BEA WebLogic Portal™®

Database Administration Guide

Version 8.1 with Service Pack 2
Document Revised: December, 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

Introduction to Database Administration for WebLogic Portal

Using a Microsoft SQL Server Database

Using an Oracle Database

Using a Sybase Database

Using a DB2 Database

The Data Dictionary

XA Support

WebLogic Portal DDL Modules

About This Document

This document explains how to set up and administer a database for WebLogic Portal. It covers the following topics:

- Chapter 1, "Introduction to Database Administration for WebLogic Portal" provides an introduction to database administration issues for WebLogic Portal.
- Chapter 2, "Using a Microsoft SQL Server Database" provides information for setting up a SQL Server environment for WebLogic Portal, and instructions for switching from the PointBase database to SQL Server.
- Chapter 3, "Using an Oracle Database" provides information for setting up an Oracle environment for WebLogic Portal, and instructions for switching from the PointBase database to Oracle.
- Chapter 6, "The Data Dictionary," provides a complete list of all of the database schemas and dictionaries.
- Chapter 7, "XA Support" describes XA support in WebLogic Portal 8.1.
- Appendix A, "WebLogic Portal DDL Modules" describes the file naming convention used for the WebLogic Portal DDL files.

Product Documentation on the dev2dev Web Site

BEA product documentation, along with other information about BEA software, is available from the BEA dev2dev Web site:

`http://dev2dev.bea.com`

To view the documentation for a particular product, select that product from the list on the dev2dev page; the home page for the specified product is displayed. From the menu on the left side of the screen, select Documentation for the appropriate release. The home page is displayed for the complete documentation set for the product and release you have selected.

Related Information

Readers of this document may find the following documentation and resources especially useful:

- For general information about Java applications, go to the Sun Microsystems, Inc. Java Web site at `http://java.sun.com`.
- For general information about XML, go to the O'Reilly & Associates, Inc. `XML.com` Web site at `http://www.xml.com`.

Contact Us!

Your feedback on the BEA WebLogic Portal 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 Portal documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Portal 8.1.

If you have any questions about this version of BEA WebLogic Portal, or if you have problems installing and running BEA WebLogic Portal, contact BEA Customer Support at **http://support.bea.com**. You can also contact Customer Support by using the contact information provided on the quick reference sheet titled "BEA Customer Support," 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
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.
`monospace text`	Indicates *user input*, as shown in the following examples: • Filenames: `config.xml` • Pathnames: *`BEAHOME`*`/config/examples` • Commands: `java -Dbea.home=BEA_HOME` • Code: `public TextMsg createTextMsg(`
	Indicates *computer output*, such as error messages, as shown in the following example: `Exception occurred during event dispatching:java.lang.ArrayIndexOutOfBoundsException: No such child: 0`
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
`{ }`	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*: `java utils.MulticastTest -n` *`name`* `[-p` *`portnumber`*`]`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. *Example*: `java weblogic.deploy [list|deploy|update]`

Convention	Item

`. . .`	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 "file1.cpp file2.cpp`
	`file3.cpp . . ."`

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

Introduction to Database Administration for WebLogic Portal

This document helps you set up and start using a database management system (DBMS) with your WebLogic Platform 8.1 system.

By default, you can run the code samples provided with WebLogic Platform with the PointBase DBMS. PointBase is a pure-Java relational database management system that BEA includes with WebLogic Platform to allow you to run code samples. It is supported only for the design, development, and verification of applications; it is not supported for production server deployment.

Using PointBase

PointBase is the default database that BEA provides. It is used for the BEA sample domains, and it is the default database used when you create a domain with the Configuration Wizard.

PointBase runs on its own server that must be running for your applications to access it. When you start WebLogic Portal server to run your applications, the PointBase server starts automatically.

PointBase Documentation

PointBase documentation, for the version of PointBase currently supported by WebLogic Server, is distributed with WebLogic Server in PDF form in the `WL_HOME\common\eval\pointbase\docs` directory. The PointBase documentation consists of the following three manuals.

- PointBase Console Guide

- PointBase System Guide
- PointBase Developer's Guide

PointBase JAR Files

Refer to the section titled “PointBase JAR Files” in the PointBase System Guide for information on the JAR files provided with WebLogic Server in the `WL_HOME\common\eval\pointbase\lib` directory.

PointBase Tools

Scripts for starting the PointBase server and the PointBase console are distributed with WebLogic Server in the `WL_HOME\common\eval\pointbase\tools` directory. Scripts are called by start scripts in the sample domains and start scripts contained in any domain created by the Configuration Wizard. These PointBase start scripts simplify starting the PointBase Server and Console within WebLogic domains.

WebLogic Portal PointBase Databases

PointBase stores all data in .dbn files and all log information in .wal files. Database properties are stored in PointBase.ini files. Data files for WebLogic Portal are named **workshop.dbn** and log files for WebLogic Portal are named **workshop$1.wal**. Pre-built PointBase data, log and PointBase.ini files for WebLogic Portal samples are included in the following directory:

```
WL_HOME\samples\domains\portal
```

By default domains created via the Configuration Wizard with the Basic WebLogic Portal Domain template would create PointBase data and log files in the following directory:

```
BEA_HOME\user_projects\portalDomain
```

Administering the WebLogic Portal PointBase Database

You can administer the default database installed with WebLogic Portal using the PointBase administrative console, or any third party database visualization and management tool that can connect via JDBC.

The PointBase Console can be launched either from the Windows Start Menu or by executing the startPointBaseConsole.cmd/.sh script located in the domain directory.

Prior to launching the PointBase Console ensure that WebLogic Server for the domain is running. You will not be able to use the PointBase Console unless WebLogic Server is running.

To Launch PointBase Console from the Windows Start menu

Go to Start > Programs > BEA WebLogic Platform 8.1 > Examples > WebLogic Portal > PointBase Console.

To launch the PointBase Console for the Portal Examples, or, if you added Start menu options for a domain created by the Configuration Wizard, navigate to that domain's PointBase Console menu option.

To Launch PointBase Console from the startPointBaseConsole script

1. Change directories to WL_HOME\samples\domains\portal.
2. Execute the appropriate start script — startPointBaseConsole.cmd or startPointBaseConsole.sh — to launch the PointBase Console for the Portal Examples. For a domain created by the Configuration Wizard navigate to that domain's home directory and execute the startPointBaseConsole.cmd/.sh script.
3. When the PointBase Console starts, it prompts you to enter connection parameters to properly connect to the database. Enter the following connection information, which is also what you will need if you use a third-party product to access the PointBase database:
   – **Driver**: com.pointbase.jdbc.jdbcUniversalDriver
   – **URL**: jdbc:pointbase:server://localhost:9093/workshop
   – **User**: weblogic
   – **Password**: weblogic

Using a Database Other Than PointBase with WebLogic Portal

Use the Supported Configurations document to determine what databases and JDBC drivers are supported. Refer to the following chapters for instructions on the following tasks:

1. Configure the database.

Note: After the database is configured, you can use the Configuration Wizard to create and load appropriate database objects and set JDBC driver settings at domain creation time.

See "WebLogic Platform - Creating WebLogic Configurations Using the Configuration Wizard" for details.

2. Create database objects.
3. Configure your domain's JDBC driver settings.

CHAPTER 2

Using a Microsoft SQL Server Database

This section describes the steps necessary to use a Microsoft SQL Server database with WebLogic Portal 8.1, and includes information on the following subjects:

- Understanding Database Configuration for WebLogic Platform
- Configuring a Microsoft SQL Server Database
- Creating Database Objects
- Configuring Your Domain’s JDBC Driver Settings
- Creating a Database for Behavior Tracking Events

Review this entire chapter and any release notes before proceeding. The steps in this chapter should be performed by a database administrator.

Understanding Database Configuration for WebLogic Platform

Typically, you use the WebLogic Configuration Wizard to configure and connect to the database that you will use to support WebLogic Platform. For more information about how to use the WebLogic Configuration Wizard, see http://edocs.bea.com/platform/docs81/confgwiz/index.html.

When using the Configuration Wizard to configure databases for use with WebLogic Platform, use the following steps.

1. Create your vendor database(s). If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

Note: If you are creating a separate database for behavior event tracking, see "Creating a Database for Behavior Tracking Events" on page 2-8.

2. Prepare the database for use with WebLogic Platform. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. After the database is configured, use the Configuration Wizard to create and load appropriate database objects and set JDBC driver settings at domain creation time.

Manually Configuring Databases

In some cases, you may need to manual configure your database(s) without the use of the Configuration Wizard.

Manual configuration should be used in the following cases:

- If your desired database was not configured via the WebLogic Configuration Wizard.
- If after running the Configuration Wizard you decide to have your domain point to a different database.
- If you would like to refresh your database with the base configuration data that comes with the product.

  Note: BEA's database creation scripts first drop all database objects and then recreate them, which means all data added since your original installation will be lost. Upon completion of the database creation scripts, only the base configuration data that is needed for the product will exist.

- When you want to create only a subset of Portal database objects, for example to create only Behavior Tracking database objects for a particular database.

Use the following steps when you need to manually configure a database.

1. Create your vendor database(s), see page 2-3. If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

2. Prepare the database for use with WebLogic Platform, see page 2-3. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. Create database objects, see page 2-5. This creates proprietary database objects that are used by WebLogic. This is done by editing the database properties file provided by BEA.

4. Configure the JDBC settings for your database using WebLogic Console Server,see page 2-6.

Configuring a Microsoft SQL Server Database

Before following the steps outlined in this chapter, you need to have already defined your SQL Server instance and the databases you need.

Be sure that you are using a supported version, see http://edocs.bea.com/platform/docs81/support/supp_plat.html#1085671.

To configure a SQL Server database, do the following:

1. Install the SQL Server client on the WebLogic Platform host and do the following:
   a. Configure it for access to your SQL Server database.
   b. Ensure that you can connect to your SQL Server database via the OSQL utility.

      See your SQL Server documentation for details.

Note: If you plan to use the Configuration Wizard to create the database objects for a new domain, you will not need to install the SQL Server Client.

2. Verify that security authentication settings for the SQL Server are set to “SQL Server and Windows.”
   a. From Enterprise Manager, right-click on the desired SQL Server.
   b. Select Properties, then select the Security tab.
   c. Under authentication, ensure that SQL Server and Windows is selected.
3. Prepare the SQL Server Database. The database creation scripts will install domain-specific tables. It is recommended that you work with a SQL Server system or database administrator to adjust the `SAMPLE` scripts and create database devices, databases, and database users for your SQL Server environment.

Notes: Multiple databases are required if you have multiple domains, or to run multiple environments using the same SQL Server instance (for example, if you want to run development and system test from a single SQL Server installation).

Be sure to back up your database(s) before installing any new database objects. See your database documentation for details.

a. Review and modify the provided sample scripts to suit your environment. These scripts are available in: `<WL_HOME>\portal\db\sql_server\2000\admin`**.** The sample scripts use the following defaults:

The following table lists the script names and the usage notes for each script.

Script Name	Description
`create_database.sql`	Create the `WEBLOGIC` database and `WEBLOGIC` database owner user login. An alias is created to make `WEBLOGIC` dbo in the database. Make the WebLogic database the default database for the WebLogic user. **Usage Notes:** You need to change database names, database owner user and password. The default names are the following: • database name: WEBLOGIC • database owner user: WEBLOGIC • password: WEBLOGIC You also need to edit the script to reflect valid disk locations for DATA and the `LOG` devices, or to adjust file sizes. `DATA` and `LOG` files should be placed on separate physical disks and away from any system database files.
`statistics.sql`	Runs `sp_updatestats` to compute database statistics needed for the database optimizer. Analyze schema should be run whenever any significant changes in database data occur. Your database administrator will typically schedule `sp_updatestats` to run periodically in your environment.

install_report.sq	Builds an informational installation report about the database objects created in the WEBLOGIC schema.
bt_create_database.sql	Create the WEBLOGIC_EVENT database and WEBLOGIC_EVENT database owner user login. An alias is created to make WEBLOGIC_EVENT dbo in the database. **Usage Notes:** You need to change database names, database owner user and password. The default names are the following: • database name: WEBLOGIC_EVENT • database owner user: WEBLOGIC_EVENT • password: WEBLOGIC_EVENT You also need to edit the script to reflect valid disk locations for DATA and the LOG devices, or to adjust file sizes. DATA and LOG files should be placed on separate physical disks and away from any system database files.

b. Run create_database.sql via OSQL as a user with System Administrator privileges (i.e. the sa user). For example:

```
osql –Usa -SSQLSERVER -e -icreate_database.sql -ocreate_database.log
```

The output from running create_database.sql is written to create_database.log. Verify that there are no errors in the log file before proceeding.

Creating Database Objects

The scripts to create Microsoft SQL Server database objects were designed to run in a Windows environment (they use the OSQL utility to create Microsoft SQL Server database objects). If you are using UNIX version of WebLogic Server with a Microsoft SQL Server database and do not have WebLogic products also installed on Windows contact BEA support for assistance.

To create WebLogic Platform database objects, use the following steps:

1. Verify that you can connect to the target database server with a valid user ID and password. For example:

```
osql -SSQLSERVER -U<userid> -P<password>
```

2. Open your domain’s db_settings.properties file for edit and comment out the database settings for PointBase.

3. Uncomment the database settings for your new target database and update the following settings for your database:
   - `server=`
   - `dblogin=`
   - `password=`
4. Initialize the database with the new settings.
   a. For Windows, navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory and double-click on the `create_db.cmd` file.
   b. Verify the results in the `db.log` file.

Note: If you are using the sample domain, run the create_db.cmd file from the following directory: `<BEA_HOME>\weblogic81\samples\domains\portal`.

5. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 2-6.

Configuring Your Domain's JDBC Driver Settings

Note: These settings do not allow support for XA functionality. For instructions on enabling XA, consult Chapter 7, "XA Support".

1. Start the WebLogic Server for your domain.
2. Login to the WebLogic Server Console.
3. Configure your new connection pools.
   a. Go to Services -> JDBC -> Connection Pools.
   b. Click Configure a new Connection Pool.
   c. Select the appropriate Database Type and Non-XA Database Driver from the drop down list boxes and click Continue.
   d. Choose a name for the new Connection Pool (For example: cgPoolN) and fill in the blanks for your vendor database. Click Continue.
   e. Test your connection to verify that you can successfully connect to your database.
   f. Create and deploy your new Connection Pool.
4. Update your data sources.

   a. From Services -> JDBC -> Data Sources, click on each data source and switch each to the newly created connection pool. Be sure to apply each change.

   b. Verify that each Data Source is changed by clicking on Data Sources and then verifying that Pool Name has been set to the new Connection Pool for each.

5. From Services -> JMS -> Stores -> cgJMSStore, switch cgJMSStore to use the new Connection Pool.
6. Stop your domain's WebLogic Server, then restart it.
7. In the WebLogic Server Console, delete the original cgPool connection pool.

   a. Go to Services -> JDBC -> Connection Pools.

   b. Right-click on the cgPool connection pool and select Delete.

Creating a Database for Behavior Tracking Events

You may want to store behavior tracking events in a different location than other WebLogic Portal database objects for increased performance. For more information about behavior tracking, see http://e-docs.bea.com/wlp/docs81/adminportal/help/SA_BehavTrackServ.html.

Note: By default, behavior tracking database objects are created in the same database as other WebLogic Portal database objects. You only need to follow these steps if you are configuring a separate database for behavior tracking events.

1. Verify that you can connect to the target database, see step 1. in "Configuring a Microsoft SQL Server Database".

2. Run bt_create_database.sql via OSQL as a user with system administrator privileges (i.e. the sa user). For example:

   ```
   osql -Usa -SSQLSERVER -e -ibt_create_database.sql
   -obt_create_database.log
   ```

   The output from running bt_create_database.sql is written to bt_create_database.log. Verify that there are no errors in the log file before proceeding.

3. Open your domain's db_settings.properties file for edit.

   a. Within the db_settings.properties file, uncomment the database settings for your new target database and update the following settings for your database:

   - server=<SERVER_NAME>
   - dblogin=WEBLOGIC_EVENT
   - password=WEBLOGIC_EVENT

4. Within the db_settings.properties file, find the p13n_modules, portal_modules and netuix_modules lines at the top of the file.

   b. Copy these 3 lines and comment out the original settings by adding a # sign.

   c. Replace the original settings as follows:

   – Replace p13n_modules=p13n au bt ds with p13n_modules=bt.

   – Replace portal_modules=cm wlcs wps collaboration sample_cm with portal_modules=

   – Replace netuix_modules=pf with netuix_modules=.

   When you are finished, the section should look like this:

```
#p13n_modules=p13n au bt ds

#portal_modules=cm wlcs wps collaboration sample_cm

#netuix_modules=pf

p13n_modules=bt

portal_modules=

netuix_modules=
```

d. Save the changes to the `db_settings.properties` file.

5. Initialize the database with the new settings.

   a. Navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, and double-click on the `create_db.cmd` file.

   b. Verify the results in the `db.log` file.

Note: If you are using the sample domain, run the create_db.cmd file from the following directory: `<BEA_HOME>\weblogic81\samples\domains\portal`.

6. Configure a connection pool to access your behavior tracking database and associate the p13n_tracking. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 2-6.

Using an Oracle Database

This section describes the steps necessary to use an Oracle database with WebLogic Portal 8.1, and includes information on the following subjects:

- Understanding Database Configuration for WebLogic Platform
- Configuring an Oracle Database
- Creating Database Objects
- Configure Your Domain's JDBC Driver Settings
- Creating a Database for Behavior Tracking Events

Review this entire chapter and any release notes before proceeding. Typically, the steps described in this chapter should be performed by an Oracle system administrator or a database administrator.

Understanding Database Configuration for WebLogic Platform

Typically, you use the WebLogic Configuration Wizard to configure and connect to the database that you will use to support WebLogic Platform. For more information about how to use the WebLogic Configuration Wizard, see http://edocs.bea.com/platform/docs81/confgwiz/index.html.

When using the Configuration Wizard to configure databases for use with WebLogic Platform, use the following steps.

1. Create your vendor database(s). If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

Note: If you are creating a separate database for behavior event tracking, see "Creating a Database for Behavior Tracking Events" on page 3-8.

2. Prepare the database for use with WebLogic Platform. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. After the database is configured, use the Configuration Wizard to create and load appropriate database objects and set JDBC driver settings at domain creation time.

Manually Configuring Databases

In some cases, you may wish to manually configure the database or manually modify your existing settings. For example, manual configuration should be used in the following cases:

- If your desired database was not configured via the WebLogic Configuration Wizard.
- If after running the Configuration Wizard you decided to have your domain point to a different database.
- If you would like to refresh your database with the base configuration data that comes with the product. (Note: All the database creation scripts first drop all database objects and recreate them, which means all data added since your original installation will be lost. Upon completion of the database creation scripts only the base configuration data that is needed for the product will exist.)
- When you want to create only a subset of Portal database objects, for example to create only Behavior Tracking database objects for a particular database.

Use the following steps when you need to manually configure a database.

1. Create your vendor database(s), see page 3-3. If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

2. Prepare the database for use with WebLogic Platform, see page 3-3. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. Create database objects, see page 3-5. This creates proprietary database objects that are used by WebLogic. This is done by editing the database properties file provided by BEA.

4. Configure the JDBC settings for your database using WebLogic Console Server, see page 3-6.

Configuring an Oracle Database

Before following the steps outlined in this chapter, you need to have already defined the Oracle databases you need.

Note the following when defining your Oracle instance and databases.

- Be sure that you are using a supported version, see http://edocs.bea.com/platform/docs81/support/supp_plat.html#1085671.
- Define a blocksize of at least 8K for increased performance.

1. Install the Oracle client software on the WebLogic Platform host.

   a. Configure a Local Net Service to access the target Oracle database.

   b. Be sure that Oracle environment variables are defined, and that the Oracle bin directory is included in the `$PATH` variable.

   c. Verify that you can connect to the target Oracle database via SQLPlus.

Note: If you plan to use the Configuration Wizard to create the database objects for a new domain, you do not need to install the Oracle Client.

2. Prepare the Oracle Database.

Notes: Multiple databases are required if you have multiple domains, or to run multiple environments using the same SQL Server instance (for example, if you want to run development and system test from a single SQL Server installation).

Be sure to back up your database(s) before installing any new database objects. See your database documentation for details.

   a. Edit the SAMPLE scripts provided in: <*WL_HOME*>/portal/db/oracle/817/admin to suit your environment.

   The database creation scripts will install domain-specific tables for each. It is recommended that you work with a database administrator to adjust the `SAMPLE` scripts, and to create the database schema owner users and tablespaces needed for your environment.

b. Review the Description and Usage Notes for each script.

Script Name	Description

`create_tablespaces.sql`	Creates data and index tablespaces.
	Usage Notes: Edits are required to modify the pathnames for the DATA_PATHNAME and INDEX_PATHNAME variables to match your local directory path structures. For example, on a UNIX system, if two disks are mounted as /usr1 and /usr2 and the Oracle SID is PROD, use the following pathnames:
	`DEFINE DATA_PATHNAME=/usr1/oradata/PROD`
	`DEFINE INDEX_PATHNAME=/usr2/oradata/PROD`
	Edits are also required if you want to change the tablespace names. The following defaults are used:
	• `WEBLOGIC_DATA`: tables for WebLogic Portal and/or WebLogic Platform
	• `WEBLOGIC_INDEX`: indexes for WebLogic Portal and/or WebLogic Platform

`create_users.sql`	Creates a `WEBLOGIC` schema owner user, establishes the users password, default and temporary tablespaces and grants privileges to that user. **Usage Notes:** Edits are required to change the schema owner user name, password and tablespace names. The following defaults are used: • database user = WEBLOGIC • database password = WEBLOIGIC • default tablespace = WEBLOGIC_DATA • temporary tablespace = TEMP
`rebuild_indexes.sql`	Rebuilds `WEBLOGIC` (schema user) indexes to move them from the `WEBLOGIC_DATA` tablespace to the `WEBLOGIC_INDEX` tablespace.
`statistics.sql`	Runs `analyze_schema` to compute database statistics needed for the Oracle optimizer. Analyze schema should be run whenever any significant changes in database data occur. Your database administrator will typically schedule `analyze_schema` to run periodically in your environment.
`install_report.sql`	Builds an informational installation report about the database objects created in the schema.
`db_size.sql`	Builds a report showing free space in database tablespaces.

Script Name	Description
bt_create_tablespaces.sql	Creates the tablespace for behavior event tracking. **Usage Notes:** Edits are required to modify the pathnames for the EVT_DATA_PATHNAME and INDEX_PATHNAME variables to match your local directory path structures. • `WEBLOGIC_DATA`: tables for WebLogic Portal and/or WebLogic Platform • `WEBLOGIC_INDEX`: indexes for WebLogic Portal and/or WebLogic Platform
bt_create_users.sql	Creates a behavior event tracking user, establishes the user's password, default and temporary tablespaces and grants privileges to that user. **Usage Notes:** Edits are required to change the schema owner user name, password and tablespace names. Edits are required to change file sizes and device names. The following defaults are used: • database user: WEBLOGIC_EVENT • password: WEBLOGIC_EVENT

c. To run these scripts from a shell, change directories to:

```
WL_HOME/portal/db/oracle/817/admin
```

d. Start SQL*Plus as the system user. For example:

sqlplus system*/manager@MYDB*

e. From SQL*Plus, execute the create_tablespaces.sql script. using the @ sign. For example:

```
@create_tablespaces.sql
```

f. From SQL*Plus, execute the create_users.sql script using the @ sign. For example,

```
@create_users.sql
```

3. Follow the steps in "Creating Database Objects" on page 3-5.

Creating Database Objects

To create WebLogic Platform database objects, use the following steps:

1. Use the following command to verify that you can connect to the target database server with a valid user ID and password:

   `sqlplus user_ID/password@DB_SID`

2. Open your domains db_settings.properties file for edit and comment out the database settings for PointBase.

3. Uncomment the database settings for your new target database and update the following settings for your database:

   - `server=`
   - `dblogin=`
   - `password=`

4. Initialize the database with the new settings.

   a. For Windows, navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, and double-click on the `create_db.cmd` file.

   b. For UNIX, navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, run `create_db.sh`.

   c. Verify the results in the `db.log` file.

