Oracle® WebLogic Integration

Tutorial: Upgrading WLI 8.x, WLI 9.x or WLI 10.x Application Source to WLI 10*g* Release 3 (10.3.1)

10g Release 3 (10.3.1)

January 2010



Tutorial: Upgrading WLI 8.x, WLI 9.x, or WLI 10.x Application Source to WLI 10g Release 3 (10.3.1), 10g Release 3 (10.3.1)

Copyright © 2007, 2010, 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. Tutorial: Overview

Tutorial Objective and Scope	1-1
Prerequisites	1-2
About the Import Wizard	1-2
Creating the Source WLI 8.x Application	1-3
Creating the Source WLI 9.x or WLI 10.x Application	1-5
Tutorial: Upgrading the WLI 8.1 or WLI 8.5 Applicatio	n Source
Application Overview	2-1
Importing the Application Source	2-2
Validating the Upgraded Application	2-12
Deploying and Publishing the Application	2-12
Creating a WebLogic Integration Domain Using the Configuration W	Vizard 2-15
Running the Application	
Post-Upgrade WLI Source Artifacts	
	Tutorial Objective and Scope. Prerequisites. About the Import Wizard. Creating the Source WLI 8.x Application Creating the Source WLI 9.x or WLI 10.x Application. Tutorial: Upgrading the WLI 8.1 or WLI 8.5 Applicatio Application Overview Importing the Application Source Validating the Upgraded Application Deploying and Publishing the Application. Creating a WebLogic Integration Domain Using the Configuration W Running the Application. Post-Upgrade WLI Source Artifacts

3. Tutorial: Upgrading the WLI 9.x and 10.x Application Source

Tutorial: Upgrading WLI 8.x, WLI 9.x, or WLI 10.x Application Source to WLI 10g Release 3 (10.3.1)



Tutorial: Overview

This tutorial describes how to upgrade WLI 8.x, WLI 9.x, or WLI 10.x applications, to the Oracle WebLogic Integration 10*g* Release 3 (10.3.1) environment. This tutorial describes the upgrade scenario using a Process Application sample created in WebLogic Platform 8.x.

This section includes the following:

- Tutorial Objective and Scope
- Prerequisites
- About the Import Wizard
- Creating the Source WLI 8.x Application
- Creating the Source WLI 9.x or WLI 10.x Application

Tutorial Objective and Scope

The objective of this tutorial is to help you upgrade your existing WLI 8.x, WLI 9.x, and WLI 10.x applications to Oracle WebLogic Integration 10g Release 3 (10.3.1). The upgrade process does not alter the source application logic and intent in any manner. The process simply upgrades the application for use in the Oracle Workshop for WebLogic workspace.

Upgrading an application to WLI 10g Release 3 (10.3.1) involves the following steps:

• Creating the Application in WLI 8.x, WLI 9.x, or WLI 10.x

Create the Application in WLI 8.x, WLI 9.x, or WLI 10.x as required, and replace the deprecated APIs for smooth upgrade to Oracle WebLogic Integration 10*g* Release 3 (10.3.1).

• Importing the Application into the Oracle WebLogic Integration 10g Release 3 (10.3.1) environment, using the Import Wizard

Import the Application including all its projects using the wizard.

• Deploying the Application in Oracle WebLogic Integration 10g Release 3 (10.3.1)

Create a server with Oracle WebLogic Integration domain, and then deploy and publish the Application.

• Running the Application to validate the upgrade process

Execute the Application using test values, and confirm the validity of the upgraded Application by executing a process.

Prerequisites

There are a few prerequisites for the smooth execution of this tutorial. For example, you require an application from the WLI 8.x, WLI, 9.x, or WLI 10.x release as the source for the upgrade exercise. The requirements for successful execution of the tutorial are as follows:

- Ensure that the application to be upgraded is built on (or upgraded to) WLI 8.1 Service Pack 4 or higher.
- Ensure that you have access to the WLI 8.1 Service Pack 4 or higher installation as you will use the Process Application tutorial sample created in that environment.

About the Import Wizard

The upgrade process from WLI 8.x, WLI 9.x, or WLI 10.x to Oracle WebLogic Integration 10g Release 3 (10.3.1) is handled by an import wizard. The wizard does not alter the logic and intent of the existing WLI 8.x, WLI 9.x, or WLI 10.x application, nor extract the application from any source repository. The wizard migrates the WLI 8.x, WLI 9.x, or WLI 10.x source artifacts into the Oracle WebLogic Integration 10g Release 3 (10.3.1) source and project model. However, it retains the WLI 8.x, WLI 9.x, or WLI 10.x Javadoc annotations as they do not require any special processing in Oracle WebLogic Integration 10g Release 3 (10.3.1). In addition, these annotations are retained to facilitate any manual processing that may be required after upgrading the application.

You can use the import wizard to do the following:

- Convert WLI 8.x, WLI 9.x, or WLI 10.x project types to Oracle WebLogic Integration 10*g* Release 3 (10.3.1) project types.
- Convert WLI 8.x Javadoc annotations to JSR 175 compliant annotations.
- Move the shared libraries from the WLI 8.x, WLI 9.x, or WLI 10.x application directory, to the J2EE EAR project that is part of the upgraded application.
- Move JSP files into a WebContent directory in the Oracle WebLogic Integration 10g Release 3 (10.3.1) environment.
- Upgrade NetUI JSP tags to Apache Beehive JSP tags, or just update existing NetUI JSP tags to the Oracle WebLogic Integration 10g Release 3 (10.3.1) compatible NetUI version JSP tags.
- Move XSD files that are in a WLI 8.x, WLI 9.x, or WLI 10.x Schema project into a Oracle WebLogic Integration 10g Release 3 (10.3.1) Utility project.
- Move Java packages and sources into src folders.

