

Oracle® WebLogic Portal

Upgrade Guide

10g Release 3 (10.3)

September 2008

ORACLE®

Copyright © 2007, 2008, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1. Overview of the Upgrade Process to WebLogic Portal 10.3

| | |
|-------------------------------------------------------|-----|
| Definitions | 1-2 |
| Portal 9.2, 10.0, 10.2 to 10.3 Upgrade Overview | 1-2 |
| Portal 8.1 to 10.3 Upgrade Overview | 1-3 |
| Library Module Changes | 1-3 |
| Supported Features Comparison | 1-4 |
| Security | 1-4 |
| Event Handling | 1-5 |
| Autonomy | 1-5 |

2. Upgrading to WebLogic Portal 10.3

| | |
|-------------------------------------------------------|-----|
| Before You Begin | 2-1 |
| Database Changes During Upgrade | 2-1 |
| DDL Files | 2-2 |
| Backing Up Your Applications and Data | 2-4 |
| Removing Unneeded Database Objects | 2-4 |
| Upgrading to WebLogic Portal 10.3 | 2-4 |
| Upgrading Your Portal Domain | 2-5 |
| Upgrading from Portal 9.2 and 10.0/10.2 to 10.3 | 2-6 |
| Upgrading from Portal 8.1 to 10.3 | 2-7 |
| Functional Changes for WebLogic Portal 10.3 | 2-8 |

A. Functional Changes Affecting Your WebLogic Portal Environment

| | |
|----------------------------------------------------------------------------------|------|
| Functional Changes from Portal 8.1 to 10.3 | A-1 |
| Upgrading to the SQL Authenticator | A-2 |
| Upgrading Federated Portals from 8.1 to 10.3 | A-3 |
| Upgrading UUPs | A-6 |
| Understanding JSP Tag Changes | A-8 |
| Upgrading Autonomy Search Services | A-9 |
| Using 8.1 Search within a 10.0 Portal Application | A-9 |
| Manually Upgrading Passwords in Content Management Repositories | A-11 |
| Maintaining Content Queries | A-11 |
| Upgrading Look & Feels | A-12 |
| Import Wizard Does Not Handle Some Legacy Jar Files | A-12 |
| Changes in Behavior Between Struts 1.1 and 1.2 | A-12 |
| Portlet State Persistence | A-13 |
| WSRP Security Compatibility | A-13 |
| Working with Encoding in HTTP Responses | A-13 |
| Disconnected Desktop Requires desktopStateShared Property | A-14 |
| Correcting Duplicate Portlet Category Names in an Upgraded Application | A-14 |
| Propagation Utility Web Application Obsolete | A-15 |
| Definition Labels Not Editable in 10.3 | A-15 |
| Propagation Inventory Compatibility | A-15 |
| Functional Changes from Portal 9.2 to 10.2/10.3 | A-15 |
| Upgrading Federated Portals | A-16 |
| Upgrading UUPs | A-16 |
| Disconnected Desktop Requires desktopStateShared Property | A-16 |
| Upgrading Search Services | A-16 |

| | |
|----------------------------------------------|------|
| WSRP Security Compatibility | A-22 |
| Upgrading Propagation Scripts. | A-22 |
| Propagation Inventory Compatibility. | A-23 |

B. Performing Database Upgrade Tasks Manually

| | |
|------------------------------------------------------------------------------|-----|
| Upgrading your Main WebLogic Portal 8.1, 9.2, or 10.0 Database | B-2 |
| Upgrading 8.1, 9.2, or 10.0 PointBase Databases | B-3 |
| Upgrading to the 10.3 WebLogic Server SQL Authenticator | B-3 |
| Upgrading Separate 8.1 Behavior Tracking Databases | B-4 |
| Upgrading Additional 8.1, 9.2, or 10.0 Content Management Databases. | B-4 |
| Upgrading a 9.2 GroupSpace Database | B-5 |
| Dropping Deprecated RDBMS Authenticator Tables After Upgrade. | B-5 |
| Dropping Deprecated Compoze Database Tables After Upgrade. | B-5 |

C. Upgrading WebLogic Portal 8.1 Projects to WebLogic Portal 10.3

| | |
|------------------------------------------------------------------------------|-----|
| WebLogic Portal 8.1 Features Not Supported in WebLogic Portal 10.3 | C-1 |
| Upgrade Considerations and Tips | C-2 |
| Command-Based and Ant Task Upgrade Not Supported. | C-2 |
| Enabling Communities Features in Upgraded Visitor Tools | C-2 |
| Upgrading Look And Feels | C-3 |
| Upgrading Custom Controls that Have Custom Properties | C-4 |
| Upgrading Tuned Thread Pools for Forked Portlets | C-4 |
| Upgraded Applications Use Struts 1.1 and Related J2EE Libraries | C-4 |
| Changes in Behavior Between Struts 1.1 and 1.2 | C-5 |
| Ampersand Entities in Portal URLs | C-5 |
| Upgrading Individual application-config.xml Files Later | C-6 |

| | |
|-------------------------------------------------------------------------------------------------|-----|
| Correcting Duplicate Portlet Category Names Before Propagating an Upgraded Application | C-6 |
|-------------------------------------------------------------------------------------------------|-----|

Overview of the Upgrade Process to WebLogic Portal 10.3

This section provides an overview of the strategies and procedures for upgrading Oracle WebLogic Portal to 10.3. You can upgrade directly to Portal 10.3 from the following WebLogic Portal applications:

- Portal 10.0 and 10.0 Maintenance Pack 1 (MP1)
- Portal 9.2 and 9.2 MP1
- Portal 8.1 SP4, SP5, and SP6

You are not required to upgrade from WebLogic Portal 8.1 to 9.2, from 9.2 to 10.0/10.2, and then from 10.0/10.2 to 10.3

The following topics are covered in this chapter:

- [Definitions](#)
- [Portal 9.2, 10.0, 10.2 to 10.3 Upgrade Overview](#)
- [Portal 8.1 to 10.3 Upgrade Overview](#)
- [Library Module Changes](#)
- [Supported Features Comparison](#)

Some of tasks associated with a WebLogic Portal upgrade are performed by running the WebLogic Upgrade Wizard. The WebLogic Upgrade Wizard is described in *[Upgrading WebLogic Application Environments](#)*.

Definitions

To clarify the different activities described by this document, a brief list of terms is included:

Migration

Moving an application and domain from a third-party technology to an Oracle product. (For example, migrating a customer from IBM to Oracle.)

Upgrade

Updating Oracle platform (and components) from older release or Service Pack to newer release or Maintenance Pack. This includes updating existing application and domain to run in a newer version, for example, 9.2 MP1 to 10.3.

The process required to upgrade an application environment depends on the scope of the application. An application environment includes a WebLogic domain and any applications and application resources associated with the domain. It may also include external resources, such as firewalls, load balancers, databases, and LDAP servers.

Interoperability

(1) The capability of an application deployed in one release or service pack to communicate with another application that is deployed in a different release or service pack. (2) The capability of WebLogic Platform components to communicate with third-party software using standard protocols.

Compatibility

Application built using one release or Service Pack running in another release or Service Pack. This might involve rebuilding the application.

Portal 9.2, 10.0, 10.2 to 10.3 Upgrade Overview

You can upgrade your WebLogic Portal 9.2, 9.2 MP1, 10.0, 10.0 MP1, and 10.2 applications to 10.3. The upgrade process involves upgrading your WebLogic Portal domain and applications. Upgrade the portal domain using the WebLogic Upgrade Wizard before you upgrade your portal application. The upgrade procedure is explained in [Chapter 2, “Upgrading to WebLogic Portal 10.3.”](#)

The WebLogic Portal APIs have been maintained in WebLogic Portal 10.3 (except for the Commerce API, which was deprecated in 10.0), and most core formats for the database and file based assets have not changed. Where changes have been made, tools are provided to upgrade you to the new format, or provide manual changes where needed.

The 9.2 and 10.0/10.2 upgrades are performed on a `.project` file. After you upgrade to Portal 10.3, you cannot go back to a previous portal version. If you are upgrading from WebLogic Portal 9.2 or 10.0/10.2 to 10.3, the PointBase database version has already been upgraded.

You can choose one of the following methods to upgrade your Portal 9.2 or 10.0/10.2 application to 10.3:

- Open an existing 9.2 or 10.0/10.2 portal application as a workspace in Workshop for WebLogic
- Import an existing exploded 9.2 or 10.0/10.2 project into Workshop for WebLogic
- Import an archived 9.2 or 10.0/10.2 project file into Workshop for WebLogic

Tip: You cannot upgrade an existing EAR application; only a project can be upgraded.

For instructions on upgrading, see [“Upgrading from Portal 9.2 and 10.0/10.2 to 10.3” on page 2-6.](#)

Portal 8.1 to 10.3 Upgrade Overview

WebLogic Portal enables you to upgrade your 8.1 SP4, SP5, and SP6 applications directly to 10.3. Most WebLogic Portal APIs have been maintained in WebLogic Portal 10.3 (except for the Commerce API, which was deprecated in 10.0), and most core formats for the database and file based assets have not changed. Where changes have been made, tools are provided to upgrade you to the new format, or provide manual changes where needed.

The upgrade process involves upgrading WebLogic Portal 8.1 portal applications and resources to WebLogic Portal 10.3. The 8.1 to 10.3 upgrade is performed on a `.work` file. If you customized how you set the domain in your start scripts, your changes will be overwritten when you run the WebLogic Portal 10.3 start scripts.

Note: When you upgrade from WebLogic Portal 8.1 to 10.3, your PointBase database is upgraded.

For instructions on upgrading, see [“Upgrading from Portal 8.1 to 10.3” on page 2-7.](#)

Library Module Changes

The Domain Upgrader tool is responsible for adding or removing library modules, as necessary. It also modifies existing library module path information to point to the new product installation.

Removed libraries are deleted from the `config.xml` file. If you upgrade from 8.1.x to 10.3, the libraries are added to your `config.xml` file. If you upgrade from 9.2 or 10.0 to 10.3, the module version number changes to 10.3 in your `config.xml` file. If you upgrade from 10.0 MP1 to 10.3, the domain upgrader removes the maintenance-lib references.

Supported Features Comparison

This section outlines significant feature changes between older version of WebLogic Portal and the Portal 10.3 release.

Security

Note: The following section applies only to an upgrade from WebLogic Portal 8.1 to 10.3.

