BEA WebLogic Portal

Migration Guide

Release 7.0 Service Pack 7
Document Date: March 2006

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Migration Guide

Part Number	Date	Software Version
N/A	March 2006	7.0 Service Pack 7

Contents

A. Viewing Code Changes Using the Migration Viewer Tool

B. Migration Files

Index

Preface

Welcome to the Migration Guide. This guide describes the steps for migrating from Release 4.0 to 7.0.

We encourage you to use the following resources, as well.

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

Providing documentation feedback 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. Please indicate that you are using the documentation for the BEA WebLogic Portal **Product Version: 7.0** release.

Contacting BEA WebSUPPORT 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 through BEA WebSUPPORT at **http://www.bea.com** or by using the contact information provided on the Customer Support Card in the product package.

CHAPTER

1 Migrating From WebLogic Portal 7.0 to WebLogic Portal 7.0 Service Packs

Migrating to Service Pack 7

BEA WebLogic Portal 7.0 Service Pack 7 (SP7) WebLogic Portal 7.0 compliant with JDK 1.4.2. No new functionality or major behavior changes were introduced.

It also incorporates service pack upgrades for all WebLogic Platform 7.0 components (WebLogic Server, WebLogic Workshop, WebLogic Integration, WebLogic Portal and WebLogic JRockit). WebLogic Platform 7.0 SP7 can be used to update all WebLogic Platform 7.0 components.

When migrating an application from 7.0, 7.0 SP1, 7.0 SP2 to 7.0 SP4, 7.0 SP4 to 7.0 SP5, 7.0 SP5 to 7.0 SP6, or 7.0 SP6 to 7.0 SP7, follow the instructions for migrating a WebLogic Portal domain at http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration.

Migrating to Service Pack 6

BEA WebLogic Platform 7.0 Service Pack 6 (SP6) incorporates service pack upgrades for all WebLogic Platform 7.0 components (WebLogic Server, WebLogic Workshop, WebLogic Integration, WebLogic Portal and WebLogic JRockit).

WebLogic Platform 7.0 SP6 can be used to update all WebLogic Platform 7.0 components.

When migrating an application from 7.0, 7.0 SP1, 7.0 SP2 to 7.0 SP4, 7.0 SP4 to 7.0 SP5, or 7.0 SP5 to 7.0 SP6, follow the instructions for migrating a WLP domain at http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration.

Migrating to Service Pack 4

BEA WebLogic Platform 7.0 Service Pack 4 (SP4) incorporates service pack upgrades for all WebLogic Platform 7.0 components (WebLogic Server, WebLogic Workshop, WebLogic Integration, WebLogic Portal and WebLogic JRockit). BEA WebLogic Platform 7.0 SP4 is the first full service pack release since WebLogic Platform 7.0 SP2.

Note: A separate 7.0 SP3 release of WebLogic Server/WebLogic Workshop is available to WebLogic Server and WebLogic Workshop support customers only. A separate WebLogic JRockit 7.0 SP3 release is also available to customers using WebLogic Server 7.0 SP3. The WebLogic Server 7.0 SP3 upgrade installers do not allow SP3 to be applied to an existing WebLogic Integration 7.0, WebLogic Portal 7.0 or full WebLogic Platform 7.0 installation.

WebLogic Platform 7.0 SP4 can be used to update all WebLogic Platform 7.0 components.

When migrating an application from 7.0, 7.0 SP1, or 7.0 SP2 to 7.0 SP4, follow the instructions for migrating a WLP domain at http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration.

Migrating to Service Pack 2

When migrating your 7.0 application to 7.0 SP1, or when migrating your 7.0 or 7.0 SP1 application to 7.0 SP2, follow the instructions for migrating a WLP Domain at http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration.

Migrating to Service Pack 1

The Configuration Wizard (introduced in WebLogic Platform 7.0) allows you to create new domains quickly and easily. If you created domains using the Configuration Wizard in WebLogic Platform 7.0, you need to migrate those domains for use with WebLogic Platform 7.0 Service Pack 1.

For most domains, migration from WebLogic Portal 7.0 to WebLogic Portal 7.0 Service Pack 1 is a three-step process:

- Step 1: Upgrade Product JAR Files in the domain directory. A migration script is provided for this purpose.

  Note: You can also revert a domain to its pre-migration state.

- Step 2: Update the Domain to Support Service Pack 1 Changes. Depending on the domain template used to generate the domain, you may need to add or modify existing scripts or files.

- Step 3: Update Startup Scripts and Configuration Files to Reference New BEA_HOME Directory Location (Non-Upgrades Only). Perform this step only if you installed WebLogic Platform 7.0 Service Pack 1 into a new directory.

  Note: If you *upgraded* your existing WebLogic Platform 7.0 installation, you can skip this step.

These steps are explained in detail in the following sections. You will need to repeat this process for *each* domain that you want to migrate.

Note: This section decribes how to migrate domains specific to WebLogic Portal. For information about migrating other WebLogic Platform domains, see "Migrating Domains Created Using the Configuration Wizard" in the *WebLogic Platform 7.0 Service Pack 1 Release Notes* at the following URL:

http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration

Step 1: Upgrade Product JAR Files

To upgrade product JAR files for a domain that you generated using the Configuration Wizard to Service Pack 1, navigate to the *BEA_HOME*\weblogic700\server\bin directory and execute one of the following commands:

Windows: `migrate.cmd` *`domain mode`*
UNIX: `migrate.sh` *`domain mode`*

Note: You will be prompted to press any key to start processing.

The following table defines the command-line arguments.

Argument	Description
domain	Full pathname of the domain directory.
mode	Migration mode. The mode can be set to one of the following values: ■ `upgrade`: Upgrade the product JAR files in the domain directory, as required. The original product JAR files are saved as `*.jar.orig`. If the timestamp of an existing product JAR file is more recent than the timestamp on the corresponding SP1 installation product JAR file, the file is skipped. This is the default mode. ■ `revert`: Reverts a domain that was migrated earlier using the backup files (`*.jar.orig`) generated. If no `*.jar.orig` files exist, the command is ignored.

For example, the following command upgrades a domain called `mydomain` located in the default user projects directory (*BEA_HOME*\user_projects):

Windows: `migrate.cmd c:\bea\user_projects\mydomain upgrade`
UNIX: `migrate.sh c:/bea/user_projects/mydomain upgrade`

The following command reverts the changes made to `mydomain` during the migration:

Windows: `migrate.cmd c:\bea\user_projects\mydomain revert`
UNIX: `migrate.sh c:/bea/user_projects/mydomain revert`

Step 2: Update the Domain to Support Service Pack 1 Changes

To update a domain that is based on the WLP Domain template to support WebLogic Platform 7.0 Service Pack 1, perform the following steps.

Note: Before adding or modifying any files, it is recommended that you backup the original files.

1. In the `web.xml` file for the tools Web application (located, by default, in the *BEA_HOME*`\user_projects\`*domain*`\beaApps\portalApp\tools\WEB-INF` directory), locate the `Customer Profile and Order Pages` security constraint and define the resources to which the security constraint applies using the `<url-pattern>` element.

The following sample excerpt from the `web.xml` file shows the required updates in **bold**:

```
<security-constraint>
  <!-- Define a resource collection -->
  <web-resource-collection>
    <web-resource-name>
      Customer Profile and Order Pages
    </web-resource-name>
    <description>
      Customer Profile and Order Pages
    </description>
  <!-- URL pattern for the resource collection -->
    <url-pattern>/tools/*</url-pattern>
    <url-pattern>/repository/*</url-pattern>
    <url-pattern>/security/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
</security-constraint>
```

Note: WebLogic Server validates each `web-resource-collection` element within a `security-constraint` element to ensure that it contains at least one URL pattern. If you have other Web applications in your domain, verify that all `web-resource-collection` elements contain at least one URL pattern, appropriate for the security constraint.

2. Copy the following files from the *BEA_HOME*\weblogic700\common\templates\domains\shared\bea\portal\webapps\tools\tools directory to the `tools` directory of your tools Web applications. Be careful not to overwrite any files that you have customized.

```
catalog\category_add_remove_items.jsp
catalog\item_property_edit.jsp
catalog\item_property_edit_mr.jsp
catalog\item_property_edit_mu.jsp
catalog\item_property_edit_sr.jsp
catalog\item_property_edit_su.jsp
catalog\item_search.jsp
usermgmt\groupuser_property_edit_mr.jsp
usermgmt\groupuser_property_edit_mu.jsp
usermgmt\groupuser_property_edit_sr.jsp
usermgmt\groupuser_property_edit_su.jsp
usermgmt\group_add_remove_users.jsp
usermgmt\group_edit.jsp
usermgmt\group_scope_property.jsp
usermgmt\user_create.jsp
```

```
usermgmt\user_edit_info.jsp
usermgmt\user_scope_property.jsp
```

3. If you have created a portal Web application in your domain, you need to perform the following steps for each application:

   a. Copy the following files from the *BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\framework directory to the framework directory of your portal Web application. Be careful not to overwrite any files that may have been created using one of these filenames.

   ```
   edit_titlebar.properties
   error\configurationerror.properties
   error\footer.inc
   error\header.inc
   error\header.properties
   error\missingformfield.properties
   error\parameters.properties
   error\pipeline.properties
   error\request.properties
   error\runtimeerror.properties
   hnav_bar.properties
   maximize_titlebar.properties
   minimize_titlebar.properties
   normal_titlebar.properties
   security\help.properties
   security\meta.inc
   vnav_bar.properties
   ```

   b. Update the following JSP files in the framework directory of your portal Web application using the corresponding files in the *BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\framework directory as a guide for comparison. Because you may have modified the JSP files to customize your portal, it is not recommended that you overwrite the existing files unless you are certain that they have not been customized.

   ```
   edit_titlebar.inc
   error\configurationerror.jsp
   error\error.jsp
   error\missingformfield.jsp
   error\parameters.jsp
   error\pipeline.jsp
   error\request.jsp
   error\runtimeerror.jsp
   error\sessiontimeout.jsp
   error\sessiontimeout.properties
   ```

```
floated_portlet.jsp
hnav_bar.jsp
maximize_titlebar.inc
minimize_titlebar.inc
normal_titlebar.inc
security\help.jsp
security\login_header.inc
security\need_group.jsp
security\new_user.jsp
security\set_password.jsp
tools\header.jsp
tools\header.properties
tools\portal_prefs.jsp
vnav_bar.jsp
```

c. Copy the following files from the *BEA_HOME*\weblogic700\common\templates\webapps\portal\baseportal\j2ee\WEB-INF\lib directory to the WEB-INF\lib directory of your portal Web applications. Be careful not to overwrite any files that you have customized.

```
ent_taglib.jar
es_taglib.jar
i18n_taglib.jar
lic_taglib.jar
p13n_servlet.jar
portal_servlet.jar
portal_taglib.jar
portlet_taglib.jar
ren_taglib.jar
res_taglib.jar
um_taglib.jar
util_taglib.jar
visitor_taglib.jar
webflow_servlet.jar
webflow_taglib.jar
weblogic-tags.jar
```

Step 3: Update Startup Scripts and Configuration Files to Reference New BEA_HOME Directory Location (Non-Upgrades Only)

Note: This step is only required if you installed WebLogic Platform 7.0 Service Pack 1 into a new directory that is separate from the WebLogic Platform 7.0 installation. If you *upgraded* your existing WebLogic Platform 7.0 installation, you can skip this step.

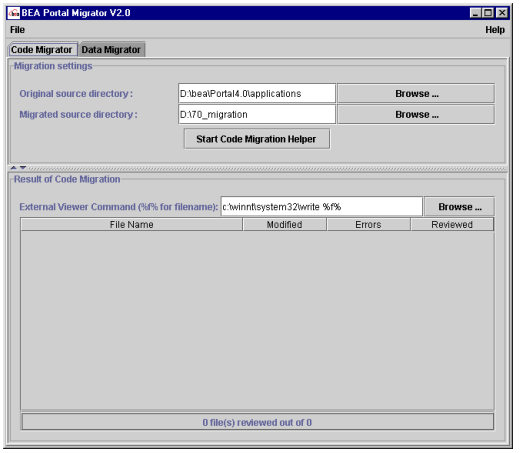
The domain startup scripts (such as, `startWebLogic`) and configuration files (such as `config.xml`) define the full pathnames to files within the *BEA_HOME* directory. You need to search for and update these full pathnames to reference the new *BEA_HOME* directory location. In addition, you must update any custom scripts, such as build scripts, that define full pathnames to the files within the *BEA_HOME* directory to reflect the new *BEA_HOME* location.

Note: Many startup scripts set environment variables in your current shell, including variables that reference your *BEA_HOME* directory. After updating the *BEA_HOME* references in script files, you should open a new shell to ensure that the latest environment settings are used.

2 Migrating From WebLogic Portal 4.0 to WebLogic Portal 7.0

This section provides information on migration from WebLogic Portal Release 4.0 to 7.0. To assist in the migration process, you will use the migration tool, as shown in Figure 2-1,

Figure 2-1 Migration Tool Code Migrator Window



The migration process consists of completing a few pre-migration steps, then migrating code, migrating data, and assembling the migrated files in the 7.0 configuration. This chapter includes the following subjects:

- Step 1: Prepare Environment and Files for Migration

Note: Step 5 contains essential tasks that you must perform in order to make your migrated 7.0 implementation run. For additional migration tasks, check the *troubleshooting* index entry. Also, for additional migration tips, check the Customer Support (http://websupport.bea.com/welcome.jsp) and BEA dev2dev (http://developer.bea.com/index.jsp) Web sites.