Creating the Source WLI 8.x Application

For the purpose of this tutorial, you must first create a Request Quote application in the WebLogic Platform 8.1 Service Pack 4, 5 or 6 release.

To create the new application in WLI 8.x, complete the following steps:

- 1. In WebLogic Workshop 8.x select the **File > New > Application...** menu option. The New Application dialog box is displayed.
- 2. In the left pane select Tutorial and in the right pane, select Tutorial: Process Application.
- 3. In the **Name:** field, type the name **sampleApp**. The New Application dialog box is displayed as shown in Figure 1-1.

New Appli	ication
All Data S Portal Proces	Service Initial: Java Control Initial: Veb Service Initial: Page Flow Initial: Page Flow Initial: Process Application Initial: Initial: Enterprise JavaBeans Initial: Hello World Process Application
Directory:	F:\bea\platform815\user_projects\applications\sampleApp Browse
<u>N</u> ame:	sampleApp
Server:	F:\bea\platform815\weblogic81\samples\domains\workst = Browse
Creates a Transform	new application containing components for the Process and Data nation tutorials.
	Create Cancel

Figure 1-1 Creating a New Application in WebLogic Platform 8.x

- 4. Click Create.
- 5. In the Application pane of WebLogic Workshop, navigate to view the **sampleApp** > **sampleAppWeb** > **requestquote** > **services** > **AvailProcessor.jws** web service file.
- 6. Double-click the file to view its **Source View** in the adjacent pane, as shown in Figure 1-2.

Application Files × AvailProcessor.jws - {sampleAppWeb}\requestquote\services\ return "BackOrder": 🔁 sampleApp 🖻 🗟 sampleAppWeb 🗄 🔄 requestquote return "BackOrder"; 🗄 🔄 services 3 😪 AvailProcessor.jws public interface Callback extends weblogic.jws.control.ServiceControl 川 AvailProcessorControl.jc 📲 PriceProcessor.iws* * @common:message-buffer enable="true" 💷 PriceProcessorControl.ic * @jws:conversation phase="finish" 📲 TaxCalc.iws public void avail(int itemID, int qty, boolean avail, String date III TaxCalcControl.icx 🛃 TaxCalcProcess.jpd static final long serialVersionUID = 1L; A TaxCalcProcessTransfor stateToString.xg 嚞 ValidateOrder.jpd 川 validatePublish.jcx Design View Source View 🗄 💼 testxml

Figure 1-2 Viewing the Web Service Source

- 7. In the highlighted text (public interface Callback extends weblogic.jws.control.ServiceControl) of the source view, as shown in Figure 1-2, replace weblogic.jws.control.ServiceControl with com.bea.control.Control.
- 8. At the beginning of the source file, replace import weblogic.jws.control.JwsContext with import com.bea.control.JwsContext.
 - **Note:** The weblogic.jws.control.ServiceControl and the weblogic.jws.control.JwsContext are deprecated. You must replace these APIs in all the web services for an error-free upgrade to WLI 10g Release 3 (10.3.1).
- 9. Complete step 5 to step 9 for the remaining two web services, PriceProcessor.jws and TaxCalc.jws.
- 10. Save and close the application.

The application, sampleApp is ready for use. You can upgrade sampleApp using the Oracle Workshop for WebLogic 10g Release 3 (10.3.1) IDE.

Creating the Source WLI 9.x or WLI 10.x Application

For the purpose of this tutorial, you must first create a Request Quote application in the WebLogic Integration 9.x or WLI 10.x release. The primary goal of this task is to create an Application in WLI 9.x or WLI 10.x and upgrade into the Oracle WebLogic Integration 10g Release 3 (10.3.1) release.

If you do not have an existing project in WLI 9.x or WLI 10.x, you need to create the new application in WLI 9.x or WLI 10.x. To do this, complete the following steps:

 In WebLogic Workshop 9.x or WLI 10.x select the File > New > Project > WebLogic Integration > Process Application menu option. The Process Application dialog box is displayed as shown in Figure 1-3.

Figure 1-3 Create a new Process Application

Trocess Application
Process Application Greates a new Process application containing EAR, Web and Utility projects.
EAR Project Name:
The EAR Project is a container for J2EE application resources. It references other projects belonging to the Process Application.
Web Project Name:
The Web Project is a container for Processes and related resources such as Transformations and Controls.
Utility Project Name:
The Utility Project is a container for Transformations, Controls, Schemas and other resources that you intend to share across other projects.
F Add Weblogic Integration System and Control Schemas to Utility Project
<back next=""> Finish Cancel</back>

2. Enter an EAR Project name, Web Project name and Utility Project name and click Finish to create the project.

Alternatively, you can skip the first two steps and import a WLI 8.x Application to the WLI 9.x or WLI 10.x workspace. To import a WLI 8.x Application:

- 1. Select **File > Import > Workshop 8.1 Application** and click Next on the Import dialog box to import the WLI 8.x application into the workspace.
- 2. Specify the path to the WLI 8.x application on the Application Import dialog box, or the archive file, open the .work file and click Next to set upgrade preferences on the Source Upgrade dialog box. See Importing the Application Source for more details.
- 3. Click Next to start the upgrade, and preview the pending upgrade actions on the Upgrade Preview dialog box. Click Finish on the Upgrade Preview dialog box to complete the project upgrade to WLI 9.x or WLI 10.x.
- 4. Save and close the project.

