BEA WebLogic Process Integrator

A Component of BEA WebLogic Integration

Using the BEA WebLogic Process Integrator Studio

BEA WebLogic Process integrator Release 2.0
Document Edition 2.0
July 2001

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Using the BEA WebLogic Process Integrator Studio

Document Edition	Date	Software Version
2.0	July 2001	2.0

Contents

About This Document

1. BEA WebLogic Process Integrator Overview

2. Getting Started with the Studio

3. Working with Plug-Ins

4. Administering Data

5. Defining and Maintaining Workflows

6. Working with Workflow Components

7. Defining Actions

8. Using Expressions and Conditions

9. Handling Workflow Exceptions

10. Monitoring Workflows

A. Understanding Event-Driven Processing

B. Example of Starting a Workflow Using an Event

C. Working with XML Entities

Index

About This Document

This document explains how to design, administer, and monitor workflows using the BEA WebLogic Process Integrator™ Studio. This document is organized as follows:

- Chapter 1, "BEA WebLogic Process Integrator Overview," provides an overview of business process and workflow concepts, and BEA WebLogic Process Integrator components.
- Chapter 2, "Getting Started with the Studio," explains how to log in to and exit from the Studio, and describes the Studio user interface.
- Chapter 3, "Working with Plug-Ins,"describes the plug-in framework that developers can use to extend the functionality provided by BEA WebLogic Process Integrator, and how to work with plug-ins.
- Chapter 4, "Administering Data," explains how to add, update, and delete organizations, users, roles, and tasks. It also describes task routing administration, calendar administration, and business operations.
- Chapter 5, "Defining and Maintaining Workflows," explains how to create, define, open, save, close, copy, and delete BEA WebLogic Process Integrator workflows.
- Chapter 6, "Working with Workflow Components," describes workflow components, including variables, starts, dones, decisions, tasks, events, and joins.
- Chapter 7, "Defining Actions," describes task, workflow, integration, and miscellaneous actions.
- Chapter 8, "Using Expressions and Conditions," describes how to create workflow expressions using the Expression Builder and the XPath wizard.

- Chapter 9, “Handling Workflow Exceptions,” describes the workflow exception handling facility, which allows you to generate, trap, and respond to internally and externally generated exception conditions.
- Chapter 10, “Monitoring Workflows,” describes the workflow monitoring facility, including its user interface, workflow statuses, workflow variables, and workflow reports.
- Appendix A, “Understanding Event-Driven Processing,” explains how events are processed by BEA WebLogic Process Integrator, and how to define an event key expression.
- Appendix B, “Example of Starting a Workflow Using an Event,” provides an example of a workflow that is started by the receipt of an XML document.
- Appendix C, “Working with XML Entities,” explains how to use the XML Finder to find XML objects, and how to manage XML objects in the XML repository

What You Need to Know

This document is intended mainly for any user involved in the workflow design process, from business analysts to system administrators. It assumes a familiarity with Java programming, XML documentation structure (XPath language structure), and business processes involved in the design of workflows.

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.beasys.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 BEA WebLogic Process Integrator 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 BEA WebLogic Process Integrator 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 BEA WebLogic Process Integrator documents contain information that is relevant to using this product:

- *Learning to Use BEA WebLogic Process Integrator*
- *Using the BEA WebLogic Process Integrator Worklist*
- *Installing and Configuring BEA WebLogic Process Integrator*
- *Programming BEA WebLogic Process Integrator Client Applications*
- *BEA WebLogic Process Integrator Javadoc*
- *BEA WebLogic Process Integrator Release Notes*

Contact Us!

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

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

If you have any questions about this version of BEA BEA WebLogic Process Integrator, or if you have problems installing and running BEA BEA WebLogic Process Integrator, contact BEA Customer Support through BEA WebSupport at **www.beasys.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
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 filenames 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`

`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
`[ ]`	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.

CHAPTER

1 BEA WebLogic Process Integrator Overview

BEA WebLogic Process Integrator is a workflow management system that you use to automate business processes. A business process is a series of interconnected business activities that produce some desired result. A graphical representation of a business process is a *workflow*. You define and monitor workflows using the BEA WebLogic Process Integrator Studio, a BEA WebLogic Process Integrator client application.

The following sections provide an overview of workflow concepts and BEA WebLogic Process Integrator components:

- Business Processes and Workflows
- Workflow Automation and Management
- Business Organizations and Workflows
- Workflow Design and Integration
- BEA WebLogic Process Integrator Components
- Defining a Workflow

Business Processes and Workflows

In order for a business organization to be effective, numerous discrete business activities must be performed in a sequence that produces the desired result. Business processes provide a way to structure and connect those business activities to achieve the desired result.

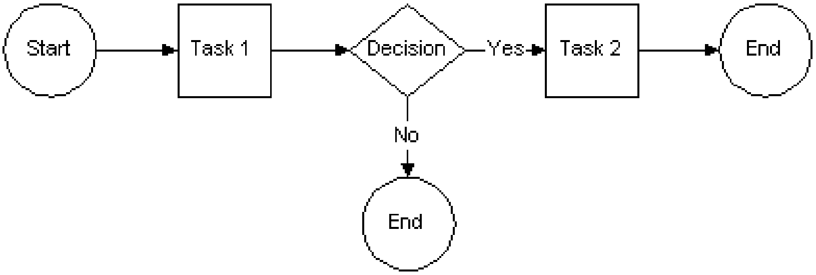
For example, in a Web-based ordering system, business activities must work in sequence so that orders are expedited and products are shipped in a timely manner. A business process provides the necessary structure to those business activities: an order is submitted, product availability is checked, customer credit is verified, the order is confirmed, and so on, until the product is shipped.

A business process, however, can be broken down into some basic elements, and the interaction between those elements can be represented graphically. In a typical business process, the basic elements are as follows:

- Start — trigger the start of a business process, and may include externally initiated events, such as order submissions.
- Tasks — perform some business activity, such as checking product availability.
- Decisions — define a business rule that determines the flow of the business process, such as deciding whether a customer's credit is satisfactory.
- End — terminate the process.

We can represent the basic elements of a business process graphically using a workflow diagram such as the one shown in the following figure.

Figure 1-1 Workflow Diagram



In this workflow diagram, a start event, such as submitting an order, starts the business process. The order flows from task to task until it encounters a decision, which might cause the flow to change direction. Eventually, the flow reaches an end event, completing the process.

Workflow diagrams such as this one provide a graphical syntax that you can use to model business processes, and to describe how numerous discrete business activities should be sequenced to produce the desired result.

Workflow Automation and Management

In an e-business world, businesses must perform effectively at a rapid pace. To achieve this level of performance, many companies automate their business processes. For these companies, software performs most of the work in a business process.

An automated business process does not, however, entirely replace people with computers. People might still be involved in an automated process, either by making discretionary decisions or by handling exceptions or problems as they arise. Perhaps more importantly, people enable automated business processes by defining, implementing, and managing them. This raises human involvement in business processes to a higher level, enabling people to focus on design and optimization, leaving the execution of routine tasks to computers.

To automate their business processes, organizations use a workflow management system: a system that defines, manages, and executes business processes using software. The order in which business activities are executed in the process is driven by a computer representation of the process called a workflow. A workflow automates a business process by controlling the sequence of activities in the process, and by invoking the appropriate resources required by the various activities in the process. These resources might be software components that perform a required activity, or they might be people who respond to messages generated by the business process.

To achieve this level of workflow automation, a workflow management system provides support in three broad areas:

- Workflow definition and execution—capturing the definition of the workflow, managing the execution of workflow processes in an operational environment, and sequencing the various activities to be performed.
- Data administration—managing the people involved in a workflow, rerouting tasks when needed, and maintaining business workflow calendars that govern the times at which workflows start and end.
- Workflow monitoring—tracking the status of workflow processes, and dynamically configuring the workflow run-time system.

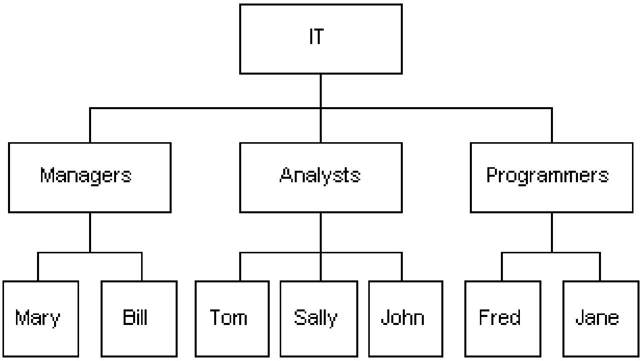
BEA WebLogic Process Integrator supports all three areas of workflow management. BEA WebLogic Process Integrator supports the execution of workflows using its run-time process engine, and the definition and monitoring of workflows with the BEA WebLogic Process Integrator Studio client application.

Business Organizations and Workflows

Businesses typically structure themselves hierarchically, starting with organizations. Each organization can represent a business function, such as accounting, human resources, or information technology, and might operate independently with separate management authority. Within each organization, a set of roles is defined. For example, in an information technology organization, the roles of manager, analyst, and programmer might be defined. For each role, specific individuals are assigned to perform the tasks within the role.

The following figure shows this kind of hierarchical business model.

Figure 1-2 Hierarchical Business Organization



BEA WebLogic Process Integrator workflows are organized around this kind of business model, with organizations defined first, followed by roles for each organization, and users for each role.

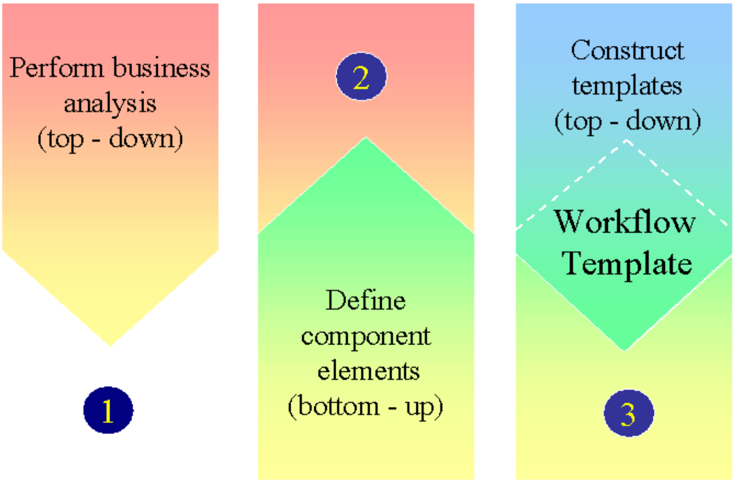
Organizations, roles, and users can be defined as follows:

- Organization—an entire company; a division, region or location within a company; a geograhical location; or any other distinction that is relevant to the particular business of a company. Using BEA WebLogic Process Integrator, one or more workflows can be defined for an organization.
- Role—a common area of responsibility, ability, or authorization level that is shared by a group of individuals who belong to a particular organization.
- User—an individual assigned to a role who has the necessary permission to perform a particular task in the workflow, such as responding to messages generated by a workflow.

Workflow Design and Integration

Theories of system design suggest that, ideally, systems should be designed using a top-down approach. In reality, systems are designed from both the top down and the bottom up. Both approaches are also used for the design of automated workflows, as the following figure shows.

Figure 1-3 Designing Workflows


Perform business analysis (top - down)
1
2
Define component elements (bottom - up)
Construct templates (top - down)
Workflow Template
3

In this design process, business analysts perform three steps:

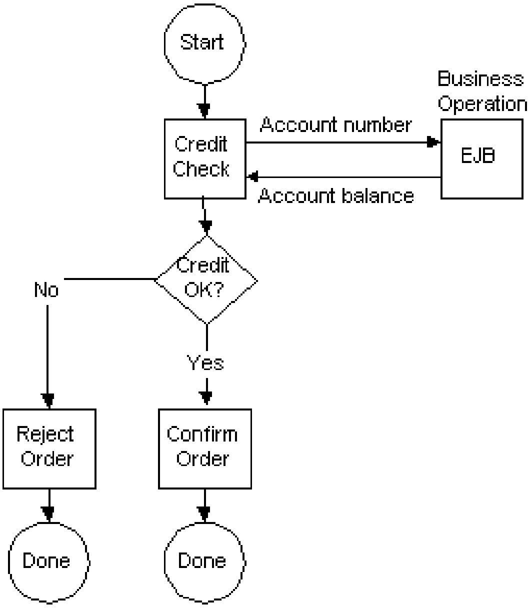
1. Define the main process flow.
2. Identify the components required by the workflow.
3. Integrate individual components into the overall design to produce the workflow template.

The template represents the flow of the business process with all of the necessary activities linked in the proper sequence. This workflow representation of the business process is not, however, fully automated at this point.

As we mention in “Workflow Automation and Management” on page 1-3 automated workflows invoke software components to perform the activities required by a business process, such as checking the account balance of a particular customer. When invoking a software component, the workflow can also pass some information to the component to identify the customer, such as an account number. The software component performs some processing to retrieve the account balance, and then returns that information to the workflow.

The software components that the workflow invokes can be built specifically for the workflow being designed, or they can be existing applications invoked by the workflow. These software components are typically Enterprise JavaBeans (EJBs), Java classes in Java applications, or application executables. To make the workflow template fully automated, the template must be integrated with these software components, as the following figure shows.

Figure 1-4 Integrating a Workflow with Business Operations

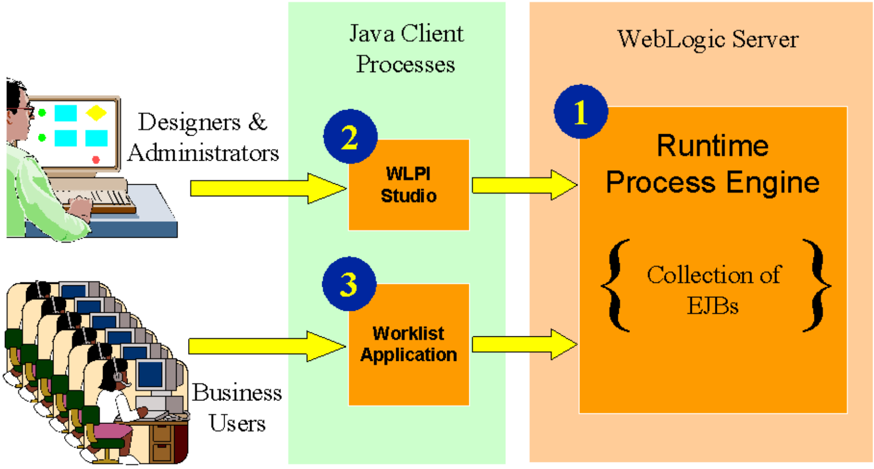


In BEA WebLogic Process Integrator, the functions performed by EJBs and Java classes, and called by workflows, are called *business operations*. The information that is passed between workflows and software components are stored in entities called *variables*. You use the BEA WebLogic Process Integrator Studio to design workflow templates and integrate them with software applications by specifying the business operations invoked by a workflow, and the variables used by a workflow to store information.

BEA WebLogic Process Integrator Components

BEA WebLogic Process Integrator is a Java 2 Enterprise Edition (J2EE) implementation of a workflow management system. BEA WebLogic Process Integrator consists of three components, as shown in the following figure.

Figure 1-5 BEA WebLogic Process Integrator Components



The central component is the run-time process engine which implements the functionality of BEA WebLogic Process Integrator through a collection of EJBs. The BEA WebLogic Process Integrator Studio and the Worklist are client applications that access the services of the process engine. Each of these components is described more fully in the following sections.

Run-Time Process Engine

The central component of BEA WebLogic Process Integrator is the run-time process engine. The process engine is built on the WebLogic Server infrastructure, so installing WebLogic Server is a prerequisite for using BEA WebLogic Process Integrator. The process engine also requires a JDBC database to store persistent data, such as workflow templates.

The process engine is a collection of EJBs that provide the following services:

- Creation and maintenance of workflow templates
- Startup and termination of running workflows
- Run-time control and management of workflow instances as they progress from task to task
- Monitoring of the status and performance of workflows

Studio Client Application

The BEA WebLogic Process Integrator Studio is a client application with a graphical user interface that business analysts, system administrators, and others can use to perform the following tasks:

- Define workflow templates
- Administer data by:
  - Adding, updating, and removing organizations, roles, and users
  - Assigning users to and removing them from roles
  - Configuring levels of permission for users
  - Rerouting tasks from one user to another for a specified period of time
  - Creating business calendars that are used to control the execution of workflows
- Monitor workflows by:
  - Displaying the status of running workflows

- Modifying the tasks associated with running workflows, such as by reassigning tasks or by forcing work to be redone
- Displaying the workload status of the system in graphical form
- Viewing historical data for workflows to identify bottlenecks and inefficiencies
- Viewing user or role worklists to maintain workflows dynamically

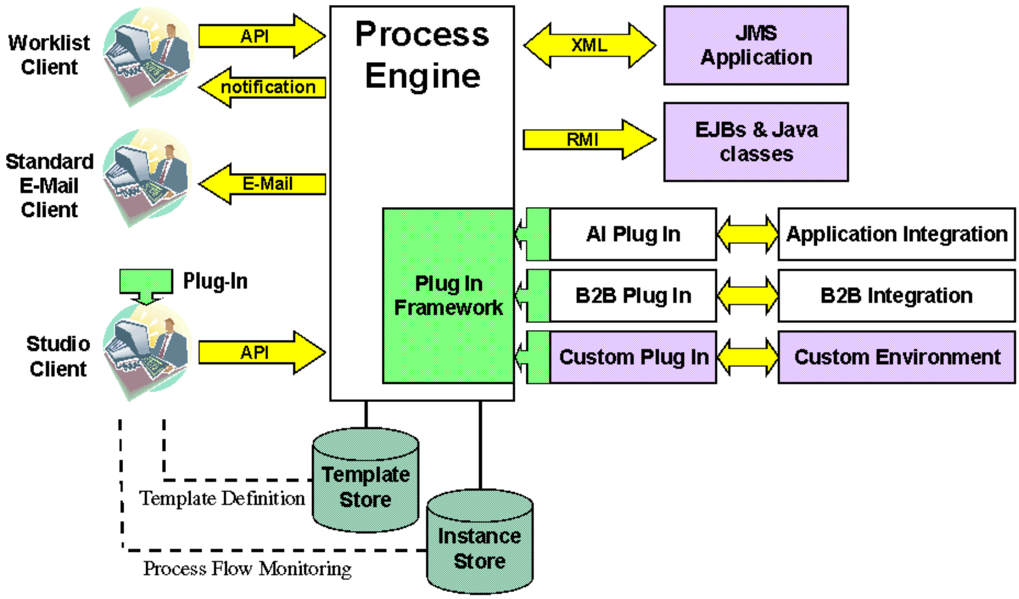
Worklist Client Application

Automated workflows sometimes need input from people. This input might include making discretionary decisions (for example, about credit information), handling exceptions, or troubleshooting problems in the workflow. The Worklist application is a client program that enables people to interact with a running workflow. That interaction is typically performed through a graphical user interface. The interface lists tasks for specific users or roles. Users or roles interact with the workflow by performing the tasks assigned to them.

System Architecture

The following figure shows the overall architecture of BEA WebLogic Process Integrator.

Figure 1-6 BEA WebLogic Process Integrator System Architecture



In addition to the process engine and the Studio and Worklist clients, the system architecture includes the following components:

- Template store — holds workflow templates created using the Studio. The template store is implemented through a JDBC database, such as Oracle or SQL Server.
- Instance store —holds running instances of workflows. When started, a workflow is converted to a workflow instance. For any workflow, multiple instances can be running at the same time. System administrators can monitor workflow instances using the Studio. The instance store is implemented through a JDBC database, such as Oracle or SQL Server.
- Plug-in framework — enables third-party developers to extend the functionality provided by BEA WebLogic Process Integrator. The Framework also enables IT departments to seamlessly integrate their systems and applications with BEA WebLogic Process Integrator, thereby simplifying the workflow definition process. Plug-ins are Java classes that are loaded at run time, and that can modify the definition and run-time behavior of selected workflow components.

During run time, the process engine can interact with other applications by performing the following tasks:

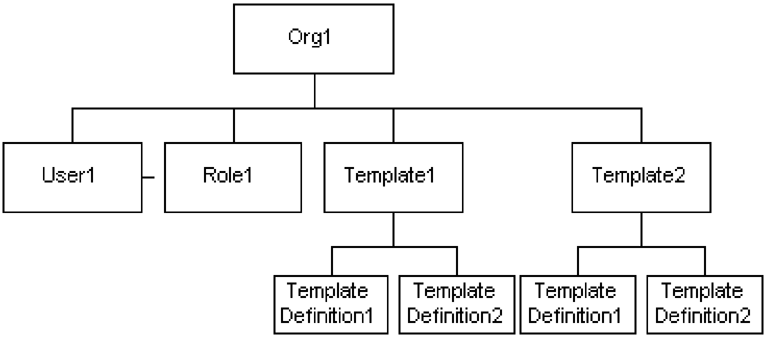
- Asynchronously sending or receiving application events in the form of XML documents or JMS messages
- Invoking EJB components and Java classes through an RMI call
- Sending e-mail messages to users of the BEA WebLogic Process Integrator system or outside parties using the SMTP protocol

The process engine also provides a public API that enables external applications to access process engine services.

Defining a Workflow

The BEA WebLogic Process Integrator workflow model is organized hierarchically, as the following figure shows.

Figure 1-7 BEA WebLogic Process Integrator Workflow Model



In this scheme, organizations are at the top. Roles and workflow templates are attached to an organization, and workflow template definitions are attached to workflow templates. Users belong to an organization, and may also be associated with roles.

To define a workflow according to this model, you perform the following steps using the BEA WebLogic Process Integrator Studio:

1. Specify an organization and its roles, and the users that belong to them.
2. Create a workflow template for the organization you specified.

   A workflow template is a folder that represents a workflow, and is attached to an organization. Give the folder a meaningful workflow name, such as Order Processing or Billing.
3. Create one or more workflow template definitions within the workflow template folder.

   A workflow template definition is a version of the workflow template. You distinguish between workflow template definitions by their effective and expiration dates. For example, you can have one workflow template definition that is valid from January to June, and a second one that is valid from July to December. At run time, the process engine starts an instance of the workflow template definition by selecting the most effective definition.

   Each workflow template definition contains workflow components. These components include the graphical representation of the workflow (called nodes), and the actions, variables, and exceptions handlers that are part of the workflow.

Workflow Components

You create a workflow template definition using four types of components: nodes, actions, variables, and exception handlers. This section describes these components.

Nodes

Nodes are geometric shapes that you use to graphically depict a workflow. The Studio provides the following six nodes:

- Start — represents the beginning of a workflow. A workflow can have the following types of start nodes: predefined, timed, manually initiated, event, or plug-in defined.
- Task — represents a unit of work or discrete activity in the workflow. During the execution of a workflow, task nodes progress through four different stages: created, activated, executed, and done. You can specify task actions in each stage. The task actions are executed as each stage occurs.

- Event — represents a wait state that is activated by the receipt of an XML message. Once an event node is activated, any actions defined within the event node are executed, and the process flow proceeds to the next node.
- Join — represents one of two conditions that must be fulfilled before the node that follows the join node is activated. There are two types of joins: AND and OR. In an AND join, the nodes linked by the join node must be satisfied before the node that follows the join node is activated. In an OR join, only one of the nodes linked by the join node must be satisfied before the node that follows the join node is activated.
- Decision — represents a condition (specified in the node) that must be evaluated. The result of the evaluation determines the path the workflow follows. Conditions can evaluate to either true or false. If a condition is verified, it is true. If it is not, it is false.
- Done — represents the end of a workflow.

Actions

Actions are units of work that you can specify in any node except a join node. An action can be as simple as assigning a task to a user, or as complicated as sending and receiving XML messages. There are several types of actions:

- Task actions — used to specify and control tasks. For example, you can specify task actions such as execute a task, mark a task as done, or assign a task to a user or a role.
- Workflow actions — used to manage an entire workflow, such as stopping the current workflow or starting a subworkflow.
- Integration actions — used to integrate the workflow with external software components or applications. This can be done by, for example, calling an executable program or sending an XML message to another application.
- Exception handling actions — used to invoke and exit an exception handler, and to make an exception handler the active one for a workflow.
- Miscellaneous actions — used to perform additional actions, such as sending e-mail, making an audit entry, or canceling a workflow event.
- Custom actions (plug-ins) — actions that you define and integrate into the workflow.