Step 1: Prepare Environment and Files for Migration

This section contains information on the following subjects:

Obtain Latest Documentation and Associated Files

To get the latest documentation and associated files, check the following:

- Release Notes for the main release and service packs.

  The release notes for any service pack provides important information about changes to the software. Review them before completing a migration. Service packs are available at the following sites:

  - General Downloads – http://commerce.bea.com/downloads/products.jsp
  - Customer Support (upgrade installers only) – http://websupport.bea.com/welcome.jsp. (You must have a valid support contract to get to the support site.)

- BEA product documentation (edocs) and BEA dev2dev Web sites at the following locations:

  - BEA Product Documentation – http://edocs.bea.com.
  - BEA dev2dev – http://developer.bea.com/index.jsp

Check Patches You Have Installed

Before you install a service pack, determine which patches are currently installed. This includes patches for all WebLogic products you use, such as WebLogic Server.

The WebLogic Portal migration is based on a WebLogic Portal 4.0 installation with the patches provided in Service Pack 2. However, before you migrate, you need to evaluate all your WebLogic patches because some patches are distributed outside of service packs for WebLogic Portal and other WebLogic products. For instance, a different version of a JAR file that the migration relies on might be included in a non-service pack patch that you have installed. Contact BEA Customer Support (http://www.bea.com/support/programs.shtml)and with their assistance determine all patches that you have installed, whether all those patches have been included in service packs, and if not determine any migration-related issues with those patches.

Address Issues Related to Migration

Carefully read each subject in the following list and take the appropriate steps, before or during the migration process, as indicated by your implementation and how each issue affects it:

- Migration Interaction With WebLogic Integration
- Important Information About the Database Migration and Sybase
- Notes on Triggers and WebLogic Portal 4.0 Service Pack 1
- Details of the E-Business Control Center Project Migration

Migration Interaction With WebLogic Integration

If you already have WebLogic Portal or WebLogic Integration and migrate to the WebLogic 7.0 Platform, you will have to migrate both WebLogic Portal and WebLogic Integration. The two products share the WebLogic Portal RDBMS realm. The configured RDBMS should contain WebLogic Integration predefined data. The installed fileRealm from WebLogic Integration should only contain the system user for WebLogic Integration and no other pre-defined data from WebLogic Integration.

These configuration issues are affected by migration as well as the Configuration Wizard. For complete information, refer to the WebLogic Integration *Platform Security Guide* at edocs.bea.com/wls/docs70/security.html.

Important Information About the Database Migration and Sybase

Sybase has relaxed its limits on column size, and WebLogic Portal has taken advantage of these relaxed limits.

If you are using Sybase, you must follow these steps in this order:

If you are currently using Sybase 11.9 or 12.0, migrate from 4.0 to 7.0. If you decide to update to Sybase 12.5, download the Sybase schema update scripts from BEA dev2dev (http://developer.bea.com/index.jsp), run them, and then update to 12.5.

If you are currently using Sybase 12.5, migrate from 4.0 to 7.0. Then download the Sybase schema update scripts from the BEA dev2dev Web site and run them.

Notes on Triggers and WebLogic Portal 4.0 Service Pack 1

Service Pack 1 omitted triggers that can negatively affect an implementation. If you are only using Service Pack 1, install Service Pack 2 before migrating.

Service packs are available through the following sites:

- General Downloads: http://commerce.bea.com/downloads/products.jsp
- Support Site (Upgrade Installers Only) http://support.bea.com/welcome.jsp

About the E-Business Control Center Project Migration

Typically, your E-Business Control Center project migration should go smoothly. However, some variations in content or currentness can produce unexpected results. If the project is not complete, the validity of the content will not be checked when it is migrated, so any corrupted files will be migrated. To avoid this, complete all work on E-Business Control Center projects before migrating. After migrating, check the projects to ensure that they function as expected, and make any necessary changes using the administration tools.

For more detail about how the projects are migrated, see "Details of the E-Business Control Center Project Migration" on page B-17.

Migrating the Default Configuration Setting for Portals

Each portal has a default configuration, which is used when an anonymous user accesses the portal. The default configuration, along with other data about the portal, is migrated in this release.

To ensure that the default configuration is migrated, sync your portal configurations to the server before you begin. If you do not do so, or if for any other reason a migrated portal does not have a default configuration set, a *default* default configuration is assigned to the portal. If you want to change the default configuration, use the WebLogic Portal Administration Tools.

Prepare the Environment

This section contains information on the following subjects:

Check Supported Platforms

Ensure that your system conforms to the specifications in the supported platforms document for WebLogic Portal 7.0, available at http://edocs.bea.com/platform/docs70/support/index.html.

Install Latest Service Pack

Be sure that you have installed the latest service pack. The instructions in this guide are written based on the assumption that you are migrating from Service Pack 2 (SP2).

Install BEA WebLogic 7.0 and Set Up Build Environment

Follow the instructions in "Installing WebLogic Platform" at http://edocs.bea.com/platform/docs70/install/index.html and any other referenced documentation in that guide to install release 7.0 correctly on your system. Be sure that your installation environment and the software are installed and set up correctly before you continue.

Set up your 7.0 build environment system—creating `.jar` files and other files—so that you will be able to modify, test, and deploy converted code files later in the migration process. This is your own build environment that you create and provide yourself.

Warning: Do not run the create_all script on your 4.0 database. You can run it to create an empty database, point your server to it, and ensure that 7.0 is installed and running correctly before you begin migrating.

Note: You will not be able to run the WebLogic Server 7.0, pointed to your 4.0 database, before you finish migrating WebLogic Portal to 7.0.

Make One or More Complete Backups of Code, Data, and Database

Completely back up your entire WebLogic Portal 4.0 system at least once, including your database schema. If some migration processes fail, you will need to restore the backup and migrate again.

Check Migration Environment

Be sure that the migration files are installed on the same computer where WebLogic Server is installed. The migration requires access to the `weblogic.jar` and license files in the BEA directory.

Sync E-Business Control Center Projects to Server

Make sure the latest version of any portal file is synced to the server so that the default configuration is captured.

Determine Whether to Modify DataMigratorBundle.properties or MigratorBundle.properties

These files control significant aspects of the migration. If you need to skip tasks in the Data Migrator (provides the ability to be done out of order) or if you need to internationalize the migration tool, see "Migration Files" on page B-1.

Warning: In a typical migration, you do not need to modify these files. Modifying them should be done only if you are an accomplished developer and have changes you must make to the migrator tool or process.

Edit the migrator.bat and migration_install.properties Files

Enter configuration settings as follows:

1. Open the <PORTAL_HOME>\migration\bin\migrator.bat or migrator.sh file. Comment out the appropriate line to indicate the database you are using, shown in Listing 2-1.

Listing 2-1 migrator.bat File

```
echo off
REM -------------------------------------------------------#
REM  Migrator Starter Script on Windows                     #
REM -------------------------------------------------------#
SETLOCAL

set DATABASE=ORACLE_THIN
REM set DATABASE=MSSQL
REM set DATABASE=SYBASE_JCONNECT
REM set DATABASE=DB2_TYPE2
if "%DATABASE%" == "" ( echo "The DATABASE variable must be uncommented in
migrator.bat"
    exit 1 )

CALL ..\..\bin\win32\set-environment.bat

REM -------------------------------------------------------#
REM         VARIABLES TO SET
REM -------------------------------------------------------#

REM Due to database specifics, you may have to set your
REM path to include dll

REM -------------------------------------------------------#
REM          The mini script
REM -------------------------------------------------------#
set MIGRATION_DIR=%WL_COMMERCE_HOME%\migration
set MIGRATION_LIB=%MIGRATION_DIR%\lib
set
MIG_CLASSPATH=%BEA_HOME%\lib\tools.jar;%MIGRATION_LIB%\migration.jar;%MIGRATION
_LIB%\apache\xerces-1_4_3\xerces.jar;%MIGRATION_LIB%\apache\xalan-j_2_0_1\xalan
.jar;%MIGRATION_LIB%\p13n_system.jar;%WEBLOGIC_HOME%\lib\weblogic.jar;%BEA_HOME
%

REM -------------------------------------------------------#
REM For testing the setup only
REM -------------------------------------------------------#
REM echo BEA_HOME=%BEA_HOME%
REM echo WEBLOGIC_HOME=%WEBLOGIC_HOME%
REM echo WL_COMMERCE_HOME=%WL_COMMERCE_HOME%
```

```
REM echo MIGRATION_DIR=%MIGRATION_DIR%
REM echo MIGRATION_LIB=%MIGRATION_LIB%
REM echo MIG_CLASSPATH=%MIG_CLASSPATH%


%JDK_HOME%\bin\java -cp %MIG_CLASSPATH%  -DMIGRATION_DIR=%migration_dir%
com.bea.commerce.migration.tools.Migrator

echo on
```

2. Open the <PORTAL_HOME>\migration\migration_install.properties file. Listing 2-2 shows the file; the items you need to address are in bold.
3. Set the line migration_start to true.
4. Uncomment the database you are using. Ensure that all other databases are commented out.
5. Change the database connection properties, shown in Listing 2-2.

Notes: The only database.version setting for oracle can be 817. If you use 9I, you should still set it to 817.

SQL Server may be set to 7 or 2000 by un-commenting the appropriate version.

For the value server=**YOURSERVER**, enter the name of the database server. For instance, for Oracle, enter the Oracle SID or the netservice name, depending on how your system is configured.

Listing 2-2 migration_install.properties File

```
#####################################################################
#
# PROPERTIES TO BE SET BY THE USER AT 'INSTALL' TIME
#
#####################################################################

# ------------------------------------------------------------
#       When you have set the properties you need, set the
#       following flag to 'true' so the migrator tool will start.
#       Otherwise, the Migrator will assume you have NOT set
#       any property and will refuse to run.
```

```
# ----------------------------------------------------------------
start_migrator=false

# ----------------------------------------------------------------
#Database Properties  // uncomment the database you are using
                      // and enter the appropriate connection values
# ----------------------------------------------------------------


# Database connection properties
#
#------Oracle Thin Driver----------------------#
#
# For oracle, replace the following:
#    @USER@, @PASSWORD@, @SERVER@, @PORT@, and @SID@
#    e.g. jdbc:oracle:thin:@localhost:1521:ORCL
#
database.connection.driver = oracle.jdbc.driver.OracleDriver
database.connection.url = jdbc:oracle:thin:@@SERVER@:@PORT@:@SID@
database.connection.props = user=@USER@;password=@PASSWORD@


#------MS SQL Server ----------------------#
#
# For SQL Server, replace the following:
#    @SERVERPORTNUMBER@, @USER@, @PASSWORD@, and @SERVER@ (in two different
locations)
#
#database.connection.driver = weblogic.jdbc.mssqlserver4.Driver
#database.connection.url =
jdbc:weblogic:mssqlserver4:@SERVER@:@SERVERPORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999


#------Sybase jConnect 5.2 ----------------------#
#
# For Sybase, replace the following:
#    @SERVER@, @PORTNUMBER@, @USER@, and @PASSWORD@
#
#database.connection.driver=com.sybase.jdbc2.jdbc.SybDriver
#database.connection.url=jdbc:sybase:Tds:@SERVER@:@PORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
```

```
#------IBMS's DB2 --------------------------------------#
#
# For DB2, replace the following:
#    @DB2_DATABASE@, @USER@, and @PASSWORD@
#
#database.connection.driver=COM.ibm.db2.jdbc.app.DB2Driver
#database.connection.url=jdbc:db2:@DB2_DATABASE@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999


#
# Database and version
#
# for Oracle users use database.version=817 for either Oracle 8.1.7 or 9i
database.name=oracle
database.version=817
#database.name=sql_server
#database.version=2000
#database.version=7
#database.name=sybase
#database.version=12
#database.name=db2
#database.version=7
```

Step 2: Migrate Data From 4.0 to 7.0

This section describes how to migrate your data to the 7.0 release. It includes information on the following subjects:

- Understand the Tool and Process
- Manually Migrate Any Databases You Have Changed
- Migrate Your Data Using the Migration Tool
- Drop 4.0 Tables When System Has Been Migrated Successfully

Understand the Tool and Process

This section provides a brief overview of the following topics:

Extent of Migration Tool Capabilities

The migration tool assists you in migrating your code and data from WebLogic Portal version 4.0 to 7.0. It is a helper utility—it migrates and annotates your code and data but does not fully complete the migration. Your team will need to perform a significant portion of the migration.

The migration tool is a stand-alone tool that does not need to interact with a running server.

Review Data Migration Process

The migration process performs a variety of tasks, including leaving some files unaltered, renaming files, creating new files, and so on. The database migration takes place on the 4.0 tables themselves; it does not create a new set of 7.0 tables without touching the 4.0 tables.

The affected tables are backed up and named `tablename_temp40`. You can drop these tables after the migration is complete and you have verified that your system is working correctly in release 7.0.

Review Data Migration Tool Tasks

This section describes the tasks that the tool completes. The order of tasks is not crucial, except for the last two. The server will not start if the Data Sync RDBMS task (in the following list) is not completed. The server will not function correctly until all tasks are completed.

You must migrate each E-Business Control Center project separately, that is, run the E-Business Control Center migration once for every project.

The following steps and descriptions are also displayed onscreen when you perform the data migration with the migration tool.

- **E-Business Control Center Data Migration** – Migrates E-Business Control Center project data one project at a time. When you click Execute, a window appears where you can specify the project to migrate. Go to the project source directory (the directory you specify should have a child directory called `application-sync`). Go to the output directory, where the migrated project will be written; this can be any directory as long as you are not overwriting existing data. If you know the Enterprise App Directory, enter it now. If you don't know it, you can enter it later using the E-Business Control Center. The migration tool will write the migrated project to this directory preserving the project name and structure.

