BEA WebLogic Portal™
BEA WebLogic Integration™

WebLogic Portal and WebLogic Integration Example

WebLogic Portal Version 4.0
WebLogic Integration Version 2.1
Document Date: July 2002

Copyright

Restricted Rights Legend

Trademarks or Service Marks

WebLogic Portal and WebLogic Integration Example

Part Number	Date	Software Version
N/A	July 2002	WebLogic Portal 4.0, SP1 WebLogic Integration 2.1, SP1

Contents

1. Overview

2. WebLogic Portal Interoperability Sample

3. WebLogic Integration Interoperability Sample

About This Document

This document describes an example of interoperability between BEA WebLogic Portal™ 4.0 SP1 and BEA WebLogic Integration™ SP 1.

This document includes the following topics:

- Chapter 1, "Overview," which describes the entry points for interoperation between WebLogic Portal and WebLogic Integration, plus an overview of asynchronous and synchronous communication.
- Chapter 2, "WebLogic Portal Interoperability Sample," which describes adding a Pipeline Component to the commerce sample application. The Pipeline Component creates an XML representation of an order and places the representation on a JMS destination. WebLogic Integration can then use this object in a workflow.
- Chapter 3, "WebLogic Integration Interoperability Sample," which describes setting up the environment for WebLogic Integration, making a purchase with the Commerce Templates, and checking the interoperation.

What You Need to Know

This document is intended for Java developers are proficient in Java programming, the J2EE specification, and XML.

e-docs Web Site

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

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the WebLogic Portal documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Portal documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

The following BEA WebLogic Portal documents contain information that is relevant to interoperability between WebLogic Portal and WebLogic Integration.

- For general information about Weblow and Pipeline Components, see the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline* in the WebLogic Portal documentation at http://edocs.bea.com/.
- For information on using the E-Business Control Center (EBCC), see the EBCC online help or the *Guide to Using the E-Business Control Center* in the WebLogic Portal documentation at http://edocs.bea.com/.

- For information about deployment, see the *Deployment Guide* in the WebLogic Portal documentation at http://edocs.bea.com/.
- For information about setting up your development environment, see the *Strategies for Developing E-Business Web Sites Deployment Guide* in the WebLogic Portal documentation at http://edocs.bea.com/.
- For more information on the WebLogic Server Administration Console, see the *BEA WebLogic Server Administration Guide* in the WebLogic Portal documentation at http://edocs.bea.com/.
- For more information about making a purchase in the WebLogic Portal Commerce Templates, see the *JSP Commerce and Campaign Tour* in the WebLogic Portal documentation at http://edocs.bea.com/.
- For more information about domain configuration in WebLogic Integration, see *Starting, Stopping, and Customizing WebLogic Integration* at http://edocs.bea.com/.
- For more information about WebLogic Integration Studio, see *Using the WebLogic Integration Studio* at http://edocs.bea.com/.
- For more information about business process management (BPM), see *Learning to Use BPM with WebLogic Integration* at http://edocs.bea.com/.

Contact Us!

Your feedback on the BEA WebLogic Portal documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Portal documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Portal 4.0 release.

If you have any questions about this version of BEA WebLogic Portal, or if you have problems installing and running BEA WebLogic Portal, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

Convention	Item
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 Overview

BEA WebLogic Portal 4.0 and BEA WebLogic Integration 2.1 provide an integration platform for e-business solutions. To help you learn how to make the best use of this functionality when you use WebLogic Portal and WebLogic Integration together, BEA provides the WebLogic Portal and WebLogic Integration Example.

This topic provides an overview of interoperation scenarios plus an introduction to the WebLogic Portal and WebLogic Integration Integration Example.

This topic includes the following sections:

- About the Example
- Interoperation Scenarios
  - WebLogic Portal Entry Points
  - WebLogic Integration Entry Points
  - Three Scenarios for Interoperation
  - Interoperability Scenario to Avoid
  - Asynchronous Scenario
  - Synchronous Scenario
- Overview of the WebLogic Portal and WebLogic Integration Example

About the Example

WebLogic Portal and WebLogic Integration are part of the BEA E-Business Platform. Each product provides the following features:

- WebLogic Portal is a suite of services and templates that enables a company to build front-end portals, marketing campaigns, e-commerce solutions, and personalized content for Web applications.
- WebLogic Integration is a package of tools and processes that the company can use to integrate those applications with the appropriate back-end enterprise systems, to streamline complex business processes, and to establish connections to business partners.

The WebLogic Portal and WebLogic Integration Example demonstrates the unique functionality produced by interoperating the two products through a scenario involving an online hardware store. The Example is delivered in two parts:

- The *WebLogic Portal* part of the Example is delivered with WebLogic Portal 4.0 Service Pack 1.
- The *WebLogic Integration* part of the Example is available for downloading from the BEA Developer Center by WebLogic Integration 2.1 customers. This part of the Example is in *Interoperating WebLogic Portal & WebLogic Integration* at the following URL:

  `http://developer.bea.com/code/wli.jsp`

Interoperation Scenarios

This section describes three scenarios for WebLogic Integration and WebLogic Portal interoperation. Interoperation is carried out through various entry points in WebLogic Portal and WebLogic Integration. These entry points allow you to create true process-level communication and data flow between Web applications and enterprise systems.

WebLogic Portal Entry Points

WebLogic Portal has the following integration points available for communication with other systems:

- Portlets – Portlets are applications that exist within a portal. JSP-based portlets can be built to integrate with enterprise systems.
- Unified User Profile (UUP) – The UUP in WebLogic Portal can integrate with any external system. The UUP Application Program Interface (API) can also be used by external systems to change, read, update, and delete data contained in the UUP.
- Webflow and Pipelines – The Webflow functionality in WebLogic Portal manages the flow of business logic and site presentation. Webflow contains discrete units of business logic called Pipeline Components, which execute narrowly focused processes. The Webflow/Pipeline Editor in the E-Business Control Center can be used to create XML-based Pipeline Components for integration to any enterprise systems.

WebLogic Integration Entry Points

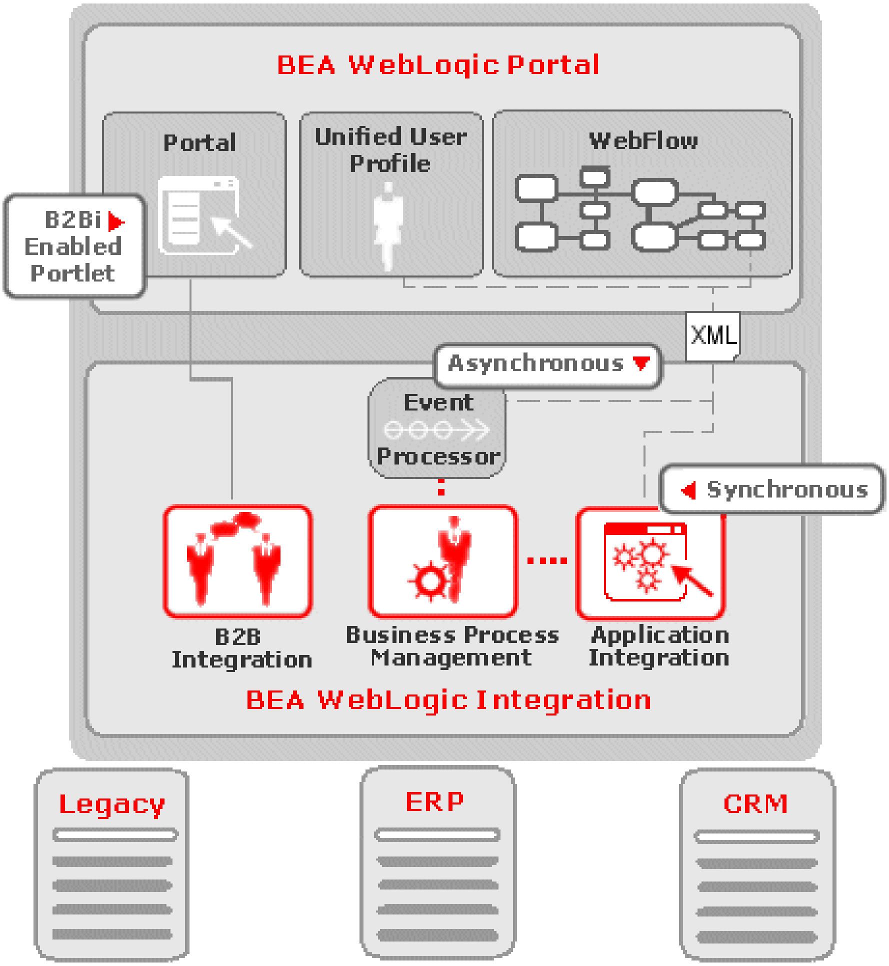
WebLogic Integration has the following entry points available for communication with other systems:

- Application integration (AI) – A system can call a J2EE Connection Architecture (J2EE CA) adapter directly using the AI entry point of WebLogic Integration. Typically, you use this for synchronous communication.
- Business-to-business integration (B2Bi) – A system can directly invoke business-to-business integration through a message API. This provides developers a way to write custom applications that communicate with a full instance of WebLogic Integration. B2Bi also provides a JavaServer Pages (JSP) tag library for building thin-client interfaces that can communicate with a hosted B2Bi enabler.
- WebLogic Integration process engine – A system can invoke the WebLogic Integration process engine synchronously using calls to a native API or asynchronously using JMS. These calls receive and send data, start processes, execute user-assigned tasks, and pass XML data among enterprise systems.

Three Scenarios for Interoperation

Figure 1-1 shows three scenarios for the interoperation of WebLogic Portal and WebLogic Integration.

Figure 1-1 Three Scenarios for Integrating WebLogic Portal and WebLogic Integration



The three scenarios shown in Figure 1-1 leverage the entry points of WebLogic Portal and WebLogic Integration, as follows:

- The B2Bi-enabled portlet is built using the WebLogic Integration JSP tag libraries and allows a user to interact with a WebLogic Integration work list from within the context of an enterprise portal.

  For example, you could create an online marketplace that provides a unified business portal for use by all the supplier partners that access a market hub. WebLogic Portal would provide an interface through a portlet and the B2Bi JSP tag library would provide the functionality.

- The synchronous approach, shows a WebLogic Portal Pipeline Component that directly communicates using XML with a J2EE CA application integration component of WebLogic Integration.

  Typically, synchronous communication is used for bi-directional, lightweight, short-lived transactions between WebLogic Portal and one other enterprise system. Examples include: order status reports, sales forecast inquiries, and package delivery tracking.

- The asynchronous approach shows a WebLogic Portal Pipeline Component generating an XML message. This message is then passed asynchronously to the WebLogic Integration Event Processor for entry into a WebLogic Integration workflow.

  Typically, an asynchronous connection is used to transact more complex interactions between WebLogic Portal and multiple enterprise systems. Some examples include: routing an order through a multistage approval and notification process, automating an employee new-hire workflow, and pushing package delivery information from the consumer origination point through inventory and on to fulfillment.

Interoperability Scenario to Avoid

This section discusses a configuration of interoperation that is not advised and not supported.

