BEA Workshop for WebLogic Samples

Current Release Information:

- What's New
- Upgrading to 10.0

Useful Links:

- Tutorials
- Tips and Tricks

Other Resources:

- Online Docs
- Dev2Dev
- Discussion Forums
- Development Blogs

Your BEA Workshop for WebLogic Platform installation includes several zipped workspaces containing sample code.

Topics in this section describe the samples provided with Workshop for WebLogic.

Topics Included in This Section

The Samples Workspace

Describes the contents of SamplesWorkspace, one of the sample workspaces provided with Workshop for WebLogic.

Tutorial-based Samples

Describes the contents of the tutorial-based workspaces. Each of these workspaces represents the result of successfully completing a Workshop for WebLogic tutorial.

Beehive NetUI Samples

Describes the contents of the Beehive NetUI samples. These samples represent importing two Beehive NetUI samples (NetUI Samples and JSF Integration) into Workshop for WebLogic web projects.

Opening a Sample Workspace

Explains how to open any of the sample workspaces provided with Workshop for WebLogic.

The Workshop for WebLogic Samples Domain

Describes the resources configured for the Workshop for WebLogic samples server domain.

Related Topics

none.

The Samples Workspace

You'll find a variety of sample workspaces installed with Workshop for WebLogic here:

BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces

One of the workspaces, named SamplesWorkspace, contains a variety of code samples and mini-applications built with Workshop for WebLogic. The remaining workspaces are based on the Workshop for WebLogic Platform tutorials. These tutorial-based samples represent the completed applications that result when you finish one of the tutorials.

For instructions on opening any of the these sample workspaces, see Opening a Sample Workspace.

The SamplesWorkspace contains the following samples. (For information on the tutorial-based samples, see Tutorial-based Samples.)

Control Samples

Database Control: Customer

Custom Control: Customer

Custom Control: Hello World

Web Service Controls (see Web Service samples below)

Web Application Samples

Hello World Web

Customer Web

FormBean Web

Web Service Samples

Synchronous Hello World

Asynchronous Hello World

Secure News Service

JSP Template Project Sample

JSP Template Project

Control Samples

Database Control: Customer

Demonstrates a Beehive database system control. This control performs basic operations on a database: select, insert, update and delete.

Sample Files

File	Role in Sample
SharedControls/src/database/ CustomerDBControl.java	Contains methods that create and query the data table PK_CUSTOMER. This control is used by the Customer Web sample to retrieve customer data. It also works in concert with the Customer Custom Control. This control extends the Beehive DB system control.

Running the Sample

This sample cannot be run independently. Instead run the Customer Web sample to invoke the methods on this control.

Custom Control: Customer

Demonstrates a basic custom control. This control mediates between the Customer Web sample and the Customer database control because its methods wrap the database control methods.

Sample Files

File	Role in Sample
SharedControls/src/database/ Customer.java	A JavaBean that defines the data structure for the Customer object. A Customer object instance corresponds to an individual record in the PK_CUSTOMER data table. An array of Customer objects corresponds to a record set from the PK_CUSTOMER data table.
SharedControls/src/database/ CustomerControl.java	The control interface for the Customer control. Contains wrapper methods for the Customer Database Control. These methods perform basic database operations: select, update, insert and delete.
SharedControls/src/database/ CustomerImpl.java	The control implementation for the Customer control.

Running the Sample

This sample cannot be run independently. Instead run the Customer Web sample to invoke the methods on this control.

Custom Control: Hello World

Demonstrates a simple custom control with one 'Hello World' method.

Sample Files

File	Role in Sample
SharedControls/src/hello/ Hello.java	The control interface class. Contains a single method signature: String hello();
SharedControls/src/hello/ HelloWorld.java	The control implementation class.

Running the Sample

This sample cannot be run independently. Instead run the Hello World Web sample to invoke the methods on this control.

Web Application Samples

Hello World Web

Demonstrates a simple web application. This web application contains one page flow (= a controller class + JSP pages). The page flow invokes the Hello World control and displays the result.