You have successfully imported an WLI 8.x application into the WLI 9.x or WLI 10.x workspace.



Tutorial: Upgrading the WLI 8.1 or WLI 8.5 Application Source

This section describes how to upgrade the application source of WLI 8.1 SP4 and higher, or WLI 8.5 applications. It discusses the following topics:

- Application Overview
- Importing the Application Source
- Validating the Upgraded Application
- Post-Upgrade WLI Source Artifacts

This tutorial does not include validation of all aspects of the sample application.

Application Overview

The application used in this tutorial creates a business process that meets the following requirements:

- Receives Request for Quote messages from clients.
- Starts the business process on receipt of the Request for Quote.
- Validates and processes the request.
- Sends the status of the Request for Quote to the client.

Importing the Application Source

This section describes how to upgrade your WebLogic Integration 8.1 Service Pack 4, 5, or 6 application source for use in the Oracle Workshop for WebLogic environment.

Start Oracle Workshop for WebLogic by selecting Start > Programs > Oracle WebLogic > Workshop for WebLogic 10gR3 from the Start menu. The Workspace Launcher dialog box is displayed as shown in Figure 2-1.

Figure 2-1 Setting a Workspace for the Upgraded Application

🖸 Workspace Launcher 🛛 🗙
Select a workspace
Oracle Workshop for WebLogic stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.
Workspace: C:\bea\user_projects\workspaces\Upgrade Browse Browse
OK Cancel

2. Specify the desired location for the upgraded application in the **Workspace** field, as shown in Figure 2-1. For this tutorial, use the upgrade workspace, under the WLI installation directory, as follows:

\$BEA_HOME\user_projects\workspaces\upgrade

- **Note:** The \$BEA_HOME location used throughout this tutorial is C:\bea\ where Oracle WebLogic Integration 10g Release 3 (10.3.1) is installed.
- 3. Click OK. The Workspace is launched, as shown in Figure 2-2.



Welcome to Oracle® Workshop for WebLogic Version 10.3	Go to the workbench
	Overview Get an overview of the features
	Find out what is new
	Go through tutorials
	Samples Try out the samples
	First Steps Take your first steps
	Web Resources Read more on the Web

4. Click **Workbench** to proceed to the workbench as shown in Figure 2-3.

Figure 2-3	Workbench
------------	-----------

🗖 Java EE - Oracle Workshop for We	bLogic						- C X
File Edit Navigate Search Project Run	Window Help						
10.80 \$.0.4.	82 8 8 8 8 8 4	0101012	1000			📑 😒 Java	EE
					••	El Crate S Se outre is not evaluate.	
Spongen Polette 22		Properties di Servers 19	Samuta				
	D errors, 0 warnings, 0 infos	Properties in particip	21499-013				*
	Description -	Resource	Path	Location			
	-		-			(a) (a) (a) (a) (a)	0 A A A

5. Select the **File > Import...** menu option to start the import procedure. The Import Source dialog box is displayed as shown in Figure 2-4.

Figure 2-4 Selecting the Import Source

🖸 Import							
Select Import a Works	hop 8.1 Applicat	ion or a Bee	hive Applicati	on		Ľ	5
Select an impor	t source:						
Gener	al YEL Process aport WSDL orkshop 8.1 App	olication					
0		< Back	Next >		Finish	Cano	el

6. Select the **Workshop 8.1 Application** option as the application source type, as shown in Figure 2-4, and click **Next**. The Application Import dialog box appears as shown in Figure 2-5.

Figure 2-5 Selecting the Import Application

🖸 Workshop 8.1 Application Upgrade
Application Import Select a Workshop 8.1 application for import and upgrade.
Only applications built with SP4 or later can be upgraded correctly. The version of the specified application cannot be determined, verify that this application satisfies the version requirements.
Workshop 8.1 Application er_projects\applications\sampleApp\sampleApp.work V Browse
 □ ♥ ♪ sampleApp □ ♥ ↓ Schemas ■ ♥ ↓ sampleAppWeb
Select All Deselect All
Options Add this prefix to project names
Target Runtime Oracle WebLogic Server v10.3
Cancel

7. Click Browse to navigate through the directory structure, and select the sampleApp.work file that you created in WLI 8.x. For more information, see Creating the Source WLI 8.x Application. As shown in Figure 2-5, the sampleApp.work file for this tutorial is located in:

C:\bea\user_projects\applications\sampleApp

After selecting the sample application from the WLI 8.x install, a list of projects in that application is displayed in the dialog box. You can choose the projects to import and upgrade by selecting the check boxes adjacent to them. However, it is recommended that you include all the projects, as most of them have inter-dependencies.

8. Click **Next.** The Source Upgrade preference selection dialog box is displayed, as shown in Figure 2-6.

Figure 2-6	Setting the	Upgrade	Preferences
------------	-------------	---------	-------------

🖸 Workshop 8.1 Appli	cation Upgrade	
Source Upgrade Upgrade preference selection	ın.	
▼ General		
Error Handling:	Log errors and continue upgrade	~
Message verbosity:	Include informational comments	~
 Properties file upgr. JPD Document Upgr JSP File Migrator op 	ader options ader options tions	
0	<pre></pre>	Cancel

You can configure your preference by selecting the various options displayed in the dialog box. These options are expanded and shown in Figure 2-7.

