BEA WebLogic® Integration

Tutorial: Upgrading WLI 8.1 or Higher and WLI 9.2 Application Source to WLI 10.2

Version: 10.2
Document Date: March 2008

Contents

CHAPTER 1

Tutorial: Overview

This tutorial describes how to upgrade BEA WebLogic Integration™ 8.1 (Service Pack 4 or 5) and WLI 9.2 applications, to the BEA WebLogic Integration 10.2 environment. This tutorial describes the upgrade scenario using a Process Application sample created in BEA WebLogic Platform™ 8.1.

This section includes the following:

Tutorial Objective and Scope

You must have in depth knowledge of WebLogic Platform, in particular, WebLogic Integration 8.1 and WLI 9.2. The objective of this tutorial is to help you upgrade your existing WebLogic Integration applications to WebLogic Integration 10.2. 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 Workshop for WebLogic Platform workspace.

Upgrading an application to WLI 10.2 involves the following steps:

- Creating the Application in WLI 8.1 or WLI 9.2

  Create the Application in WLI 8.1 or WLI 9.2 as required, and replace the deprecated APIs for smooth upgrade to WLI 10.2.

- Importing the Application into the WLI 10.2 environment, using the Import Wizard

  Import the Application including all its projects using the wizard.

- Deploying the Application in WLI 10.2

  Create a server with 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 8.1 or WLI 9.2 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 8.1 or WLI 9.2 to WLI 10.2 is handled by an import wizard. The wizard does not alter the logic and intent of the existing WLI 8.1 or WLI 9.2 application, nor extract the application from any source repository. The wizard migrates the WLI 8.1 or WLI 9.2 source artifacts into the WLI 10.2 source and project model. However, it retains the WLI 8.1 or WLI 9.2 Javadoc annotations as they do not require any special processing in WLI 10.2. 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 8.1 or WLI 9.2 project types to WLI 10.2 project types.

- Convert WLI 8.1 Javadoc annotations to JSR 175 compliant annotations.
- Move the shared libraries from the WLI 8.1 or WLI 9.2 application directory, to the J2EE EAR project that is part of the upgraded application.
- Move JSP files into a WebContent directory in the WLI 10.2 environment.
- Upgrade NetUI JSP tags to Apache Beehive JSP tags, or just update existing NetUI JSP tags to the WLI 10.2 compatible NetUI version JSP tags.
- Move XSD files that are in a WLI 8.1 or WLI 9.2 Schema project into a WLI 10.2 Utility project.
- Move Java packages and sources into src folders.

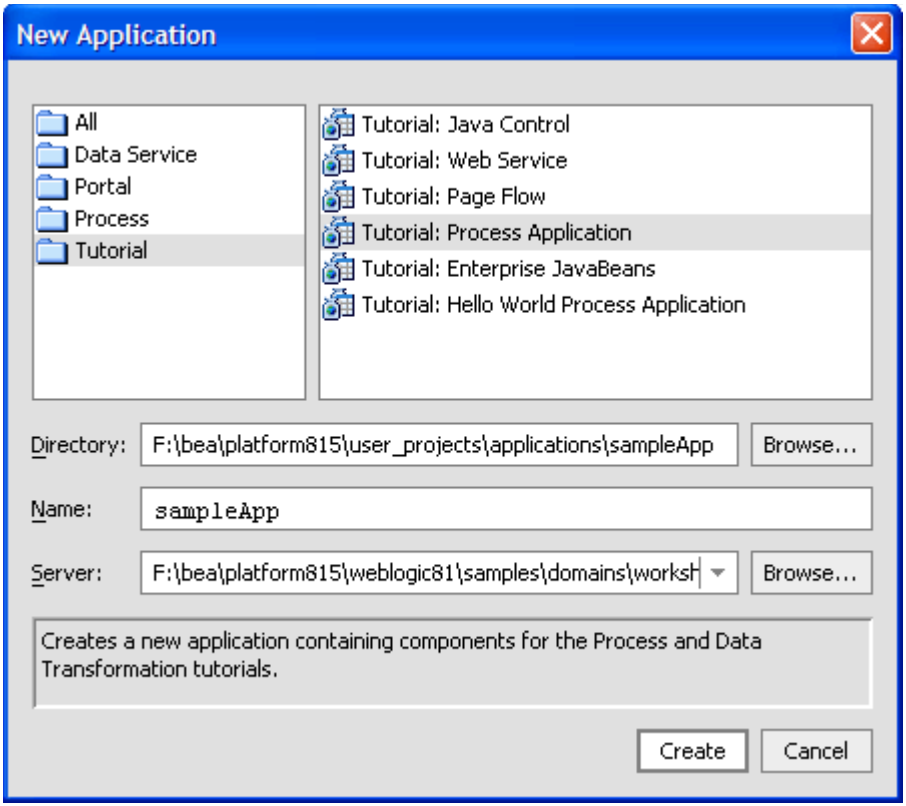
Creating the Source WLI 8.1 Application

For the purpose of this tutorial, you must first create a Request Quote application in the WebLogic Platform 8.1 Service Pack 5 release. Applications created using Service Pack 4 or 6 can also be upgraded using this tutorial.

To create the new application in WLI 8.1, complete the following steps:

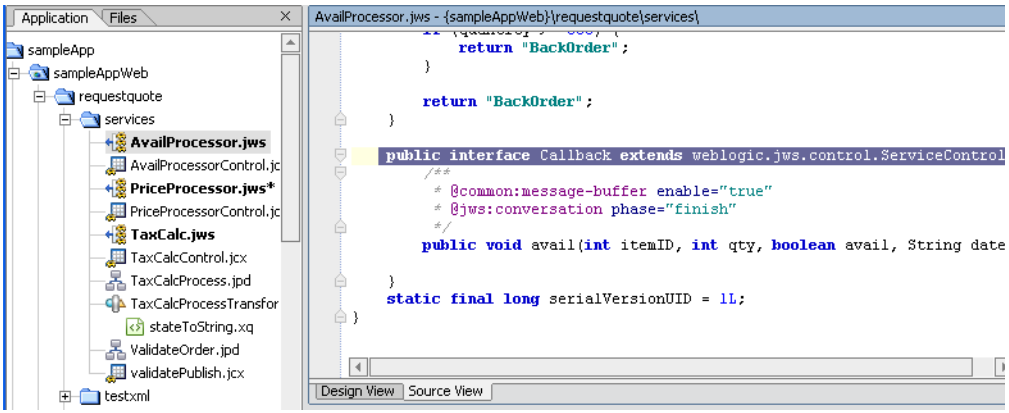
1. In WebLogic Workshop 8.1 select the **File > New > Application...** menu option. The New Application dialog box is displayed.
2. In the left pane select **Tutorial** and in the right pane, select **Tutorial: Process Application**.
3. In the **Name:** field, type the name **`sampleApp`**. The New Application dialog box is displayed as shown in Figure 1-1.

Figure 1-1 Creating a New Application in WebLogic Platform 8.1



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.

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 10.2.

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 Workshop for WebLogic Platform 10.2 IDE.