Sample Files

File	Role in Sample
WebApps/src/helloWorld/ HelloWorldController.java	The controller class. This controller class contains one action method, begin(). This method invokes the Hello World control, places the result of the invocation in a action output, and forwards to the index.jsp page.
WebApps/WebContent/ helloWorld/index.jsp	This JSP displays the result of the Hello World control invocation. The result data is passed to the page through an action output/page input.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebApps/src/helloWorld/HelloWorldController.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
   In the dropdown **Domain home** select **<BEA_HOME>/weblogic100/samples/domains/workshop** where <BEA_HOME> is the root directory of your installation.
   Click **Finish**.

   If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

4. The index.jsp page will display the results of calling the Hello World control.

Customer Web

Demonstrates a customer data management web application. This web application allows web access to a customer database.

Sample Files

File	Role in Sample
WebApps/src/database/ DatabaseController.java	The controller class. This controller class contains one action method, getCustomers(). This method invokes the Customer control, places the result of the invocation in a action output, and forwards to the index.jsp page.
WebApps/WebContent/ database/index.jsp	This JSP displays the result of the Customer control invocation. The result data (a record set from a database) is passed to the page through an action output/page input.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebApps/src/database/DatabaseController.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
   In the dropdown **Domain home** select **<BEA_HOME> / weblogic100 / samples / domains / workshop** where <BEA_HOME> is the root directory of your installation.
   Click **Finish**.

   If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

4. The index.jsp page will display the results of calling the Customer control.

FormBean Web

Demonstrates how to submit a multi-selectable list using a form bean.

Sample Files

File	Role in Sample
WebApps/src/formbean/ FormBeanController.java	The controller class. This controller class contains the action method displayChoice() that handles the user submit.
WebApps/WebContent/ formbean/display.jsp	Displays the items selected by the user after submission.
WebApps/WebContent/ formbean/index.jsp	Displays the multi-select form to the user.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebApps/src/formbean/FormBeanController.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
   In the dropdown **Domain home** select **<BEA_HOME>/weblogic100/samples/domains/workshop** where <BEA_HOME> is the root directory of your installation.
   Click **Finish**.

   If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

Web Service Samples

Synchronous Hello World

Demonstrates a simple, synchronous web service, web service control, and client.

Sample Files

File	Role in Sample
WebServiceClients/src/clients/HelloWorldControlClient.java	Client web service for HelloWorld.java (via the intermediary control HelloWorldServiceControl.java)
WebServiceClients/src/controls/HelloWorldServiceControl.java	Intermediary control between HelloWorldControlClient.java and HelloWorld.java
WebServices/src/services/HelloWorld.java	Base web service for this sample.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebServiceClients/src/clients/HelloWorldControlClient.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
   In the dropdown **Domain home** select **<BEA_HOME>/weblogic100/samples/domains/workshop** where <BEA_HOME> is the root directory of your installation.

Click **Finish**.

If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

4. In the **WebLogic Test Client**, click **getHelloWorldMessage**.

Asynchronous Hello World

Demonstrates an asynchronous web service, web service control, and client.

Sample Files

File	Role in Sample
WebServiceClients/src/clients/HWCallbackClient.java	Client web service for HWCallback.java (via the intermediary control HWCallbackServiceControl.java)
WebServiceClients/src/controls/HWCallbackServiceControl.java	Intermediary control between HWCallbackClient.java and HWCallback.java
WebServices/src/services/HWCallback.java	Base web service for this sample.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebServiceClients/src/clients/HWCallbackClient.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
In the dropdown **Domain home** select **<BEA_HOME>/weblogic100/samples/domains/workshop** where <BEA_HOME> is the root directory of your installation.
Click **Finish**.

   If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

4. In the **WebLogic Test Client**, enter a value in the **name** field and click **start**.

Secure News Service

Demonstrates a secure asynchronous web service, web service control, and client. Features role-based access filters and a secure callback from the base web service to the client.

Sample Files

File	Role in Sample
WebServiceClients/src/clients/security/NewsServiceControlCaller.java	Client web service for SecureNewsService.java (via the intermediary control NewsServiceControl.java)
WebServiceClients/src/controls/NewsServiceControl.java	Intermediary control between NewsServiceControlCaller.java and SecureNewsService.java
WebServiceClients/src/controls/secureCallbackServerNewsServiceService.wsdl	WSDL for the web service SecureNewsService.java
WebServices/src/services/security/SecureNewsService.java	Base web service for this sample.

Running the Sample

1. Open the SamplesWorkspace according to these instructions.

2. Right-click on WebServiceClients/src/clients/security/NewServiceControlCaller.java and select **Run As > Run on Server**.