Variables

Variables hold run-time values and data obtained usually from outside sources, such as external applications.

Variables can hold values that are returned by business operations, that are extracted from XML documents, or that are set explicitly by workflow actions. Variables can also be used by the workflow for several other purposes, such as to evaluate a condition in a decision node, or to store the result of a response from the Worklist client application.

BEA WebLogic Process Integrator supports the following types of variables: Boolean, date, double, entity EJB, integer, Java object, session EJB, string, and XML.

Exception Handlers

Exception handlers are used to generate, trap, and respond to exception conditions that are generated internally or externally. Exceptions can be abnormal conditions that you define in the workflow and that you want to trap, or they can be server exceptions that you trap and respond to accordingly. Workflows can contain exception handlers that are provided by the system or that are defined by you.

CHAPTER

2 Getting Started with the Studio

The following sections describe the user interface of the Studio, and how to log in to and exit from the Studio:

- Logging In to the Studio
- Overview of the Studio Interface
- Toggling Between Normal and Interface View
- Exiting the Studio

Logging In to the Studio

1. Select the first step in this procedure that is appropriate for your platform and working environment:

   - To start the Studio from the Start Menu on a Windows system, go to step 2.
   - To start the Studio from a Command Prompt window on a Windows system, go to step 3.
   - To start the Studio on a UNIX system, go to step 4.

2. Choose Start→Programs→BEA WebLogic E-Business Platform→WebLogic Integration 2.0→Process Integrator→Start Studio to display the Logon dialog box.

   Go to step 5.

3. Go to the following directory:

   ```
   wlintegration2.0\processintegrator\bin
   ```

   `wlintegration2.0` is the product directory in which you installed the BEA WebLogic Process Integrator software.

   Type `studio` at the prompt to display the Logon dialog box.

   Go to step 5.

4. Go to the following directory:

   ```
   wlintegration2.0/processintegrator/bin
   ```

   `wlintegration2.0` is the product directory in which you installed the BEA WebLogic Process Integrator software.

   Execute the `studio.sh` script by entering the following command at the prompt:

   ```
   sh studio.sh
   ```

   The Logon dialog box is displayed.

5. Enter your user name and password in the appropriate fields. If you have not yet been assigned a user name and password for WebLogic Process Integrator, enter a default user name and password. For a list of default user names and passwords, see “WebLogic Process Integrator Users and Passwords” in *Installing and Configuring BEA WebLogic Process Integrator*.

6. In the Server [:port] field, specify the system that is running the WebLogic Process Integrator server application as follows:

   `t3://`*`host`*`:`*`port`*

   - *`host`* is the computer name or IP address of the system that is running the WebLogic Process Integrator server. Specify `localhost` if the server is running on the same computer as the Studio application.
   - *`port`* is the number you specified for the listen port when you installed BEA WebLogic Server. The default is 7001

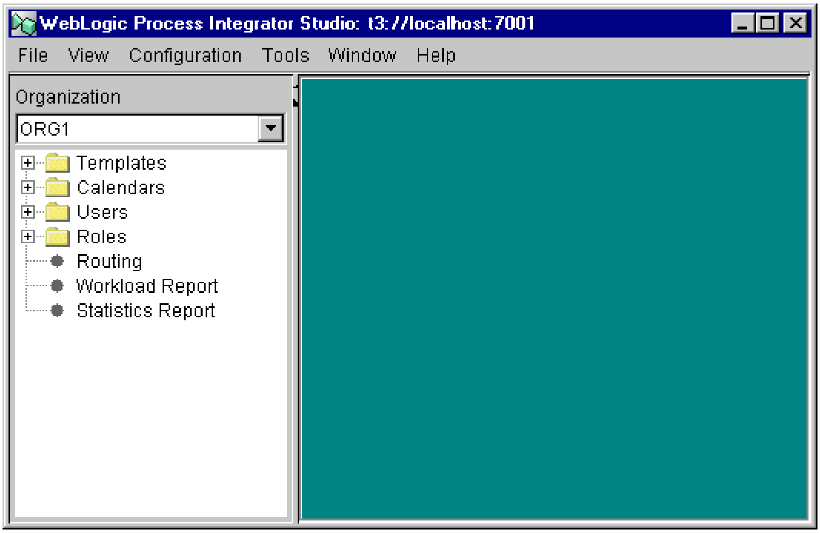
   To log in to clustered servers, enter the following in the Server[:port] field:

   `t3://`*`host1`*`,`*`host2`*`,`*`host3`*`:`*`port`*

   In this case, *`host1`*, *`host2`*, and *`host3`* are the computer names or IP addresses of the clustered WebLogic Process Integrator servers.

7. Click OK to display the main window of the Studio.

Figure 2-1 BEA WebLogic Process Integrator Studio Main Window



Overview of the Studio Interface

This section describes the parts of the Studio user interface, and the functions they provide.

Menu Options

The following sections provide information about using the menu bar in the Studio.

File Menu

Choose File→*Option* to perform the functions described in the following table.

Menu Option	Function
Logon	Log on to the BEA WebLogic Process Integrator server.
Logoff	Log off from the BEA WebLogic Process Integrator server.
Exit	Log off from the BEA WebLogic Process Integrator server and exit the BEA WebLogic Process Integrator Studio.

View Menu

Choose View→*Option* to perform the functions described in the following table.

Menu Option	Function
Refresh	Update the information if changes are made to the database by other client applications.
Use color on flowcharts	View the workflow diagrams in color or black and white.
Sync selection with tree	Synchronize the workflow components in the folder tree hierarchy with the workflow components in the drawing area. For example, clicking the Start node in a workflow diagram opens the corresponding Start node folder in the folder tree hierarchy.
Look and feel	Change the look and feel of the Studio display. The choices are: ■ Metal ■ CDE/Motif ■ Windows
Interface View . . .	Display additional visual objects to represent subworkflows, XML documents, and Plug-ins. Selecting this menu item displays the Interface View Preferences dialog box. For details about setting interface view preferences, see “Setting Interface View Preferences” on page 2-15. If a start, event, or done node contains a customized property defined by a plug-in, a small plug-in icon is displayed in the upper-right corner of the node icon.

Configuration Menu

Choose Configuration→*Option* to perform the functions described in the following table.

Menu Option	Function
Organizations	Define organizations that represent different business entities, geographical locations, or any other distinction relevant to the particular business of the company. For details about defining organizations, see "Maintaining Organizations" on page 4-5.
Business Operations	Define business operations representing a method call on an EJB or Java Class instance. For details about defining business operations, see "Defining a Business Operation" on page 4-37.
Events	Define event key expressions. For instructions about defining event key expressions, see Appendix A, "Understanding Event-Driven Processing."
Plugins	View and configure the available plug-ins. For details about configuring plug-ins, see Chapter 3, "Working with Plug-Ins."
Permissions	Define the user and role permission levels. For details about defining permission levels, see "Setting Permissions for Users" on page 4-23 and "Setting Permissions for Roles" on page 4-14.
Role Mappings	Map the currently-defined roles to WebLogic Server groups. For details about mapping roles, see "Changing the Mapping for Roles" on page 4-13.

Tools Menu

Choose Tools→*Option* to perform the functions described in the following table.

Menu Option	Function
Export Package . . .	Export workflow objects as a jar file. For details about exporting packages, see "Exporting and Importing Workflow Packages" on page 5-24.

Menu Option	Function
Import Package . . .	Import workflow objects from a jar file. For details about importing packages, see “Exporting and Importing Workflow Packages” on page 5-24.
Show XML Finder . . .	Display the XML finder dialog box to manage the XML repository. For details about managing the XML repository using the XML finder, see “Managing the XML Repository” on page C-7.

Help Menu

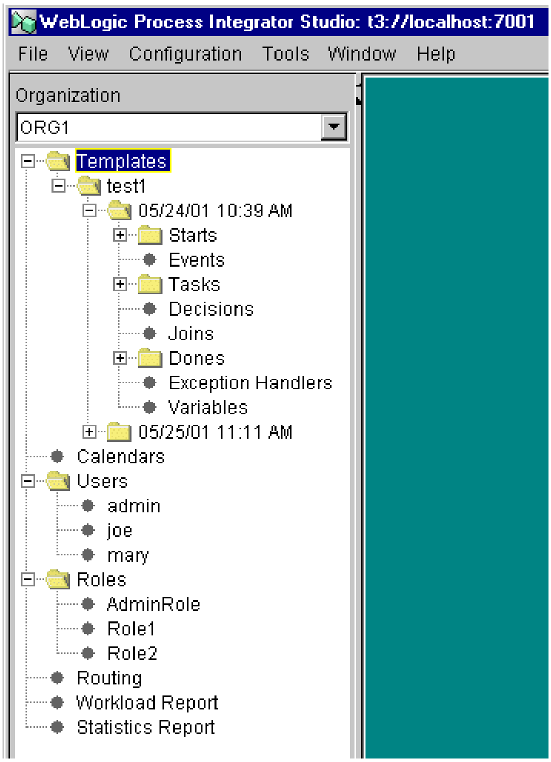
Choose Help→*Option* to perform the functions described in the following table.

Menu Option	Function
Help Topics	Access the online help files for the Studio.
Plugin Help	Access the online help files for the loaded plug-ins.
About WebLogic Process Integrator Studio . . .	Provide software version information about the BEA WebLogic Process Integrator Studio.

Folder Tree Display

The BEA WebLogic Process Integrator Studio interface contains a folder tree display, which organizes workflow components in a standard tree structure.

Figure 2-2 BEA WebLogic Process Integrator Folder Tree Display



Organization
ORG1

Note: Information is displayed for the current organization only. To select a different organization, use the Organization drop-down list at the top of the screen.

To view details about any of the following items shown in the folder tree display, double-click the item to expand it:

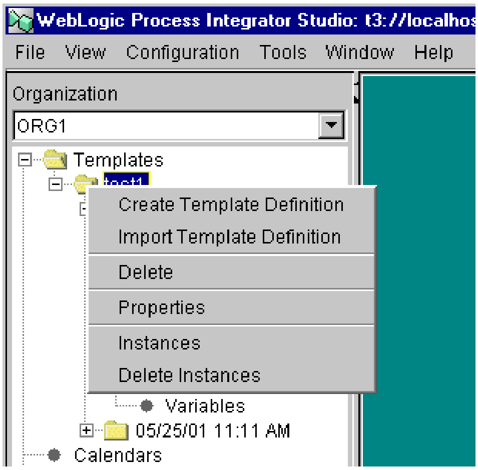
- Templates
- Calendars
- Users
- Roles
- Routing
- Workload Report
- Statistics Reports

For example, double-click the Templates folder to display a list of workflow templates. Double-click a workflow template to display a list of all workflow template definitions. Expanding a particular workflow template definition displays folders containing the tasks, decisions, events, joins, starts, dones, and variables for that workflow template definition.

You can right-click most items in the folder tree to display a menu containing options relevant to the highlighted item. Choosing any menu option displays a corresponding dialog box on the right-hand side of the screen. However, right-click menus in the folder tree are context specific. Depending on the context in which you are accessing the right-click menu, certain options may not be accessible. For example, if your security realm does not permit you to add users and roles to WebLogic Process Integrator, the Create Role and Add Users menu options will not be accessible to you.

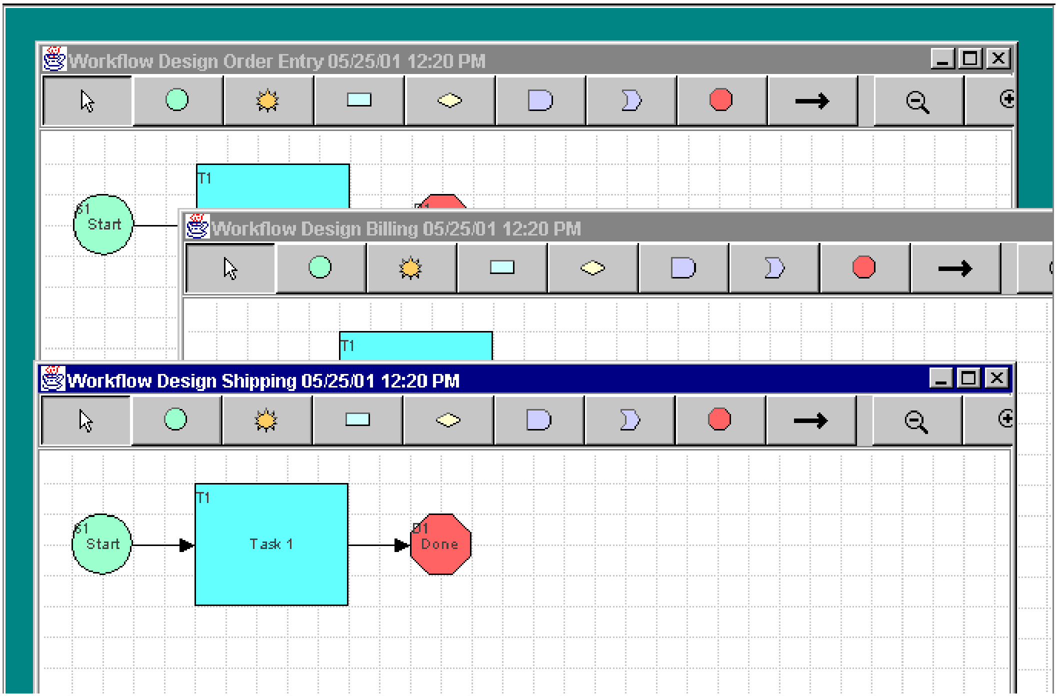
For example, right-click on an existing workflow template to display a menu with the options shown in the following figure:

Figure 2-3 Workflow Template Menu Options


WebLogic Process Integrator Studio: t3://localhos
File View Configuration Tools Window Help
Organization
ORG1
Templates
test1
Create Template Definition
Import Template Definition
Delete
Properties
Instances
Delete Instances
Variables
05/25/01 11:11 AM
Calendars

You can have multiple workflow diagrams open simultaneously. You can view more than one workflow, and toggle back and forth between them.

Figure 2-4 Multiple Workflow Diagrams


Workflow Design Order Entry 05/25/01 12:20 PM
Workflow Design Billing 05/25/01 12:20 PM
Workflow Design Shipping 05/25/01 12:20 PM
T1
S1
Start
Task 1
D1
Done

Workflow Drawing Area and Toolbar

The workflow drawing area, shown in the following figure, is where you create your workflow diagrams. The workflow toolbar contains the shapes representing workflow nodes and the connector you use to define a workflow.

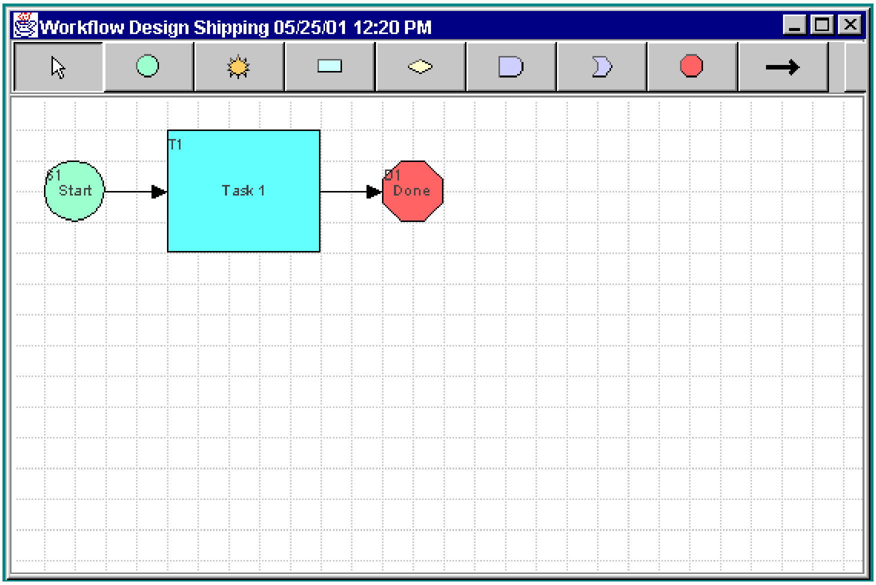
To open the workflow drawing area, you either create a new workflow template definition or open an existing workflow template definition.

Once the drawing area is open, you can perform the following tasks:

- Place a shape in the drawing area by clicking the shape on the toolbar, placing your cursor on the drawing area, and then clicking again to drop the shape onto the drawing area.
- Move a shape within the diagram by clicking and dragging the shape with your mouse.

Refer to Chapter 5, "Defining and Maintaining Workflows," for detailed instructions on performing these tasks.

Figure 2-5 Workflow Drawing Area



The following table describes the various nodes the Studio provides to define a workflow template definition.

Table 2-1 Workflow Template Definition Nodes

Symbol	Shape Name	Purpose
🟢	Start	Indicates the start of the workflow.
▭	Task	Defines a task in the workflow.
◇	Decision	Represents a condition in the workflow that evaluates to True or False. (If the condition is verified, it is True. If it is not, it is False.)
⊃	And	Allows joining of one or more task, decision, or event nodes with an AND condition.
☽	Or	Allows joining of one or more task, decision, or event nodes with an OR condition.
🔴	Done	Indicates the end of the workflow.
✹	Event	Represents an event that can be triggered either internally or externally by an XML message. Subactions can be performed and/or workflow variables can be set as the result of the trigger of the event.
→	Draw Connection	Used to connect workflow nodes. The arrow directs you to the subsequent task in the flow. **Note:** In the case of a Decision, when the arrow is drawn from a Decision shape, the Create Connection dialog box prompts you to specify whether the connection is True or False.

Relocating the Toolbar

By default, the workflow toolbar appears at the top of the drawing area. Depending on the size of the Studio window or drawing area, you might not be able to see the entire toolbar. To see the entire toolbar, relocate the toolbar to another part of the Studio drawing area, or to another location on your desktop.

To relocate the toolbar:

1. Click the background area of the toolbar (for example, the space between the Draw Connection and Zoom Out shapes).
2. Drag the toolbar to the desired location.

If you drag the toolbar to another location on your desktop outside of the Studio drawing area, the toolbar becomes a separate window on your desktop, which you can minimize, maximize, and close.

Toggling Between Normal and Interface View

Workflows often interface to different objects, such as subworkflows and business operations. At times, it might be useful to see the objects to which a workflow interfaces. You can view workflows using two views: normal and interface.

A normal view displays the components in the workflow diagram, but not the objects to which the workflow interfaces. An interface view displays the components in the workflow diagram, plus the following objects to which the workflow can interface:

- Subworkflows — other workflows to which the current workflow connects
- Business operations — software components, such as EJBs and Java classes, that the workflow invokes to perform a function
- Inbound and outbound XML documents — XML documents that are coming into the workflow, or that the workflow is sending to another workflow or application.
- Plug-ins — custom workflow components

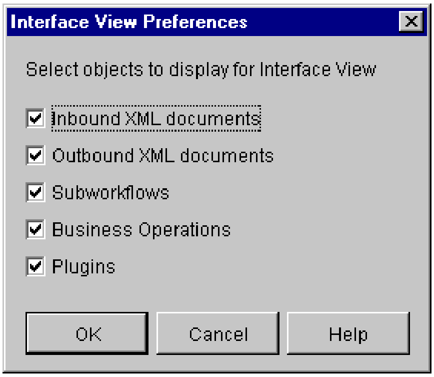
To toggle between a normal and interface view, click the following button in the toolbar:

Setting Interface View Preferences

Within an interface view, you can choose which objects to display in the workflow diagram. To specify the objects you want to view:

1. Choose View→Interface View to display the Interface View Preferences dialog box.

Figure 2-6 Interface View Preferences Dialog Box



Interface View Preferences

Select objects to display for Interface View

☑ Inbound XML documents

☑ Outbound XML documents

☑ Subworkflows

☑ Business Operations

☑ Plugins

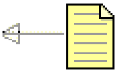
2. Select the check boxes for the objects you want to display, or clear the check boxes for the objects you do not want to display.

   The settings are system-wide; that is, they apply to all template definitions. They are saved from session to session in the Studio; they are not saved as part of the template definition.

For details about displaying workflow objects in an interface view, see “Toggling Between Normal and Interface View” on page 2-14.

Viewing Inbound XML Document Data

In an interface view, the following icon represents an inbound XML document in the workflow diagram.

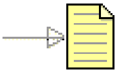


Note: The arrow points toward the parent template node to indicate inbound direction.

An inbound XML document icon is displayed for start and event nodes that respond to incoming XML documents. Hover the mouse pointer over the XML document icon in the workflow diagram to display a text box containing the type (inbound or outbound), root, and key of the XML document.

Viewing Outbound XML Document Data

In an interface view, the following icon represents an outbound XML document in the workflow diagram.



Note: The arrow points away from the parent template node to indicate outbound direction.

An outbound XML document icon is displayed if either a Send XML to Client or Post XML Event action is defined for an object. You can obtain information about the XML document as follows:

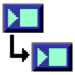
- Hover the mouse pointer over the XML document icon in the workflow diagram to display a text box containing the type (inbound or outbound), root, and key of the XML document.

- Double-click the XML document icon to display the dialog box for the defined action; either Send XML to Client or Post XML Event. Alternatively, you can select the icon, right-click on it and select Properties from the pop-up menu to display the appropriate dialog box.

For details about the Send XML to Client and Post XML Event actions, see “Using Integration Actions” on page 7-32.

Viewing Subworkflow Data

In an interface view, the following icon represents a subworkflow in the workflow diagram.



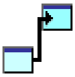
A subworkflow icon is displayed if a Start Workflow action is defined for an object. You can obtain information about the subworkflow as follows:

- Hover the mouse pointer over the subworkflow icon to display a text box containing the name of the called subworkflow.
- Double-click the icon to display the Start Workflow dialog box. Alternatively, you can select the subworkflow icon, right-click the icon, and select Properties to display the Start Workflow dialog box.

For details about the Start Workflow action, see “Start Workflow” on page 7-25.

Viewing Business Operation Data

In an interface view, the following icon represents a business operation in the workflow diagram.



A business operation icon is displayed if a Perform Business Operation action is defined for an object. You can obtain information about the business operation as follows:

- Hover the mouse pointer over the business operation icon in the workflow diagram to display a text box containing the name of the business operation to be performed.
- Double-click the business operation icon to display the Perform Business Operation dialog box. Alternatively, you can select the icon, right-click the icon, and select Properties to display the Perform Business Operation dialog box.

For details about business operations, see “Defining a Business Operation” on page 4-37.

Viewing Plug-In Data

In an interface view, the following icon, or a custom icon, represents a plug-in action in the workflow diagram.

If a node contains a plug-in action, you can obtain information about the action as follows:

- Hover the mouse pointer over the plug-in icon in the workflow diagram to display a text box containing a description of the plug-in action.
- Double-click the plug-in icon to display the corresponding plug-in action dialog box. Alternatively, you can select the icon, right-click the icon, and select Properties to display the plug-in action dialog box.

Note: If a start, event, or done node contains a customized property defined by a plug-in, a small plug-in icon is displayed in the upper-right corner of the node icon.

Exiting the Studio

To exit the Studio:

1. If you want to save changes to a workflow, right-click the workflow template definition and choose Save from the menu that is displayed.
2. Choose File→Exit, or select the System Close box. A dialog box asks you to confirm your decision. Click Yes to exit.

Note: If you try to exit the Studio without saving your changes, a dialog box prompts you to do so.

CHAPTER

3 Working with Plug-Ins

BEA WebLogic Process Integrator provides a plug-in framework that developers can use to extend the functionality provided by BEA WebLogic Process Integrator. The following sections explain plug-ins and how to work with them:

- What Is a Plug-In?
- Viewing Plug-Ins
- Loading Plug-Ins
- Configuring Plug-Ins

What Is a Plug-In?

A plug-in is a group of Java classes that extend the functionality provided in selected workflow components. Plug-ins provide a way to customize existing BEA WebLogic Process Integrator features so they are more appropriate for your environment, and to add functionality that is specific to your environment.

You access the functionality defined by a plug-in through existing Studio dialog boxes that have been modified to include a selection for the plug-in. For example, BEA WebLogic Process Integrator provides several default methods in the Start node to trigger the start of a workflow: timed, manual, called, and event. You select the trigger method using the Start Properties dialog box.