Note: If you are using the sample domain, run the create_db.cmd/sh file from the following directory: `<BEA_HOME>\weblogic81\samples\domains\portal`.

5. Follow the steps in "Configure Your Domain's JDBC Driver Settings" on page 3-6.

Configure Your Domain's JDBC Driver Settings

Note: These settings do not allow support for XA functionality. For instructions on enabling XA, consult Chapter 7, "XA Support".

1. Start the WebLogic Server for your domain.

2. Login to the WebLogic Server Console.

3. Configure your new connection pools.

   a. Go to Services -> JDBC -> Connection Pools.

   b. Click Configure a new Connection Pool.

   c. Select the appropriate Database Type and Non-XA Database Driver from the drop down list boxes and click Continue.

   d. Choose a name for the new Connection Pool (For example: cgPoolN) and fill in the blanks for your vendor database. Click Continue.

   e. Test your connection to verify that you can successfully connect to your database.

   f. Create and deploy your new Connection Pool.

4. Update your data sources.

   a. From Services -> JDBC -> Data Sources, click on each data source and switch each to the newly created connection pool. Be sure to apply each change.

   b. Verify that each Data Source is changed by clicking on Data Sources and then verifying that Pool Name has been set to the new Connection Pool for each.

5. From Services -> JMS -> Stores -> cgJMSStore, switch cgJMSStore to use the new Connection Pool.
6. Stop your domain's WebLogic Server, then restart it.
7. In WebLogic Server Console, delete the original cgPool connection pool.

   a. Go to Services -> JDBC -> Connection Pools.

   b. Right-click on the cgPool connection pool and select Delete.

8. Move indexes to the WEBLOGIC_INDEX tablespace by executing rebuild_indexes.sql from SQLPLUS. This should be done while WebLogic Server is not running, and is recommended for performance.

Creating a Database for Behavior Tracking Events

You may want to store behavior tracking events in a different location than other WebLogic Portal database objects for increased performance. For more information about behavior tracking, see http://e-docs.bea.com/wlp/docs81/adminportal/help/SA_BehavTrackServ.html.

Note: By default, behavior tracking database objects are created in the same database as other WebLogic Portal database objects. You only need to following these steps if you are configuring a separate database for behavior tracking events.

1. Verify that you can connect to the target database, see step 1. in “Configuring an Oracle Database”.

2. From SQL*Plus, execute the bt_create_tablespaces.sql script. using the @ sign. For example:

   ```
   @bt_create_tablespaces.sql
   ```

3. From SQL*Plus, execute the bt_create_users.sql script using the @ sign. For example,

   ```
   @bt_create_users.sql
   ```

4. Open your domain’s `db_settings.properties` file for edit.

   a. Within the db_settings.properties file, uncomment the database settings for your new target database and update the following settings for your database:

   - `server=<SERVER_NAME>`
   - `dblogin=WEBLOGIC_EVENT`
   - `password=WEBLOGIC_EVENT`

   b. Within the db_settings.properties file, find the p13n_modules, portal_modules and netuix_modules lines at the top of the file.

   c. Copy these 3 lines and comment out the original settings by adding a # sign.

   d. Replace the original settings as follows:

   - Replace `p13n_modules=p13n au bt ds` with `p13n_modules=bt`.
   - Replace `portal_modules=cm wlcs wps collaboration sample_cm` with `portal_modules=`
   - Replace `netuix_modules=pf` with `netuix_modules=`.

   When you are finished, the section should look like this:

   ```
   #p13n_modules=p13n au bt ds
   ```

```
#portal_modules=cm wlcs wps collaboration sample_cm
#netuix_modules=pf
p13n_modules=bt
portal_modules=
netuix_modules=
```

   e. Save the changes to the `db_settings.properties` file.

5. Initialize the database with the new settings.

   a. For Windows, navigate to the `\\bea\user_projects\domains\portalDomain` directory, and double-click on the create_db.cmd file.

   b. For UNIX, navigate to the `\\bea\user_projects\domains\portalDomain` directory, run create_db.sh.

   c. Verify the results in the db.log file.

Note: If you are using the sample domain, run the create_db.cmd/sh file from the following directory: `\\bea\weblogic81\samples\domains\portal`.

6. Configure a connection pool to access your behavior tracking database and associate the p13n_tracking. Follow the steps in "Configure Your Domain's JDBC Driver Settings" on page 3-6.

Using a Sybase Database

This section describes the steps necessary to use a Sybase database with WebLogic Portal 8.1, and includes information on the following subjects:

- Understanding Database Configuration for WebLogic Platform
- Configuring a Sybase Database
- Creating Database Objects
- Configuring Your Domain's JDBC Driver Settings
- Creating a Database for Behavior Tracking Events

Typically, the steps in this chapter are performed by a database administrator.

Note: Review this entire chapter and any release notes before proceeding.

Understanding Database Configuration for WebLogic Platform

Typically, you use the WebLogic Configuration Wizard to configure and connect to the database that you will use to support WebLogic Platform. For more information about how to use the WebLogic Configuration Wizard, see http://edocs.bea.com/platform/docs81/confgwiz/index.html.

When using the Configuration Wizard to configure databases for use with WebLogic Platform, use the following steps.

1. Create your vendor database(s). If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

Note: If you are creating a separate database for behavior event tracking, see "Creating a Database for Behavior Tracking Events" on page 4-9.

2. Prepare the database for use with WebLogic Platform. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. After the database is configured, use the Configuration Wizard to create and load appropriate database objects and set JDBC driver settings at domain creation time.

Manually Configuring Databases

In some cases, you may wish to manually configure the database or manually modify your existing settings. For example, manual configuration should be used in the following cases:

- If your desired database was not configured via the WebLogic Configuration Wizard.
- If after running the Configuration Wizard you decided to have your domain point to a different database.
- If you would like to refresh your database with the base configuration data that comes with the product. (Note: All the database creation scripts first drop all database objects and recreate them, which means all data added since your original installation will be lost. Upon completion of the database creation scripts only the base configuration data that is needed for the product will exist.)
- When you want to create only a subset of Portal database objects, for example to create only Behavior Tracking database objects for a particular database.

Use the following steps when you need to manually configure a database.

1. Create your vendor database(s),see page 4-3. If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

2. Prepare the database for use with WebLogic Platform, see page 4-3. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.

3. Create database objects, see page 4-7. This creates proprietary database objects that are used by WebLogic. This is done by editing the database properties file provided by BEA.

4. Configure the JDBC settings for your database using WebLogic Console Server, see page 4-8.

Configuring a Sybase Database

Before following the steps outlined in this chapter, you need to have already defined your Sybase instance. Note the following when defining your Sybase instance.

- Be sure that you are using a supported version, see http://edocs.bea.com/platform/docs81/support/supp_plat.html#1085671.
- Define a pagesize of at least 8K to support WebLogic Portal's use of wide tables, wide columns, and larger indexes.

  If your Sybase instance uses 2k or 4k pages, create a new Sybase instance with an 8K page size. Sybase provides a migration utility to migrate data between servers of different page sizes. A technical white paper on the Sybase migration process can be found at: http://www.sybase.com/detail/printthis/1,6907,1021203,00.html.
- For WebLogic Portal 7.0 users who are upgrading to version 8.1, ensure that the following WebLogic Portal 7.0 script has been run: `bea\weblogic700\portal\db\sybase\125\migrate\migrate_to_125.sql`

1. Install the Sybase client software on the WebLogic Platform host and do the following:

   a. Configure it connect to the target Sybase instance.

   b. Verify that you can connect to the target instance via isql. For example,

   ```
   isql -Usa -Ppassword -SMySybase
   ```

2. Prepare the Sybase database. The database creation scripts install domain-specific tables. You should work with your database administrator to adjust the `SAMPLE` scripts, and to create the database schema owner users and tablespaces needed for your environment.

Notes: Multiple databases are required if you have multiple domains, or to run multiple environments using the same SQL Server instance (for example, if you want to run development and system test from a single SQL Server installation).

Be sure to back up your database(s) before installing any new database objects. See your database documentation for details.

a. Review and modify the provided sample scripts to suit your environment. These scripts are provided in WL_HOME/portal/db/sybase/125/admin.

Script Name	Description
create_devices.sql	Create database devices. **Usage Notes:** Database devices must be created by a user with system administrator privileges (normally the 'sa' user). "D:\DATAFILE" and "E:\LOGFILE" specifications in this script must be changed to reflect valid disk locations for your environment. Optimally, data and log devices would be placed on separate physical disks, which reside on separate controllers. Edits are required to change file sizes and device names. The following default names are used: • data device: WEBLOGIC_DATA • log device: WEBLOGIC_LOG
create_database.sql	Create the database and login. An alias is added to the dbo (database owner user) of the database. The devices created by create_devices.sql are used. **Usage Notes:** Edits are required to reflect name or size changes from create_devices.sql. Edits are required to change the default database name and/or database owner user. The following defaults are used: • data device: WEBLOGIC_DATA • log device: WEBLOGIC_LOG • database name: WEBLOGIC • database owner user: WEBLOGIC • password: WEBLOGIC If the database you are creating is a development database your database administrator may want to uncomment and set the "truncate log on checkpoint" database option. If your application will use WebLogic Workshop page flows or RowSet controls uncomment and set the 'DDL in transaction' to true option to true to allow database table create commands to work properly.

Script Name	Description
statistics_build.sql	Builds statistics.sql to update table and index statistics for the database optimizer. Statistics should be should be updated whenever any significant changes in database data occurs. Your database administrator should schedule update statistics to run periodically in your environment.
install_report_build.sql **install_report_static.sql**	Builds an informational installation report about the database objects created by the WEBLOGIC user.

Script Name	Description
bt_create_devices.sql	Creates behavior tracking database devices. **Usage Notes:** Database devices must be created by a user with system administrator privileges (normally the 'sa' user). "D:\DATAFILE" and "E:\LOGFILE" specifications in this script must be changed to reflect valid disk locations for your environment. Optimally, data and log devices would be placed on separate physical disks, which reside on separate controllers. Edits are required to change file sizes and device names. The following default names are used: • data device: WEBLOGIC_EVENT_DATA • log device: WEBLOGIC_EVENT_LOG
bt_create_database.sql	Create the WEBLOGIC_EVENT database and WEBLOGIC_EVENT database owner user login. An alias is created to make WEBLOGIC_EVENT dbo (database owner user) in the database. **Usage Notes:** Edits are required to change database names, database owner user and password. Edits are required to reflect valid disk locations for DATA and the LOG devices, or to adjust file sizes. DATA and LOG files should be placed on separate physical disks and away from any system database files. The following defaults are used: • data device: WEBLOGIC_EVENT_DATA • log device: WEBLOGIC_EVENT_LOG • database name: WEBLOGIC • database owner user: WEBLOGIC • password: WEBLOGIC

b. Run create_devices.sql as a user with system administrator privileges. For example,

```
isql -Usa -SMYSYBASE -e -icreate_devices.sql -ocreate_devices.log
```

c. Run create_database.sql via isql as a user with System Administrator privileges (i.e. the sa user).

```
isql -Usa -SMYSYBASE -e -icreate_database.sql -ocreate_database.log
```

Output from the above is written to the file specified after the "-o" parameter. The log file is stored in the same directory that the script resides. Verify that each log file contains no errors for database object creation.

d. Statistics and install report scripts will be run automatically by the create_db.cmd/.sh scripts. Ensure that your database administrator schedules update statistics to be run periodically for your WebLogic Portal database.

3. Follow the steps in "Creating Database Objects" on page 4-7.

Creating Database Objects

To create WebLogic Platform database objects, use the following steps:

1. Verify that you can connect to the target database. Use the following command syntax to verify that you can connect to the target database server using the default schema owner user created by running `create_database.sql`.

```
isql -UWEBLOGIC -SMYSYBASE
```

2. Open your domain's `db_settings.properties` file for edit and comment out the database setting for PointBase.

3. Uncomment the database settings for your new target database and update the following settings for your database:

   - `server=`
   - `dblogin=`
   - `password=`

4. Initialize the database with the new settings.

   a. Navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, and double-click on the `create_db.cmd` file.

   b. Verify the results in the `db.log` file.

Note: If you are using the sample domain, run the create_db.cmd/sh file from the following directory: `<WL_HOME>\samples\domains\portal`.

5. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 4-8.

Configuring Your Domain's JDBC Driver Settings

Note: These settings do not allow support for XA functionality. For instructions on enabling XA, see Chapter 7, “XA Support.”.

1. Start the WebLogic Server for your domain.
2. Log on to the WebLogic Server Console.
3. Configure your new connection pools.
   a. Go to Services -> JDBC -> Connection Pools.
   b. Click Configure a new Connection Pool.
   c. Select the appropriate Database Type and Non-XA Database Driver from the drop down list boxes and click Continue. See the Supported Configurations documentation for JDBC drivers supported by WebLogic Platform, http://edocs.bea.com/platform/docs81/support/supp_plat.html#1085671.
   d. Choose a name for the new Connection Pool (For example: cgPoolN) and fill in the blanks for your vendor database. Click Continue.
   e. Test your connection to verify that you can successfully connect to your database.
   f. Create and deploy your new Connection Pool.
4. Update your data sources.
   a. From Services -> JDBC -> Data Sources, click on each data source and switch each to the newly created connection pool. Be sure to apply each change.
   b. Verify that each Data Source is changed by clicking on Data Sources and then verifying that Pool Name has been set to the new Connection Pool for each.
5. From Services -> JMS -> Stores -> cgJMSStore, switch cgJMSStore to use the new Connection Pool.
6. Stop your domain’s WebLogic Server, then restart it.
7. In WebLogic Server Console, delete the original cgPool connection pool.
   a. Go to Services -> JDBC -> Connection Pools.
   b. Right-click on the cgPool connection pool and select Delete.

Creating a Database for Behavior Tracking Events

You may want to store behavior tracking events in a different location than other WebLogic Portal database objects for increased performance. For more information about behavior tracking, see http://e-docs.bea.com/wlp/docs81/adminportal/help/SA_BehavTrackServ.html.

Note: By default, behavior tracking database objects are created in the same database as other WebLogic Portal database objects. You only need to following these steps if you are configuring a separate database for behavior tracking events.

1. Verify that you can connect to the target Sybase instance, see step 1. in "Configuring a Sybase Database".

2. Run `bt_create_devices.sql` via isql as a user with system administrator privileges. For example,

```
isql -Usa -SMYSYBASE -e -ibt_create_devices.sql
-obt_create_devices.log
```

3. Run `bt_create_database.sql` via isql as a user with system administrator privileges (i.e. the sa user).

```
isql -Usa -SMYSYBASE -e -ibt_create_database.sql
-obt_create_database.log
```

4. Open your domain's `db_settings.properties` file for edit.

   a. Within the `db_settings.properties` file, uncomment the database settings for your new target database and update the following settings for your database:

   - `server=<SERVER_NAME>`
   - `dblogin=WEBLOGIC_EVENT`
   - `password=WEBLOGIC_EVENT`

5. Within the `db_settings.properties` file, find the p13n_modules, portal_modules and netuix_modules lines at the top of the file.

   b. Copy these 3 lines and comment out the original settings by adding a # sign.

   c. Replace the original settings as follows:

   – Replace `p13n_modules=p13n au bt ds` with `p13n_modules=bt`.

   – Replace `portal_modules=cm wlcs wps collaboration sample_cm` with `portal_modules=`

   – Replace `netuix_modules=pf` with `netuix_modules=`.

When you are finished, the section should look like this:

```
#p13n_modules=p13n au bt ds

#portal_modules=cm wlcs wps collaboration sample_cm

#netuix_modules=pf

p13n_modules=bt

portal_modules=

netuix_modules=
```

d. Save the changes to the `db_settings.properties` file.

6. Initialize the database with the new settings.

   a. Navigate to the `\\<BEA_HOME>\user_projects\domains\portalDomain` directory, and double-click on the create_db.cmd file.

   b. Verify the results in the db.log file.

Note: If you are using the sample domain, run the create_db.cmd file from the following directory: `\\<WL_HOME>\samples\domains\portal`.

7. Configure a connection pool to access your behavior tracking database and associate the p13n_tracking. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 4-8.

Using a DB2 Database

This section describes the steps necessary to use a DB2 database with WebLogic Portal 8.1, and includes information on the following subjects:

- Understanding Database Configuration for WebLogic Platform
- Configuring a DB2 Database
- Creating Database Objects
- Configuring Your Domain's JDBC Driver Settings
- Creating a Database for Behavior Tracking Events

Typically, the steps in this chapter should be performed by a database administrator.

Note: Review this entire chapter and any release notes before proceeding.

Understanding Database Configuration for WebLogic Platform

Typically, you use the WebLogic Configuration Wizard to configure and connect to the database that you will use to support WebLogic Platform. For more information about how to use the WebLogic Configuration Wizard, see http://edocs.bea.com/platform/docs81/confgwiz/index.html.

When using the Configuration Wizard to configure databases for use with WebLogic Platform, use the following steps.

1. Create your vendor database(s). If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.

Note: If you are creating a separate database for behavior event tracking, see "Creating a Database for Behavior Tracking Events" on page 5-8.

2. Prepare the database for use with WebLogic Platform. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.
3. After the database is configured, use the Configuration Wizard to create and load appropriate database objects and set JDBC driver settings at domain creation time.

Manually Configuring Databases

In some cases, you may wish to manually configure the database or manually modify your existing settings. For example, manual configuration should be used in the following cases:

- If your desired database was not configured via the WebLogic Configuration Wizard.
- If after running the Configuration Wizard you decided to have your domain point to a different database.
- If you would like to refresh your database with the base configuration data that comes with the product. (Note: All the database creation scripts first drop all database objects and recreate them, which means all data added since your original installation will be lost. Upon completion of the database creation scripts only the base configuration data that is needed for the product will exist.)
- When you want to create only a subset of Portal database objects, for example to create only Behavior Tracking database objects for a particular database.

Use the following steps when you need to manually configure a database.

1. Create your vendor database(s) see page 5-3. If you want to use behavior event tracking in a production environment, consider using a separate database for behavior event tracking.
2. Prepare the database for use with WebLogic Platform,see page 5-3. BEA provides several sample initialization scripts that need to be modified and run on the vendor database before using the database with WebLogic Platform.
3. Create database objects, see page 5-6. This creates proprietary database objects that are used by WebLogic. This is done by editing the database properties file provided by BEA.
4. Configure the JDBC settings for your database using WebLogic Console Server, see page 5-7.

Configuring a DB2 Database

Before following the steps outlined in this chapter, you need to have already defined your DB2 instance and the databases you need.

Be sure that you are using a supported version, see http://edocs.bea.com/platform/docs81/support/supp_plat.html#1085671.

1. Install the DB2 client software and configure it to connect to the target DB2 database. See your DB2 documentation for more information.
2. Verify that you can connect to the target database through the Command Line Processor (CLP).
3. Prepare the DB2 database. The database creation scripts will install domain-specific tables for each. It is recommended that you work with a database administrator to adjust the SAMPLE scripts, and to create the database objects (users, passwords, tablespaces, etc.) needed for your environment.

Notes: Multiple databases are required if you have multiple domains, or to run multiple environments using the same SQL Server instance (for example, if you want to run development and system test from a single SQL Server installation).

Be sure to back up your database(s) before installing any new database objects. See your database documentation for details.

a. Review and modify the provided sample scripts to suit your environment. The scripts are located in <*WL_HOME*>/portal/db/db/db2/8/admin.

The following table lists the script names and the usage notes for each script.

Script Name	Description
create_user.sql	Grant createtab, bindadd and connect DB2 privileges to the WEBLOGIC schema owner user. **Usage Notes:** Because IBM DB2 databases authenticate users via the operating system (OS), you need to create an OS user that will own database schema objects. Edits are required to change the schema owner user name. The default schema owner user name and password are the following: • schema owner user: WEBLOGIC • schema owner user password: WEBLOGIC
create_bufferpool.sql	Create an 8K bufferpool, if needed. **Usage Notes:** DB2 must be stopped and restarted to utilize new bufferpools. Edits are required to change the 8K bufferpool name. The default tablespace names and bufferpool (TEMPSPACE) names are the following: • WEBLOGIC_DATA_4K: Tables for WebLogic Portal and/or WebLogic Platform with rowsize smaller than 4K. • WEBLOGIC_DATA_8K: Tables for WebLogic Portal and/or WebLogic Platform with rowsize larger than 4K and smaller than 8K. • TEMPSPACE_8K: Temp space utilized by tables created in the WEBLOGIC_DATA_8K tablespace.
create_temp_tablespaces.sql	Create 4K and 8K regular tablespaces. **Usage Notes:** Edits are required to specify valid physical disk locations for your environment.
statistics_build.sql	Builds a file of "runstats" commands for each table that will compute database statistics needed for the database optimizer. Runstats should be run whenever any significant changes in database data occur. Your database administrator will typically schedule "runstats to run periodically in your environment.

Script Name	Description
install_report.sql	Builds an informational installation report about the database objects created in the WEBLOGIC schema.
bt_create_tablespace.sql	Creates the WEBLOGIC_EVENT_DATA tablespace. **Usage Notes:** Edits are required to modify the pathnames for the DATA_PATHNAME and INDEX_PATHNAME variables to match your local directory path structures. The default tablespace names and bufferpool (TEMPSPACE) names are the following: • WEBLOGIC_DATA_4K: Tables for WebLogic Portal and/or WebLogic Platform with rowsize smaller than 4K. • WEBLOGIC_DATA_8K: Tables for WebLogic Portal and/or WebLogic Platform with rowsize larger than 4K and smaller than 8K. • TEMPSPACE_8K: Temp space utilized by tables created in the WEBLOGIC_DATA_8K tablespace.
bt_create_users.sql	Creates the WEBLOGIC_EVENT schema owner user, establishes the user's password, default and temporary tablespaces and grants privileges to that user. **Usage Notes:** Edits are required to change the schema owner user name, password and tablespace names. The default schema owner user name and password are the following: • schema owner user: WEBLOGIC • schema owner user password: WEBLOGIC

b. From the CLP tool, navigate to the directory that contains the scripts. For example, type:

```
WL_HOME/portal/db/DB2/8/admin
```

c. From CLP, connect to the database you want to work with. For example, type:

```
Db2 connect to <DATABASE> user <USER_NAME> password <PASSWORD>
```

d. From CLP, run `create_bufferpool.sql`, if needed. You may not need to create a new 8K bufferpool if you already have one to utilitize. For example, type,

```
Db2 -tf create_bufferpool.sql -v
```

e. Restart your database instance.

f. From CLP, run `create_temp_tablespaces.sql`. For example, type,

```
Db2 -tf create_temp_tablespaces.sql -v
```

g. From CLP, run `create_user.sql`. For example, type,

```
Db2 -tf create_user.sql -v
```

4. Follow the steps in "Creating Database Objects" on page 5-6.

Creating Database Objects

To create BEA Platform database objects, use the following steps:

1. From DB2-CLP, use the following command to verify that you can connect to the target database server with a valid user ID and password:

   ```
   db2 connect to <DATABASE> user <USERNAME> password <PASSWORD>
   ```

2. Open your domain's `db_settings.properties` file for edit and comment out the database settings for PointBase.

3. In the `db_settings.properties` file for your domain, uncomment the database settings for your new target database and update the following settings for your database:

   - `server=`
   - `dblogin=`
   - `password=`

4. Create the database.

   a. For Windows, navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, and double-click on the create_db.cmd file.

   b. For UNIX, navigate to the `<BEA_HOME>\user_projects\domains\portalDomain` directory, run create_db.sh.

   c. Verify the results in the `db.log` file.