- **Data Sync RDBMS Migration** – Clears the data from DATA_SYNC_ITEM, DATA_SYNC_APPLICATION, and DATA_SYNC_SCHEMA_URI and creates a trigger which removes unreferenced DATA_SYNC_APPLICATION, DATA_SYNC_SCHEMA_URI, and DATA_SYNC_VERSION rows. This step is repeatable.

Note: If you have modified your database tables, the following tasks might be affected; the previous two are not.

- **Portal RDBMS Migration** – Saves PORTAL_PAGE_P13N data in PORTAL_PAGE_P13N_TMP40 table. Drops foreign key constraints to PORTAL_PAGE_P13N and then drop the table. Creates the 7.0 version of PORTAL_PAGE_P13N and inserts saved data back to it. Adds foreign keys, checks constraints, and triggers referring to PORTAL_PAGE_P13N. This step is repeatable.

- **ENTITLEMENT_RULESET RDBMS Migration** – Creates ENTITLEMENT_RULESET_TEMP40 and inserts ENTITLEMENT_RULESET data into it, which is used as the data source for transforming the documents contained in the RULESET_DOCUMENT column. The ENTITLEMENT_RULESET RDBMS migration also must be run before Entitlement Ruleset Data Migration.

Warning: This step *not* repeatable after the next step, Entitlement Ruleset Data Migration, is run.

- **Entitlement Ruleset Data Migration** – Migrates entitlement ruleset data. The ENTITLEMENT_RULESET table is updated with the transformed RESULTSET_DOCUMENTs.

Manually Migrate Any Databases You Have Changed

If you changed the definition of any of your 4.0 database tables, you need to run those migrations manually.

Note: This can be a complex procedure. Study the migration tasks, their dependencies, and the SQL involved carefully before you begin. We recommend that you run the migration on a copy of your 4.0 data first.

Use the migration tool to complete migration tasks that are not affected by your modifications. Use an SQL tool to run the appropriate migration SQL statements affected by your modifications. Do the modifications in the order they are listed in "Review Data Migration Tool Tasks" on page 2-12.

Follow these steps to migrate your changed databases:

1. List all the database changes you have made.
2. Read the data migration descriptions and dependencies in the "Review Data Migration Tool Tasks" on page 2-12 to determine which tasks are affected by your database changes.
3. Locate the appropriate `.sql` file for your database, which is stored in the `<PORTAL_HOME>\migration` directory. The `.sql` files contain all the SQL you need to run the data migration steps except for terminate statements. Review this SQL and make any necessary changes required by the tool you will use to execute these scripts. These changes may include statement delimiters, backslashes, and a commit statement.
4. In your `.sql` file, locate the tasks you identified previously. Each set of SQL is preceded by a commented line and the task name, as shown. Part of the `oracle-817-v4_0to7_0.sql` file is shown in Listing 2-3.

Listing 2-3 Partial oracle-817-v4_0to7_0.sql File

```
-- ==============================

-- ENTITLEMENT_RULESET RDBMS Migration

CREATE TABLE ENTITLEMENT_RULESET_TEMP40 ( APPLICATION_NAME VARCHAR2(100) NOT
NULL, RULESET_URI VARCHAR2(254) NOT NULL, RULESET_DOCUMENT CLOB NULL,
CREATION_DATE DATE DEFAULT SYSDATE NOT NULL, MODIFIED_DATE DATE DEFAULT SYSDATE
NOT NULL)

INSERT INTO ENTITLEMENT_RULESET_TEMP40 ( APPLICATION_NAME, RULESET_URI,
RULESET_DOCUMENT, CREATION_DATE, MODIFIED_DATE ) SELECT APPLICATION_NAME,
RULESET_URI, RULESET_DOCUMENT, CREATION_DATE, MODIFIED_DATE FROM
ENTITLEMENT_RULESET
```

5. For each identified migration task, create new SQL statements and edit the SQL as appropriate for your database changes. Create one file of SQL statements for every migration task you need to modify.
6. Run the SQL statements on the database using the SQL tool of your choice, such as SQLPlus.

Migrate Your Data Using the Migration Tool

To complete data migration steps with the migration tool, follow the steps in this section.

Warning: Be sure you have edited the `migration_install.properties` file in the `PORTAL_HOME\migration` directory. If you have not entered the appropriate information, an error message appears when you attempt to run the migrator.

Use a database viewing/editing tool to check the data migration process as it progresses. After each task that affects databases, you can view your 4.0 database and the changes that took place.

1. Shut down WebLogic Portal Server.

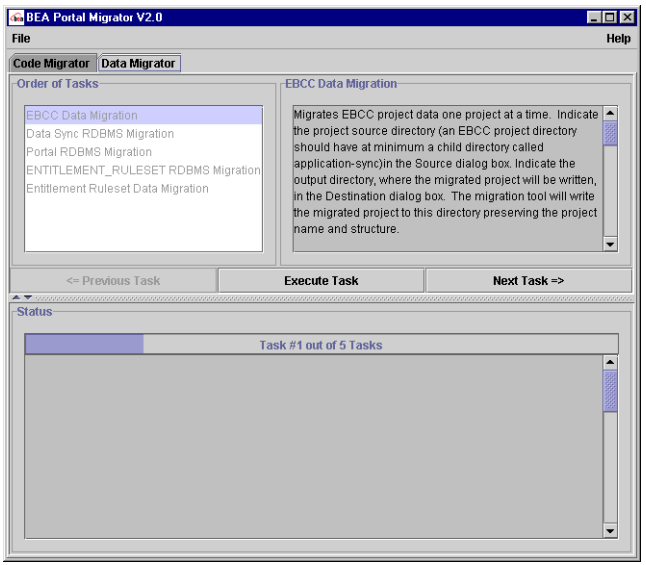
2. If you have not done so already, make *two* complete backups of your data, including both the databases and your properties files. Include the behavior tracking/events database, if it is being used.

3. Start the migration tool by double-clicking or using the command line to run the `migrator.bat` or `migrator.sh` file in the `<PORTAL_HOME>\migration\bin` directory.

Note: If you encounter problems, open the `migrator.bat` file and be sure all settings are correct.

4. Click the tab labeled Data Migrator, if it is not already on top, as shown in Figure 2-2.

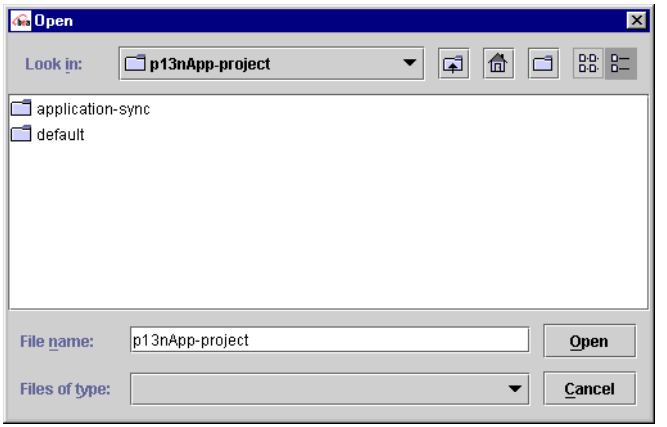
Figure 2-2 Data Migration Window



5. Select the first task in the list and click **Execute Task**.

6. For the first task only, E-Business Control Center Data Migration, another window appears. Enter the source for your first E-Business Control Center project, as shown in Figure 2-3. This should be the project, the parent directory of the `application-sync` directory.

Figure 2-3 Selecting the E-Business Control Center Project Source Directory



7. After the E-Business Control Center Project Migration window appears, enter the appropriate values for each of the three fields:

   - E-Business Control Center Project Source Directory: The project directory, such as `D:\BEA\Portal4.0\applications\p13nApp-project`.
   - E-Business Control Center Project Destination Directory: This can be any directory. Be sure not to overwrite existing files. An example is `D:\70_migration`.

Note: You can enter a project destination directory that does not yet exist and it will be created.

   - Enterprise App Root Directory: The enterprise application directory that this project works within. If you do not know yet know the directory, you can leave it blank and enter it later using the E-Business Control Center. For more information, see the E-Business Control Center online help.

   The project directories appears in your destination directory in the same structure, with the same directory names.

8. Examine the status messages that appear, which show the progress of the migration.

   Regarding messages in the data migration window Many of the messages are extremely detailed, and contain valuable information, sometimes indicating the directory where a migrated or new file was created. Tables

affected by the migration task are listed. At the bottom of each set of comments for the task, a message states whether the migration was successful.

The `migration.log` file contains more complete information. It contains all messages displayed in the window, as well as additional information such as the stack trace.

If any other message appears stating that there were problems migrating, use the error messages onscreen to address the problem, then run the task again. Some tasks require that you restore the database from a backup, then fix the problem and run the task again. If this is the case, the description in the window says so. If the problem stems from modifications you have made to the database, see "Manually Migrate Any Databases You Have Changed" on page 2-14.

9. Migrate all other E-Business Control Center projects.
10. Click **Next Task** and complete the other migration tasks.
11. When the task has completed successfully, click **Next Task** and continue until all tasks are completed. (If you have omitted tasks by editing `DataMigratorBundle.properties`, as described on "MigratorBundle.properties and DataMigratorBundle.properties Files" on page B-5, you can skip them in the list; otherwise, you must migrate every task in sequence.)

Manually Updating Events to the Latest XSLT

If you are not using events in WebLogic Portal 4.0, ignore this section.

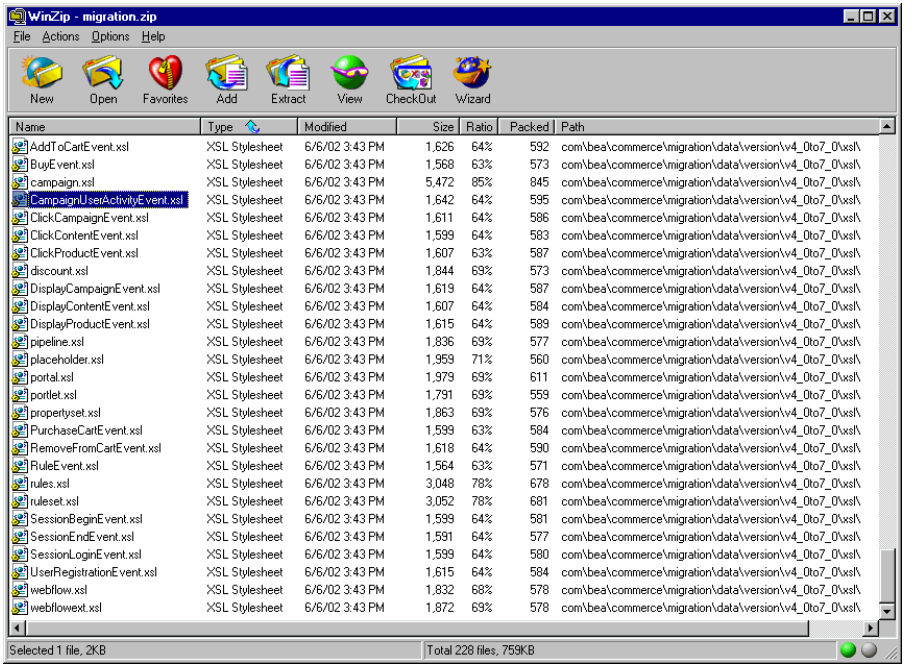
Events are actions taken by users, such as clicking a particular option in your site. You can choose to track events in a database, and analyze them with the tool of your choice, such as Broadbase. (WebLogic Portal does not provide analysis tools for events.)

We recommend that you simply create the new events database for the 7.0 format and start recording events in that database. Events are analyzed via third party tools so migration is not necessary.

Events in WebLogic Portal 7.0 use the most recent XSLT. If you want to use Xylon or a similar tool to update your WebLogic Portal 4.0 events to the latest format, you can use the provided `.XSL` files.

The files are in the `migration.jar` file, shown in Figure 2-4. Only files that are named `*Event.xsl` are relevant. Use Xylon or another tool of your choice and the `*Event.xsl` files to apply the new XSLT to your 4.0 events.

Figure 2-4 XSL Files for Event Updating



Drop 4.0 Tables When System Has Been Migrated Successfully

The migration process does not drop the backup tables it creates (`tablename_temp40`). Drop these 4.0 tables after you have successfully run your site for a satisfactory amount of time.

Step 3: Migrate Code From 4.0 to 7.0

Use the information in this section to migrate the code. The information in this section is organized as follows:

- Review Code Migration Process
- Use the Migration Viewer
- Migrate Code
- Migrate JavaServer Pages

Note: To speed up the migration process, multiple people can run the migration simultaneously. For instance, if you use a source code control system, one developer could check out and migrate each of the main subdirectories within your main release 4.0 directory.

Warning: To prevent one person's changes overwriting another's, be sure that no one is migrating a subdirectory of code that another person is migrating.

Review Code Migration Process

The code migrator analyzes `.java` files and either makes the appropriate change or flags the necessary portions for change, without updating the code. No matter what the change, a comment is added to the file. The code migrator assists you in the migration process, but you must still review all code files and make any necessary changes before the migration is complete. The changes and comments are put in a new copy of the files, created in a location you specify before beginning the migration.

The amount of code the tool can migrate varies drastically from one implementation to another.

You can migrate both standard `.java` files and JSPs by precompiling the JSPs; running the code migration tool; and making the appropriate updates to JSPs, tags, and any other dependent code.