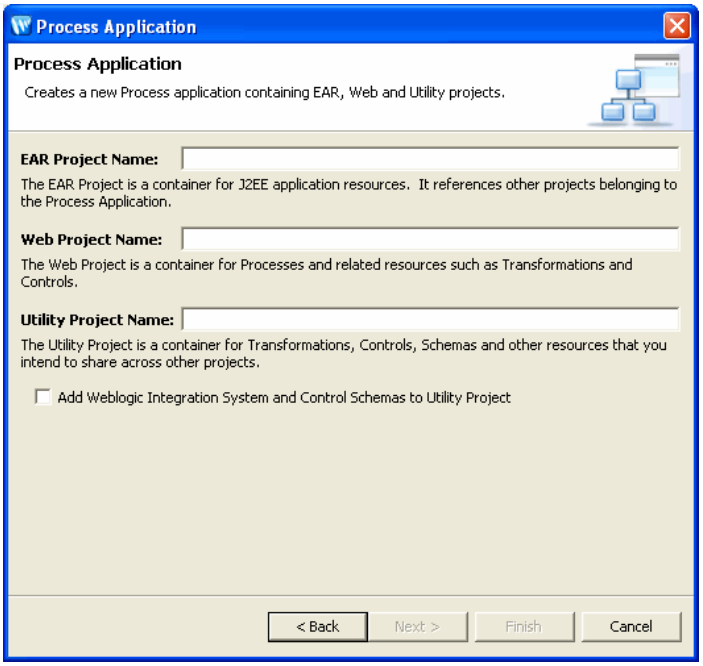
Creating the Source WLI 9.2 Application

For the purpose of this tutorial, you must first create a Request Quote application in the WebLogic Integration 9.2 release. The primary goal of this task is to create an Application in WLI 9.2 and upgrade into the WLI 10.2 release.

If you do not have an existing project in WLI 9.2, you need to create the new application in WLI 9.2. To do this, complete the following steps:

1. In WebLogic Workshop 9.2 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



2. Enter an EAR Project name, Web Project name and Utility Project name and click Finish to create the project.
3. Alternatively you can skip the first two steps and select **File > Import > Workshop 8.1 Application** and click Next on the Import dialog box to import the 8.1 RequestQuote application into the workspace.
4. Specify the path to the WLI 8.1 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.
5. 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 9.2.

6. Save and close the project. You have successfully imported an WLI 8.1 application into the WLI 9.2 workspace.

CHAPTER 2

Tutorial: Upgrading the WLI 8.1 or WLI 8.5 Application Source

This section describes how to upgrade the application source. 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:

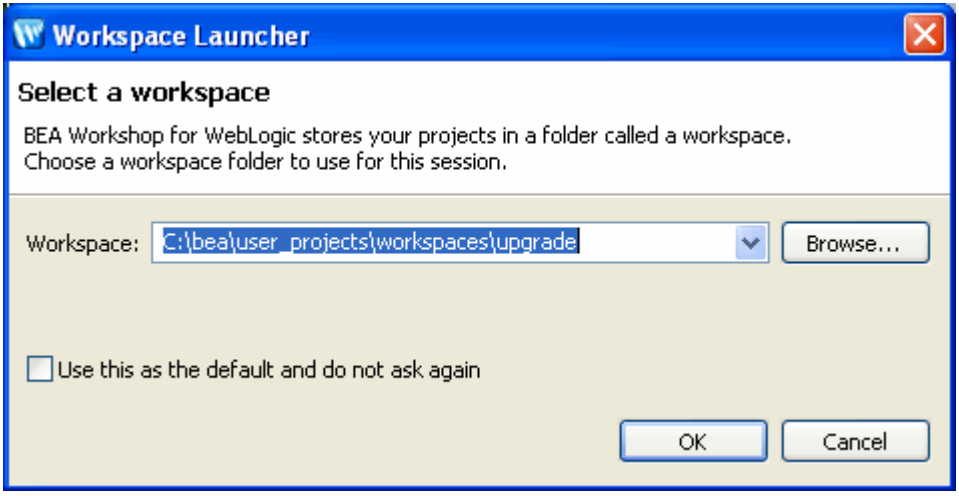
- Receives Request for Quote messages from clients.
- Starts the business process on receipt of the Request for Quote.
- Validates and processes the request.
- Sends the status of the Request for Quote to the client.

Importing the Application Source

This section describes how to upgrade your WebLogic Integration 8.1 Service Pack 4 or 5 application source for use in the Workspace Studio 1.1 environment.

1. Start Workspace Studio by selecting **Start > Programs > BEA > Workspace Studio 1.1** 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



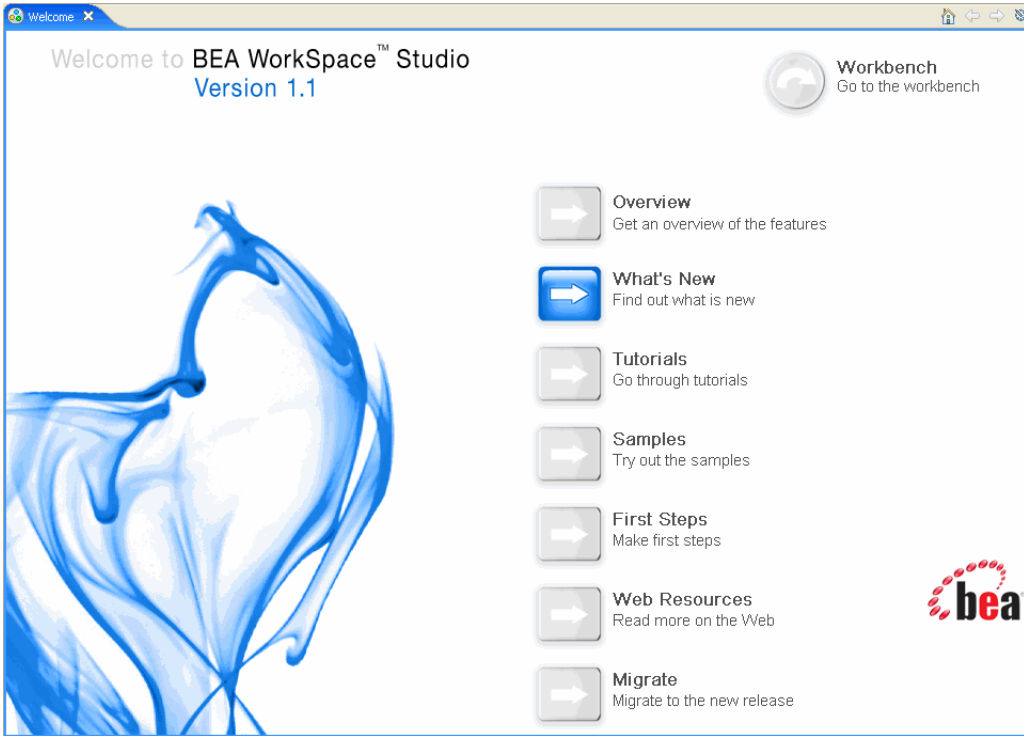
2. Specify the desired location for the upgraded application in the **Workspace** field, as shown in Figure 2-1. For this tutorial, use the `upgrade` workspace, under the WebLogic Platform installation directory, as follows:

   `$BEA_HOME``\user_projects\workspaces\upgrade`

Note: The *`$BEA_HOME`* location used throughout this tutorial is `C:\bea\` where WebLogic Platform 10.2 is installed.