Figure 2-7 Source Upgrade Preference Selection

6	Error Handling:	Log errors and continue upgrade
1	Message verbosity:	Include informational comments
→ Pro	operties file upgra	ader options
	Delete copied res Resource bundle files along with other sour remain there, they co	ource bundle files from the web content folder ; (, properties) are copied from a web projects content folder to its source folde co files. If none of the JSP pages in the web content folder rely on these files an be safely deleted.
- JPI	D Document Upgr	ader options
^X	Query Options	
	Upgrade XQ2002 By default, upgrade Select the check box	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft
	Upgrade XQ2002 By default, upgrade i Select the check box version so that the J	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version, above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries.
▼ JSI	Upgrade XQ2002 By default, upgrade Select the check box version so that the J	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries.
ا د در	Upgrade XQ2002 By default, upgrade Select the check box version so that the J P File Migrator op/ ISP Tags	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries.
ا بر در در	Upgrade XQ2002 By default, upgrade i Select the check box version so that the J P File Migrator opi SP Tags Replace BEA NetL	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. tions
ا اکر ب	Upgrade XQ2002 By default, upgrade Select the check box version so that the J P File Migrator opl SP Tags Replace BEA Nett By default, upgrade 10.0 server runkime, compatibility tags wit are the default for n	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. It ags with Apache Beehive tags. replaces the 6.1 NetUI tags with a set of NetUI tags that are compatible with the select the check box above to cause upgrade to replace these NetUI h the enhanced NetUI tags from the Apache Beehive project. The Beehive tag aw web projects.
- JSI	Upgrade XQ2002 By default, upgrade Select the check box version so that the J P File Migrator opl SP Tags Replace BEA Nett By default, upgrade 10.0 server runtime, compatibility tags wit are the default for nn You can also convert	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. .above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. It ags with Apache Beehive tags. replaces the 6.1 NetUI tags with a set of NetUI tags that are compatible with t Select the check box above to cause upgrade to replace these NetUI h the enhanced NetUI tags from the Apache Beehive project. The Beehive tag aw web projects. to the Beehive tags any time after upgrade, on all or a subset of JSPs.
• JSI	Upgrade XQ2002 By default, upgrade i Select the check box version so that the J P File Migrator opl SP Tags Replace BEA NetL By default, upgrade 10.0 server runtime. Compatibility tags wit are the default for n You can also convert See upgrade docume	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. tions If tags with Apache Beehive tags. replaces the 6.1 NetUI tags with a set of NetUI tags that are compatible with t Select the check box above to cause upgrade to replace these NetUI In the enhanced NetUI tags from the Apache Beehive project. The Beehive tag sw web projects. to the Beehive tags any time after upgrade, on all or a subset of JSPs. intation for code changes required if you convert to the Beehive tags.
 ✓ JSI □ □<!--</td--><td>Upgrade XQ2002 By default, upgrade i Select the check box version so that the J P File Migrator opi SP Tags Replace BEA Nett By default, upgrade 10.0 server runtime. compatibility tags wit are the default for n You can also convert See upgrade docume</td><td>to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. tions JI tags with Apache Beehive tags. replaces the 8.1. NetUI tags with a set of NetUI tags that are compatible with t Select the check box above to cause upgrade to replace these NetUI the enhanced NetUI tags from the Apache Beehive project. The Beehive tag ew web projects. to the Beehive tags any time after upgrade, on all or a subset of JSPs. intation for code changes required if you convert to the Beehive tags.</td>	Upgrade XQ2002 By default, upgrade i Select the check box version so that the J P File Migrator opi SP Tags Replace BEA Nett By default, upgrade 10.0 server runtime. compatibility tags wit are the default for n You can also convert See upgrade docume	to XQ2004 keeps the inline xqueries which are compliant with xquery 2002 draft version. above to cause upgrade to convert the inline xqueries to xquery 2004 draft PD node editor will be able to parse the inline xqueries. tions JI tags with Apache Beehive tags. replaces the 8.1. NetUI tags with a set of NetUI tags that are compatible with t Select the check box above to cause upgrade to replace these NetUI the enhanced NetUI tags from the Apache Beehive project. The Beehive tag ew web projects. to the Beehive tags any time after upgrade, on all or a subset of JSPs. intation for code changes required if you convert to the Beehive tags.

These options are available under specific categories, as listed below, depending on their function.

- **General** – You can set the error-handling options and the type of content you want to record. Use the default value for both the properties, as described in Table 2-1.

Property	Description	
Error Handling	Determines how the errors are handled and the subsequent action you would like to implement. Error messages are always recorded in a log file, but your selection here determines how they are delivered to you. There are three options for this field, namely:	
	Log errors and continue upgrade (default option)	
	Log errors but abort upgrade	
	Display dialog on errors	
Message Verbosity	Determines the type of messages that will be captured during the upgrade process. There are three types of messages that are captured: information, warning, and error. These three message types are reflected in the three options for this field, namely:	
	• Include informational comments (default option) – All three message types	
	Include warning comments – Only warning and error type messages	
	Include error comments – Only error messages	

Table 2-1 Setting General Preferences

- NetUI Project Upgrader options Selecting the Use WebLogic J2EE Shared Libraries
 option lets you use the WebLogic J2EE shared libraries, without having to duplicate
 runtime . jar files across projects. You can ignore this option as it usually does not
 impact the Oracle WebLogic Integration applications. By default, this option is selected
 for upgrade.
- Properties File Upgrader options The Delete copied resource bundle files from the web content folder option removes any unnecessary .properties files. If none of the JSP pages in the web content folder require .properties files, they can be deleted. You can ignore this option as it usually does not impact the WebLogic Integration applications. By default, this option is not selected for upgrade.
- JPD Document Upgrader options Select the Upgrade XQ2002 to XQ2004 option as it converts XQuery statements from XQuery 2002 draft version to XQuery 2004 draft version. This option lets the JPD node editor parse through the inline XQuery statements and convert them to the 2004 version. By default, this option is not selected for upgrade.
 - Ensure that you select the **Upgrade XQ2002 to XQ2004** option. For more information on upgrading XQuery files, see the **Upgrading XQuery Code** section of the *Transforming Data Using XQuery Mapper* document.

