Server, you must set environment variables and start the Java Virtual Machine (JVM) with the `weblogic.Server` class name and a specific set of parameters. WebLogic Commerce Server and WebLogic Personalization Server provide two files, `set-environment` and `StartCommerce`, that you can use to start the server, and on Microsoft Windows platforms, it creates a shortcut on the Start menu. If you prefer, instead of using `set-environment` and `StartCommerce`, you can set the required environment variables and start the JVM in any way that your operating system supports.

This topic contains the following sections:

- Starting the Server on UNIX
- Starting the Server on Windows
- Setting Environment Variables
- Starting the JVM

Starting the Server on UNIX

To start the server on UNIX, enter the following command from a WebLogic Commerce Server and WebLogic Personalization Server host, where `$WL_COMMERCE_HOME` is the directory into which you installed WebLogic Commerce Server and WebLogic Personalization Server:

```
$WL_COMMERCE_HOME/StartCommerce.sh
```

The `StartCommerce.sh` calls `set-environment.sh` to set environment variables. Then it passes to the JVM the class name and parameters that start the server.

For information on starting the server without using the `StartCommerce.sh` and `set-environment.sh` files, refer to “Setting Environment Variables” on page 5-3 and “Starting the JVM” on page 5-5.

Starting the Server on Windows

To start the server on a Windows WebLogic Commerce Server and WebLogic Personalization Server host, do one of the following:

- Click Start → Programs → WebLogic Commerce Server *release-number* → Start Commerce Server
- From a command prompt, enter the following command:
  ```
  %WL_COMMERCE_HOME%\StartCommerce.bat
  ```

The Start Commerce Server command on the Start menu is a shortcut to `StartCommerce.bat`. The `StartCommerce.bat` calls `set-environment.bat` to set environment variables. Then it passes to the JVM the class name and parameters that start the server.

For information on starting the server without using the `StartCommerce.bat` and `set-environment.bat` files, refer to “Setting Environment Variables” on page 5-3 and “Starting the JVM” on page 5-5.

Setting Environment Variables

Before starting the server, you must set environment variables and add directories to the system path. Although the `set-environment` and `StartCommerce` files do this for you, you can set the variables in any other way that your operating system supports.

Set the following environment variables:

Note: The examples in the following list are from a Windows `set-environment.bat` file.

- `WL_COMMERCE_HOME={` Directory into which you installed WebLogic Commerce Server and WebLogic Personalization Server }
- `JDK_HOME={` Directory into which you installed the JDK that WebLogic Commerce Server and WebLogic Personalization Server require }
- `WLCS_ORACLE_HOME={` If you are using Oracle, set this variable to the directory in which you installed the Oracle client software. }
- `WEBLOGIC_HOME={` Directory into which you installed WebLogic Server }
- `DATABASE={ CLOUDSCAPE | ORACLE_OCI_815` },
  where `CLOUDSCAPE` is the sample database and `ORACLE_OCI_815` is the Oracle jDriver.
- `SYSTEM_NAME={` The top directory of the server instance. For more information about multiple server instances, refer to "Setting Up Multihoming" in *Running and Maintaining the WebLogic Server.*}
- `SYSTEM_HOME={ %WL_COMMERCE_HOME%` (Windows) |
  `$WEBLOGIC_COMMERCE_HOME` (UNIX) }

- `DB_CLASSPATH=`
  - For the Cloudscape sample database,
    {
    `%WEBLOGIC_HOME%\eval\cloudscape\lib\cloudscape.jar;%WEBLOGIC_HOME%\eval\cloudscape\lib\tools.jar;%WEBLOGIC_HOME%\eval\cloudscape\lib\client.jar`
    }
  - For the WebLogic jDriver, set the variable to null by defining the variable and leaving the value blank. For example, `DB_CLASSPATH=`
- `JDK_TOOLS=` { The pathname of the `tools.jar` file in the JDK that WebLogic Commerce Server and WebLogic Personalization Server require. }