Note: If you are using the sample domain, run the create_db.cmd/sh file from the following directory: `<BEA_HOME>\weblogic81\samples\domains\portal`.

5. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 5-7.

Configuring Your Domain's JDBC Driver Settings

Note: These settings do not allow support for XA functionality. For instructions on enabling XA, consult "XA Support" on page 7-1.

1. Start the WebLogic Server for your domain.
2. Login to the WebLogic Server Console.
3. Configure your new connection pools.
    a. Go to Services -> JDBC -> Connection Pools.
    b. Click Configure a new Connection Pool.
    c. Select the appropriate Database Type and Non-XA Database Driver from the drop down list boxes and click Continue.
    d. Choose a name for the new Connection Pool (For example: cgPoolN) and fill in the blanks for your vendor database. Click Continue.
    e. Test your connection to verify that you can successfully connect to your database.
    f. Create and deploy your new Connection Pool.
4. Update your data sources.
    a. From Services -> JDBC -> Data Sources, click on each data source and switch each to the newly created connection pool. Be sure to apply each change.
    b. Verify that each Data Source is changed by clicking on Data Sources and then verifying that Pool Name has been set to the new Connection Pool for each.
5. From Services -> JMS -> Stores -> cgJMSStore, switch cgJMSStore to use the new Connection Pool.
6. Stop your domain's WebLogic Server, then restart it.
7. In the WebLogic Server Console, delete the original cgPool connection pool.
    a. Go to Services -> JDBC -> Connection Pools.
    b. Right-click on the cgPool connection pool and select Delete.

Creating a Database for Behavior Tracking Events

You may want to store behavior tracking events in a different location than other WebLogic Portal database objects for increased performance. For more information about behavior tracking, see http://e-docs.bea.com/wlp/docs81/adminportal/help/SA_BehavTrackServ.html.

Note: By default, behavior tracking database objects are created in the same database as other WebLogic Portal database objects. You only need to following these steps if you are configuring a separate database for behavior tracking events.

1. Verify that you can connect to the target database, see step 1. in "Configuring a DB2 Database".
2. From CLP, run the bt_create_temp_tablespaces.sql script. For example, type,

   ```
   Db2 -tf bt_reate_temp_tablespaces.sql -v
   ```

3. From CLP, run bt_create_user.sql. For example, type,

   ```
   Db2 -tf bt_create_user.sql -v
   ```

4. Open your domain's `db_settings.properties` file for edit.
   a. Within the db_settings.properties file, uncomment the database settings for your new target database and update the following settings for your database:
      - `server=<SERVER_NAME>`
      - `dblogin=WEBLOGIC_EVENT`
      - `password=WEBLOGIC_EVENT`

   b. Within the db_settings.properties file, find the p13n_modules, portal_modules and netuix_modules lines at the top of the file.

   c. Copy these 3 lines and comment out the original settings by adding a # sign.

   d. Replace the original settings as follows:
      - Replace `p13n_modules=p13n au bt ds` with `p13n_modules=bt`.
      - Replace `portal_modules=cm wlcs wps collaboration sample_cm` with `portal_modules=`
      - Replace `netuix_modules=pf` with `netuix_modules=`.

      When you are finished, the section should look like this:

      ```
      #p13n_modules=p13n au bt ds
      ```

```
#portal_modules=cm wlcs wps collaboration sample_cm
#netuix_modules=pf
p13n_modules=bt
portal_modules=
netuix_modules=
```

e. Save the changes to the `db_settings.properties` file.Save the changes to the db_settings.properties file.

5. Initialize the database with the new settings.

   a. For Windows, navigate to the `\\bea\user_projects\domains\portalDomain` directory, and double-click on the create_db.cmd file.

   b. For UNIX, navigate to the `\\bea\user_projects\domains\portalDomain` directory, run create_db.sh.

   c. Verify the results in the db.log file.

Note: If you are using the sample domain, run the create_db.cmd/sh file from the following directory: `\\bea\weblogic81\samples\domains\portal`.

6. Configure a connection pool to access your behavior tracking database and associate the p13n_tracking. Follow the steps in "Configuring Your Domain's JDBC Driver Settings" on page 5-7.

The Data Dictionary

This section describes the database objects for each component of WebLogic Portal. The information in this section is collectively known as the data dictionary.

Information Provided

For each component of WebLogic Portal, the following information is provided:

- An entity-relationship diagram
- A detailed description of each database table, including:

  Table Name
  The predefined name for the Table

  Table Description
  A detailed description of the contents and purpose for the table in WebLogic Portal database schema.

  Column Name
  The predefined name for the column

  Data Type
  The predefined characteristics for the column.

Note: Data types vary slightly by DBMS. For instance, Columns defined as BLOB data types in Oracle, DB2, and PointBase would be defined as TEXT columns in Microsoft SQL Server and Sybase.

Null Value

Whether or not null values are allowed to be stored for the column.

Column Description

A detailed description of the contents and purpose for the column including Primary Key (PK-) and Foreign Key (FK-) designations.

Portal Database Components Covered

This section includes information on the following subjects:

- Behavior Tracking Database Objects
- Commerce Services Database Objects
- Product Catalog Database Tables
- Order and Discount Database Objects
- Personalization Database Objects
- Data Synchronization Database Objects
- WebLogic Portal Services Database Objects
- Portal Framework Database Objects
- Portal Framework Database Objects
- Content Management Database Objects
- Localization Database Objects
- Tracked Anonymous User Database Objects

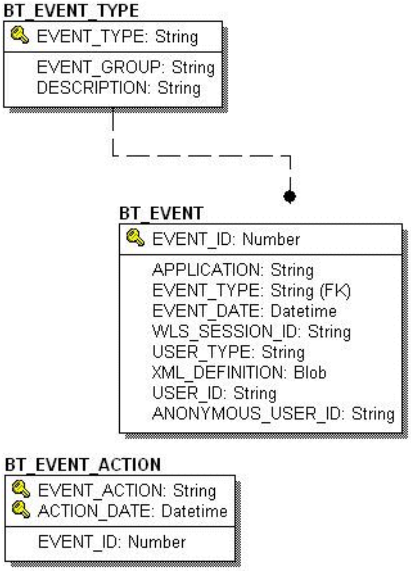
Note: Appendix A, "WebLogic Portal DDL Modules" identifies the filenames and location of DDL (database definition language) files for each set of Portal Database objects.

Behavior Tracking Database Objects

To record how online visitors are interacting with your Web site, you can record event information to a database. These kinds of events are called Behavior Tracking events. Analytics Marketing systems can then analyze these events offline to evaluate visitor behavior and transactional data. You can use the knowledge gained from analysis to create and optimize personalization rules, set up product offers, and develop interactive marketing campaigns. This section describes the requirements and database objects needed to log event data for analytical use.

Three tables are provided for the Behavior Tracking data. The BT_EVENT table stores all event data. The BT_EVENT_ACTION table logs actions used by third-party vendors against the recorded event data, and the BT_EVENT_TYPE table references event types and categories in the EVENT table. Figure 6-1 shows an entity-relation diagram for the WebLogic Portal Behavior Tracking Database objects.

Figure 6-1 Entity-Relation Diagram for the Behavior Tracking Database



The BT_EVENT_TYPE Database Table

This table references event types and categories in the BT_EVENT table. This table is static.

Table 6-1 BT_EVENT_TYPE Table Metadata

Column Name	Data Type	Null Value	Description
EVENT_TYPE	VARCHAR(30)	Not Null	PK - A unique, system-generated number used as the record ID.

Table 6-1 BT_EVENT_TYPE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
EVENT_GROUP	VARCHAR(10)	Not Null	The event category group associated with the event type.
DESCRIPTION	VARCHAR(50)	Null	A description of the EVENT_TYPE.

To record custom events, you must create an entry in this table. If a custom event does not have a record in this table, you cannot persist it to the BT_EVENT table.

The BT_EVENT Database Table

This table stores all Behavior Tracking event data.

Table 6-2 The BT EVENT Table Metadata

Column Name	Data Type	Null Value	Description
EVENT_ID	NUMBER	Not Null	PK - A unique, system-generated number used as the record ID.
APPLICATION	VARCHAR (30)	Not Null	The application that created the event.
EVENT_TYPE	VARCHAR(30)	Not Null	FK - Set to BT_EVENT_TYPE. A string identifier showing which event was fired.
EVENT_DATE	DATE	Not Null	The date and time of the event.
WLS_SESSION_ID	VARCHAR(254)	Not Null	A unique, WebLogic Server-generated number assigned to the session.
XML_DEFINITION	CLOB	Null	An XML document that contains the specific event information for each event type. It is stored as a CLOB (Character Large Object). See Table 6-3.
USER_ID	VARCHAR(50)	Null	The user ID associated with the session and event. If the user has not logged in this column will be null.

As shown in Table 6-2, the BT_EVENT table has six columns; each column corresponds to a specific event element. Five of the EVENT table's columns contain data common to every event type. The XML_DEFINITION column contains all information from these five columns plus event data that is unique to each event type. An XML document is created specifically for each event type. The data elements corresponding to each event type are captured in the XML_DEFINITION column of the EVENT table. These elements are listed in Table 6-3.

Table 6-3 XML_DEFINITION Data Elements

Event	Data Element
AddToCartEvent	application event-date event-type session-id user-id sku quantity unit-list-price currency application-name
BuyEvent	application event-date event-type session-id user-id sku quantity unit-price currency application-name order-line-id
CampaignUserActivityEvent	application event-date event-type session-id user-id campaign-id scenario-id

Table 6-3 XML_DEFINITION Data Elements (Continued)

Event	Data Element

ClickCampaignEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`
	`campaign-id`
	`scenario-id`
	`application-name`
	`placeholder-id`

ClickContentEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`

ClickProductEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`
	`sku`
	`category-id`
	`application-name`

DisplayCampaignEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`
	`campaign-id`
	`scenario-id`
	`application-name`
	`placeholder-id`

Table 6-3 XML_DEFINITION Data Elements (Continued)

Event	Data Element

DisplayContentEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`

DisplayProductEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`document-type`
	`document-id`
	`sku`
	`category-id`
	`application-name`

PurchaseCartEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`total-price`
	`order-id`
	`currency`
	`application-name`

RemoveFromCartEvent	`application`
	`event-date`
	`event-type`
	`session-id`
	`user-id`
	`sku`
	`quantity`
	`unit-price`
	`currency`
	`application-name`

Table 6-3 XML_DEFINITION Data Elements (Continued)

Event	Data Element
RuleEvent	application event-date event-type session-id user-id ruleset-name rule-name
SessionBeginEvent	application event-date event-type session-id user-id
SessionEndEvent	application event-date event-type session-id user-id
SessionLoginEvent	application event-date event-type session-id user-id
UserRegistrationEvent	application event-date event-type session-id user-id

The BT_EVENT_ACTION Database Table

This table logs actions used by third-party vendors against the recorded event data.

Table 6-4 BT_EVENT_ACTION Table Metadata

Column Name	Data Type	Null Value	Description
EVENT_ACTION	VARCHAR(30)	Not Null	The event action taken such as BEGIN EXPORT or END EXPORT. This field is one of the table's primary keys.
ACTION_DATE	DATE	Not Null	The date and time of the event. This field is one of the table's primary keys.
EVENT_ID	NUMBER	Null	The ID of the event that corresponds with the event action taken.

Commerce Services Database Objects

The metadata for items in Commerce services Product Catalog are based on the Dublin Core Metadata Open Standard. This standard offers a number of advantages for a Web-based catalog. For more information about the Dublin Core Metadata Open Standard, please see http://purl.org/dc.

Figure 6-2 and Figure 6-3 show the Entity-Relation for the WebLogic Portal Commerce services core Product Catalog database objects.

Figure 6-2 Entity-Relation Diagram for the Core Product Catalog Tables

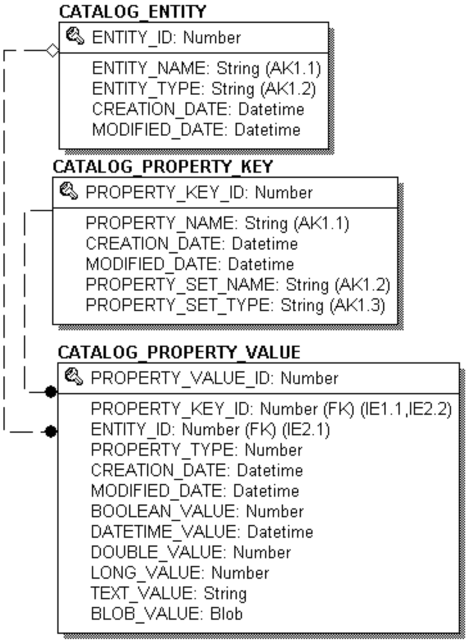
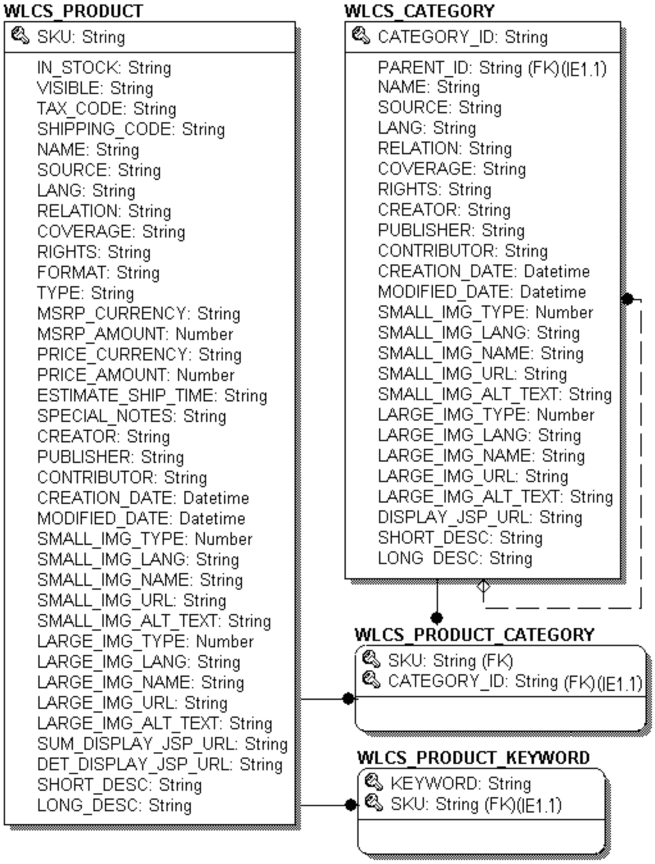


Figure 6-3 Entity-Relation Diagram for the Core Product Catalog Tables - continued



Product Catalog Database Tables

The following tables compose the product catalog database.

- The CATALOG_ENTITY Database Table
- The CATALOG_PROPERTY_KEY Database Table
- The CATALOG_PROPERTY_VALUE Database Table
- The WLCS_CATEGORY Database Table

- The WLCS_PRODUCT Database Table
- The WLCS_PRODUCT_CATEGORY Database Table
- The WLCS_PRODUCT_KEYWORD Database Table

The CATALOG_ENTITY Database Table

Unique identification numbers for configurable entities.

Table 6-5 CATALOG_ENTITY Table Metadata

Column Name	Data Type	Null Value	Description
ENTITY_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as a record identifier.
ENTITY_NAME	VARCHAR(200)	Not Null	The name of the entity.
ENTITY_TYPE	VARCHAR(100)	Not Null	The type of entity (e.g., User, Group, etc.)
CREATION_DATE	DATE	Not Null	The time and date the record was created.
MODIFIED_DATE	DATE	Not Null	The time and date the record was last modified.

The CATALOG_PROPERTY_KEY Database Table

Unique identification numbers for scoped property names that are associated with configurable entities.

Table 6-6 CATALOG_PROPERTY_KEY Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_KEY_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as a record identifier.
PROPERTY_NAME	VARCHAR(100)	Not Null	The name of the property (formerly PROPERTY_NAME from the WLCS_PROP_ID table).
CREATION_DATE	DATE	Not Null	The time and date the record was created.
MODIFIED_DATE	DATE	Not Null	The time and date the record was last modified.

Table 6-6 CATALOG_PROPERTY_KEY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
PROPERTY_SET_NAME	VARCHAR(100)	Null	The name of the property set (formerly the SCOPE_NAME from WLCS_PROP_ID).
PROPERTY_SET_TYPE	VARCHAR(100)	Null	The type of property set (for example, USER)

The CATALOG_PROPERTY_VALUE Database Table

Boolean, timestamp, float, integer, text, and user-defined (object) property values that are associated with configurable entities.

Table 6-7 CATALOG_PROPERTY_VALUE Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_VALUE_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as a record identifier.
PROPERTY_KEY_ID	NUMBER(15)	Not Null	FK - A system-generated value and foreign key to the PROPERTY_KEY column.
ENTITY_ID	NUMBER(15)	Not Null	FK - A system-generated value and foreign key to the ENTITY column.
PROPERTY_TYPE	NUMBER(1)	Not Null	Valid entries are: 0=Boolean, 1=Integer, 2=Float, 3=Text, 4=Date and Time, 5=User-Defined (BLOB)
CREATION_DATE	DATE	Not Null	The time and date the record was created.
MODIFIED_DATE	DATE	Not Null	The time and date the record was last modified.
BOOLEAN_VALUE	NUMBER(1)	Null	The value for each boolean property identifier.
DATETIME_VALUE	DATE	Null	The value for each date and time property identifier.

Table 6-7 CATALOG_PROPERTY_VALUE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
DOUBLE_VALUE	NUMBER	Null	The value associated with each float property identifier.
LONG_VALUE	NUMBER(20)	Null	The value associated with the integer property.
TEXT_VALUE	VARCHAR(254)	Null	The value associated with the text property.
BLOB_VALUE	BLOB	Null	The value associated with the user-defined property.

The WLCS_CATEGORY Database Table

Categories in the Commerce database. The descriptions shown in the table reflect the “recommended best practice” for the use of that field by the Dublin Core standard.

Table 6-8 WLCS_CATEGORY Table Metadata

Column Name	Data Type	Null Value	Descriptions
CATEGORY_ID	VARCHAR(20)	Not Null	PK - A unique identifier for a category; the primary key for this table. This field cannot be NULL. All other fields in the WLCS_CATEGORY table can be NULL.
PARENT_ID	VARCHAR(20)	Null	The value of the CATEGORY_ID of the parent category in the hierarchy of categories that comprise your product catalog. If this is a top-level user-defined category, the PARENT_ID will be com.beasys.ROOT.
NAME	VARCHAR(50)	Null	The name of the category in the product catalog.
SOURCE	VARCHAR(30)	Null	A reference to a category from which the present category is derived.

Table 6-8 WLCS_CATEGORY Table Metadata (Continued)

Column Name	Data Type	Null Value	Descriptions
LANG	VARCHAR(30)	Null	A language of the intellectual content of the category. The recommended best practice for the values of the language element is defined by RFC 1766, which includes a two-letter Language Code (taken from the ISO 639 standard), such as: en for English; fr for French, or de for German. The language code can, optionally, be followed by a two-letter Country Code (taken from the ISO 3166 standard [ISO3166]). For example, en-uk for English used in the United Kingdom.
RELATION	VARCHAR(30)	Null	A reference to a related category.
COVERAGE	VARCHAR(30)	Null	The extent or scope of the content of the category.
RIGHTS	VARCHAR(30)	Null	Information about rights held in and over the category.
CREATOR	VARCHAR(50)	Null	An entity primarily responsible for making the content of the category.
PUBLISHER	VARCHAR(50)	Null	An entity responsible for making the category available.
CONTRIBUTOR	VARCHAR(50)	Null	An entity responsible for making contributions to the content of the category.
CREATION_DATE	DATE	Null	A date associated with an event in the life cycle of the category. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 and follows the YYYY-MM-DD format.

Table 6-8 WLCS_CATEGORY Table Metadata (Continued)

Column Name	Data Type	Null Value	Descriptions
MODIFIED_DATE	DATE	Null	A date associated with an event in the life cycle of the category, such as an update or insert by the DBLoader program that is provided with the Commerce services. The recommended best practice for encoding the date value is defined in a profile of ISO 8601 and follows the YYYY-MM-DD format.
SMALL_IMG_TYPE	NUMBER(3)	Null	A type field of your own design that relates to the graphic. For example, you can implement your own numbering scheme, such as: 0 = display a low resolution graphic for users with low bandwidth. 1 = display a high resolution graphic for users with high bandwidth.
SMALL_IMG_LANG	VARCHAR(30)	Null	The language of the thumbnail image for the category. For related information, see the description of the LANG column.
SMALL_IMG_NAME	VARCHAR(50)	Null	The name of the thumbnail image for the category.
SMALL_IMG_URL	VARCHAR(254)	Null	The URL of the thumbnail image for the category.
SMALL_IMG_ALT_TEXT	VARCHAR(254)	Null	The alternate text to display when the user has their cursor over the thumbnail image for the category, or if they have disabled the display of graphics in their browser settings.

Table 6-8 WLCS_CATEGORY Table Metadata (Continued)

Column Name	Data Type	Null Value	Descriptions
LARGE_IMG_TYPE	NUMBER(3)	Null	A type field of your own design that relates to the graphic. For example, you can implement your own numbering scheme, such as: 0 = display a low resolution graphic for users with low bandwidth. 1 = display a high resolution graphic for users with high bandwidth.
LARGE_IMG_LANG	VARCHAR(30)	Null	The language of the full-size image for the category. For related information, see the description of the LANG column.
LARGE_IMG_NAME	VARCHAR(50)	Null	The name of the full-size image for the category.
LARGE_IMG_URL	VARCHAR(254)	Null	The URL of the full-size image for the category.
LARGE_IMG_ALT_TEXT	VARCHAR(254)	Null	The alternate text to display when the user has their cursor over the full-size image for the category, or if they have disabled the display of graphics in their browser settings.
DISPLAY_JSP_URL	VARCHAR(254)	Null	The URL to the JSP used to display the category. For example: /commerce/catalog/includes/category.jsp
SHORT_DESC	VARCHAR(50)	Null	A short description of the content of the category.
LONG_DESC	VARCHAR(254)	Null	A long description of the content of the category.

The WLCS_PRODUCT Database Table

Item records in the Commerce database.

Table 6-9 WLCS_PRODUCT Table Metadata

Column Name	Data Type	Null Value	Description
SKU	VARCHAR(40)	Not Null	PK - A unique identifier (the “Stock Keeping Unit,” or SKU) for a product item. This field is the table’s primary key and cannot be NULL. All other fields in the WLCS_PRODUCT table can be NULL.
IN_STOCK	VARCHAR(1)	Null	A flag to indicate whether the product item is in stock. 0 equates to false, 1 equates to true.
VISIBLE	VARCHAR(1)	Null	Indicates whether the item should be displayed to the user. Enter 1 if visible or 0 if not visible. If not specified in the database, the default is 1.
TAX_CODE	VARCHAR(10)	Null	The code used by the TAXWARE system to identify the specific tax category to which this item belongs.
SHIPPING_CODE	VARCHAR(10)	Null	The code used by the shipping company for this item.
NAME	VARCHAR(100)	Null	A name given to the product item.
SOURCE	VARCHAR(30)	Null	A reference to another product item from which the present item is derived.