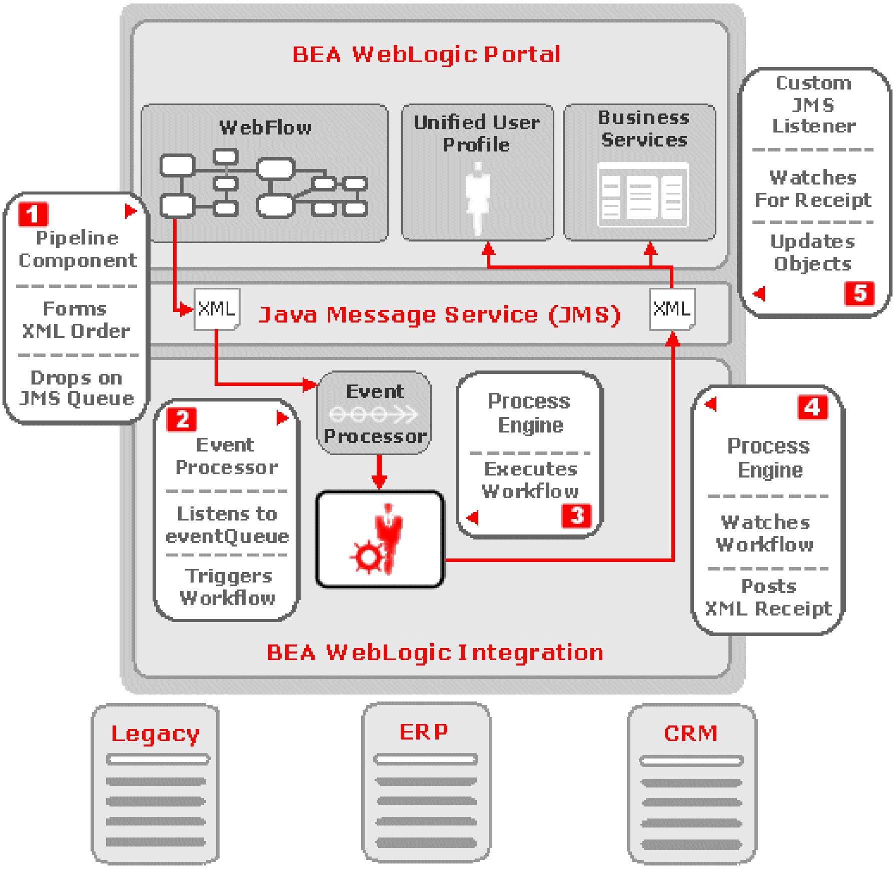
A distributed transaction is a transaction that updates multiple resource managers, such as databases, in a coordinated manner. In WebLogic Portal, the Java Transaction API (JTA) orchestrates a two-phase commit when such a distributed transaction completes.

The two-phase commit ensures that all the participants in the transaction commit their updates simultaneously. JTA coordinates this commit with databases that are updated using Open Group's XA protocol. WebLogic Integration 2.1 supports Open Group’s XA transaction protocol but WebLogic Portal 4.0 does not. Therefore, it is not advised to include WebLogic Portal business objects in distributed XA transactions. However, if you require this configuration, professional services may be able to help you.

Asynchronous Scenario

Figure 1-2 shows the suggested asynchronous, round-trip integration between WebLogic Portal and WebLogic Integration.

Figure 1-2 Asynchronous Round-Trip Integration



Step 1: A commerce order originates in WebLogic Portal's Webflow. A Pipeline Component in that Webflow transforms the order into an XML representation and drops it into a JMS queue in WebLogic Integration.

Step 2: The WebLogic Integration Event Processor listens for XML messages on an internal JMS queue. After receiving the order message, it passes the order message to the appropriate message-driven bean for processing and then on to WebLogic Integration process engine for participation in a workflow.

Step 3: The WebLogic Integration process engine moves the order along a workflow.

Step 4: Once the workflow execution has completed, WebLogic Integration process engine drops an order receipt back on to a JMS queue.

Step 5: A WebLogic Portal-built listener receives the order receipt and updates the WebLogic Portal order message with the new information and also increments a custom frequent-buyer property in the UUP for use in later personalization scenarios.

The WebLogic Portal product provides a sample Pipeline Component the creates an XML message and places it on a JMS queue. This sample is discussed fully in the Chapter 2, "WebLogic Portal Interoperability Sample."

Synchronous Scenario

Because of performance issues, you generally use this method for simple data requests that do not have the possibility of timing out the run-time application. To set up synchronous communication, you need to create a Pipeline Component that communicates directly with WebLogic Integration by invoking an API. WebLogic Integration provides a way to connect enterprise systems to an application server through adaptors you can build. The following documents provide the necessary information:

- To create a Pipeline Component, see Creating a New Pipeline Component in *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*.
- For information on using WebLogic Integration, see the WebLogic Integration documentation at http://edocs.bea.com/wlintegration/v2_1/index.htm.
- For information on building adaptors, see *Introducing Application Integration* in the WebLogic Integration Application Integration documentation.

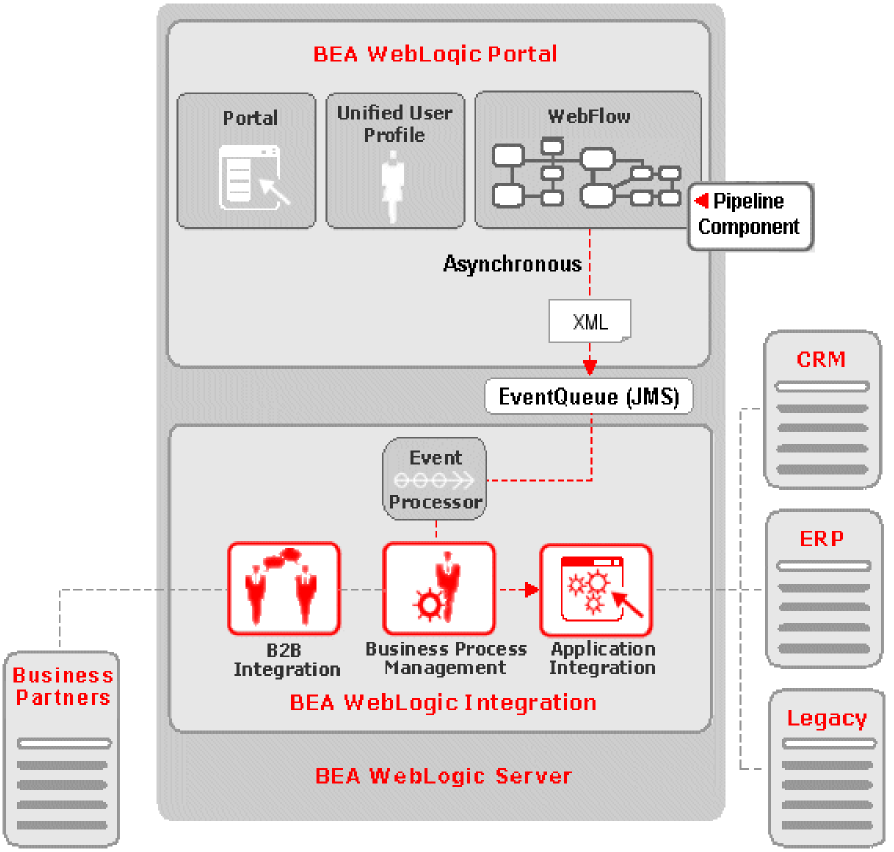
Overview of the WebLogic Portal and WebLogic Integration Example

The WebLogic Portal and WebLogic Integration Example demonstrates asynchronous communication. It consist of two parts: the WebLogic Portal sample, which is included with the WebLogic Portal product, and the WebLogic Integration sample, which can be downloaded from the BEA Developer Center at `http://developer.bea.com/code/wli.jsp`.

The example shows how a purchase order generated in WebLogic Portal can be used in business processes. In the Example, a sample Pipeline Component in the WebLogic Portal commerce application (wlcsApp) converts a purchase order to an XML representation and then places the order representation on a Java Message Service (JMS) queue in WebLogic Integration. The business process management (BPM) component of WebLogic Integration then processes the order. The WebLogic Portal part of the Example demonstrates a catalog and a set of campaigns for an online hardware store. The goal is to enable a user to browse through the online catalog and request a product for purchase.

The following figure shows the process flow for the Example.

Figure 1-3 Process Flow of the Integration Example



When a customer submits a request, a purchase order is generated in WebLogic Portal. A WebLogic Portal Pipeline Component called `ConvertOrderRepPC` converts the order to a JMS XML message and sends it to a JMS queue (`com.bea.wlpi.EventQueue`) to which a BPM workflow is subscribed. Subsequently, the BPM event listener retrieves the message from the JMS queue and processes the message. The message either starts a workflow or triggers a workflow event listened to by a running workflow instance. The WebLogic Integration part of the Example retrieves the XML message from the JMS queue and forwards the data to a database, using the sample DBMS adapter provided by WebLogic Integration for use with its application integration component.

The XML message is used as the input document to the workflow. The workflow responds with two actions:

- It parses the XML message using XPath and passes all the input data to the application integration service.
- It calls an application integration service that is defined when the user deploys an application view for the DBMS adapter.

The application view service updates the database by updating two tables created by the user: `WLI_ORDER` and `WLI_ORDER_LINES`. Both tables reside in the WebLogic Integration repository.

2 WebLogic Portal Interoperability Sample

This topic provides instructions for setting up an example of interoperability between BEA WebLogic Portal 4.0 and BEA WebLogic Integration 2.1 that uses asynchronous communication. As previously mentioned, included with WebLogic Portal is a sample Pipeline Component (`ConvertOrderRepPC`). This Pipeline component converts a purchase order generated by the commerce sample application in WebLogic Portal to an Extensible Markup Language (XML) representation and places the order representation on a Java Message Service (JMS) queue (`eventQueue`) in WebLogic Integration. The WebLogic Integration sample picks up the message and uses it in business processes. In the following sections, you will create a Pipeline that contains the sample Pipeline Component, and then add the completed Pipeline to a Webflow:

- About the Order Representation Example
  - Use the Webflow and Pipeline Editor For Editing Pipeline Components and Webflows
  - Editor Tools Needed
- Creating a New Pipeline and Adding it to a Webflow
  - Step 1: Create a New Pipeline Component
  - Step 2: Adding the New Pipeline Component to the Webflow
  - Step 3: Synchronize the Webflow to the Application

About the Order Representation Example

The sample Pipeline Component `ConvertOrderRepPC` is already built. To use it, you first create a new Pipeline that contains `ConvertOrderRepPC` and then put the new Pipeline in a Webflow. After you complete this process and when you complete an order using the commerce sample application, the order representation is placed on WebLogic Integration's `eventQueue`.

The following sections contain the information you need to complete this process.

Use the Webflow and Pipeline Editor For Editing Pipeline Components and Webflows

To create a new Pipeline and add the Pipeline to your Order Webflow, you should use the Webflow and Pipeline Editors in the E-Business Control Center. These editors are the preferred editing method for the Webflow and Pipeline XML configuration files.

Caution: If you create invalid entries outside of the Editors by hand-editing the XML configuration files in a text editor, the behavior of the Editor may be unpredictable.

Both the E-Business Control Center online help and “Customizing and Extending Webflow” in the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline* contain general information on using the E-Business Control Center and adding Pipeline Components.

Editor Tools Needed

To use the Webflow/Pipeline Editor, you should to be familiar with its tools and component representations. Table 2-1 provides information about the tools and component representations in used for creating the sample Pipeline and adding it to a Webflow. For more information about using the Webflow/Pipeline Editor, see “Using the Webflow and Pipeline Editors” in the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*.