To extend the default methods, a developer can create a plug-in that specifies a custom event to trigger a workflow, such as receiving an e–mail message. This might be the preferred method for starting workflows in your environment. For you to be able to select the custom event, the developer also modifies the Start Properties dialog box so it contains a selection for the custom event.

A plug-in can extend the functionality of the following workflow components:

- Workflow templates
- Workflow template definitions
- Start nodes
- Event nodes
- Done nodes
- Variables
- Actions
- Functions (that are part of expressions)

If a plug-in is developed for any of these workflow components, the Studio dialog box for these components is also modified to include a way for you to access the functionality provided by the plug-in.

Once a plug-in is developed and is deployed on WebLogic Server, it is available for use by BEA WebLogic Process Integrator. When BEA WebLogic Process Integrator starts, it checks WebLogic Server to see if any plug-ins are available on the server.

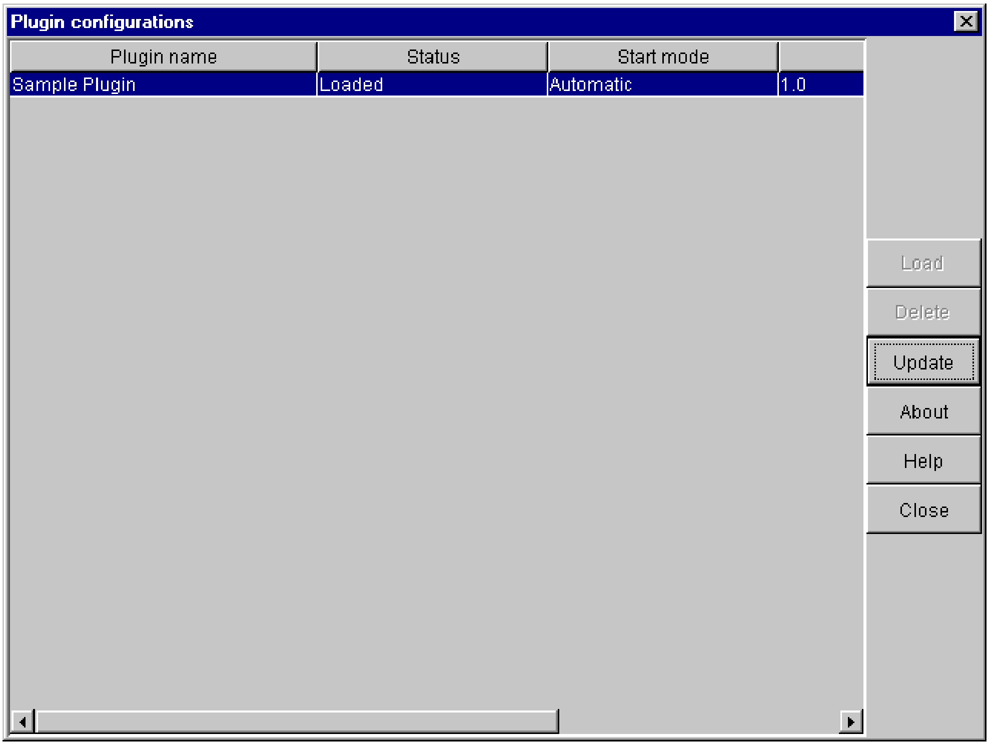
Before you can use an available plug-in, you must load it in BEA WebLogic Process Integrator to activate it. You might also need to specify some configuration settings for the plug-in before you can use it.

Note: To load or configure a plug-in, you must have the necessary permission. For details about permission levels, see "About Levels of Permission" on page 4-2.

Viewing Plug-Ins

To view a plug-in, choose Configuration→Plugins to display the Plugin configurations dialog box.

Figure 3-1 Plugin Configurations Dialog Box



Plugin configurations

Plugin name	Status	Start mode	
Sample Plugin	Loaded	Automatic	1.0

The information displayed for each plug-in is explained in the following table.

Plugin Name	The name for the plug-in as specified by the plug-in software.
Status	■ Loaded—The plug-in is loaded ■ Unloaded—The plug-in is available but has not been loaded ■ Missing—The plug-in has a registered configuration, but is not deployed or available.
Start mode	■ Automatic—The plug-in is loaded each time the server is started ■ Manual—The plug-in must be loaded manually each time you start the server ■ Disabled—The plug-in cannot be loaded

Version	The software version for the plug-in
Vendor	The company that supplied the plug-in

You can also obtain information about a plug-in by selecting the plug-in from the list and clicking About.

Loading Plug-Ins

To load a plug-in:

1. Choose Configuration→Plugins to display the Plugin configurations dialog box.
2. Select the plug-in you want to load.
3. Click Load.

   The status of the plug-in changes to Loaded in the list.

   You can configure the plug-in as described in the following section.

Configuring Plug-Ins

To configure a plug-in:

1. Choose Configuration→Plugins to display the Plugin configurations dialog box.
2. Select the plug-in you want to configure.
3. Click Update. A configuration dialog box is displayed.
4. Select the start mode for the plug-in by clicking one of the following buttons:
   - Automatic — Select this option if you want to load the plug-in each time the server is started

- Manual — Select this option if you want to load the plug-in manually as required
- Disabled — Select this option to disable the plug-in functionality

Note: The start mode does not become effective until the next time the server is started.

5. Specify configuration settings for the plug-in as appropriate.

   For details about defining configuration settings for a plug-in, see the online help for that plug-in. To access plug-in help, press the F1 key (for context-sensitive plug-in help), or choose Help->Plugin Help from the main menu in the Studio and select the appropriate plug-in help from the menu.

6. Click OK to complete the configuration or Cancel to cancel the operation.

Deleting a Configuration

You can delete a configuration for a plug-in if you no longer need the configuration. When you delete a configuration, you do not delete the plug-in itself. You just delete its registered configuration.

To delete a configuration:

1. From the main menu in the Studio, choose Configuration→Plugins to display the Plugin configurations dialog box.
2. Select the plug-in whose configuration you want to delete.

   Note: You cannot delete the configuration of a plug-in unless the status of that plug-in is Missing.

3. Click Delete. The registered configuration for the selected plug-in is deleted, but the plug-in remains visible in the Plugin configurations dialog box.

   The following table describes the actions that occur when you restart the server after a plug-in configuration has been deleted.

If the server is restarted and the plug-in is . . .	Then . . .
Not deployed	The plug-in manager does not search for the plug-in and the plug-in is not listed in the Plugin configurations dialog box.
Deployed	The plug-in is automatically loaded using the default configuration values defined by the plug-in.

CHAPTER

4 Administering Data

The following sections explain data administration concepts and tasks within BEA WebLogic Process Integrator:

- Understanding Security
- Configuring Security
- Maintaining Organizations
- Maintaining Users
- Administering Task Routing
- Administering Task Routing
- Administering Business Calendars
- Understanding Business Operations

Understanding Security

In BEA WebLogic Process Integrator, you maintain security for your workflows and their resources by doing the following:

- Mapping role and user information in BEA WebLogic Process Integrator to BEA WebLogic Server security realms
- Specifying levels of permission for roles and users to protect and control access to BEA WebLogic Process Integrator functions

About Security Realms

WebLogic Server provides security for applications through a service called a security realm. A security realm is a logical grouping of users and groups. A user is a specific individual who performs a certain task, such as programming or sales. A group is a collection of users who perform the same task. In this scheme, Group A might represent a collection of users who are programmers, and Group B might represent a collection of users who are sales people. Within a security realm, administrators can specify the levels of access users and groups have to workflows and other resources.

BEA WebLogic Process Integrator maintains information about roles and users in WebLogic Server security realms. When you define users and roles in BEA WebLogic Process Integrator, you need to specify their relationship to users and groups in WebLogic Server. You do this by mapping the roles you define in the Studio to the groups in a security realm in WebLogic Server.

About Levels of Permission

Levels of permission enable you to protect and control access to Studio functionality. Roles and users can perform the tasks shown in the following table only if they have the corresponding level of permission.

Table 4-1 Levels of Permission

Permission Level	Allows User or Role to . . .
Configure System	Make application configuration changes, such as adding, updating, and deleting business calendars
Configure Components	■ Define, update, and delete business operations ■ Load and configure plug–ins
Administer User	■ Manage users, roles, and organizations ■ Specify levels of permission for users and roles ■ Work with task routing specifications
Monitor Instance	■ Monitor instances, but not change them ■ Monitor business calendars, but not change them ■ Work with workload and statistics reports
Create Template	Create templates
Delete Template	Delete templates
Execute Template	Execute templates within a particular organization.

A role in BEA WebLogic Process Integrator inherits the levels of permission for the WebLogic Server group to which it belongs. In the Studio, you can add or remove levels of permission for a role. Any changes you make to a role are reflected in the group to which the role belongs, and affect any other roles that belong to the group.

A user inherits the levels of permission for role to which it belongs. In the Studio, you can add levels of permission for a user, but you cannot remove the permissions a user inherits from the role to which it belongs. Any permissions you add to a user are specific to the user, and are not reflected in the role to which the user belongs.

Available Functions

A security realm in WebLogic Server can be manageable or nonmanageable. This is defined when the realm is implemented on WebLogic Server. A manageable realm is one in which an application can make updates to groups and users. A nonmanageable realm is one in which an application can only list the groups and users

BEA WebLogic Process Integrator detects automatically whether a realm is manageable or nonmanageable. The type of security realm BEA WebLogic Process Integrator detects determines which data administration functions are available in the Studio.

You can perform the following tasks using the Studio dialog boxes only if BEA WebLogic Process Integrator detects a manageable realm:

- Map a role to a WebLogic Server group
- Add or delete a user
- Add a user to an organization
- Remove a user from an organization
- Add a user to a role
- Remove a user from a role

If BEA WebLogic Process Integrator detects a nonmanageable realm, you can perform the following tasks using the Studio dialog boxes:

- See a list of users and roles. The functions provided in the Studio dialog boxes to perform add, delete, and mapping tasks are dimmed, so you cannot select them.
- Update data managed by BEA WebLogic Process Integrator, and not by WebLogic Server in a security realm, such as a business calendar attached to a user, role, or organization.

If you want to add users to a nonmanageable realm, you must add them using the appropriate data administration functions for the realm. You cannot add them using the Studio dialog boxes.

Configuring Security

To configure security for roles and users, you need to perform tasks in both WebLogic Server and the Studio as follows:

1. In WebLogic Server, define a group in a security realm. For information about this task, see the WebLogic Server documentation.
2. In the Studio:
   a. Add an organization.

   b. Define a role within the organization, and map the role to a WebLogic Server group.

   c. If necessary, modify the levels of permission for the role. Roles inherit the permissions of the groups to which they belong. Changing the levels of permission for a role also changes the permissions in the group, and affects any other roles that belong to the group.

   d. Define users for the role.

   e. If necessary, define additional levels of permission for users. Users inherit the permissions of the role to which they belong, but you can add other levels of permission to users that are not defined for the role to which the users belong.

For information about performing these tasks using the Studio, see the sections that follow.

Maintaining Organizations

Use the Organization option to define organizations, which can represent different business entities, geographical locations, or any other distinction relevant to the particular business of the company. Organizations are specific to BEA WebLogic Process Integrator, and do not correspond to any groups on WebLogic Server.

The Organization drop-down list is located on the main menu of the BEA WebLogic Process Integrator main window. Selecting an organization from the drop-down list displays information in the folder tree for that organization only.

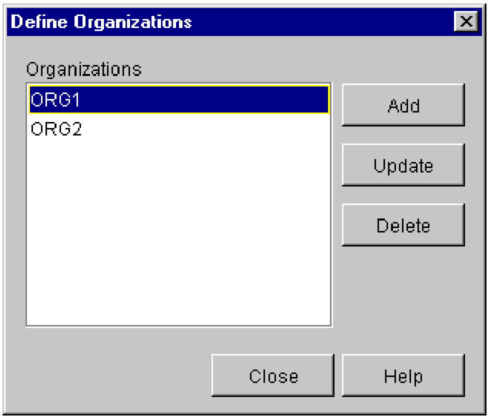
Adding an Organization

The Add Organization facility adds an organization to the WebLogic Process Integrator database.

To add an organization:

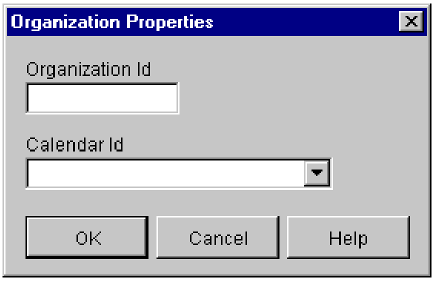
1. From the Configuration menu, choose Organizations to display the Define Organizations dialog box.

Figure 4-1 Define Organizations Dialog Box



2. In the Define Organizations dialog box, click Add to display the Organization Properties dialog box.

Figure 4-2 Organization Properties Dialog Box



3. Enter a meaningful ID for the organization in the Organization Id field.
4. From the Calendar Id drop-down list, select a calendar to assign to the organization. For details about this feature, see “Administering Business Calendars” on page 4-28.
5. Click OK.

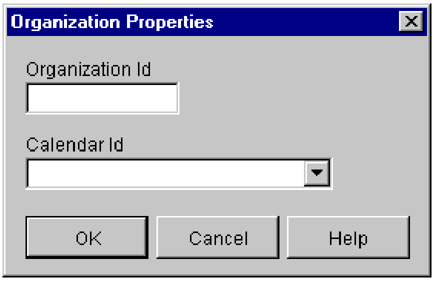
Updating an Organization

The Update Organization facility allows you to update the business calendar of an existing organization.

To update an organization:

1. From the Configuration menu, choose Organizations.
2. In the Define Organizations dialog box, highlight the organization to update.
3. Click Update to display the Organization Properties dialog box.

Figure 4-3 Organization Properties Dialog Box



4. Make changes as needed to the Organization Id or Calendar Id fields, and click OK.

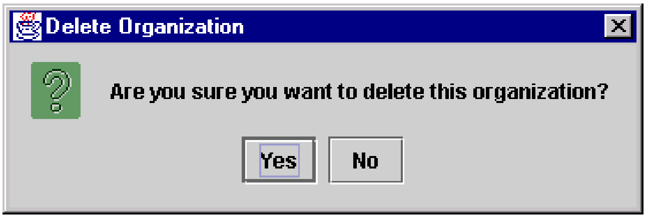
Deleting an Organization

The Delete Organization facility allows you to remove an organization from the WebLogic Process Integrator database.

To delete an organization:

1. From the Configuration menu, choose Organizations.
2. In the Define Organizations dialog box, highlight the organization to delete.
3. Click Delete to display the Delete Organization dialog box.

Figure 4-4 Delete Organization Dialog Box



Are you sure you want to delete this organization?

4. Click Yes to confirm or No to cancel.

Maintaining Roles

Use the Roles feature to create, update, and delete roles, and to map roles to WebLogic Server groups.

A role is a common area of responsibility, ability, or authorization level that is shared by a group of individuals. A role can only be a member of one organization, but you can use the same name in multiple organizations. For example, you can have a role named *Supervisor* defined in Org1 and Org2. The name of the role is the same, but, in actual fact, the roles are different. *Supervisor* in Org1 is not the same as *Supervisor* in Org2, even though the names are the same.

Roles are mapped to groups in WebLogic Server. Using the previous roles and organizations as an example, you can map *Supervisor* in Org1 to a group called *SupervisorOrg1*, and *Supervisor* in Org2 to a group called *SupervisorOrg2*.

Roles are displayed within each organization. Selecting an organization from the Organizations drop-down list in the BEA WebLogic Process Integrator main window and double-clicking the Roles folder displays the roles belonging to that particular organization. Double-clicking an existing role displays the properties for that role.

Right-clicking a role displays a menu with the following options:

- Properties — displays the Role dialog box, where you create a new role or update a selected role.
- Delete — delete the highlighted role from the WebLogic Server security realm. To delete a role, all of the role's tasks must first be reassigned to other roles.
- Open Role Worklist — opens a list of tasks assigned to the role. For details, see "Viewing Role Worklists" on page 10-13.
- Close Role Worklist — closes the list of tasks assigned to the role.

A role inherits the levels of permission of the group to which it belongs. After you create a role, you can specify different levels of permission for the role. Any changes you make to the levels of permission for a role are reflected in the group, and affect any other roles that belong to the group.

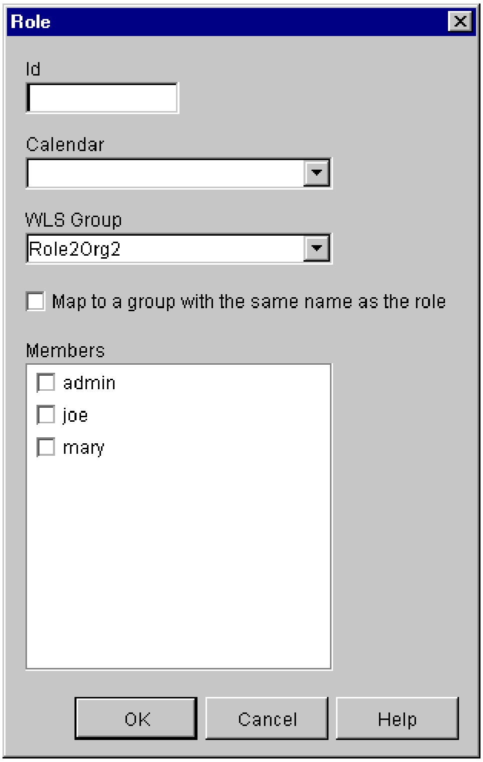
Creating a Role

The Create Role facility allows you to create a new role in the WebLogic Server security realm, and to map the role to a group in WebLogic Server. This facility is accessible only if WebLogic Process Integrator Studio is operating in a manageable realm.

To add a new role:

1. Right-click the Roles folder and choose Create Role from the menu to display the Role dialog box.

Figure 4-5 Role Dialog Box



Id

Calendar

WLS Group: Role2Org2

□ Map to a group with the same name as the role

Members

□ admin

□ joe

□ mary

2. Enter a meaningful ID for the role in the Id field.

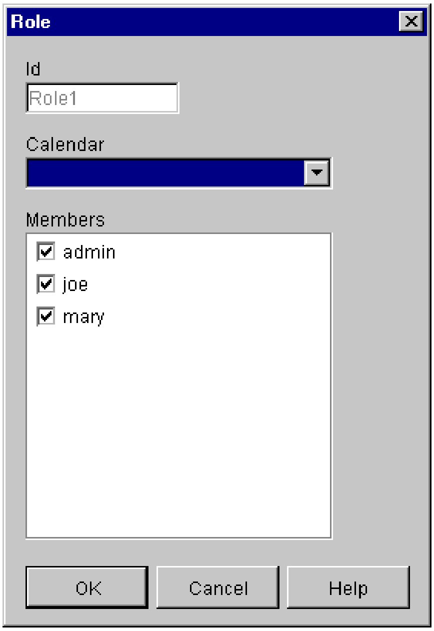
3. Use the Calendar drop-down list to assign a calendar to the role. For details about this feature, see "Administering Business Calendars" on page 4-28.
4. Use the WLS Group drop-down list to assign the role to a group in WebLogic Server.
5. To create a new group in WebLogic Server with the same name as the role, click the following check box: Map to a group with the same name as the role.
6. In the Members section of the dialog box, select the check box(es) to the left of the users that you will make members of this role.
7. Click OK to save the new role. Click Cancel to cancel the operation.

Updating a Role

To update an existing role:

1. Right-click the role and choose Properties from the menu to display the Role dialog box.

Figure 4-6 Role Dialog Box



Role

Id

Role1

Calendar

Members

☑ admin

☑ joe

☑ mary

2. Make changes as needed to the calendar and role members.

3. Click OK to save the update. Click Cancel to cancel the operation.

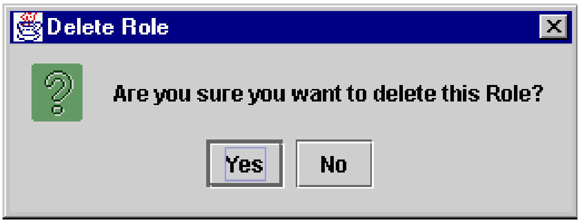
Deleting a Role

Use the Delete Role dialog box to delete a role from the WebLogic Server security realm and WebLogic Process Integrator database.

To delete a role:

1. Right-click the role and choose Delete from the pop-up menu to display the Delete Role dialog box.

Figure 4-7 Delete Role Dialog Box



Delete Role

Are you sure you want to delete this Role?

2. To delete the role, click Yes. To cancel the delete, click No.

Changing the Mapping for Roles

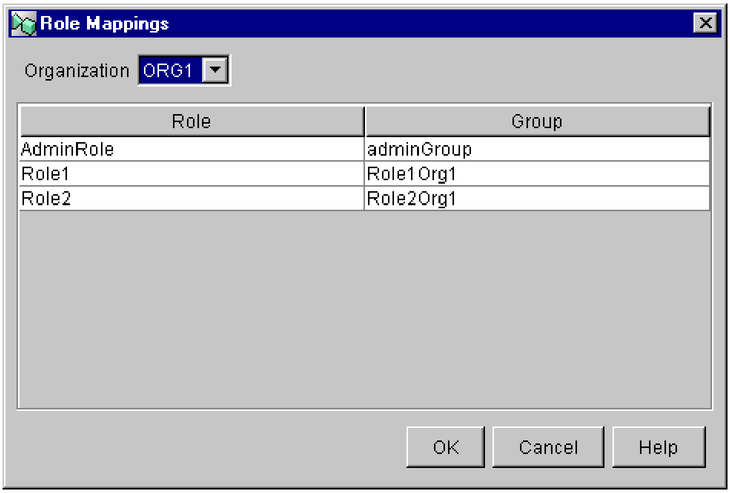
You can change the group in a security realm to which a role is mapped. You might want to do this if you decide that you want the role to be mapped to a group that is different from the one you mapped the role to when you created the role.

When you create a new role, you can map the role to an existing WebLogic Server group or you can create a new group. For details, see “Creating a Role” on page 4-10.

To change the mapping for a role:

1. From the main Studio window, choose Configuration→Role Mappings.The Role Mappings dialog box is displayed.

Figure 4-8 Role Mappings Dialog Box



Organization ORG1

Role	Group
AdminRole	adminGroup
Role1	Role1Org1
Role2	Role2Org1

2. From the Organization drop-down list, select the organization containing the roles you want to assign to WebLogic Server groups. The current mappings are displayed in a table.

3. In the table, select the role or group you want to remap.

   A drop-down arrow appears to the right of the group name to be changed.

4. Click the drop-down arrow and, from the drop-down list, select the new WebLogic Server group to which you want to map the role.

5. Click OK to complete the procedure, or Cancel to cancel the operation.

Setting Permissions for Roles

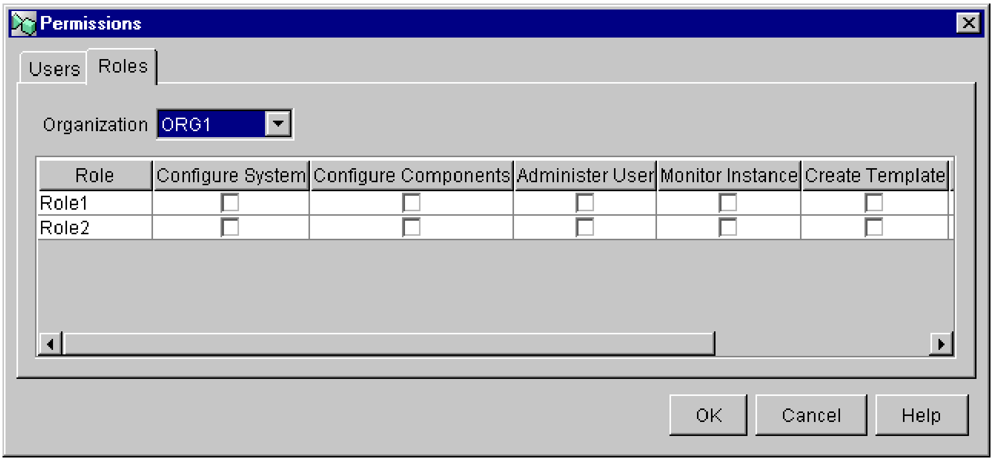
A role inherits the levels of permission for the WebLogic Server group to which it belongs. You can add and remove levels of permission for a role. Any changes you make to a role are also reflected in the group to which the role belongs.