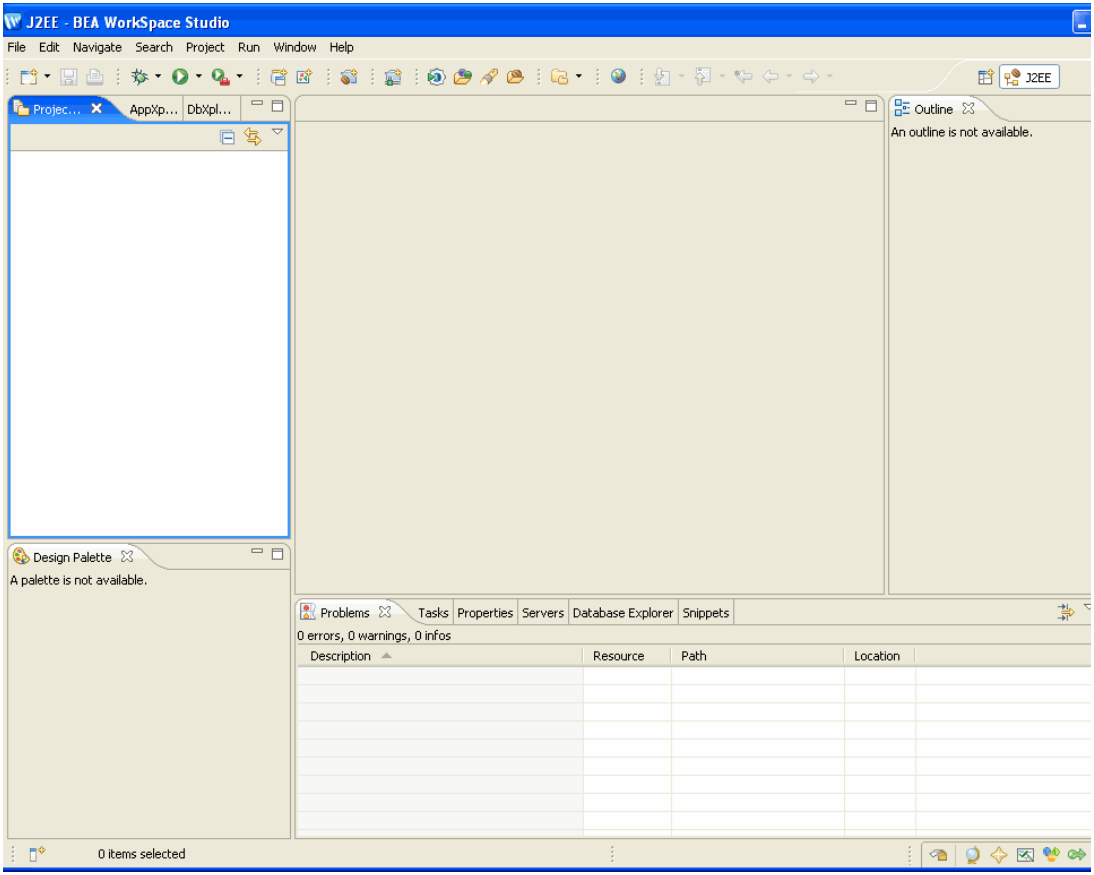
3. Click **OK**. The Workspace Studio 1.1 is launched, as shown in Figure 2-2.

Figure 2-2 Workspace Studio



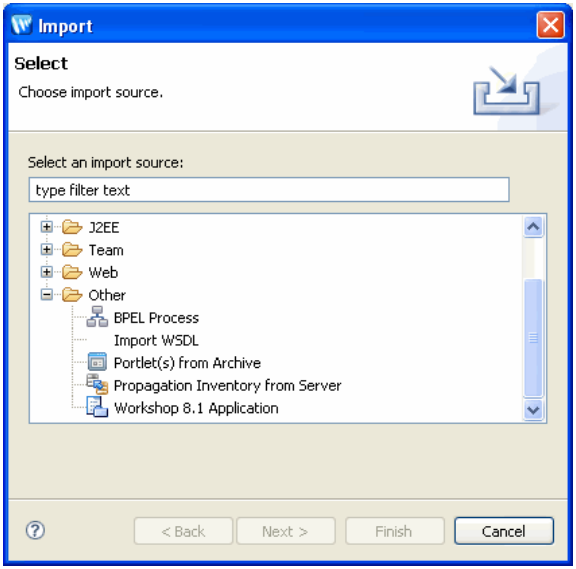
4. Click **Workbench** to proceed to the workbench as shown in Figure 2-3.

Figure 2-3 Workbench



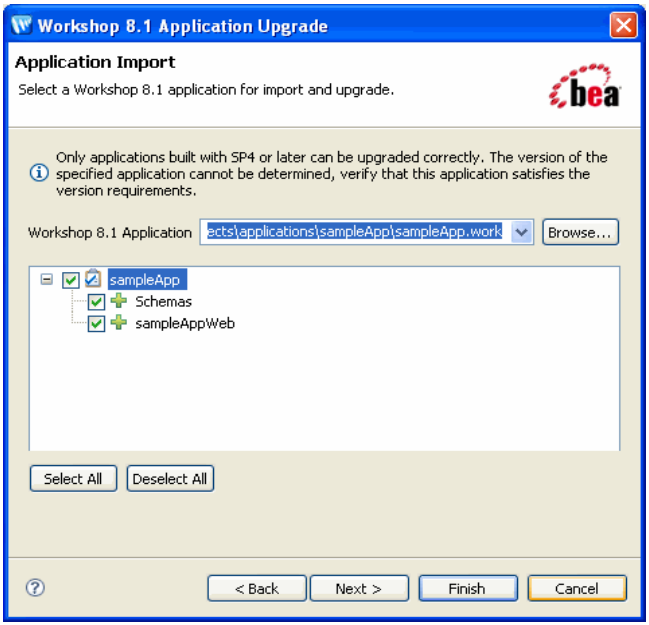
5. Select the **File > Import...** menu option to start the import procedure. The Import Source dialog box is displayed as shown in Figure 2-4.

Figure 2-4 Selecting the Import Source



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



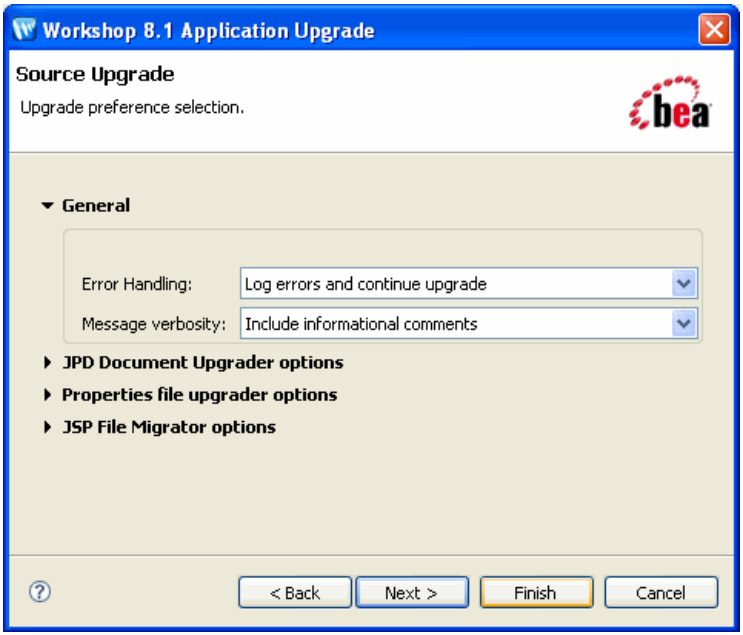
7. Click **Browse** to navigate through the directory structure, and select the **`sampleApp.work`** file that you created in WLI 8.1 SP5 or higher. For more information, see Creating the Source WLI 8.1 Application. As shown in Figure 2-5, the **`sampleApp.work`** file for this tutorial is located in:

   ```
   C:\bea\user_projects\applications\sampleApp
   ```

   After selecting the sample application from the WLI 8.1 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.