3. If you *have not* already set up a server, follow these instructions: In the **Run On Server** dialog, select **Manually define a new server**, confirm that the node **BEA Systems > BEA WebLogic v10.0 Server** is selected and click **Next**.
   In the dropdown **Domain home** select **<BEA_HOME>/weblogic100/samples/domains/workshop** where <BEA_HOME> is the root directory of your installation.
   Click **Finish**.

   If you *have* already set up a server, follow these instructions: In the **Run On Server** dialog, confirm that the node **localhost > BEA WebLogic v10.0 Server** is selected and click **Finish**.

4. In the **WebLogic Test Client**, enter the username/password pair **weblogic/weblogic** and click **subscribeToNews**.

JSP Template Project Sample

JSP Template Project

Demonstrates the structure of a JSP template project. JSP template projects are used to populate the list of available JSP/JSF templates. For syntax rules for a JSP template project see Authoring JSP Template Projects.

Sample Files

File	Role in Sample
TemplateProject/templateProject.xml	The template project configuration file. This file defines one JSP template. (Note that Multiple templates may be defined.)
TemplateProject/images/	Image directory for the template.
TemplateProject/WebContent/index.jsp	The template page. This page is copied each time a new JSP page is created and this template is selected.

Running the Sample

This sample cannot be run independently. See any of the JSP pages in the WebApps project (inside WebApps/WebContent/) for examples of JSP pages created with this template.

When you open the SamplesWorkspace, none of web application projects (WebApps, WebServices, and WebServiceClients) use this template as their default JSP template. To set this template project as the default JSP template for these projects. Right-click one of the web projects and select **Properties > File Templates > Default JSP Template**. Place a check mark next to **Use Project Settings** and **BEA Branded NetUI JSP**. Click **OK**.

The selected template will be applied whenever a new JSP file is created by selecting **New > Workshop JSP**. Note that the template is *not* applied when you select **New > JSP**.

Related Topics

Opening a Sample Workspace

Tutorial-based Samples

This topic describes each of the tutorial-based samples. Each of these samples represents the results of completing one of the Workshop for WebLogic tutorials. For instructions on running these samples, see the sample's corresponding tutorial.

AdvancedWSTutorial

This is the completed result of the Advanced Web Service Tutorial. It features web services and web service controls.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/AdvancedWSTutorial

EJBTutorial

This sample is what you'd have if you completed the EJB Tutorial. The sample features very simple entity and session beans.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/EJBTutorial

JSFTutorial

This is the result of the JSF Tutorial. This sample features Beehive NetUI used along with Java Server Faces.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/JSFTutorial

TimerTutorial

This sample is the completed result of the Timer Tutorial.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/TimerTutorial

WebAppTutorial

The complete result of a tutorial on calling a database control from a web application, this sample features Beehive technologies, custom and system controls, and data access.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/WebAppTutorial

WebSvcTutorial

This is the completed result of the introductory web services tutorial.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.
samples_1.0.0/workspaces/WebSvcTutorial

Note: you may see WSDL validation errors associated with this sample. These errors are incorrect: to turn off WSDL validation, select **Window > Preferences > Validation > uncheck WSDL Validator**.

WorkspaceTutorialGetStart

This sample is the completed result of a "getting started" tutorial through which you can become acquainted with Workshop for WebLogic.

Location: BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.
samples_1.0.0/workspaces/WorkspaceTutorialGetStart

Related Topics

Tutorials in Workshop for WebLogic

Beehive NetUI Samples

Two Beehive NetUI samples (NetUI Samples and JSF Integration) have been packaged as Workshop for WebLogic web projects. This gives you a chance to explore the Beehive sample code using the Workshop for WebLogic development tools. It is especially useful to explore these samples using the Page Flow perspective and the JSF related tools.

Note that minimal changes have been made to the Beehive sample code. Only a few JSP validation errors have been corrected (because Eclipse JSP validation is especially scrict). The warnings generated by the Beehive sample code have been left in place. These warnings do not block you from deploying and running the sample code.

The Beehive samples are located in a ZIP archive at:

BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/BeehiveSamples/BeehiveSamples.zip

For instructions on opening the Beehive samples, see Opening a Sample Workspace.

To deploy and run the samples, right-click on one of the following files and select **Run As > Run On Server**.