WebLogic Portal 8.1 included a WebLogic Portal-specific RDBMSAuthenticator. This has been deprecated. WebLogic Server 9.2 contains a new default SQLAuthenticator authentication provider, which contains an RDBMS user store for users and groups. Oracle recommends upgrading to the new WebLogic Server SQLAuthenticator.

When you run the WebLogic Upgrade Wizard to upgrade your domain, it determines whether or not you are using the RDBMSAuthenticator in your 8.1 installation. If the RDBMSAuthenticator is detected, the WebLogic Upgrade Wizard prompts you to choose whether to upgrade to the WebLogic SQLAuthenticator as your default authentication provider or continue to use your existing RDBMS user store. Replacing the existing authentication provider with the new WebLogic Server SQLAuthenticator upgrades all content, including personalization features. You can also choose to manually upgrade your personalization features to the Portal 10.3 RDBMS user store later.

Choose one of the following options when upgrading your user store:

- Upgrade users and groups – Choose to automatically upgrade your users and groups from WebLogic Portal 8.1 to the new RDBMS user store, which is part of the new default WebLogic SQLAuthenticator authentication provider. When you run the WebLogic Upgrade Wizard and it detects the Portal 8.1 RDBMSAuthenticator, you can select the Upgrade RDBMSAuthenticator option. Selecting this option replaces the existing authentication provider with the new SQLAuthenticator authentication provider and upgrades your user store, including users and groups. Your `config.xml` file is also updated.
- Do not upgrade users and groups – Choose to continue to use the RDBMS user store from the default RDBMSAuthenticator in your WebLogic Portal 8.1 installation. When you run

the WebLogic Upgrade Wizard and it detects your Portal 8.1 RDBMSAuthenticator, you can select the Do not upgrade RDBMSAuthenticator option. You can choose to manually upgrade your users and groups to the Portal 9.2 RDBMS user store later.

If you do not upgrade your user store during the domain upgrade process, you can perform a manual upgrade later. The script to upgrade from the WebLogic Portal-specific RDBMS Authenticator to the WebLogic SQL Authenticator is

```
<WLPORTAL_HOME>/p13n/db/<dbms>/upgrade_fromdbmsauth_towlssqlauth.sql.
```

For additional information, see [“Upgrading 8.1, 9.2, or 10.0 PointBase Databases” on page B-3](#).

Note: If you upgrade a WebLogic Portal 8.1 application to 10.3 and you use the `UserProviderControl.createUser()` class in the upgraded domain, you might see a `javax.security.auth.login.LoginException` error when a new user attempts to log into WebLogic Portal. This occurs because by default new users in a WebLogic Portal 10.3 domain are created in the `SQLAuthenticator` and not in a migrated authentication provider (which normally is configured with a JAAS flag set to `REQUIRED`). Since the WebLogic Portal domain Upgrade Wizard does not adjust your JAAS settings or remove your existing authentication provider, you must adjust the JAAS setting or delete the authentication provider (as appropriate) to avoid this exception.

Event Handling

Events can target content to a desired audience.

Note: Events will fire for a content repository that was upgraded to 10.3 (unless you turned event tracking turned off at the repository level). Events can include repository configuration changes, as well as content additions, updates, and deletions to the repository. Events do not fire for content in an 8.1 or 9.2 repository that was not upgraded. Events are fired for content that is added, updated, or removed from that repository.

If you created a separate behavior tracking database in version 8.1 or 9.2, upgrade it as described in [“Upgrading Separate 8.1 Behavior Tracking Databases” on page B-4](#).

Autonomy

A new Autonomy engine is installed with WebLogic Portal 9.2 and higher. It is disabled by default after upgrade so it can be properly configured before launching. Autonomy services that have been enabled in WebLogic Portal 8.1 are also disabled after upgrade. For information about installing and configuring search services, see the [Integrating Search Guide](#).

Overview of the Upgrade Process to WebLogic Portal 10.3

In addition, the Autonomy APIs and the all content management APIs, including search, were deprecated in WebLogic Portal 9.2. If you want to continue to use the deprecated APIs, you need to manually add the older Autonomy APIs to your application, as described in [“Using 8.1 Search within a 10.0 Portal Application” on page A-9](#).

For information about upgrading WebLogic Portal 8.1 to the new version of Autonomy that is included with WebLogic Portal 10.3, see [“Upgrading Autonomy Search Services” on page A-9](#).

Upgrading to WebLogic Portal 10.3

This chapter describes upgrade tasks related to upgrading your WebLogic Portal *product* to 10.3.

This chapter contains the following sections:

- [Before You Begin](#)
- [Upgrading to WebLogic Portal 10.3](#)
- [Functional Changes for WebLogic Portal 10.3](#)

Before You Begin

You can upgrade directly to WebLogic Portal 10.3 from 8.1, 9.2, and 10.0/10.2 portal applications. For information on upgrading earlier versions to WebLogic Portal 8.1, see the [Upgrade Guide for Version 8.1](#).

Database Changes During Upgrade

The WebLogic Upgrade Wizard executes database scripts to add and modify database tables for WebLogic Portal 10.3. Before you run the Upgrade Wizard to perform database upgrade tasks, you should perform full backups of all WebLogic Portal databases.

Note: Oracle 8.1.7 (DBMS version and drivers) is no longer supported. Upgrade to either Oracle 9i or 10G by following your vendor's instructions before you upgrade to WebLogic 10.3.

WebLogic Portal 8.1 supports only the Oracle Thin driver. A WebLogic Portal 8.1 domain that was configured with either the `weblogic.jdbcx.oracle.OracleDataSource` or the

`weblogic.jdbc.oracle.OracleDriver` driver is not supported. The domain might upgrade successfully, but the following error occurs when you start the server:

```
<Warning> <JDBC> <BEA-001129> <Received exception while creating connection
for pool "portalDataSourceAlwaysXA": Invalid Oracle URL specified>
<Error> <Deployer> <BEA-149205> <Failed to initialize the application
'portalDataSourceAlwaysXA' due to error
weblogic.application.ModuleException: - with nested exception:
[weblogic.common.ResourceException: Invalid Oracle URL specified]
```

When you run the 10.3 Upgrade Wizard, the default is to automatically upgrade the following two databases:

- The main WebLogic Portal database
- The GroupSpace database (for 9.2 to 10.3 upgrades)

You can choose to defer database upgrades and do them manually. Certain database upgrade tasks, such as dropping deprecated database objects, upgrading separate behavior tracking databases, and upgrading separate content management databases require a manual database upgrade.

For additional information on how to perform database upgrade tasks manually, see [Appendix B, “Performing Database Upgrade Tasks Manually.”](#)

DDL Files

[Table 2 -1](#) lists the .SQL files containing the Data Definition Language (DDL) for the database upgrade from WebLogic Portal 8.1 to 9.2.

Table 2 -1 Files Containing the DDL for Upgrading from 8.1 to 9.2

| Directory Name | File Name |
|--------------------------------|------------------------------------------------------------------------------------------------|
| <WLPORTAL_HOME>/p13n/db/<DBMS> | p13n9_create.tables.sql |
| | p13n9_create.indexes.sql |
| | upgrade_fromdbmsauth_towlssqlauth.sql (if the domain is configured for the DBMS authenticator) |
| | dep9_drop_tables.sql (manual upgrade only) |

Table 2 -1 Files Containing the DDL for Upgrading from 8.1 to 9.2

| Directory Name | File Name |
|----------------------------------------|--------------------------------------------|
| <WLPORTAL_HOME>/content-mgmt/db/<DBMS> | cm9_create_tables.sql |
| | cm9_create_indexes.sql |
| | cm9_create_fkeys.sql |
| | cm9_create_triggers.sql |
| | cmv9_create_tables.sql |
| | cmv9_create_indexes.sql |
| <WLPORTAL_HOME>/portal/db/<DBMS> | pf9_create_tables.sql |
| | pf9_create_fkeys.sql |
| | pf9_create_views.sql |
| | comm_create_tables.sql |
| | comm_create_fkeys.sql |
| | comm_create_indexes.sql |
| | comm_create_views.sql |
| | comm_create_triggers.sql |
| | dep9_drop_tables.sql (manual upgrade only) |

[Table 2-2](#) lists the .SQL files containing the DDL for the database upgrade from WebLogic Portal 9.2 to 10.0.

Table 2-2 Files Containing the DDL for Upgrading from 9.2 to 10.0

| Directory Name | File Name |
|----------------------------------------|-------------------------|
| <WLPORTAL_HOME>/portal/db/<DBMS> | pf10_create_tables.sql |
| <WLPORTAL_HOME>/content-mgmt/db/<DBMS> | cm10_create_tables.sql |
| | cmv10_create_tables.sql |

Table 2-3 Files Containing the DDL for Upgrading from 10.0/10.2 to 10.3

| Directory Name | File Name |
|----------------------------------------|-----------------------------------------------------------------------------------|
| <WLPORTAL_HOME>/p13n/db/<DBMS> | p13n102_create.tables.sql p13n102_drop.tables.sql p13n103_create_tables.sql |
| <WLPORTAL_HOME>/portal/db/<DBMS> | pf102_create_tables.sql pf102_drop_tables.sql |
| <WLPORTAL_HOME>/content-mgmt/db/<DBMS> | cmv102_create_tables.sql |

Backing Up Your Applications and Data

Before you upgrade your application environment, you should manually back up the domain and any external application and application database resources in a separate process. You should back up the relevant information on all machines in the domain. The wizard backs up only the domain directory and does not preserve file permissions.

Removing Unneeded Database Objects

You can clean up an existing database (any database prior to Portal 10.3) that contains unused and deprecated WebLogic Commerce Services database objects.

Run the following scripts to remove the objects from the database:

- <WLPORTAL_HOME>/portal/db/<DBMS>/wlcs_drop_fkeys.sql
- <WLPORTAL_HOME>/portal/db/<DBMS>/wlcs_drop_tables.sql

Upgrading to WebLogic Portal 10.3

The upgrade process requires you to upgrade your portal domain and your portal application. Upgrading from the following WebLogic Portal versions is supported: 9.2, 9.2 MP1, 10.0, 10.0 MP1, and 10.2 applications.

This section contains the following topics:

- [Upgrading Your Portal Domain](#)

- [Upgrading from Portal 9.2 and 10.0/10.2 to 10.3](#)
- [Upgrading from Portal 8.1 to 10.3](#)

Upgrading Your Portal Domain

This section provides upgrade instructions for WebLogic Portal database and metadata files.