- JSP File Migrator options When selected, the Replace NetUI tags with Apache Beehive tags option upgrades the WebLogic Integration 8.x supported NetUI tags with Apache Beehive compatible NetUI tags. You can ignore this option as it usually does not impact the Oracle WebLogic Integration applications. By default, this option is not selected for upgrade.
- **Note:** You can convert projects or selective JSPs to use Apache Beehive compatible NetUI tags even after the upgrade exercise. However, all new projects created in Oracle WebLogic Integration 10g Release 3 (10.3.1) use the Apache Beehive compatible NetUI tags.
- 9. Click **Next**. The Progress Information dialog box is displayed, as shown in Figure 2-8. At this step, the application being upgraded is assessed and a detailed list of upgrade tasks that need to be performed (and not performed) is generated.

Figure 2-8 Tracking Upgrade Progress

Progress Information	
Generating report for file: PriceAvailTransformations.dtf	
- moving source artifact : validatePublish.jcx	
	Cancel

After completing the upgrade process, the **Upgrade Preview** dialog box is displayed, as shown in Figure 2-9. The preview dialog box provides a summary of changes that will be implemented during the upgrade task.



🖸 Workshop 8.1 Application Upgrade	
Upgrade Preview	
Preview the pending upgrade actions.	
Below is a list of files and a description of changes that will be made:	
E 🖉 Source Upgrade Actions	
E (î) \Schemas	
(i) \sampleAppWeb\WebContent\WEB-INF\weblogic.xml	
IsampleAppWeb\src\requestquote\FileQuote, jcx IsampleAppWeb\src\requestquote\FileQu	
i) \sampleAppWeb\src\requestquote\PriceAvailTransformations.dtf	
🛓 🗄 \sampleAppWeb\src\requestquote\RequestQuote.jpd	
i (i) \sampleAppWeb\src\requestquote\RequestQuoteTransformation.dtf	
) sampleAppWeb\src\requestquote\RequestQuoteavailProcessorGetAvail.xq	
(i) \sampleAppWeb\src\requestquote\RequestQuoteavailProcessor_avail.xq	
IsampleAppWeb\src\requestquote\RequestQuotepriceProcessorGetPrice.xq	
(1) \sampleAppWeb\src\requestquote\RequestQuotepriceProcessor_returnPrice.xq	
() \sampleAppWeb\src\requestquote\RequestQuotetaxCalculationRequestTaxRate.xq	
Image: SampleAppWeb\src\requestquote\TutorialJoin.dtf	
() \sampleAppWeb\src\requestquote\convertAvailListToXML.xq	
() \sampleAppWeb\src\requestquote\convertAvailXMLtoXMLObj.xq	
(1) \sampleAppWeb\src\requestquote\convertPriceListToXML.xq	
(1) \sampleAppWeb\src\requestquote\convertPriceXMLtoXMLObj.xq	
() \sampleAppWeb\src\requestquote\join.xq	
IsampleAppweb(src(requestquote(services(AvailProcessor.)ws	
IsampleAppweb(src(requestquote(services(AvailProcessorControl.)cx	
(sampleAppWeb(src/requestquote(services(PriceProcessor.)ws	
(sampleAppWeb(src)requestquote(services)Thcervices)Sol Condition(cx	
(sampleAppWeb(sic)requestquote(services)TaxCalc.)ws	
SampleAppWeb(src)requestquote(services)TaxCalcProcess.ind	
Sample App Web)src/requestquate/services/TaxCalcProcessTransformation.dtf	
SampleAppWeb\src\reguestquote\services\\alidateOrder.ipd	
(i) \sampleAppWeb\src\reguestguote\services\stateToString.xg	
🗄 🗀 Files requiring no upgrade	
Caracteria Back Next > Finish Caracteria	incel

10. Click **Finish** when you are satisfied with the items listed for upgrade, and those that do not require to be upgraded. The **Upgrade Preview** dialog box is refreshed, and it displays the upgrade progress status at the bottom of the dialog box, as shown in Figure 2-10.

Figure 2-10 Upgrade Progress Indicator



- 11. The last step is to import the test data from the WLI 8.x application.
 - a. In Project Explorer, browse to the requestquote folder in SampleAppWeb/Webcontent directory. Right click on the requestquote folder and select the Import option.
 - b. On the Import dialog box, select General > File System and click Next.
 - c. Click Browse on the File System dialog box to locate the SampleApp/SampleAppWeb/requestquote/testxml folder in the WLI 8.x application created earlier. Click OK. Select the testxml check box if not selected. Click Finish to complete importing the test data.

After the upgrade is complete, Workshop builds the workspace for the application in the IDE. A log of the import process is displayed in the Problems pane at the bottom of the dialog box; while a log of the upgrade process is displayed in the Upgrade pane. You can customize the type of information that is displayed by making the appropriate changes in the Upgrade Preferences dialog box.