Table 6-9 WLCS_PRODUCT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
LANG	VARCHAR(30)	Null	A language of the intellectual content of the category. The recommended best practice for the values of the language element is defined by RFC 1766, which includes a two-letter Language Code (taken from the ISO 639 standard), such as: en for English; fr for French, or de for German. The language code can, optionally, be followed by a two-letter Country Code (taken from the ISO 3166 standard [ISO3166]). For example, en-uk for English used in the United Kingdom.
RELATION	VARCHAR(30)	Null	A reference to a related product item.
COVERAGE	VARCHAR(30)	Null	The extent or scope of the content of the product item.
RIGHTS	VARCHAR(30)	Null	Information about rights held in and over the item.
FORMAT	VARCHAR(30)	Null	The physical or digital manifestation of the item.
TYPE	VARCHAR(30)	Null	The nature or genre of the content of the item.
MSRP_CURRENCY	VARCHAR(30)	Null	The currency type of the manufacturer's recommended price.
MSRP_AMOUNT	NUMBER(16,4)	Null	The manufacturer's recommended price.
PRICE_CURRENCY	VARCHAR(30)	Null	The currency type of our catalog price for this item.
PRICE_AMOUNT	NUMBER(16,4)	Null	Our current price for this item in the catalog.
ESTIMATE_SHIP_TIME	VARCHAR(100)	Null	Inventory: number of days/weeks before the item can be shipped.
SPECIAL_NOTES	VARCHAR(100)	Null	Inventory related message to display with the item.

Table 6-9 WLCS_PRODUCT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATOR	VARCHAR(50)	Null	An entity primarily responsible for making the content of the product item.
PUBLISHER	VARCHAR(50)	Null	An entity responsible for making the product item available.
CONTRIBUTOR	VARCHAR(50)	Null	An entity responsible for making contributions to the content of the product item.
CREATION_DATE	DATE	Null	A date associated with an event in the life cycle of the product item. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 and follows the YYYY-MM-DD format.
MODIFIED_DATE	DATE	Null	A date associated with an event in the life cycle of the item, such as an update or insert by the DBLoader program that is provided with the Commerce services. The recommended best practice for encoding the date value is defined in a profile of ISO 8601 and follows the YYYY-MM-DD format.
SMALL_IMG_TYPE	NUMBER(3)	Null	A type field of your own design that relates to the graphic. For example, you can implement your own numbering scheme, such as: 0 = display a low resolution graphic for users with low bandwidth. 1 = display a high resolution graphic for users with high bandwidth.
SMALL_IMG_LANG	VARCHAR(30)	Null	The language of the thumbnail image for the item. For related information, see the description of the LANG column.
SMALL_IMG_NAME	VARCHAR(50)	Null	The name of the thumbnail image for the item.

Table 6-9 WLCS_PRODUCT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
SMALL_IMG_URL	VARCHAR(254)	Null	The URL of the thumbnail image for the category.
SMALL_IMG_ALT_TEXT	VARCHAR(254)	Null	The alternate text to display when the user has their cursor over the thumbnail image for the item, or if they have disabled the display of graphics in their browser settings.
LARGE_IMG_TYPE	NUMBER(3)	Null	A type field of your own design that relates to the graphic. For example, you can implement your own numbering scheme, such as: 0 = display a low resolution graphic for users with low bandwidth. 1 = display a high resolution graphic for users with high bandwidth.
LARGE_IMG_LANG	VARCHAR(30)	Null	The language of the full-size image for the item. For related information, see the description of the LANG column.
LARGE_IMG_NAME	VARCHAR(50)	Null	The name of the full-size image for the item.
LARGE_IMG_URL	VARCHAR(254)	Null	The URL of the full-size image for the item.
LARGE_IMG_ALT_TEXT	VARCHAR(254)	Null	The alternate text to display when the user has their cursor over the full-size image of the item, or if they have disabled the display of graphics in their browser settings.
SUM_DISPLAY_JSP_URL	VARCHAR(254)	Null	The URL to the JSP used to display the item in summary form. For example: /commerce/catalog/includes/itemsummary.jsp

Table 6-9 WLCS_PRODUCT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
DET_DISPLAY_JSP_URL	VARCHAR(254)	Null	The URL to the JSP used to display the item in detailed form. For example: /commerce/catalog/includes/ itemdetails.jsp
SHORT_DESC	VARCHAR(254)	Null	A short description of the content of the product item.
LONG_DESC	VARCHAR(2000)	Null	A long description of the content of the product item.

The WLCS_PRODUCT_CATEGORY Database Table

Shows which product items are associated with product categories.

Table 6-10 WLCS_PRODUCT_CATEGORY Table Metadata

Column Name	Data Type	Null Value	Description
SKU	VARCHAR(40)	Not Null	PK - A unique identifier (the “Stock Keeping Unit,” or SKU) for an item. FK to WLCS_PRODUCT.
CATEGORY_ID	VARCHAR(20)	Not Null	PK - A unique identifier for a category. FK to WLCS_CATEGORY.

The WLCS_PRODUCT_KEYWORD Database Table

Keywords that you associate with each product item. The keywords enable rapid retrieval of item records via the search functions on the Web site’s pages or Administration pages.

Table 6-11 WLCS_PRODUCT_KEYWORD Table Metadata

Column Name	Data Type	Null Value	Description
KEYWORD	VARCHAR(30)	Not Null	PK - Contains a keyword that you associate with the product item assigned to the unique SKU. Recommendation—for a given item, select a value from a controlled vocabulary or formal classification scheme implemented in your company.
SKU	VARCHAR(40)	Not Null	PK - A unique identifier (the "Stock Keeping Unit," or SKU) for an item. FK to WLCS_PRODUCT.

Order and Discount Database Objects

Figure 6-4 and Figure 6-5 show the Entity-Relation diagram for the WebLogic Portal order and discount objects.

Figure 6-4 Entity-Relation Diagram for the Commerce Tables

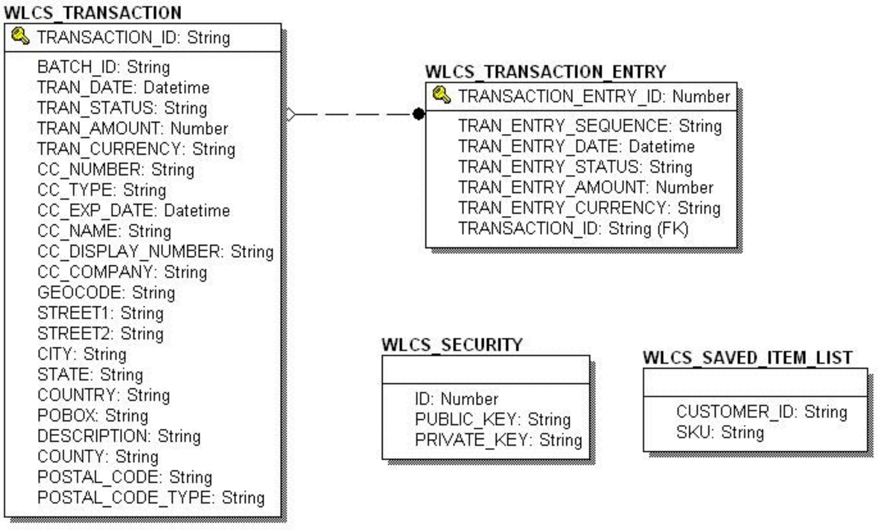
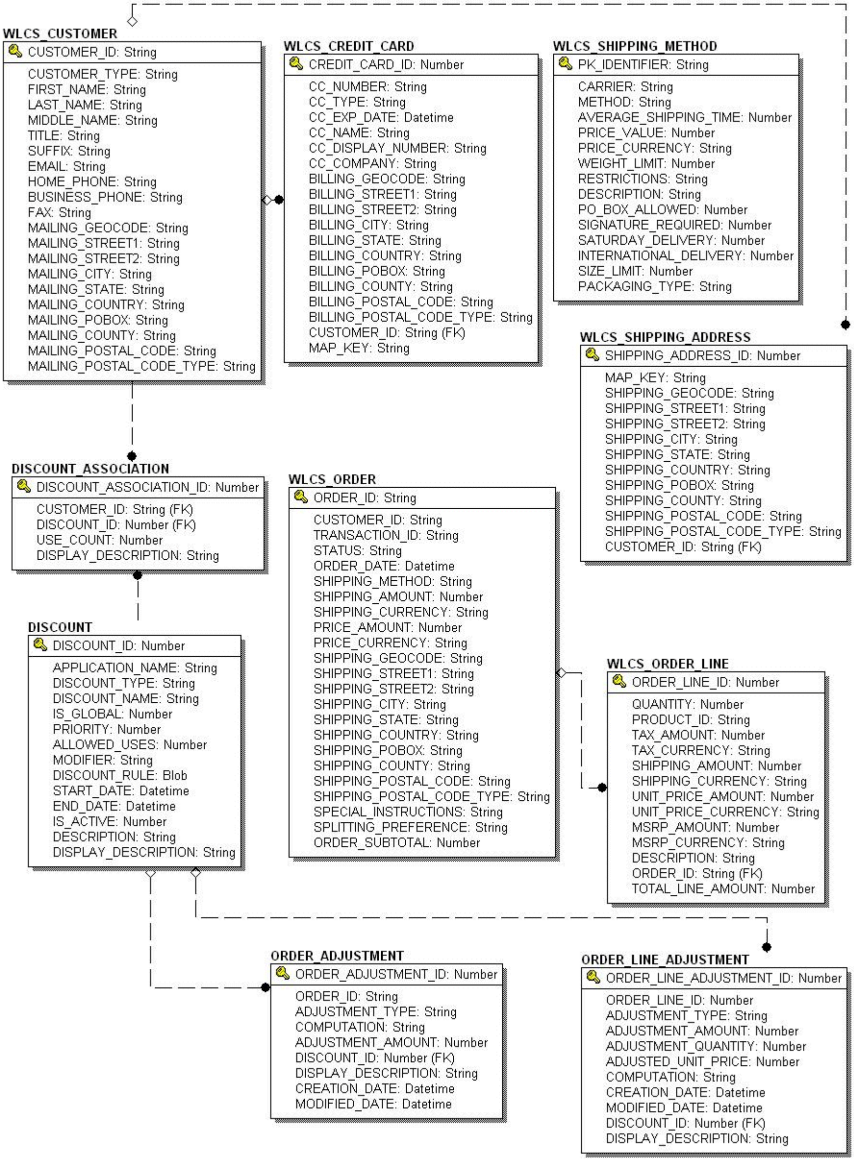


Figure 6-5 Entity-Relation Diagram for the Commerce Tables - continued



The Order Processing Data Dictionary Tables

The Commerce services order management system has the following tables:

- The DISCOUNT Database Table
- The DISCOUNT_ASSOCIATION Database Table
- The ORDER_ADJUSTMENT Database Table
- The ORDER_LINE_ADJUSTMENT Database Table
- The WLCS_CREDIT_CARD Database Table
- The WLCS_CUSTOMER Database Table
- The WLCS_ORDER Database Table
- The WLCS_ORDER_LINE Database Table
- The WLCS_SAVED_ITEM_LIST Database Table
- The WLCS_SECURITY Database Table
- The WLCS_SHIPPING_ADDRESS Database Table
- The WLCS_SHIPPING_METHOD Database Table
- The WLCS_TRANSACTION Database Table
- The WLCS_TRANSACTION_ENTRY Database Table

The DISCOUNT Database Table

One or more discount records for every DISCOUNT_SET record.

Table 6-12 DISCOUNT

Column Name	Data Type	Null Value	Description
DISCOUNT_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
APPLICATION_NAME	VARCHAR(100)	Not Null	FK—foreign key to the DISCOUNT_SET table.

Table 6-12 DISCOUNT (Continued)

Column Name	Data Type	Null Value	Description
DISCOUNT_TYPE	VARCHAR(10)	Not Null	The type of discount offered. It is used for an *order* or for an *order line item*.
DISCOUNT_NAME	VARCHAR(254)	Not Null	The name of the discount.
IS_GLOBAL	NUMBER(1)	Not Null	A flag showing whether or not this discount can be used globally.
PRIORITY	NUMBER(3)	Not Null	The level of priority this discount has over other discounts.
ALLOWED_USERS	NUMBER(10)	Not Null	The number of times the discount may be used.
MODIFIER	VARCHAR(254)	Not Null	Describes the actual discount to be applied. This is XML.
DISCOUNT_RULE	CLOB	Not Null	The method used to select items for discount. This is XML.
START_DATE	DATE	Not Null	The starting date and time of the discount
END_DATE	DATE	Not Null	The ending date and time of the discount.
IS_ACTIVE	NUMBER(1)	Not Null	A flag that determines whether the discount is active or not. Active=1, Not active=0
DESCRIPTION	VARCHAR(254)	Null	The discount description.
DISPLAY_DESCRIPTION	VARCHAR(254)	Null	The discount description used for display purposes only.

The DISCOUNT_ASSOCIATION Database Table

Associates each customer with a discount and maintains information regarding the times the customer has used each discount.

Table 6-13 DISCOUNT_ASSOCIATION

Column Name	Data Type	Null Value	Description
DISCOUNT_ASSOCIATION_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
CUSTOMER_ID	VARCHAR(20)	Not Null	FK—foreign key to the DISCOUNT_SET table.
DISCOUNT_ID	NUMBER(15)	Not Null	FK—foreign key to the DISCOUNT_SET table.
USE_COUNT	NUMBER(10)	Not Null	The number of times the discount has been used.
DISPLAY_DESCRIPTION	VARCHAR(254)	Null	The discount description used for display purposes only.

The ORDER_ADJUSTMENT Database Table

Information about a discount taken at the order level (for example, $20.00 off any order between 1/1/02 and 1/31/02.)

Table 6-14 ORDER_ADJUSTMENT

Column Name	Data Type	Null Value	Description
ORDER_ADJUSTMENT_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
ORDER_ID	VARCHAR(20)	Not Null	FK—foreign key to the DISCOUNT_SET table.
ADJUSTMENT_TYPE	VARCHAR(20)	Null	The type of adjustment being made to the order line item (e.g., order line discount, shipping discount, etc.)
COMPUTATION	VARCHAR(254)	Not Null	The number of times the discount has been used.

Table 6-14 ORDER_ADJUSTMENT (Continued)

Column Name	Data Type	Null Value	Description
ADJUSTMENT_AMOUNT	NUMBER(16,4)	Not Null	The discount description used for display purposes only.
DISCOUNT_ID	NUMBER(15)	Null	FK—foreign key to the DISCOUNT table.
DISPLAY_DESCRIPTION	VARCHAR(254)	Null	The description used for display purposes only. Depending on the nature of the discount, the DISPLAY_DESCRIPTION is generated from either the Discount service or Campaign service.
CREATION_DATE	DATE	Not Null	The date and time the order adjustment was created.
MODIFIED_DATE	DATE	Null	The date and time the order adjustment record was last modified.

The ORDER_LINE_ADJUSTMENT Database Table

Information about a discount taken at the order line item level (for example, 10% off SKU “Power Drill”).

Table 6-15 ORDER_LINE_ADJUSTMENT Table Metadata

Column Name	Data Type	Null Value	Description
ORDER_LINE_ADJUSTMENT_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
ORDER_LINE_ID	NUMBER(15)	Not Null	A unique identifier for each line in a customer’s shopping cart. This field is the table’s primary key and cannot be NULL. All other fields in the WLCS_ORDERLINE table can be NULL.
ADJUSTMENT_TYPE	VARCHAR(20)	Null	The type of adjustment being made to the order line item (e.g., order line discount, shipping discount, etc.)
ADJUSTMENT_AMOUNT	NUMBER(16,4)	Not Null	The dollar amount of the adjustment.
ADJUSTMENT_QUANTITY	NUMBER(16,4)	Not Null	The quantity amount for the adjustment.

Table 6-15 ORDER_LINE_ADJUSTMENT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
ADJUSTED_UNIT_PRICE	NUMBER(16,4)	Not Null	The adjusted unit price of the specific line item.
COMPUTATION	VARCHAR(254)	Not Null	The computation for determining ADJUSTED_UNIT_PRICE.
CREATION_DATE	DATE	Not Null	The date and time the adjustment record was created.
MODIFIED_DATE	DATE	Null	The date and time the adjustment record was last modified.
DISCOUNT_ID	NUMBER(15)	Null	FK—a foreign key to the discount used from the DISCOUNT table.
DISPLAY_DESCRIPTION	VARCHAR(254)	Null	The adjustment description used for display purposes.

The WLCS_CREDIT_CARD Database Table

Information related to a customer's credit card(s) in the order processing database.

Table 6-16 WLCS_CREDIT_CARD Table Metadata

Column Name	Data Type	Null Value	Description and Recommendations
CREDIT_CARD_ID	NUMBER(15)	Not Null	A unique identifier for the credit card. This field is the table's primary key and cannot be NULL. All other fields in the WLCS_CREDIT_CARD table can be NULL.
CC_NUMBER	VARCHAR(200)	Null	The customer's credit card number. This is encrypted if is.encryption.enable is set to true in the weblogiccommerce.properties file.
CC_TYPE	VARCHAR(20)	Null	The customer's credit card type, such as VISA or MasterCard.

Table 6-16 WLCS_CREDIT_CARD Table Metadata (Continued)

Column Name	Data Type	Null Value	Description and Recommendations
CC_EXP_DATE	DATE	Null	The expiration date on the customer's credit card.
CC_NAME	VARCHAR(50)	Null	The credit card holder's name.
CC_DISPLAY_NUMBER	VARCHAR(20)	Null	The version of the credit card number that is displayed (all Xs except last 4-digits).
CC_COMPANY	VARCHAR(50)	Null	The name of the credit card company.
BILLING_GEOCODE	VARCHAR(2)	Null	The code used by the TAXWARE system to identify taxes for the order based on jurisdiction.
BILLING_STREET1	VARCHAR(30)	Null	The first line in the customer's billing address.
BILLING_STREET2	VARCHAR(30)	Null	The second line in the customer's billing address.
BILLING_CITY	VARCHAR(30)	Null	The city in the customer's billing address.
BILLING_STATE	VARCHAR(40)	Null	The state in the customer's billing address.
BILLING_COUNTRY	VARCHAR(40)	Null	The country in the customer's billing address.
BILLING_POBOX	VARCHAR(30)	Null	The post office box in the customer's billing address.
BILLING_COUNTY	VARCHAR(50)	Null	The county in the customer's billing address.
BILLING_POSTAL_CODE	VARCHAR(10)	Null	The postal (ZIP) code in the customer's billing address.
BILLING_POSTAL_CODE_TYPE	VARCHAR(10)	Null	Format or type of postal code, generally determined by country (such as ZIP code in the United States).

Table 6-16 WLCS_CREDIT_CARD Table Metadata (Continued)

Column Name	Data Type	Null Value	Description and Recommendations
CUSTOMER_ID	VARCHAR(20)	Null	A unique identifier for the customer.
MAP_KEY	VARCHAR(60)	Null	Key that maps multiple credit cards with a single customer.

The WLCS_CUSTOMER Database Table

Information about the customer in the order processing database.

Table 6-17 WLCS_CUSTOMER Table Metadata

Column Name	Data Type	Null Value	Description
CUSTOMER_ID	VARCHAR(20)	Not Null	A unique identifier for the customer. This field is the table's primary key and cannot be NULL. All other fields in the WLCS_CUSTOMER table can be NULL.
CUSTOMER_TYPE	VARCHAR(20)	Null	A label for the customer (such as preferred, standard, or business).
FIRST_NAME	VARCHAR(30)	Null	The customer's first name.
LAST_NAME	VARCHAR(30)	Null	The customer's last name.
MIDDLE_NAME	VARCHAR(30)	Null	The customer's middle name.
TITLE	VARCHAR(10)	Null	The customer's preferred title, such as Mr., Mrs., or Ms.
SUFFIX	VARCHAR(10)	Null	The customer's preferred suffix, such as Jr.or Sr.
EMAIL	VARCHAR(80)	Null	The customer's email address.
HOME_PHONE	VARCHAR(15)	Null	The customer's home phone number.
BUSINESS_PHONE	VARCHAR(20)	Null	The customer's business phone number.
FAX	VARCHAR(15)	Null	The customer's fax number.

Table 6-17 WLCS_CUSTOMER Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MAILING_GEOCODE	VARCHAR(2)	Null	The code used by the TAXWARE system to identify taxes for the order based on jurisdiction.
MAILING_STREET1	VARCHAR(30)	Null	The first line in the customer's street address.
MAILING_STREET2	VARCHAR(30)	Null	The second line in the customer's street address.
MAILING_CITY	VARCHAR(30)	Null	The city in the customer's address.
MAILING_STATE	VARCHAR(40)	Null	The state in the customer's address.
MAILING_COUNTRY	VARCHAR(40)	Null	The country in the customer's address.
MAILING_POBOX	VARCHAR(30)	Null	The post office box in the customer's address.
MAILING_COUNTY	VARCHAR(50)	Null	The county in the customer's address.
MAILING_POSTAL_CODE	VARCHAR(10)	Null	The postal (ZIP) code in the customer's address.
MAILING_POSTAL_CODE_TYPE	VARCHAR(10)	Null	Format or type of postal code, generally determined by country (such as ZIP code in the United States).

The WLCS_ORDER Database Table

Information about a customer's specific order in the order-processing database. The Commerce services product does not populate the SHIPPING_AMOUNT, SHIPPING_CURRENCY, PRICE_AMOUNT, or PRICE_CURRENCY columns.

Table 6-18 WLCS_ORDER Table Metadata

Column Name	Data Type	Null Value	Description
ORDER_ID	VARCHAR(20)	Not Null	PK - A unique identifier for the order. This field is the table's primary key and cannot be NULL. All other fields in the WLCS_ORDER table can be NULL.
CUSTOMER_ID	VARCHAR(20)	Null	A unique identifier for the customer.
TRANSACTION_ID	VARCHAR(25)	Null	A unique identifier for the transaction.
STATUS	VARCHAR(20)	Null	The status of the order.
ORDER_DATE	DATE	Null	The date the order was placed.
SHIPPING_METHOD	VARCHAR(40)	Null	The method by which the order is to be shipped.
SHIPPING_AMOUNT	NUMBER(16,4)	Null	The shipping amount for the order.
SHIPPING_CURRENCY	VARCHAR(10)	Null	The currency associated with the shipping amount.
PRICE_AMOUNT	NUMBER(16,4)	Null	The price of the order.
PRICE_CURRENCY	VARCHAR(10)	Null	The currency associated with the price.
SHIPPING_GEOGODE	VARCHAR(2)	Null	The code used by the TAXWARE system to identify taxes for the order based on jurisdiction.
SHIPPING_STREET1	VARCHAR(30)	Null	The first line in the customer's shipping address.
SHIPPING_STREET2	VARCHAR(30)	Null	The second line in the customer's shipping address.

Table 6-18 WLCS_ORDER Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
SHIPPING_CITY	VARCHAR(30)	Null	The city in the customer's shipping address.
SHIPPING_STATE	VARCHAR(40)	Null	The state in the customer's shipping address.
SHIPPING_COUNTRY	VARCHAR(40)	Null	The country in the customer's shipping address.
SHIPPING_POBOX	VARCHAR(30)	Null	The post office box in the customer's shipping address.
SHIPPING_COUNTY	VARCHAR(50)	Null	The county in the customer's shipping address.
SHIPPING_POSTAL_CODE	VARCHAR(10)	Null	The postal (ZIP) code in the customer's shipping address.
SHIPPING_POSTAL_CODE_TYPE	VARCHAR(10)	Null	Format or type of postal code, generally determined by country, such as ZIP code in the United States.
SPECIAL_INSTRUCTIONS	VARCHAR(254)	Null	Any special shipping instructions associated with the order.
SPLITTING_PREFERENCE	VARCHAR(254)	Null	The splitting preferences for the customer's order.
ORDER_SUBTOTAL	NUMBER(16,4)	Null	The sum of all the TOTAL_LINE_AMOUNT columns in the WLCS_ORDER_LINE table for that specific order.