To set the levels of permission for a role:

1. From the main Studio window, choose Configuration→Permissions. The Permissions dialog box is displayed

2. Select the Roles tab.

Figure 4-9 Permissions Dialog Box



Organization: ORG1

Role	Configure System	Configure Components	Administer User	Monitor Instance	Create Template
Role1	□	□	□	□	□
Role2	□	□	□	□	□

3. From the Organization drop-down list, select the organization for which you want to set the role permissions.

   The Permissions dialog box displays all the roles in the organization and the permissions currently assigned.

4. Select or clear the check boxes as desired.

5. Click OK to accept the changes, or Cancel to cancel the operation.

Maintaining Users

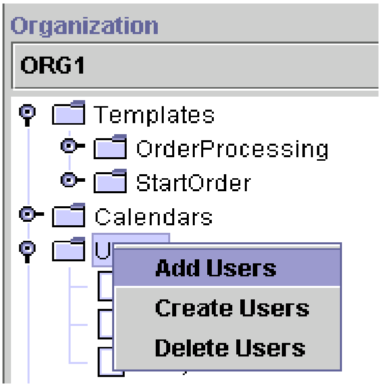
A user is an individual who has permissions to perform certain tasks. Use the Users feature to create, update, and delete users in a security realm in WebLogic Server. You can also add or remove users already defined in a security realm from organizations.

The User folder is located in the BEA WebLogic Process Integrator folder tree. Expanding it displays a list of users who have already been defined for the current organization. Double-clicking on an existing user displays the properties for that user.

Right-clicking the User folder displays a menu with the following options:

- Add Users — adds a user already defined in a WebLogic Server security realm to the current organization.
- Create Users — adds a user to a WebLogic Server security realm.
- Delete Users — deletes defined users from a WebLogic Server security realm.

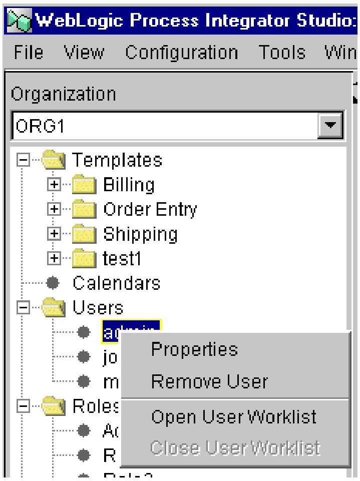
Figure 4-10 User Folder Menu Options



Right-clicking a user within the User folder displays a menu with the following options:

- Properties — displays the User dialog box, where you create a new user or update a selected user.
- Remove User — removes the highlighted user from the current organization. To remove a user, all of the user's tasks must first be reassigned to other users.
- Open User Worklist — opens a list of tasks assigned to the user. For details, see "Viewing User Worklists" on page 10-12.
- Close User Worklist — closes the list of tasks assigned to the user.

Figure 4-11 Users Menu Options



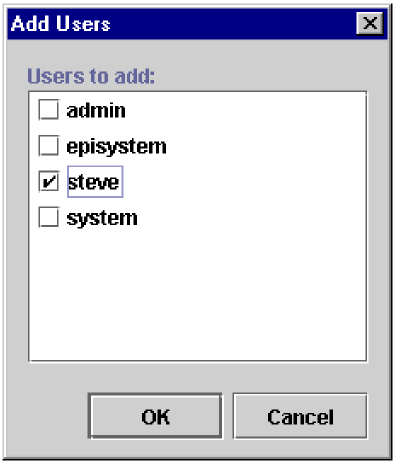
Adding a User to the Current Organization

The Add User facility adds a user that is already defined in the WebLogic Server security realm to the current WebLogic Process Integrator organization. This facility is accessible only if WebLogic Process Integrator Studio is operating within a manageable realm.

To add a user to the current WebLogic Process Integrator organization:

1. Right-click the Users folder, and select Add Users to display the Add Users dialog box.

Figure 4-12 Add Users Dialog Box

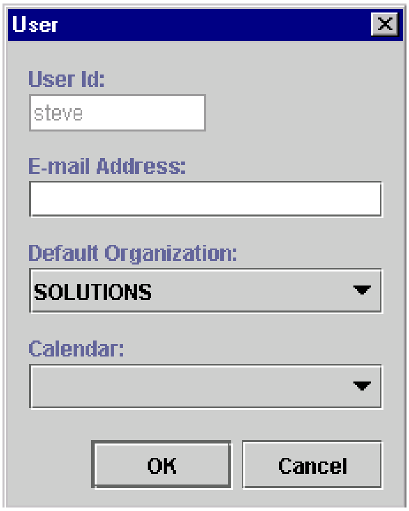


Users to add:

- □ admin
- □ episystem
- ☑ steve
- □ system

2. A list of valid users is displayed. (Valid users are those users registered in WebLogic Server). Click the check box to the left of the user to add, and click OK.
3. The user is added to the folder tree. Right-click the user name, and choose Properties from the pop-up menu to display the User dialog box.

Figure 4-13 User Dialog Box



User Id: steve

E-mail Address:

Default Organization: SOLUTIONS

Calendar:

The User dialog box contains the following fields:

- User Id — uniquely defines the user. The user enters this ID to log in to BEA WebLogic Process Integrator applications. In the User dialog box, this field is already populated with the name of the user selected in Step 2.
- E-mail Address — enter the e-mail address of the user.
- Default Organization — assign one default organization to the user.
- Calendar — select the appropriate business calendar to assign to the user. For details about this feature, see "Administering Business Calendars" on page 4-28.

4. Click OK to save the new user information, or click Cancel to cancel the operation.

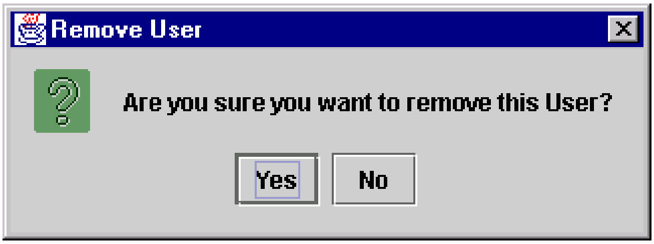
Removing a User from the Current Organization

The Remove User facility removes a user that is already defined in the WebLogic Server security realm from the current organization. It does not remove the user from the security realm.

To remove a user from the current WebLogic Process Integrator organization:

1. Right-click a user and choose Remove to display the Remove User dialog box.

Figure 4-14 Remove User Dialog Box



Are you sure you want to remove this User?

2. To remove the user, click Yes. To cancel the delete, click No.

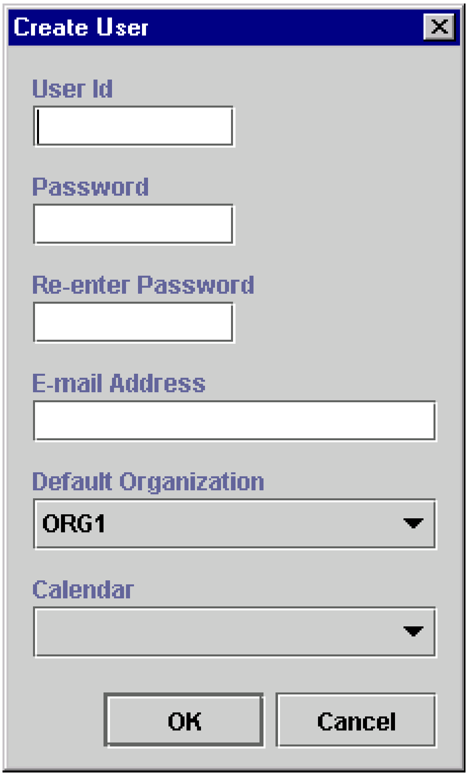
Creating a User

The Create User option adds a user to a WebLogic Server security realm, and is available only if the security realm is a manageable one.

To create a user in the security realm and WebLogic Process Integrator database:

1. Right-click the Users folder, and select Create Users to display the Create User dialog box.

Figure 4-15 Create User Dialog Box



Create User

User Id

Password

Re-enter Password

E-mail Address

Default Organization

ORG1

Calendar

2. Enter values in the following fields, and then click OK.

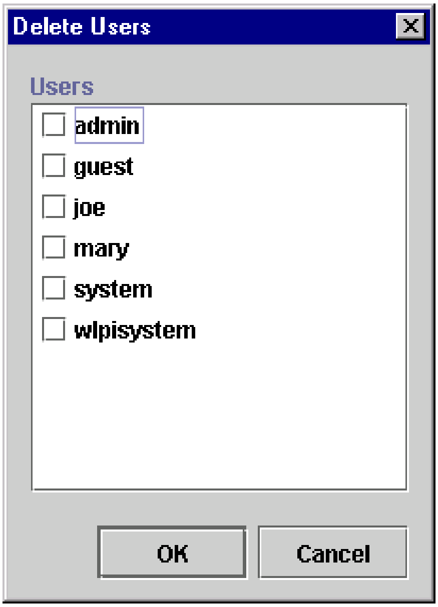
    - User Id — uniquely defines the user. The user enters this ID to log in to BEA WebLogic Process Integrator client applications.
    - Password and Re-enter Password — enter a password for the user.
    - E-mail Address — optionally enter the e-mail address of the user.
    - Default Organization — select a default organization for the user.
    - Calendar — optionally select a business calendar for the user.

Deleting a User

To delete defined users from the WebLogic Server security realm and WebLogic Process Integrator database:

1. Right-click the Users folder, and select Delete Users to display the Delete Users dialog box, which contains a complete list of valid users for the current organization.

Figure 4-16 Delete Users Dialog Box



□ admin
□ guest
□ joe
□ mary
□ system
□ wlpisystem

2. Select the checkbox to the left of each user that you will delete, and click OK.

   A confirmation dialog box prompts you to confirm your selection.

3. Click Yes to delete the selected user(s).

Updating User Properties

To update user properties:

1. Right-click a user and choose Properties to display the User dialog box.
2. Make the necessary changes to the E-mail Address, Default Organization, and Calendar.
3. Click OK to save the update the user information, or click Cancel to cancel the operation.

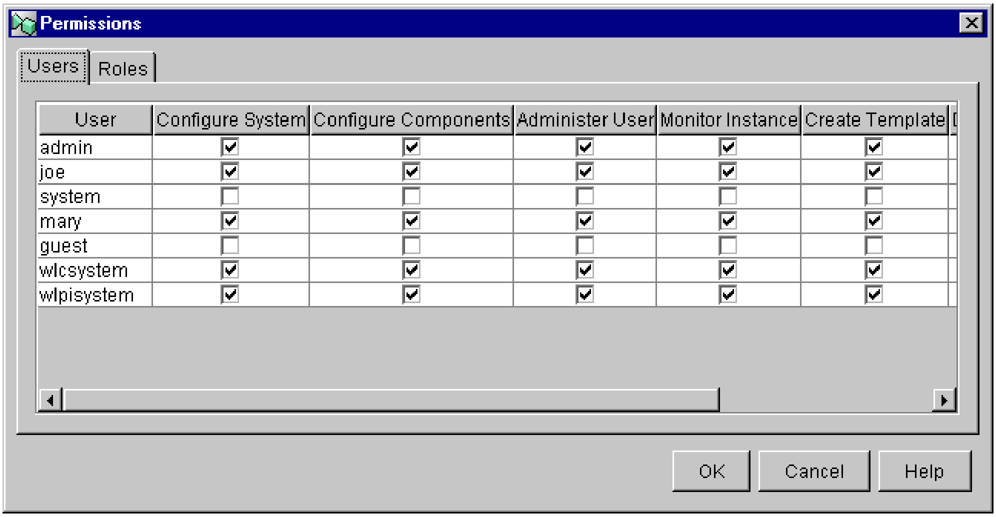
Setting Permissions for Users

A user inherits the levels of permission for the role to which it belongs. You can add other levels of permission to a user that are not defined for the role to which it belongs, but you cannot remove the permissions a user inherits from the role to which it belongs. Any permissions you add to a user are specific to the user, and are not reflected in the role to which the user belongs.

To set levels of permission for a user:

1. From the main Studio window, choose Configuration→Permissions. The Permissions dialog box is displayed.
2. Select the Users tab (if it is not already selected).

Figure 4-17 Permissions Dialog Box



User	Configure System	Configure Components	Administer User	Monitor Instance	Create Template
admin	☑	☑	☑	☑	☑
joe	☑	☑	☑	☑	☑
system	□	□	□	□	□
mary	☑	☑	☑	☑	☑
guest	□	□	□	□	□
wlcsystem	☑	☑	☑	☑	☑
wlpisystem	☑	☑	☑	☑	☑

The Permissions dialog box displays all the currently defined users, and the permissions currently assigned. If a permission is checked, but grayed out, you cannot remove that permission because it was inherited from the role to which the user belongs.

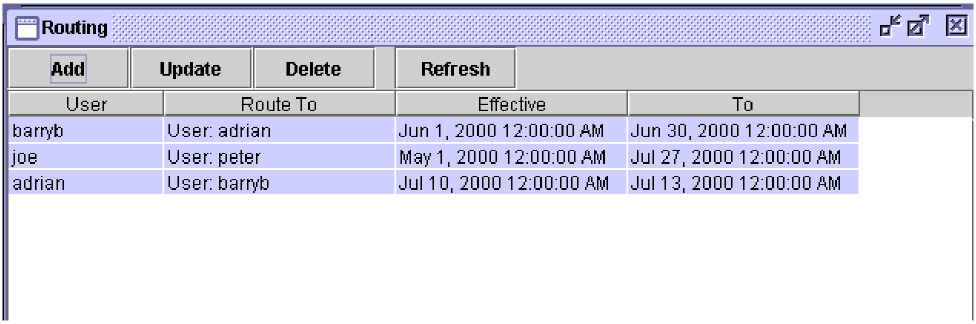
3. Select or clear the check boxes as desired.
4. Click OK to accept the changes, or Cancel to cancel the operation.

Administering Task Routing

Use the Routing feature to reroute currently assigned tasks from one user to another user or role for a specified, albeit temporary, period of time.

Right-click Routing in the folder tree to display a menu containing two options: Open and Close. Choose Open to display the Routing dialog box. The Routing dialog box contains a list of defined routings, showing users whose tasks were rerouted and the users or roles to whom the tasks were rerouted. The Effective and To dates represent the time period for which the tasks are rerouted.

Figure 4-18 Routing Dialog Box



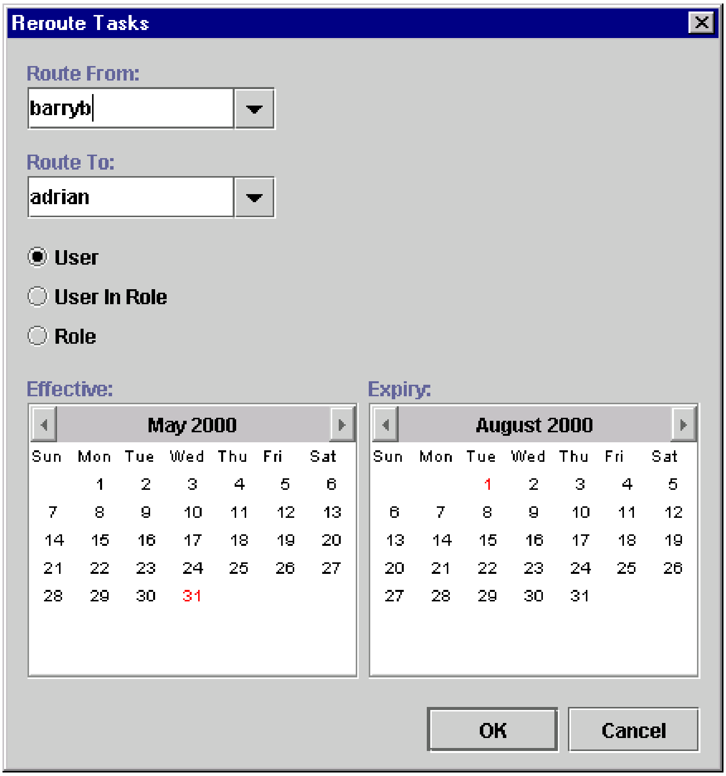
User	Route To	Effective	To
barryb	User: adrian	Jun 1, 2000 12:00:00 AM	Jun 30, 2000 12:00:00 AM
joe	User: peter	May 1, 2000 12:00:00 AM	Jul 27, 2000 12:00:00 AM
adrian	User: barryb	Jul 10, 2000 12:00:00 AM	Jul 13, 2000 12:00:00 AM

Adding a Routing Specification

To reroute a task:

1. In the Routing dialog box, click Add to display the Reroute Tasks dialog box.

Figure 4-19 Reroute Tasks Dialog Box



Reroute Tasks

Route From: barryb

Route To: adrian

● User

○ User In Role

○ Role

Effective:

May 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Expiry:

August 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

2. From the Route From drop-down list, select the user from which all tasks should be rerouted.

3. From the Route To drop-down list, select the User, User In Role, or Role to which all tasks should be rerouted, and select the corresponding radio button.

   Note: When you assign to a User in Role, the system performs workload balancing by first reviewing the number of tasks assigned to all users in the role, selecting the user with the least number of assigned tasks, and assigning all of the rerouted tasks to this user.

4. Specify an Effective date by selecting a month and year, and clicking on a day in the month display.

5. In a similar fashion, specify the Expiry date for the task rerouting.
6. Click OK to save the rerouting specification. Click Cancel to cancel the operation.

Updating a Task Routing Specification

To update a task rerouting specification:

1. Right-click the Routing feature in the folder tree, and choose Open from the pop-up menu to display the Routing dialog box.
2. In the list of displayed task reroutings, locate and highlight the task rerouting you want to update.
3. Click Update to display the Reroute Tasks dialog box.
4. Make changes as needed to the Route To, Effective, and Expiry values.
5. Click OK to save the changes. Click Cancel to cancel the operation.

Deleting a Task Routing Specification

To delete a task rerouting specification:

1. Right-click the Routing feature in the folder tree, and choose Open from the pop-up menu to display the Routing dialog box.
2. In the list of displayed task routings, locate and highlight the task routing you wish to delete.
3. Click Delete to display the Delete Reroute dialog box.
4. To delete the task rerouting, click Yes. To cancel the delete, click No.

Refreshing the Task List

Click Refresh in the Routing dialog box to refresh the rerouting task list and display any changes that have been made since you first invoked the Routing dialog box.

Administering Business Calendars

The BEA WebLogic Process Integrator business calendar feature defines the operating hours for the current organization. Business calendars make possible business time-related calculations, such as "Set a task’s due date to three business days from today."

Calendars can be associated with the following, as described in the sections indicated:

- Organizations (See “Adding an Organization” on page 4-6)
- Users (See “Adding an Organization” on page 4-6)
- Roles (See “Creating a Role” on page 4-10)
- Time-related actions:
  - Set Task Due Date action (See “Set Task Due Date” on page 7-10)
  - Timed Event action (See “Timed Event” on page 7-68)

You can define and assign the same business calendar to organizations, users, roles, and actions. You can also assign different business calendars to users, roles, and actions within the same organization.

Calendar assignment within BEA WebLogic Process Integrator is hierarchical in nature. The hierarchy places time-related actions at the lowest level, followed by roles and users, and finally, organizations at the highest level. If a time-related action is not assigned a calendar, it will, by default, be assigned the calendar of the user or role to which it is assigned. If a user or role is not assigned a calendar, it will, by default, be assigned the business calendar (organization level). In other words, calendar assignment is made at the most detailed component level.

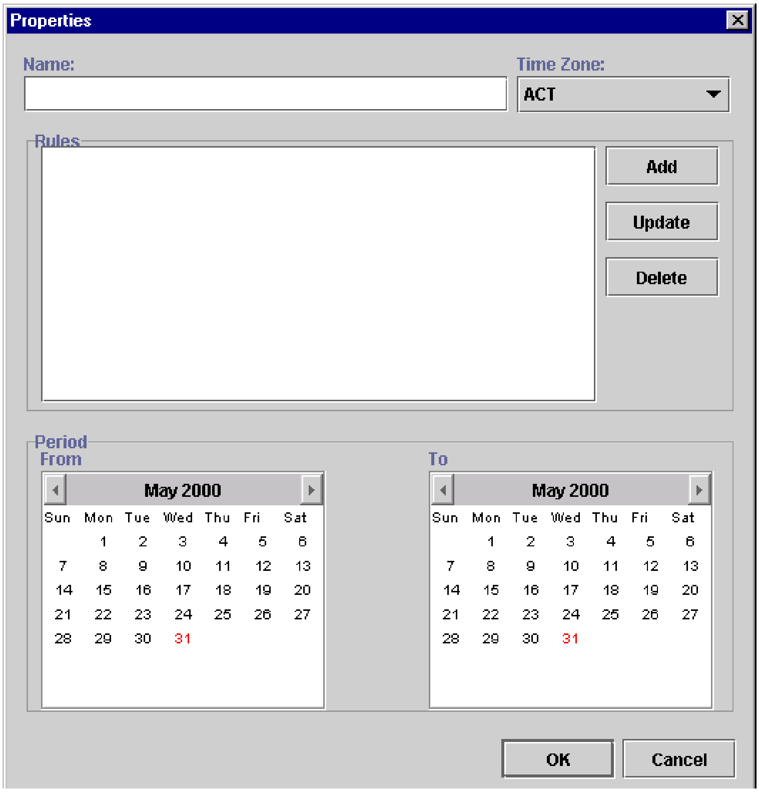
Business calendars are rule-based. The calendar facility leads you through the definition of each rule, much as the BEA WebLogic Process Integrator Expression Builder allows you to select from lists of functions, operators, variables, and so on in order to create an expression.

Creating a Calendar

To create a new calendar:

1. Right-click Calendars in the folder tree, and choose Create Calendar to display the Calendar Properties dialog box.

Figure 4-20 Calendar Properties Dialog Box



Name:

Time Zone: ACT

From

May 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

To

May 2000

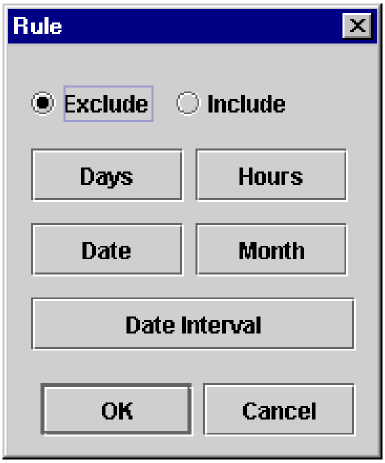
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

2. Enter a meaningful name for the calendar in the Name field.

3. Select a time zone for the calendar from the Time Zone drop-down list.

4. To add a rule, click Add. This displays the Rule dialog box.

Figure 4-21 Rule Dialog Box



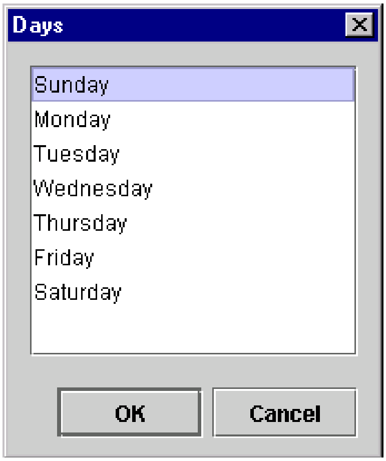
● Exclude ○ Include

5. Select either Exclude or Include. These buttons determine the method by which you will define your calendar rules (by exclusion or by inclusion). It is recommended that you use one or the other method and not both throughout your rule defining process.

6. Define calendar rules using the following buttons:

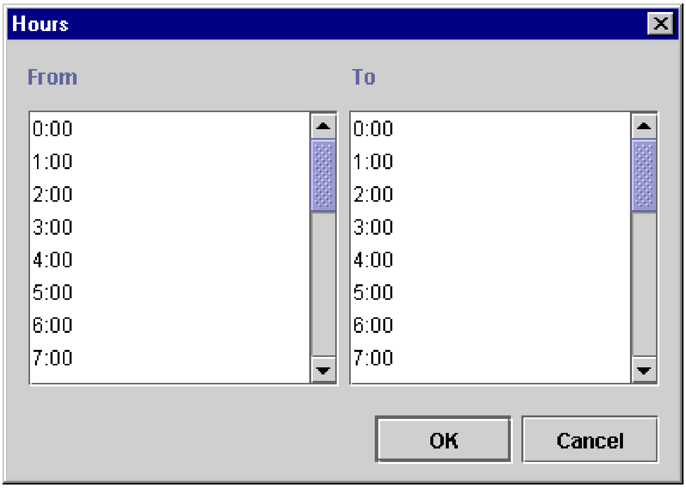
   - Days — displays the Days dialog box. Excludes or includes certain calendar days. (Hold the CTRL key down to select more than one day of the week from the Days dialog box.)