Note: For the purpose of this tutorial, you can ignore the warning and information type error messages displayed in the Problems pane.

The following sections help you validate the application in the Oracle WebLogic Integration 10*g* Release 3 (10.3.1) environment. The tasks include deploying and running the application in a Oracle WebLogic Integration domain.

Validating the Upgraded Application

This section describes the steps involved in validating the upgraded application in the Oracle WebLogic Integration 10g Release 3 (10.3.1) environment. The task involves deploying the application on the server, publishing it, and subsequently running the application with some test values.

Deploying and Publishing the Application

Perform the following tasks to deploy the application after a successful upgrade.

 In the Package Explorer pane, navigate through the directory structure to view the sampleAppWeb > src > requestquote > RequestQuote.java file, as shown in Figure 2-11.



Figure 2-11 Selecting the RequestQuote.java File

 Right-click the RequestQuote.java file, and select Run As > Run On Server menu option. The Run on Server dialog box is displayed, as shown in Figure 2-12.

Figure 2-12 Defining a New Server to Deploy the Application

🖸 Run On Server			
Define a New Se	rver		
Choose the type of	server to create		
How do you want to	select the server?		
Choose an exis	ting server		
 Manually define 	a new server		
Server's host name:	localhost		~
		Downloa	d additional server adapters
Select the server typ	e:		
type filter text			
Apache Apache	is, Inc. hnology DC4J Standalone Server 10.1.3 DC4J Standalone Server 10.1.3.1 WebLogic Server v10.3		
Server runtime: Ora	cle WebLogic Server v10.3 erver when running this project		 Installed Runtimes
0	< Back	Next >	Finish Cancel

- 3. Select the **Manually define a new server** check box, and fill out the form, as displayed in Figure 2-12.
- 4. Click **Next**. The **Run On Server** dialog box is refreshed with the **Specify a WebLogic domain directory** page, as shown in Figure 2-13.

Figure 2-13 Defining a WebLogic Server

🖸 Run On Serve	er	
Oracle WebLogi	jic Server v10.3 .ogic domain directory	ORACLE [®]
Name: Ora	acle WebLogic Server v10.3 at localhost	
Server Type: 💿 L	Local 🔘 Remote	
Configuration Domain Directory:	:: Lick here to launch Configuration Wizard to crea	Browse)
Disable Auton	matic Publishing To Server	
Always start	: WebLogic Server in debug mode / degrade performance of some applications when not debugging, but om running to debugging an application without restarting the server.	allows Eclipse
0	< Back Next > Finish	Cancel

In following section, you will create a new Oracle WebLogic Integration domain in the Oracle WebLogic Server.

Creating a WebLogic Integration Domain Using the Configuration Wizard

Complete the following steps to set up the Oracle WebLogic Integration domain to deploy the sampleApp application.

1. Select the **Click here to launch Configuration Wizard to create a new domain** option in Run On Server dialog box to start the Oracle WebLogic Configuration Wizard as shown in Figure 2-14.

Oracle WebLogic Configuration Wizard Welcome Choose between creating and extending a domain. Based on your selection, the Configuration Wizard guides you through the steps to generate a new or extend an existing domain. Orceate a new WebLogic domain Create a new WebLogic domain Create a webLogic domain in your projects directory. O Extend an existing WebLogic domain. Extend an existing WebLogic domain. Use this option to add applications and services, or to override existing database access (IDBC) and messaging (IMS) settings. You can also incorporate additional functionality in your domain, for example, by including AquaLogic Service Bus. Egt Previous

Figure 2-14 Creating a New WebLogic Domain

- Select the Create a new WebLogic domain option, and click Next. The Select Domain Source page is displayed.
- 3. Select the **Workshop for WebLogic 10.3** and the **WebLogic Integration** options, and click **Next**. The Configure Administrator Username and Password page is displayed.
- 4. Enter weblogic in the User name, User password, and the Confirm user password fields, and click Next. The Configure Server Start Mode and JDK page is displayed.
- In the WebLogic Domain Startup Mode pane, select Development Mode. In the JDK Selection pane, select the Sun SDK and click Next. The Customize Environment and Service Settings page is displayed.
- 6. Select **Yes** and click **Next**. The Configure the Administrator Server page is displayed. Click **Next** to proceed without making any changes.

 Similarly, do not make any changes in subsequent Configure Managed Servers, Configure Machines, and Configure the JDBC Data Source pages. Click Next and skip to the Run Database Scripts page, as shown in Figure 2-15.

Figure 2-15 Running Database Scripts

Oracle WebLogic Configuration Wizard		
Run Database Scripts If your connections tested OK, you may now run database scr select the desired database loading options and database vers	ipts. For each JDBC data source, sion, and click Run Scripts.	ORACLE
Available JDBC Data Sources cqDataSource cgDataSource cgDataSource portalDataSource bnmArchDataSource Available SQL Files and Database Loading Options Drop/Create P13N Database Objects	Results	
DB Version: 51 T Run Scripts	Log File: C:\bea\WLSERV~1.3\common\lib\jdbc.log	Browse Previous Next

- 8. From the list of Available JDBC Data Sources, select the **p13nDataSource** option, as shown in Figure 2-15 and click **Run Scripts**. On successful execution, the Results pane will display **Database Load Successful!**.
- 9. Again, from the list of Available JDBC Data Sources, scroll down and select cgDataSource-nonXA and click Run Scripts again. On successful execution, the Results pane will display Database Load Successful!.
- 10. Click **Next**. The Configure JMS File Stores page is displayed. Do not make any changes and click **Next**. The Review WebLogic Domain page is displayed, as shown in Figure 2-16, which summarizes the contents of the domain and reflects the options you have selected.