The WLCS_ORDER_LINE Database Table

Information about each line of a customer's shopping cart in the order processing database.

Table 6-19 WLCS_ORDER_LINE Table Metadata

Column Name	Data Type	Null Value	Description
ORDER_LINE_ID	NUMBER(15)	Not Null	PK - A unique identifier for each line in a customer's shopping cart. This field is the table's primary key and cannot be NULL. All other fields in the WLCS_ORDERLINE table can be NULL.
QUANTITY	NUMBER(16,4)	Null	The quantity of the item in the shopping cart.
PRODUCT_ID	VARCHAR(40)	Null	An identification number for the item in the shopping cart.
TAX_AMOUNT	NUMBER(16,4)	Null	The tax amount for the order.
TAX_CURRENCY	VARCHAR(10)	Null	The currency associated with the tax amount.
SHIPPING_AMOUNT	NUMBER(16,4)	Null	The shipping amount for the order.
SHIPPING_CURRENCY	VARCHAR(10)	Null	The currency associated with the shipping amount.
UNIT_PRICE_AMOUNT	NUMBER(16,4)	Null	The unit price amount for the item.
UNIT_PRICE_CURRENCY	VARCHAR(10)	Null	The currency associated with the unit price.
MSRP_AMOUNT	NUMBER(16,4)	Null	The MSRP amount for the item.
MSRP_CURRENCY	VARCHAR(10)	Null	The currency associated with the MSRP amount.
DESCRIPTION	VARCHAR(254)	Null	The name of the item that is part of the order.
ORDER_ID	VARCHAR(20)	Null	FK - A unique identifier for the order.

Table 6-19 WLCS_ORDER_LINE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
TOTAL_LINE_AMOUNT	NUMBER(16,4)	Null	The total discounted price for the line item. UNIT_PRICE_AMOUNT (less any discount) times the QUANTITY.

The WLCS_SAVED_ITEM_LIST Database Table

Information about the customer's saved shopping cart items in the order processing database.

Table 6-20 WLCS_SAVED_ITEM_LIST Table Metadata

Column Name	Data Type	Null Value	Description
CUSTOMER_ID	VARCHAR(20)	Null	A unique identifier for the customer.
SKU	VARCHAR(40)	Null	A unique identifier (the Stock Keeping Unit or SKU) for a product item.

The WLCS_SECURITY Database Table

Persists public and private keys for encryption and decryption purposes in the order processing database. This table is meant for internal use by the Commerce services product.

Table 6-21 WLCS_SECURITY Table Metadata

Column Name	Data Type	Null Value	Description
ID	NUMBER(5)	Null	A unique identifier for the key pair. This field is the table's primary key and cannot be NULL.
PUBLIC_KEY	VARCHAR(2000)	Null	The public key to be used for encryption/decryption of credit cards.
PRIVATE_KEY	VARCHAR(2000)	Null	The private key to be used for encryption/decryption of credit cards.

The WLCS_SHIPPING_ADDRESS Database Table

Information related to a customer's shipping address(es) in the order processing database.

Table 6-22 WLCS_SHIPPING_ADDRESS Table Metadata

Column Name	Data Type	Null Value	Description
SHIPPING_ADDRESS_ID	NUMBER(15)	Not Null	PK - A unique identifier for the shipping address.
MAP_KEY	VARCHAR(60)	Null	Key that maps multiple shipping addresses with a single customer.
SHIPPING_GEOCODE	VARCHAR(2)	Null	The code used by the TAXWARE system to identify taxes for the order based on jurisdiction.
SHIPPING_STREET1	VARCHAR(30)	Null	The first line in the customer's shipping address.
SHIPPING_STREET2	VARCHAR(30)	Null	The second line in the customer's shipping address.
SHIPPING_CITY	VARCHAR(30)	Null	The city in the customer's shipping address.
SHIPPING_STATE	VARCHAR(40)	Null	The state in the customer's shipping address.
SHIPPING_COUNTRY	VARCHAR(40)	Null	The country in the customer's shipping address.
SHIPPING_POBOX	VARCHAR(30)	Null	The post office box in the customer's shipping address.
SHIPPING_COUNTY	VARCHAR(50)	Null	The county in the customer's shipping address.
SHIPPING_POSTAL_CODE	VARCHAR(10)	Null	The postal (zip) code in the customer's shipping address.
SHIPPING_POSTAL_CODE_TYPE	VARCHAR(10)	Null	Format or type of postal code, generally determined by country, such as ZIP code in the United States.

Table 6-22 WLCS_SHIPPING_ADDRESS Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CUSTOMER_ID	VARCHAR(20)	Null	A unique identifier for the customer.

The WLCS_SHIPPING_METHOD Database Table

Information about the shipping method in the order processing database.

Table 6-23 WLCS_SHIPPING_METHOD Table Metadata

Column Name	Data Type	Null Value	Description
PK_IDENTIFIER	VARCHAR(20)	Not Null	PK - A unique identifier for the shipping method.
CARRIER	VARCHAR(40)	Null	The carrier being used to ship the order, such as UPS or FedEx.
METHOD	VARCHAR(40)	Null	The method by which the order is to be shipped, such as Air, 2nd Day Air, or Parcel Post.
AVERAGE_SHIPPING_TIME	NUMBER	Null	The average number of days it will take the order to arrive.
PRICE_VALUE	NUMBER(16,4)	Null	The amount it will cost to ship the order.
PRICE_CURRENCY	VARCHAR(10)	Null	The currency associated with the PRICE_VALUE column, such as dollars, pounds, or lira.
WEIGHT_LIMIT	NUMBER(16,4)	Null	The weight limit for the shipment.
RESTRICTIONS	VARCHAR(254)	Null	Any restrictions associated with the shipment.
DESCRIPTION	VARCHAR(254)	Null	A description of the shipping method, such as FedEx Overnight or Standard.
PO_BOX_ALLOWED	NUMBER	Null	Specifies whether or not the shipment can be left at a post office box.
SIGNATURE_REQUIRED	NUMBER	Null	Specifies whether or not a signature is required upon receipt of the shipment.

Table 6-23 WLCS_SHIPPING_METHOD Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
SATURDAY_DELIVERY	NUMBER	Null	Specifies whether or not the shipment can be delivered on Saturday.
INTERNATIONAL_DELIVERY	NUMBER	Null	Specifies whether or not international delivery is an option.
SIZE_LIMIT	NUMBER(16,4)	Null	The size limit for the shipment.
PACKAGING_TYPE	VARCHAR(50)	Null	The packaging type for the shipment.

The WLCS_TRANSACTION Database Table

Data for every payment transaction in the order processing database.

Table 6-24 WLCS_TRANSACTION Table Metadata

Column Name	Data Type	Null Value	Description
TRANSACTION_ID	VARCHAR(25)	Not Null	PK - A unique identifier for the transaction.
BATCH_ID	VARCHAR(15)	Null	A unique identifier of a batch submitted for settlement, as returned by the Payment Web service. This field need not be populated for other external payment services.
TRAN_DATE	DATE	Null	The date of the transaction (that is, date on which the transaction was first started).
TRAN_STATUS	VARCHAR(20)	Null	The current status of the transaction (Settled, Authorized, MarkedForSettle, PendingSettle, Retry, or Settled).
TRAN_AMOUNT	NUMBER(16,4)	Null	The most recent amount applied to the transaction. MarkForSettle amounts can be different from the authorization amount.
TRAN_CURRENCY	VARCHAR(30)	Null	The currency of the transaction.

Table 6-24 WLCS_TRANSACTION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CC_NUMBER	VARCHAR(200)	Null	The customer's credit card number. This is encrypted if `is.encryption.enable` is set to `true` in the `weblogiccommerce.properties` file.
CC_TYPE	VARCHAR(20)	Null	The customer's credit card type, such as VISA or MasterCard.
CC_EXP_DATE	DATE	Null	The expiration date on the customer's credit card.
CC_NAME	VARCHAR(50)	Null	The credit card holder's name.
CC_DISPLAY_NUMBER	VARCHAR(20)	Null	The version of the credit card number that is displayed (displays all Xs except last 4-digits).
CC_COMPANY	VARCHAR(50)	Null	The name of the credit card company.
GEOCODE	VARCHAR(2)	Null	The code used by the TAXWARE system to identify taxes for the order based on jurisdiction.
STREET1	VARCHAR(30)	Null	The first line in the customer's street address.
STREET2	VARCHAR(30)	Null	The second line in the customer's street address.
CITY	VARCHAR(30)	Null	The city in the customer's address.
STATE	VARCHAR(40)	Null	The state in the customer's address.
COUNTRY	VARCHAR(40)	Null	The country in the customer's address.
POBOX	VARCHAR(30)	Null	The post office box in the customer's address.
DESCRIPTION	VARCHAR(30)	Null	Any additional data. Can be `NULL`.
COUNTY	VARCHAR(50)	Null	The county in the customer's address.

Table 6-24 WLCS_TRANSACTION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
POSTAL_CODE	VARCHAR(10)	Null	The postal (ZIP) code in the customer's address.
POSTAL_CODE_TYPE	VARCHAR(10)	Null	Format or type of postal code, generally determined by country, such as Zip code in the United States.

The WLCS_TRANSACTION_ENTRY Database Table

Logs the different states a payment transaction has passed through in the order processing database.

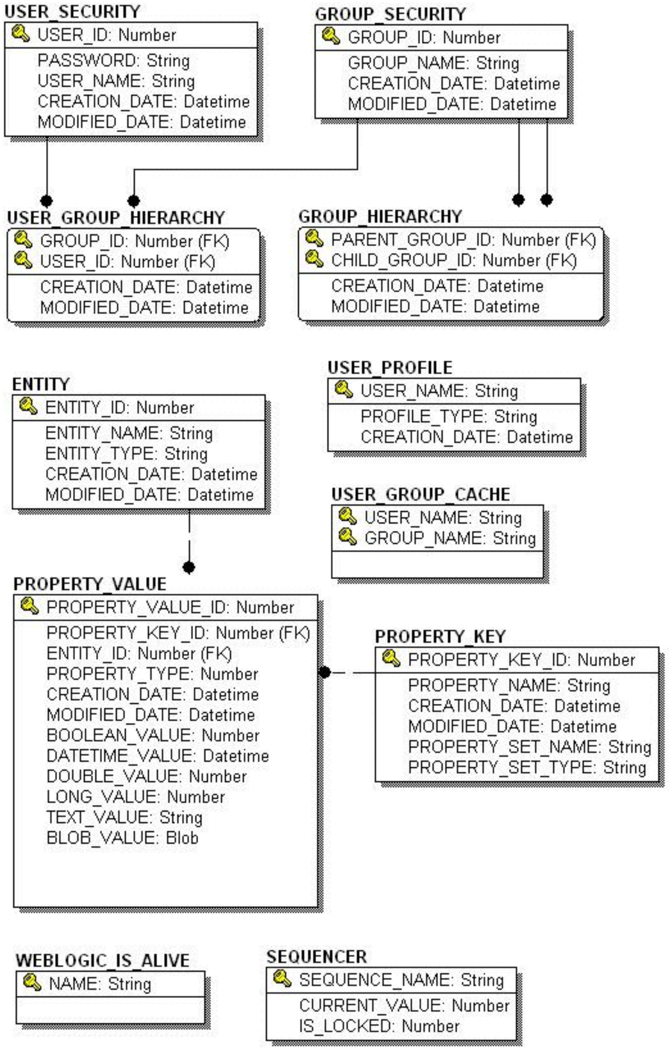
Table 6-25 WLCS_TRANSACTION_ENTRY Table Metadata

Column Name	Data Type	Null Value	Description
TRANSACTION_ENTRY_ID	NUMBER(25)	Not Null	PK - A unique identifier for the transaction entry.
TRAN_ENTRY_SEQUENCE	VARCHAR(30)	Null	Represents the running count per transaction.
TRAN_ENTRY_DATE	DATE	Null	The date of the log entry.
TRAN_ENTRY_STATUS	VARCHAR(20)	Null	The status of the transaction when this entry was made.
TRAN_ENTRY_AMOUNT	NUMBER(16,4)	Null	The amount of the transaction when the log entry was made.
TRAN_ENTRY_CURRENCY	VARCHAR(30)	Null	The currency of the transaction.
TRANSACTION_ID	VARCHAR(25)	Null	A unique identifier for the transaction.

Personalization Database Objects

This section provides information about the database objects for WebLogic Portal personalization features. Figure 6-6 shows an Entity Relation diagram for the WebLogic Portal Personalization database objects.

Figure 6-6 Entity-Relation Diagram for WebLogic Portal Personalization



The Portal Personalization Database Tables

In this section, WebLogic Portal personalization tables are arranged alphabetically as a data dictionary.

The following tables compose the portal personalization database:

- The GROUP_HIERARCHY Database Table
- The GROUP_SECURITY Database Table
- The USER_GROUP_CACHE Database Table
- The USER_GROUP_HIERARCHY Database Table
- The USER_PROFILE Database Table
- The USER_SECURITY Database Table
- The ENTITY Database Table
- The PROPERTY_KEY Database Table
- The PROPERTY_VALUE Database Table
- The SEQUENCER Database Table
- The WEBLOGIC_IS_ALIVE Database Table

The GROUP_HIERARCHY Database Table

This table stores relationship information between groups.

Table 6-26 GROUP_HIERARCHY Table Metadata

Column Name	Data Type	Null Value	Description
PARENT_GROUP_ID	NUMBER(15)	Not Null	PK - The parent group identifier. FK to the ENTITY table.
CHILD_GROUP_ID	NUMBER(15)	Not Null	PK - The child group identifier. FK to the ENTITY table.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The GROUP_SECURITY Database Table

This table holds all groups that a user could be given membership to for security authentication of the rdbms realm.

Table 6-27 GROUP_SECURITY Table Metadata

Column Name	Data Type	Null Value	Description
GROUP_ID	NUMBER(15)	Not Null	PK – a unique, system-generated number used as the record identifier.
GROUP_NAME	VARCHAR(200)	Not Null	The name of the group.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The USER_GROUP_CACHE Database Table

In the event of a deep group hierarchy, this table will flatten the group hierarchy and enables quick group membership searches.

Note: The startup process GroupCache is disabled by default. This table will only be used if enabled.

Table 6-28 USER_GROUP_CACHE Table Metadata

Column Name	Data Type	Null Value	Description
`USER_NAME`	`VARCHAR(200)`	Not Null	PK - A user's name.
`GROUP_NAME`	`VARCHAR(200)`	Not Null	PK - A group name.

The USER_GROUP_HIERARCHY Database Table

This table allows you to store associated users and groups.

Table 6-29 USER_GROUP_HIERARCHY Table Metadata

Column Name	Data Type	Null Value	Description
`GROUP_ID`	`NUMBER(15)`	Not Null	PK - and FK – to `USER_SECURITY.USER_ID`

Table 6-29 USER_GROUP_HIERARCHY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
USER_ID	NUMBER(15)	Not Null	PK - and FK – to GROUP_SECURITY.GROUP_ID
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The USER_PROFILE Database Table

This table associates users with profiles (such as the WLCS_CUSTOMER user profile). User profiles use property sets to organize the properties that they contain.

Table 6-30 USER_PROFILE Table Metadata

Column Name	Data Type	Null Value	Description
USER_NAME	VARCHAR(200)	Not Null	PK - The name of the user.
PROFILE_TYPE	VARCHAR(100)	Not Null	A type of profile associated with the user (such as WLCS_Customer).
CREATION_DATE	DATE	Not Null	The date and time this record was created.

The USER_SECURITY Database Table

This table holds all the user records for security authentication of the rdbms realm.

Table 6-31 USER_SECURITY Table Metadata

Column Name	Data Type	Null Value	Description
USER_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as the record identifier.
USER_NAME	VARCHAR(200)	Not Null	The user’s name.
PASSWORD	VARCHAR(50)	Null	The user’s password.
CREATION_DATE	DATE	Not Null	The date and time this record was created.

Table 6-31 USER_SECURITY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The ENTITY Database Table

Some objects in WebLogic Portal implement a Java interface called ConfigurableEntity. Any ConfigurableEntity within the system will have an entry in this table.

Table 6-32 ENTITY Table Metadata

Column Name	Data Type	Null Value	Description
ENTITY_ID	NUMBER(15)	Not Null	PK - A unique, sequence-generated number used as the record identifier.
ENTITY_NAME	VARCHAR(200)	Not Null	The name of the ConfigurableEntity.
ENTITY_TYPE	VARCHAR(100)	Not Null	Defines what type of ConfigurableEntity this is.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The PROPERTY_KEY Database Table

Contents: Any property assigned to a ConfigurableEntity has a unique PROPERTY_ID. This identifier and associated information is stored here.

Primary Key: PROPERTY_KEY_ID.

Table 6-33 PROPERTY_KEY Table Metadata

Column Name	Data Type	Null Value	Description and Recommendations
PROPERTY_KEY_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as the record identifier.
PROPERTY_NAME	VARCHAR(100)	Not Null	The name of the property.

Table 6-33 PROPERTY_KEY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description and Recommendations
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.
PROPERTY_SET_NAME	VARCHAR(100)	Null	The name of the property set.
PROPERTY_SET_TYPE	VARCHAR(100)	Null	The type the property set.

The PROPERTY_VALUE Database Table

This table stores property values for boolean, datetime, float, integer, text, and user-defined properties.

Table 6-34 PROPERTY_VALUE Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_VALUE_ID	NUMBER(15)	Not Null	PK – a unique, system-generated number used as the record identifier.
PROPERTY_KEY_ID	NUMBER(15)	Not Null	FK - to PROPERTY_KEY.PROPERTY_KEY_ID
ENTITY_ID	NUMBER(15)	Not Null	FK – to ENTITY.ENTITY_ID
PROPERTY_TYPE	NUMBER(1)	Not Null	Valid entries are: 0=Boolean, 1=Integer, 2=Float, 3=Text, 4=Date and Time, 5=User-Defined (BLOB)
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.
BOOLEAN_VALUE	NUMBER(1)	Null	The value for each boolean property identifier.
DATETIME_VALUE	DATE	Null	The value for each date and time property identifier.

Table 6-34 PROPERTY_VALUE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
DOUBLE_VALUE	NUMBER	Null	The value associated with each float property identifier.
LONG_VALUE	NUMBER(20)	Null	The value associated with the integer property.
TEXT_VALUE	VARCHAR(254)	Null	The value associated with the text property.
BLOB_VALUE	BLOB	Null	The value associated with the user-defined property.

The SEQUENCER Database Table

The SEQUENCER table is used to maintain all of the sequence identifiers (for example, property_meta_data_id_sequence, and so on) used in the application.

Table 6-35 SEQUENCER Table Metadata

Column Name	Data Type	Null Value	Description
SEQUENCE_NAME	VARCHAR(50)	Not Null	PK – A unique name used to identify the sequence.
CURRENT_VALUE	NUMBER(15)	Not Null	The current value of the sequence.
IS_LOCKED	NUMBER(1)	Not Null	This flag identifies whether or not the particular SEQUENCE_ID has been locked for update. This column is being used as a generic locking mechanism that can be used for multiple database environments.

The WEBLOGIC_IS_ALIVE Database Table

This table is used by the JDBC connection pools to insure the connection to the database is still alive.

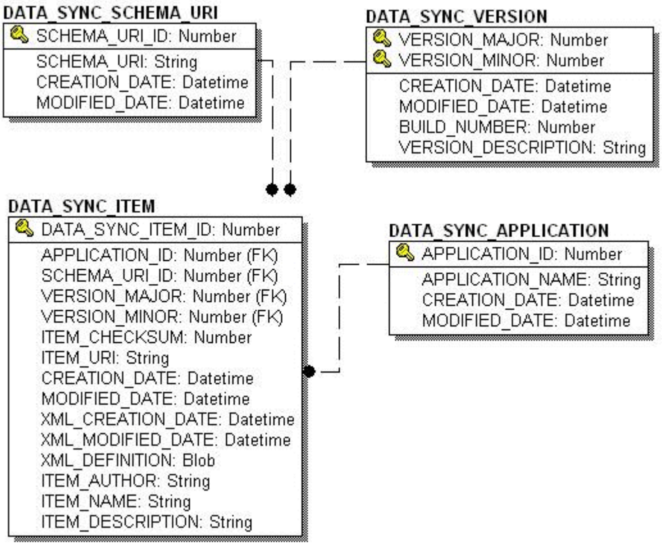
Table 6-36 WEBLOGIC_IS_ALIVE Table Metadata

Column Name	Data Type	Null Value	Description
NAME	VARCHAR(100)	Not Null	Used by the JDBC connection pools to insure the connection to the database is still alive.

Data Synchronization Database Objects

This section provides information about the database objects for WebLogic Portal data synchronization features. Figure 6-7 shows an Entity Relation diagram for WebLogic Portal data synchronization database objects.

Figure 6-7 Entity-Relation Diagram for WebLogic Portal Data Synchronization



The Data Synchronization Database Tables

In this section, WebLogic Portal data synchronization objects tables are arranged alphabetically as a data dictionary.

The following tables compose the data synchronization database:

- The DATA_SYNC_APPLICATION Database Table
- The DATA_SYNC_ITEM Database Table
- The DATA_SYNC_SCHEMA_URI Database Table
- The DATA_SYNC_VERSION Database Table

The DATA_SYNC_APPLICATION Database Table

This table holds the various applications available for the data synchronization process..

Table 6-37 DATA_SYNC_APPLICATION Table Metadata

Column Name	Data Type	Null Value	Description
APPLICATION_ID	NUMBER(15)	Not Null	PK - A unique, system-generated number used as the record identifier.
APPLICATION_NAME	VARCHAR(100)	Not Null	The deployed J2EE application name. (This should match the name in the WebLogic Server console.)
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The DATA_SYNC_ITEM Database Table

This table stores all the data items to be synchronized.

Table 6-38 DATA_SYNC_ITEM Table Metadata

Column Name	Data Type	Null Value	Description
DATA_SYNC_ITEM_ID	NUMBER(15)	Not Null	PK - A unique, system-generated number used as the record identifier.
APPLICATION_ID	NUMBER(15)	Not Null	FK – to DATA_SYNC_APPLICATON.APPLICATION_ID
SCHEMA_URI_ID	NUMBER(15)	Not Null	FK – to DATA_SYNC_SCHEMA_URI.SCHEMA_URI_ID
VERSION_MAJOR	NUMBER(15)	Not Null	FK – to DATA_SYNC_VERSION.VERSION_MAJOR
VERSION_MINOR	NUMBER(15)	Not Null	FK – to DATA_SYNC_VERSION.VERSION_MINOR
ITEM_CHECKSUM	NUMBER(15)	Not Null	A generated number representing the contents of the XML_DEFINITION column.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.
XML_MODIFIED_DATE	DATE	Not Null	The date and time the XML file was last modified.
XML_CREATION_DATE	DATE	Not Null	The date and time the XML file was created.
XML_DEFINITION	CLOB	Not Null	The XML representation of the data item to be synchronized.
ITEM_URI	VARCHAR(254)	Not Null	The path on the file system of the data item to be synchronized.
ITEM_AUTHOR	VARCHAR(200)	Null	Metadata info—the o/s login.
ITEM_NAME	VARCHAR(100)	Null	Metadata info—the full path to the item.
ITEM_DESCRIPTION	VARCHAR(254)	Null	Metadata info—a general description of the item to be synchronized.