Table 2-1 Pipeline Editor Tools and Elements

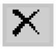
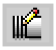
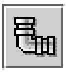
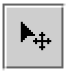
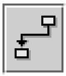
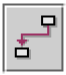
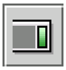
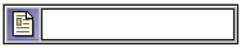
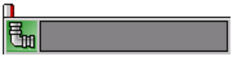
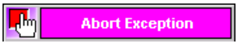
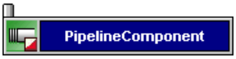
Representation	Webflow Component	Description
	Delete Button	Deletes the selected Webflow component.
	Pipeline Component Editor Button	Opens the Pipeline Component Editor for specifying the parameters of a Pipeline Component.
	Pipeline Node Tool	Adds a new Pipeline Node to the Editor canvas.
	Selection Tool	Selects and moves Pipeline Components, event transitions, and exception transitions.
	Event Tool	Adds an event transition between two nodes.
	Exception Tool	Adds an exception transition between two nodes, or a self-referring exception transition.
	Root Node Tool	Designate one of the Pipeline Components already on the Editor canvas as the Root Node for the current Pipeline.
	Presentation Node	Represents a page displayed to visitors interacting with your Web application.
	Pipeline Node	A processor Webflow component that encapsulates business logic for a Web application.
	Abort Exception Node	Represents the place where a Pipeline's exceptions connect to, because their destinations actually exist outside the Pipeline Editor (in the Webflow Editor).
	Pipeline Component Node	A specialized Webflow component that performs tasks related to the application's underlying business logic. Pipeline Components exist within Pipelines, and may be implemented as Java objects or stateless session EJBs.

Table 2-1 Pipeline Editor Tools and Elements (Continued)

Representation	Webflow Component	Description
	Connection Port	A small graphical device on a node edge that represents where an event or exception is connected to that node.

Creating a New Pipeline and Adding it to a Webflow

This section contains step-by-step instructions for creating the `sendOrderToWLI` Pipeline and adding it to the `sampleapp_order` Webflow using the E-Business Control Center. The following list shows the major steps needed to complete this task:

Step 1: Create a New Pipeline Component

Step 2: Adding the New Pipeline Component to the Webflow

Step 3: Synchronize the Webflow to the Application

Step 1: Create a New Pipeline Component

This section contains the steps needed to create the Pipeline Component used by the Webflow.

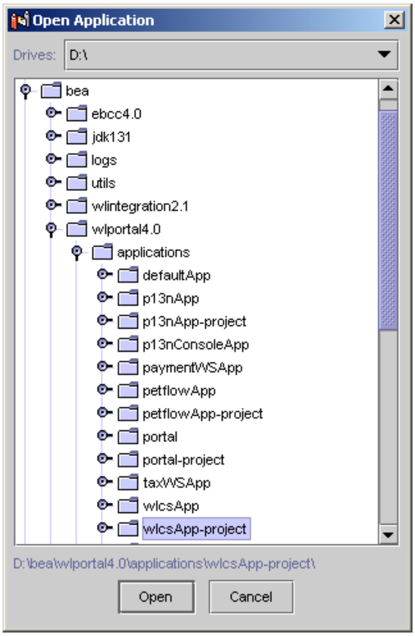
Note: Before you can use the E-Business Control Center, you must start the E-Business Control Center. For more information on starting the E-Business Control Center, see "Starting the E-Business Control Center" in the *Guide to Using the E-Business Control Center*.

1. Open the wlcsApp application, as follows:

   a. In the main toolbar of the E-Business Control Center, click the Open folder icon.

b. In the Open Application dialog box, navigate to the wlcsApp folder (`BEA_HOME` → `wlportal4.0` → `applications` → `wlcsAPP-Project`) as shown in Figure 2-1.

Figure 2-1 Open Application Window



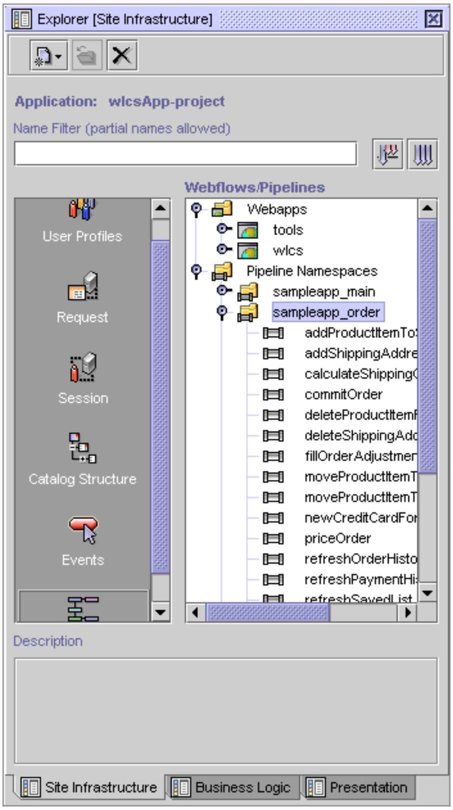
Drives: D:\

D:\bea\wlportal4.0\applications\wlcsApp-project\

c. Select the wlcsApp-project folder, and then click the Open button.

2. In the Explorer window, click the Site Infrastructure tab, as shown in Figure 2-2

Figure 2-2 Explorer Window: Webflows/Pipelines



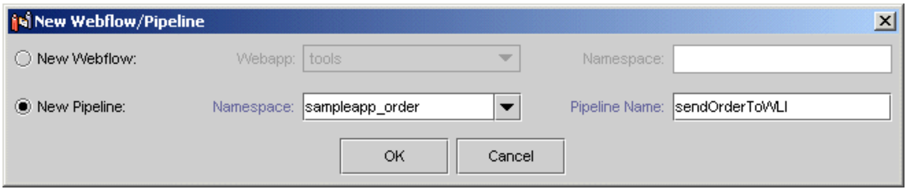
3. In the left pane, click Webflow/Pipelines. A list of Webflows and Pipelines appears.

4. In the Webflows/Pipelines list, open the Pipeline Namespaces folder.

5. Open the `sampleapp_order` namespace.

6. Click the New button, and then select Webflow/Pipeline. The New Webflow/Pipeline window opens, as shown in Figure 2-3.

Figure 2-3 New Webflow/Pipeline Window



New Webflow/Pipeline

○ New Webflow: Webapp: tools Namespace:

● New Pipeline: Namespace: sampleapp_order Pipeline Name: sendOrderToWLI

7. Create the new Pipeline as follows:

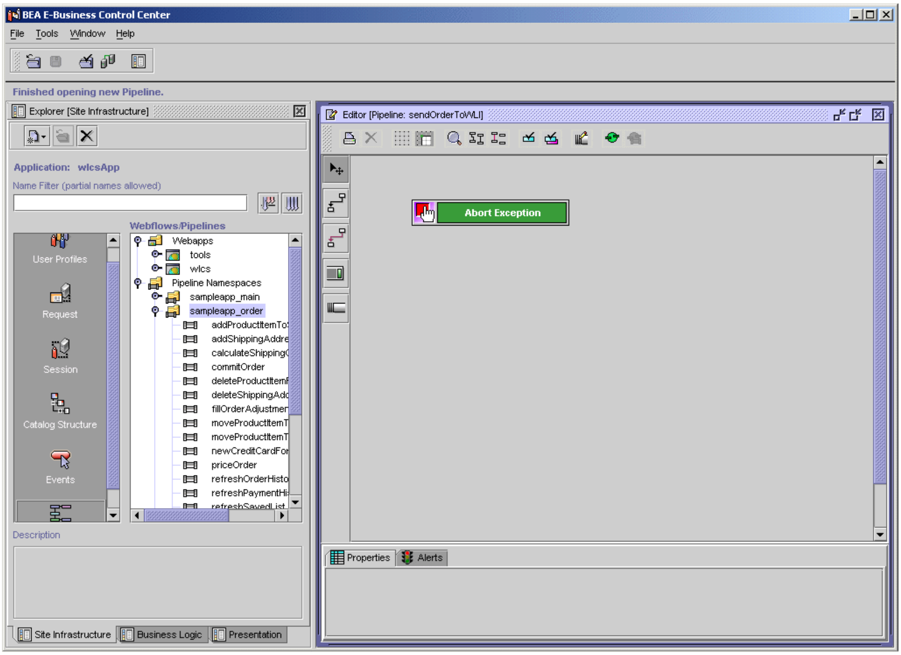
   a. Select the New Pipeline radio button.

   b. In the Namespace list box, select `sampleapp_order`.

   c. In the Pipeline Name field, enter “`sendOrderToWLI`”.

   d. Click the OK button. The Editor appears, as shown in Figure 2-4.

Figure 2-4 Editor [Pipeline: sendOrderToWLI]



BEA E-Business Control Center

Finished opening new Pipeline.

Explorer [Site Infrastructure]

Application: wlcsApp

Editor [Pipeline: sendOrderToWLI]

8. Create the `ConvertOrderRepPC` Pipeline Component as follows:

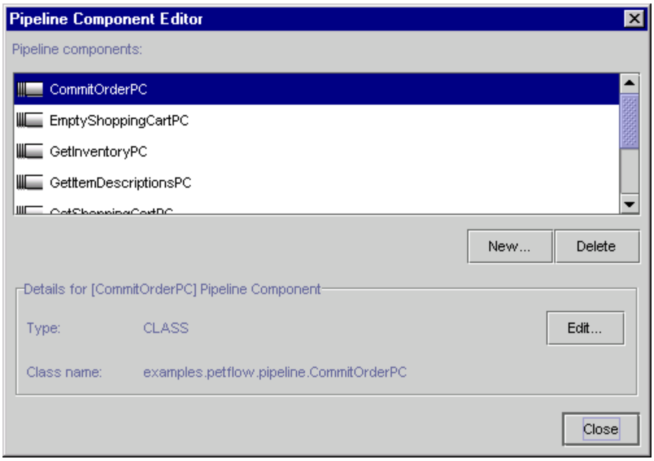
a. Move the Abort Exception Node down near the bottom of the canvas.

a. Click the Pipeline Component Node tool. Cross-hairs appear in the Webflow Editor Canvas.

b. Place the cross-hairs to the place above the Abort Exception Node where you want to place the new Pipeline Component, and then click. The new Pipeline Component Node is inserted on the Editor canvas in that location.

c. Click the Pipeline Component Editor button. The Pipeline Component Editor window opens, as shown in Figure 2-5.

Figure 2-5 Pipeline Component Editor



Pipeline Component Editor

Pipeline components:

- CommitOrderPC
- EmptyShoppingCartPC
- GetInventoryPC
- GetItemDescriptionsPC
- GetShoppingCartPC

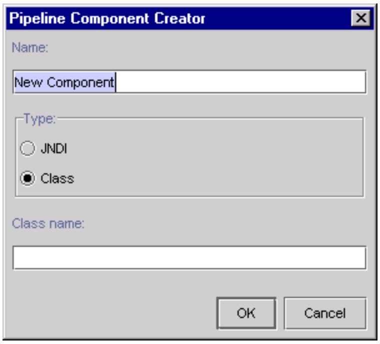
Details for [CommitOrderPC] Pipeline Component

Type: CLASS

Class name: examples.petflow.pipeline.CommitOrderPC

d. In the Pipeline Component Editor window, click the New button. The Pipeline Component Creator window opens, as shown in Figure 2-6.