As you prepare to upgrade your portal application, review the Workshop for WebLogic upgrade documentation available by choosing **Help > Help Contents > Oracle Workshop for WebLogic User's Guide > Upgrading to Workshop for WebLogic 10.x**. The Workshop for WebLogic documentation includes step-by-step instructions for using the Import Wizard, and detailed information about what happens during the upgrade process and any required manual pre- or post-upgrade tasks.

Perform the following steps to upgrade your portal domain:

1. Verify that the WebLogic domain is not running.
2. Upgrade the portal domain using the WebLogic Upgrade Wizard before you upgrade your portal application. To start the WebLogic Upgrade Wizard in graphical mode and upgrade a WebLogic domain on a Windows platform:

- Run `upgrade.cmd` from the `<WLPORTAL_HOME>\common\bin` directory

Note: Do not use the menu option **Start > All Programs > Oracle Products > WebLogic Server 10.x > Tools > Domain Upgrade Wizard**. This launches the WebLogic Server domain upgrade wizard.

Note: The WebLogic Upgrade Wizard is described in detail in [Upgrading WebLogic Application Environments](#).

The wizard upgrades your portal, content management, and personalization database data. It optionally upgrades your 8.1 RDBMS Authenticator to the WebLogic 10.3 SQL Authenticator. For more information on upgrading your user store, see “[Security](#)” on [page 1-4](#).

3. Read [Appendix A, “Functional Changes Affecting Your WebLogic Portal Environment”](#).
4. As needed, upgrade individual applications as described in the [Production Operations Guide](#).

Tip: If you customized how you set the domain in your start scripts, your changes will be overwritten when you run the WebLogic Portal 10.3 start scripts. You should manually

copy any start script (and `setDomainEnv.cmd/sh`) modifications you made and want to preserve in the upgraded domain.

Upgrading from Portal 9.2 and 10.0/10.2 to 10.3

You can upgrade your WebLogic Portal 9.2, 9.2 MP1, 10.0, 10.0 MP1, or 10.2 applications to 10.3. The WebLogic Portal APIs have been maintained in WebLogic Portal 10.3 (except for the Commerce API, which was deprecated in 10.0), and most core formats for the database and file based assets have not changed. For general information on this type of upgrade, see [“Portal 9.2, 10.0, 10.2 to 10.3 Upgrade Overview”](#) on page 1-2.

The upgrade process involves upgrading WebLogic Portal applications and resources to WebLogic Portal 10.3. The 9.2 and 10.0/10.2 upgrades are performed on the Workshop for WebLogic Eclipse project folder. After you upgrade to Portal 10.3, you cannot go back to a previous portal version.

Choose one of the following ways to upgrade your Portal 9.2 or 10.0/10.2 application to 10.3:

- Open an existing 9.2 or 10.0/10.2 portal application as a workspace in Workshop for WebLogic
- Import an existing exploded 9.2 or 10.0/10.2 project into Workshop for WebLogic
- Import an archived 9.2 or 10.0/10.2 project file into Workshop for WebLogic

Tip: You can choose to point your upgraded application to a new or upgraded 10.3 domain.

Perform the following steps to upgrade a 9.2 or 10.0/10.2 application to 10.3:

1. In Workshop for WebLogic, ensure that **Project > Build Automatically** is enabled to resolve any build errors. By default, it is enabled.
2. Upgrade your domain using the WebLogic Upgrade Wizard. For more information, see [“Upgrading Your Portal Domain”](#) on page 2-5.
3. Perform one of the following steps in Workshop for WebLogic to upgrade an existing WebLogic Portal 9.2 and 10.0 application:
 - a. Open an existing portal application as a workspace in Workshop for WebLogic – Choose **File > Switch Workspace**, locate the existing workspace, and click **OK**. You can also choose the existing portal application in the Workspace Launcher when you launch Workshop for WebLogic, and click **OK**.

Tip: To open an existing portal application as a workspace, ensure that the workspace and the application are created and accessed on the same machine. The project directory stores non-transferable preferences.

- b. Import an exploded 9.2 or 10.0 project into Workshop for WebLogic – Choose **File > Import > General > Existing Projects into Workspace**, and click **Next**. Click **Select root directory**, locate the project’s directory, and click **Finish**. Click **Finish** again on the next dialog box to complete the upgrade.
- c. Import an archived 9.2 or 10.0 project file (such as a .ZIP file) into Workshop for WebLogic – Choose **File > General > Existing Projects into Workspace**, and click **Next**. Click **Select archive file**, locate the archive file, and click **Finish**. Click **Finish** again on the next dialog box to complete the upgrade.

Upgrading from Portal 8.1 to 10.3

WebLogic Portal enables you to upgrade your 8.1 SP4, SP5, and SP6 applications directly to 10.3. Most WebLogic Portal APIs have been maintained in WebLogic Portal 10.3 (except for the Commerce API, which was deprecated in 10.0), and most core formats for the database and file based assets have not changed. Where changes have been made, tools are provided to upgrade you to the new format, or provide manual changes where needed. For general information on this type of upgrade, see [“Portal 8.1 to 10.3 Upgrade Overview” on page 1-3](#).

The upgrade process involves upgrading WebLogic Portal 8.1 portal applications and resources to WebLogic Portal 10.3. The 8.1 to 10.3 upgrade is performed on a `.work` file. When you upgrade from WebLogic Portal 8.1 to 10.3, your PointBase database is upgraded.

The high-level steps in the upgrade process include the following:

1. Upgrade your domain using the WebLogic Upgrade Wizard. For more information, see [“Upgrading Your Portal Domain” on page 2-5](#).
2. Upgrade existing WebLogic Portal 8.1 SP4, SP5, and SP6 applications to run in WebLogic Portal 10.3. You can do this automatically using the Import utility that is provided in Workshop for WebLogic. For additional details on importing an 8.1 application, see [Appendix , “Upgrading WebLogic Portal 8.1 Projects to WebLogic Portal 10.3.”](#)

Tip: If you customized how you set the domain in your start scripts, your changes will be overwritten when you run the WebLogic Portal 10.3 start scripts.

Functional Changes for WebLogic Portal 10.3

Review the functional changes that are described in [Appendix A, “Functional Changes Affecting Your WebLogic Portal Environment.”](#) If any manual upgrade tasks are required for your particular environment, perform those tasks as instructed.

Functional Changes Affecting Your WebLogic Portal Environment

This appendix describes functional changes in WebLogic Portal 10.3 that affect your upgraded environment and might require you to perform manual tasks.

This chapter includes the following sections:

- [Functional Changes from Portal 8.1 to 10.3](#)
- [Functional Changes from Portal 9.2 to 10.2/10.3](#)

Functional Changes from Portal 8.1 to 10.3

The following sections describe changes that occur when you upgrade directly from WebLogic Portal 8.1 to 10.3. You might be required to perform manual tasks.

This section includes these topics:

- [Upgrading to the SQL Authenticator](#)
- [Upgrading Federated Portals from 8.1 to 10.3](#)
- [Upgrading UUPs](#)
- [Understanding JSP Tag Changes](#)
- [Upgrading Autonomy Search Services](#)
- [Using 8.1 Search within a 10.0 Portal Application](#)
- [Manually Upgrading Passwords in Content Management Repositories](#)

- [Maintaining Content Queries](#)
- [Upgrading Look & Feels](#)
- [Import Wizard Does Not Handle Some Legacy Jar Files](#)
- [Changes in Behavior Between Struts 1.1 and 1.2](#)
- [Portlet State Persistence](#)
- [WSRP Security Compatibility](#)
- [Working with Encoding in HTTP Responses](#)
- [Disconnected Desktop Requires desktopStateShared Property](#)
- [Correcting Duplicate Portlet Category Names in an Upgraded Application](#)
- [Propagation Utility Web Application Obsolete](#)
- [Definition Labels Not Editable in 10.3](#)
- [Propagation Inventory Compatibility](#)

Upgrading to the SQL Authenticator

The RDBMS Authenticator was supported in 8.1, but was deprecated in Portal 9.2 and all later releases. The RDBMS Authenticator was replaced by the SQL Authenticator.

To enable support in the Upgrade Wizard for upgrading a domain with an RDBMS Authenticator, a manual step is required. Perform the following workaround before you upgrade from an 8.1 RDBMS Authenticator to Portal 10.3:

1. Update your `setDomainEnv.cmd/sh` variable `weblogic.alternateTypesDirectory` to include the path to the deprecated provider:
`<WLPORTAL_HOME>/p13n/deprecated/lib/security.`
2. Run the Upgrade Wizard to perform the upgrade. The Upgrade Wizard also removes references to the deprecated RDBMS Authenticator from the domain's `config.xml` file.

Tip: If you did not upgrade your user store during the domain upgrade process, you can perform a manual upgrade later. Run the `upgrade_fromdbmsauth_tosqlauth.sql` script to upgrade from the WebLogic Portal-specific RDBMS Authenticator to the

WebLogic SQL Authenticator. The script is located in the
<WLPORTAL_HOME>\p13n\db\<DBMS>\ directory.

Upgrading Federated Portals from 8.1 to 10.3

New features added in WebLogic Portal 10.0 to support federated portal propagation require you to perform the upgrade procedures described in this section.

This section includes the following topics:

- [Overview](#)
- [Upgrading Producer and Consumer Applications](#)
- [Upgrading Only the Producer Applications](#)
- [Upgrading Only the Consumer Applications](#)
- [Listing Producer Handles](#)
- [Updating Producer Registration Handles](#)

Overview

WebLogic Portal 10.3 supports features of WSRP 2.0 that permit a more flexible and practical approach to propagation of federated portals. With WebLogic Portal 10.3, the consumer applications in staging and production environments can point to separate producers. The primary advantage of this new capability is that you can create and modify remote (proxy) portlets in a staging environment in isolation from the production environment.

Before WebLogic Portal 10.0, if you wanted to propagate WSRP consumer applications, the consumers on the source and destination systems had to point to the same producer. This configuration, which is described in detail in “[WSRP Propagation](#)” in the *WebLogic Portal 9.2 Production Operations Guide*, included several limitations.