Figure 4-22 Days Dialog Box



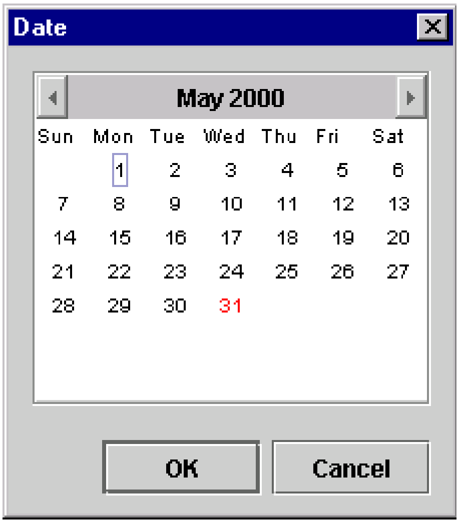
- Hours — displays the Hours dialog box. Excludes or includes certain hours of the day. Select a From and To value.

Figure 4-23 Hours Dialog Box



- Date — displays the Date dialog box. Excludes or includes a certain date.

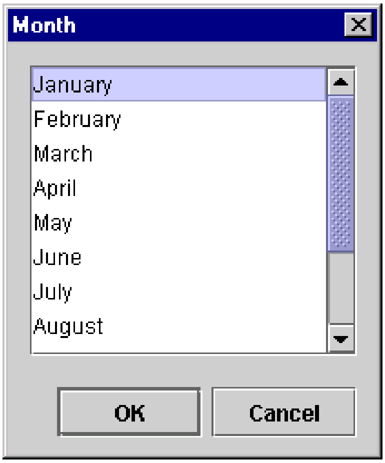
Figure 4-24 Date Dialog Box



Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

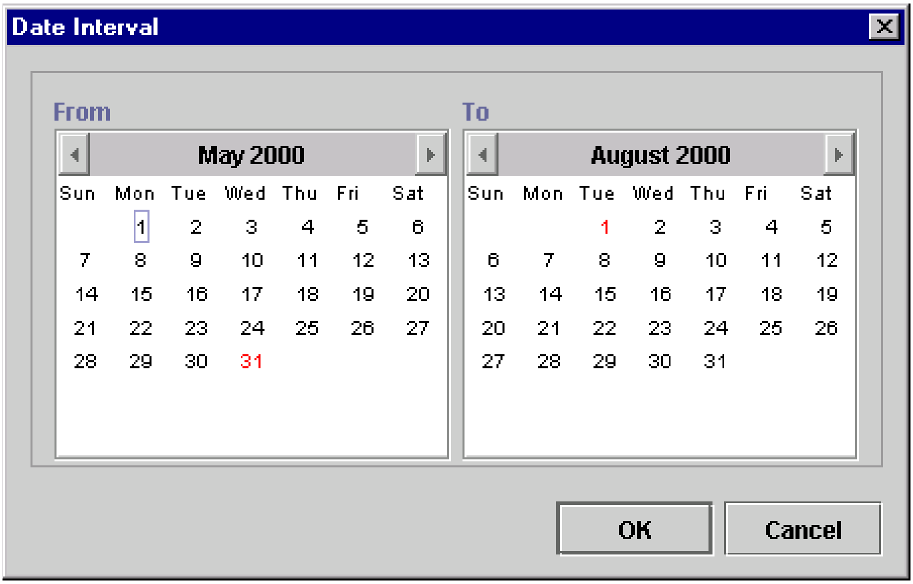
- Month — displays the Month dialog box. Excludes or includes certain month(s) of the year. (Hold the CTRL key down while clicking with your mouse to select more than one month of the year from the Month dialog box.)

Figure 4-25 Month Dialog Box



- Date Interval — displays the Date Interval dialog box. Enables you to specify a specific date interval to be excluded or included.

Figure 4-26 Date Interval Dialog Box



Date Interval

From

May 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

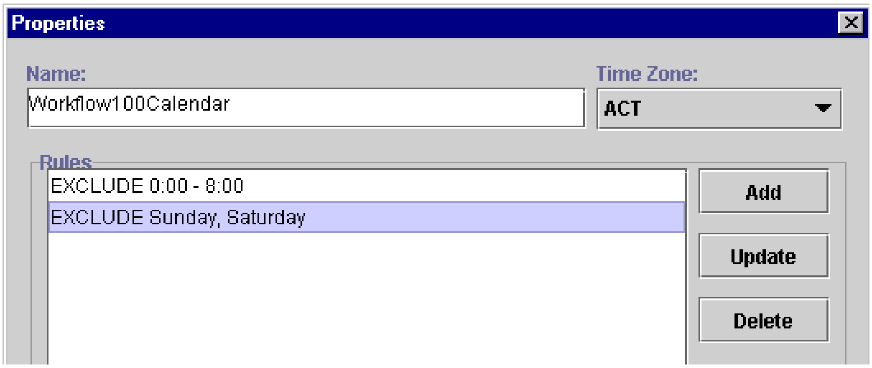
To

August 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

New calendar rules are displayed in the Rules area of the Calendar Rules dialog box.

Figure 4-27 Calendar Rules



Properties

Name: Workflow100Calendar

Time Zone: ACT

Rules

EXCLUDE 0:00 - 8:00

EXCLUDE Sunday, Saturday

Note: You can update or delete a calendar rule by highlighting the rule in the Rules portion of the Calendar Properties dialog box, clicking Update or Delete, and following the prompts.

7. Once you define your calendar rules (using either the Exclusion or Inclusion method), use the Period From: and To: calendars to specify the period for which this calendar will be in effect.
8. Click OK to save your changes, or click Cancel to terminate the operation.

Updating a Calendar

To update an existing calendar:

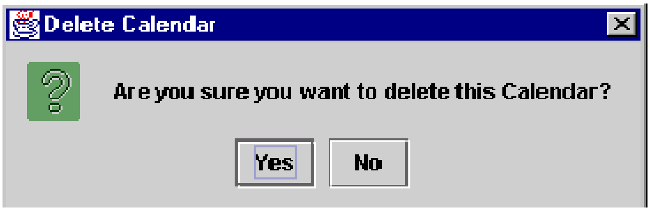
1. In the folder tree, right-click the calendar you want to update, and choose Properties.
2. Make the necessary changes to the business days and hours.
3. Click OK to save the changes. Click Cancel to cancel the operation.

Deleting a Calendar

To delete an existing calendar:

1. Right-click the calendar you want to delete and choose Delete to display the Delete Calendar dialog box.

Figure 4-28 Delete Calendar Dialog Box


Delete Calendar

Are you sure you want to delete this Calendar?

2. To delete the calendar, click Yes. To cancel the delete, click No.

Understanding Business Operations

The Business Operations facility enables you to display all Enterprise JavaBeans (EJBs) and Java classes that are registered in WebLogic Server, as well as their remote methods and parameters. Using the Business Operations facility, you can create customized actions that invoke existing applications or applications that are built specifically for the workflow.

EJBs and Java classes are software components that a workflow invokes to perform a business activity, such as checking the account balance of a customer. The workflow invokes the software component by calling one of its methods to perform the activity. In BEA WebLogic Process Integrator, a business operation represents the method call. The business operation connects the workflow to the software components that automate the business process.

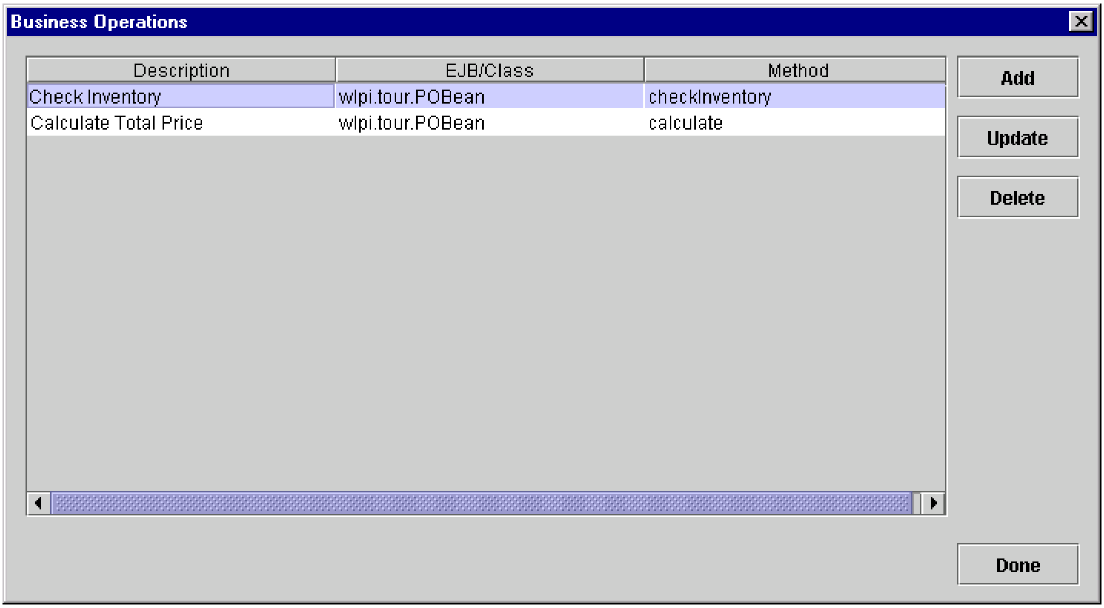
You use the Business Operations facility to define the business operation, and the Perform Business Operation action to perform the method call. You can assign the results of the method call to a workflow variable.

Ideally, technical personnel with detailed knowledge about the software components would define Business Operations. Workflow designers would then use the Business Operations in the appropriate place in the workflow to perform the business activity invoked by the Business Operation.

Using the Business Operations Dialog Box

In the BEA WebLogic Process Integrator Studio main window, choose Business Operations from the Configuration menu. This displays the Business Operations dialog box.

Figure 4-29 Business Operations Dialog Box



Business Operations

Description	EJB/Class	Method
Check Inventory	wlpi.tour.POBean	checkInventory
Calculate Total Price	wlpi.tour.POBean	calculate

The Business Operation dialog box (accessed from the BEA WebLogic Process Integrator Studio Configuration menu) contains the following:

- Description — description assigned to the business operation. For more information, see “Defining a Business Operation” on page 4-37.
- EJB/Class — Enterprise JavaBean (EJB) or Java class to be invoked.
- Method — method in the EJB or Java class to be called.
- Add — invokes the Define Business Operation dialog box, where you define business operations.

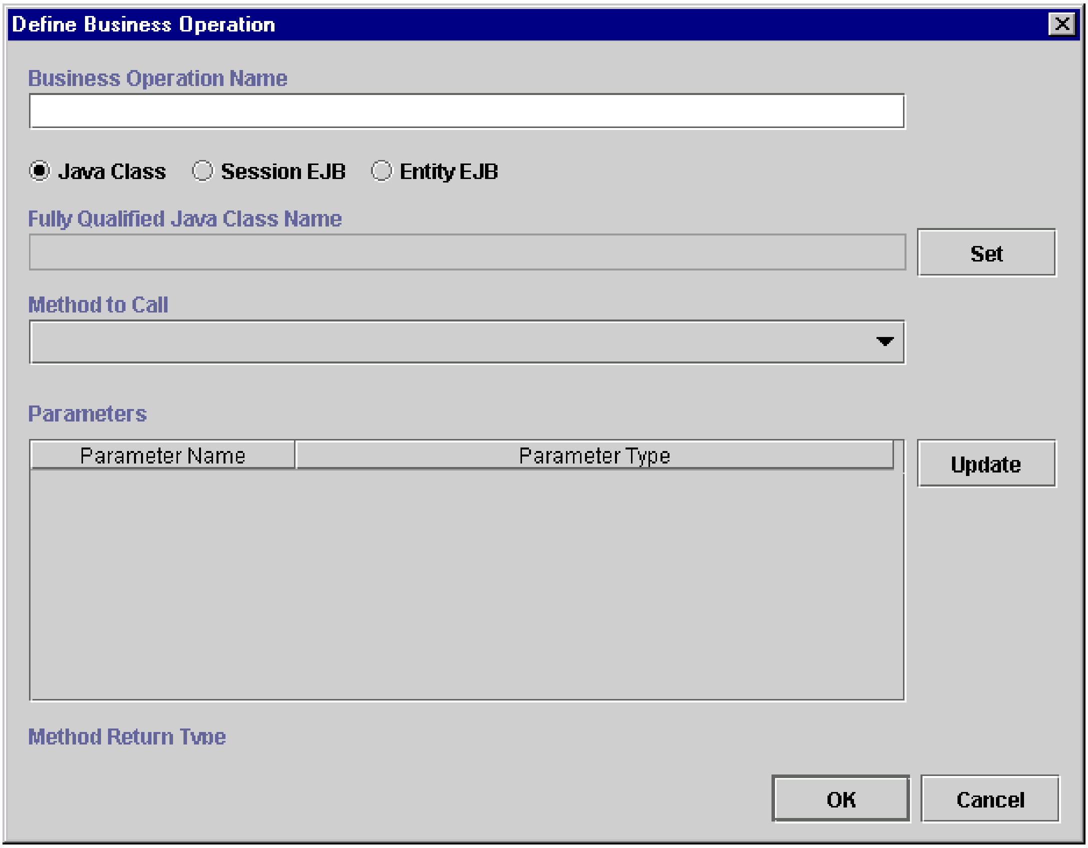
- Update — updates the highlighted business operation; invoke the Define Business Operation dialog box and make changes as needed.
- Delete — deletes the highlighted business operation.

Defining a Business Operation

To define a BEA WebLogic Process Integrator business operation, proceed as follows:

1. From the Business Operations dialog box, click Add to display the Define Business Operation dialog box.

Figure 4-30 Define Business Operation Dialog Box



● Java Class ○ Session EJB ○ Entity EJB

2. A business operation represents a method call on an EJB or Java class instance. In the Name field, enter a description of the business operation that is meaningful to a business analyst. A business operation that returns the number of items available in stock might be called, for example, `CheckInventory`.

3. Specify the software component the business operation is invoking. The choices are:

   - Java Class — the business operation calls a method in a Java class on WebLogic Server. Only Java classes that implement the serializable interface can be used with business operations.

   - Session EJB — the business operation calls a method in a session EJB on WebLogic Server. Session EJBs represent transient data, and persists only for the duration of a transaction.

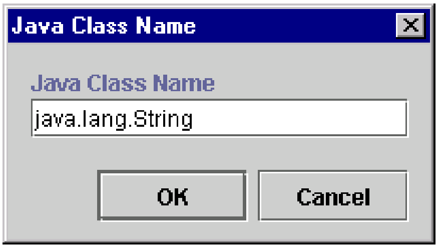
   - Entity EJB — the business operation calls a method in an entity EJB on WebLogic Server. Entity EJBs represent persistent data, and persist for at least the duration of the workflow instance and, often, beyond the life of a workflow instance.

Defining Business Operations for Java Classes

To define a business operation for a Java class, proceed as follows:

1. Select the Java Class radio button.
2. Click Set to display the Java Class Name dialog box.

Figure 4-31 Java Class Name Dialog Box

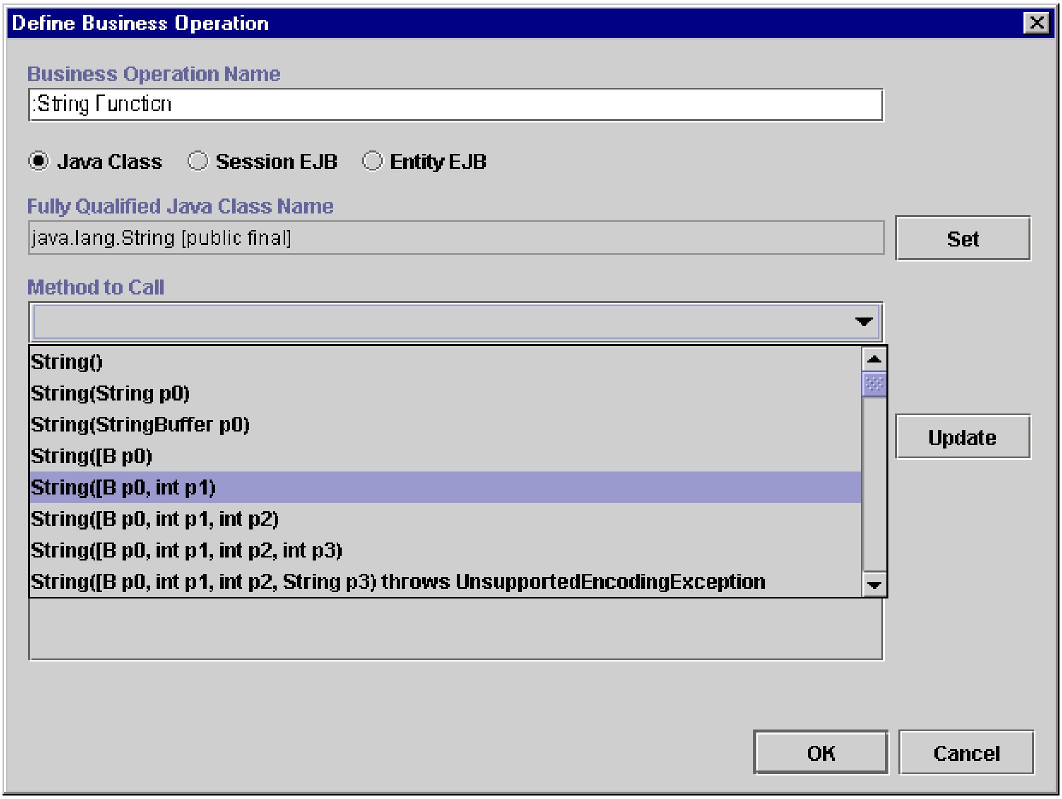


Java Class Name

Java Class Name: java.lang.String

3. Enter a fully qualified Java class name (for example, java.lang.String), and click OK. The Java class name is set in the Fully Qualified Java Class Name field of the Define Business Operation dialog box.
4. Use the Method to Call drop-down list to select a method to call upon invocation of the business operation in the workflow. There are three types of Java class methods in the list:
   - Constructor Type — these method types are first in the list and begin with the word `String`. Constructor methods return an object of the type you entered as the Java Class Name.
   - Method Type — these method types are second in the list. They are invoked on Java objects and can return any type.
   - Static Type — these are last in the list and begin with the word `Static`. Static methods require no object to be invoked and can return any type.

Figure 4-32 Method to Call Drop-down List



Define Business Operation

Business Operation Name

String Function

● Java Class ○ Session EJB ○ Entity EJB

Fully Qualified Java Class Name

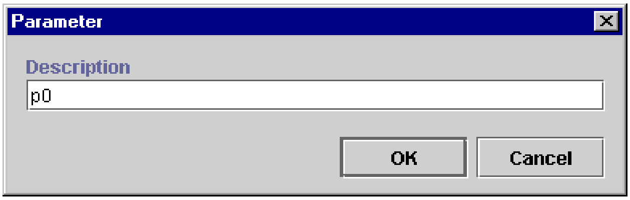
java.lang.String [public final]

Method to Call

String()
String(String p0)
String(StringBuffer p0)
String([B p0)
String([B p0, int p1)
String([B p0, int p1, int p2)
String([B p0, int p1, int p2, int p3)
String([B p0, int p1, int p2, String p3) throws UnsupportedEncodingException

5. The Parameter Name and Parameter Type fields are populated with the parameter values of the selected method. You can optionally rename the parameter and give it a more meaningful name by highlighting the parameter and clicking Update to display the Parameter dialog box.

Figure 4-33 Parameter Dialog Box


Parameter

Description

p0

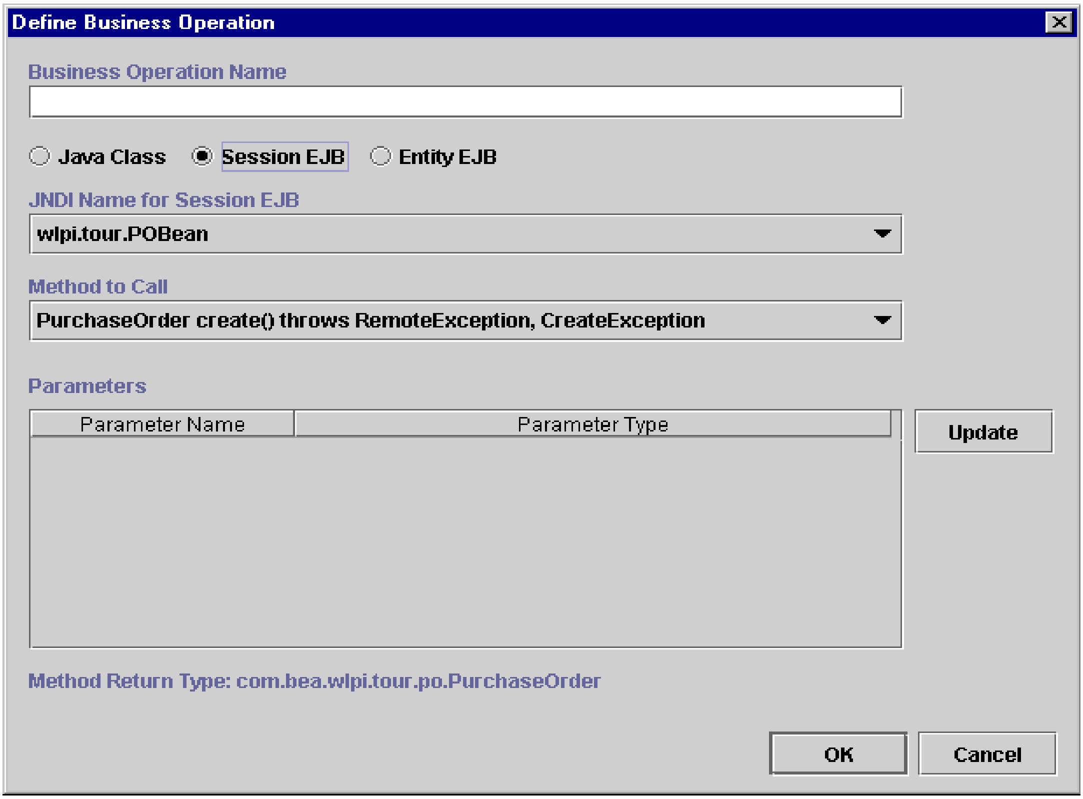
6. Change the parameter name in the Description field, and click OK.

Defining Business Operations for Session EJBs

To define a business operation for a Session EJB (Enterprise Java Bean), proceed as follows:

1. Select the Session EJB radio button to display fields relevant to Session EJBs in the Define Business Operation dialog box.

Figure 4-34 Define Business Operation: Session EJB



Define Business Operation

Business Operation Name

○ Java Class ● Session EJB ○ Entity EJB

JNDI Name for Session EJB: wlpi.tour.POBean

Method to Call: PurchaseOrder create() throws RemoteException, CreateException

Parameters

Parameter Name	Parameter Type

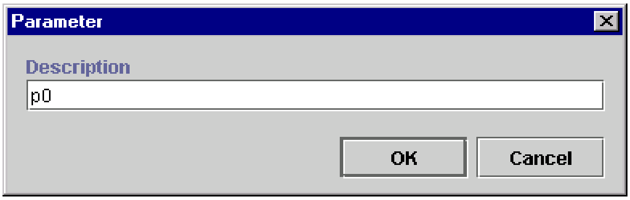
Method Return Type: com.bea.wlpi.tour.po.PurchaseOrder

2. From the JNDI Name for Session EJB drop-down list, select the JNDI (Java Naming and Directory Interface) name for the Session EJB. JNDI is a service provided by WebLogic Server; this service stores objects much like a database. Here, you are selecting the JNDI name for the Session EJB; a Session EJB is an object stored in this service.