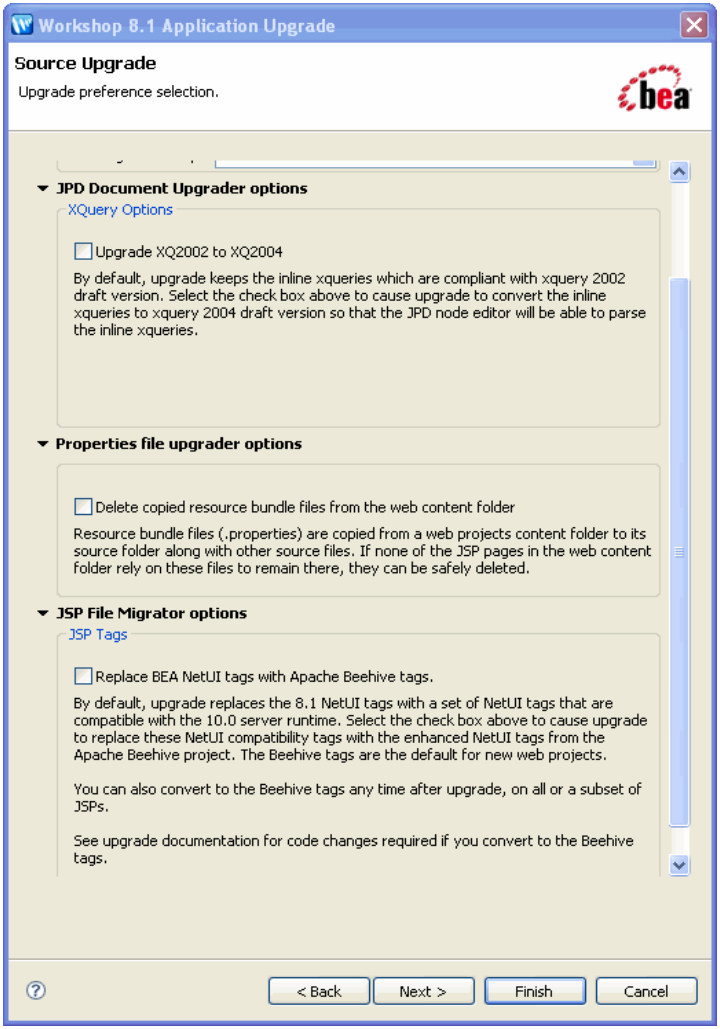
8. Click **Next.** The Source Upgrade preference selection dialog box is displayed, as shown in Figure 2-6.

Figure 2-6 Setting the Upgrade Preferences



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



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.

Table 2-1 Setting General Preferences

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

– **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 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.

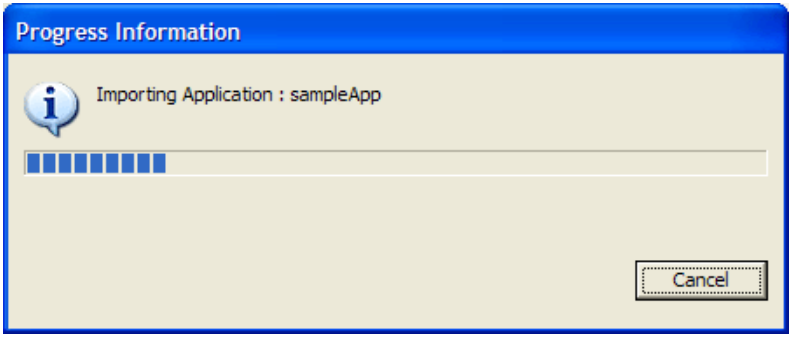
  - Ensure that you select the **Upgrade XQ2002 to XQ2004** option. For more information on upgrading XQuery files, see the Upgrading XQuery Code section of the *Transforming Data Using XQuery Mapper* document.

– **JSP File Migrator options** – When selected, the **Replace BEA NetUI tags with Apache Beehive tags** option upgrades the WebLogic Integration 8.1 supported NetUI tags with Apache Beehive compatible NetUI tags. You can ignore this option as it usually does not impact the 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 WLI 10.2 use the Apache Beehive compatible NetUI tags.

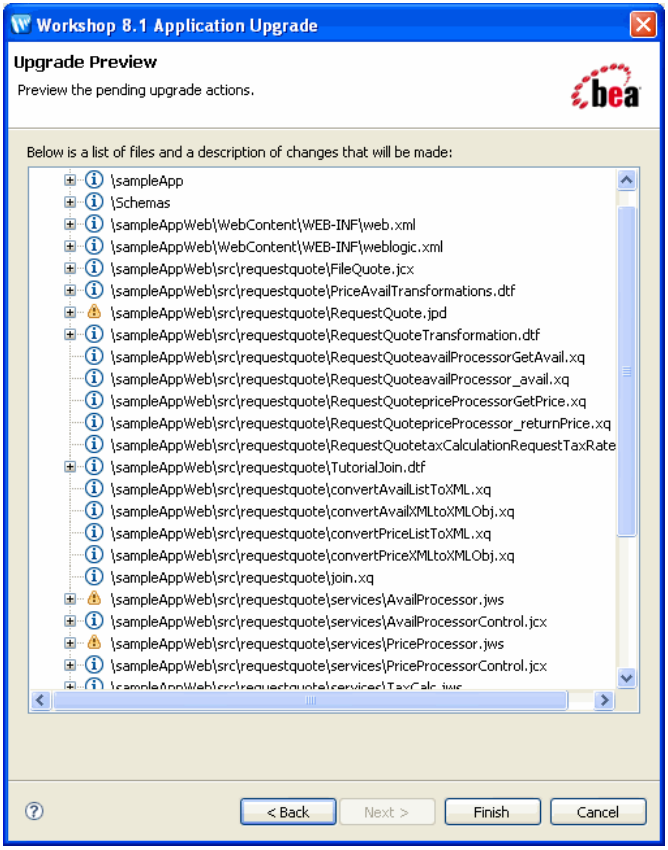
9. Click **Next**. The Progress Information dialog box is displayed, as shown in Figure 2-8. At this step, the application being upgraded is assessed and a detailed list of upgrade tasks that need to be performed (and not performed) is generated.

Figure 2-8 Tracking Upgrade Progress



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.

Figure 2-9 Preview of the Upgrade Actions



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 8.1 application.

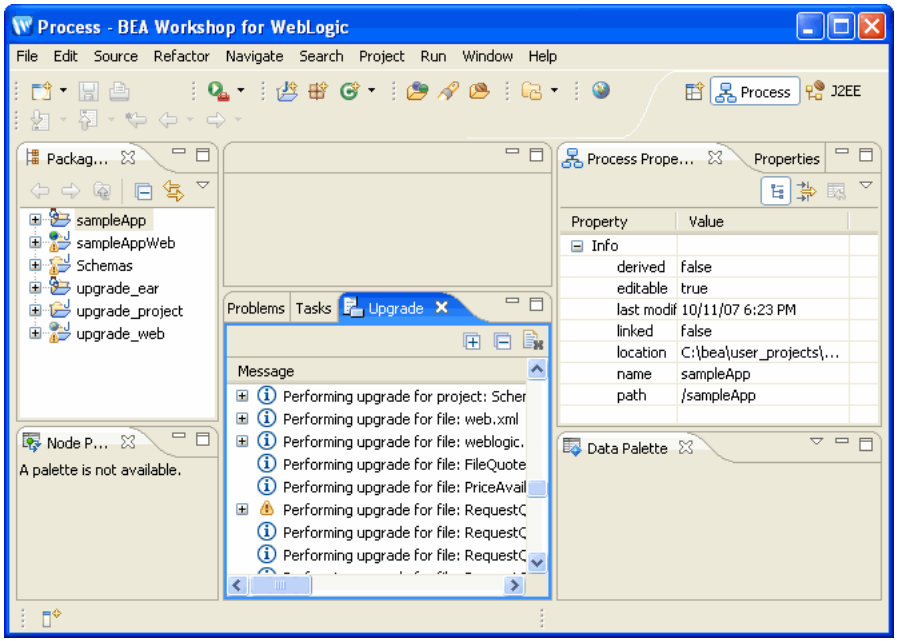
    a. In Project Explorer, browse to the requestquote folder in SampleAppWeb/Webcontent directory. Right click on the requestquote folder and select the Import option.