This section explains the upgrade procedure for federated portals. The procedures described here apply to the following scenarios:

- [Upgrading Producer and Consumer Applications](#)
- [Upgrading Only the Producer Applications](#)
- [Upgrading Only the Consumer Applications](#)

Upgrading Producer and Consumer Applications

It is recommended that you upgrade both your consumer and producer applications to WebLogic Portal 10.0 in your source (staging) and destination (production) environments. Doing so allows you to take advantage of new propagation features and simplifies the propagation process.

1. Upgrade your consumer applications to WebLogic Portal 10.0. Perform the upgrade in both the source and destination environments.
 - a. If you ever performed a bidirectional propagation (propagated from staging to production and then back to staging again) or if propagation fails, you need to follow this sub-step. If neither of these conditions applies to you, skip this sub-step and proceed to Step 2.

Obtain the producer registration handles from each consumer application from both the source and destination systems. To do this, run the List Producers JSP utility as described in [“Listing Producer Handles” on page A-6](#).

2. Upgrade your producer application to WebLogic Portal 10.0. Perform the upgrade in both the staging and production environments.
 - a. If you ever performed a bidirectional propagation (propagated from staging to production and then back to staging again) or if propagation fails, you need to follow this sub-step. If neither of these conditions applies to you, skip this sub-step and proceed to Step 3.

Update the producer registration handles. To do this, run the Update Registration Handles JSP utility as described in [“Updating Producer Registration Handles” on page A-6](#).

3. Propagate the consumer application from the staging to the production environment using the propagation tools. See the [Production Operations Guide](#) for information on propagation.

Tip: When you propagate, you can adjust the scope to include only the Portal Framework resources.

This completes the upgrade process. It is now possible for consumer applications in staging and production environments to point to separate producers.

Note: If you want to retain a configuration where consumers in staging and production point to the same producer, you can do so; however, limitations described in [WSRP Propagation](#) in the *Portal 9.2 Production Operations Guide* no longer apply.

See the [Production Operations Guide](#) for detailed information on propagating portals.

Upgrading Only the Producer Applications

If you upgrade your domain and producer application but not the consumers, you need to follow the upgrade procedure described in this section. The procedure described in this section applies equally whether the consumer application(s) are currently at WebLogic Portal 8.1.x or 9.2.

Note: If you upgrade only the producer, you are required to use the propagation model described in [WSRP Propagation](#) in the *Portal 9.2 Production Operations Guide*. In this model, consumer applications in both staging and production environments must point to the same producer.

Note: If you upgrade only the producer, you must perform [step 2](#) and [step 3](#) below each time you propagate remote portlets in your consumer applications.

1. Upgrade your producer application to WebLogic Portal 10.0.
2. Obtain the producer registration handles from each consumer application on both the source and destination system. To do this, run the List Producers JSP utility as described in [“Listing Producer Handles”](#) on page A-6.
3. Update the producer registration handles for each upgraded producer application on the destination system. To do this, run the Update Registrations JSP utility as described in [“Updating Producer Registration Handles”](#) on page A-6.

See the *Production Operations Guide* for detailed information on propagating portals.

Upgrading Only the Consumer Applications

If you upgrade your consumer application(s) but not the producer(s), you need to follow the upgrade procedure described in this section. The procedure described in this section applies equally whether the producer application is currently at WebLogic Portal 8.1.x or 9.2.

Note: If you want to retain a configuration where consumers in staging and production point to the same producer, you can do so; however, limitations described in [WSRP Propagation](#) in the *WebLogic Portal 9.2 Production Operations Guide* will no longer apply.

Note: If you upgrade only the consumer(s), you are required to use the propagation model described in [WSRP Propagation](#) in the *WebLogic Portal 9.2 Production Operations Guide*. In this model, consumer applications in both staging and production environments must point to the same producer.

Note: If you upgrade only the consumers, steps 2 and 3 are recommended each time you propagate your consumer applications.

1. Upgrade your consumer applications to WebLogic Portal 10.0. Perform the upgrade in both the staging and production environments.

Note: If you are currently running a configuration where consumer applications in staging and production environments point to the same producer, the following steps are optional but recommended.

2. (Optional/Recommended) Obtain the producer registration handles from each consumer application on both the source and destination system. To do this, run the List Producers JSP utility as described in [“Listing Producer Handles” on page A-6](#).
3. (Optional/Recommended) Update the producer registration handles for each upgraded producer application on the destination system. To do this, run the Update Registrations JSP utility as described in [“Updating Producer Registration Handles” on page A-6](#).

Note: If you upgrade only the consumers, steps 2 and 3 are recommended each time you propagate your consumer applications.

Listing Producer Handles

You can run the List Producer JSP utility (`listProducers.jsp`) on both the staging and production systems on which your consumer applications are deployed. This utility obtains the registration handles for producers that have been previously added to your consumers.

See the [Production Operations Guide](#) for instructions.

Updating Producer Registration Handles

You can run the Update Registrations JSP utility (`updateRegistrations.jsp`) on both the staging and production systems. This utility updates the registration handles for each consumer application the currently references a given producer.

See the [Production Operations Guide](#) for instructions.

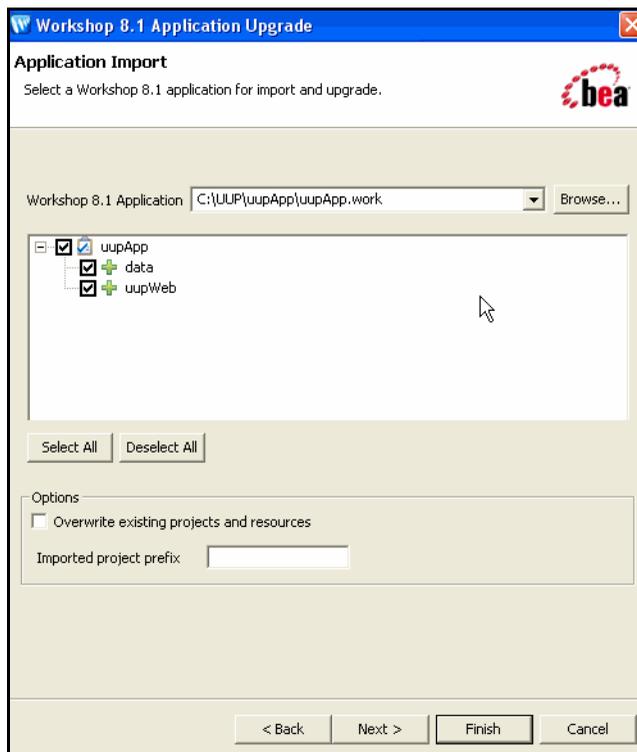
Upgrading UUPs

When you upgrade a Unified User Profile (UUP) from WebLogic Portal 8.1 to 10.3, the `p13n_ejb.jar` file is deleted and replaced with a new version of the WebLogic Portal 9.2 file. The new `p13n_ejb.jar` file is packaged in the library modules that ship with WebLogic Portal 9.2.

Perform the following steps to upgrade a UUP configured in WebLogic Portal 8.1 to WebLogic Portal 9.2:

1. Start Workshop for WebLogic and create a new Workspace.
2. Create a new portal domain. Do not create a Portal EAR Project. For instructions on creating a new domain, see the [Portal Development Guide](#).
3. Import your Portal 8.1 UUP application into your new environment by choosing **File > Import**.
4. In the Import dialog, open the **Other** folder, select **Workshop 8.1 Application**, and click **Next**.
5. In the Application Import dialog, click **Browse** and locate your 8.1 UUP application. Select the `.work` file and click **Open**. Verify that the check boxes for the UUP application are selected and click **Next**, as shown in [Figure A-1](#).

Figure A-1 Locate the 8.1 UUP Application



6. In the Source Upgrade dialog, click **NetUI Project Upgrader options** and select the **Use WebLogic J2EE Shared Libraries** check box. You can also click **JSP File Migrator**

options and select the **Replace Oracle NetUI tags with Apache Beehive tags** check box (if desired) and click **Finish**.

7. After the upgrade finishes, verify that the following actions occurred:
 - The `p13n-ejb.jar` file was removed from the `EARContent` directory of the UUP application.
 - The UUP EJB JAR file (for example, `UUPEXAMPLE.jar`) exists in the `EARContent` directory of the UUP application.
 - The UUP EJB JAR file is referenced in a module entry in the `application.xml` file in the `<UUPApplication>/EARContent/META-INF/` directory.
 - As an example, the cache entry below was added to the `p13n-cache-config.xml` file in the `<UUPApplication>/EARContent/META-INF/` directory:

```
<p13n:cache>
  <p13n:name>UUPEXAMPLECache</p13n:name>
  <p13n:description>Cache for UUP Example</p13n:description>
  <p13n:time-to-live>60000</p13n:time-to-live>
  <p13n:max-entries>100</p13n:max-entries>
</p13n:cache>
```
 - Verify that the User Profile file (for example, `UUPEXAMPLE.usr`) file exists in the `data/src/userprofiles/` directory (or where your Datasync folder exists).
8. Associate your portal application with your WebLogic Server by selecting the server in the **Servers** tab, right-clicking the server, and choosing **Add and Remove Projects**. Select the portal application from the **Available Projects** section, click **Add**, and then click **Finish**.
9. Build and publish your application. Verify the application by starting the WebLogic Server Administration Console and clicking **Deployments**. Verify that the UUP application is active. Then open the UUP application by expanding the tree and verifying that the UUP JAR file appears as an EJB.

Understanding JSP Tag Changes

Two JSP tags, `<ugm:login>` and `<ugm:logout>`, are deprecated in WebLogic Portal 9.2 and 10.0. The `<ugm:login>` and `<ugm:logout>` JSP tags were moved from the `ugm_taglib.jar` file to the `auth_taglib.jar` file.

After you upgrade to 10.3, you should use the `<auth:login>` and `<auth:logout>` JSP tags. The attributes and parameters are the same.

Upgrading Autonomy Search Services

If you are upgrading from WebLogic Portal 8.1 and used Autonomy search, you need to do the following to upgrade to the new version of Autonomy that is included with WebLogic Portal 10.3. The upgrade process does not automatically migrate your existing search settings and databases to the new search capabilities. You need to manually configure Autonomy search to work with your upgraded applications.

The new version of Autonomy is installed into the `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10` directory. For more information about configuring Autonomy search, see the [Integrating Search Guide](#).