3. The Method to Call drop-down list contains a list of methods (or operations) that you can perform on the Session EJB by calling it. Select the method to call from the drop-down list.

4. The Parameter Name and Parameter Type fields are populated with the parameter values of the selected method. You can optionally rename the parameter and give it a more meaningful name by highlighting the parameter and clicking the Update button to display the Parameter dialog box.

Figure 4-35 Parameter Dialog Box



Description: p0

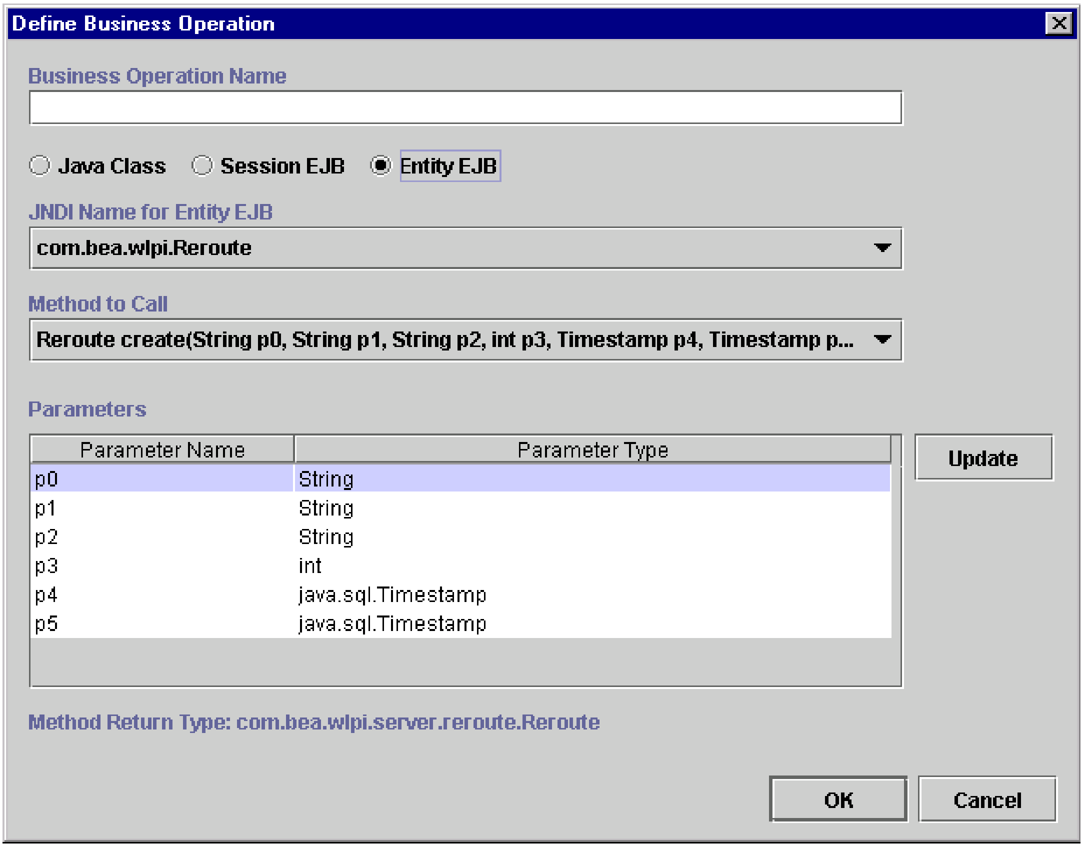
5. Change the parameter name in the Description field, and click OK.

Defining Business Operations for Entity EJBs

To define a business operation for an Entity EJB (Enterprise Java Bean), proceed as follows:

1. Select the Entity EJB radio button to display fields relevant to Session EJBs in the Define Business Operation dialog box.

Figure 4-36 Define Business Operation: Entity EJB



Define Business Operation

Business Operation Name

○ Java Class ○ Session EJB ● Entity EJB

JNDI Name for Entity EJB

com.bea.wlpi.Reroute

Method to Call

Reroute create(String p0, String p1, String p2, int p3, Timestamp p4, Timestamp p...

Parameters

Parameter Name	Parameter Type
p0	String
p1	String
p2	String
p3	int
p4	java.sql.Timestamp
p5	java.sql.Timestamp

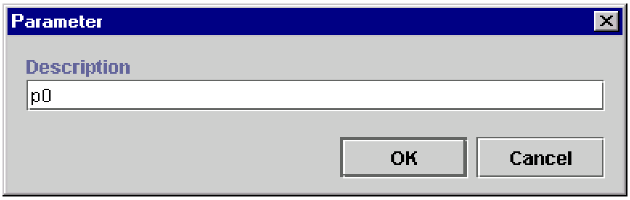
Method Return Type: com.bea.wlpi.server.reroute.Reroute

2. From the JNDI Name for Entity EJB drop-down list, select the JNDI (Java Naming and Directory Interface) name for the Entity EJB. JNDI is a service provided by WebLogic Server; this service stores objects much like a database. Here, you are selecting the JNDI name for the Entity EJB; an Entity EJB is an object stored in this service.

3. The Method to Call drop-down list contains a list of methods (or operations) that you can perform on the Entity EJB by calling it. Select the method to call from the drop-down list.

4. The Parameter Name and Parameter Type fields are populated with the parameter values of the selected method. You can optionally rename the parameter and give it a more meaningful name by highlighting the parameter and clicking Update to display the Parameter dialog box.

Figure 4-37 Parameter Dialog Box



Description
p0

5. Change the parameter name in the Description field, and click OK.

Once you have defined all of your business operations, click OK on the Define Business Operations dialog box to confirm your changes. The business operations are added to the list of valid business operations in the Business Operation dialog box.

You use the Perform Business Operation action to invoke a defined business operation. For more information, see “Perform Business Operation” on page 7-57.

CHAPTER

5 Defining and Maintaining Workflows

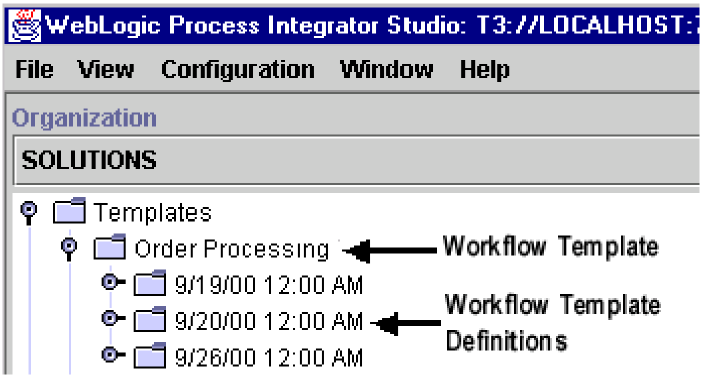
The following sections introduce you to BEA WebLogic Process Integrator workflow templates and workflow template definitions. It also explains how to perform a number of workflow template definition and maintenance tasks.

- Understanding Workflow Templates and Workflow Template Definitions
- Opening Workflow Template Definitions
- Saving and Closing Workflow Template Definitions
- Printing Workflow Template Definitions
- Copying Workflow Template Definitions
- Deleting Workflow Template Definitions
- Deleting Workflow Templates
- Understanding Effective and Expiry Dates
- Exporting and Importing Workflow Packages
- Exporting Workflow Template Definitions
- Importing Workflow Template Definitions

Understanding Workflow Templates and Workflow Template Definitions

A workflow template is, in essence, a folder or a container for BEA WebLogic Process Integrator workflow template definitions. Each workflow template can hold one or more BEA WebLogic Process Integrator workflow template definitions. BEA WebLogic Process Integrator workflow template definitions are identified in the folder tree by an Effective and Expiry date. For details, see "Understanding Effective and Expiry Dates" on page 5-23.

Figure 5-1 Workflow Templates and Workflow Template Definitions

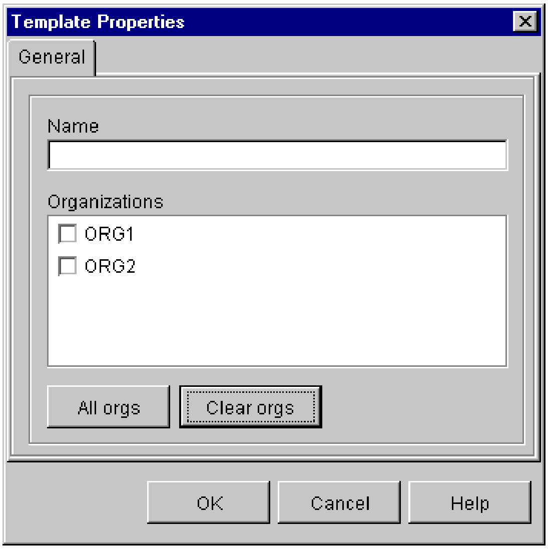

WebLogic Process Integrator Studio: T3://LOCALHOST:
File View Configuration Window Help
Organization
SOLUTIONS
Templates
Order Processing
9/19/00 12:00 AM
9/20/00 12:00 AM
9/26/00 12:00 AM
Workflow Template
Workflow Template Definitions

Creating a New Workflow Template

To create a new workflow template:

1. Right-click on the Templates folder in the BEA WebLogic Process Integrator main window.
2. Select Create Template from the pop-up menu to display the Template Properties dialog box.

Figure 5-2 Template Properties Dialog Box



Note: If a plug-in is defined for workflow templates, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. In the Name field, enter a meaningful name for the workflow template. All workflow template definitions defined within this workflow template use this name for identification, once they are instantiated (placed into run time) and become workflow instances. The name must be unique.
4. In the Organizations section of the dialog box, select the organization (or organizations) under which you will define this workflow template.

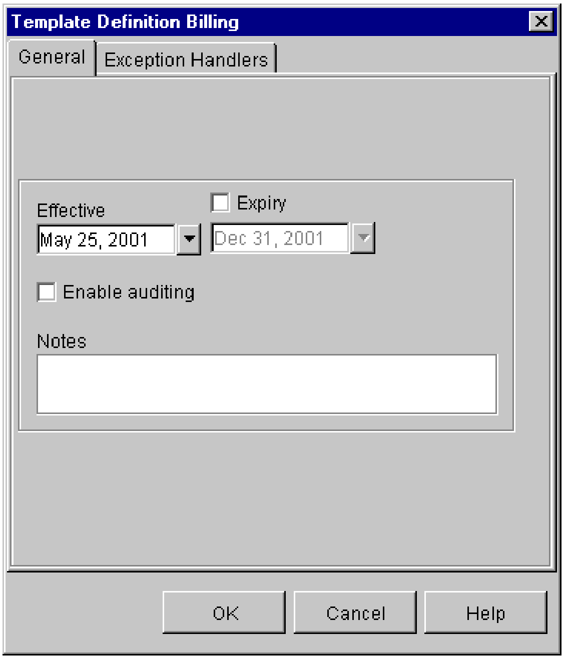
5. To make the workflow template available to all organizations, click All Orgs; to make the workflow template available to no organizations, click Clear Orgs.
6. Click OK. The new workflow template appears under Templates in the folder tree of the BEA WebLogic Process Integrator main window.

Creating a New Workflow Template Definition

To create a new workflow template definition, proceed as follows:

1. Right-click the workflow template that will contain the new workflow template definition in the folder tree.
2. Choose Create Template Definition from the pop-up menu to display the Template Definition dialog box.

Figure 5-3 Template Definition Dialog Box



Effective: May 25, 2001

□ Expiry: Dec 31, 2001

□ Enable auditing

Notes

3. In the Template Definition dialog box, specify the following:

- Effective — select the exact date and time when this workflow template definition will become effective.
- Expiry — optionally select this checkbox to set an expiry date for the workflow template definition. For more information, see "Understanding Effective and Expiry Dates" on page 5-23.

Note: If the Expiry option is not selected, the workflow template definition will always be valid and effective.

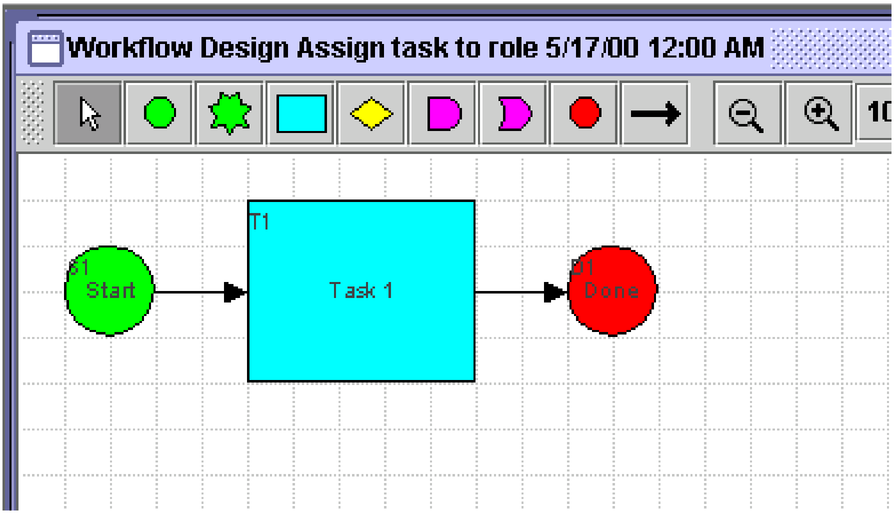
- Enable auditing — allows you post an XML message with audit and error information to the WebLogic Process Integrator JMS topic: wlpi.AuditTopic. The default JMS topic for audit entries is wlpi.AuditTopic. You can change the default topic in the config.xml file.
- Notes — free-form optional field where general comments describing the workflow template definition can be entered.

4. Click OK to save changes and display the workflow drawing area.

In the workflow drawing area, a default workflow template definition is presented along with a toolbar containing drawing shapes used for defining the workflow template. The default workflow template definition contains three shapes: Start, Task, and Done. Modify and define properties for the default workflow template definition as needed.

More than one workflow template definition diagram can be opened simultaneously, and you can toggle between these within BEA WebLogic Process Integrator.

Figure 5-4 Workflow Drawing Area



Specifying Workflow Template Definition Properties

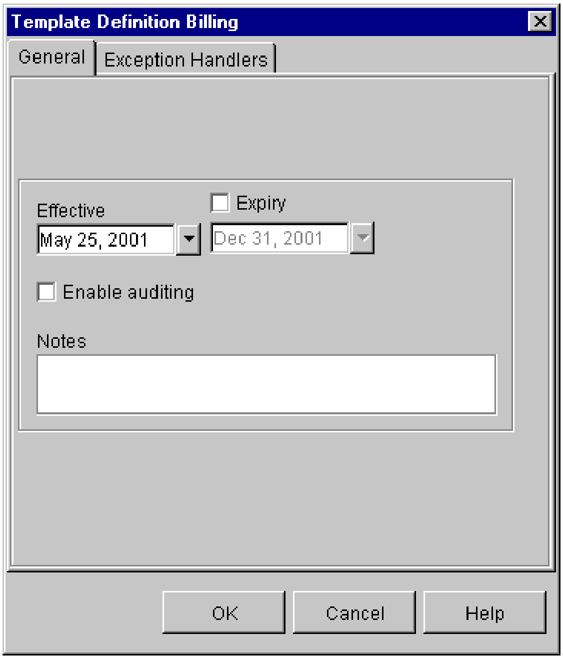
To specify the properties of a workflow template definition:

1. Right-click anywhere in the workflow drawing area, and choose Properties from the pop-up menu to display the Template Definition dialog box. Alternately, display the Template Definition dialog box by right-clicking the workflow template definition in the folder tree and selecting Properties from the pop-up menu.

   The Template Definition dialog box contains two tabs:

   - General — changes general workflow template definition values as needed. For details, see “Creating a New Workflow Template Definition” on page 5-4.
   - Exception Handlers — assigns system and custom-defined workflow exception handlers to the workflow template definition. For details, see Chapter 7, “Defining Actions.”

Figure 5-5 Template Definition Dialog Box: General Tab



Template Definition Billing

Effective: May 25, 2001

□ Expiry: Dec 31, 2001

□ Enable auditing

Notes:

Note: If a plug-in is defined for workflow template definitions, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

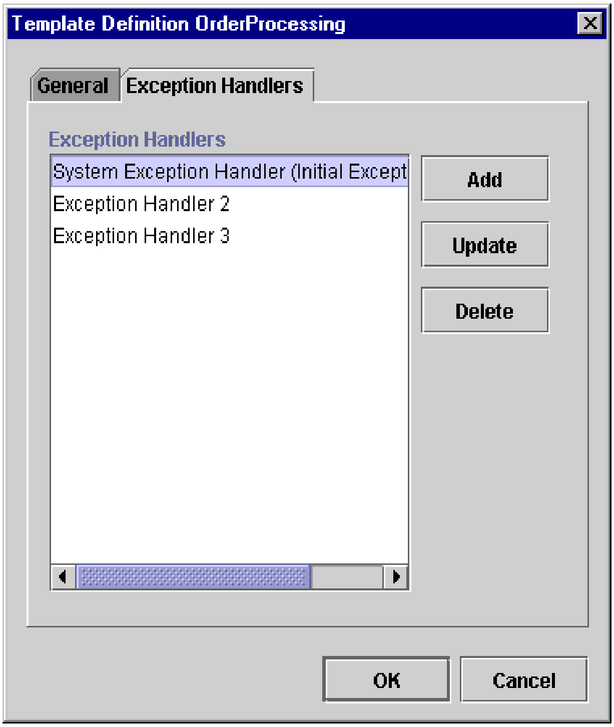
2. In the General tab of the Template Definition dialog box, specify the following:

- ID — expression specifying the label the system will generate at run time to distinguish instances of the same workflow template definition. (Workflow instances are created when a workflow template definition is placed into run time. There can be multiple instances of the same workflow template definition.) This Id is used to tag both the workflow template definition instance and tasks within the workflow template definition, giving the user contextual information at run time. For example, an instance of a workflow template definition can include date and time, invoice number, customer name, and other relevant information that differentiates it from others.

Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

- Active — if selected, indicates that the workflow template definition is active and available for processing. A new workflow template definition defaults to the inactive status and must be activated before it can be processed. Often workflow template definitions that are currently in development are inactive to prevent premature invocation. It is important to note that inactive workflow template definitions cannot be instantiated (placed into run time). Also, imported workflow template definitions are marked inactive and must be marked active in order for them to be processed.
- Effective — exact date and time when the workflow template definition will become effective and can be instantiated.
- Expiry — optional selection to set an expiry date for the workflow template definition using the available drop-down list.
- Enable audit to JMS (Java Message Service) topic — allows the workflow template definition analyst to send audit messages to any external program that may be listening on a particular JMS topic; the external program can then do what it needs to do with the audit information.
- Notes — free-form optional field where general comments describing the workflow template definition can be entered.

Figure 5-6 Template Definition Dialog Box: Exception Handlers Tab



Note: If a plug-in is defined for workflow template definitions, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. In the Exception Handlers tab of the Template Definition dialog box, select a system or custom-defined exception handler to assign to the workflow template definition.

4. Click OK to save changes. If you click Cancel, any pending changes to the fields in the Template Definition dialog box are discarded.

5. Use the buttons situated to the right of the Exception Handlers tab as follows:

   - Add — access the Exception Handler Properties dialog box to create a new exception handler.

- Update — highlight an exception handler in the list and access the Exception Handler Properties dialog box to update the selected exception handler.
- Delete — highlight an exception handler in the list, and click Delete to delete the exception handler. A confirmation window is displayed. Click OK to confirm the delete.

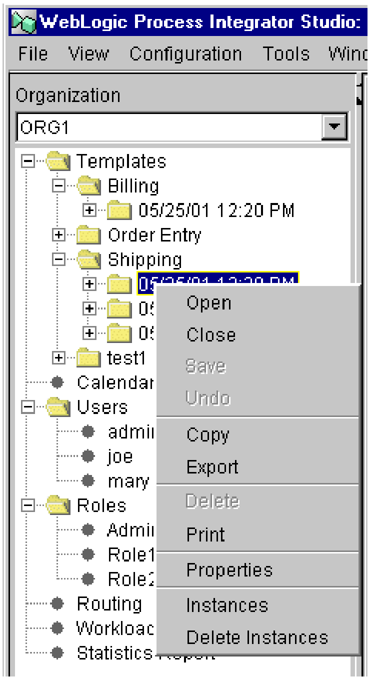
Note: For detailed instructions on using the Exception Handler Properties dialog box, see Chapter 9, “Handling Workflow Exceptions.”

Opening Workflow Template Definitions

To open an existing workflow template definition:

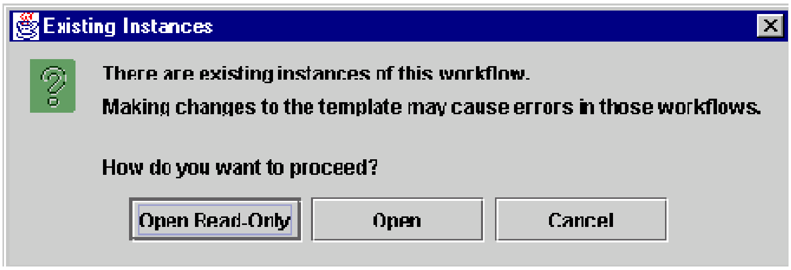
1. Right-click the workflow template definition in the folder tree, and choose Open from the pop-up menu.

Figure 5-7 Workflow Template Definition Menu



The Existing Instances dialog box is displayed, if there are existing instances of the selected workflow template definition, meaning that the selected workflow template definition has been started and an instance of the template definition exists in the server. Multiple instances of a single workflow template definition can exist.

Figure 5-8 Existing Instances Dialog Box



There are existing instances of this workflow.
Making changes to the template may cause errors in those workflows.

How do you want to proceed?

This message warns that changing a workflow template definition might cause a problem with the workflow template definitions currently in process.

2. Select from the following options:

- Open Read-Only — opens the selected workflow template definition for viewing purposes only.

Note: Opening a workflow template definition in read-only mode prevents you from modifying a workflow. In fact all user interface options relating to making changes to workflows are not accessible. For example, you cannot access the OK button on a given Properties dialog box, shapes are browned out and cannot be added to the workflow drawing area, and so forth.

- Open — opens the selected workflow template definition for view and update.
- Cancel — closes this dialog box.

Saving and Closing Workflow Template Definitions

When you make changes within BEA WebLogic Process Integrator Studio, you must save them. If you make changes to a workflow template definition and attempt to close the workflow template definition in the drawing area without saving it, you will be prompted to save.

In the BEA WebLogic Process Integrator Studio folder tree, an asterisk (*) appears to the left of each workflow template definition that needs to be saved. From the folder tree, you save by right-clicking the workflow template definition and choosing Save from the pop-up menu. From the workflow drawing area, you save by right-clicking in the drawing area and choosing Save.

A workflow template definition cannot be modified unless it is opened. When a definition is not open, you cannot modify its values. Once you open a definition, it is locked and cannot be opened by another user. When you close a definition, it is unlocked and available for use by another user.