Figure 2-6 Pipeline Component Creator



Pipeline Component Creator

Name: New Component

Type:
○ JNDI
● Class

Class name:

e. In the Name field, enter "`ConvertOrderRepPC`".

f. Select Class as the Type.

g. Enter the following in the Class name field:

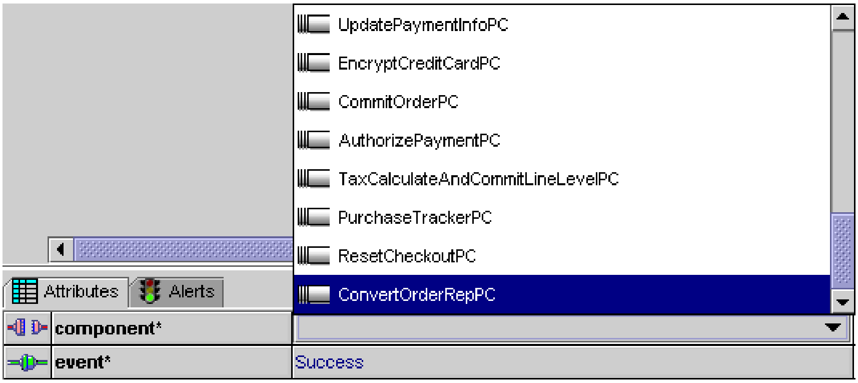
```
examples.wlcs.sampleapp.order.pipeline.ConvertOrderRepPC
```

h. Click the OK button. The Pipeline Component Creator window closes.

i. In the Pipeline Component Editor window, click the Close button.

j. In the Properties Editor, select `ConvertOrderRepPC` from the pop-up list, as shown in Figure 2-7. The Pipeline Component Node changes from Pipeline Component to `ConvertOrderRepPC`.

Figure 2-7 Properties Editor—Component Selection



UpdatePaymentInfoPC
EncryptCreditCardPC
CommitOrderPC
AuthorizePaymentPC
TaxCalculateAndCommitLineLevelPC
PurchaseTrackerPC
ResetCheckoutPC
ConvertOrderRepPC

component*	
event*	Success

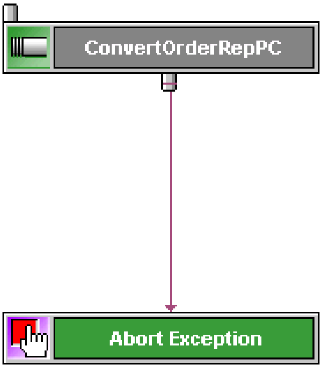
9. Connect the `ConvertOrderRepPC` node to the Abort Exception node, as follows:

a. Click the Exception tool.

b. Position the transition by moving the mouse to bottom edge of the `CovertOrderRepPC` node. A solid orange square indicates an acceptable connection location, and the cursor changes to indicate a transition addition.

c. Hold, and drag the mouse to the Abort Exception node. Release the mouse to connect the transition to the Abort Exception node (Figure 2-8).

Figure 2-8 ConvertOrderRepPC – Abort Exception Connection

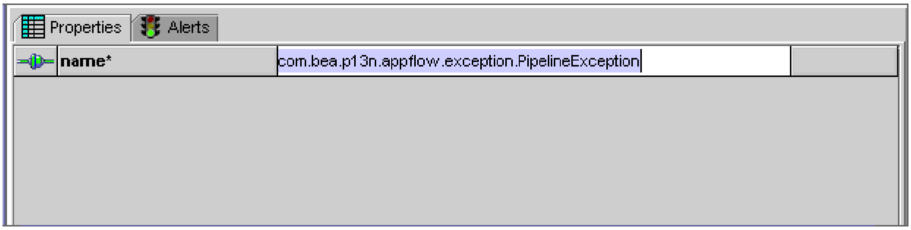


d. Select the Exception event. The event line turns orange.

e. In the Properties Editor, select the second column (Figure 2-9), and enter the package name to the exception class, as follows:

```
com.bea.p13n.appflow.exception.PipelineException
```

Figure 2-9 Properties Editor—Exception Value



name*	com.bea.p13n.appflow.exception.PipelineException

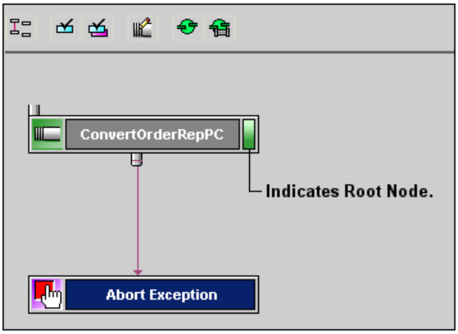
10. Make the `ConvertOrderRepPC` node a Root Node, as follows:

a. Click the Root Node Tool.

b. Move the arrow over the `ConvertOrderRepPC` node. It changes to a green rectangle.

c. Click the green rectangle on the `ConvertOrderRepPC` node. The node is marked with a green stripe to the right of the name, as shown in Figure 2-10.

Figure 2-10 ConvertOrderRepPC—Completed



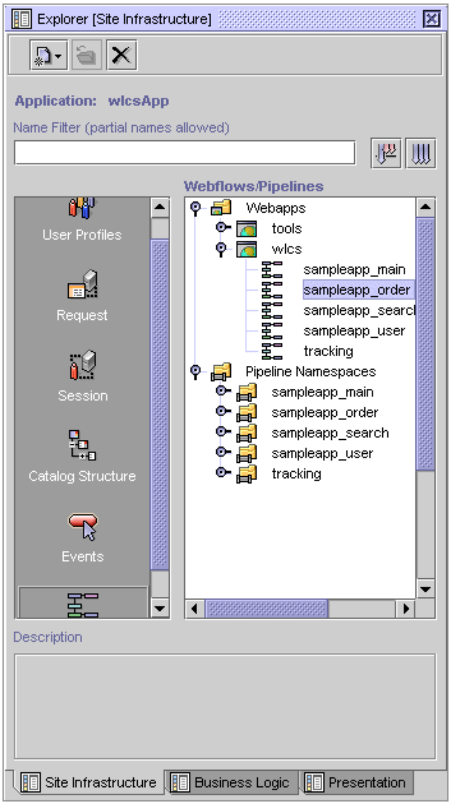
11. To save the new Pipeline, select File → Save from the E-Business Control Center Main toolbar.

Step 2: Adding the New Pipeline Component to the Webflow

This section contains step-by-step instructions for using the Webflow Editor in the E-Business Control Center to add the `sendOrderToWLI` Pipeline to the `sampleapp_order` Webflow.

1. In the right pane of the Explorer, open the wlcs Webflow, and then double-click `sampleapp_order`, as shown in Figure 2-11.

Figure 2-11 Explorer Window: Webflows/Pipelines



Explorer [Site Infrastructure]

Application: wlcsApp

Name Filter (partial names allowed)

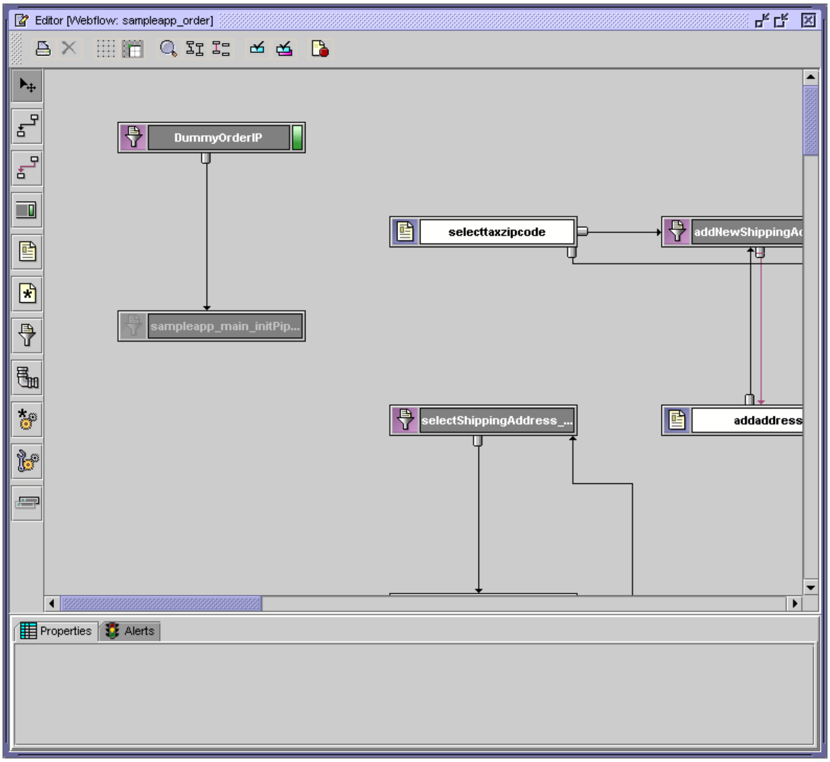
Webflows/Pipelines

- Webapps
  - tools
  - wlcs
    - sampleapp_main
    - sampleapp_order
    - sampleapp_searcl
    - sampleapp_user
    - tracking
- Pipeline Namespaces
  - sampleapp_main
  - sampleapp_order
  - sampleapp_search
  - sampleapp_user
  - tracking

Description

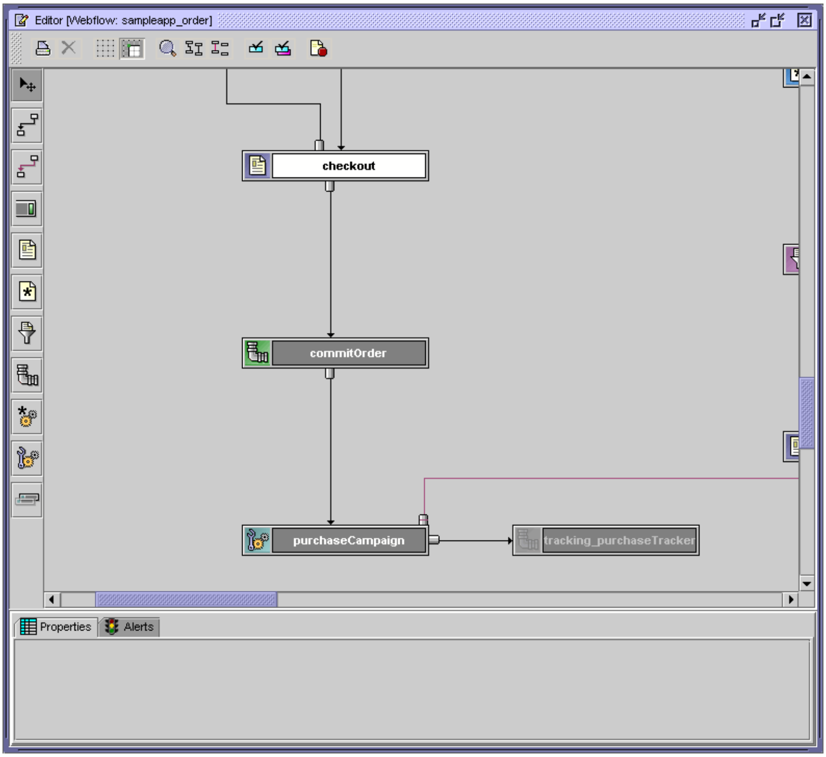
The Webflow editor opens as shown in Figure 2-12.

Figure 2-12 Editor [Webflow: sample_order]—Top


Editor [Webflow: sampleapp_order]
DummyOrderIP
sampleapp_main_initPip...
selecttaxzipcode
addNewShippingA
selectShippingAddress_...
addaddress
Properties
Alerts

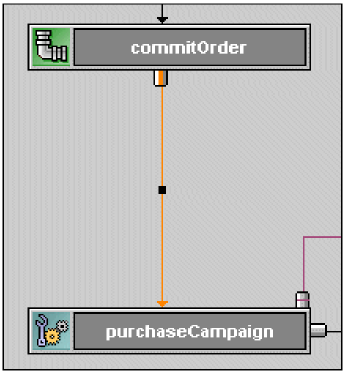
2. Scroll down to near the bottom left of the canvas, as shown in Figure 2-13.