After installing and configuring Autonomy search, do the following to upgrade your application to use the new features:

1. Modify your WebLogic Portal 10.3 configuration files to match any Autonomy customizations you used. [Table A-1](#) lists the files should be modified. If other configuration files were used that reference Autonomy search tools, you need to modify those as well.
2. Edit any start scripts that may start the 8.1 version of Autonomy services to ensure that they are not restarted. These are stopped automatically when you upgrade. For instructions on editing Autonomy start scripts, see the [Integrating Search Guide](#).

Table A-1 Mapping of WebLogic Portal 8.1 Configuration Files to WebLogic Portal 10.3 Configuration Files

| File used with WebLogic Portal 8.1 | File used with WebLogic Portal 10.3 |
|------------------------------------------|-------------------------------------|
| <code>PortalSearchDRE.cfg</code> | <code>AutonomyIDOLServer.cfg</code> |
| <code>PortalSearchDiSH.cfg</code> | <code>AutonomyDiSH.cfg</code> |
| <code>PortalSearchHTTPFetch.cfg</code> | <code>HTTPFetch.cfg</code> |
| <code>PortalSearchAutoIndexer.cfg</code> | <code>FileSystemFetch.cfg</code> |

Using 8.1 Search within a 10.0 Portal Application

If you wrote applications using the WebLogic Portal `com.bea.query` classes or using the 8.1 Autonomy API and want to continue using these applications without using the new 9.2 version of the Autonomy API, you need to manually add the older, deprecated APIs to your application.

However, if you continue to use the 8.1 API (either WebLogic Portal or Autonomy's), those APIs will take priority over the 10.0 API and are NOT compatible with the 10.0 Autonomy engine. This means that some 10.0 content management features will not work, such as full-text search. If you want to continue to use the 8.1 version of Autonomy with your portal application, you must manually add the older APIs to your installation.

To add the older Autonomy version to your 10.0 application:

1. Upgrade your domain and application to 10.0.
2. Copy `autonomyCompat.jar` and `autonomyClient.1.5.0.jar` from `<WLPORAL_HOME>/info-mgmt/lib/thirdparty/search/81-compat-only` into the `application_home/APP-INF/lib` directory.
3. Using the Portal Administration Console, turn off full-text search for your repository. For more information, see the *Content Management Guide*.
4. Modify your domain start script to ensure that the 8.1 versions of the Autonomy are started rather than the new versions. For more information about start scripts, see the [WebLogic Server documentation](#).
5. Start your domain.
6. Verify that search functionality is working.
 - a. Index some content into Autonomy. For example, use `FileSystemFetch`.
 - b. Execute a search in the Portal Search portlet.
 - c. Verify that results are returned and no exceptions are encountered.
 - d. Add content to the WLP Repository instance using the content management tools.
 - e. Ensure that no exceptions are displayed; this would only occur if attempting to index the content, which will not occur if the preceding steps have been successfully executed.

The 8.1 Autonomy APIs and engines are now running.

Upgrading to Use 10.0 Search after Using 8.1 Search with a 10.0 Portal Application

If you want to upgrade to use the 10.0 version of Autonomy, after you have completed the steps in [“Using 8.1 Search within a 10.0 Portal Application”](#) on page A-9, you can reverse the implementation.

To upgrade to use the 10.0 version of Autonomy (after you have implemented 8.1 Autonomy with 9.2):

1. Locate and update any usages of the `com.bea.query.*` classes or any calls to the Autonomy client APIs in your applications and replace them with the appropriate calls to the 10.0 version of the Autonomy API.
2. Remove the `autonomyCompat.jar` and `autonomyClient1.5.0.jar` file from the `app-inf/lib` directory.
3. Modify your domain start script to execute the 10.0 version of the Autonomy engines.
4. Configure the 10.0 Autonomy engines to index your content. For more information, see the [Integrating Search Guide](#).
5. Using the Portal Administration Console, turn on full-text search for your repository. For more information, see the [Content Management Guide](#).
6. Run the startup script.

Manually Upgrading Passwords in Content Management Repositories

After the upgrade is complete, you must manually re-enter the passwords for your third-party repositories using the Administration Console; see the [Content Management Guide](#) for more information about editing repository settings.

Until you manually re-enter the passwords for your third-party repositories, you cannot access those repositories.

Maintaining Content Queries

In WebLogic 8.1 through WebLogic Portal 8.1 SP5, content query expressions were generated differently due to an order of precedence problem. The order of precedence was not maintained when executing a content query expression. For example, the following expression: `(a && (b || c))`, gets evaluated/executed as `(a && b || c)`.

This problem was fixed in Weblogic Portal 10.0 such that the order of precedence is now maintained when executing a content query expression. However, if you want to continue using the WebLogic Portal 8.1 through WebLogic Portal SP5 query behavior, you need to modify your domain scripts to define the following system property:

```
-Dwlp.disable.content.rule.fix=true.
```

Upgrading Look & Feels

Portal Look & Feels in WebLogic Portal 8.1 used two configuration files for skins and skeletons (in the `/skins/<skin_name>` and `/skeletons/<skeleton_name>` directories): `skin.properties` and `skeleton.properties`. Both were text files, and `skeleton.properties` was optional.

In WebLogic Portal 10.0, both files are XML, and both are required.

To upgrade a WebLogic Portal 8.1 Look and Feel to the WebLogic Portal 10.3 format:

1. Verify that the portal application that contains the Look and Feel has been converted to WebLogic Portal 10.0, as described in the [Portal Development Guide](#).
2. Open the Look & Feel in Workshop for WebLogic and re-save it. The configuration files are automatically converted to the new XML format.

Import Wizard Does Not Handle Some Legacy Jar Files

The `cm_taglib.jar` and the `pz_compat_taglib.jar` are deleted when you upgrade and Import Wizard flags all JSPs that refer to these taglibs, which have an unsupported taglib URIs. The JSPs will fail.

The `cm_taglib.jar` file was not installed by default in a new 8.1 web application, but if you added it to your application for backward compatibility, you must handle this file manually in your upgraded application.

Change all references to the `cm_taglib.jar` and the `pz_compat_taglib.jar` so that they use supported tags and APIs, and delete the obsolete jar files.

Changes in Behavior Between Struts 1.1 and 1.2

WebLogic Portal support for Struts is slightly different in 10.3 if you upgrade to Struts 1.2.

Struts 1.1 support in WebLogic Portal is the same as in previous releases, with the `struts-adapter` taglibs mapped to URIs using `web.xml`. You can use the `struts-1.1.war` library module instead of the new `struts-1.2.war` library module.

If you are upgrading to Struts 1.2, instead of mapping the `struts` and `struts-adapter` taglibs using `web.xml`, WebLogic Portal now relies on the JSP 1.2 implicit taglib mapping, wherein any `.tld` files in the `META-INF` directory in a JAR are implicitly mapped by the web container to the URI specified in the `tld`. For WebLogic Portal, these are in `struts-adapter.jar`, in the path `META-INF/tlds`.

Choose to use one of these two methods to upgrade to Struts 1.2:

- Modify all JSPs that use the struts taglibs to reference `http://bea.com/struts/adapter/tags-html` and `http://bea.com/struts/adapter/tags-nested` for the HTML and nested taglibs, and `http://struts.apache.org/tags-*` for the remainder of the taglibs that our adapter does not override.
- Extract the `.tlds` from both `struts.jar` (in `struts-1.1.war`) and from `struts-adapter.jar` (in our portal web library module) and copy them to `WEB-INF/tlds`. This allows for the case where you want to continue using the explicit tld mapping via `web.xml`.

Portlet State Persistence

In WebLogic Portal 8.1, minimized portlet states were persisted only for the session. You can use a workaround, described in the [Upgrade Guide for Version 8.1](#), to set up a backing file that controls the states of portlets under the desktop.

The solution used in 8.1 will continue to work if you depend on this behavior. See the [9.2 Portlet Guide](#) for instructions and an example.

WSRP Security Compatibility

Producer and consumer applications developed with WebLogic Portal 10.3 are compatible with producers and consumers developed with WebLogic Portal 8.1. That is, a portal developed with WebLogic Portal 10.0 can consume portlets deployed in a WebLogic Portal 8.1 domain. Similarly, portlets exposed in a WebLogic Portal 10.3 producer can be consumed by an 8.1 consumer.

However, if you want to use your own key for the 8.1 or 9.2 consumer, you need to follow the procedures outlined the chapter “[Establishing WSRP Security with SAML](#)” in the [Federated Portals Guide](#).

Working with Encoding in HTTP Responses

This section describes changes in how the encoding is set on the HTTP response.

Setting Encoding for 8.1

In WebLogic Portal 8.1, the following methods were used to set the encoding:

1. Examine the `.portal` file for a `directive.page` element. If that element is present, obtain the encoding from an attribute there.
2. If the element is not present, use the JSP encoding configuration, which looks at the `<encoding>` element in the `<jsp-param>` section of the `web.xml` file; the default is ISO-8859-1 if these elements are missing.

Both of these mechanisms are now deprecated.

Setting Encoding for 10.3

See the [10.3 Portal Development Guide](#) for instructions on setting encoding and editing encoding in Workshop for WebLogic.

Disconnected Desktop Requires `desktopStateShared` Property

For compatibility with 8.1 and prior releases, and to support the few desirable uses of a disconnected desktop, WebLogic Portal 10.3 provides a new, but deprecated, boolean property called `desktopStateShared` for the `StandalonePortletURL` and the associated JSP tag. You can use this property to retain the previous “disconnected desktop” behavior. The default value of this property and attribute will be `true`, causing the portlet to be connected to a desktop.

Explicitly setting either the `path` or `contextualPath` properties on the URL or tag also disassociates the resulting standalone portlet from its originating desktop. This also holds true for any URLs generated within the context of the standalone portlet—setting `path` or `contextualPath` causes the resulting URLs to become disassociated with the originating desktop.

Correcting Duplicate Portlet Category Names in an Upgraded Application

In 8.1 and previous releases of WebLogic Portal it was possible, though not recommended, to create more than one portlet category with the same name, at the same level in the hierarchy. In 10.3, this operation is not permitted. (You can use the same name for more than one category, but they must not be “peers” in the hierarchy.)

When you upgrade a portal application to 10.3, any duplicate portlet category names that were used previously are preserved. It is extremely important that you edit these category names to be unique; otherwise the WebLogic Portal propagation tools might cause unexpected results, or errors might occur during the propagation process.