The code migration tool focuses on the following:

- Import statements (packages and fully qualified classes)
- Variable declarations
- Object instantiations
- Variable references
- Method calls
- Field references
- Castings
- Instance of statements
- `extends` and `implements` statements
- `return` statements

Use the Migration Viewer

The changes to code in release 7.0 are described in the `migrinfo.html` file, included in the `<PORTAL_HOME>\migration\doc` directory. To search and sort the information, use the Migration Viewer tool. For more information, see "Viewing Code Changes Using the Migration Viewer Tool" on page A-1.

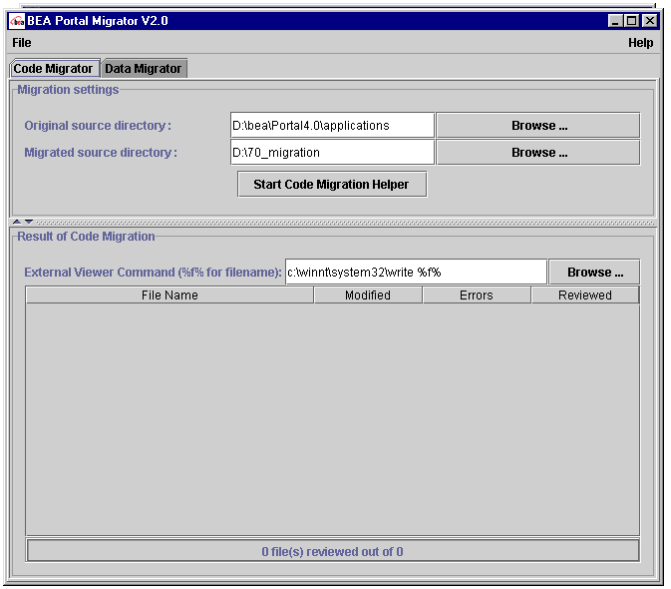
This is extremely valuable information, providing details on what classes, packages, methods, and other API elements have changed in this release. Use the Migration Viewer concurrently with the migration tool as you migrate code and JSPs. As you review each migrated file and the migration comments, use the Viewer to get more information on how you need to accommodate the changes to the API.

Migrate Code

This section contains the steps for using the utility to migrate your code to WebLogic Portal 7.0.

1. If the migration utility is not already running, start it by double-clicking the `migrator.bat` or `migrator.sh` file at `\migration\bin`.
2. Select the **Code Migrator** tab, if it is not already on top, as shown in Figure 2-5.

Figure 2-5 Code Migrator Tab

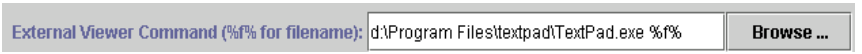


3. Specify a source directory and a destination directory.

Warning: If more than one person is migrating code, be sure that no one is migrating a subdirectory that another person is already migrating.

4. Enter the path to the executable file for the editor that you want to use to view and edit code files, as shown in Figure 2-6. After the migration, you can open any of your code files by selecting a file in the list at the bottom of the code migrator window.

Figure 2-6 Entering Path to External Viewer



Note: You must retain the space and `%f%` at the end of the path; this is the placeholder for the code file that you are editing.

5. When the migration has run, the migrated files are listed in the bottom half of the window. You can open the files in your preferred editor by double-clicking them in the window.

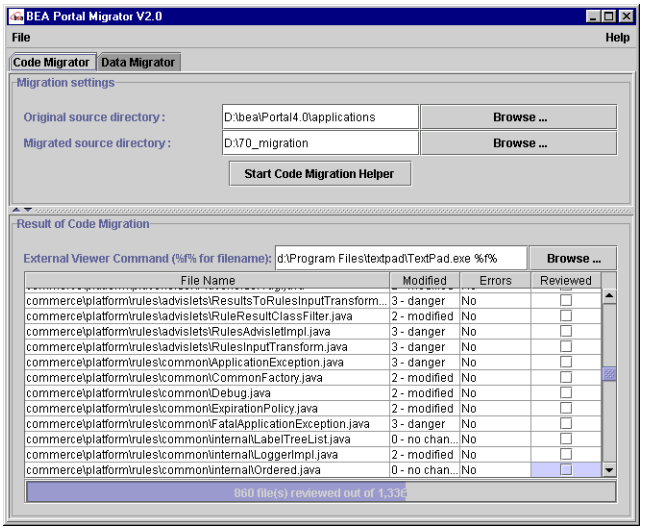
6. The migration process migrates all the code in the source directory in one process; you do not need to migrate one file at a time. Status messages and the migrated file names appear during the process.

   When you are ready to migrate the entire source directory, click **Start Code Migration Helper**.

   If errors occur, refer to the `migration.log` file in the `PORTAL_HOME\migration` directory, the status messages displayed onscreen, and the migration notes added to the migrated code files.

7. When the migration process is complete, the files are displayed with information about what issues the migration utility encountered, whether the file was changed, and so on. Figure 2-7 shows an example.

Figure 2-7 List of Migrated Files and Status



The effect of the migration on each file is indicated by the level 0 through 3, with 3 being most in need of review and additional changes The following list describes each level:

- **0 - no change:** No changes was made to the file. These files are marked as reviewed in the window.
- **1 - comments:** A migration note was added; review the file to determine what changes to make.

- **2 - modified:** The file was modified; review the file to determine whether the change was correct and to view the migration note.
- **3 - danger:** A significant change needs to be made to the file; for instance, it might contain code extending a class that no longer exists.

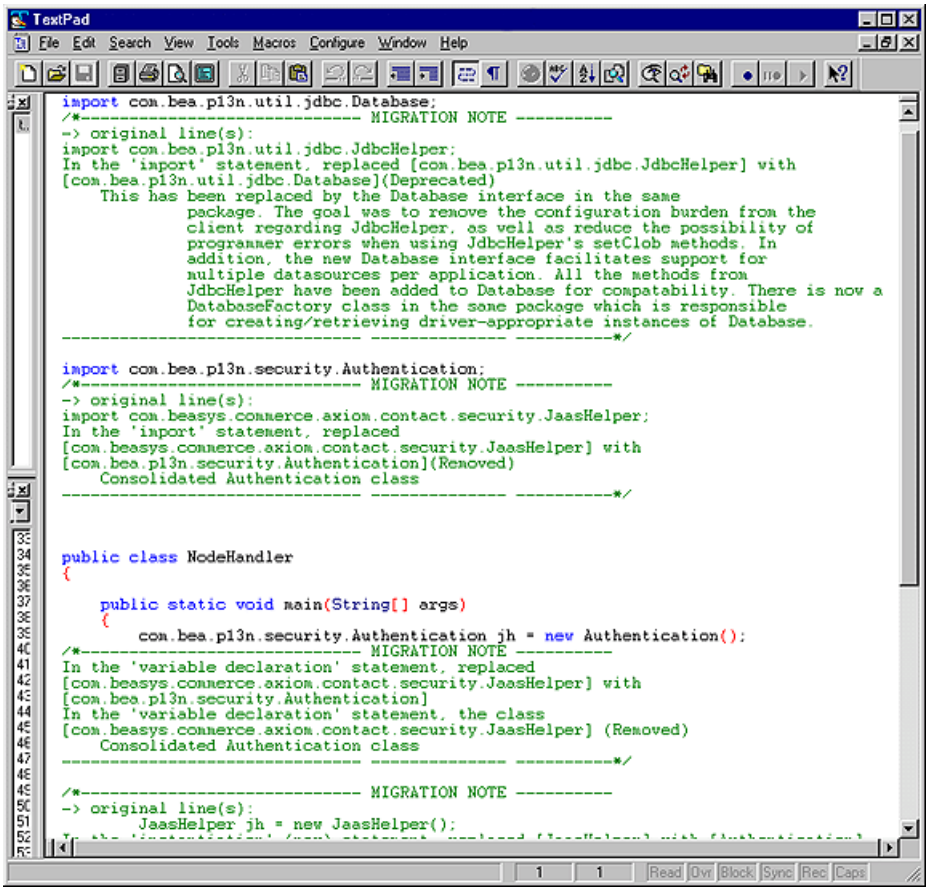
Note: Errors occur on empty statements (often preceded by static initialization blocks). For details, see the `migration.log` file. The log clearly shows the sources of problems caused in the migration. If you see that errors were caused by an empty statement, remove the empty statement and run the code migration again.

8. In your destination directory, review each of the migrated files to see what notes were added and make additional changes to the files according to the migration notes.

Note: Use the Migration Viewer to help you determine what changes you need to make and how to make them based on how files were migrated and the generated comments in each migrated file. For instructions on using the Migration Viewer, see "Viewing Code Changes Using the Migration Viewer Tool" on page A-1.

An example of a migrated file and migration notes is shown in Figure 2-8.

Figure 2-8 Migrated Code File With Notes



9. After you have completely finished with each file, mark the **Reviewed** box in the migrator utility window.

Migrate JavaServer Pages

This section covers the changes to JavaServer Pages (JSPs) in release 7.0 and what you need to accommodate those changes.

Note: After migration is complete, you need to consider how to make changes to JSPs based on the information in "Configure Webflows and JSPs" on page 2-31.

1. Create `.java` files for each of your JSPs. (For two ways to do this, see "Generating .java Files for JSPs" on page 2-26.)

   The `.java` files are created in a path starting at your WEB-INF directory for the application. For instance, if the JSPs are at `mywebapp\myJSPs\`*`JSPfilenames`*`.JSP`, the generated `.java` files are created at `WEB-INF\_tmp*\jsp_compiled\_myJSPs\*`*`JSPfilename`*`.java`.

2. To run the code migrator on the generated `.java` files, complete the steps in the previous section, "Migrate Code" on page 2-21. For a destination directory, select a location that is not in your `PORTAL_HOME` directory.

   Before you can begin reviewing and making changes to the JSPs, move the JSPs the appropriate directory in the `PORTAL_HOME` directory. They need to be in the 7.0 environment for you to verify whether the changes were effective. Typically this should be `<PORTAL_HOME>\applications\wlcsApp-project\`*`webappdir`*`\`.

3. Review each migrated generated `.java` file and the comments and changes in each. The relevant line number in the JSP is referenced in the migration note in the `.java` file and the corresponding JSP filename and path is added to the top of each `.java` file.

4. Make the necessary changes in your JSPs.

Repeat these steps until no more conversions or warnings show up in the code migrator.

Generating .java Files for JSPs

You can do this in a number of ways:

- Add the following to your `web.xml` file, then open each JSP:

```
<!-- JSP configuration -->
  <jsp-descriptor>
    <jsp-param>
      <param-name>keepgenerated</param-name>
      <param-value>true</param-value>
```

```
 </jsp-param>
</jsp-descriptor>
```

- **Quicker:** You can use `jspcPrepare.bat` to do this, which is included in the `<PORTAL_HOME>\migration\bin` directory. The file is shown in this directory. You can either edit the appropriate paths in the file and then run `jspsPrepare.bat`, or run each command separately from the command line, supplying the appropriate values.

Step 4: Migrate Other WebLogic Products

If you are using prior versions of WebLogic Server, WebLogic Integration, or the other WebLogic products, you should migrate them now. Refer to the migration documentation for migration to 7.0, at http://edocs.bea.com/platform/docs70/index.html.

View WebLogic Platform 7.0 Documentation

The WebLogic Platform 7.0 documentation contains important, useful information about the WebLogic Server, WebLogic Integration, and WebLogic Workshop, including the new directory structure organized by domain. For more information, see http://edocs.bea.com/platform/docs70/index.html.

Step 5: Assemble Migrated Files and Perform Additional Configuration Tasks

After all files are migrated, you need to organize and configure them to work in the WebLogic Portal 7.0 environment, as shown in the following list:

1. Ensure All Files and Databases Are Migrated

2. Create a New 7.0 Project Directory
3. Move Files to the New Domain
4. Configure Webflows and JSPs
5. Set Database Connection Variable Values in the Start Script for Each Domain
6. Set the Value of the mem_args Variable
7. Change How Start Scripts Call WebLogic Server
8. Edit the config.xml File to Set Two-Phase MBean Deployment
9. Add Database Information to config.xml
10. Update Any Hard-Coded Paths in Configuration Files
11. Add References to EJBs to Configuration Files
12. Add weblogic-application.xml File to All Applications
13. Copy New JAR Files to Migrated Web Application Directory Structure
14. Follow Instructions in WebLogic Server Upgrade (Migration) Guide
15. Complete a Build of Your Java Source and EJBs
16. Configure Classpath to Use Xerces
17. Check for Additional Migration Information

Ensure All Files and Databases Are Migrated

Before you complete the tasks in this section, be sure all necessary migration has been completed.

- Be sure you have completed all the migration tasks for code migration and data migration in "Step 1: Prepare Environment and Files for Migration" on page 2-2 through "Step 4: Migrate Other WebLogic Products" on page 2-27.
- Be sure you have addressed all issues referred to in "Address Issues Related to Migration" on page 2-4.

- If you did not use the Migration Viewer while migrating code, use it now to read about all API changes in release 7.0, and address the changes as necessary in your implementation. For more information, see "Viewing Code Changes Using the Migration Viewer Tool" on page A-1.
- Be sure you have reviewed the WebLogic Platform 7.0 documentation, in particular information regarding migration and what's new in this release.

Create a New 7.0 Project Directory

The directory structure for release 7.0 is significantly different, so create a new build environment, as described in the following steps:

1. Create a new project directory, including all your build scripts.
2. In that project directory, run the Configuration Wizard to create a new domain in the project. Make sure that it is a WebLogic Portal portal domain; you can make several types of domains using the wizard.