The DATA_SYNC_SCHEMA_URI Database Table

This table holds information pertaining to each of the governing schemas used by various documents.

Table 6-39 DATA_SYNC_SCHEMA_URI Table Metadata

Column Name	Data Type	Null Value	Description
SCHEMA_URI_ID	NUMBER(15)	Not Null	PK - A unique, system-generated number used as the record identifier.
SCHEMA_URI	VARCHAR(254)	Not Null	The governing schema of the document.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.

The DATA_SYNC_VERSION Database Table

This table is not being used currently. It is reserved for future use and is expected to accommodate data synchronization versioning. As a result, this table only holds one record.

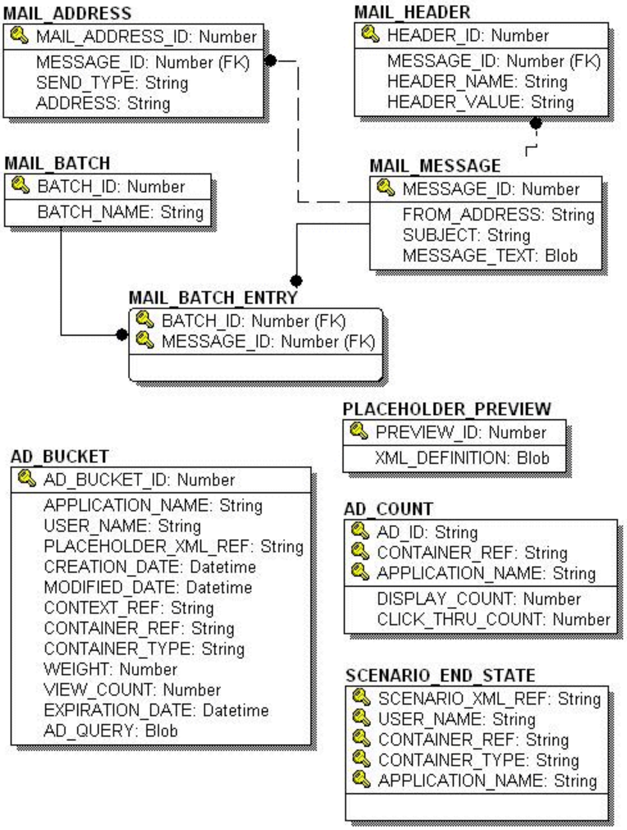
Table 6-40 DATA_SYNC_VERSION Table Metadata

Column Name	Data Type	Null Value	Description
VERSION_MAJOR	NUMBER(15)	Not Null	PK - The current record has a value of zero.
VERSION_MINOR	NUMBER(15)	Not Null	PK - The current record has a value of zero.
CREATION_DATE	DATE	Not Null	The date and time the record was created.
MODIFIED_DATE	DATE	Not Null	The date and time the record was last modified.
BUILD_NUMBER	NUMBER(15)	Null	The build number associated with the version.
VERSION_DESCRIPTION	VARCHAR(30)	Null	A description of the particular sync version.

WebLogic Portal Services Database Objects

This section provides information about the database objects for WebLogic Portal Services features. Figure 6-8 shows an Entity Relation diagram for WebLogic Portal services database objects.

Figure 6-8 Entity-Relation Diagram for WebLogic Portal Services



The Portal Services Database Tables

In this section, WebLogic Portal Services objects tables are arranged alphabetically as a data dictionary.

The following tables compose the Portal services database:

- The AD_BUCKET Database Table
- The AD_COUNT Database Table
- The PLACEHOLDER_PREVIEW Database Table
- The MAIL_ADDRESS Database Table
- The MAIL_BATCH Database Table
- The MAIL_BATCH_ENTRY Database Table
- The MAIL_HEADER Database Table
- The MAIL_MESSAGE Database Table
- The SCENARIO_END_STATE Database Table

The AD_BUCKET Database Table

This table maintains content queries for ads.

Table 6-41 AD_BUCKET Table Metadata

Column Name	Data Type	Null Value	Description
AD_BUCKET_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number used as the record identifier.
USER_NAME	VARCHAR (200)	Not Null	The user's name associated with the ad.
PLACEHOLDER_XML_REF	VARCHAR(254)	Not Null	The location identifier of the XML-based placeholder definition file.
APPLICATION_NAME	VARCHAR(100)	Not Null	The name of the application for which the ad has been scoped.
CONTEXT_REF	VARCHAR(254)	Null	The scenario unique identifier.
CONTAINER_REF	VARCHAR(254)	Null	The campaign unique identifier.

Table 6-41 AD_BUCKET Table Metadata

Column Name	Data Type	Null Value	Description
CONTAINER_TYPE	VARCHAR(50)	Null	Identifies the service associated with the CONTAINER_REF.
WEIGHT	NUMBER(15)	Null	A weighted scheme used in prioritizing one placeholder over another.
VIEW_COUNT	NUMBER(15)	Null	*Disabled. Reserved for future use.*
EXPIRATION_DATE	DATE	Null	The date and time the ad expires or becomes invalid.
CREATION_DATE	DATE	Not Null	The date and time this record was created.
MODIFIED_DATE	DATE	Not Null	The date and time this record was last modified.
AD_QUERY	CLOB	Null	The actual content query.

The AD_COUNT Database Table

This table tracks the number of times the ads are displayed and clicked though.

Table 6-42 AD_COUNT Table Metadata

Column Name	Data Type	Null Value	Description
AD_ID	VARCHAR(254)	Not Null	PK - A unique, system-generated number used as the record identifier.
CONTAINER_REF	VARCHAR(254)	Not Null	PK - The campaign unique identifier.
APPLICATION_NAME	VARCHAR(100)	Not Null	PK - The name of the application for which the ad clicks or views were scoped
DISPLAY_COUNT	NUMBER(15)	Not Null	The number of times the ad has been displayed.
CLICK_THROUGH_COUNT	NUMBER(15)	Not Null	The number of times the ad has been clicked on.

The PLACEHOLDER_PREVIEW Database Table

This table is used as a mechanism to hold the placeholder for previewing purposes only.

Table 6-43 PLACEHOLDER_PREVIEW Table Metadata

Column Name	Data Type	Null Value	Description
PREVIEW_ID	NUMBER	Not Null	PK—a unique, system generated number used as the record identifier.
XML_DEFINITION	CLOB	Null	The representation of the expression to be previewed.

The MAIL_ADDRESS Database Table

This table stores all of the address info for e-mail purposes.

Table 6-44 MAIL_ADDRESS Table Metadata

Column Name	Data Type	Null Value	Description
MAIL_ADDRESS_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
MESSAGE_ID	NUMBER(15)	Not Null	FK—foreign key to the MAIL_MESSAGE table.
ADDRESS	VARCHAR(254)	Not Null	Stores the various e-mail addresses on the distribution list.
SEND_TYPE	VARCHAR(4)	Not Null	Determines how the ADDRESS should be included on the distribution. Possible values are TO, CC, or BCC.

The MAIL_BATCH Database Table

This table establishes a batch for each mailing.

Table 6-45 MAIL_BATCH Table Metadata

Column Name	Data Type	Null Value	Description
BATCH_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
BATCH_NAME	VARCHAR(254)	Not Null	The name of the mail message batch.

The MAIL_BATCH_ENTRY Database Table

This table is used to correlate the mail batch with the specific mail message.

Table 6-46 MAIL_BATCH_ENTRY Table Metadata

Column Name	Data Type	Null Value	Description
BATCH_ID	NUMBER(15)	Not Null	PK and FK—a unique, system-generated number to be used as the record ID.
MESSAGE_ID	NUMBER(15)	Not Null	PK and FK—foreign key to the MAIL_MESSAGE table.

The MAIL_HEADER Database Table

This table contains all of the header information specific to the e-mail message.

Table 6-47 MAIL_HEADER Table Metadata

Column Name	Data Type	Null Value	Description
HEADER_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
MESSAGE_ID	NUMBER(15)	Not Null	FK—foreign key to the MAIL_MESSAGE table.
HEADER_NAME	VARCHAR(50)	Null	The name of the mail message header.
HEADER_VALUE	VARCHAR(254)	Null	The value of the mail message header.

The MAIL_MESSAGE Database Table

This table contains the specifics of the mail message (e.g., the subject line, text, etc.).

Table 6-48 MAIL_MESSAGE Table Metadata

Column Name	Data Type	Null Value	Description
MESSAGE_ID	NUMBER(15)	Not Null	PK—a unique, system-generated number to be used as the record ID.
FROM_ADDRESS	VARCHAR(254)	Null	Identifies who is sending the message.
SUBJECT	VARCHAR(128)	Null	Stores the mail message subject.
MESSAGE_TEXT	CLOB	Null	Holds the content of the mail message.

The SCENARIO_END_STATE Database Table

This table identifies when a user is no longer eligible to participate in a particular scenario.

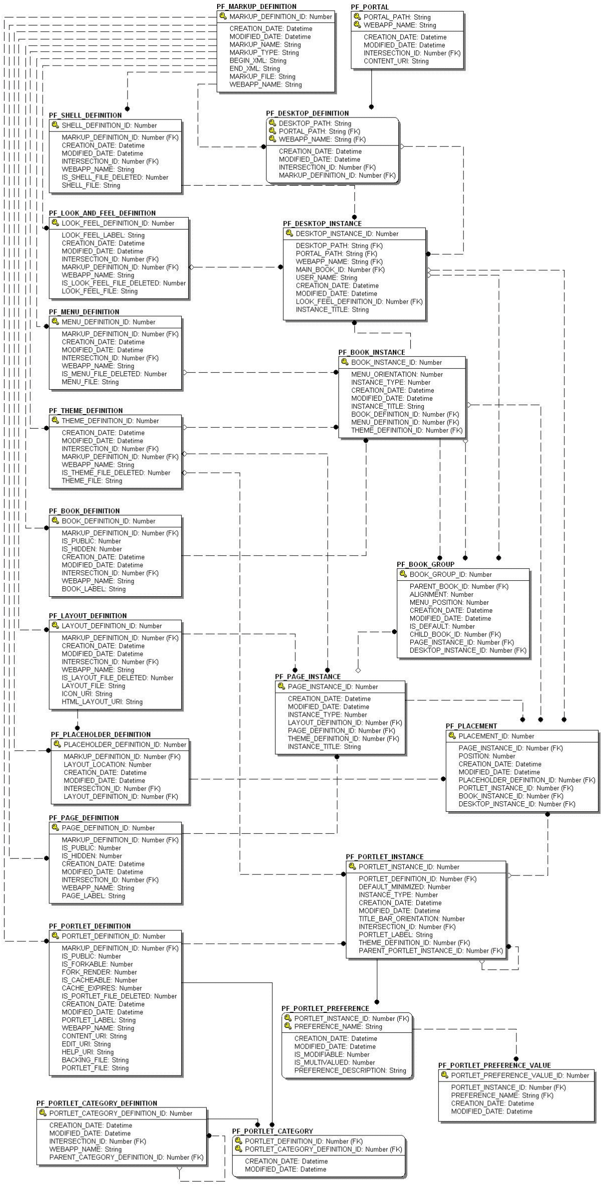
Table 6-49 SCENARIO_END_STATE Table Metadata

Column Name	Data Type	Null Value	Description
SCENARIO_XML_REF	VARCHAR(20)	Not Null	PK—The identifier for the XML-based scenario definition file.
USER_NAME	VARCHAR(200)	Not Null	PK—the user ID. (FK to WLCS_USER.IDENTIFIER)
CONTAINER_REF	VARCHAR(254)	Not Null	PK—the campaign unique identifier. (FK to CAMPAIGN.CAMPAIGN_UID)
CONTAINER_TYPE	VARCHAR(50)	Not Null	PK—At this time this column will always hold the string 'Campaign'.
APPLICATION_NAME	VARCHAR(100)	Not Null	PK—The deployed J2EE application name. This should match the name in the WebLogic Server console.)

Portal Framework Database Objects

This section documents the database objects for the WebLogic Portal package. Figure 6-9 shows the Entity Relation diagram for the WebLogic Portal Framework database objects.

Figure 6-9 Entity-Relation Diagram for the Portal Framework Tables



The Portal Framework Database Tables

In this section, WebLogic Portal Services tables are arranged alphabetically as a data dictionary. The following tables compose the Portal Framework database:

- The PF_BOOK_DEFINITION Database Table
- The PF_BOOK_GROUP Database Table
- The PF_BOOK_INSTANCE Database Table
- The PF_DESKTOP_DEFINITION Database Table
- The PF_DESKTOP_INSTANCE Database Table
- The PF_LAYOUT_DEFINITION Database Table
- The PF_LOOK_AND_FEEL_DEFINITION Database Table
- The PF_MARKUP_DEFINITION Database Table
- The PF_MENU_DEFINITION Database Table
- The PF_PAGE_DEFINITION Database Table
- The PF_PAGE_INSTANCE Database Table
- The PF_PLACEHOLDER_DEFINITION Database Table
- The PF_PLACEMENT Database Table
- The PF_PORTAL Database Table
- The PF_PORTLET_CATEGORY Database Table
- The PF_PORTLET_CATEGORY_DEFINITION Database Table
- The PF_PORTLET_DEFINITION Database Table
- The PF_PORTLET_INSTANCE Database Table
- The PF_PORTLET_PREFERENCE Database Table
- The PF_PORTLET_PREFERENCE_VALUE Database Table
- The PF_SHELL_DEFINITION Database Table
- The PF_THEME_DEFINITION Database Table

The PF_BOOK_DEFINITION Database Table

This table defines a BOOK portal library resource, Books are used to aggregate PAGES and other BOOKS.

Table 6-50 PF_BOOK_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
BOOK_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
IS_PUBLIC	NUMBER	Not Null	A boolean flag indicating if this book definition will be displayed to the public. When end users create books they are not marked as public.
IS_HIDDEN	NUMBER	Not Null	A boolean flag indicating if this book definition will be hidden from the menu. If a page or book is hidden it does not prevent it from being displayed it is only a hint to the menu control to not display a tab for the given book or page. The page or book may be activated via a link or a backing file.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application (as defined in the config.xml) to which the portal resource is scoped.

Table 6-50 PF_BOOK_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
BOOK_LABEL	VARCHAR(40)	Null	A moniker used to reference this portal resource for development purposes. This is the same as the bookDefinitionLabel in WebLogic Worshop. If a label is not supplied at creation time the BOOK_DEFINITION_ID prefixed with a 'B' is used. This label can be supplied to APIs to activate books or pages.

The PF_BOOK_GROUP Database Table

This table represent a child page or book placement on the parent book. A single record in the table represents one placement on a book. This table also identifies a customized grouping of Books and Pages. Customized groupings are represented and aggregated around the DESKTOP_INSTANCE_ID.

Table 6-51 PF_BOOK_GROUP Table Metadata

Column Name	Data Type	Null Value	Description
BOOK_GROUP_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PARENT_BOOK_ID	NUMBER	Not Null	FK to PF_BOOK_INSTANCE that identifies the parent BOOK_INSTANCE_ID.
ALIGNMENT	NUMBER	Not Null	The alignment is a 'hint' to the menu skeleton JSP to indicate if the tab should be aligned on the left or right of the tab bar. A skeleton may choose to implement this feature or ignore it.
MENU_POSITION	NUMBER	Not Null	The order in the tab menu this page or book will appear on the parent book. Order does not need to be contiguous.
CREATION_DATE	DATE	Not Null	The date and time the row was created.

Table 6-51 PF_BOOK_GROUP Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
IS_DEFAULT	NUMBER	Not Null	A boolean flag indicating that this is the default page or book on the parent book.
CHILD_BOOK_ID	NUMBER	Null	FK to PF_BOOK_INSTANCE that identifies the child BOOK_INSTANCE_ID.
PAGE_INSTANCE_ID	NUMBER	Null	FK to PF_BOOK_INSTANCE.
DESKTOP_INSTANCE_ID	NUMBER	Null	FK to PF_DESKTOP_INSTANCE. If this book grouping is an administrators' or end user's customization. This will be non null and point to the admins or users desktop. If this field is null it represents the library's view.

The PF_BOOK_INSTANCE Database Table

This table identifies an instance of the BOOK_DEFINITION. There is always at least one book instance, namely the primary instance. All other instances represent customization by administrators or end users.

Table 6-52 PF_BOOK_INSTANCE Table Metadata

Column Name	Data Type	Null Value	Description
BOOK_INSTANCE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MENU_ORIENTATION	NUMBER	Not Null	The orientation is a hint to the book skeleton JSP and the menu skeleton JSP to display the tabs on the top, left, right, or bottom of the main book. The skeletons may choose to ignore this field.
INSTANCE_TYPE	NUMBER	Not Null	The type of book instance: 1=Primary, 3=Admin, 4=User

Table 6-52 PF_BOOK_INSTANCE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATION_DATE	DATE	Not Null	The date and time the row was created
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INSTANCE_TITLE	VARCHAR(255)	Null	An end user customized title for this BOOK. This title is not internationalized as it is only used by the end user and there should be no need to do so. If the end user does not customize his books title then this will be null and the L10N_RESOURCE title will be used.
BOOK_DEFINITION_ID	NUMBER	Not Null	FK to PF_BOOK_DEFINITION
MENU_DEFINITION_ID	NUMBER	Null	FK to PF_MENU_DEFINITION. Maybe null as not every book is required to have a menu.
THEME_DEFINITION_ID	NUMBER	Null	FK to PF_THEME_DEFINITION

The PF_DESKTOP_DEFINITION Database Table

This table defines a desktop definition. Desktops may be created from template (.portal files) or from existing resources.

Table 6-53 PF_DESKTOP_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
DESKTOP_PATH	VARCHAR(40)	Not Null	Part of the PK- identifies the partial url path to the desktop.
PORTAL_PATH	VARCHAR(40)	Not Null	Part of the PK and FK to PF_PORTAL- identifies the partial url path to this desktop and parent portal.

Table 6-53 PF_DESKTOP_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
WEBAPP_NAME	VARCHAR(80)	Not Null	Part of the PK and FK to PF_PORTAL. This is the name of the webapp (as defined in the config.xml file) this desktop is scoped to.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	**NUMBER**	**Not Null**	**FK to L10N_INTERSECTION. The BOOK_INSTANCE_ID of the main or default PF_BOOK_INSTANCE for the desktop.**
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION

The PF_DESKTOP_INSTANCE Database Table

This table identifies a customized or localized instance of a desktop

Table 6-54 PF_DESKTOP_INSTANCE Table Metadata

Column Name	Data Type	Null Value	Description
DESKTOP_INSTANCE_ID	NUMBER	Not Null	PK- identifies the partial url path to the desktop.
DESKTOP_PATH	VARCHAR(40	Not Null	FK to PF_DESKTOP_DEFINITION
PORTAL_PATH	VARCHAR(40)	Not Null	FK to PF_DESKTOP_DEFINITION
WEBAPP_NAME	VARCHAR(80)	Not Null	FK-to PF_DESKTOP_DEFINITION
MAIN_BOOK_ID	NUMBER	Not Null	FK to BOOK_INSTANCE_ID of the main or default PF_BOOK_INSTANCE for the desktop

Table 6-54 PF_DESKTOP_INSTANCE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
USER_NAME	VARCHAR(200)	Null	NULL if the desktop instance is not for a particular user or admin. The name of the user if the user has customized his/her desktop.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
LOOK_FEEL_DEFINITION_ID	NUMBER	Null	FK to PF_LOOK_AND_FEEL_DEFINITION
INSTANCE_TITLE	VARCHAR(20)	Null	An end user customized title for this DESKTOP. Note this title is not internationalized as it is only used by the end user and there should be no need to do so. If the end user does not customize his desktops title then this will be null and the L10N_RESOURCE title will be used.
SHELL_DEFINITION_ID	NUMBER	Not Null	FK to PF_SHELL_DEFINITION.

The PF_LAYOUT_DEFINITION Database Table

This table defines a LAYOUT portal library resource which is used as a specification for determining the location of items on a page. For every layout definition there is a corresponding.layout file. By updating the.layout file you are updating this record.

Table 6-55 PF_LAYOUT_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
LAYOUT_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION.

Table 6-55 PF_LAYOUT_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application that portal resource is scoped to.
IS_LAYOUT_FILE_DELETED	NUMBER	Not Null	A boolean indicating that the file associated with this layout was removed from the file system. If the layout is not being used then the record will get deleted outright. This flag is only set to true when the .layout file is deleted and the layout is still in use. You may either return the .layout file and this flag will automatically get reset, or remove the layout with a replacement layout in the admin tools.
LAYOUT_FILE	VARCHAR(255)	Null	The name and location of the file associated with this layout definition.
ICON_URI	VARCHAR(255)	Null	The URI that identifies the ICON for this layout definition.
HTML_LAYOUT_URI	VARCHAR(255)	Null	The URI for the HTML for this layout definition, the htlp file is used by the admin and visitor tools to provide a visual display that emulates the real layout.

The PF_LOOK_AND_FEEL_DEFINITION Database Table

This table defines a LOOK and FEEL portal library resource or template for assignment to DESKTOPs that control how a portal renders.

Table 6-56 PF_LOOK_AND_FEEL_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
LOOK_FEEL_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
LOOK_FEEL_LABEL	VARCHAR(40)	Not Null	A moniker used to reference this portal resource for development purposes.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.
IS_LOOK_FEEL_FILE_DELETED	NUMBER	Not Null	A boolean indicating that the file associated with this look and feel was removed from the file system. If the look and feel is not being used then the record will get deleted outright. This flag is only set to true when the .laf file is deleted and the look and feel is still in use. You may either return the .laf file and this flag will automatically get reset, or remove the look and feel with a replacement look and feel in the WebLogic Administration Portal.
LOOK_FEEL_FILE	VARCHAR(255)	Not Null	The fully qualified file path (from the web app) to the location of the .laf file associated with this look and feel definition.

The PF_MARKUP_DEFINITION Database Table

This table defines the MARKUP (blueprint, design, model) for a portal library resource.

Table 6-57 PF_MARKUP_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
MARKUP_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
MARKUP_NAME	VARCHAR(255)	Not Null	The file name and location which contains the definition of this portal object.
MARKUP_TYPE	VARCHAR(20)	Not Null	The type of portal resource that this markup defines.
BEGIN_XML	VARCHAR(2000)	Not Null	The first 2000 characters of XML definition of this portal object.
END_XML	VARCHAR(2000)	Null	The last 2000 characters of the XML definition of this portal object.
MARKUP_FILE	VARCHAR(255)	Null	Location of the file containing the markup definition.
WEBAPP_NAME	VARCHAR(80)	Null	Name of the J2EE Web Application to which the portal resource is scoped.

The PF_MENU_DEFINITION Database Table

This table defines a MENU portal library resource or template which can be assigned to a BOOK INSTANCE.

Table 6-58 PF_MENU_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
MENU_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.

Table 6-58 PF_MENU_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.
IS_MENU_FILE_DELETED	NUMBER	Not Null	A boolean indicating that the file associated with this menu was removed from the file system. If the menu is not being used then the record will get deleted outright. This flag is only set to true when the.menu file is deleted and the menu is still in use. You may either return the .menu file and this flag will automatically get reset, or remove the menu with a replacement menu in the WebLogic Administration Portal.
MENU_FILE	VARCHAR(255)	Not Null	The fully qualified path (from the Web application) to the location of the .menu file associated with this menu definition.