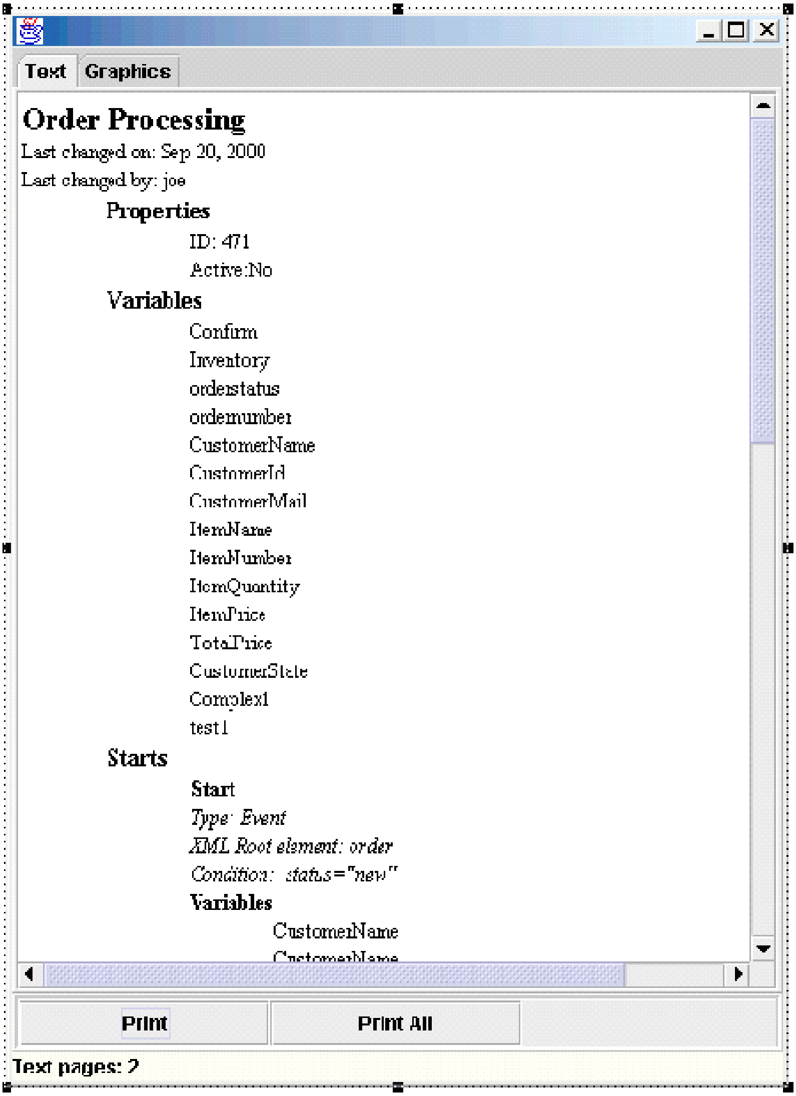
Printing Workflow Template Definitions

You can print workflow template definition diagrams from WebLogic Process Integrator. There are two methods for invoking the print facility.

- Right-click the workflow template definition in the folder tree of the Studio and select Print from the pop-up menu.
- Right-click anywhere in the workflow template definition diagram in the drawing area and select Print from the pop-up menu.

The Text tab contains information relevant to each workflow node, including action and node notes, as well as details on actions within tasks, events, decisions and done nodes, including subactions within actions.

Figure 5-9 Print Workflow Text

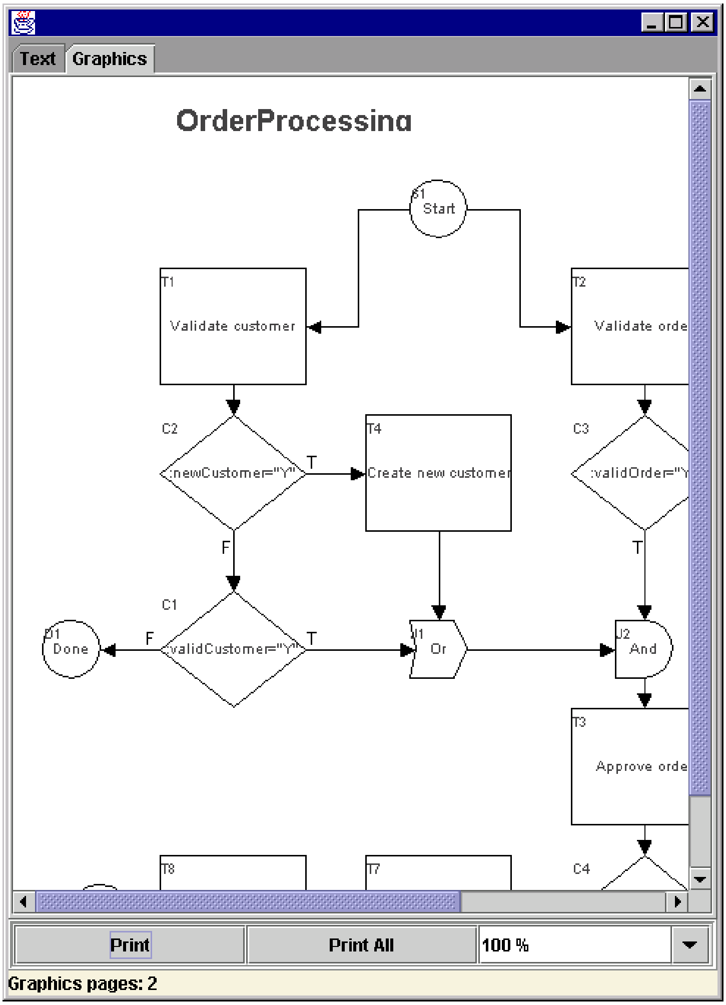


```
Order Processing
Last changed on: Sep 20, 2000
Last changed by: joe
          Properties
                    ID: 471
                    Active:No
          Variables
                    Confirm
                    Inventory
                    orderstatus
                    ordernumber
                    CustomerName
                    CustomerId
                    CustomerMail
                    ItemName
                    ItemNumber
                    ItemQuantity
                    ItemPrice
                    TotalPrice
                    CustomerState
                    Complex1
                    test1
          Starts
                    Start
                    Type: Event
                    XML Root element: order
                    Condition:  status="new"
                    Variables
                              CustomerName
                              CustomerName
```

Text pages: 2

The Graphics tab contains a diagram of the workflow template definition.

Figure 5-10 Print Workflow Graphics



100 %

Graphics pages: 2

Click Print to print the information on the selected tab, or click Print All to print both the information contained in the Text tab and the diagram contained in the Graphics tab.

In the Print dialog box, select the appropriate settings to print the workflow.

Copying Workflow Template Definitions

BEA WebLogic Process Integrator allows you to take an existing workflow template definition, copy it, and paste it as a new workflow template definition into the folder tree. When this is done, all of the assigned workflow template definition properties are copied as well. This saves you time when you must create more than one workflow template definition with similar properties. The properties of the new workflow template definition can be modified as needed.

In addition, BEA WebLogic Process Integrator allows you to copy nodes (shapes) within workflow template definitions and paste them into the current workflow template definition or other open workflow template definitions. All properties assigned to nodes are copied as well.

Note: The copied (new) workflow template definition will not be marked active. You must mark it active in the workflow Template Definition Properties dialog box in order for it to be processed in the workflow.

Copying Entire Workflow Template Definitions

To create a new workflow template definition based on an existing workflow template definition and its assigned properties, in the folder tree, right-click the workflow template definition to be copied, and choose Copy from the pop-up menu. A copy of the selected workflow template definition is instantly pasted as the last workflow template definition in the folder tree, and the new workflow template definition is opened and displayed in the drawing area. The new workflow template definition contains the same effective and expiry dates as the original workflow template definition. See “Understanding Effective and Expiry Dates” on page 5-23 for more information on this topic.

To rename the new workflow template definition, right-click anywhere in the drawing area, choose Properties from the pop-up menu to access the Template Definition dialog box, and make changes as needed. Modifying the Expiry and Effective dates changes the name of the workflow template definition in the folder tree and reflects the new dates.

Copying Nodes Within Workflow Template Definitions

You can copy a shape representing a node within a workflow template definition and paste it into the current workflow template definition or into another open workflow template definition. Any properties that have been defined within the node are also copied. To do so, proceed as follows:

1. Click the node that you will copy.
2. Right-click in the drawing area, and select Copy from the pop-up menu.
3. Place your cursor in either the current workflow template definition or another open workflow template definition, depending on where you want to paste the shape, and right-click.
4. From the pop-up menu, select Paste.

The shape is pasted in the indicated workflow template definition. Modify the properties of the node as needed.

Copying Actions Within or Between Workflow Node Shapes

BEA WebLogic Process Integrator also allows you to copy actions within or between shapes representing a node within a workflow template definition. Any properties that have been defined within the action are also copied. The ability to copy actions saves you time.

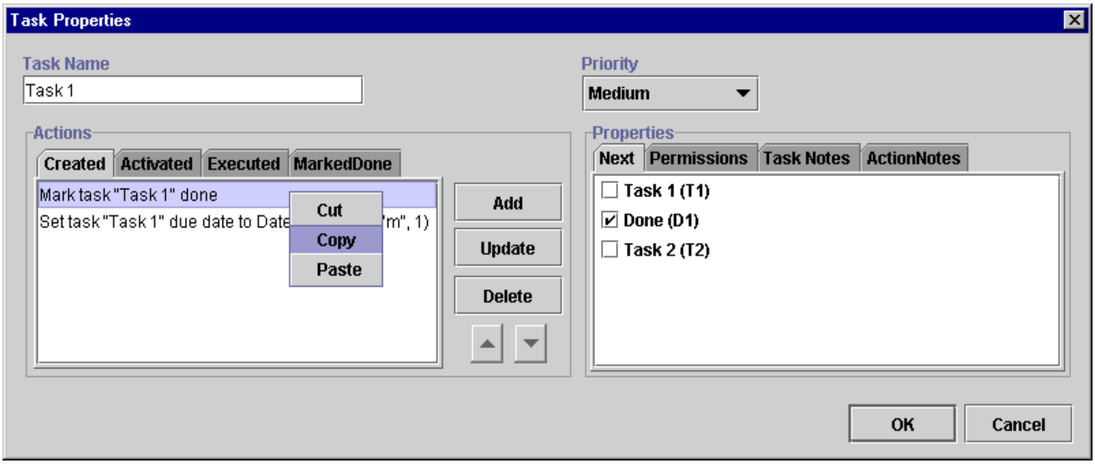
Copying Actions Within Workflow Nodes

The ability to copy actions within workflow nodes is useful when, for example, you must define a new action within a workflow task node that will contain properties similar to an already defined action within that task node. By copying the existing action and pasting it as a new action within the same task node, you save yourself time. Rather than having to define a whole new action, which might be a complex action, you might only have to make one or two modifications to the pasted action.

To copy actions within the same node, proceed as follows:

1. From the drawing area, right-click the node shape containing the defined action that you will copy.
2. Select Properties from the pop-up menu to display the Properties dialog box for that node shape.
3. In the Actions portion of the window, select the appropriate tab containing the action to be copied: Created, Activated, Executed, or MarkedDone.
4. Right-click the defined action that you will copy, and select Copy from the pop-up menu.

Figure 5-11 Copy Action



Task Name: Task 1

Priority: Medium

Mark task "Task 1" done

Set task "Task 1" due date to Date 'm", 1)

☐ Task 1 (T1)

☑ Done (D1)

☐ Task 2 (T2)

5. In the Actions portion of the window, select the appropriate tab where you will paste the action: Created, Activated, Executed, or MarkedDone.
6. Right-click anywhere in the Actions portion of the window.
7. Select Paste from the pop-up menu.
8. Depending on the copied action, a dialog box appears that corresponds to the copied action.
9. Make changes, if needed, to the action definition, and Click OK. The action is copied into the Actions portion of the window.

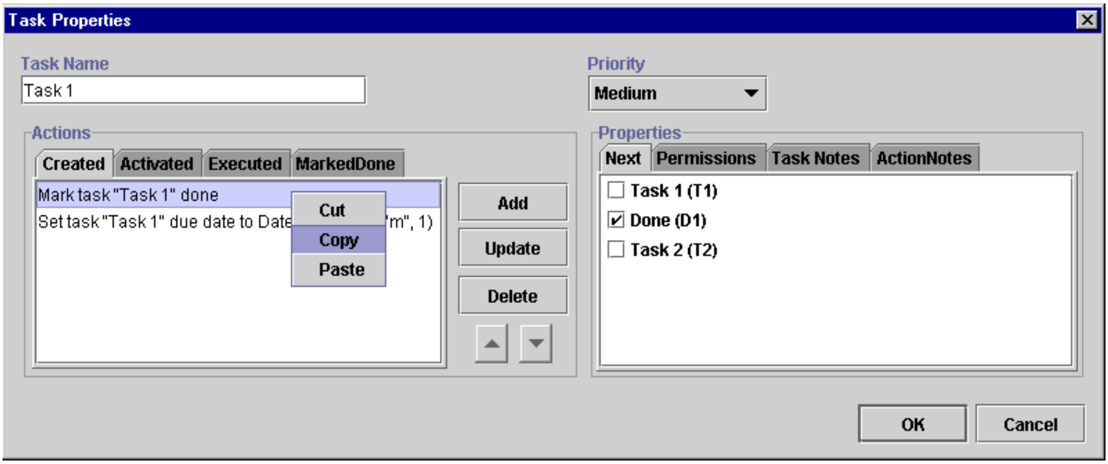
Copying Actions Between Workflow Nodes

The ability to copy actions between workflow nodes benefits you in the same manner as the ability to copy actions within workflow nodes. However, rather than copying actions within the same workflow node, you copy actions within one node and paste them into another node within the same workflow template definition. The ability to copy actions between nodes simplifies and speeds up the workflow definition process. When an action is copied, all of its properties are also copied, so you do not need to redefine all of the properties in the pasted node.

To copy actions between nodes, proceed as follows:

1. From the drawing area, right-click the node shape containing the defined action that you will copy.
2. Select Properties from the pop-up menu to display the Properties dialog box for that node shape.
3. In the Actions portion of the window, select the appropriate tab containing the action to be copied: Created, Activated, Executed, or MarkedDone.
4. Right-click the defined action that you will copy, and select Copy from the pop-up menu.

Figure 5-12 Copy Action


Task Properties

Task Name: Task 1

Priority: Medium

Actions

Mark task "Task 1" done
Set task "Task 1" due date to Date 'm", 1)

Properties

☐ Task 1 (T1)
☑ Done (D1)
☐ Task 2 (T2)

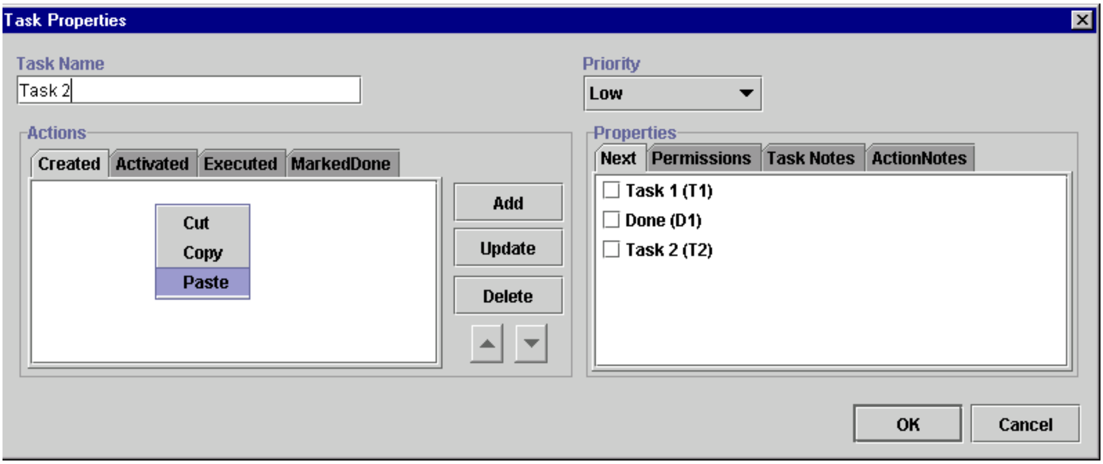
5. From the drawing area, right-click the node shape to which you will paste the defined action.

6. Select Properties from the pop-up menu to display the Properties dialog box for that node shape.

7. In the Actions portion of the window, select the appropriate tab to which the action is to be copied: Created, Activated, Executed, or MarkedDone.

8. Right-click anywhere in the Actions portion of the screen, and select Paste from the pop-up menu.

Figure 5-13 Paste Action



Task Name: Task 2

Priority: Low

9. Depending on the pasted action, a dialog box appears that corresponds to the action.

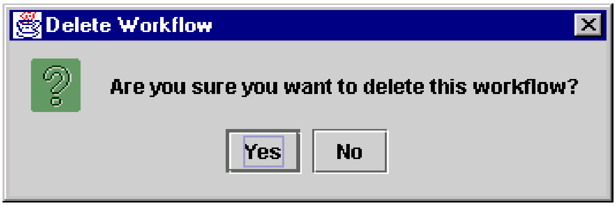
10. Make changes, if needed, to the action definition, and Click OK. The action is pasted into the Actions portion of the window.

Deleting Workflow Template Definitions

To delete a workflow template definition, right-click the workflow template definition name in the folder tree, and choose Delete from the pop-up menu. The Delete Workflow dialog box asks you to confirm or cancel the delete.

Note: The workflow template definition must be closed in the drawing area before you delete it, or you will receive an error message.

Figure 5-14 Delete Workflow Dialog Box



Are you sure you want to delete this workflow?

When a workflow template definition is deleted, the following also occurs:

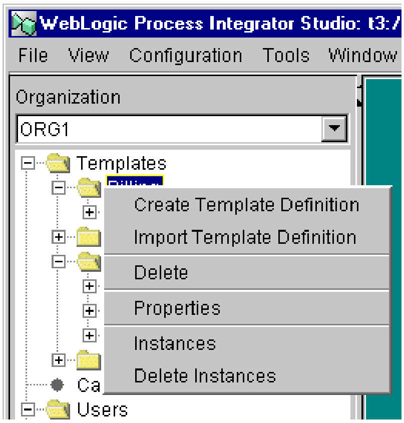
- All instances of that workflow template definition are deleted (including all tasks regardless of their current status).
- All history relating to that workflow template definition is deleted and is therefore no longer available in statistical reports.
- Workload reports that refer specifically to the deleted workflow template definition are changed to refer instead to all remaining workflow template definitions.

Deleting Workflow Templates

You can delete a workflow template from BEA WebLogic Process Integrator, in which case all workflow template definitions within the workflow template are also deleted. To delete a workflow template (and all of its workflow template definitions), proceed as follows:

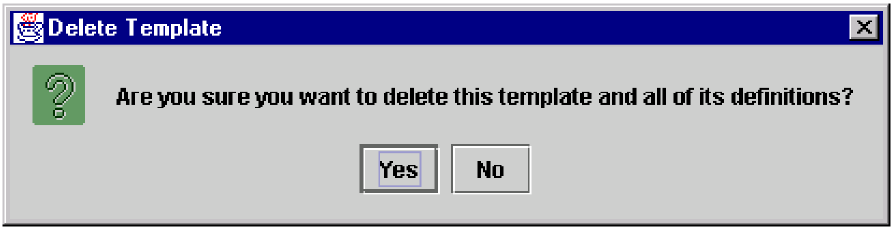
1. Right-click on the workflow template name in the folder tree.

Figure 5-15 Delete Workflow Template



2. Choose Delete to display the Delete Template dialog box.

Figure 5-16 Delete Template Dialog Box



Are you sure you want to delete this template and all of its definitions?

3. Click Yes to delete the workflow template and all of its workflow template definitions, or click No to cancel the delete.

Understanding Effective and Expiry Dates

In the folder tree, a workflow template definition name consists simply of its Expiry and Effective dates. The effective and expiry dates represent the range of time within which the workflow template definition is available for instantiation.

The Effective and Expiry Date features are useful in that they allow you to make workflow template definitions valid for exact periods of time. Perhaps you have a cyclical business, which requires you to run a particular workflow template definition for the first six months of the year, followed by a second workflow template definition having different tasks or variables the remaining six months of the year.

Rather than having to manually inactivate one workflow template definition and activate another at different times, you can define separate workflow template definitions having different properties and make them available at the right time by managing the effective and expiry dates.

Within a single workflow template, a workflow template definition's timeline cannot overlap that of another workflow template definition. In other words, only one workflow template definition can run at a time within one workflow template.

You cannot have two active definitions with the same effective date. When trying to make a workflow template definition active, WebLogic Process Integrator prevents you from doing so, if another definition with the same effective date exists.

When a workflow template definition is started, the WebLogic Process Integrator server creates a new instance (or session) of that workflow template definition. Regardless of how many workflow template definitions exist for a given workflow template, the WebLogic Process Integrator server chooses only one of these definitions to instantiate. It chooses the most effective of the active workflow template definitions. For example, consider three definitions defined with the following effective dates:

- January to March
- April to June
- July to September

If you instantiate your workflow in February, the most effective of the active workflow template definitions for the workflow template is selected, in this case, the one with the effective date of January to March.

Exporting and Importing Workflow Packages

WebLogic Process Integrator provides the capability of exporting and importing workflow objects as Java Archive (JAR) files. This capability allows you to export and import templates, template definitions, business operations, business calendars, plug-ins, event key tables, and XML repository items. Organizations, users, and roles cannot be exported or imported.

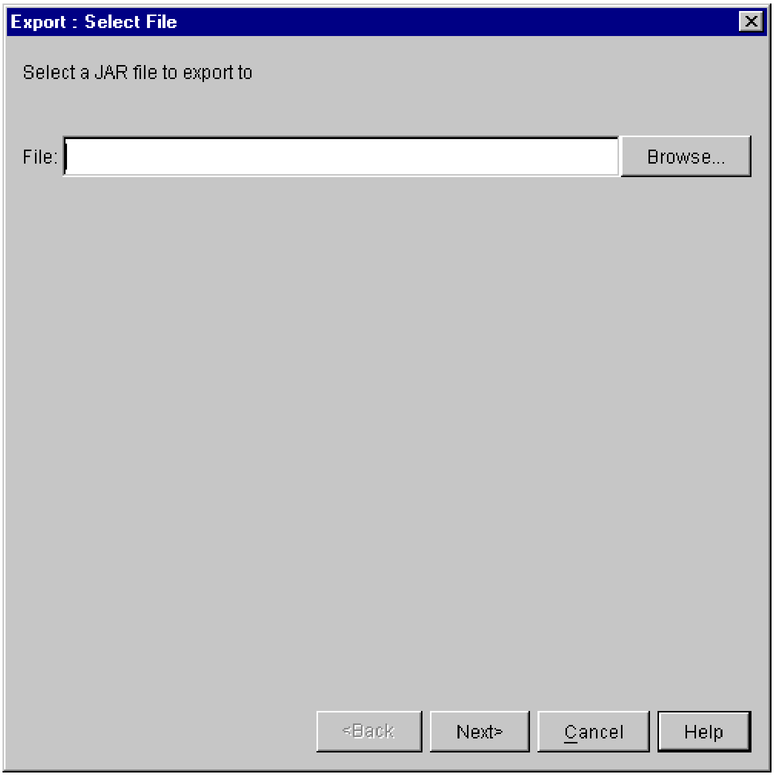
Exported packages can be marked as *published* packages. When a published package is imported, the objects in the package cannot be edited. In addition, you cannot create a new template definition in a published template. You can, however, delete published templates and template definitions.

You can also assign password protection to exported packages to prevent them from unauthorized import.

Exporting Packages

1. Choose Tools→Export Package to display the Export: Select File dialog box.

Figure 5-17 Export: Select File Dialog Box



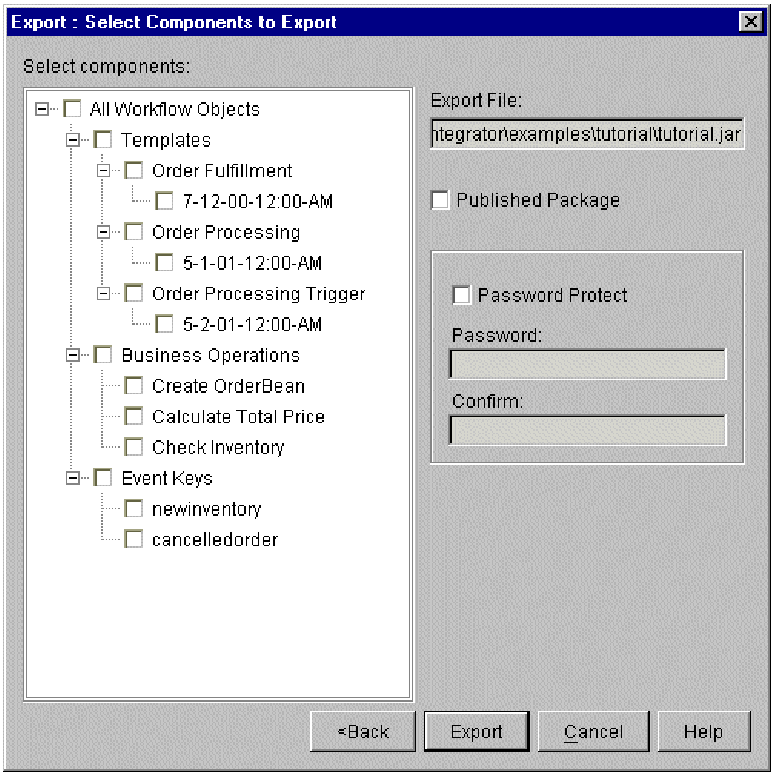
2. Specify the full pathname of the JAR file to which you want to export the package. If you know the full pathname, you can type it directly in the File: field. Alternatively, click Browse to navigate through the file system and select the desired file.