Figure 2-13 Editor [Webflow: sample-order]—Bottom



3. Select the event between the `CommitOrder` and `purchaseCampaign` nodes, as shown in Figure 2-14, and then click the Delete button.

Figure 2-14 Event Selection



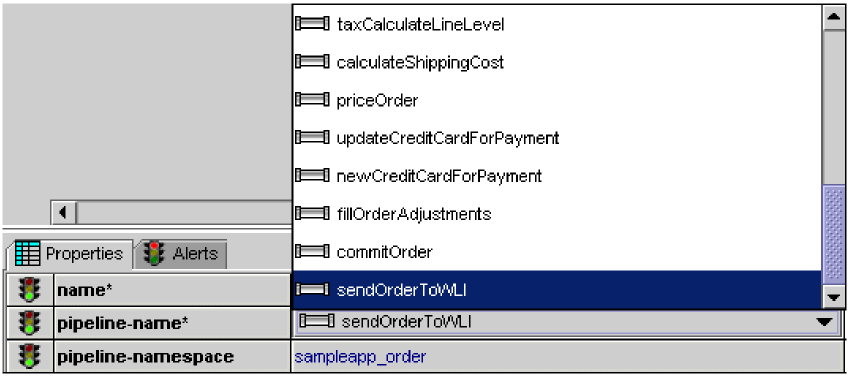
4. Add the `sendOrderToWLI` Pipeline to the Webflow, as follows:

   a. Click the Pipeline Node tool. Cross-hairs appear in the Webflow Editor Canvas.

   b. Place the cross-hairs between `commitOrder` and `tracking_purchaseTracker` nodes, and then click. The new Pipeline Node is inserted on the Editor canvas in that location.

   c. In the Properties Editor, select the second column of the pipeline-name* row, and then select `sendOrderToWLI` from the pop-up list, as shown in Figure 2-15.

Figure 2-15 Properties Editor—Component Selection



name*	
pipeline-name*	sendOrderToWLI
pipeline-namespace	sampleapp_order

5. Connect the `sendOrderToWLI` Pipeline node to the other Pipeline nodes, as follows:

a. With the Event tool connect the commitOrder node to the sendOrderToWLI node, click the Event you just created, and then in the Properties Editor → Properties tab, change the name of the event to "success" and then press Enter.

b. With the Event tool, from the right Connection Port on the top of the sendOrderToWLI node, connect the sendOrderToWLI node to the tracking_purchaseTracker node.

c. Click the Event you just created, then in the Properties Editor → Properties tab, change the name of the event to "success" (if necessary), and then press Enter.

d. Using the Exception tool, from the left Connection Port on the top of the sendOrderToWLI node, connect the sendOrderToWLI node to the sampleapp_main_error Presentation Node, which is located near the middle right of the Webflow.

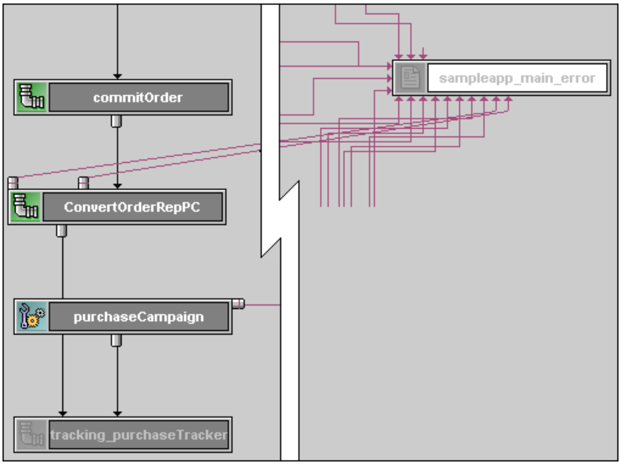
e. Click the Exception you just created, then in the Properties Editor → Properties tab, change the name of the exception to "com.bea.p13n.applfow.exception.PipelineException" if necessary, and then press Enter.

f. Using the Exception tool, make a second connection from the sendOrderToWLI node to the sampleapp_main_error Presentation Node.

g. Click the Exception you just created, then in the Properties Editor → Properties tab, change the name of the exception to "java.rmi.RemoteException" and then press Enter.

Figure 2-16 shows the Webflow Editor canvas after this process is completed. The nodes have been moved to make the drawing easier to see.

Figure 2-16 Event Transitions



Note: You can move the Connection Ports, add elbows, and route the transitions. For more information, see "Using the Webflow and Pipeline Editors" in the *Guide to Managing Presentation and Business Logic: Using Webflow and Pipeline*.

6. To save the changes, select File → Save from the E-Business Control Center Main toolbar.

Step 3: Synchronize the Webflow to the Application

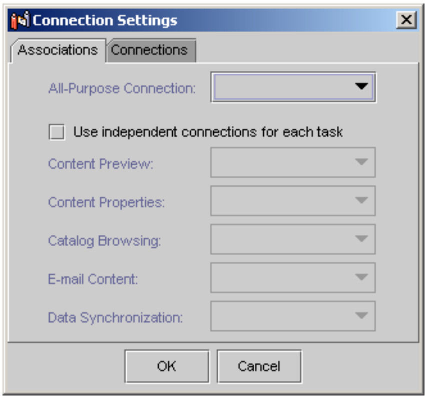
The Webflow and Pipeline Component that you just added must be synchronized to the wlcs application in order for the Example to work.

Warning: All application data is synchronized at once. If you and other developers concurrently synchronize data to a single enterprise application, it is possible to overwrite each others' work or create sets of changes that are incompatible and difficult to debug. To prevent this possibility, synchronize to separate instances of your application. For more information on how to set up your development environment, see "Milestone 4: Set Up a Development Site" in the *Strategies for Developing E-Business Web Sites* documentation.

To synchronize the new Pipeline Component and the modified Webflow, take the following steps:

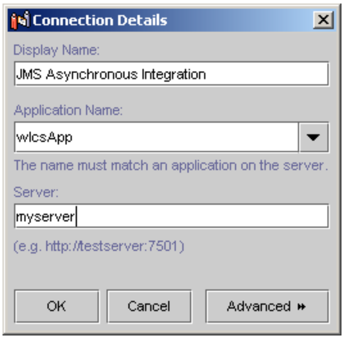
1. Start WebLogic Portal.
2. After WebLogic Portal starts, in the E-Business Control Center, go to Tools → Connection Settings. The connections Settings dialog box appears, as shown in Figure 2-17.

Figure 2-17 Connections Settings



3. In the Connection Settings dialog box, click the Connections tab, and then select New. The Connection Details dialog box appears, as shown in Figure 2-18.

Figure 2-18 Connection Details



Display Name: JMS Asynchronous Integration

Application Name: wlcsApp

The name must match an application on the server.

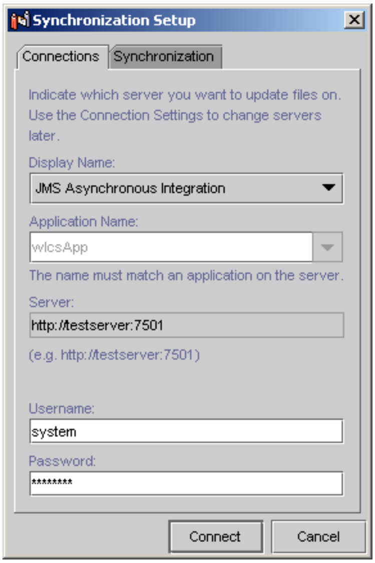
Server: myserver

(e.g. http://testserver:7501)
OK
Cancel
Advanced »

4. In the Display Name field enter any name you want, select wlcsApp from the Application Name drop list, enter the name of your server, and then click the OK button. The Connection Details dialog box closes.
5. In the Connection Setting dialog box, click the OK button. The dialog box closes.

6. In the E-Business Control Center, click the Synchronize button. The Synchronize Setup dialog box appears, as shown in Figure 2-19.

Figure 2-19 Synchronization Setup



Indicate which server you want to update files on. Use the Connection Settings to change servers later.

Display Name: JMS Asynchronous Integration

Application Name: wlcsApp

The name must match an application on the server.

Server: http://testserver:7501

(e.g. http://testserver:7501)

Username: system

Password: ********

7. From the Display Name drop list, select the name you created in the Connection Details dialog box, enter your Username and Password for WebLogic Portal, and then click Connect. The Checking for Incomplete Files dialog box appears.

8. In the Checking for Incomplete Files dialog box, click the Continue button, even if incomplete documents exist. The Synchronizing dialog box appears.

9. After synchronizing finishes, click the Close button.

This completes the synchronization procedure. How to set up the WebLogic Integration part of the Example and verifying successful execution of the purchase order is described in Chapter 3, "WebLogic Integration Interoperability Sample."

3 WebLogic Integration Interoperability Sample

Before setting up WebLogic Integration, you should have already configured the WebLogic Portal sample as described in Chapter 2, “WebLogic Portal Interoperability Sample.” This document provides the following information:

Description of the WebLogic Integration Part of the Example

The WebLogic Integration part of the WebLogic Portal and WebLogic Integration Example is a JMS asynchronous application designed to show how the business process management (BPM) component of WebLogic Integration can be used with WebLogic Portal. For more information see *Learning to Use BPM with WebLogic Integration*.

Prerequisites For Running the Example

- You must set up a WebLogic Integration environment in the following directory: `WLI_HOME\config\samples`. For details, see “Domain Configuration Requirements” in *Starting, Stopping, and Customizing WebLogic Integration*.
- The InstallAnywhere installer for WebLogic Integration 2.1 is not bundled with a JRE. You must put the `<jdk-install-directory>/bin` in your system `PATH` environment variable. On Windows, you can update the `PATH` in the System Environment Variables control panel. On UNIX, you can define:

  ```
  %PATH=<BEA_HOME>/JDK131/BIN:$path
  ```

  ```
  %export PATH
  ```

  where `<BEA_HOME>` is the location of the BEA home directory.
- You must run the Example on the same machine on which WebLogic Portal is running.
- The port for the Example should be 7001.

Configuring and Verifying the WebLogic Integration Sample

This section contains step-by-step instructions for configuring and verifying the WebLogic Integration part of the Example. The following list contains the major steps needed to complete this task:

- Step 1: Install the WebLogic Integration Example Files
- Step 2: Create the Sample Tables
- Step 3: Import the WebLogic Integration Example Components
- Step 4: Deploy the orderAppView Sample Application
- Step 5: Make a Purchase in WebLogic Portal
- Step 6: How to Verify Successful Execution

Step 1: Install the WebLogic Integration Example Files

Before you can run the two-part Example, you must set up your environment for WebLogic Integration. To install the files for WebLogic Integration sample, complete the following steps:

1. Create a directory called `JAI` in `WLI_HOME\config\samples`, where `WLI_HOME` is directory in which you installed WebLogic Integration.
2. Place the `jai_sample1.jar` into the new directory.
3. Open a command/shell window and go to the newly created directory.
4. Extract all the files from `jai_sample1.jar` into your current folder by entering the following command: `jar -xvf jai_sample1.jar`. This creates the following three directories:
   - `META-INF`
   - `schema`
   - `workflows`

Step 2: Create the Sample Tables

1. Navigate to the `WLI_HOME\config\samples\JAI\schema` directory.
2. To create the necessary sample tables, execute `create_jai_cloudscape.bat/sh` or `jai_<db>.sql` file, where `db` is `oracle` or `mssql`. Make sure of the following:
   - You have switched WebLogic Integration to the database you are using.
   - The WebLogic Integration repository tables are set up.
   - The Database Management System (DBMS) is set up.
   - The same database is used for both the WebLogic Integration repository and the Interoperability Sample.
   - The Interoperability Sample and WebLogic Portal are installed on the same machine.