The PF_PAGE_DEFINITION Database Table

This table defines a PAGE portal library resource or template which can be assigned to a PAGE INSTANCE.

Table 6-59 PF_PAGE_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
PAGE_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.

Table 6-59 PF_PAGE_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
IS_PUBLIC	NUMBER	Not Null	A boolean indicating this page definition is public. Only public page definitions are ever exposed to 'visitors'
IS_HIDDEN	NUMBER	Not Null	A boolean indicating this page is hidden. The hidden flag is a hint to the menu not render a tab for this page. The page can still be displayed by other methods (links, events).
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.
PAGE_LABEL	VARCHAR(40)	Null	A moniker used to reference this portal resource for development purposes.

The PF_PAGE_INSTANCE Database Table

This table identifies an instance of the page definition - there is always at least one instance per definition.

Table 6-60 PF_PAGE_INSTANCE Table Metadata

Column Name	Data Type	Null Value	Description
PAGE_INSTANCE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.

Table 6-60 PF_PAGE_INSTANCE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INSTANCE_TYPE	NUMBER	Not Null	1=Primary, 3=Admin, 4=User
LAYOUT_DEFINITION_ID	NUMBER	Not Null	FK to PF_LAYOUT_DEFINITION
PAGE_DEFINITION_ID	NUMBER	Not Null	FK to PF_PAGE_DEFINITION
THEME_DEFINITION_ID	NUMBER	Null	FK to PF_THEME_DEFINITION
INSTANCE_TITLE	VARCHAR(255)	Null	A DESKTOP or USER customized title for this PAGE. This instance title is only valid to end users as it cannot and need not be localized.

The PF_PLACEHOLDER_DEFINITION Database Table

This table defines a PLACEHOLDER portal library resource or template which has a LAYOUT definition and can be assigned to a PLACEMENT.

Table 6-61 PF_PLACEHOLDER_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
PLACEHOLDER_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
LAYOUT_LOCATION	NUMBER	Not Null	The location of this placeholder in the layout. This is used when swapping layouts as portlets in one layout's location will be moved to the other layouts location with the same id. If the other layout does not have the same number of placeholders the modulus of the location by number of locations will be used.
CREATION_DATE	DATE	Not Null	The date and time the row was created

Table 6-61 PF_PLACEHOLDER_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
LAYOUT_DEFINITION_ID	NUMBER	Not Null	FK to PF_LAYOUT_DEFINITION

The PF_PLACEMENT Database Table

Each record in this table represents a single placement of a book or portlet on a page.

Table 6-62 PF_PLACEMENT Table Metadata

Column Name	Data Type	Null Value	Description
PLACEMENT_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PAGE_INSTANCE_ID	NUMBER	Not Null	FK to PF_PAGE_INSTANCE
POSITION	NUMBER	Not Null	The position within the placeholder this placement lies (placeholders can contain more then one placement).
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
PLACEHOLDER_DEFINITION_ID	NUMBER	Not Null	FK to PF_PLACEHOLDER_DEFINITION
PORTLET_INSTANCE_ID	NUMBER	Null	FK to PF_PORTLET_INSTANCE
BOOK_INSTANCE_ID	NUMBER	Null	FK to PF_BOOK_INSTANCE

Table 6-62 PF_PLACEMENT Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
DESKTOP_INSTANCE_ID	NUMBER	Null	FK to PF_DESKTOP_INSTANCE. If this placement grouping is an admins or end user's customization. This will be non null and point to the admins or users desktop. If this field is null it represents the library's view.

The PF_PORTAL Database Table

This table identifies a PORTAL application library resource or template which can be associated with a DESKTOP definition.

Table 6-63 PF_PORTAL Table Metadata

Column Name	Data Type	Null Value	Description
PORTAL_PATH	VARCHAR(40)	Not Null	PK - partial primary key and partial URL to this portal
WEBAPP_NAME	VARCHAR(80)	Not Null	PK-Name of the J2EE Web Application to which the portal resource is scoped.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
CONTENT_URI	VARCHAR(255)	Null	Defines an optional URI to be forwarded to when only the portal portion of the URL is supplied. This URL (JSP or .portal) can be used to forward to a default desktop or display a list of desktops available under this portal.

The PF_PORTLET_CATEGORY Database Table

This table associates a PORTLET CATEGORY resource with a PORTLET DEFINITION.

Table 6-64 PF_PORTLET_CATEGORY Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PORTLET_CATEGORY_ DEFINITION_ID	NUMBER	Not Null	FK to PF_PORTLET_CATEGORY_ DEFINITION
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.

The PF_PORTLET_CATEGORY_DEFINITION Database Table

This table identifies a PORTLET CATEGORY and PORTLET CATEGORY hierarchy resource or template for association with a PORTLET resource.

Table 6-65 PF_PORTLET_CATEGORY_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_CATEGORY_ DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.
PARENT_CATEGORY_ DEFINITION_ID	NUMBER	Null	FK to PF_PORTLET_CATEGORY_ DEFINITION that identifies the parent portlet category. NULL if this is a top level category.

The PF_PORTLET_DEFINITION Database Table

This table identifies the characteristics of a PORTLET library resource or template which can used as the user interfaces for a web application.

Table 6-66 PF_PORTLET_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
IS_PUBLIC	NUMBER	Not Null	A boolean indicating this portlet definition is public. Only public portlet definitions are ever exposed to 'visitors'.
IS_FORKABLE	NUMBER	Not Null	A boolean indicating this portlet supports multi threading.
FORK_RENDER	NUMBER	Not Null	A boolean - _is_ multi-threading being used for this portlet, can only be true if IS_FORKABLE is true
IS_CACHEABLE	NUMBER	Not Null	A boolean - _can_ this portlet use render caching
CACHE_EXPIRES	NUMBER	Not Null	Is this portlet using caching and if so what is the ttl, -1 indicates off, 0..n indicates a ttl for the cache, can only have a value other then -1 if IS_CACHEABLE is true.
IS_PORTLET_FILE_DELETED	NUMBER	Not Null	A boolean that indicates that the PORTLET_FILE associated with this object has been removed from the file system. This flag is only set to true when the .portlet file is deleted and the portlet is still in use. You may either return the .portlet file and this flag will automatically get reset, or remove the portlet in the WebLogic Administration Portal.

Table 6-66 PF_PORTLET_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATION_DATE	DATE	Not Null	The date and time the row was created
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
PORTLET_LABEL	VARCHAR(80)	Not Null	A moniker used to reference this portal resource for development purposes.
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.
CONTENT_URI	VARCHAR(255)	Not Null	The content URI for this portlet (JSP, HTML) may be null for JAVA (JSR168) portlets.
EDIT_URI	VARCHAR(255)	Null	The Edit mode URI (JSP) for this portlet (if the portlet supports edit mode).
HELP_URI	VARCHAR(255)	Null	The Help mode URI (JSP) for this portlet (if the portlet supports help mode).
BACKING_FILE	VARCHAR(255)	Null	The optional backing file (java class name) for this portlet. Backing classes must implement JspBacking or extend AbstractJspBacking.
PORTLET_FILE	VARCHAR(255)	Null	The (*.portlet) file describing the controls that make up the portlet

The PF_PORTLET_INSTANCE Database Table

This table identifies a customized or localized instance of a Portlet.

Table 6-67 PF_PORTLET_INSTANCE Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_INSTANCE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PORTLET_DEFINITION_ID	NUMBER	Not Null	FK to PF_PORTLET_DEFINITION
DEFAULT_MINIMIZED	NUMBER	Not Null	A boolean that indicates this portlet is to be displayed in the minimized state by default.
INSTANCE_TYPE	NUMBER	Not Null	Type codes for the portlet instance. Valid values: 1=Primary, 3=Admin, 4=User.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
TITLE_BAR_ORIENTATION	NUMBER	Null	A hint to the skeleton file to display this portlets titlebar in the (top, left, right or bottom) location. Not all skeletons may implement this and therefore may not have any effect.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
PORTLET_LABEL	VARCHAR(80)	Not Null	A moniker used to reference this portal resource for development purposes.
THEME_DEFINITION_ID	NUMBER	Null	FK to PF_THEME_DEFINITION
PARENT_PORTLET_INSTANCE_ID	NUMBER	Null	FK to PF_PORTLET_INSTANCE that identifies the parent portlet instance. NULL if this is a top level portlet instance.

The PF_PORTLET_PREFERENCE Database Table

This table identifies preference values for the portlet instance.

Table 6-68 PF_PORTLET_PREFERENCE Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_INSTANCE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PREFERENCE_NAME	VARCHAR(40)	Not Null	An optional name associated with the preference values
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
IS_MODIFIABLE	NUMBER	Not Null	A boolean, indicating whether the name/value of this preference may be modified by portlets.
IS_MULTIVALUED	NUMBER	Not Null	A boolean, indicating whether a preference may have more than one value.
PREFERENCE_DESCRIPTION	VARCHAR(255)	Null	An optional description of the portlet preferences.

The PF_PORTLET_PREFERENCE_VALUE Database Table

This table maintains values of portlet preferences. There is a one to many correspondence between the records in the PF_PORTLET_PREFERENCE table and this table.

Table 6-69 PF_PORTLET_PREFERENCE_VALUE Table Metadata

Column Name	Data Type	Null Value	Description
PORTLET_PREFERENCE_VALUE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PORTLET_INSTANCE_ID	NUMBER	Not Null	FK to PF_PORTLET_PREFERENCE
PREFERENCE_NAME	VARCHAR(40)	Not Null	FK to PF_PORTLET_PREFERENCE

Table 6-69 PF_PORTLET_PREFERENCE_VALUE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
PREFERENCE_VALUE	VARCHAR(255)	Null	The actual value for this preference.

The PF_SHELL_DEFINITION Database Table

This table represents a shell definition. There is a one-to-one correspondence between records in this table and .shell files.

Table 6-70 PF_SHELL_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
SHELL_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.

Table 6-70 PF_SHELL_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
IS_SHELL_FILE_DELETED	NUMBER	Not Null	A boolean indicating that the file associated with this shell was removed from the file system. If the shell is not being used then the record will get deleted outright. This flag is only set to true when the .shell file is deleted and the shell is still in use. You may either return the .shell file and this flag will automatically get reset, or remove the shell with a replacement in the WebLogic Administration Portal.
SHELL_FILE	VARCHAR(255)	Not Null	The name of the .shell file contained in the applications framework/markup/shell directory backing this shell definition

The PF_THEME_DEFINITION Database Table

This table represents a theme definition. There is a one to one correspondence between records in this table and .theme files.

Table 6-71 PF_THEME_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
THEME_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
INTERSECTION_ID	NUMBER	Not Null	FK to L10N_INTERSECTION
MARKUP_DEFINITION_ID	NUMBER	Not Null	FK to PF_MARKUP_DEFINITION
WEBAPP_NAME	VARCHAR(80)	Not Null	Name of the J2EE Web Application to which the portal resource is scoped.

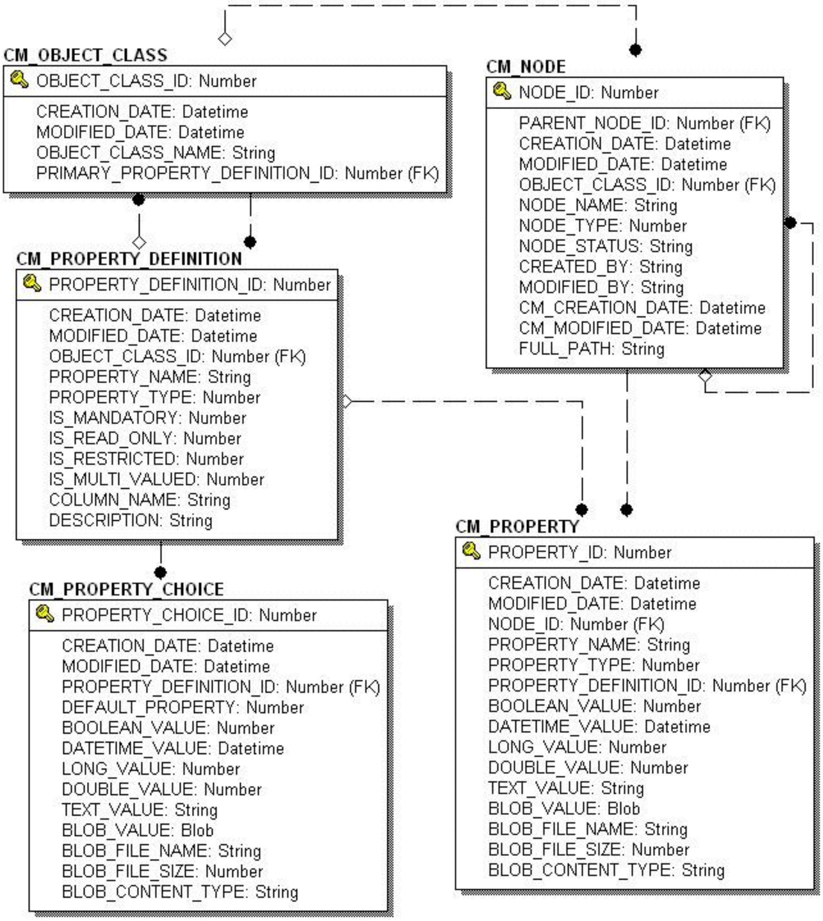
Table 6-71 PF_THEME_DEFINITION Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
`IS_THEME_FILE_DELETED`	`NUMBER`	Not Null	A boolean indicating that the file associated with this theme was removed from the file system. If the theme is not being used then the record will get deleted outright. This flag is only set to true when the .theme file is deleted and the theme is still in use. You may either return the .theme file and this flag will automatically get reset, or remove the theme in WebLogic Administration Portal.
`THEME_FILE`	`VARCHAR(255)`	Not Null	The name of the `.theme` file contained in the applications framework/markup/theme directory backing this theme definition.

Content Management Database Objects

Figure 6-10 shows the logical Entity-Relation diagram for the WebLogic Portal Content Management tables.

Figure 6-10 Entity-Relation Diagram for the Content Management Tables



The Content ManagementData Dictionary Tables

The Content Management system has the following tables:

- The CM_NODE Database Table
- The CM_OBJECT_CLASS Database Table
- The CM_PROPERTY Database Table

- The CM_PROPERTY_CHOICE Database Table
- The CM_PROPERTY_DEFINITION Database Table

The CM_NODE Database Table

In the CM_NODE table a node represents an element in a hierarchy which can either be a "Hierarchy Node" or a "Content Node". A hierarchy node can contain both other hierarchy and content nodes while a content node can only contain other content nodes. Nodes may contain Properties based on the ObjectClass (schema) defined for it.

Both Content and Hierarchy Nodes may contain an ObjectClass and Properties. All nodes have a path that uniquely identifies it within the repository.

Table 6-72 CM_NODE Table Metadata

Column Name	Data Type	Null Value	Description
NODE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
PARENT_NODE_ID	NUMBER	Null	FK-The nodes parent record ID (NODE_ID)
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
OBJECT_CLASS_ID	NUMBER	Null	FK-The object class ID associated to the node.
NODE_NAME	VARCHAR(50)	Not Null	The name of the node. The name is unique relative to it’s siblings. The name must not contain forward or backward slashes.
NODE_TYPE	NUMBER	Not Null	The node type. Either 1 for Hierarchy Node or 2 for Content Node.
NODE_STATUS	VARCHAR(40)	Null	The status of the node. The available values are defined by the application as property definition choices.

Table 6-72 CM_NODE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATED_BY	VARCHAR(100)	Not Null	The ID of the user that created the node.
MODIFIED_BY	VARCHAR(100)	Null	The ID of the user that last modified the node.
CM_CREATION_DATE	DATE	Not Null	The date and time the row was created. Maintained by the application.
CM_MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. Maintained by the application.
FULL_PATH	VARCHAR(254)	Null	AK-Each node has a path that uniquely identifies it within the repository. The path is defined in a unix-like format such as /a/b/c where "/" is the root and "a" ("a" is the Nodes NODE_NAME) is the root's child. The path must always begin with "/" and never end with it. So neither of the following are valid: a/b/c/d or /a/b/d/d/.

The CM_OBJECT_CLASS Database Table

The ObjectClass is the schema for a Node. It has both an id and a name that uniquely identifies it within a content repository. An ObjectClass may have PropertyDefinitions associated with it that define the shape of Properties required for a Node. This does not mean that the Property must contain a value, but simply that the Property must exist for the Node.

It may have a primary PropertyDefinition that defines the primary content Property for a Node. This allows for the definition of content in the schema since the schema does not distinguish between content and meta-content. A Node is only considered valid in the repository if its Properties conform to its ObjectClass PropertyDefinitions.

Table 6-73 CM_OBJECT_CLASS Table Metadata

Column Name	Data Type	Null Value	Description
OBJECT_CLASS_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.

Table 6-73 CM_OBJECT_CLASS Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
OBJECT_CLASS_NAME	VARCHAR(100)	Not Null	AK-A unique name for the object class.
PRIMARY_PROPERTY_DEFINITION_ID	NUMBER	Null	FK-The PROPERTY_DEFINITION_ID for the primary CM_PROPERTY_DEFINITION table row that defines the content for a node associated to the object class.

The CM_PROPERTY Database Table

The CM_PROPERTY table identifies a property which is a name value pair, with the name being unique relative to the CM_NODE and the value is either a Date, BLOB, Boolean, Number, Float, or Varchar.

Only one value should be set on a given row, if the value is a BLOB then all of the BLOB_ columns may be set. If the IS_MULTIVALUED column is set to 1 then there will be multiple rows with the same property name and same NODE_ID. A property may represent both the content and meta-content for a Node.

Table 6-74 CM_PROPERTY Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
NODE_ID	NUMBER	Not Null	FK-The ID of the node that contains the property

Table 6-74 CM_PROPERTY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
PROPERTY_NAME	VARCHAR(100)	Not Null	The name of the property. It must be unique relative to it's node
PROPERTY_TYPE	NUMBER	Not Null	The type of the property: BOOLEAN = 0; NUMBER = 1; FLOAT = 2; VARCHAR = 3; DATE = 4; BLOB = 5;
PROPERTY_DEFINITION_ID	NUMBER	Null	FK-The ID of the property definition to which this property must conform.
BOOLEAN_VALUE	NUMBER	Null	True (1) for the Property if the PROPERTY_TYPE is Boolean (PROPERTY_TYPE=0).
DATETIME_VALUE	DATE	Null	The datetime value for the Property if the PROPERTY_TYPE is DATE (PROPERTY_TYPE=4).
LONG_VALUE	NUMBER	Null	The long number or integer value for the Property if the PROPERTY_TYPE is NUMBER(PROPERTY_TYPE=1).
DOUBLE_VALUE	FLOAT	Null	The floating point decimal number value for the Property if the PROPERTY_TYPE is FLOAT(PROPERTY_TYPE=2).
TEXT_VALUE	VARCHAR(254)	Null	The textual property value for the Property if the PROPERTY_TYPE is VARCHAR(PROPERTY_TYPE=3).
BLOB_VALUE	BLOB	Null	The binary large object for the Property if the PROPERTY_TYPE is BLOB(PROPERTY_TYPE=5)
BLOB_FILE_NAME	VARCHAR(50)	Null	The name of the file associated with the BLOB_VALUE.
BLOB_FILE_SIZE	NUMBER	Null	The size of the file in bytes associated with the BLOB_VALUE.
BLOB_CONTENT_TYPE	VARCHAR(100)	Null	The content type (mime type and characterset) for the BLOB_VALUE. e.g. "text/html;charset=iso8859-1"

The CM_PROPERTY_CHOICE Database Table

This table identifies the valid values or choices for a PropertyDefinition (row in the CM_PROPERTY_DEFINITION table). A property choice can identify a default choice (DEFAULT_PROERTY=1) which means that if the creator of a Property does not choose different values, it will be set as a Property value.

If the PropertyChoice value is defined as NULL (no value is supplied for the PROPERTY_TYPE), it allows for an empty choice. For example, a Property that has a String type (or TEXT_VALUE) could have three PropertyChoices - "blue", "red" * and null.

Table 6-75 CM_PROPERTY_CHOICE Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_CHOICE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
PROPERTY_DEFINITION_ID	NUMBER	Not Null	FK-The ID of the property definition that contains the property choice
DEFAULT_PROPERTY	NUMBER	Not Null	Set to 1 if the property choice is a default, or 0 if it is not.
BOOLEAN_VALUE	NUMBER	Null	True (1) for the Property if the PROPERTY_TYPE is BOOLEAN (PROPERTY_TYPE=0).
DATETIME_VALUE	DATE	Null	The dietitian value for the Property if the PROPERTY_TYPE is DATE (PROPERTY_TYPE=4).
LONG_VALUE	NUMBER	Null	The long number or integer value for the Property if the PROPERTY_TYPE is NUMBER(PROPERTY_TYPE=1).
DOUBLE_VALUE	FLOAT	Null	The floating point decimal number value for the Property if the PROPERTY_TYPE is FLOAT(PROPERTY_TYPE=2).

Table 6-75 CM_PROPERTY_CHOICE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
TEXT_VALUE	VARCHAR(254)	Null	The textual property value for the Property if the PROPERTY_TYPE is VARCHAR(PROPERTY_TYPE=3).
BLOB_VALUE	BLOB	Null	The binary large object for the Property if the PROPERTY_TYPE is BLOB(PROPERTY_TYPE=5) .
BLOB_FILE_NAME	VARCHAR(50)	Null	The name of the file associated with the BLOB_VALUE.
BLOB_FILE_SIZE	NUMBER	Null	The size of the file in bytes associated with the BLOB_VALUE.
BLOB_CONTENT_TYPE	VARCHAR(100)	Null	The content type (mime type and characterset) for the BLOB_VALUE. Eg. "text/html;charset=iso8859-1"

The CM_PROPERTY_DEFINITION Database Table

The PropertyDefinition table defines the shape of a property. It describes the property type (blob, boolean, varchar, float, date, number), whether it is required, whether it is editable, the default value and restricted values, if applicable. A PropertyDefinition may have 0..n PropertyChoices.

This is a list of values that may be selected for a Property's values. Rules for a PropertyDefinition are as follows. If the PropertyDefinition contains a reference, it may not be multi-valued, or binary. If the PropertyDefinition is binary, it may not be multi-valued or restricted and may only have one PropertyChoice.

If the PropertyDefinition is boolean, it may not be multi-valued. If the PropertyDefinition is restricted then the Property's value(s) must be contained in the PropertyChoice list, or be null.

For example: consider a PropertyDefinition named "color". It has PropertyChoices "blue", "green", and "red". If the PropertyDefinition is restricted then the value of a Property defined by this PropertyDefinition may not have a value that isn't "green", "red", "blue", or null.