    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 8.1 application created earlier. Click OK. Select the testxml checkbox 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, as shown in Figure 2-11. You can customize the type of information that is displayed by making the appropriate changes in the Upgrade Preferences dialog box. For more information about the type of error messages to be displayed, see Figure 2-6.

Figure 2-11 Import and Upgrade Progress Log in the BEA Workshop for WebLogic Platform IDE



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 WLI 10.2 environment. The tasks include deploying and running the application in a WebLogic Integration domain.

Validating the Upgraded Application

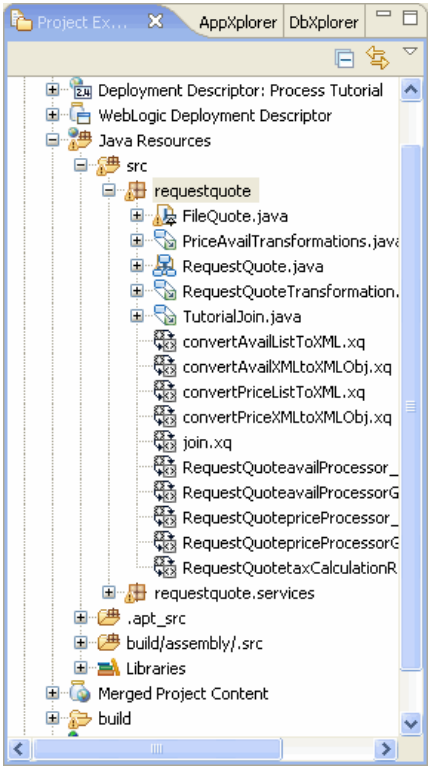
This section describes the steps involved in validating the upgraded application in the WLI 10.2 environment. The task involves deploying the application on the server, publishing it, and subsequently running the application with some test values.

Deploying and Publishing the Application

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

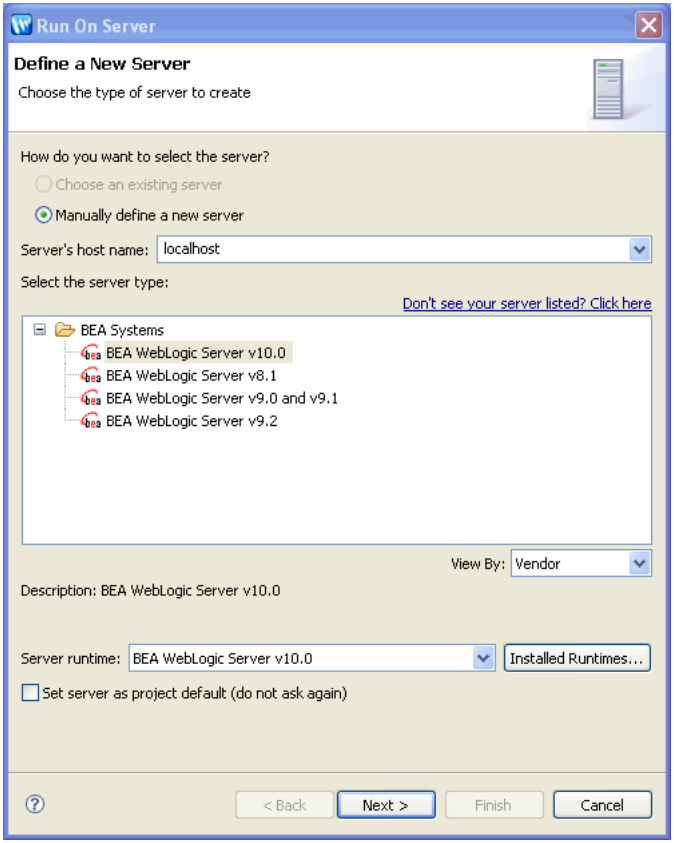
1. In the Package Explorer pane, navigate through the directory structure to view the **sampleAppWeb > src > requestquote > RequestQuote.java** file, as shown in Figure 2-12.

Figure 2-12 Selecting the RequestQuote.java File



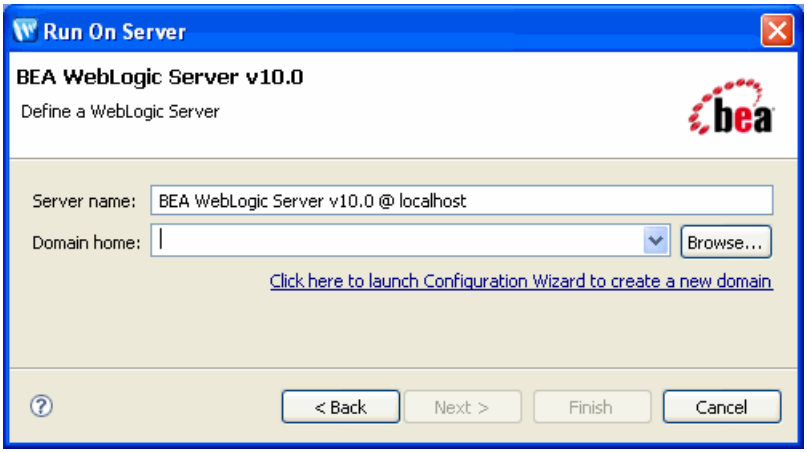
2. Right-click the **RequestQuote.java** file, and select **Run As > Run On Server** menu option. The **Run on Server** dialog box is displayed, as shown in Figure 2-13.

Figure 2-13 Defining a New Server to Deploy the Application



3. Select the **Manually define a new server** check box, and fill out the form, as displayed in Figure 2-13.
4. Click **Next**. The **Run On Server** dialog box is refreshed with the **Define a WebLogic Server** page, as shown in Figure 2-14.