Propagation Utility Web Application Obsolete

The Propagation Utility web application, `propagation.war`, became obsolete as of WebLogic Portal 9.2. This application was introduced in a patch for WebLogic Portal 8.1 SP4 and later incorporated into WebLogic Portal 8.1 SP5. If you are upgrading a WebLogic Portal web application in which you previously installed the file `propagation.war` into the root directory of any WebLogic Portal enterprise applications, it is recommended that you remove the file before or after upgrading to WebLogic Portal 10.3.

Definition Labels Not Editable in 10.3

In WebLogic Portal 8.1, the capability to edit definition labels existed, but was not recommended. Modifying the definition label could have unintended implications; for example, exposing a protected resource or breaking WSRP (which uses the definition label as the portlet handle).

As of WebLogic Portal 9.2, this functionality has been replaced by a much richer ability to move portal resources (books, pages, desktops) between production and development environments, without losing user customizations or changing labels. These new features include XIP and the propagation utility. XIP allows you to target individual portal resources to import and export between development and production systems, and you can specify the scope (library, admin, or visitor). For more information about WebLogic Portal's propagation tools, see the [Production Operations Guide](#).

Propagation Inventory Compatibility

WebLogic Portal inventories saved with WebLogic Portal 8.1 or 9.2 cannot be used with WebLogic Portal 10.0 propagation tools.

Functional Changes from Portal 9.2 to 10.2/10.3

The following sections describe changes that occur when you upgrade from WebLogic Portal 9.2 or 9.2 MP1 to 10.2/10.3. You might be required to perform manual tasks.

This section includes the following topics:

- [Upgrading Federated Portals](#)
- [Upgrading UUPs](#)
- [Disconnected Desktop Requires desktopStateShared Property](#)
- [Upgrading Search Services](#)

- [WSRP Security Compatibility](#)
- [Upgrading Propagation Scripts](#)
- [Propagation Inventory Compatibility](#)

Upgrading Federated Portals

The procedure for upgrading federated portals from 9.2 to 10.3 is identical to the procedure for updating from 8.1 to 10.3. See [“Upgrading Federated Portals from 8.1 to 10.3”](#) on page A-3 for detailed instructions.

Upgrading UUPs

Your WebLogic Portal 9.2 or 9.2 MP1 UUP automatically works in WebLogic Portal 10.3. You do not need to upgrade your 9.2 UUP.

Disconnected Desktop Requires desktopStateShared Property

For backward compatibility to WebLogic Portal 8.1 and previous releases, and to support the few desirable uses of a disconnected desktop, WebLogic Portal 9.2 provided a new, but deprecated, boolean property called `desktopStateShared` for the `StandalonePortletURL` and the associated JSP tag. This property is retained in 10.3 and is still deprecated.

For more details, see [Disconnected Desktop Requires desktopStateShared Property](#).

Upgrading Search Services

Upgrading search services involves updating your Autonomy configuration files and re-indexing your content.

Optionally, you can choose to migrate your existing Autonomy installation to use the 10.0 Autonomy installation directory. WebLogic Portal 10.3 ships with the same version of Autonomy as was shipped with WebLogic Portal 9.2, but the installation directory has changed. The new installation directory is: `<WLPORTAL_HOME>\content-mgmt\thirdparty\autonomy-wlp10`.

For production environments, you should continue to use the 10.3 installation directory (`<WLPORTAL_HOME>\content-mgmt\thirdparty\autonomy-wlp10`).

For development environments, Oracle recommends discontinuing the use of the previous installation directory and using the new installation directory.

New multiple language search, query, and indexing capabilities have been added. The default 10.3 configuration supports the 9.2 configuration without alteration. For more information about the new capabilities, see [Multi-language Searching and Indexing](#) in *Integrating Search*.

Note: You **MUST** re-index your WLP content within your production environment; see the [Integrating Search Guide](#) for detailed instructions.

Repository Configuration Settings for Using CM Full-Text Search

In WebLogic Portal 9.2, CM full-text search connects to Autonomy during startup if the repository configuration has both of the following settings:

- `search-is-enabled=TRUE`
- `search-indexing-is-enabled=TRUE`

In WebLogic Portal 10.3, the settings for connections require:

- `fulltext-search-is-enabled=TRUE`
- `search-indexing-is-enabled=TRUE`

The 10.3 settings more accurately reflect that CM full-text search is being used for a repository.

Because a WLP 9.2 full-text search application requires `search-indexing-is-enabled=TRUE` to use CM full-text search indexing, this setting already applies. However, you should verify the 10.3 repository configuration and update if necessary.

Update Autonomy Configuration Files

Update all of necessary Autonomy configuration files to use the new location, see [Configuring Autonomy on Your Target Server](#) in the *Integrating Search Guide* for more details on Autonomy configuration files.

Supporting Double Quotes in WLP Content Search Queries

To continue to support double-quotes in your search queries, you must make the following changes to your BEACMRepoFetch.cfg file.

To support double-quotes (for example, “quotes”):

1. On the Autonomy host, edit the file at:

```
//autonomyhome/operatingsystem/internal/BEACMRepoFetch/BEACMRepoFetch.cfg
```

For example, on a Linux host, this might be at:

```
<WLPORAL_HOME>/content-mgmt/thirdparty/autonomy/linux-intel/internal/
```

2. In the [BEACMRpoIDXImport] section, add the line:

```
ImportBifIncludeQuotes=true
```

Supporting Japanese Characters in your WLP Content Queries

To continue support of querying Japanese characters, you must make the following modifications to your `AutonomyIDOLServer.cfg` file:

1. In the [FieldProcessing] section, do the following:
 - a. Increment the [FieldProcessing] Number field value by one.
 - b. Add a line of the form `19=SetLanguage` using one number greater than the largest number listed in the [FieldProcessing] section.

For example, change from this:

```
[FieldProcessing]
Number=19
0=SetIndexFields
...
18=SetMoreReferenceFields
```

To this:

```
[FieldProcessing]
Number=20
0=SetIndexFields
...
18=SetMoreReferenceFields
19=SetLanguage
```

2. Before the [Properties] section, add the following section:

```
[SetLanguage]
Property=LanguageFields
PropertyFieldCSVs=*/DRELANGUAGETYPE
```

3. In the [Properties] section, add a line of the form `19=LanguageFields` using one number greater than the largest number listed in the [Properties] section.

For example, change from this:

```
[Properties]
0=IndexFields
...
18=MoreReferenceFields
```

To this:

```
[Properties]
0=IndexFields
...
18=MoreReferenceFields
19=LanguageFields
```

4. Before the `[Security]` section, add the following section:

```
[LanguageFields]
LanguageType=TRUE
```

5. Save your changes.

Note: Additional Japanese language and encoding functionality has been added to the 10.3 installation for searching, queries, and indexing. For more information about the new capabilities, see [Multi-language Searching and Indexing](#) in *Integrating Search*.

Migrating your Autonomy Installation to the 10.3 Directory

Optionally, you can migrate your 9.2 Autonomy configuration to the 10.3 installation.

Note: For upgraded production environments, Oracle recommends continuing to use the existing 9.2 Autonomy location.

To migrate your 9.2 Autonomy configuration and data to the new 10.3 location:

1. [Copy Required Files to the New Location.](#)
2. [Export Indexed Data from the 9.2 IDOL Location.](#)
3. [Import 9.2 Indexed Data to the 10.3 IDOL Location.](#)
4. Re-index all WLP content; see the *Integrating Search Guide* for detailed instructions.

Copy Required Files to the New Location

On the IDOL server, copy the following folders from the 9.2 location to the 10.3 location.

1. From the `\\wlp-92\IDOLserver\IDOL\content\` folder, copy the following subfolders to the new content folder in the `\\wlp-10\IDOLserver\IDOL\content\` location, and copy the following subfolders to the new content `\\wlp-102\IDOLserver\IDOL\content\` location:

- `\dynterm folder\`
- `\main`
- `\nodetable`
- `\numeric`
- `\refindex`
- `\sortfield`
- `\status`
- `\storedstate`
- `\tagindex`

2. From the Portal 9.2 `\\wlp-92\IDOLserver\IDOL\indextasks\` folder, copy the following subfolders to the new Portal 10.3 `\\wlp-103\IDOLserver\IDOL\indextasks\` folder:

- `\failed`
- `\incoming`
- `\main`
- `\outgoing`
- `\queue`
- `\temp`

3. From the Portal 9.2 `\\wlp-92\IDOLserver\IDOL\agentstore\` folder, copy the following subfolders to the new Portal 10.3 `\\wlp-103\IDOLserver\IDOL\agentstore\` folder:

- `\dynterm folder`
- `\main`
- `\nodetable`
- `\numeric`
- `\refindex`
- `\sortfield`

- \status
 - \storedstate
 - \tagindex
4. From the Portal 9.2 \\wlp-92\IDOLserver\IDOL\category\ folder, copy the following subfolders to the new Portal 10.3 \\wlp-103\IDOLserver\IDOL\category\ folder:
 - \cluster
 - \imex
 - \taxonomy
 - \category
 - \failed
 - \outgoing
 5. From the Portal 9.2 \\wlp-92\IDOLserver\IDOL\community\ folder, copy the following subfolders to the new Portal 10.3 \\wlp-103\IDOLserver\IDOL\agentstore\ folder in the 10.0 location:
 - \users
 - \temp
 - \queue

Export Indexed Data from the 9.2 IDOL Location

You need to export all indexed data from your 9.2 IDOL Server and re-import it into the 10.3 location.

Use the following command from your browser to export indexed content. Use the correct host name and port for your Autonomy configuration. You also need to indicate a file name, including a directory, to which to index content.

`http://host:port/DREEXPORTIDX?FileName<filename>&Compress<>true\false>`

Import 9.2 Indexed Data to the 10.3 IDOL Location

After exporting your indexed data from the 9.2 location, use the following command to import the content to the new WebLogic Portal 10.3 location. Use the correct host name and port for your Autonomy configuration and the file you created when you exported your data.

`http://host:port/DREADD?<filename>`

Re-Index WLP Content

After completing the above sections, you must re-index your WLP content. For detailed instructions on re-indexing WLP content, see the [Integrating Search Guide](#).

WSRP Security Compatibility

Producer and consumer applications developed with WebLogic Portal 10.3 are compatible with producers and consumers developed with WebLogic Portal 9.2. That is, a portal developed with WebLogic Portal 10.3 can consume portlets deployed in a WebLogic Portal 9.2 domain. Similarly, portlets exposed in a WebLogic Portal 10.3 producer can be consumed by a 9.2 consumer.