Step 3: Import the WebLogic Integration Example Components

1. Start the WebLogic Integration server for the Interoperability Sample from the command window:

   ```
   WLI_HOME\config\samples\startweblogic.cmd/sh
   ```

2. After WebLogic Integration is running, start the WebLogic Integration Studio, as follows:

   Windows: Start Menu → Programs → BEA WebLogic E-Business Platform → WebLogic Integration 2.1 → Studio

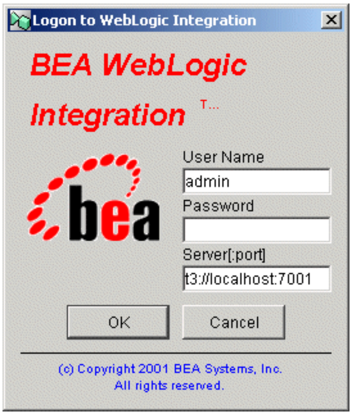
   UNIX: Go to the `WLI_HOME/bin` directory and execute the `studio.sh` script by entering the following command:

   ```
   sh studio.sh
   ```

   WebLogic Integration Studio starts and the Logon dialog box appears, as shown in Figure 3-1.

Note: For more information about starting WebLogic Integration, see “Starting and Logging On to the Studio” in *Using the WebLogic Integration Studio*.

Figure 3-1 WebLogic Integration Logon Dialog Box



User Name: admin

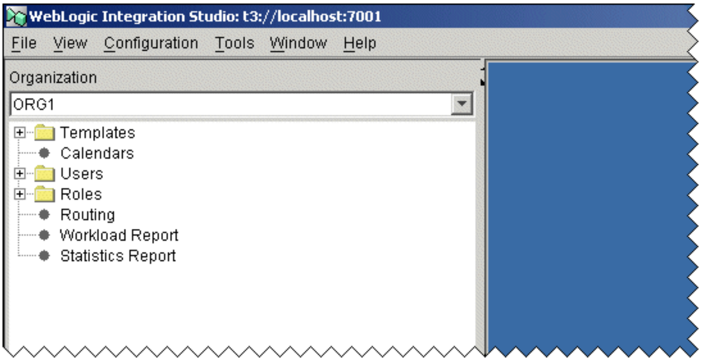
Password:

Server[:port]: t3://localhost:7001

(c) Copyright 2001 BEA Systems, Inc.
All rights reserved.

3. Log on to the WebLogic Integration Studio using “`security`” as the password. The Studio opens as shown in Figure 3-2.

Figure 3-2 WebLogic Integration Studio



Organization: ORG1

4. Import the `OrderWorkflow.jar` file as follows:

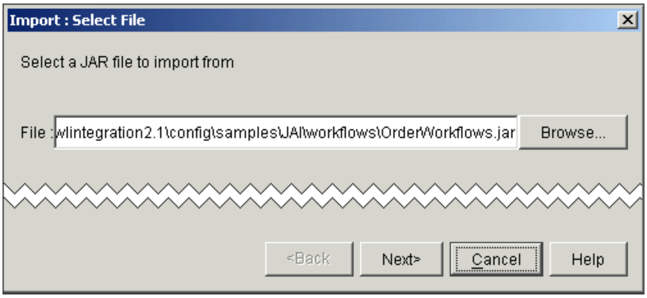
   a. Tools → Import Package. The Import: Select File dialog box is displayed.

   b. Click the Browse button and import the following file:

      `WLI_HOME\config\samples\jai\workflows\OrderWorkflow.jar`

c. After the imported file appears in the File field (Figure 3-3), click the Next button.

Figure 3-3 Import Select File Window



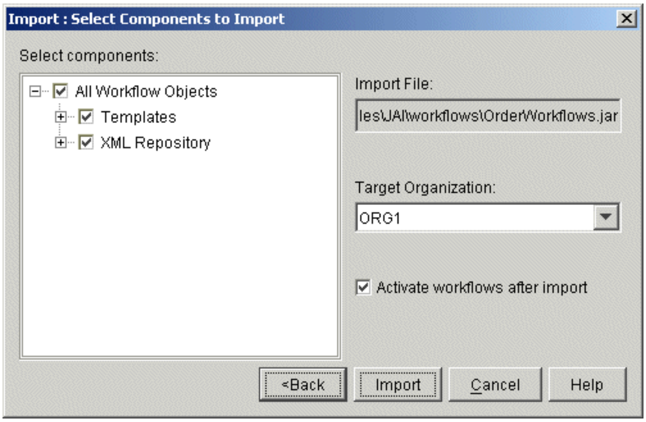
Import : Select File

Select a JAR file to import from

File : wlintegration2.1\config\samples\JAI\workflows\OrderWorkflows.jar

d. In the Target Organization drop list, choose `ORG1`, select the Activate workflows after import check box, and then click the Import button (Figure 3-4).

Figure 3-4 Select Components to Import Window



Import : Select Components to Import

Select components:

- ☑ All Workflow Objects
  - ☑ Templates
  - ☑ XML Repository

Import File: les\JAI\workflows\OrderWorkflows.jar

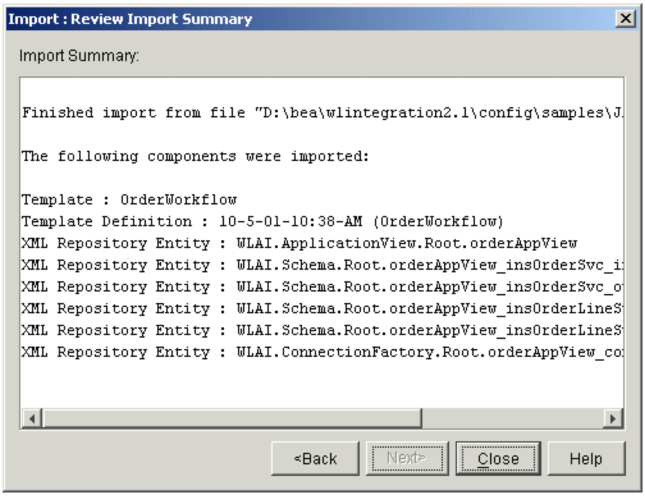
Target Organization: ORG1

☑ Activate workflows after import

Note: If the workflows are not activated, the sample will not run.

The components are imported as shown in Figure 3-5.

Figure 3-5 Review Import Summary



```
Finished import from file "D:\bea\wlintegration2.1\config\samples\J

The following components were imported:

Template : OrderWorkflow
Template Definition : 10-5-01-10:38-AM (OrderWorkflow)
XML Repository Entity : WLAI.ApplicationView.Root.orderAppView
XML Repository Entity : WLAI.Schema.Root.orderAppView_insOrderSvc_i
XML Repository Entity : WLAI.Schema.Root.orderAppView_insOrderSvc_o
XML Repository Entity : WLAI.Schema.Root.orderAppView_insOrderLineS
XML Repository Entity : WLAI.Schema.Root.orderAppView_insOrderLineS
XML Repository Entity : WLAI.ConnectionFactory.Root.orderAppView_co
```

5. Click the Close button.

Step 4: Deploy the orderAppView Sample Application

1. Open the Application View Console as follows:

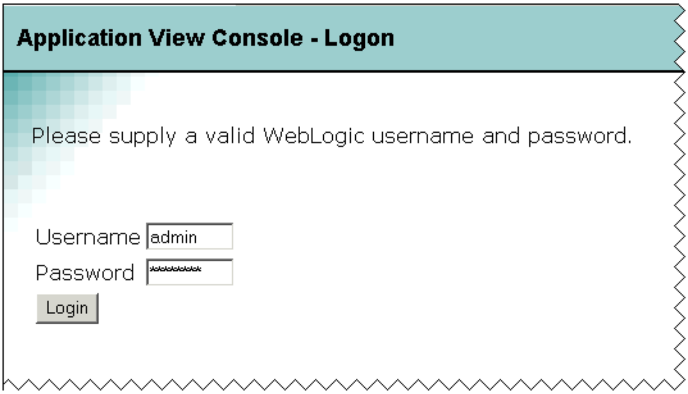
   Windows: Start Menu → Programs → BEA WebLogic E-Business Platform → WebLogic Integration 2.1 → Application View Console.

   UNIX: Open a Web browser and to go the following URL:

   ```
   http://WLP-WLI_machine:7501/wlcs
   ```

   The Application View Console - Logon opens as shown in Figure 3-6.

Figure 3-6 Application View Console - Logon



Application View Console - Logon

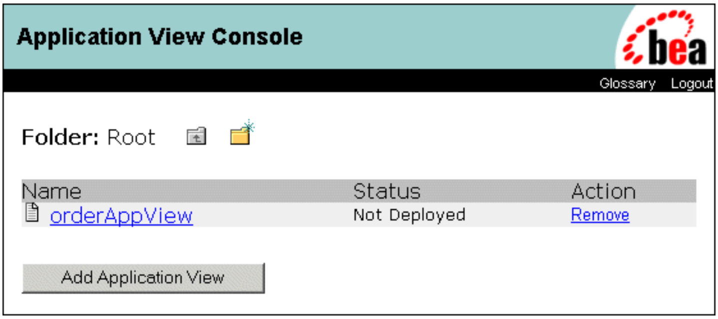
Please supply a valid WebLogic username and password.

Username admin

Password ********

2. Enter "`admin`" and "`security`" as the Username and Password, and then click the Login button. The window changes to show the console, as shown in Figure 3-7.

Figure 3-7 Application View Console



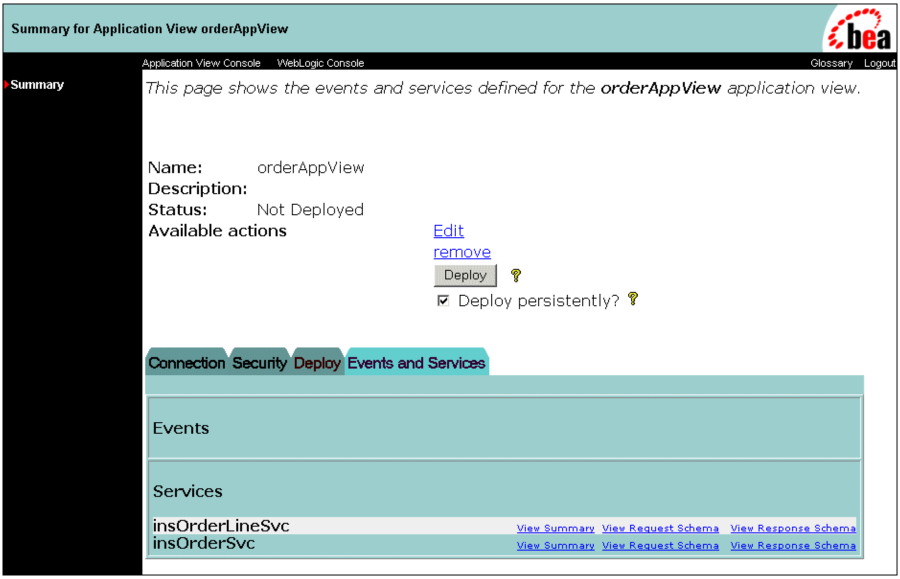
Application View Console

Folder: Root

Name	Status	Action
orderAppView	Not Deployed	Remove

3. Select the Application View named orderAppView. A summary of the selected application view is displayed, as shown in Figure 3-8.