   For more information on the Configuration Wizard, see *Using the Configuration Wizard* at http://edocs.bea.com/platform/docs70/confgwiz/index.html.

Note: You *must* create a new domain instead of rearranging your existing directories and files. Web application tools for tasks such as creating users and syncing have changed significantly, and 4.0 versions do not work in release 7.0.

Move Files to the New Domain

1. Copy all migrated Java source files to the appropriate directories in your new project.
2. Copy the webapp that you are migrating to the enterprise app that you created with the Configuration Wizard. For example, assume the Configuration Wizard created a directory `DOMAIN_DIR/beaApps/portalApp`. Copy your Web application directory into the `portalApp` directory.
3. Manually copy the E-Business Control Center project files for your portal Web application only from the migrated E-Business Control Center data file directory into the E-Business Control Center project directory created by the Configuration

Wizard. That directory is `DOMAIN_DIR/beaApps/portalApp-project`. Copy only the files for your Web application, not other Web applications such as the `tools` web application, which will only exist if your Web application is a portal. The following directories are among those we recommend that you copy some or all files from:

```
application-sync\pipelines
application-sync\entitlements
application-sync\entitlements\GlobalEntitlements
application-sync\Portlets
application-sync\userprofiles
application-sync\webapps\your_web_app
```

For more information on the directory structure, review the *WebLogic Server Administration Guide*, at http://edocs.bea.com/wls/docs70/adminguide/overview.html. To learn more about domains, see the *Creating and Configuring WebLogic Server Domains* guide http://edocs.bea.com/wls/docs70/admin_domain/index.html.

The gist of the change in directory structure is illustrated in Figure 2-9 and Figure 2-10. Reassemble the migrated files into the appropriate directories.

Figure 2-9 WebLogic Portal Applications in the WebLogic Portal 4.0 Directory Structure

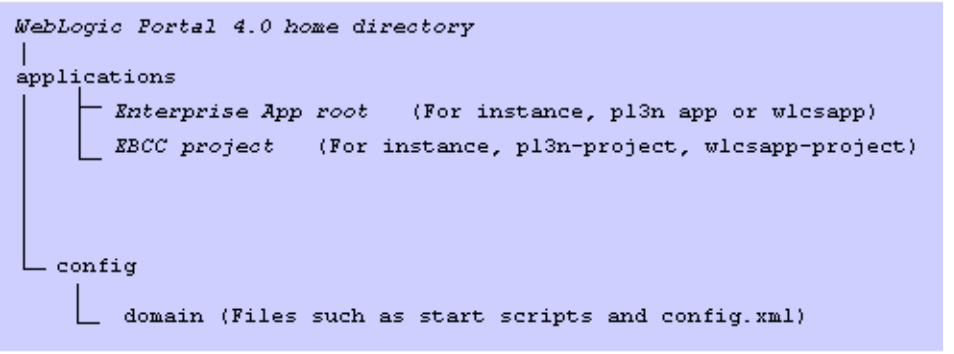
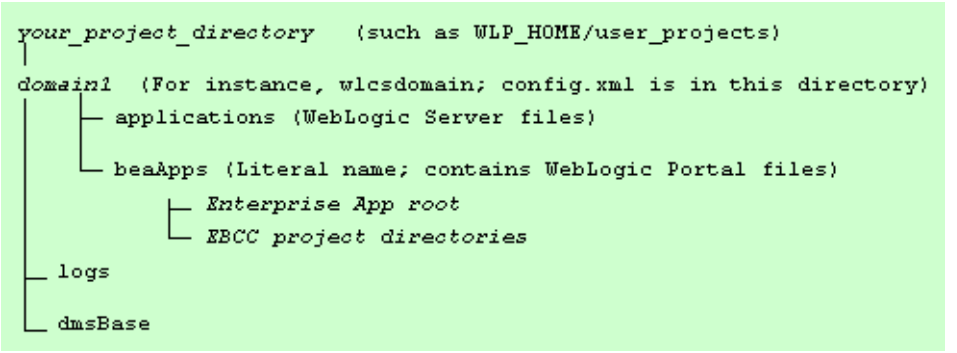


Figure 2-10 WebLogic Portal Applications in the WebLogic Platform 7.0 Directory Structure



Configure Webflows and JSPs

The information in this section helps you to incorporate the latest WebLogic Portal 7.0 portal functionality in your migrated portal while maintaining compatibility with any changes you may have made to the standard portal webflows or JSPs. This is a manual process whose difficulty varies depending on these changes.

Use the instructions in the following sections to make the appropriate configurations.

- Determining Whether You Need to Perform These Steps
- Before You Begin
- Configuring Webflow Files
- Configuring JSP files

Determining Whether You Need to Perform These Steps

If you did not use a portal in your Web application, skip this section. If you do not know if there is a portal in your webapp, look for the following files in your `application-sync/webapps/yourWebAppName` directory:

- `portal.wf`
- `security.wf`

- tools.wf
- user_account.wf

If these files do not exist, you can skip this section. If you have these files, continue to the next section.

Before You Begin

Required base version of WebLogic Portal These instructions apply to migrating from WebLogic Portal 4.0 Service Pack 2 only. If you are using Service Pack 1 or earlier, you *must* upgrade to SP2 first. There are differences between SP2 and previous versions that affect these instructions.

In this section, <SAMPLE_DIR> is used to refer to the directory <BEA_HOME>/weblogic700/samples/portal/sampleportalDomain/beaApps

Configuring Webflow Files

Portal uses three webflow files that must be updated to work with WebLogic Portal 7.0. They are security, tools, and user_account. (The portal webflow is also used by Portal Web applications, but there were no changes to it from release 4.0 to 7.0).

Each webflow is represented by a file of the same name with the .wf extension.

If you have not made changes You do not have to do anything; simply use the new versions in the new domain and Web application you created.

If you have made changes You have two potential paths to take for the changed files; it is up to you to decide what is easiest, based on the extent of your changes.

- You can apply your changes as needed to the new files. This is the recommended strategy.
- Alternately, you can manually apply the new 7.0 changes to your files, then copy them to the new Web application you created. If you choose this path, consult the "Portal Webflow Changes" on page B-19, which has a list of the changes needed to make for each of the files.

Note: You can find these new files in the following directory:

<SAMPLE_DIR>\beaApps\sampleportal-project\application-sync\webapps\sampleportal

Configuring JSP files

Some of the JSP files that make up the portal framework have changed from WebLogic Portal 4.0 to WebLogic Portal 7.0. These changes must be applied to your Web application for it to function properly. There are also some new files that you must add to your migrated Web application.

- New Directories
- New Files
- Modified Files

New Directories

Two new directories are now in your Web application, in the new domain you created:

```
<WEBAPP>\registration
<WEBAPP>\util
```

New Files

These are the new files; you do not need to do anything to them, just note that they are there.

```
<WEBAPP>\framework\tools\change_name.jsp
<WEBAPP>\framework\tools\change_name.properties
```

Modified Files

The following list shows that files that have been updated for WebLogic Portal 7.0. You can:

- Use these files in place of your own, in which case do not copy your own files to the new domain and Web application
- Reconcile these files with your own, if you have changed your own version of these files

If you need to reconcile the files, examine each one before you copy it to determine if any changes need to be preserved. If you have made any changes to a file, it is up to you to resolve the differences with the 7.0 version of the files. Decide if it is easier to re-apply your changes to the new 7.0 files or make the 7.0 changes to your existing files and then make the changes.

```
<WEBAPP>\framework\edit_titlebar.inc
<WEBAPP>\framework\header_links.inc
<WEBAPP>\framework\hnav_bar.jsp
<WEBAPP>\framework\maximize_titlebar.inc
<WEBAPP>\framework\minimize_titlebar.inc
<WEBAPP>\framework\normal_titlebar.inc
<WEBAPP>\framework\resourceURL.inc
<WEBAPP>\framework\titlebar.jsp
<WEBAPP>\framework\vnav_bar.jsp
<WEBAPP>\framework\security\badlogin.properties
<WEBAPP>\framework\security\login.properties
<WEBAPP>\framework\tools\portal_prefs.jsp
<WEBAPP>\framework\tools\portal_prefs.properties
<WEBAPP>\framework\tools\select_portal_pages.jsp
<WEBAPP>\framework\tools\select_portal_pages.properties
<WEBAPP>\framework\tools\select_portlets.jsp
<WEBAPP>\framework\tools\select_portlets.properties
<WEBAPP>\framework\tools\select_skins.properties
```

Note: If you have problems opening a migrated project, check the `.eaprj` file. If the "EnterpriseAppRoot" element has an uppercase E at the beginning, change it to a lowercase e.

Set Database Connection Variable Values in the Start Script for Each Domain

Database information, including the database connection information, is specific to each domain. WebLogic Platform 7.0 has changed so that the database connection information is defined in the start script and assigned to a variable, which is used by `set-environment.bat` or `set-environment.sh`.

Therefore, for each start script you have, such as `start-portal.bat`, where you call WebLogic Server and other items, set the definitions for variables called in `set-environment`. Do this by editing the database connection values in the `dbsettings.properties` file, which is created by the Domain Wizard, as follows:

1. Uncomment the information for the database you are using.
2. Enter the correct values for the database connection variables (enter the right values, instead of `@variablename@`).

Set the Value of the mem_args Variable

The `mem_args` variable is new in this release and appears in all start scripts. You must assign a value to this variable or your migrated system will not run and the server will appear to hang. This variable tells WebLogic Server how to handle memory.

1. Assign the appropriate amount of memory. The following example shows how to set the memory to 128 MB of RAM.

   ```
   MEM_ARGS="-Xms128m -Xmx128m -XX:MaxPermSize=128m"
   ```

2. Choose a value appropriate for your application. If your server locks up or crashes with a Hotspot VM error, you might need to increase this number. This is due to an issue with the Hotspot VM.

Change How Start Scripts Call WebLogic Server

Formerly, each start script made a Java call to WebLogic Server directly. In the WebLogic Platform 7.0 release, this has changed. Each start script must call the server as follows:

```
call %WLP_HOME%\bin\win32\startWebLogic.cmd
```

Replace your start scripts' server calls with the above line. The `WLP_HOME` variable's value is set when you install WebLogic Portal 7.0.

Edit the config.xml File to Set Two-Phase MBean Deployment

A new feature in this release is two-phase Mbean deployment. This is covered in the WebLogic Server documentation for this release (http://edocs.bea.com/wls/docs70/index.html.

You must set up WebLogic Portal to accommodate the feature by editing the `config.xml` file.

Add the following, shown in Listing 2-4, to the application tag in your `config.xml` file.

Note: You can set the value to true or false, but the attribute must be there for every application.

Listing 2-4 Attribute to Add to config.xml

```
<Application
  Deployed="true" TwoPhase="false"
  Name="p13nConsoleApp"
  Path="D:/bea/weblogic700/samples/portal/p13nDomain/beaApps/p13nConsoleApp">
  <WebAppComponent
    Name="p13nConsole"
    ServletReloadCheckSecs="300"
    Targets="p13nServer"
    URI="p13nConsole"
```

```
  />
</Application>
```

Add Database Information to config.xml

Modify `DOMAIN_DIR/config.xml` to use the appropriate database driver. For information about doing so, see the *Administration Guide* at http://edocs.bea.com/wlp/docs70/index.htm.

If you need any database connection pools or data sources besides the default commercePool, add them to `DOMAIN_DIR/config.xml`. Make sure that you add an appropriate line to `DOMAIN_DIR/fileRealm.properties` for each pool that you create. For example, if you add a connection pool named "oraclePool", add this line to `fileRealm.properties as follows`:

```
acl.reserve.weblogic.jdbc.connectionPool.oraclePool=everyone
```

Update Any Hard-Coded Paths in Configuration Files

Search all of the following files and update the paths so that they are correct for the new directory structure:

- `config.xml`
- `application.xml`
- `application-config.xml`
- `weblogic-application.xml` (New in this release)

Add References to EJBs to Configuration Files

Add entries for your EJBs, if any, and your Web application(s) to the following:

```
<DOMAIN_DIR>/beaApps/portalApp/application.xml
<DOMAIN_DIR>/config.xml
```

Add weblogic-application.xml File to All Applications

The new `weblogic-application.xml` file lets you tune WebLogic Server features for use with WebLogic Portal applications.

The file *must* be present for your application to run. Copy it to the META-INF directory for every application. For example:

```
weblogic700\samples\portal\p13nDomain\beaApps\p13nApp\META-INF
```

To use the file for tuning, see the WebLogic Server documentation at http://edocs.bea.com/wls/docs70/index.html.

Copy New JAR Files to Migrated Web Application Directory Structure

Delete all JAR files that are not specific to your application from the `<migrated_webapp>/WEB-INF/lib` directory, and replace them with all of the JAR files from this directory:

```
<WLS_HOME>/weblogic700/samples/portal/sampleportalDomain/beaApps/sampleportal/sampleportal/WEB-INF/lib
```

Follow Instructions in WebLogic Server Upgrade (Migration) Guide

Follow the WebLogic Server upgrade instructions at http://e-docs.bea.com/wls/docs70/upgrade/index.html. Pay particular attention to changes to EJB-QL in the descriptors for CMP entity beans.

Complete a Build of Your Java Source and EJBs

Complete a build of your java source and EJBs. Remember, you must use your build scripts to perform the build.

Configure Classpath to Use Xerces

If your source uses the Xerces parser, add the JAR file `<WLS_HOME>/weblogic700/lib/xerces.jar` to the startup classpath for your server.

Check for Additional Migration Information

Any additional migration information of a similar nature is posted on the Customer Support (http://websupport.bea.com/welcome.jsp) and BEA dev2dev (http://developer.bea.com/index.jsp) Web sites.