Figure 2-14 Defining a WebLogic Server



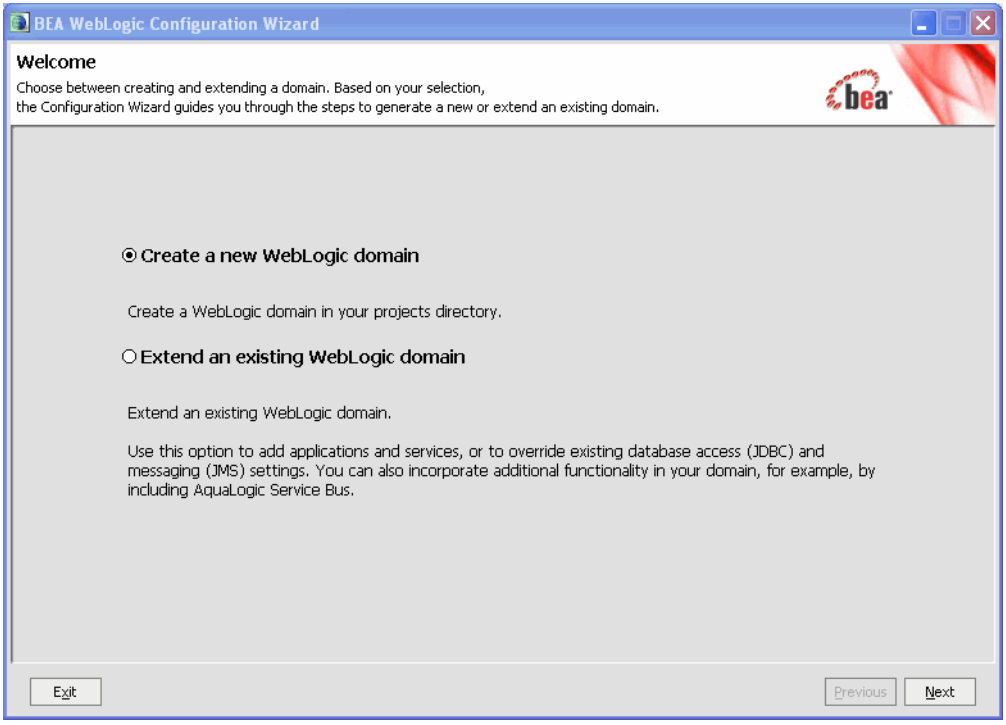
In following section, you will create a new WebLogic Integration domain in the WebLogic Server.

Creating a WebLogic Integration Domain Using the Configuration Wizard

Complete the following steps to set up the 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 BEA WebLogic Configuration Wizard as shown in Figure 2-15.

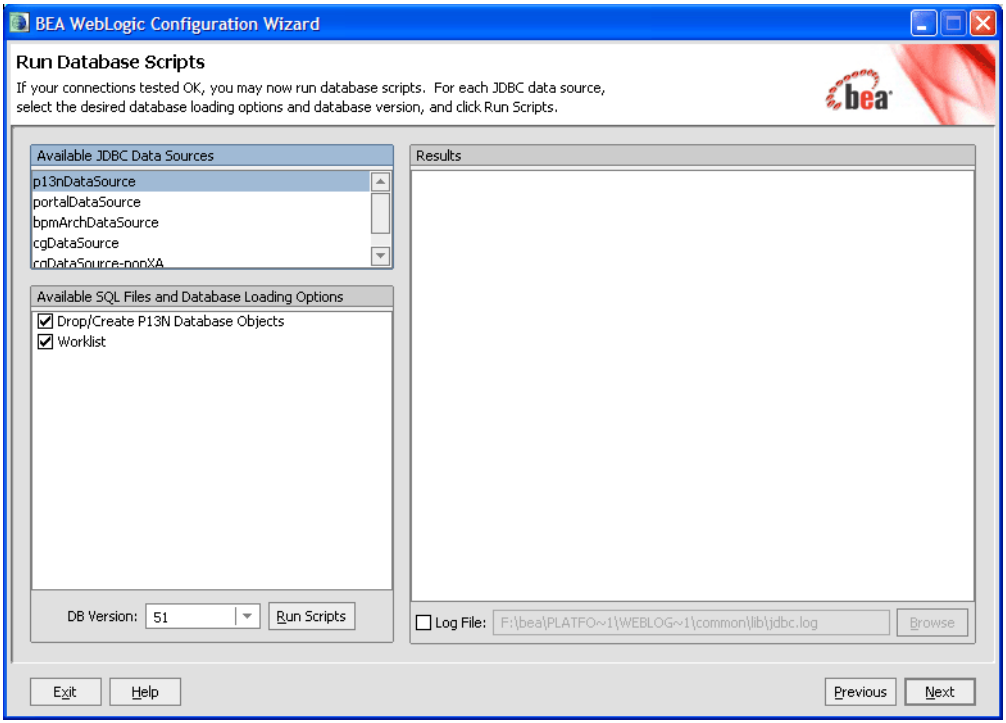
Figure 2-15 Creating a New WebLogic Domain



2. Select the **Create a new WebLogic domain** option, and click **Next**. The Select Domain Source page is displayed.
3. Select the **Workshop for WebLogic Platform** 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.
5. 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.

7. 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 2-16.

Figure 2-16 Running Database Scripts

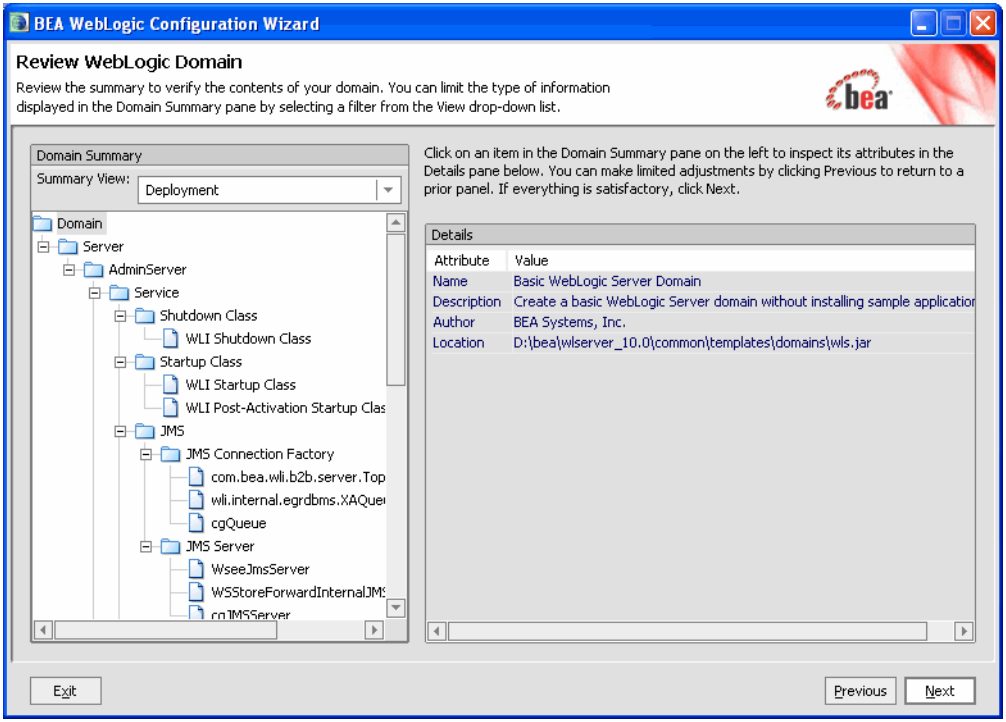


8. From the list of Available JDBC Data Sources, select the **p13nDataSource** option, as shown in Figure 2-16 and click **Run Scripts**. On successful execution, the Results pane will display **`Database Load Successful!`**.

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 2-17, which summarizes the contents of the domain and reflects the options you have selected.