Project	File to Run
netui-samples	netui-samples/src/Controller.java
netui-jsf	netui-jsf/src/Controller.java

Related Topics

Opening a Sample Workspace

Opening a Sample Workspace

The BEA Workshop for WebLogic Platform installation includes several workspaces that contain sample project code. The samples are installed in ZIP files that you can open with Workshop for WebLogic. This topic describes how to import these sample projects into Workshop for WebLogic.

You'll find a list of the installed sample workspaces at The Samples Workspace and Tutorial-based Samples.

How to Open a Sample Workspace in Workshop for WebLogic

To open a sample workspace for the first time, or to recreate a samples workspace in its original state, follow these instructions:

1. Select **File > Switch Workspace**. If you want to open the sample projects in their own directory, then create a new empty workspace directory, or select the directory where the sample zip file resides. If you want the sample projects to be co-located with other, already existing projects, then select the workspace where those projects reside.
2. Select **File > Import > General > Existing Projects into Workspace > Select archive file > Browse to the ZIP file BEA_HOME/workshop100/workshop4WP/eclipse/plugins/com.bea.workshop.product.wl.samples_1.0.0/workspaces/<SomeWorkspace>/<SomeWorkspace>.zip and click Open > Click Finish.**

   This will cause the ZIP file to decompress into the workspace directory.

Feel free to modify the samples as you wish. You can always recreate the samples workspace in its original state by running through the procedure above starting with a new empty workspace directory. However, be aware that WebLogic Server does not allow deployment of projects with duplicate names from different locations on disk. See the note below to avoid this pitfall.

Duplicate Project Names Not Supported

WebLogic Server does not support the deployment of projects with the same name from two different locations on disk.

For this reason, anytime you deploy a sample or run through a tutorial, you should check your server for previous deployments of the same code. If your server already has a deployment of the sample or tutorial code, you should either (1) undeploy any previous code, (2) create a new server domain, or (3) deploy the code from the same location as the previous deployment.

Related Topics

none

The Workshop for WebLogic Samples Domain

This topic describes the samples server domain designed to run the Workshop for WebLogic samples.

The Workshop for WebLogic samples (both the SamplesWorkspace and the tutorial-based samples) can be deployed and run on the samples domain, located at:

BEA_HOME/weblogic100/samples/domains/workshop

Many individual samples may not run on other server domains, because of specific resource dependencies.

Domain Resources

The Workshop for WebLogic samples domain is configured as follows:

- The domain contains a single WebLogic Server instance: the administration server for the domain.
- The default listen port is 7001.
- A PointBase database is configured for the sample applications; the JNDI name for the data source is `samplesDataSource`. Two other data sources are configured named `cgDataSource` and `cgDataSource-nonXA`.
- A JMS Server named `WseeJmsServer` is provided.
- Numerous library modules are configured, to support logging, Apache Beehive, Java Server Pages, Java Server Faces, and Apache Struts.
- The startup script for the domain (BEA_HOME/weblogic100/samples/domains/workshop/bin/startWebLogic.cmd|sh) is set to development mode. For more information on development and production modes see Tuning WebLogic Server in the WebLogic Server documentation.

For more configuration details see the main server configuration file at BEA_HOME/weblogic100/samples/domains/workshop/config/config.xml.

Using the Database Explorer

The samples Database Control: Customer and WebAppTutorial automatically create datatables when they are deployed and run. You can explore these datatables by following this procedure:

Window > Show View > Other > Data > Database Explorer > Ok

2. On the **Database Explorer** tab, right-click **Connections** and select **New Connection**.

3. In the N**ew Connection** dialog, in the **Database** field, enter `FooBar` (any value works),
   in the **JDBC driver class** field, enter `com.pointbase.jdbc.jdbcUniversalDriver`,
   in the **Class location** field, browse to `<BEA_HOME>/weblogic100/common/eval/pointbase/lib/pbclient51.jar`,
   in the **Connection URL** field, enter `jdbc:pointbase:server://localhost:9093/weblogic_eval`,
   in the **User ID** field, enter `weblogic`,
   in the **Password** field, enter `weblogic`.
   Click **Finish**.

Deploying and Running the Samples

Instructions for deploying and running samples in the SamplesWorkspace available in the topic The Samples Workspace.

Instructions for deploying and running the tutorial-based samples are available in the topic Tutorial-based Samples.

Related Topics

none