However, if you want to use your own key for the 9.2, 10.0, 10.2, or 10.3 consumer, you need to follow the procedures outlined in the chapter “Establishing WSRP Security with SAML” in the [Federated Portals Guide](#).

Upgrading Propagation Scripts

If you created any propagation Ant scripts in 9.2, you must either manually update them or regenerate the scripts in 10.0. This change is required because several JAR file names have changed and the package name for all propagation Ant tasks has changed.

If you regenerate your scripts using Workshop for WebLogic, the correct filenames and package name will be used automatically. If you must update your scripts manually (for instance, if you created them manually in the first place), you need to make the following changes:

The propagation ant tasks are now located in the package `com.bea.propagation.ant.taskdefs`.

The following table lists the 9.2 JAR file names and the updated names for 10.0:

| 9.2 JAR Name | 10.0 JAR Name |
|-----------------------------------|-------------------------------------|
| <code>p13n_prop.jar</code> | <code>propagation.jar</code> |
| <code>p13n_prop_online.jar</code> | <code>propagation_online.jar</code> |
| <code>p13n_prop_ant.jar</code> | <code>propagation_ant.jar</code> |

Propagation Inventory Compatibility

WebLogic Portal inventories saved with WebLogic Portal 8.1 or 9.2 cannot be used with WebLogic Portal 10.3 propagation tools.

Functional Changes Affecting Your WebLogic Portal Environment

Performing Database Upgrade Tasks Manually

This appendix describes how to perform database upgrade tasks manually if you do not use the WebLogic Upgrade Wizard to upgrade from 8.1, 9.2, or 10.0/10.2 to WebLogic Portal 10.3.

See the `README.txt` file and the `upgrade_db.properties` file in the `<WLPORTAL_HOME>\common\lib\upgrade\db` directory for detailed information about the files used in the upgrade and specific upgrade steps.

This appendix contains the following sections:

- [Upgrading your Main WebLogic Portal 8.1, 9.2, or 10.0 Database](#)
- [Upgrading 8.1, 9.2, or 10.0 PointBase Databases](#)
- [Upgrading to the 10.3 WebLogic Server SQL Authenticator](#)
- [Upgrading Separate 8.1 Behavior Tracking Databases](#)
- [Upgrading Additional 8.1, 9.2, or 10.0 Content Management Databases](#)
- [Upgrading a 9.2 GroupSpace Database](#)
- [Dropping Deprecated RDBMS Authenticator Tables After Upgrade](#)
- [Dropping Deprecated Compoze Database Tables After Upgrade](#)

Upgrading your Main WebLogic Portal 8.1, 9.2, or 10.0 Database

If you did not upgrade your main WebLogic Portal database from 8.1, 9.2, or 10.0/10.2 to 10.3 using the WebLogic Upgrade Wizard, you can perform the upgrade manually. If you have a 9.2 GroupSpace database, you must also upgrade that database. See [“Upgrading a 9.2 GroupSpace Database” on page B-5](#) for more information.

Perform the following steps to upgrade your WebLogic Portal database:

1. Shut down WebLogic Server.
2. Back up your database data as described by your database vendor.
3. Edit the `<WLPORTAL_HOME>\common\lib\upgrade\db\upgrade_db.properties` file for your database environment. Replace the @ symbols and the text between the symbols with the correct values for `@DB_USER@`, `@DB_PASSWORD@`, `@DB_HOST@`, `@DB_PORT@`, and `@DB_NAME@`.

Note: For PointBase, follow the instructions in [“Upgrading 8.1, 9.2, or 10.0 PointBase Databases” on page B-3](#).

4. In the `upgrade_db.properties` file, modify and uncomment the `files=` setting for your portal database. Follow the instructions in the `upgrade_db.properties` file for your specific type of upgrade (9.2 to 10.3 or 8.1 to 10.3.)
5. Run the following script: `upgrade_db.cmd/.sh`.
6. If you are upgrading from 8.1 to 10.3, run the `upgrade_db_data.cmd/.sh` script to upgrade database data.

Tip: To upgrade your user store, follow the instructions in [“Upgrading to the 10.3 WebLogic Server SQL Authenticator” on page B-3](#).

7. If you are upgrading a main WebLogic Portal database from 8.1 SP4, SP5, or SP6, determine if patch # CR244936 was applied. If this patch was applied, manual upgrade of the main WebLogic Portal database is complete. This patch is described in WebLogic Portal 8.1 SP5 Release Notes (<http://edocs.beasys.com/wlp/docs81/relnotes/relnotes.html#1117746>) as follows:

CR237251 – If markup contains more than 4000 bytes, an attempt to store it in the database causes an error. If your database does not contain a PF_MARKUP_XML table that has been populated with data (for example, Select count(*) from PF_MARKUP_XML returns 0 rows) and the PF_MARKUP_DEFINITION table does not contain the BEGIN_XML and END_XML columns, then this patch was not applied.

8. If you are upgrading a main WebLogic Portal database from 8.1 SP4, SP5, or SP6 and patch # CR244936 has not been applied, run the following script against your database:

```
<WLPORTAL_HOME>\portal\db\<DBMS>\pf9_drop_columns.sql
```

Upgrading 8.1, 9.2, or 10.0 PointBase Databases

To manually upgrade your PointBase database from 8.1, 9.2, or 10.0, you must first run the WebLogic Upgrade Wizard to perform a domain upgrade, and select **No** to skip the database upgrade. You must also copy the database files themselves (weblogic_eval.dbn and weblogic_eval\$.wal) into the <WLPORTAL_HOME>\common\upgrade\db directory and then copy them back to the domain directory.

See the README.txt file for the specific steps to manually upgrade your PointBase database.

Upgrading to the 10.3 WebLogic Server SQL Authenticator

The RDBMS Authenticator was supported in 8.1, but was deprecated in Portal 9.2 and all later releases. The RDBMS Authenticator was replaced by the SQL Authenticator.

To enable support in the Upgrade Wizard for upgrading a domain with an RDBMS Authenticator, a manual step is required. Perform the following workaround before you upgrade from an 8.1 RDBMS Authenticator to Portal 10.3:

1. Update your setDomainEnv.cmd/sh variable weblogic.alternateTypesDirectory to include the path to the deprecated provider:


```
<WLPORTAL_HOME>/p13n/deprecated/lib/security.
```
2. Run the Upgrade Wizard to perform the upgrade. The Upgrade Wizard also removes references to the deprecated RDBMS Authenticator from the domain's config.xml file.

Tip: If you did not upgrade your user store during the domain upgrade process, you can perform a manual upgrade later. Use the following script to upgrade from the WebLogic Portal-specific RDBMS Authenticator to the WebLogic SQL Authenticator:

```
<WLPORTAL_HOME>\p13n\db\<DBMS>\ upgrade_fromdbmsauth_tosqlauth.sql.
```

Upgrading Separate 8.1 Behavior Tracking Databases

If you created a separate behavior tracking database in 8.1, you can upgrade it manually.

Perform the following steps to upgrade a separate behavior tracking database:

1. Shut down WebLogic Server.
2. Back up your database data as described by your database vendor.
3. Edit the settings in the `upgrade_db.properties` file for your behavior tracking database. Replace the @ symbols and the text between the symbols with the correct values for @DB_USER@, @DB_PASSWORD@, @DB_HOST@, @DB_PORT@, and @DB_NAME@.
4. In the `upgrade_db.properties` file, modify and uncomment the `files=` setting for your behavior tracking database. Follow the instructions in the `upgrade_db.properties` file.
5. Run the following script: `upgrade_db.cmd/.sh`.

Upgrading Additional 8.1, 9.2, or 10.0 Content Management Databases

The default content management database is upgraded automatically. If you created an additional content management database in 8.1, 9.2, or 10.0, you can upgrade it manually.

Perform the following steps to upgrade an additional Content Management database:

1. Shut down WebLogic Server.
2. Back up your database data as described by your database vendor.
3. Update the settings in the `upgrade_db.properties` file for your content management database. Replace the @ symbols and the text between the symbols with the correct values for @DB_USER@, @DB_PASSWORD@, @DB_HOST@, @DB_PORT@, and @DB_NAME@.
4. In the `upgrade_db.properties` file, modify and uncomment the `files=` setting for your content management database. Follow the instructions in the `upgrade_db.properties` file.

5. Run the following script: `upgrade_db.cmd / .sh`.

Upgrading a 9.2 GroupSpace Database

If you did not use the WebLogic Upgrade Wizard to upgrade your WebLogic Portal 9.2 GroupSpace repository database, you can manually upgrade it.

Perform the following steps to upgrade a GroupSpace database:

1. Shut down WebLogic Server.
2. Back up your database data as described by your database vendor.
3. In the `upgrade_db.properties` file, modify and uncomment the `files=` setting for your GroupSpace database. Follow the instructions in the `upgrade_db.properties` file.
4. Run the following script: `upgrade_db.cmd / .sh`.

Dropping Deprecated RDBMS Authenticator Tables After Upgrade

After you upgrade to the WebLogic Server SQL Authenticator, you can drop the tables associated with the WebLogic Portal RDBMS Authenticator using the following script:

```
<WLPORTAL_HOME>\p13n\db\<DMBS>\dep9_drop_tables.sql
```

Dropping Deprecated Compoze Database Tables After Upgrade

After you upgrade to WebLogic Portal 10.3, you can drop the tables associated with Compoze/Collaboration using the following script:

```
<WLPORTAL_HOME>\portal\db\<DMBS>\dep9_drop_tables.sql
```

Performing Database Upgrade Tasks Manually

Upgrading WebLogic Portal 8.1 Projects to WebLogic Portal 10.3

Before you attempt to upgrade a WebLogic Portal 8.1 SP4+ application to WebLogic Portal 10.3, you need to become familiar with the Workshop for WebLogic upgrade steps, and any related limitations. For complete information about upgrading your portal application in Workshop for WebLogic, see the [Workshop for WebLogic for WebLogic upgrade](#).

The Workshop for WebLogic upgrade documentation assumes that your application was developed using WebLogic Workshop for WebLogic 8.1 SP4+. If it was not, you must refactor your code so that it builds and runs in the WebLogic Workshop for WebLogic IDE 8.1 SP4+ before using the tools described here to upgrade to Workshop for WebLogic 10.3.