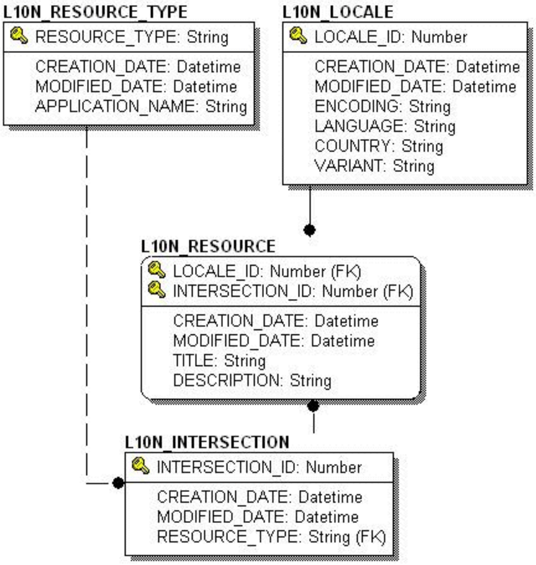
Table 6-76 CM_PROPERTY_DEFINITION Table Metadata

Column Name	Data Type	Null Value	Description
PROPERTY_DEFINITION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This columns data is maintained via a database trigger.
OBJECT_CLASS_ID	NUMBER	Not Null	FK-The OBJECT_CLASS_ID of the property definitions CM_OBJECT_CLASS.
PROPERTY_NAME	VARCHAR(100)	Not Null	The name associated with the property definition. The combination of PROPERTY_NAME AND OBJECT_CLASS_ID for an Alternate Key for the CM_PROPERTY_DEFINITION table.
PROPERTY_TYPE	NUMBER	Not Null	The type of the property; BOOLEAN = 0; NUMBER = 1; FLOAT = 2; VARCHAR = 3; DATE = 4; BLOB = 5;
IS_MANDATORY	NUMBER	Not Null	True if the value of a property must be set.
IS_READ_ONLY	NUMBER	Not Null	True if the value of a property should not be set by an end-user.
IS_RESTRICTED	NUMBER	Not Null	True if the value of a property should come from the property choice values.
IS_MULTI_VALUED	NUMBER	Not Null	True if there may be multiple rows with the same property name, node_id, but different property_IDs.
COLUMN_NAME	VARCHAR(30)	Null	The name of a column added to the CM_NODE table that defines an explicit property.
DESCRIPTION	VARCHAR(254)	Null	A description of the property definition.

Localization Database Objects

This section documents the database objects for the WebLogic Portal package. Figure 6-11 shows the Entity Relation diagram for the WebLogic Portal Localization database objects.

Figure 6-11 Entity-Relation Diagram for the Localization Tables



The Localization Dictionary Tables

The following tables support Localization:

- The L10N_INTERSECTION Database Table
- The L10N_LOCALE Database Table
- The L10N_RESOURCE Database Table
- The L10N_RESOURCE_TYPE Database Table

The L10N_INTERSECTION Database Table

This table is used to tie an application resource (menu, portlet, etc.) to a localized title and description.

Table 6-77 L10N_INTERSECTION Table Metadata

Column Name	Data Type	Null Value	Description
INTERSECTION_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
RESOURCE_TYPE	VARCHAR(80)	Not Null	FK to L10N_RESOURCE_TYPE.

The L10N_LOCALE Database Table

This table defines the characteristics of a locale that are needed to localize an application.

Table 6-78 L10N_LOCALE Table Metadata

Column Name	Data Type	Null Value	Description
LOCALE_ID	NUMBER	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
ENCODING	VARCHAR(20)	Not Null	The encoding that will be used by the locale. The default encoding is UTF-8.
LANGUAGE	CHAR(2)	Not Null	Lowercase two-letter ISO-639 language code that will be used by the locale. e.g. en, a.
COUNTRY	CHAR(2)	Null	Uppercase two-letter ISO-3166 country code that will be used by the locale. e.g. US, UK.

Table 6-78 L10N_LOCALE Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
VARIANT	VARCHAR(40)	Null	Vendor and browser specific code variant code that will be used by the locale. e.g. WIN, MAC, UNIX.

The L10N_RESOURCE Database Table

This table is used to define the localized title and description of a localized resource.

Table 6-79 L10N_RESOURCE Table Metadata

Column Name	Data Type	Null Value	Description
LOCALE_ID	NUMBER	Not Null	PK and FK to L10N_LOCALE.
INTERSECTION_ID	NUMBER	Not Null	PK and FK to L10N_INTERSECTION.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
TITLE	VARCHAR(80)	Not Null	A localized title for the object, typically used for display purposes. e.g. Name of the portal or portlet.
DESCRIPTION	VARCHAR(500)	Null	A localized description of the object.

The L10N_RESOURCE_TYPE Database Table

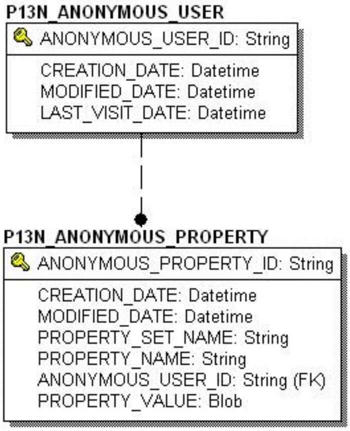
Table 6-80 L10N_RESOURCE _TYPE Table Metadata

Column Name	Data Type	Null Value	Description
RESOURCE_TYPE	VARCHAR(80)	Not Null	PK-type of Resource to be Localized. e.g. BOOK, DESKTOP, DESKTOP CATEGORY.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
APPLICATION_NAME	VARCHAR(100)	Not Null	The name of the application that the resource belongs to. APPLICATION_ NAME is currently set to PORTAL for all types of resources to be localized.

Tracked Anonymous User Database Objects

This section documents the database objects for the WebLogic Portal package. Figure 6-11 shows the Entity Relation diagram for the WebLogic Portal Anonymous User database objects.

Figure 6-12 Entity-Relation Diagram for the Anonymous User Tables



The Tracked Anonymous User Dictionary Tables

The following tables support Anonymous Users:

- The P13N_ANONYMOUS_PROPERTY Database Table
- The P13N_ANONYMOUS_USER Database Table

The P13N_ANONYMOUS_PROPERTY Database Table

This table is used store the properties associated with the tracked anonymous user.

Table 6-81 P13N_ANONYMOUS_PROPERTY Table Metadata

Column Name	Data Type	Null Value	Description
ANONYMOUS_PROPERTY_ID	VARCHAR(128)	Not Null	PK—a unique, system-generated number to be used as the record ID.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
PROPERTY_SET_NAME	VARCHAR(100)	Not Null	The name of the property set for which the tracked anonymous user data is set.

Table 6-81 P13N_ANONYMOUS_PROPERTY Table Metadata (Continued)

Column Name	Data Type	Null Value	Description
PROPERTY_NAME	VARCHAR(100)	Not Null	The name of the property.
ANONYMOUS_USER_ID	VARCHAR(128)	Not Null	The foreign key that maps to the primary key of the same.
PROPERTY_VALUE	LONG RAW	Not Null	The value ''. Must implement `java.io.Serializable`.

The P13N_ANONYMOUS_USER Database Table

This table is used to store the tracked anonymous user data.

Table 6-82 P13N_ANONYMOUS_USER Table Metadata

Column Name	Data Type	Null Value	Description
ANONYMOUS_USER_ID	VARCHAR(128)	Not Null	The foreign key that maps to the primary key of the same.
CREATION_DATE	DATE	Not Null	The date and time the row was created.
MODIFIED_DATE	DATE	Not Null	The date and time the row was last modified. This column's data is maintained via a database trigger.
LAST_VISIT_DATE	DATE	Null	Date the tracked anonymous user last updated the data.

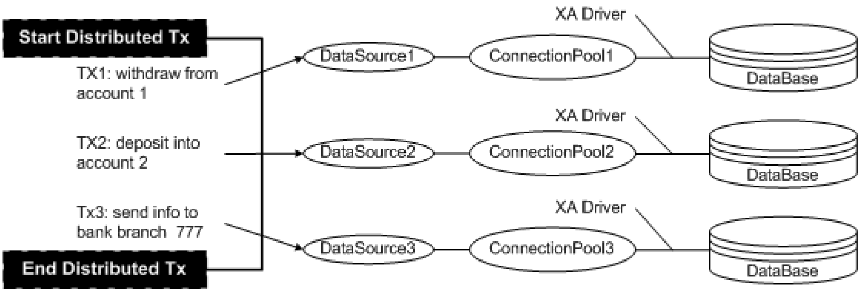
XA Support

This section describes XA support in WebLogic Portal 8.1. For additional information about using an XA configuration, see http://edocs.bea.com/platform/docs81/confgwiz/index.html.

About XA

An XA (global) transaction can span multiple databases. For example, it's common for multiple application components doing database work to be invoked as part of a single transaction. Without XA, the only way for this to work is if all transaction participants use the same database connection. With XA, you can use a TxDataSource in WLS for distributed transactions so that application components can use a different database connection for each part of the transaction.

Figure 7-1 XA Transaction Diagram



What Makes a Transaction Distributed?

WebLogic Server 8.1 enables distributed transactions by providing support for XA drivers. An XA driver can guarantee atomicity of updates across multiple participating resources. Two things are required: Configuration settings and application code.

XA transaction support requires two configuration steps:

- Change the Connection Pool driver to use an XA Driver.
- Make sure the data source for that pool is a TxDataSource instead of a DataSource.

When using an XA driver, a *TxDataSource can only point to a single connection pool.* Put another way, you cannot have multiple TxDataSources pointing to the same connection pool.

XA-Compliant Code

Configuration is only part of the process for making your application XA-compliant. You must also ensure that your application code is honoring the requirements for participating in an XA transaction. For WebLogic Portal, the following data sources associated with XA-compliant code:

- contentDataSource
- cgDataSource
- portalFrameworkPool
- commercePool

Note: If you are using commerce.jar (which is not a default part of a portal application), then the commercePool data source must point to a non-XA connection pool. Commerce code is not XA-compliant.

Using XA versus Non-XA Mode: Programming Applications

This section deals with Oracle specifically, but the programming principles apply to writing XA-compliant code in general. Applications participating in distributed transaction (XA) mode can use the JDBC 2.0 Core API the same way as in local transaction (non-XA) mode, with the following exceptions:

- Connections have to be obtained via the JDBC 2.0 javax.sql.DataSource API, but not through the deprecated java.sql.DriverManager or java.sql.Driver API.

- When used in WebLogic Server, you must access your connection pool using a TxDataSource, not a DataSource, in order to use XA.
- Attempting to enable autocommit mode by calling the java.sql.Connection.setAutoCommit method on the Connection will throw an SQLException.
- Attempting to complete the distributed transaction by calling java.sql.Connection.commit or java.sql.Connection.rollback methods will throw an SQLException.

Deciding When to Use TxDataSource Instead of DataSource

Contrary to what the name implies, a DataSource is transactional. It differs from a TxDataSource in that it can support local transactions only.

The following criteria indicate the use of a TxDataSource instead of a DataSource:

- The EJB container is managing your transactions.
- All EJBs in an application must use a TxDataSource, and each TxDataSource must point to the *same* connection pool.
- Multiple databases are updated within a single transaction
- Multiple resources, such as a database and the Java Messaging Service (JMS), are accessed during a transaction

For more information on this subject, consult the JDBC FAQ:

http://e-docs.bea.com/wls/docs81/faq/jdbc.html#499904

Using XA

This section explains some critical factors in providing XA support to your applications.

Combining XA and Non-XA Drivers in the Same Application

You can configure one connection pool to use an XA-compliant driver, and another to use a non-XA driver. Then simply point your DataSources to the appropriate pool. Local and global transactions may not be mixed using the same data source.

Mixing XA and Non-XA Transactions

You don't have to worry about transaction management, mixing XA and non-XA, if you use a TxDataSource, as described in the section Using TxDataSource on a Non-XA Connection Pool.

The transaction manager (XA) handles suspending the existing XA transaction and starts the local transaction using the DataSource (non-XA) connection. The developer, when using a DataSource to do local transaction work, must commit and close the connection (rolling back if any exceptions occur). This is because, if an XA transaction is started afterwards, the connection will still be in a 'local' mode, and the XA driver may not accept being in local mode.

Using TxDataSource on a Non-XA Connection Pool

You can use a TxDataSource to point to 'commercePool' when it's not using an XA driver. The platform detects that no XA transaction is active, and will handle things such that there is no extra overhead (such as unrequired XA-related calls).

Configuring XA

This section includes examples of config.xml files that illustrate XA configuration by example.

Note: These are only partial config.xml files to illustrate the database configurations with XA. They are not complete config.xml files

Supporting WebLogic Content Management

The Content Management mechanism provided out of the box with WebLogic Portal 8.1 employs a special datasource to track content objects. This datasource, called p13n_sequencerDataSource, must be hosted by the same database as the content management repository.

To configure the data source, go to the WLS console --> Services --> JDBC --> Data Sources and click on 'Configure a new JDBC Data Source'. Set the name of the Data Source to "p13n_sequencerDataSource" and set the JNDI name to "p13n.sequencerDataSource". The JDBC Connection Pool must match the one used by contentDataSource. Since the p13n_sequencerDataSource does not support XA, it should point to a non-xa driver.

Examples of XA Configuration

This section includes partial config.xml files meant to show what settings are required to support XA in applications.

XA for Oracle in Portal Domain

Listing 7-1 illustrates XA support for Oracle in a Portal Domain.

Note: 'cgJMSStore' and 'cgPool' will use 'auxPool'

Listing 7-1 Configuration for XA on Oracle in a Portal Domain

```
<?xml version="1.0" encoding="UTF-8"?>
<Domain Name="portalDomain">
    <!-- This pool should NEVER be configured with an XA driver -->
    <JDBCConnectionPool CapacityIncrement="1"
        DriverName="oracle.jdbc.driver.OracleDriver"
        InitialCapacity="5" MaxCapacity="50" Name="auxPool"
        Properties="user=<yourUserName>;" Password="<yourPassword>"
        Targets="<yourServerName>"
URL="jdbc:oracle:thin:@<TNSNAME>:<port>:<dbHostMachineName>"/>
<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it. In this case, it's the
'commercePool' TxDataSource -->
    <JDBCConnectionPool CapacityIncrement="1"
        DriverName="oracle.jdbc.xa.client.OracleXADataSource"
        KeepXAConnTillTxComplete="true"
        InitialCapacity="5" MaxCapacity="20" Name="portalPool"
        Properties="user=<yourUserName>;" Password="<yourPassword>"
        Targets="<yourServerName>"
URL="jdbc:oracle:thin:@<TNSNAME>:<port>:<dbHostMachineName>"/>
<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it. In this case, it's the
'contentDataSourcl' TxDataSource -->
    <JDBCConnectionPool CapacityIncrement="1"
        DriverName="oracle.jdbc.xa.client.OracleXADataSource"
        KeepXAConnTillTxComplete="true"
        InitialCapacity="5" MaxCapacity="15" Name="contentPool"
        Properties="user=<yourUserName>;" Password="<yourPassword>"
        RefreshMinutes="0" Targets="<yourServerName>"
        URL="jdbc:oracle:thin:@<TNSNAME>:<port>:<dbHostMachineName>"/>

<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it.  In this case, it's the
''portalFrameworkPool" TxDataSource  -->
    <JDBCConnectionPool CapacityIncrement="1"
        DriverName="oracle.jdbc.xa.client.OracleXADataSource"
        KeepXAConnTillTxComplete="true"
        InitialCapacity="5" MaxCapacity="20" Name="portalFrameworkPool"
        Properties="user=<yourUserName>;" Password="<yourPassword>"
        Targets="<yourServerName>"
        URL="jdbc:oracle:thin:@<TNSNAME>:<port>:<dbHostMachineName>"/>

<!-- First, the non-XA data sources.  These must only point to 'auxPool' -->
    <JDBCDataSource
        JNDIName="p13n.trackingDataSource"
        Name="p13n_trackingDataSource" PoolName="auxPool"
        Targets="portalServer"/>
       <JDBCDataSource JNDIName="p13n.sequencerDataSource"
        Name="p13n_sequencerDataSource" PoolName="auxPool"
        Targets="portalServer"/>
       <JDBCDataSource JNDIName="p13n.dataSyncDataSource"
        Name="p13n_dataSyncDataSource" PoolName="auxPool"
        Targets="portalServer"/>

<!-- Although ebusiness can't partipate in XA, it must still use a
TxDataSource (because it uses EJBs), and point to the non-XA pool -->
```

```
    <JDBCTxDataSource
        JNDIName="weblogic.jdbc.jts.ebusinessPool" Name="ebusinessPool"
        PoolName="auxPool" Targets="portalServer"/>

<!-- Next, the XA data sources.  Each one must point to its own connection
pool -->
    <JDBCTxDataSource
        JNDIName="weblogic.jdbc.jts.commercePool" Name="commercePool"
        PoolName="portalPool" Targets="portalServer"/>
    <JDBCTxDataSource
        JNDIName="contentDataSource" Name="contentDataSource"
        PoolName="contentPool" Targets="portalServer"/>
    <JDBCTxDataSource
        JNDIName="portalFrameworkPool" Name="portalFrameworkPool"
        PoolName="portalFrameworkPool" Targets="portalServer"/>

<!-- The JMS data stores should all point to the non-XA pool and have a null
prefix -->
     <JMSJDBCStore ConnectionPool="auxPool" Name="cgJMSStore"
PrefixName=""/>
</Domain>
```

XA for SQL Server in a Portal Domain

Note: 'cgJMSStore' and 'cgPool' will use 'auxPool'

Listing 7-2 Configuration Settings for XA on SQL Server in a Portal Domain

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Domain Name="portalDomain">

    <!-- This pool should NEVER be configured with an XA driver -->

    <JDBCConnectionPool CapacityIncrement="1"

        DriverName="weblogic.jdbc.sqlserver.SQLServerDriver"

        InitialCapacity="5" MaxCapacity="50" Name="auxPool"

        Properties="user=<yourUsername>;ServerName=<TNSNAME>;

        PortNumber=<port>" Password="<yourPassword>"

        Targets="<yourServerName>"

        URL="jdbc:bea:sqlserver://<dbHostMachineName>:<port>"/>


<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it.  In this case, it's the
'commercePool' TxDataSource -->

    <JDBCConnectionPool CapacityIncrement="1"

        DriverName="weblogic.jdbcx.sqlserver.SQLServerDataSource"

        KeepXAConnTillTxComplete="true"

        InitialCapacity="5" MaxCapacity="20" Name="portalPool"

        Properties="user=<yourUsername>;ServerName=<TNSNAME>;

        PortNumber=<port>" Password="<yourPassword>"

        Targets="<yourServerName>"

        URL="jdbc:bea:sqlserver://<dbHostMachineName>:<port>"/>


<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it.  In this case, it's the
'contentDataSourcl' TxDataSource -->

    <JDBCConnectionPool CapacityIncrement="1"

        DriverName="weblogic.jdbcx.sqlserver.SQLServerDataSource"

        KeepXAConnTillTxComplete="true"
        InitialCapacity="5" MaxCapacity="15" Name="contentPool"
        Properties="user=<yourUsername>;ServerName=<TNSNAME>;
        PortNumber=<port>" Password="<yourPassword>"
        Targets="<yourServerName>"
        URL="jdbc:bea:sqlserver://<dbHostMachineName>:<port>"/>

<!-- This pool has been configured with an XA driver.  As for all XA pools,
only one TxDataSource may connect to it.  In this case, it's the
''portalFrameworkPool" TxDataSource  -->
    <JDBCConnectionPool CapacityIncrement="1"
        DriverName="weblogic.jdbcx.sqlserver.SQLServerDataSource"
        KeepXAConnTillTxComplete="true"
        InitialCapacity="5" MaxCapacity="20" Name="portalFrameworkPool"
        Properties="user=<yourUsername>;ServerName=<TNSNAME>;
        PortNumber=<port>" Password="<yourPassword>"
        Targets="<yourServerName>"
        URL="jdbc:bea:sqlserver://<dbHostMachineName>:<port>"/>
<!-- First, the non-XA data sources.  These must only point to 'auxPool' -->
    <JDBCDataSource
        JNDIName="p13n.trackingDataSource"
        Name="p13n_trackingDataSource" PoolName="auxPool"
Targets="portalServer"/>
    <JDBCDataSource JNDIName="p13n.sequencerDataSource"
        Name="p13n_sequencerDataSource" PoolName="auxPool"
Targets="portalServer"/>
    <JDBCDataSource
        JNDIName="p13n.dataSyncDataSource"
        Name="p13n_dataSyncDataSource" PoolName="auxPool"
Targets="portalServer"/>
```

```
<!-- Although ebusiness can't partipate in XA, it must still use a
TxDataSource (because of it uses EJBs), and point to the non-XA pool -->
    <JDBCTxDataSource
        JNDIName="weblogic.jdbc.jts.ebusinessPool" Name="ebusinessPool"
        PoolName="auxPool" Targets="portalServer"/>
<!-- Next, the XA data sources.  Each one must point to its own connection
pool -->
    <JDBCTxDataSource
        JNDIName="weblogic.jdbc.jts.commercePool" Name="commercePool"
        PoolName="portalPool" Targets="portalServer"/>
    <JDBCTxDataSource
        JNDIName="contentDataSource" Name="contentDataSource"
        PoolName="contentPool" Targets="portalServer"/>
    <JDBCTxDataSource
        JNDIName="portalFrameworkPool" Name="portalFrameworkPool"
        PoolName="portalFrameworkPool" Targets="portalServer"/>
<!-- The JMS data stores should all point to the non-XA pool and have a null
prefix -->
     <JMSJDBCStore ConnectionPool="auxPool" Name="cgJMSStore"
PrefixName=""/>
</Domain>
```

XA for Sybase in a Portal Domain

1. For XA transaction support, install the license for Distributed Transaction Management (DTM) and configure the Sybase server as follows:

   ```
   sp_configure "enable DTM",1
   go
   ```

```
sp_configure "enable xact coordination",1

go

    sp_configure "dtm lock timeout",30

go
```

2. Copy the sample xa_config file from the SYBASE_INSTALL\OCS-12_0\sample\xa-dtm subdirectory up three levels to SYBASE_INSTALL,where SYBASE_INSTALL is the directory of your Sybase server installation. For example: $SYBASE_INSTALL\xa_config.
3. Edit the xa_config file. In the first [xa] section, modify the sample server name to reflect the correct server name. See the "Using Adaptive Distributed Transaction Features" Sybase product manual for additional information.
4. To prevent deadlocks when running transactions, enable row level locking by default:

```
sp_configure "lock scheme",0,datarows

go
```

APPENDIX A

WebLogic Portal DDL Modules

WebLogic Portal DDL Modules

WebLogic Portal Database Definition Language (DDL) modules are provided in directories with the following format:

```
<WL_HOME>/portal/db/<dbms name>/<dbms version>
```

For example:

```
<WL_HOME>/portal/db/pointbase/44
<WL_HOME>/portal/db/oracle/817
<WL_HOME>/portal/db/oracle/9i
```

Note: The same WebLogic Portal DDL is used for both Oracle 8.1.7 and 9i databases as indicated by a readme.txt file in the oracle/9i directory.This directory naming structure offers the ability to have distinct DDL between dbms versions.

Data inserts for bootstrap data that must be inserted into tables in each WebLogic Portal databases are contained in the following:

```
<WL_HOME>/portal/db/data/required/xx_insert_system_required_data.sql
```

WebLogic Portal DDL is provided in files named as follows:

```
xx_create_fkeys.sql
xx_create_indexes.sql
xx_create_tables.sql
xx_create_triggers.sql
```

```
xx_create_views.sql
xx_drop_constraints.sql
xx_drop_fkeys.sql
xx_drop_indexes.sql
xx_drop_tables.sql
xx_drop_views.sql
```

where the xx is a prefix from the table below:

Prefix	Description
au	Anonymous user
bt	Behavior Tracking
cm	Content Management
collaboration*	Compoze portlets
doc**	Document management
ds	Data synchronization
er**	Entitlement ruleset
p13n	WebLogic Portal Personalization
pf	WebLogic Portal Framework and Localization
portal**	WebLogic Portal 7.0
sample_cm	Content Management types data
sample_portal**	Sample Portal 7.0
wlcs	WebLogic Commerce Services
wps	WebLogic Portal Services

Table notes:

* Database object definitions for portlets from Compoze Software

** Database object definitions for WebLogic Portal 7.0 which has been retained for Compatibility Mode