Step 6: Verify the Migration From 4.0 to 7.0

Start the server. Validate the application using the 'Validate' button in the E-Business Control Center. Perform a datasync either from the E-Business Control Center, or using the sync.bat file in the EBCC project directory. At this point, the portal Web application should now be properly synced, but some data must be migrated using the following manual DB task:

1. In a sql tool for your DB, perform the following command:

   ```
   SELECT PORTAL_XML_REF From PORTAL;
   ```

   The result will be returned in the following format:

   ```
   <enterpriseapp>/<webapp> (i.e. portal/stockportal)
   ```

2. Edit this to reflect your enterprise app name, which by default was set to portalApp by the domain wizard tool.

3. For example:

   ```
   UPDATE PORTAL SET PORTAL_XML_REF='portalApp/stockportal' where
   PORTAL_XML_REF='portal/stockportal';
   ```

If Entitlement Rules Exist

If the ENTITLEMENT_RULESET table in WebLogic 4.0 is not empty at migration, a problem may arise because of a difference in the schema used to evaluate entitlements. If this problem arises, the following error will appear in the console:

Listing 2-5 ENTITLEMENT_RULESET Error

```
Caused by: org.xml.sax.SAXParseException: General Schema Error:
Grammar with uri:
http://www.bea.com/servers/p13n/xsd/rules/corerules/2.1 , can not
be found; schema namespace may be wrong:  Xerces supports schemas
from the "http://www.w3.org/2001/XMLSchema"; namespace or the
instance document's namespace may not match the targetNamespace of
the schema.
```

This problem can be resolved in three ways:

- Use the migrator tool to update and move the data from the 4.0 Database schema.
- Delete the contents of the ENTITLEMENT_RULESET table, and then use the WebLogic Administration Portal to re-create the entitlements.
- Extract the Entitlements from the ENTITLEMENT_RULESET table and update them to reference the correct DTD.

Step 7: Next Steps

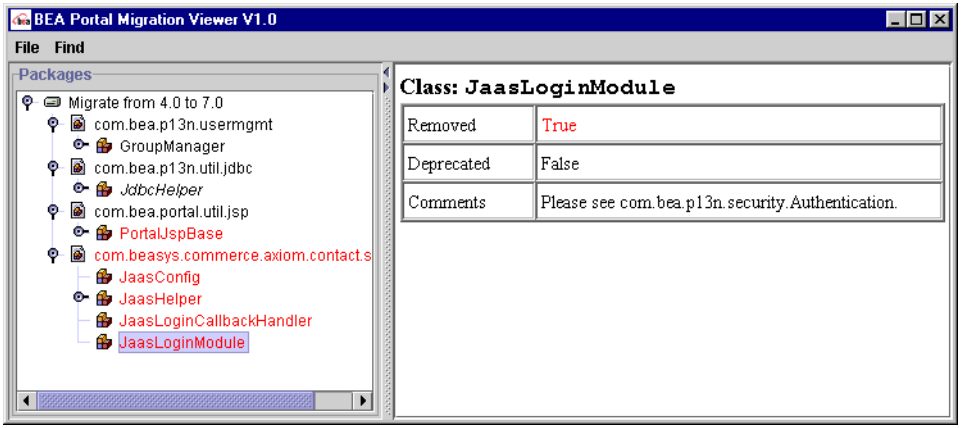
Now that you have successfully completed migration, complete the next steps for continuing migration and using WebLogic Portal 7.0.

Use the online documentation site to locate the documentation you need for additional learning at http://edocs.bea.com/wlp/docs70/index.html.

A Viewing Code Changes Using the Migration Viewer Tool

The Migration Viewer provide with this release provides quick, easy access to information about API changes in WebLogic Portal 7.0. With the viewer, you can easily see how many of the changes affect your implementation, and how much work will be involved in your migration efforts.The viewer is shown in Figure A-1.

Figure A-1 Migration Viewer Tool



Note: This information contained in the Migration Viewer is also provided in an HTML file (`migrinfo.html`).

The Migration Viewer includes the following key features:

- A GUI environment for easier scanning of changes.
- Visual cues about types of changes (italics for deprecation, red for removal).
- Icons indicating packages, classes, fields, and methods.
- A find feature allowing you to search for a keyword in packages, classes, fields, and methods; in names and in comments.

Migration Viewer Files

The zip file contains the following files:

- `<PORTAL_HOME>/migration/lib/migration_viewer.jar`: the required class files and images
- `<PORTAL_HOME>/migration/bin/viewer.bat`: a Windows batch script for starting the viewer
- `<PORTAL_HOME>/migration/bin/viewer.sh`: a UNIX sh script for starting the viewer

Using the Migration Viewer

You can perform two main types of tasks in the viewer:

- Use the main GUI window to view information about the listed packages, classes, fields, and methods
- Use the Find window to search for specific keywords

Viewing Information in the Main Window

To view information in the Main Window, take the following steps:

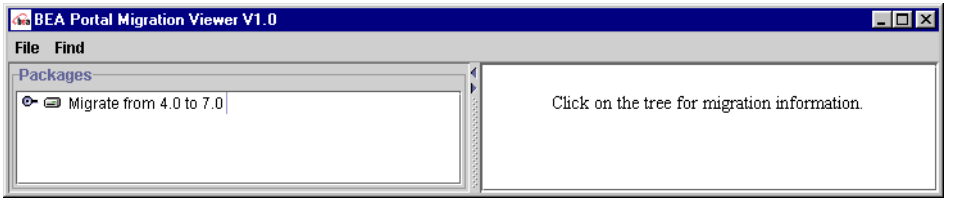
1. Start the Migration viewer, as follows:

   Windows: `<PORTAL_HOME>\migration\bin\viewer.bat`

   UNIX: `<PORTAL_HOME>/migration/bin/viewer.sh` (UNIX) file.

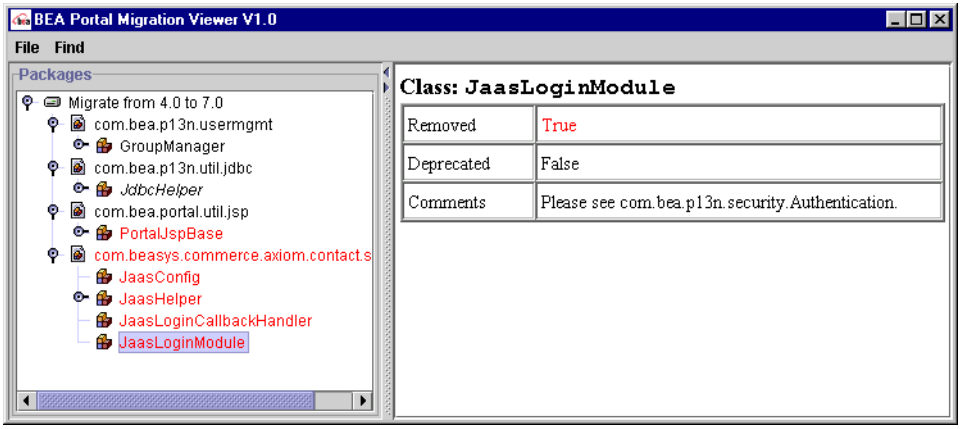
   The Viewer should appear as shown in Figure A-2:

Figure A-2 Migration Viewer Tool—Initial View



2. In the left panel, click the **Key** icon next to "Migrate from 4.0 to 7.0". The packages for this migration appear in the left panel, and information about the selected item appears in the right panel.
3. Expand the icons and select the packages, classes, fields, and methods you want information about, as shown in Figure A-3.

Figure A-3 Migration Viewer Tool—Subsequent View After Selecting



Legend for cues in the left panel:

- Red text indicates a removed item.
- Italicized text indicates a deprecated item.
- A key icon indicates a package.
- A boxes icon indicates a class.
- An **M** icon indicates a method.
- An **f** icon indicates a field.

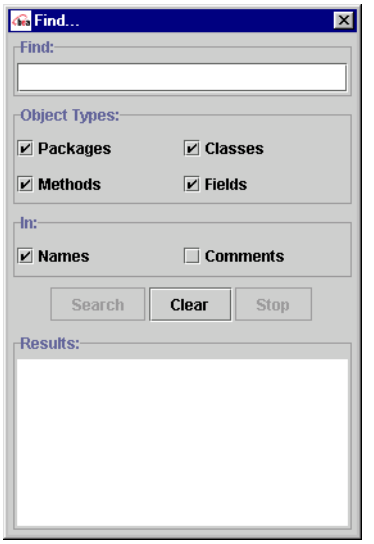
Searching by Keyword in the Find Window

If you want to find a particular keyword anywhere in the API change descriptions, use the Find function.

1. From the Migration Viewer Find menu, choose **Find**.

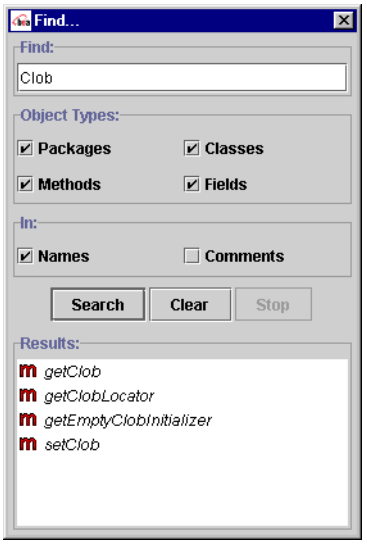
   The Find window appears, as shown in Figure A-4.

Figure A-4 Migration Viewer Tool—Finding, Step 1



2. Type the keyword in the **Find** field.
3. Unmark any types or locations where you do not want to search for the keyword.
4. Click **Search**; the results will appear in the Results window, as shown in Listing A-5.

Figure A-5 Migration Viewer Tool—Finding, Step 2



5. Double-click any item in the Results window; the item is displayed in the main Migration Viewer window.

B Migration Files

The migration parent directory contains programs and support files for the migration process. This section also contains information on the E-Business Control Center project migration.

migration_install.properties File

The `migration_install.properties` file in the migration root directory contains information you need to specify for the migration tool, such as your database and database connection data. Listing B-1

Listing B-1 MigratorInstall.properties

```
######################################################################
## PROPERTIES TO BE SET BY THE USER AT INSTALL TIME
######################################################################
# ------------------------------------------------------------
##       When you have set the properties you need, set the
#       following flag to 'true' so the migrator tool will start.
#       Otherwise, the Migrator will assume you have NOT set
#       any property and will refuse to run.
## ------------------------------------------------------------
start_migrator=false
# ------------------------------------------------------------
##Database Properties
## ------------------------------------------------------------
# Database connection properties
#------Oracle Thin Driver----------------------#
#
# For oracle, replace the following:
#    @USER@, @PASSWORD@, @SERVER@, @PORT@, and @SID@
#    e.g. jdbc:oracle:thin:@localhost:1521:ORCL
#
database.connection.driver = oracle.jdbc.driver.OracleDriver
database.connection.url = jdbc:oracle:thin:@@SERVER@:@PORT@:@SID@
database.connection.props = user=@USER@;password=@PASSWORD@
#------MS SQL Server ----------------------#
#
# For SQL Server, replace the following:
#    @SERVERPORTNUMBER@, @USER@, @PASSWORD@, and @SERVER@ (in two different
locations)
#
#database.connection.driver = weblogic.jdbc.mssqlserver4.Driver
#database.connection.url =
jdbc:weblogic:mssqlserver4:@SERVER@:@SERVERPORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
#------Sybase jConnect 5.2 ----------------------#
#
# For Sybase, replace the following:
#    @SERVER@, @PORTNUMBER@, @USER@, and @PASSWORD@
#
#database.connection.driver=com.sybase.jdbc2.jdbc.SybDriver
#database.connection.url=jdbc:sybase:Tds:@SERVER@:@PORTNUMBER@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
```

```
orConnectionSecs=999999999999
#------IBMS's DB2 ---------------------------------------#
#
# For DB2, replace the following:
#    @DB2_DATABASE@, @USER@, and @PASSWORD@
#
#database.connection.driver=COM.ibm.db2.jdbc.app.DB2Driver
#database.connection.url=jdbc:db2:@DB2_DATABASE@
#database.connection.props =
user=@USER@;password=@PASSWORD@;server=@SERVER@;weblogic.t3.waitForConnection=t
rue;weblogic.t3.waitSecondsForConnection=999999999999;weblogic.jts.waitSecondsF
orConnectionSecs=999999999999
# Database and version
# for Oracle users use database.version=817 for either Oracle 8.1.7 or 9i
database.name=oracle
database.version=817
#database.name=sql_server
#database.version=2000
#database.version=7
#database.name=sybase
#database.version=12
#database.name=db2
#database.version=7
# This is used to continue a task even if required statements fail.
# Use it to get a longer list of failed actions
database.continue.despite.failure=false
```

migrator.bat and migrator.sh File

The `migrator.bat` (Windows) and `migrator.sh` (UNIX) start the migration tool.

Listing B-2 shows the `migrator.bat` file.