  For example, `JDK_TOOLS=%JDK_HOME%\lib\tools.jar`
- `JAVA_CLASSPATH=` { The classes that boot WebLogic Commerce Server and WebLogic Personalization Server. } For example, `%JDK_TOOLS%;%WEBLOGIC_HOME%\lib\weblogic510sp6boot.jar;%WEBLOGIC_HOME%\classes\boot`
- `WLPS_CLASSPATH=` { The classes required for the WebLogic Personalization Server runtime environment. } For example, `%WL_COMMERCE_HOME%\license;%WL_COMMERCE_HOME%\classes;%WL_COMMERCE_HOME%\lib\rules.jar;%WL_COMMERCE_HOME%\lib\jrulesserviceprovider.jar`
- `WLCS_CLASSPATH=` { The classes required for the WebLogic Commerce Server run-time environment } For example, `%WLPS_CLASSPATH%;%WL_COMMERCE_HOME%\deploy\bmp\classes;%WL_COMMERCE_HOME%\eval\win32\Taxware\classes`
- `WEBLOGIC_CLASSPATH=` { The classes required for the WebLogic Server run-time environment. } For example, `%WEBLOGIC_HOME%\lib\weblogic510sp6.jar;%WEBLOGIC_HOME%\lib\WebLogic_RDBMS.jar;%WEBLOGIC_HOME%\license;%WEBLOGIC_HOME%\classes;%WEBLOGIC_HOME%\lib\weblogicaux.jar;%WEBLOGIC_HOME%\lib\weblogic-tags-510.jar;%WLCS_CLASSPATH%;%DB_CLASSPATH%`
- `SCRIPTS_CLASSPATH=` { The classes required to run WebLogic Commerce Server and WebLogic Personalization Server scripts. } For example, `%JAVA_CLASSPATH%;%WEBLOGIC_CLASSPATH%`

Add the following directories to the system path:

- The directory that contains the WebLogic Server binary files. For example, `%WEBLOGIC_HOME%\bin`
- The directory that contains the CyberCash binary files. For example, `%WEBLOGIC_HOME%\eval\win32\CyberCash\bin`
- The directory that contains the TAXWARE binary files. For example, `%WEBLOGIC_HOME%\eval\win32\Taxware\bin`
- If you are using the Oracle jDriver, the directory that contains the driver and the directory that contains the Oracle client binary files. For example, `%WEBLOGIC_HOME%\bin\oci815_8;%WLCS_ORACLE_HOME%\bin`

Starting the JVM

After setting environment variables, start the JVM by entering the following command on a single line:

```
%JDK_HOME%\bin\java -ms64m -mx128m

-server

-classpath %JAVA_CLASSPATH%

-Dweblogic.class.path=%WEBLOGIC_CLASSPATH%

-Dweblogic.system.name=%SYSTEM_NAME%

-Dweblogic.system.home=%SYSTEM_HOME%

-Dweblogic.home=%WEBLOGIC_HOME% -Djava.security.manager

-Djava.security.policy=%WEBLOGIC_HOME%\weblogic.policy

-Dcommerce.properties=%WL_COMMERCE_HOME%\weblogiccommerce.propert
ies

-Dweblogic.properties=%WL_COMMERCE_HOME%\weblogic.properties

-Dpipeline.properties=%WL_COMMERCE_HOME%\pipeline.properties

-Dwebflow.properties=%WL_COMMERCE_HOME%\webflow.properties

weblogic.Server
```

Note: The `-server` option activates the HotSpot server VM. Before starting the server on Windows, you must download and install the server VM in addition to the standard JDK 1.3 installation. To verify whether the HotSpot server VM is installed, navigate to the `jdk/jdk1.3.0/jre/bin` directory. If the server VM is installed, the `bin` directory includes a `server` subdirectory.

For information on adjusting command options to optimize performance for a production environment, refer to the following topics:

- "Use the HotSpot Server Virtual Machine" in the WebLogic Commerce Server and WebLogic Personalization Server *Performance Tuning Guide*.
- "JVM Execution Parameters" in the *WebLogic Server Performance Tuning Guide*.

Index