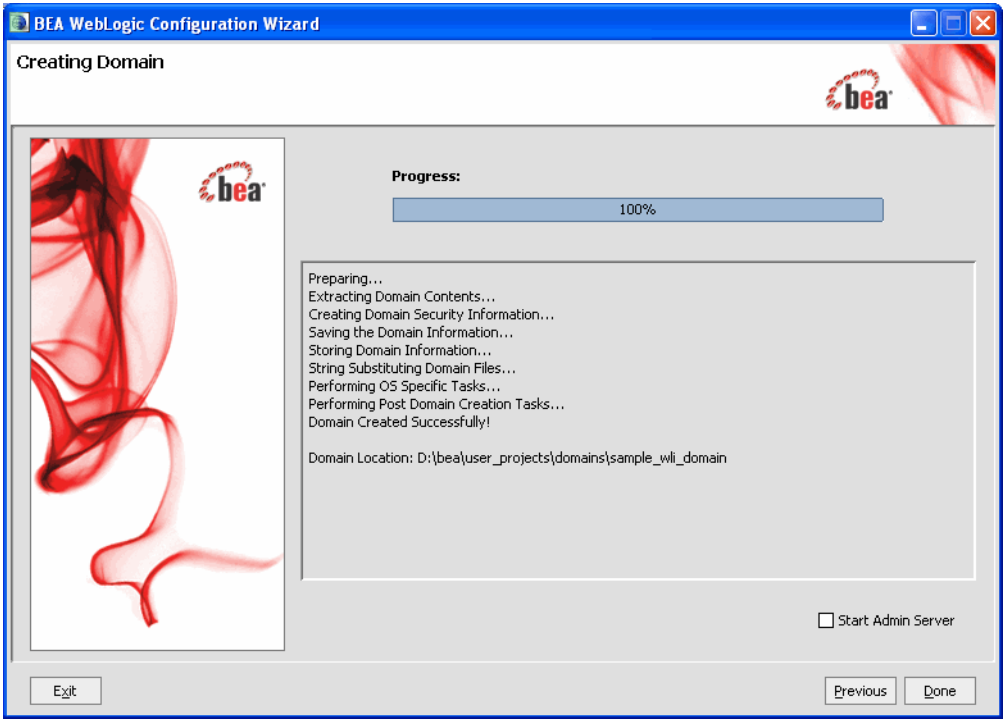
Figure 2-17 Review the Domain Configuration Settings



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 2-18.

Figure 2-18 Successful Creation of a Domain



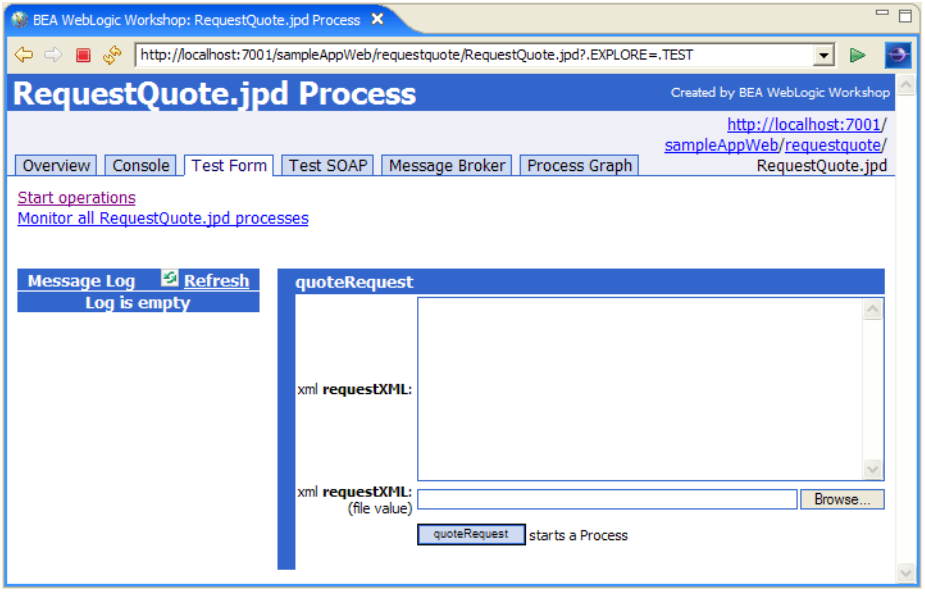
13. Click **Done** to conclude the Domain creation process.
14. After creating the Domain using the Configuration Wizard, return to the Run On Server dialog box as shown in Figure 2-14, to publish the application on the server with the domain you just created.
15. Click **Browse.** Navigate to the folder 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 2-19.

Figure 2-19 Published Application Ready for Validation



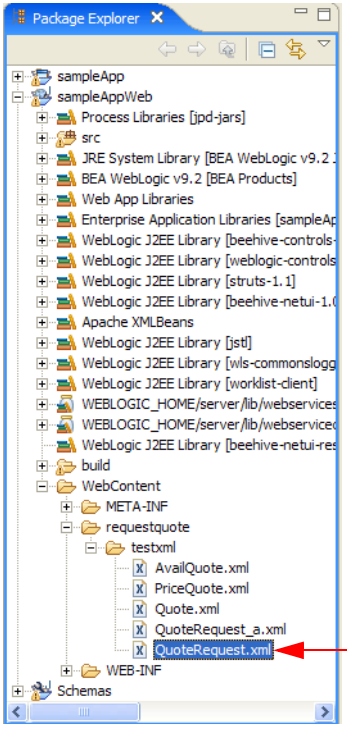
The Application is now ready for the final validation process. The following section contains details on how to run the application.

Running the Application

This is the final step in validating the upgraded Application and the WLI 8.x to WLI 10.2 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.

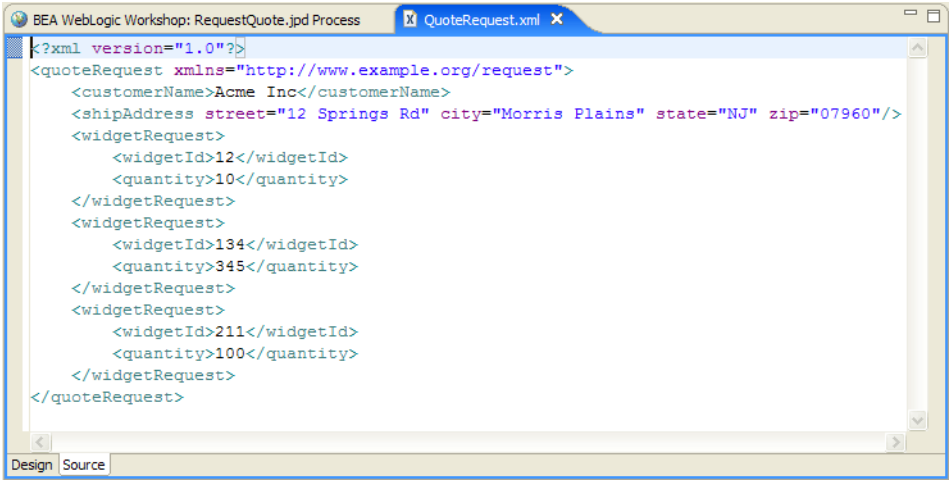
1. Locate the **`QuoteRequest.xml`** file that contains the test values required to validate the upgraded Application. The file is in the `WebContent` folder in the **Package Explorer** pane, as shown in Figure 2-20.

Figure 2-20 Locating the XML Document for Validation



2. Right-click the **`QuoteRequest.xml`** file, and select **Open With > Text Editor** to display its contents in the IDE browser, as shown in Figure 2-21.