Figure 2-16 Review the Domain Configuration Settings

- 11. Click **Next** to proceed without making any changes. The Create WebLogic Domain page is displayed.
- 12. Enter a name in the **Domain Name** field, for example sample_wli_domain, and click **Create**. The domain creation process starts. On successful creation, the Create Domain dialog box appears as shown in Figure 2-17.

Oracle WebLogic Configuration	Wizard	
Creating Domain		ORACLE
	Progress:	
	Preparing Extracting Domain Contents Creating Domain Security Information Saving the Domain Information Storing Domain Information String Substituting Domain Files Performing OS Specific Tasks Performing Post Domain Creation Tasks Domain Created Successfully! Domain Location: C:\bea\user_projects\domains\sample_wli_domain	
ORACLE		Start Admin Server
Exit		Previous Done

Figure 2-17 Successful Creation of a Domain

- 13. Click **Done** to conclude the Domain creation process.
- 14. After creating the Domain using the Configuration Wizard, return to the Run On Server dialog box as shown in Figure 2-13, to publish the application on the server with the domain you just created.
- 15. Click **Browse.** Navigate to the folder in which the new WebLogic Integration Domain sample_wli_domain was created. Select the domain. The new domain for this tutorial is located at:

```
C:\bea\user_projects\domains\sample_wli_domain
```

- 16. Click **Next** on the Run On Server dialog box to display the Add and Remove Projects page. Ensure the **sampleApp** project is listed in the **Configured projects** column.
- 17. Click **Finish** to publish the Application. This process may take some time if you are starting the server for the first time.

After successfully deploying and publishing the Application, a browser pane is opened in the IDE, displaying the **Overview** tab of **RequestQuote.jpd Process**.

18. Click the **Test Form** tab in the browser pane, as shown in Figure 2-18.

Figure 2-18 Published Application Ready for Validation

RequestQuote.jpd F	rocess		Created by Oracle Webl	.ogic Workshop 🗠
Overview Console Test Form Tes	st SOAP Message	Broker Process Graph	<u>http://localhost:7001/Sam</u> requestquote/Reque	pleAppWeb/ estQuote.jpd
<u>Start operations</u> Monitor all RequestQuote.jpd processes				
Message Log 🛛 🖉 <u>Refresh</u>	quoteRequest			
Log is empty				~
	xml requestXML:			
				~
	xml requestXML: (file value)			Browse
	[quoteRequest starts a Process	1	
				~

The Application is now ready for the final validation process. The following section contains details on how to run the application.

Running the Application

This is the final step in validating the upgraded Application and the WLI 8.x to Oracle WebLogic Integration 10g Release 3 (10.3.1) upgrade procedure. The primary task of this section is to provide a set of test values to execute the Application, and to verify the results.

1. Locate the **QuoteRequest.xml** file that contains the test values required to validate the upgraded Application. The file is in the WebContent folder in the **Package Explorer** pane, as shown in Figure 2-19.



Figure 2-19 Locating the XML Document for Validation

 Right-click the QuoteRequest.xml file, and select Open With > Text Editor to display its contents in the IDE browser, as shown in Figure 2-20.





- 3. Copy all the contents of the source file, and close the file.
- 4. Paste the contents in the **xml requestXML** field of the Test Form tab, as shown in Figure 2-21.

Another way to provide the test values is to click **Browse** in the **Test Form** tab of the browser pane, as shown in Figure 2-18. Subsequently, select the C:\bea\user_projects\workspaces\upgrade\sampleAppWeb\WebContent\request quote\testxml**QuoteRequest.xml** file and click **Open**.

Figure 2-21 Providing the Test Values for Validation



5. Click **quoteRequest** to start the process. On successful completion of the process, the **Test Form** tab is refreshed, as shown in Figure 2-22.

Figure 2-22 Successful Execution

③ Oracle WebLogic Workshop: RequestQuote.jpd	Process 🛛	- B)
🗢 🔿 🔳 🛷 http://localhost:7001/SampleA	AppWeb/requestquote/RequestQuote.jpd?.EXPLORE=.TEST8	a.LOGENTRY=0&.LOGID=12169751: 🔽 🕨 📑
RequestQuote.jpd Pi	rocess	Created by Oracle WebLogic Workshop 📤
Overview Console Test Form Test	SOAP Message Broker Process Graph	http://localhost:7001/SampleAppWeb/ requestquote/RequestQuote.jpd
Start operations Monitor all RequestOuote.jpd processes Continue this conversation		
Message Log 🦉 Refresh	Service Request quoteRequest	
1216975138054 • Monitor • Graph • → quoteRequest Instance 1216975138054 is Running. Ixt Clear Log	Submitted at Fri Jul 25 14:08:58 IST 2008 .CONVERSATIONID = _ID_ .EXECREQUEST = true .CONVPHASE = .START requestML = <xml ?="" version="1.0"> <quoterequest 12="" diy="Morris Plains
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
<widgetRequest>
</widgetRequest>
</widgetRequest>
</widgetRequest></td><td>" rd"="" springs="" xmls="http://www.example.org/request'
<usibondress struet="> * state="NJ" zip="07960"/></quoterequest></xml>	
	Submitted at Eri Jul 25 14:08:58 IST 2008	
	ExecResponse: <null>/n</null>	
	Processing Request	
	Submitted at FII Jul 25 17:00:50 151 2000	×

The WLI 8.x Application is now successfully imported, published, and validated in the Oracle WebLogic Integration 10g Release 3 (10.3.1) workspace.