Figure 3-8 Summary for Application View orderAppView



This page shows the events and services defined for the ***orderAppView*** *application view.*

Name: orderAppView
Description:
Status: Not Deployed
Available actions

☑ Deploy persistently?

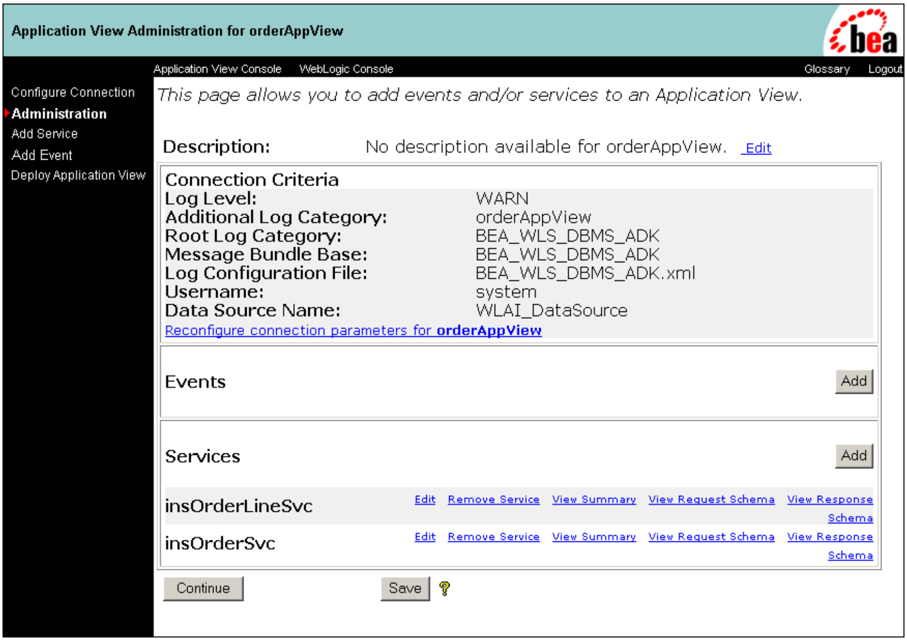
Events

Services

insOrderLineSvc
insOrderSvc

4. On the Summary page, click the Edit link. An Administration page is displayed as shown in Figure 3-9.

Figure 3-9 Application View Administration for orderAppView



This page allows you to add events and/or services to an Application View.

Description: No description available for orderAppView.

Connection Criteria

Log Level:	WARN
Additional Log Category:	orderAppView
Root Log Category:	BEA_WLS_DBMS_ADK
Message Bundle Base:	BEA_WLS_DBMS_ADK
Log Configuration File:	BEA_WLS_DBMS_ADK.xml
Username:	system
Data Source Name:	WLAI_DataSource

Reconfigure connection parameters for **orderAppView**

Events

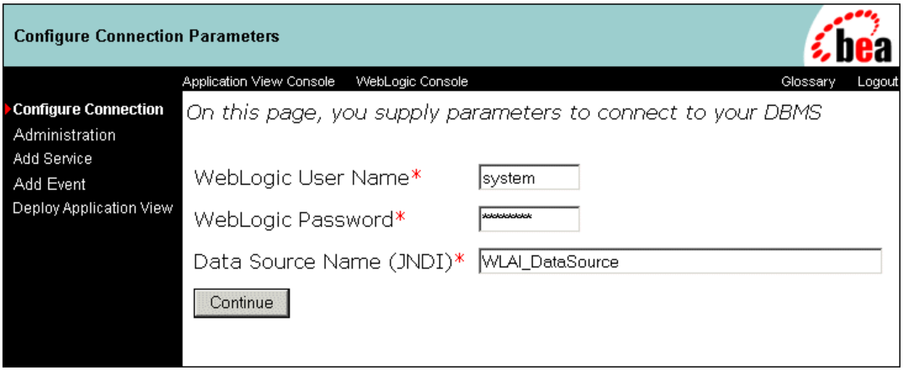
Services

insOrderLineSvc

insOrderSvc

5. Click Reconfigure connection parameters for orderAppView. The Configuration Connection Parameters page is displayed, as shown in Figure 3-10.

Figure 3-10 Configure Connection Parameters



On this page, you supply parameters to connect to your DBMS

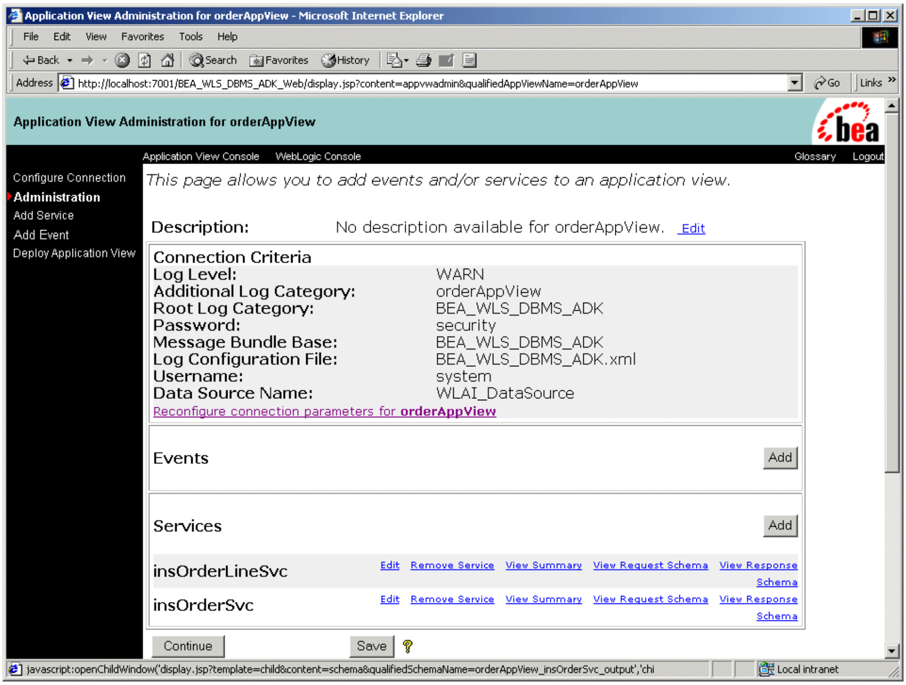
WebLogic User Name* system

WebLogic Password* ********

Data Source Name (JNDI)* WLAI_DataSource

6. Enter “`system`” and “`security`” in the WebLogic User Name and Password fields, and then click the Continue button. The Administration page is displayed, as shown in Figure 3-11.

Figure 3-11 Application View Administration for orderAppView—Parameters Set



Address: http://localhost:7001/BEA_WLS_DBMS_ADK_Web/display.jsp?content=appvwadmin&qualifiedAppViewName=orderAppView

Application View Administration for orderAppView

This page allows you to add events and/or services to an application view.

Description: No description available for orderAppView.

Connection Criteria

Log Level:	WARN
Additional Log Category:	orderAppView
Root Log Category:	BEA_WLS_DBMS_ADK
Password:	security
Message Bundle Base:	BEA_WLS_DBMS_ADK
Log Configuration File:	BEA_WLS_DBMS_ADK.xml
Username:	system
Data Source Name:	WLAI_DataSource

Reconfigure connection parameters for **orderAppView**

Events

Services

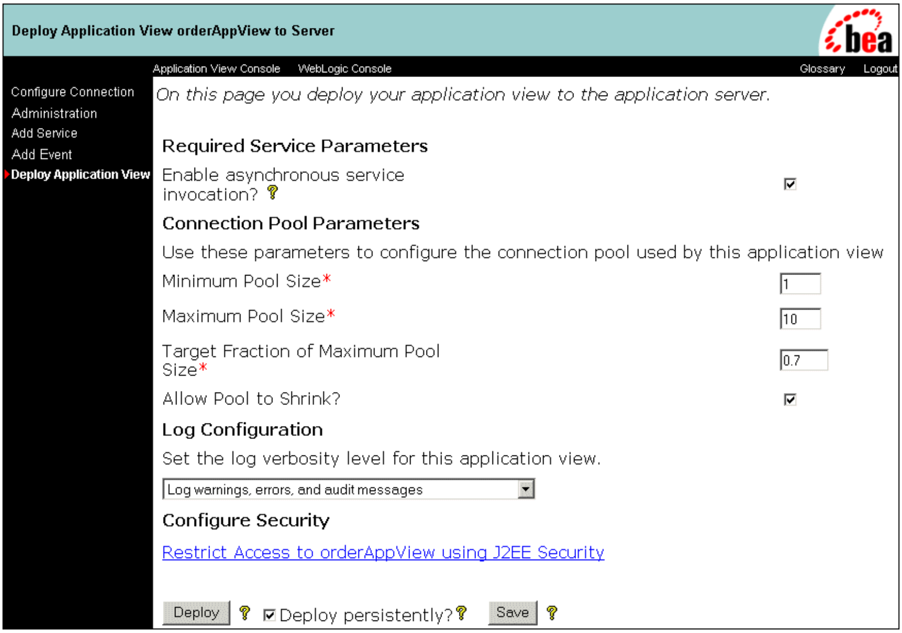
insOrderLineSvc

insOrderSvc

javascript:openChildWindow('display.jsp?template=child&content=schema&qualifiedSchemaName=orderAppView_insOrderSvc_output','chi

7. To deploy the specified application, click the Continue button. The Deploy Application View window is displayed, as shown in Figure 3-12.

Figure 3-12 Deploy Application View orderAppView to Server



Deploy Application View orderAppView to Server

On this page you deploy your application view to the application server.

Required Service Parameters

Enable asynchronous service invocation? ☑

Connection Pool Parameters

Use these parameters to configure the connection pool used by this application view

Minimum Pool Size* 1

Maximum Pool Size* 10

Target Fraction of Maximum Pool Size* 0.7

Allow Pool to Shrink? ☑

Log Configuration

Set the log verbosity level for this application view.

Log warnings, errors, and audit messages

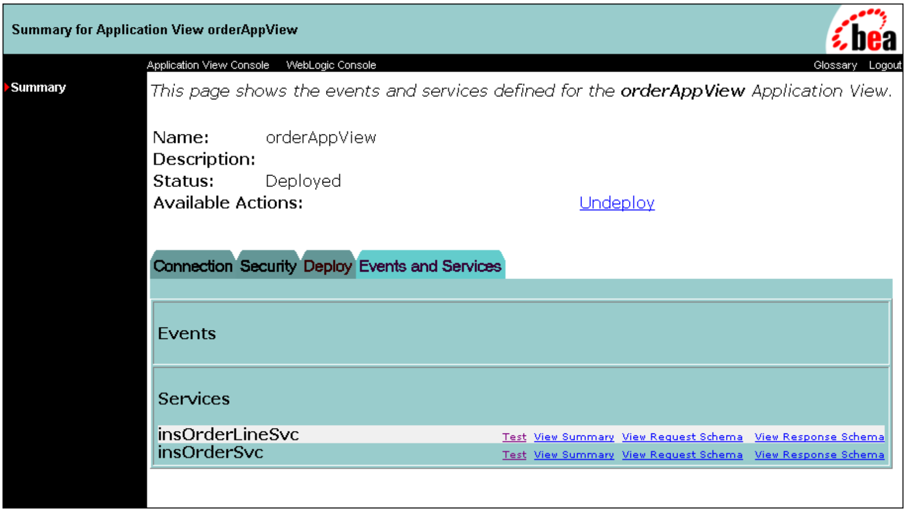
Configure Security

Restrict Access to orderAppView using J2EE Security

Deploy ☑ Deploy persistently? Save

8. Click the Deploy button. The application is deployed as shown in Figure 3-13.

Figure 3-13 Summary for Application View orderAppView



Summary for Application View orderAppView

This page shows the events and services defined for the **orderAppView** Application View.