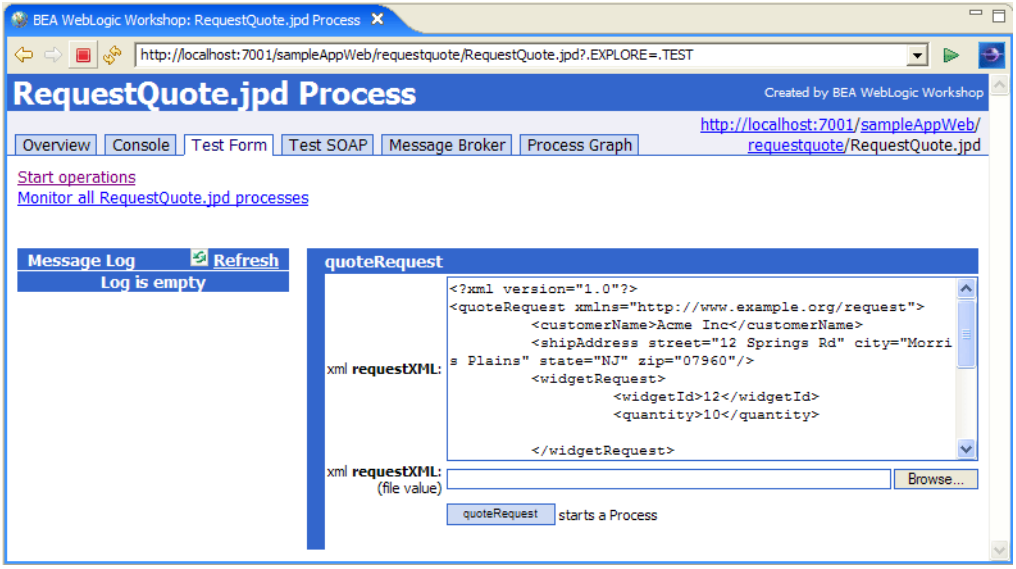
Figure 2-21 Test Values for Validation



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 2-22.

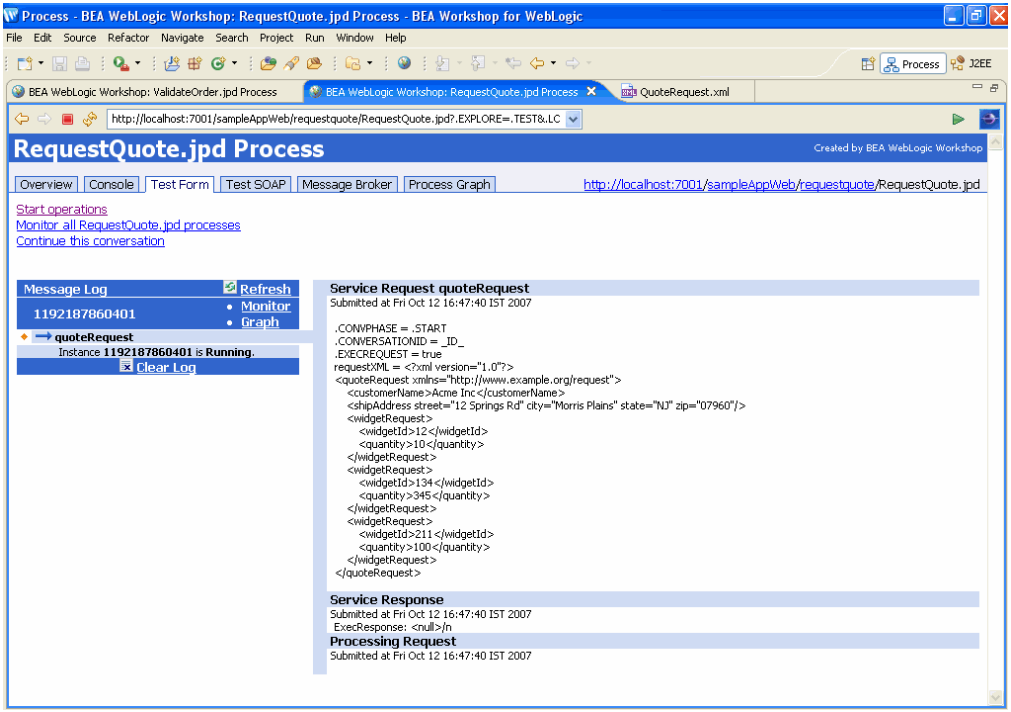
   Another way to provide the test values is to click **Browse** in the **Test Form** tab of the browser pane, as shown in Figure 2-19. Subsequently, select the `C:\bea\user_projects\workspaces\upgrade\sampleAppWeb\WebContent\requestquote\testxml\QuoteRequest.xml` file and click **Open**.

Figure 2-22 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 2-23.

Figure 2-23 Successful Execution



The WLI 8.1 Application is now successfully imported, published, and validated in the WLI 10.2 workspace.

Post-Upgrade WLI Source Artifacts

This section summarizes the post-upgrade 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 8.1 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 WLI 10.2 release uses the XQuery version 2004 as the default.
- All controls, WebLogic Integration included, are converted to use Apache Beehive with a `.java` file extension.
- The WLI 8.1 channel files are moved to the WLI 10.2 Utility projects although they are not modified in any manner.
- The upgraded application is adapted to the WLI 10.2 workspace, with an additional project that is created during upgrade. This additional EAR project combines the other projects into a J2EE application.

CHAPTER 3

Tutorial: Upgrading the WLI 9.2 Application Source

This section contains information on how to upgrade the WLI 9.2 application source and view it in the WebLogic Integration 10.2 environment. This tutorial does not include validation of all the aspects of the sample application. Weblogic Integration supports binary compatibility upgrade from WLI 9.2 to WLI 10.2. For more information, see Compatibility Statement for WebLogic Server.

Before you upgrade the WLI 9.2 application source, ensure that you do the following:

- In Workshop for WebLogic Platform, create a WLI 9.2 runtime in the WLI 10.2 environment.

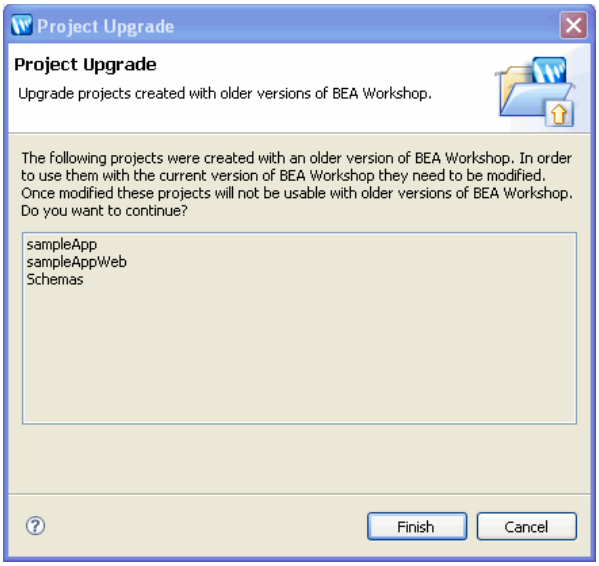
  To create a new runtime, select **File > New > Other > Server > Server > Next**. Select **BEA WebLogic Server v 9.2** and click **Next**. Use the wizard to define the paths to WebLogic Home and Domain home for the new WLI 9.2 runtime.

Note: If the WLI 9.2 runtime is created after you import a WLI 9.2 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 version 9.2 Workshop perspective if it is open before you close the project in WLI 9.2. In WLI 9.2, right click on the Workshop perspective and click Close. In WLI 10.2, 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 WLI 10.2.

To import the WLI 9.2 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 WLI 10.2.

Figure 3-1 Project Upgrade



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 WLI 10.2. 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 10.0 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 9.2 application in the WLI 10.2 workspace.