Post-Upgrade WLI Source Artifacts

This section summarizes the post-upgrade Oracle WebLogic Integration source artifacts.

- The following source file types are renamed to have a . java file extension:
 - DTF: Data Transformation File
 - JPD: Java Process Definition file
 - JCS: Java control source file
 - JCX: Java control extensions file
- The WLI 8.x XQuery files are updated with a comment, indicating that they belong to the XQuery version 2002. This comment helps differentiate the XQuery files as the Oracle WebLogic Integration 10g Release 3 (10.3.1) release uses the XQuery version 2004 as the default.
- All controls, Oracle WebLogic Integration included, are converted to use Apache Beehive with a . java file extension.
- The WLI 8.x channel files are moved to the Oracle WebLogic Integration 10g Release 3 (10.3.1) Utility projects although they are not modified in any manner.
- The upgraded application is adapted to the Oracle WebLogic Integration 10g Release 3 (10.3.1) workspace, with an additional project that is created during upgrade. This additional EAR project combines the other projects into a J2EE application.

Tutorial: Upgrading the WLI 8.1 or WLI 8.5 Application Source



3-1

Tutorial: Upgrading the WLI 9.x and 10.x Application Source

This section contains information on how to upgrade the WLI 9.x or WLI 10.x application source and view it in the Oracle WebLogic Integration 10g Release 3 (10.3.1) environment. This tutorial does not include validation of all the aspects of the sample application. Oracle WebLogic Integration supports binary compatibility upgrade from WLI 9.x or WLI 10.x to Oracle WebLogic Integration 10g Release 3 (10.3.1). For more information, see Compatibility Statement for Oracle WebLogic Server.

Before you upgrade the WLI 9.x or 10.x application source, ensure that you do the following:

• In Oracle Workshop for WebLogic, create a WLI 9.x or WLI 10.x runtime in the Oracle WebLogic Integration 10g Release 3 (10.3.1) environment.

To create a new runtime, select **File > New > Other > Server > Server > Next**. Select **Oracle WebLogic Server v 10.3** and click **Next**. Use the wizard to define the paths to WebLogic Home and Domain home for the new WLI 9.x or WLI 10.x runtime.

- **Note:** If the WLI 9.x or WLI 10.x runtime is created after you import a WLI 9.x or WLI 10.x project, you must invoke a project clean and rebuild to remove the project errors. To invoke a project clean and rebuild, select **Project > Clean**.
- Close the WLI 9.x or WLI 10.x Workshop perspective if it is open before you close the project in WLI 9.x or WLI 10.x. In WLI 9.x or WLI 10.x, right click on the Workshop perspective and click Close. In Oracle WebLogic Integration 10g Release 3 (10.3.1), the default perspective is the J2EE perspective.
- Ensure that all projects in your workspace are open. You can close a project in the IDE and it still remains in the workspace. Closed projects might create errors when you open the workspace in Oracle WebLogic Integration 10g Release 3 (10.3.1).

To import the WLI 9.x or WLI 10.x project, select **File > Import > General > Existing Projects into Workspace**. Specify the path to the project or archived project. Workshop opens the project and first identifies and lists the components of the project that require upgrade as shown in Figure 3-1. Select Finish to upgrade the projects to Oracle WebLogic Integration 10g Release 3 (10.3.1).

Figure 3-1 Project Upgrade

🖸 Project Upgrade 📃 🗖 🗙
Project Upgrade Upgrade projects created with older release of Eclipse and/or related plugins.
The following projects need to be modified in order to be used with this Eclipse installation. Once modified these projects may not be usable with older Eclipse installations. Do you want to continue? Sample_Util SampleApp_Ear SampleApp_Web
⑦ Finish Cancel

Once the project is modified, it appears in the Project Explorer pane. Workshop builds the workspace for the application in the IDE. A log of the process is displayed in the Problems pane at the bottom of the dialog box containing all the errors encountered during the application upgrade to Oracle WebLogic Integration 10g Release 3 (10.3.1). You must resolve all the errors before you can deploy and publish the project.

Note: For the purpose of this tutorial, you can ignore the warning and information type error messages displayed in the Problems pane.

To deploy, publish, and validate the project:

- In the Package Explorer pane:
 - Navigate through the directory structure to view the sampleAppWeb > src > requestquote > RequestQuote.java file.
 - Right click on the file and select **Run As > Run on Server.**

- Select WebLogic Server 10.3 from the list of existing servers and click Finish in the Run on Server window.
- After successfully deploying and publishing the Application, a browser pane is opened in the IDE, displaying the **Overview** tab of the **RequestQuote.jpd** process.
- Click the **Test Form** tab in the browser pane.
- Locate the **QuoteRequest.xml** file that contains the test values required to validate the upgraded Application. The file is under the WebContent folder in the **Package Explorer** pane. Right click the **QuoteRequest.xml** file and select **Open With > Text Editor** to display its contents in the IDE browser.
- Copy all the contents of the source file and paste the contents into the **xml requestXML** field of the Test Form tab.
- Click **quoteRequest** to start the process. On successful completion of the process, the **Test Form** tab is refreshed, and the instance is reported as completed.

You have now successfully imported, published, and validated an WLI 9.x or WLI 10.x application in the Oracle WebLogic Integration 10*g* Release 3 (10.3.1) workspace.