Listing B-2 Example migrator.bat File

```
echo off
REM -------------------------------------------------------#
REM  Migrator Starter Script on Windows                     #
REM -------------------------------------------------------#
SETLOCAL
```

```
set DATABASE=ORACLE_THIN
REM set DATABASE=MSSQL
REM set DATABASE=SYBASE_JCONNECT
REM set DATABASE=DB2_TYPE2
if "%DATABASE%" == "" ( echo "The DATABASE variable must be uncommented in
migrator.bat"
    exit 1 )
CALL ..\..\bin\win32\set-environment.bat
REM ------------------------------------------------------#
REM         VARIABLES TO SET
REM ------------------------------------------------------#
REM Due to database specifics, you may have to set your
REM path to include dll
REM ------------------------------------------------------#
REM          The mini script
REM ------------------------------------------------------#
set MIGRATION_DIR=%WL_COMMERCE_HOME%\migration
set MIGRATION_LIB=%MIGRATION_DIR%\lib
set
MIG_CLASSPATH=%BEA_HOME%\lib\tools.jar;%MIGRATION_LIB%\migration.jar;%MIGRATION
_LIB%\apache\xerces-1_4_3\xerces.jar;%MIGRATION_LIB%\apache\xalan-j_2_0_1\xalan
.jar;%MIGRATION_LIB%\p13n_system.jar;%WEBLOGIC_HOME%\lib\weblogic.jar;%BEA_HOME
%
REM ------------------------------------------------------#
REM For testing the setup only
REM ------------------------------------------------------#
REM echo BEA_HOME=%BEA_HOME%
REM echo WEBLOGIC_HOME=%WEBLOGIC_HOME%
REM echo WL_COMMERCE_HOME=%WL_COMMERCE_HOME%
REM echo MIGRATION_DIR=%MIGRATION_DIR%
REM echo MIGRATION_LIB=%MIGRATION_LIB%
REM echo MIG_CLASSPATH=%MIG_CLASSPATH%
%JDK_HOME%\bin\java -cp %MIG_CLASSPATH%  -DMIGRATION_DIR=%migration_dir%
com.bea.commerce.migration.tools.Migrator
echo on
```

MigratorBundle.properties and DataMigratorBundle.properties Files

Many of the settings for the migration process are stored in `MigratorBundle.properties` and `DataMigratorBundle.properties`. These files are in the `migration.jar` file located at `<PORTAL_HOME>\migration\lib` and unjarred to the following location during migration.

```
<MIGRATION_HOME>\com\bea\commerce\migration\tools\
```

- `DataMigratorBundle.properties` contains descriptions of and settings for the data migration tasks. It includes settings for whether a task is skippable—whether you can proceed to another task without completing it first. Making a task skippable is typically the only reason you need to modify this file.
- `MigratorBundle.properties` contains settings for the code migration process. It contains tool settings for data and code migration, including strings for the migration GUI, error messages, and so on. You typically do not need to edit this file unless you need to do for internationalization purposes.

Note: If you unjar or modify these files, you must rejar them.

The files are shown in Listing B-3 and Listing B-4.

Listing B-3 MigratorBundle.properties

```
#####################################################################
#  MIGRATOR's Resources
#####################################################################
#####################################################################
# Code Migration Mapper: Versions available
#
#  The structure points to XML; however, this
#  is all internal here. If we decided to go
#  for an actual product, let's put this in
#  a 'migratorconfig.xml' file.
#  Structure: versions give you all the roots, separated by commas: ","
#             for each root, there are two resources:
#              1. menuname, appears in the version combo-box
#              2. resource, is used to initialize the Mapper for Code Migration
```

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

# example with many versions:
mapper_resource=com.bea.commerce.migration.code.version.v4_0to7_0.MapperBundle


#####################################################################
#
#  PROPERTIES FOR THE GUI
#
#####################################################################
title=  BEA Portal Migrator V2.0
migratorWelcome= Welcome to the BEA Portal Migrator, V2.0
migratorIcon=/com/bea/commerce/migration/tools/images/migratorIcon.gif

#
# For the Migrator Menu bar
#
helpURL=http://www.bea.com/
fileMenu=File
fileAccess=To exit this tool.
exitMenu=Exit
exitAccess=To exit this tool.
HelpMainMenu=Help
HelpMainAccess=To get help on this tool

overviewHelp=Overview and Strategy
overviewAccess=What is migration and how to use it

codeMigratorHelp=Code Migrator Help
codeMigratorAccess=How to use the code migrator
codeMigratorHelpURL=http://www.bea.com/

dataMigratorHelp=Data Migrator Help
dataMigratorAccess=How to use the data migrator
dataMigratorHelpURL=http://www.sun.com/

# SHOULD BE FOUND UNDER '<migrator_dir>\doc'
overviewHelpURL=mainhelp.html

#
# The log message on starting a session
#
startLogMessage= ------------------ New Migration Session ---------------



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

```
#
# Code Migration GUI
#
#####################################################################

#
# JTable Columns/yes-no
#
fileNameColumn=File Name
modifiedColumn=Modified
errorColumn= Errors
reviewedColumn=Reviewed

yes=Yes
no=No

#
# Code Migration: directory selection
#
codeMigratorTitle = Code Migrator

codeMigratorDirTitle = Migration settings
originalDir    = Original source directory :
destinationDir = Migrated source directory :
chooseDir = Browse ...
chosenDir = Using directory:
noChosenDir = No directory selected
dirFilter = directories
startMigration = Start Code Migration Helper

#
#  External Viewer's command
#
external_viewer_command=c:\\winnt\\system32\\write %f%

#
# Code Migration: Analyzed file results
#
codeMigrationResultTitle = Result of Code Migration
analysisProgress = Progress of Files Analysis
externalViewerLabel=External Viewer Command (%f% for filename):
reviewed= {0} file(s) reviewed out of {1}
analyzed= {0} file(s) analyzed out of {1}

#
# Code Migration: Recommendation Strings
#
migratorChangeHeader=// WARNING: THE MIGRATOR HELPER MODIFIED THIS FILE, MAKE SURE
TO REVIEW THE CHANGES
```

```
migratorNote   =/*----------------------------- MIGRATION NOTE ----------
migratorNoteEnd=------------------------------ -------------- ----------*/\n
originalLines=-> original line(s):
importReplaced=In the ''import'' statement, replaced [{0}] with [{1}]
importWarning=WARNING: In the ''import'' statement, the package or class [{0}]
castReplaced=In the ''cast'' statement, replaced [{0}] with [{1}]
castWarning=WARNING: In the ''cast'' statement, the class [{0}]
instanceofReplaced=In the ''instance of'' statement, replaced [{0}] with [{1}]
instanceofWarning=WARNING: In the ''instance of'' statement, the class [{0}]
newObjectReplaced=In the ''instantiation'' (new) statement, replaced [{0}] with
[{1}]
newSameObject=In the ''instantiation'' (new) statement, the class [{0}]
varDeclarationReplaced=In the ''variable declaration'' statement, replaced [{0}]
with [{1}]
varDeclarationSameObject=In the ''variable declaration'' statement, the class
[{0}]
returnReplaced=In the ''return statement'', replaced [{0}] with [{1}]
returnWarning=WARNING: In the ''return statement'', , the class [{0}]
objectReferenceWarning=WARNING: the variable reference by [{0}] is of type [{1}]
fieldReplaced=In the ''field reference'' statement [{0}], variable of type [{1}],
replaced with [{2}]
fieldReferenceWarning=WARNING: found a field referenced by [{0}] for variable type
[{1}]
methodReferenceWarning=WARNING: found a method referenced by [{0}] for variable
type [{1}]
methodReplaceMessaqe=The following replacements have been specified for this
method:\n\tnew package: [{0}]\n\tnew class: [{1}]\n\tnew method: [{2}]\n\tnew
arguments: [{3}]
implementsDanger=DANGER: this class implements [{0}] now moved to: [{1}]
implementsWarning=WARNING: this class implements [{0}] see notes below
extendsDanger=DANGER: this class extends [{0}] now moved to: [{1}]
extendsWarning=WARNING: this class extends [{0}] see notes below
removedStr=(Removed)
deprecatedStr=(Deprecated)
noAdditionalDoc=[There was no additional documentation available regarding this
change]

#
# Code Migration: Result Strings
#
level0= 0 - no change
level1= 1 - comments
level2= 2 - modified
level3= 3 - danger

#####################################################################
#
# Data Migration GUI
#
```

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

dataMigratorTitle= Data Migrator

taskProgress= Task #{0} out of {1} Tasks

apply=Execute Task
back=<= Previous Task
next=Next Task =>

taskPanelTitle= Task
description=Description
status=Status

taskListTitle= Order of Tasks

startTask= \n -------------- Start of {0} ------------------ \n
endTask  = \n --------------  End of {0}  ------------------ \n


#####################################################################
#
# Data Migration - EBCCDataMigration 4.0 to 7.0 specific
#
#####################################################################

# String for the EBCCDataMigDialog
ebcc_dialog_title=EBCC Project Migration
ebcc_proj_scr_dir=EBCC Project Source Directory:
ebcc_proj_dest_dir=EBCC Project Destination Directory:
ejb_app_root_dir=Enterprise Application Root Directory:
ebcc_dialog_ok_button_text=OK
ebcc_dialog_cancel_button_text=Cancel

# String for the EBCCDataMigration task
app_sync_dir_name=application-sync
failed_verify_dir_name=failed-verification
ebcc_proj_file_ext=eaprj
proj_name_token=PROJ_NAME_TOKEN
app_root_token=APP_ROOT_TOKEN
encoding_token=ENCODING_TOKEN
ebcc_proj_file_template=<?xml version="1.0"
encoding="ENCODING_TOKEN"?>\n<project name="PROJ_NAME_TOKEN" version="1.0">\n\t
\

<server>\n\t\t<EnterpriseAppRoot>APP_ROOT_TOKEN</EnterpriseAppRoot>\n\t</server
>\n</project>

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

```
#
# Data Migration Strings
#
######################################################################

xml_schema_namespace=http://www.w3.org/2001/XMLSchema-instance
properties_to_XML_schema=properties-to-xml-1_0_1.xsd


# specify the document encoding to be used by the data migration tool when
# writing XML file, if not set here the default is UTF-8
xml_document_encoding=UTF-8

#
# For resources as Stream, will find xsl in .jar file at
# right location
#
xsl_resource = xsl/

#
# the relative location of the XML schemas
#
schema_location = /lib/schema/7_0

#
# Whenever a task needs an 'OK' flag, here it is
#
taskOK= : The task completed successfully.
taskError= : *** The task compeleted with errors (check migration.log file) or
was aborted ***

#
# For WebFlowMigrate and PipelineMigrate Caller.
#

genericConversion=Convert "{0}" to a generic properties XML format.
fileNotFound=Could not find: "{0}".
genericOK=The generic XML "{0}" was written out without error.
validation=Validate the file: "{0}".
validationError=Error: validating the file:"{0}".
validationOK=OK: validating the file:"{0}".
specificConversion=Convert the generic file, "{0}",  to a specific file, "{1}".
storing=Store result to: "{0}".

webflowEndMessage=Copy the file "{0}" to the proper WebApp location, and use \n \
   the Webflow Editor to modify and set up the Webflow for that WebApp. \n
  Once the Webflow is properly set up, use data sync to push it to the appropriate
server.
```

```
pipelineEndMessage=Copy the file "{0}" to the proper WebApp location, and use \n \
   the Webflow Editor to modify and set up the pipelines for that WebApp. \n \
   You will also want to move the tracking pipelines to a ''tracking'' namespace,
in \
   a file called "tracking.pln". \
   Once the pipelines are properly set up, use data sync to push it to the
appropriate server.


migrationError=There was an error with this migration task. \
            Please refer to the log file (tmp\migration.bat) for more information.

#
# Schema/Property Set migration strings
#
schemaError=Error while retrieving data from WLCS_SCHEMA.\n Cannot continue.
schemaOK=Retrieved data from WLCS_SCHEMA: OK.
schemaMigrationComplete=\nThis task completed satisfactorily.\n Please review the
messages and take any eventual corrective actions.
schemaBadType=Check for WLCS_PROP_MD properties of type "5" or "6", which are not
migrated by this tool. \n
schemaUnmigratedProps=\nA number of schemas were not migrated due to their
SCHEMA_GROUP_NAME.\n \
                      You will need to deal with them on a case by case basis,
checking in \n  \
      the WLCS_SCHEMA table, using the SCHEMA_GROUP_NAME. \n \
      The following list gives you an overview:
schemaUnmigrated=\t{1} Schemas not migrated, group name = "{0}" .

#####################################################################
#
# Logger Levels (prepended to log messages based on log 'type'
#
#####################################################################
logLevel1=Message:
logLevel2=Warning:
logLevel3=Error  :

#####################################################################
#
# Error Messages
#
#####################################################################
errorTitle=Error

editInstallFileError=Before running the Migrator, \n "{0}" must be edited
correctly.

URLerror=Could not open browser properly. \nCheck
```

```
http://edocs.bea.com/index.html.

mapperError= The Code Migrator''s configuration file (mapper) could not be
initialized!
sameDirError=Source and destination CANNOT be the same directory.
isNotDirError=: is NOT a directory.

viewerError=Could NOT spawn viewer on file with:
readError=Cannot read the file \n
createError=Cannot create the file \n
writeError=Cannot write to the file \n

storeConfig=store configuration settings
restoreConfig=restore configuration settings

restoreConfigError=Could not restore Code Migration settings from file:
storeConfigError=Could not store Code Migration settings to file:

storeAnalysisError=Could not persist the result of the analysis to file:
restoreAnalysisError=Could not read the result of the analysis from file:

restoreSettingsError=Could not read settings from file:
storeSettingsError  =Could not persist settings to file:

fileError= {0} could not be handled by the Code Migrator.\n Check the original
file.

dataMigratorResourceError=Could not read configuration for task {0}.
dataMigratorInitError = Could not initialize the Data Migrator:

error=* Error * :

################################################################################
###
#
# Status and error messages for Data Migration
#
################################################################################
###

#
# for com.bea.commerce.migration.data.CallerBase
#
data.callerbase.validate.doc.name=\nValidating transformed version of {0}
data.callerbase.validate.failed=Validation of {0} failed\n
data.callerbase.validate.success=Validation of {0} succeeded\n
data.callerbase.validate.IOException=An IOException occurred during
validation...
data.callerbase.validate.ValidationException=A ValidationException occurred
```

```
during validation...
data.callerbase.validate.start=Performing validation

#
# for com.bea.commerce.migration.data.version.XMLcaller
#
XMLcaller.retrieveXML.DBConnectionException=A DBConnectionException occurred
while attempting to retrieve document(s)...
XMLcaller.retrieveXML.SQLException=A SQLException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.IOException=An IOException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.ParserConfigurationException=A
ParserConfigurationException occurred while attempting to retrieve document(s)...
XMLcaller.retrieveXML.SAXException=A SAXException occurred while attempting to
retrieve document(s)...
XMLcaller.retrieveXML.UnsupportedArgumentTypeException=One or more invalid
arguments where used while attempting to retrieve document(s)...
XMLcaller.transform.doc.InvalidTransformerStateException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.doc.TransformerFailureException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.docs.InvalidTransformerStateException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...
XMLcaller.transform.docs.TransformerFailureException=An
InvalidTransformerStateException occurred while attempting to transform
document(s)...


#
# for
com.bea.commerce.migration.data.version.v4_0to7_0.EntitlementRulesetMigration
#
entitlement_ruleset.exe.retrieve=Retrieving ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.found=There are {0} ENTITLEMENT_RULESET entries to
migrate.
entitlement_ruleset.exe.transform.begin=Beginning transformation of
ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.transform.end=Ending transformation of
ENTITLEMENT_RULESET entries...
entitlement_ruleset.exe.validate.begin=Beginning validation of
ENTITLEMENT_RULESET documents...
entitlement_ruleset.exe.validate.success=All ENTITLEMENT_RULESET documents
validated...
entitlement_ruleset.exe.validate.failed=Some ENTITLEMENT_RULESET documents
failed validation. Check log file for errors.
```

```
entitlement_ruleset.exe.saving=Persisting transformed ENTITLEMENT_RULESET
documents.
entitlement_ruleset.exe.rule_set.not.found=No ENTITLEMENT_RULESET documents
found.
entitlement_ruleset.exe.abort=Terminating Entitlement Ruleset Migration due to
failure.
entitlement_ruleset.update.missing.clob=The ENTITLEMENT_RULESET row specified by
APPLICATION_NAME [{0}] and RULESET_URI [{1}] had no existing RULESET_DOCUMENT and
could not be updated.  There may be a data integrity problem in your orginal
ENTITLEMENT_RULESET data.

#
# for com.bea.commerce.migration.data.version.v4_0to7_0.EBCCDataMigration
#
EBCCDataMigration.exe.abort=Aborting migration task...
EBCCDataMigration.exe.failed=Migration task failed...
EBCCDataMigration.processAuto.create.projectfile=Creating EBCC project file
EBCCDataMigration.processAuto.starting.mig=Beginning migration of EBCC project
named: {0}\n
EBCCDataMigration.processAuto.end.mig=Finished migration of  EBCC project named:
{0}
EBCCDataMigration.processFile.copyfile=Copying file with unknown type named: {0}
EBCCDataMigration.processFile.processfile=Processing file named {0}
EBCCDataMigration.processFile.validation.success=Validation success for
transformed doc named: {0} \n
EBCCDataMigration.processFile.validation.failed=Validation failed for
transformed doc named: {0} \n
EBCCDataMigration.processFile.skip.validation=Skipping validation for the
document named: {0}.  The document may not be complete.\n
EBCCDataMigration.main.success=Successful completion
EBCCDataMigration.main.failed=Task failed
EBCCDataMigration.processCommandLine.invalid.num.args=Invalid number of
arguments...
EBCCDataMigration.processCommandLine.invalid.arg.value=The command line argument
{0} had an null or zero length value
EBCCDataMigration.processCommandLine.invalid.arg=The following invalid command
line argument was encountered: {0}
EBCCDataMigration.printUsage.usage=Usage :
```