This appendix focuses on topics that are specifically related to upgrading WebLogic Portal applications, and contains the following sections:

- [WebLogic Portal 8.1 Features Not Supported in WebLogic Portal 10.3](#)
- [Upgrade Considerations and Tips](#)

WebLogic Portal 8.1 Features Not Supported in WebLogic Portal 10.3

Webflows and pipelines were deprecated in WebLogic Portal 9.2 and are no longer supported; use page flows in place of these deprecated features.

Upgrade Considerations and Tips

The following sections describe some considerations and tips that might be useful when you upgrade portal applications, and it describes some situations where you might need to perform some manual tasks after upgrading your portal application.

This section contains the following topics:

- [Command-Based and Ant Task Upgrade Not Supported](#)
- [Enabling Communities Features in Upgraded Visitor Tools](#)
- [Upgrading Look And Feels](#)
- [Upgrading Custom Controls that Have Custom Properties](#)
- [Upgrading Tuned Thread Pools for Forked Portlets](#)
- [Upgraded Applications Use Struts 1.1 and Related J2EE Libraries](#)
- [Changes in Behavior Between Struts 1.1 and 1.2](#)
- [Ampersand Entities in Portal URLs](#)
- [Upgrading Individual application-config.xml Files Later](#)
- [Correcting Duplicate Portlet Category Names Before Propagating an Upgraded Application](#)

Command-Based and Ant Task Upgrade Not Supported

Workshop for WebLogic provides a command-driven upgrade (upgradeStarter) and an ant task-based upgrade. WebLogic Portal does not support these upgrade alternatives; you must use the Import Wizard.

Enabling Communities Features in Upgraded Visitor Tools

Unmodified Visitor Tools code from WebLogic Portal 8.1.4+ is upgraded as part of Import Wizard processing. However, the WebLogic Portal 10.3 communities-based components are disabled by default, because projects developed with 8.1.4+ are not communities-aware.

As part of the upgrade process, the Import Wizard creates a `communities-config.xml` file in the EAR project `/META-INF` directory that defines whether or not the communities-related Visitor Tools are enabled. To enable this functionality, set the flag in the `enable-community-tools` attribute to `true`, as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<communities-config
xmlns="http://www.bea.com/ns/portal/100/communities-config">
  <enable-community-tools>true</enable-community-tools>
</communities-config>
```

Upgrading Look And Feels

Portal Look And Feels in WebLogic Portal 8.1.4+ used two configuration files for skins and skeletons (in the `/skins/skin_name` and `/skeletons/skeleton_name` directories): `skin.properties` and `skeleton.properties`. Both were text files, and `skeleton.properties` was optional.

In WebLogic Portal 10.3, both files are now XML.

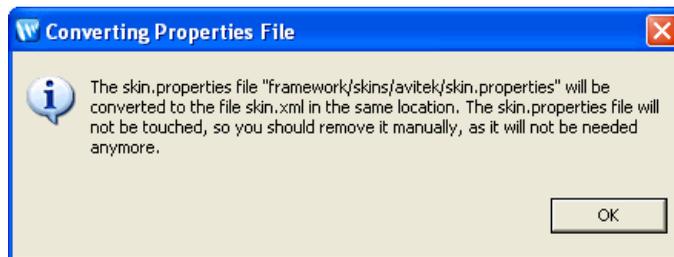
For WebLogic Portal 10.3, it is necessary to use `skin.xml` and `skeleton.xml` to take advantage of new Look And Feel features, although you can still use legacy configurations. Upgrading your old `.properties` files is required.

To upgrade a WebLogic Portal 8.1 Look And Feel to the WebLogic Portal 10.3 format:

1. Make sure the portal application containing the Look And Feel has been converted to WebLogic Portal 10.3, as described in this chapter and in the Workshop for WebLogic documentation.
2. Open the Look And Feel file (`.laf` file).

WebLogic Portal automatically upgrades the associated `.properties` file; informational messages describe the change that occurs during upgrade. [Figure C-1](#) shows an example:

Figure C-1 Conversion of Properties File when Look And Feel File is Opened



3. Click **OK** to complete the upgrade.

Upgrading Custom Controls that Have Custom Properties

WebLogic Portal 8.1 custom control annotation definitions are not upgraded to WebLogic Portal 10.3. The means for defining annotations is based on the Java 5 annotations model. To upgrade controls written for 8.1, you must rewrite the annotations definition in keeping with the new model.

For more information on upgrading your custom annotations, take a look at the Apache Beehive source code for its system controls. These provide annotations that use the new model.

For information on how the control context APIs have changed from WebLogic 8.1, see “Handling Context API Changes” in the Workshop for WebLogic online help.

Upgrading Tuned Thread Pools for Forked Portlets

If you optimized the thread pool used for an 8.1 application that uses forked rendering or forked pre-rendering, and you want to keep those optimizations in the upgraded application, you will need to perform some manual tasks after upgrading.

WebLogic Portal 10.3 uses WebLogic Server’s CommonJ WorkManager infrastructure for forked portlet pre-render and render. WorkManagers have similar but not identical configuration parameters, behavior, and deployment options. When you upgrade an 8.1.4+ application, any existing customizations to the portalRenderQueue thread pool will not be automatically applied to the default WorkManager used for forking. To tune this WorkManager, configure a WorkManager and associate it with the name `wm/portalRenderQueueWorkManager`. For more information about WorkManagers and thread usage in WebLogic Server 10.0, refer to *“Using Work Managers to Optimize Scheduled Work”* in the WebLogic Server documentation.

Upgraded Applications Use Struts 1.1 and Related J2EE Libraries

The Import Wizard includes the Struts 1.1 shared J2EE library for upgraded applications. The Struts 1.1 J2EE library is used to maintain the operability of your upgraded application; before an upgraded application can use Struts 1.2, any code in page flows or portlets with explicit dependencies on Struts 1.1 would need to be manually updated. (For more information, refer to *“Changes in Behavior Between Struts 1.1 and 1.2”* on page C-5.

For new portal web projects, the Struts 1.2 shared J2EE library is included by default, but you can choose Struts 1.1 if desired. Oracle strongly recommends that you keep the default setting to use Struts 1.2 with new applications. Even with upgraded applications, it would be beneficial to use

Struts 1.2 if possible, and you might want to experiment with Struts 1.2 to see if any explicit dependencies on Struts 1.1 actually exist.

Changes in Behavior Between Struts 1.1 and 1.2

WebLogic Portal support for Struts is slightly different if you upgrade to Struts 1.2.

Struts 1.1 support in WebLogic Portal will be the same as in previous releases, with the `struts-adapter` taglibs mapped to URIs using `web.xml`. In this case, you should use the `struts-1.1.war` J2EE library instead of the new `struts-1.2.war` J2EE library.

For applications upgrading to Struts 1.2, instead of mapping the `struts` and `struts-adapter` taglibs using `web.xml`, WebLogic Portal now relies on the JSP 1.2 implicit taglib mapping, wherein any `.tld` files in the `META-INF` directory in a JAR are implicitly mapped by the web container to the URI specified in the `tld`. In the case of WebLogic Portal, these are in `struts-adapter.jar`, in the path `META-INF/tlds`.

You can choose to use one of these two methods to upgrade to Struts 1.2:

- Modify all JSPs that use the `struts` taglibs to reference `http://bea.com/struts/adapter/tags-html` and `http://bea.com/struts/adapter/tags-nested` for the HTML and nested taglibs, and `http://struts.apache.org/tags-*` for the remainder of the taglibs that the portal adapter does not override.
- Extract the `.tlds` from both `struts.jar` (in `struts-1.1.war`) and from `struts-adapter.jar` and copy them to `WEB-INF/tlds`. This allows for the case where you want to continue using the explicit `tld` mapping via `web.xml`.

Ampersand Entities in Portal URLs

In past releases, WebLogic Portal used the configuration file `url-template-config.xml` for configuring the form of WebLogic Portal-generated URLs. WebLogic Portal now uses the Beehive equivalent: `beehive-url-template-config.xml`. The previous configuration file contained an element (`generate-xml-amp-entity`) that caused URLs to be generated with ampersand entities instead of characters for parameter separators, that is:

```
http://www...?arg1=foo&amp;arg2=bar (entity)
```

instead of:

```
http://www...?arg1=foo&arg2=bar (character)
```

In the absence of this configuration element, URLs were generated with ampersand characters.

The Beehive equivalent configuration element for ampersand entities is located in the NetUI configuration file, `beehive-netui-config.xml`. The Beehive default for ampersand entities is the opposite of the previous Portal default; in the absence of the configuration element, URLs are generated with ampersand entities.

Portal Framework now uses the NetUI configuration file as the source for this configuration element, and also uses the Beehive semantics. This means that by default, URLs generated by Portal contain ampersand entities. Note that this applies only to HTML configurations – XHTML configurations force ampersand entities in URLs, regardless of configuration setting.

If cases exist where you want to ensure that URLs are generated with characters rather than ampersand entities, you must add that configuration element to the `beehive-netui-config.xml` file.

Upgrading Individual `application-config.xml` Files Later

Applications created using WebLogic Portal Version 8.1.4+ used the file `META-INF/application-config.xml` for configuration of various MBeans. In WebLogic Portal 10.3, MBeans have been changed to descriptor beans. Settings previously contained within `application-config.xml` are now in the appropriate descriptor bean configuration files, including:

- `content-config.xml`
- `p13n-config.xml`
- `p13n-cache-config.xml`
- `p13n-security-config.xml`
- `wps-config.xml`

When you import a Version 8.1.4+ application into Workshop for WebLogic, the import process performs the necessary conversion. For individual instances of the 8.1 `application-config.xml` file that you might want to use, you can right-click the file name in the Package Explorer to access the **Upgrade** context menu, which will convert the file.

Correcting Duplicate Portlet Category Names Before Propagating an Upgraded Application

In past releases of WebLogic Portal it was possible, though not recommended, to create more than one portlet category with the same name, at the same level in the hierarchy. In WebLogic

10.3, this operation is not permitted. (You can use the same name for more than one category, but they must not be “peers” in the hierarchy.)

When you upgrade a portal application to 10.3, any duplicate portlet category names that were used previously are preserved. It is extremely important that you edit these category names to be unique; otherwise the WebLogic Portal propagation tools might cause unexpected results, or errors might occur during the propagation process.

Upgrading WebLogic Portal 8.1 Projects to WebLogic Portal 10.3