Name: orderAppView
Description:
Status: Deployed
Available Actions: Undeploy

Events

Services

insOrderLineSvc Test View Summary View Request Schema View Response Schema
insOrderSvc Test View Summary View Request Schema View Response Schema

This completes setting up the WebLogic Integration part of the Example. Before continuing you must have configured and synchronized the WebLogic Portal part of the Example as described in Chapter 2, “WebLogic Portal Interoperability Sample.”

Step 5: Make a Purchase in WebLogic Portal

To place an order on the WebLogic Integration JMS `eventQueue`, you must first make a purchase in WebLogic Portal. Purchase of one or more items creates an order that is placed in the queue.

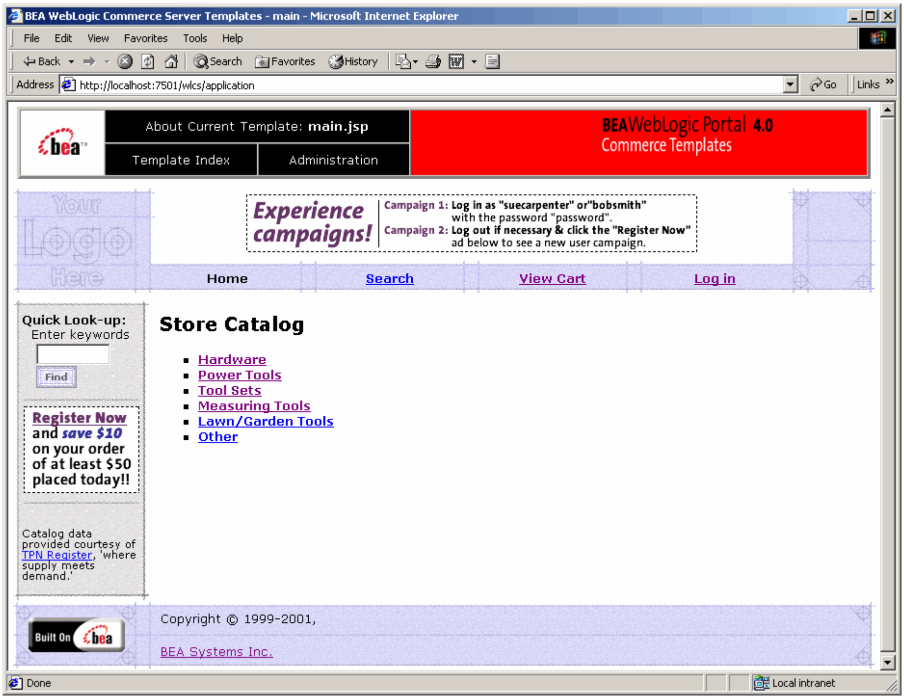
1. Make sure that the servers for both WebLogic Portal and WebLogic Integration are started.

2. Launch the WebLogic Portal Commerce Templates using the following URL:

   ```
   http://WLP-WLI_machine:7501/wlcs
   ```

   Figure 3-14 shows the starting page for the Commerce Template, which simulates a shopping Web site.

Figure 3-14 Commerce Templates Home Page



Address: http://localhost:7501/wlcs/application

About Current Template: **main.jsp**

Experience campaigns!

Campaign 1: Log in as "suecarpenter" or"bobsmith" with the password "password".

Campaign 2: Log out if necessary & click the "Register Now" ad below to see a new user campaign.

Quick Look-up:
Enter keywords

Register Now and ***save $10*** on your order of at least $50 placed today!!

Catalog data provided courtesy of TPN Register, 'where supply meets demand.'

Store Catalog

- Hardware
- Power Tools
- Tool Sets
- Measuring Tools
- Lawn/Garden Tools
- Other

Copyright © 1999-2001,

BEA Systems Inc.

3. Following the instructions displayed on the screen, purchase one or more items from the site. For more information, see the tour of the *JSP Commerce and Campaign Tour*.

   After the order is completed, WebLogic Portal converts the order into an XML representation and then places the order representation on a Java Message Service (JMS) queue in WebLogic Integration.

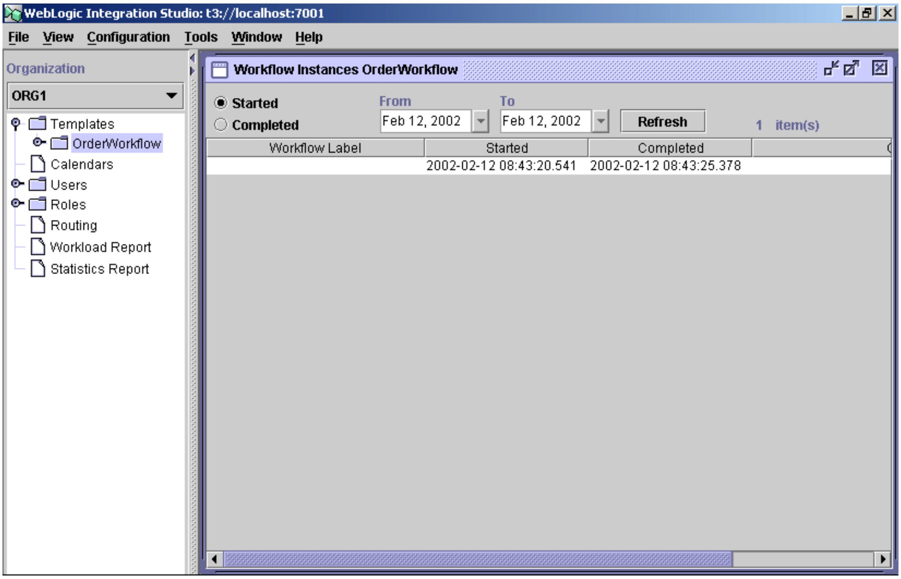
Step 6: How to Verify Successful Execution

To verify that the Example has executed successfully, completing the following steps:

1. Go to WebLogic Integration Studio.
2. In the left pane, right click the 10/5/10 12:38PM folder again, and then select Instances.

3. The Workflow Instances OrderWorkflow opens and shows the purchase instance, as shown in Figure 3-15.

Figure 3-15 Workflow Instances OrderWorkflow Window



● Started
○ Completed

From: Feb 12, 2002
To: Feb 12, 2002

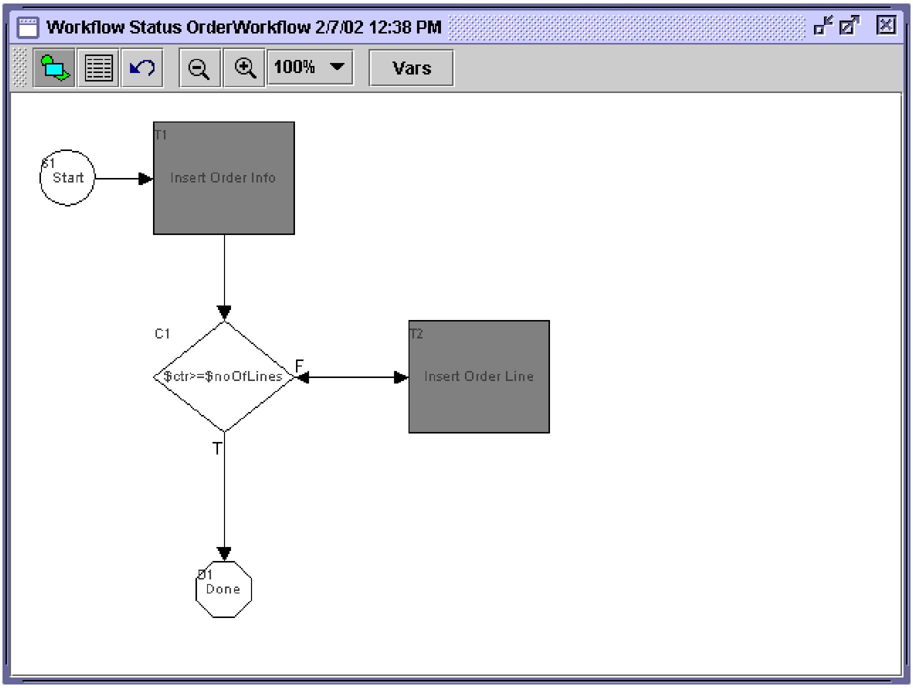
1 item(s)

Workflow Label	Started	Completed
	2002-02-12 08:43:20.541	2002-02-12 08:43:25.378

4. To verify that the values being used for the variables match the details of your order, take the following steps:

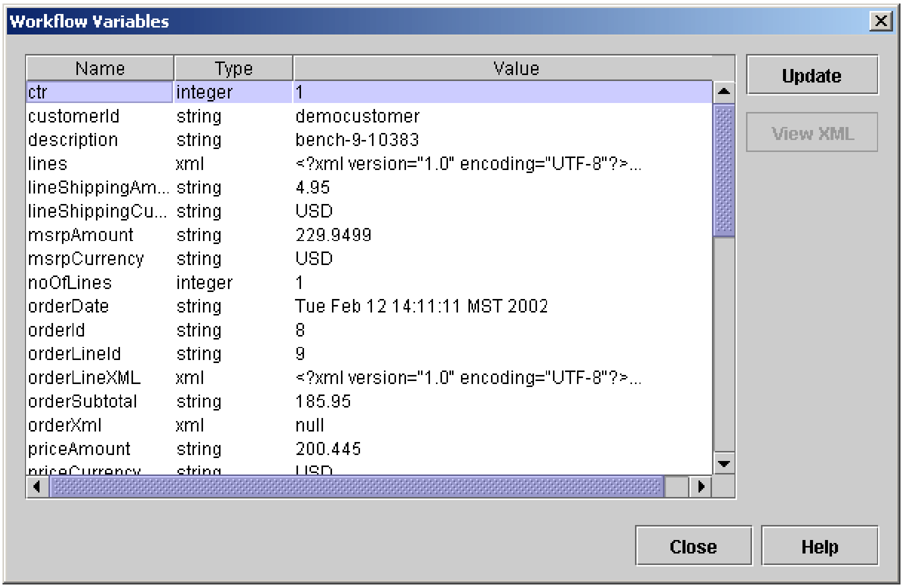
   a. In the Workflow Instances window, double click the order. The Window status screen appears, as shown in Figure 3-16.

Figure 3-16 Workflow Status OrderWorkflow Window



b. Click the Vars (Variables) button. The Workflow Variables window opens and shows the purchase information, as shown in Figure 3-17.

Figure 3-17 Workflow Variables Window



Name	Type	Value
ctr	integer	1
customerId	string	democustomer
description	string	bench-9-10383
lines	xml	<?xml version="1.0" encoding="UTF-8"?>...
lineShippingAm...	string	4.95
lineShippingCu...	string	USD
msrpAmount	string	229.9499
msrpCurrency	string	USD
noOfLines	integer	1
orderDate	string	Tue Feb 12 14:11:11 MST 2002
orderId	string	8
orderLineId	string	9
orderLineXML	xml	<?xml version="1.0" encoding="UTF-8"?>...
orderSubtotal	string	185.95
orderXml	xml	null
priceAmount	string	200.445
priceCurrency	string	USD

This completes verifying the WebLogic Portal and WebLogic Integration Example.

Index