Listing B-4 DataMigratorBundle.properties

```
#################################################################
#
#  Data Migrator's  Task Specification
#
```

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

#
# To create a new task:
#  1. add it to the list of tasks (tasks=...) in the right order
#  2. copy another task and replace the root (as in the task list)
#     and set the right values for those elements
#  3. in the DESCRIPTION of the task, state whether it is required
#     task and indicate alternatives if that task is skippable.
# The list of all tasks: the ORDER is quite important, as
# it determines the dependency of tasks
#
tasks=EBCCDataMigration, rdbms-data-sync, rdbms-portal,
rdbms-entitlement-ruleset, EntitlementRulesetMigration
#
# The EBCCDataMigration task
#
EBCCDataMigration_title=EBCC Data Migration

EBCCDataMigration_description=Migrates EBCC project data one project at a time.
Indicate the project source directory (an EBCC project directory should have at
minimum a child directory called application-sync)in the Source dialog box.
Indicate the output directory, where the migrated project will be written, in the
Destination dialog box.  The migration tool will write the migrated project to
this directory preserving the project name and structure.

EBCCDataMigration_classname=com.bea.commerce.migration.data.version.v4_0to7_0.E
BCCDataMigration
EBCCDataMigration_gif=null
EBCCDataMigration_skippable=true
#
# The rdbms-data-sync task
#
rdbms-data-sync_title=Data Sync RDBMS Migration
rdbms-data-sync_description=Resets Data Sync tables prior to post-migration data
sync.
rdbms-data-sync_classname=com.bea.commerce.migration.data.version.v4_0to7_0.Dat
aSyncDB
rdbms-data-sync_gif=null
rdbms-data-sync_skippable=true

#
# The rdbms-portal task
#
rdbms-portal_title=Portal RDBMS Migration
rdbms-portal_description=Migrates Portal tables.
rdbms-portal_classname=com.bea.commerce.migration.data.version.v4_0to7_0.Portal
DB
rdbms-portal_gif=null
```

```
rdbms-portal_skippable=true

#
# The rdbms-entitlement-ruleset task
#
rdbms-entitlement-ruleset_title=ENTITLEMENT_RULESET RDBMS Migration
rdbms-entitlement-ruleset_description=Creates new 7.0 ENTITLEMENT_RULESET table.
rdbms-entitlement-ruleset_classname=com.bea.commerce.migration.data.version.v4_
0to7_0.EntitlementRulesetDB
rdbms-entitlement-ruleset_gif=null
rdbms-entitlement-ruleset_skippable=true
#
# The EntitlementRulesetMigration task
#
EntitlementRulesetMigration_title=Entitlement Ruleset Data Migration
EntitlementRulesetMigration_description=Migrates Entitlement Ruleset Data.
EntitlementRulesetMigration_classname=com.bea.commerce.migration.data.version.v
4_0to7_0.EntitlementRulesetMigration
EntitlementRulesetMigration_gif=null
EntitlementRulesetMigration_skippable=true
```

Migration Log File

The Migration Viewer records the actions in the `migration.log` file in the `migration` directory. This file is created when you run the migration.

The log file also includes system properties information such as tool versions, service packs installed, and other migration environment information, Use this file if you encounter difficulties or have questions while using the migration tool. The file contains more complete information than the messages on screen, and it contains the entire stack trace for the migration process.

.SQL Files for Customized Database Conversion

All the SQL used to migrate 4.0 databases to 7.0 databases for each database and version is included in the `migration` directory. There is one file for each type and version of database; for example, `oracle-817-v4_0to7_0.sql`. If you made modifications to your databases, you will need to use these files to create the appropriate modified SQL statements and run them. Instructions are included in "Manually Migrate Any Databases You Have Changed" on page 2-14.

Migration Viewer and migrinfo.html for API Changes

The `migrinfo.html` file contains the API changes for this release. Use the Migration Viewer as you migrate code to help in your migration efforts. See "Viewing Code Changes Using the Migration Viewer Tool" on page A-1.

Details of the E-Business Control Center Project Migration

Typically, your E-Business Control Center project migration should go smoothly. However, some variations in content or currentness can produce different results. Review this information before you continue.

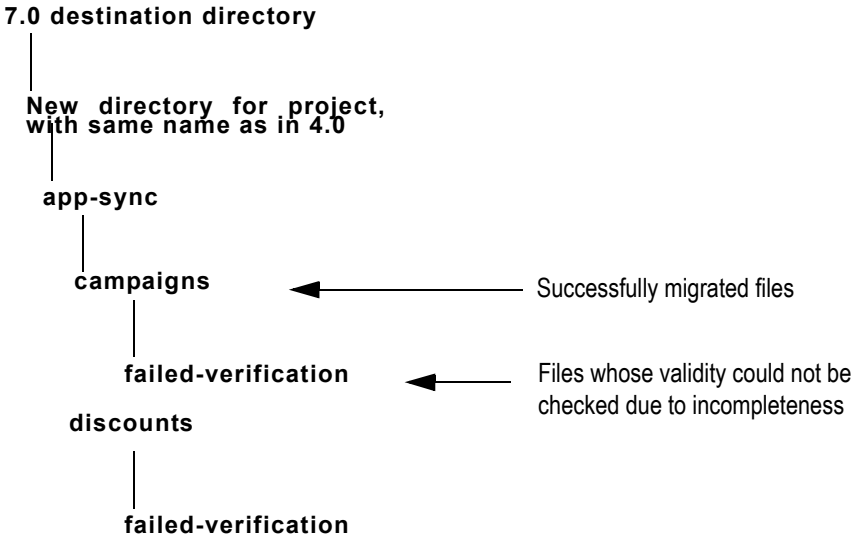
Completeness and validity are evaluated All projects are transformed during migration to a different format. The migration process also determines whether the project file contains valid data. However, if the project is incomplete, the validity cannot be checked.

- If the file is complete but corrupt or otherwise invalid, the file is migrated to a different migration folder, whose name you define. The default name for the directory is `failed-verification`. You can change the name of this folder; see "Edit the migrator.bat and migration_install.properties Files" on page 2-7.
- If the file is incomplete, the validity is not checked. The file is copied over to the standard migration destination directory. You also set this by following the instructions in "Edit the migrator.bat and migration_install.properties Files" on page 2-7.

Because incomplete files are not checked for validity, it is a good idea to ensure that the files are complete before you migrate.

The directories are shown in Figure B-1.

Figure B-1 Destination Directories



Determination of completeness The determination of whether a file is incomplete is set by the `iscomplete` attribute in the file, with the following exceptions:

- In portal and portlet documents, instead of an `iscomplete` attribute in the root tag at the top of the file, there is a separate `iscomplete` tag.
- `.wfx` files are templates, and the migration process always attempts to validate them.
- `.pln` files have an `incomplete` attribute; if the value is `false`, the file is incomplete; if the attribute does not appear, the file is complete.

If incomplete or invalid files are migrated or copied To correct any problems with the resulting copied or migrated files, use the E-Business Control Center to enter the correct or full information. If they are badly corrupted, that is, invalid, you might not be able to open them in the E-Business Control Center. Incomplete files can open without problems.

Portal Webflow Changes

Note: This section is supplementary information for "Configure Webflows and JSPs" on page 2-31.

This section lists the changes to three of the portal Webflows that your Portal Web application will need. If your Web application is not a portal, you do not need these changes.

The changed or added nodes are listed for each Webflow, but you should open a copy of the 7.0 version of the Webflow for a detailed reference of how each new and changed node should look.

Tools Webflow

Presentation Nodes

- `portal-prefs.jsp` – The outgoing event `link.change_name` event to node `change_name.jsp` has been added.
- `change-name.jsp` – This is a new node in 7.0; add it and all associated events.

Input Processors

- `displayNameSpecialCharacterProcessor` – This is a new node in 7.0; add it and all associated events.

Wildcard Processor Nodes

- `com.bea.p13n.appflow.webflow.forms.MissingFormFieldException` – This is a new exception in 7.0.
- `com.bea.p13n.appflow.exception.InvalidArgumentException` – This is a new exception in 7.0.

Security Webflow

Input Processors

- `loginProcessor` – The outgoing event `success` to `groupProcessor` and the outgoing event `failure` event to `badlogin.jsp` have been added.
- `specialCharacterProcessor` – This is a new node in 7.0; add it and all associated events.

user_account Webflow

Presentation Nodes

- `new_user.jsp` – Removed outgoing event `button.go` (to `userProcessor`) and added outgoing event `button.create_user_account` to `createAccountFormProcessor`.
- `account_info.jsp` – This is a new node in 7.0; add it and all associated events.

Input Processors

- `postLoginProcessor` – The outgoing event `success` event to `portalRefreshProcessor` in the `portal` Webflow has been added, and the destination for the outgoing event `user.create` has been changed from `dispatchUserRegEventProcessor` to `createUserProfileProcessor`.
- `getAccountProcessor` – This is a new node in 7.0; add it and all associated events.
- `createAccountFormProcessor` – This is a new node in 7.0; add it and all associated events.
- `updateAccountFormProcessor` – This is a new node in 7.0; add it and all associated events.
- `updateUserProfileProcessor` – This is a new node in 7.0; add it and all associated events.
- `createUserProfileProcessor` – This is a new node in 7.0; add it and all associated events.
- portalRefreshProcessor - Two new events were added. The user.login event also goes to the portalRefreshProcessor. The user.create event goes to the createUserProfileProcessor.

Wildcard Presentation Nodes

- `link.account_info` – This is a new wildcard in 7.0.
- `link.new_user` – This is a new exception in 7.0.

Index

E

F

G

H

J

R

S

T

V

W

X