3. Click Next.

   If you specified an existing JAR file that is not password protected, the Export: Select Components to Export dialog box is displayed.

If you specified an existing JAR file that is password protected, the Export: Enter Password dialog box is displayed. Enter the necessary password and click Next to display to display the Export: Select Components to Export dialog box, or click Back to specify an alternate file.

Figure 5-18 Export: Select Components to Export Dialog Box



Export : Select Components to Export

Select components:

- ⊟ □ All Workflow Objects
  - ⊟ □ Templates
    - ⊟ □ Order Fulfillment
      - □ 7-12-00-12:00-AM
    - ⊟ □ Order Processing
      - □ 5-1-01-12:00-AM
    - ⊟ □ Order Processing Trigger
      - □ 5-2-01-12:00-AM
  - ⊟ □ Business Operations
    - □ Create OrderBean
    - □ Calculate Total Price
    - □ Check Inventory
  - ⊟ □ Event Keys
    - □ newinventory
    - □ cancelledorder

Export File: ntegrator\examples\tutorial\tutorial.jar

□ Published Package

□ Password Protect

Password:

Confirm:

4. Complete the fields as described in the following table.

In this field . . .	Do this . . .
Select components	Specify the components to be exported by selecting or deselecting the appropriate check boxes. The export feature does not enforce template definition integrity so you can deselect any object to exclude it from the exported package. This field displays a tree-view of all the workflow objects in the system. If, in the Export: Selecte File dialog box, you selected an existing JAR file on your system in which to export the package, the items in the tree are prechecked, based on the contents of the selected JAR file. **Note:** Check boxes at the folder level have the following three states: ■ Checked—This state indicates that all subobjects are selected. ■ Unchecked—This state indicates that no subobjects are selected. ■ Partially checked (checked with gray background)—This state indicates that at least one subobject is selected. If you select a template definition to be exported, any subworkflows or business operations referenced by the template definition are automatically selected. However, business calendars, event keys, and XML entities are not automatically selected. Therefore, you must manually select these items if you want them to be included in the exported package. If you export a template definition that specifies a due date for a task, and the due date is set using a business calendar, be sure to export the calendar along with the template definition. Similarly, if you export a template definition that uses the XSL Transform action, be sure to export the XML entities in the XML repository required by the XSL Transform action. The XML entities could be the XSLT template document or the XML input document.
Export File	Leave as is. This field is populated with the full pathname of the JAR file you specified in step 2. If you want to change the content of this field, click Back to return to the Export: Select File dialog box, and enter the new information in the File field.

In this field . . .	Do this . . .
Published Package	Select this check box to export the package as a published package. The objects of a published package are read-only and cannot be edited after they are imported to the target system.
Password Protect	1. Select this check box if you want to assign a password to protect the package from unauthorized import. 2. Enter the password in the Password: and Confirm: fields. When you assign a password to a package, the user on the target system must specify this password before they can import the package.

3. Click Export to begin the export.

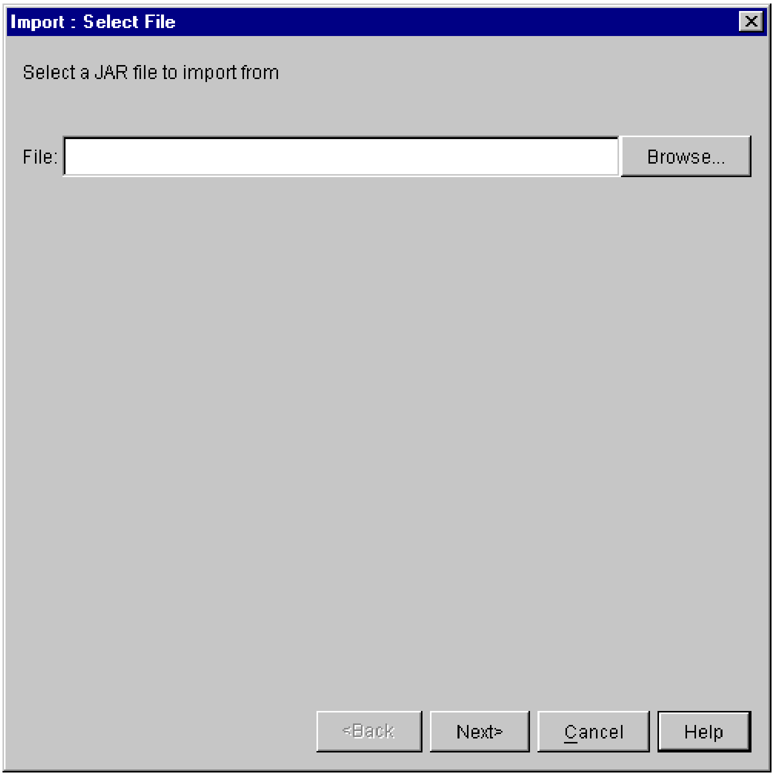
   After the export operation is complete, a Review Export Summary report is displayed indicating if the export operation was successful.

4. Click Close to close the Review Export Summary dialog box, or Back to return to the previous screen.

Importing Packages

1. Choose Tools→Import Package to display the Import: Select File dialog box.

Figure 5-19 Import: Select File Dialog Box



Import : Select File

Select a JAR file to import from

File:

2. Specify the full pathname of the JAR file to import. If you know the full pathname, you can type it directly in the File: field. Alternatively, click Browse to navigate through the file system and select the desired file.

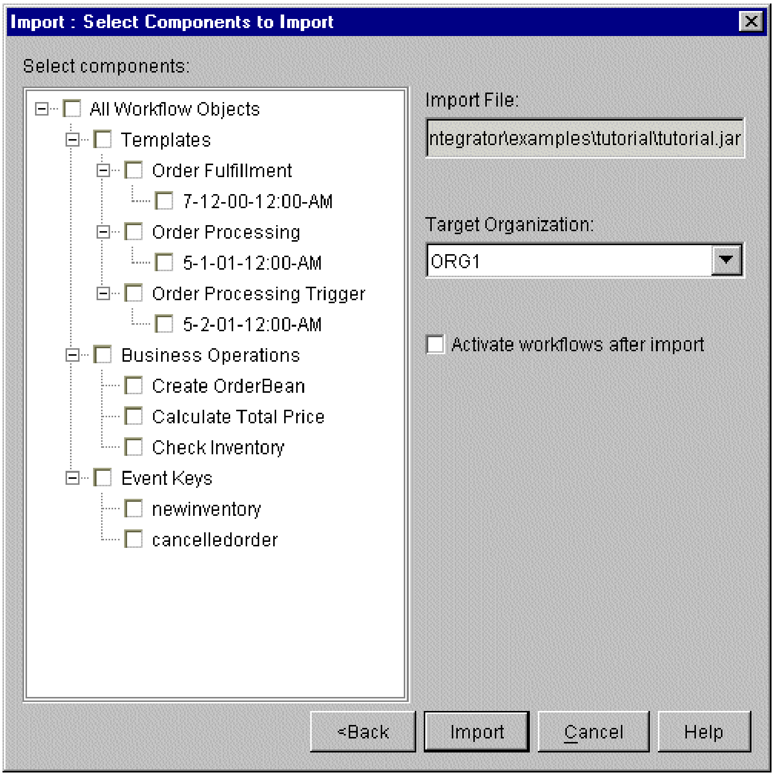
3. Click Next.

   If the package you want to import is not password protected, the Import: Select Components to Import dialog box is displayed.

   If the package you want to import is password protected, the Import: Enter Password dialog box is displayed. Enter the necessary password and click Next to display the Import: Select Components to Import dialog box.

Figure 5-20 Import: Select Components to Import Dialog Bo

x



Import : Select Components to Import

Select components:

- □ All Workflow Objects
  - □ Templates
    - □ Order Fulfillment
      - □ 7-12-00-12:00-AM
    - □ Order Processing
      - □ 5-1-01-12:00-AM
    - □ Order Processing Trigger
      - □ 5-2-01-12:00-AM
  - □ Business Operations
    - □ Create OrderBean
    - □ Calculate Total Price
    - □ Check Inventory
  - □ Event Keys
    - □ newinventory
    - □ cancelledorder

Import File: ntegrator\examples\tutorial\tutorial.jar

Target Organization: ORG1

□ Activate workflows after import

4. Complete the fields as described in the following table.

In this field . . .	Do this . . .
Select components	This field displays a tree-view of all the workflow objects in the JAR file to be imported. All tree items are selected by default. Clear the check box for any object that you do not want to import. If you import a template definition that specifies a due date for a task, and the due date is set using a business calendar, be sure to import the calendar along with the template definition. **Note:** Since business calendars are referenced indirectly by organizations, users, and roles, the references are not maintained during the import process. After importing the package, be sure to reassign the necessary organizations, users, and roles to the business calendar. Similarly, if you import a template definition that uses the XSL Transform action, be sure to import the XML entities in the XML repository required by the XSL Transform action. The XML entities could be the XSLT template document or the XML input document.
Import File	Leave as is. This field is populated with the full pathname of the JAR file you specified in step 2. If you want to change the content of this field, click Back to return to the previous dialog box, where you can enter new information in the File field.
Organization	From the drop-down menu, select the organization to which the imported template definition should be assigned.
Activate after import	Select this check box if you want the imported template definitions to be activated automatically after they are imported. **Note:** If a template definition has unresolved references to objects, it cannot be activated.

5. Click Import to begin the import operation, or Cancel to cancel the operation.

   Note: During the import process, a warning is displayed if you are attempting to import an object with the same name as an object that already exists on your system. You can overwrite templates, business operations, and business calendars. You are not permitted to overwrite template definitions.

If you are importing a template definition from a published package (that is, a package that you previously exported and marked as published), and a template does not exist on the target system, a published template is automatically created. If, however, you are importing a published template definition into an existing template, the existing template must also be published. You can not import a published template definition into an unpublished template.

Similarly, if you are importing a template definition from an unpublished package and a template does not exist on the target system, an unpublished template is automatically created. To import an unpublished template definition into an existing template, the existing template must also be unpublished. You can not import an unpublished template definition into an published template.

After the import operation is complete, a Review Import Summary report is displayed. The report lists all the objects that were imported and any problems that were encountered during the import operation.

Click Close to close the Review Export Summary dialog box, or Back to return to the previous screen.

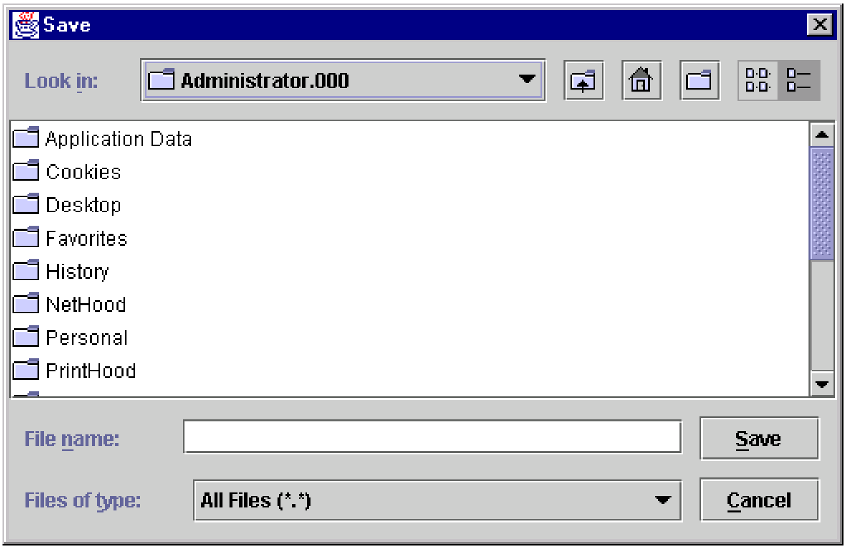
Exporting Workflow Template Definitions

For compatibility with earlier versions of BEA WebLogic Process Integrator, you can export a workflow template definition, in XML format, from BEA WebLogic Process Integrator to any drive mapped in your computer.

To export a workflow template definition:

1. In the folder tree, right-click the template definition to be exported.
2. Choose Export from the pop-up menu.The Save dialog box is displayed.

Figure 5-21 Save Dialog Box



Look in: Administrator.000

File name:

Files of type: All Files (*.*)

3. Select the drive and the directory in which you want to save the exported workflow template definition.

4. Click Save to export the template definition, or Cancel to cancel the operation.

Importing Workflow Template Definitions

For compatibility with earlier versions of BEA WebLogic Process Integrator, you can import a workflow template definition (that was previously exported), in XML format, into BEA WebLogic Process Integrator to create a new workflow template definition.

Note: Imported workflow template definitions are always marked as "inactive." Before an imported workflow template definition can be instantiated, you must change the definition to "active" in the Template Definition dialog box. For details, see "Creating a New Workflow Template Definition" on page 5-4.

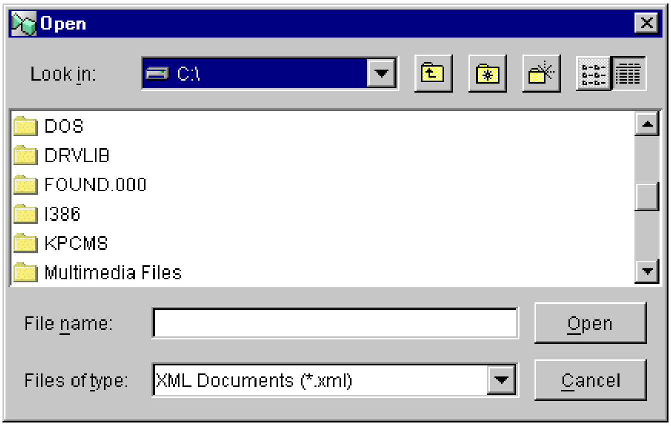
To import a workflow template definition, proceed as follows:

1. In the BEA WebLogic Process Integrator folder tree, right-click the workflow template into which you will import the workflow template definition.

   Note: If a template does not exist, you must create one before you can import a template definition. For details about creating a template, see "Creating a New Workflow Template" on page 5-3.

2. From the pop-up menu, select Import Template Definition to display the Open dialog box.

Figure 5-22 Open Dialog Box



Open

Look in: C:\

DOS
DRVLIB
FOUND.000
I386
KPCMS
Multimedia Files

File name:

Files of type: XML Documents (*.xml)

3. In the Open dialog box, select the current location of the exported workflow template definition file on your hard drive, and click Open.

   Note: You cannot import a workflow template definition using a name that is different from the one that it was assigned when it was exported. If you attempt to do so, you will receive a warning message.

4. After the file is imported, an import confirmation dialog box is displayed that asks you to confirm the import. Click Yes to import the workflow template definition.

   Note: In some cases, importing workflows may display warning messages. When a workflow template definition contains a Perform Business Operation or Start Workflow action, the action must be redefined when the it is imported in a workflow.

CHAPTER

6 Working with Workflow Components

The following sections describe how to use workflow components:

- Adding Workflow Components to a Workflow Diagram
- Deleting Components from a Workflow Diagram
- Understanding Variables
- Defining Start
- Defining Dones
- Defining Decisions
- Defining Tasks
- Defining Events
- Using Joins (And/Or)

Adding Workflow Components to a Workflow Diagram

After you first create a workflow template definition, the workflow toolbar and workflow drawing area are displayed on the right side of the WebLogic Process Integrator Studio main window.

By default, the drawing area already contains three shapes (components) that you must define: start, task, and done. If you will be adding a shape to an existing workflow diagram, right-click the workflow template definition in the folder tree and choose Open from the pop-up menu to open the workflow diagram.

Add new shapes to the diagram by clicking the shape in the toolbar, then clicking in the workflow diagram where you want to drop the shape. To identify icons to the toolbar, see “Workflow Drawing Area and Toolbar” on page 2-12.

Deleting Components from a Workflow Diagram

Delete a component from a workflow diagram in one of two ways:

- Select the shape in the drawing area, right-click, and choose Delete from the pop-up menu.
- In the folder tree, right click the Start node and choose Delete from the pop-up menu.

A confirmation dialog box is always presented.

Understanding Variables

Not all workflow template definitions require variables. However, for those workflow template definitions containing processes that require variables, you should define these before you begin defining the workflow template definition. You can also add these variables as you create the workflow template definition.

Process variables have a workflow-global scope. That is, a single variable instance is shared by all objects within a workflow template definition instance. Process variables are therefore defined at the workflow template definition level in the folder tree and are referred to as workflow variables.

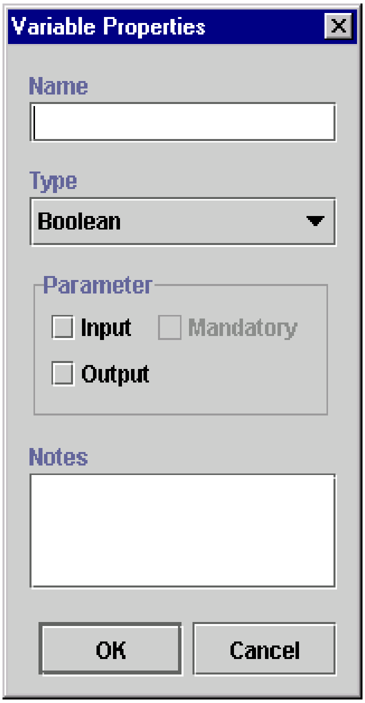
Each workflow template definition can have a set of variables associated with it. Variables can hold values that are returned by business operations, that are extracted from XML documents, or that are set explicitly by workflow actions. Variables can also be used by the workflow for several other purposes, such as to evaluate a condition in a decision node, or to store the result of a response from the Worklist client application.

Defining Variables

To define a variable:

1. In the folder tree, right-click Variables under the appropriate workflow template definition, and choose Create Variable to display the Variable Properties dialog box.

Figure 6-1 Variable Properties Dialog Box



Type
Boolean

Note: If a plug-in is defined for variables, this properties dialog box will contain the new variables in the Type field. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

2. Complete the following fields:

- Name — enter a meaningful name for the variable, such as ItemNumber.
- Type — select one of the following variable types.

Variable Type	Contains . . .
Boolean	Boolean value True or False
Date	Java date object
Double	Double type integer

Variable Type	Contains . . .
Entity EJB	Reference to an Entity EJB called by the Perform Business Operation action (see “Perform Business Operation” on page 7-57)
Integer	Short type integer
Java Object	Reference to a Java class called by the Perform Business Operation action (see “Perform Business Operation” on page 7-57)
Session EJB	Reference to a Session EJB called by the Perform Business Operation action (see “Perform Business Operation” on page 7-57)
String	Character string
XML	XML document, and can be used by the Set Workflow Variable action (see “Set Workflow Variable” on page 7-27)

- Parameter — input or output. For Input, specify whether the parameter is a mandatory one.

  The parameter is used by a calling workflow to pass values to and receive values from a called subworkflow. The input parameter contains the values passed to the subworkflow, and the output parameter contains the return values from the subworkflow. For details about starting another workflow, see “Start Workflow” on page 7-25.

- Notes — a free-form text field that can be used to make notes for the variable.

Updating Variables

To update an existing variable:

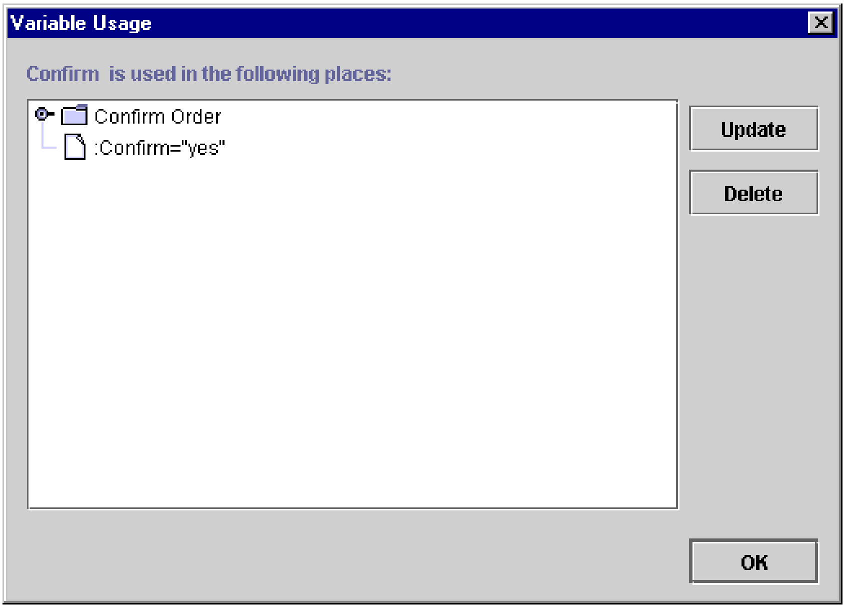
1. Right-click an existing variable in the folder tree and choose Properties from the pop-up menu. The Variable Properties dialog box is displayed.

2. Make changes to the variable as needed, and click OK.

Locating Variables in the Workflow

To see where a variable is used within the workflow, right-click the variable in the folder tree and choose Usage from the pop-up menu. This displays the Variable Usage dialog box.

Figure 6-2 Variable Usage Dialog Box



Confirm is used in the following places:

Confirm Order

:Confirm="yes"

The Variable Usage dialog box lists the nodes within a workflow template definition in which the selected variable is used and the value assigned to that variable. Variables can be used in all workflow nodes except joins and dones.

In the preceding figure, the Confirm variable is used in the Confirm Order task and has a value of `:Confirm="yes"`.

The dialog box contains the following buttons:

- Update — updates the selected action. Opens the action dialog box to allow the variable expression to be edited.
- Delete — deletes the selected action.
- OK — closes the Variable Usage dialog box.

Variable Conversion Rules

The following table details the implicit conversions that are handled automatically by WebLogic Process Integrator between different variable types when values of one type are assigned to variables of another type.

Table 6-1 Variable Conversions

Value Converted To	String	Integer	Double	Date	Boolean	XML	Node List[1]
String	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]
Integer	X[9]	X[10]	X[11]	NS	X[12]	X	X[13]
Double	X[14]	X[15]	X[16]	NS	X[17]	X	X[18]
Date	X	NS	NS	X[19]	NS	X	X
Boolean	X[20]	X[21]	X[22]	NS	X[23]	X	X[24]
XML	X[25]	NS	NS	NS	NS	X[26]	X[27]

Notes: NS means Not Supported. If one of the unsupported conversions are attempted, an error message is displayed indicating that the conversion is illegal.

Additionally, the conversion rules are not fully supported for calculations in expressions. For example, assume you have the following expression, and you want to assign the result to a string type variable:

```
XPath(“your_expression”) + 2
```

If `XPath(“your_expression”)`evaluates to a text value of 5, the value is not automatically converted to a string, and an error message is displayed.

To convert the result to a string, you need to indicate explicitly that you want the result of the expression converted. The expression needs to be modified as follows:

```
tostring XPath(“your_expression”) + 2
```

The following are notes relating to certain conversions in the preceding table:

1. org.w3c.dom.NodeList, which is typically the output of Xpath() function. The Xpath() function also returns Double, Boolean, String, and Integer types.
2. As it is.
3. Java Integer.toString() output.
4. Java Double.toString() output.
5. Java Date.toString() output.
6. Java Boolean.toString() output: “true” and “false”.
7. DOM serializing.
8. DOM serializing.
9. An ILLEGAL_CONVERSION warning WorkflowException is thrown if parsing cannot occur (NumberFormatException).
10. As it is.
11. Double’s Integer value.
12. 1 if true, 0 if false.
13. DOM serialized to string, then convert to Integer. (See note 9.)
14. An ILLEGAL_CONVERSION warning WorkflowException is thrown if parsing cannot occur (NumberFormatException).
15. Integer’s Double value.
16. As it is.
17. 1.0 if true, 0 if false.
18. DOM serialized to string, then convert to Double. (See note 14.)
19. As it is.
20. True if string value is “true”, “t”, “1”; false if string value is “false”, “f”, “0”. Any other string value throws an ILLEGAL_CONVERSION warning WorkflowException.
21. False if Integer value is 0, true if any other value.

22. False if Double value is 0, true if any other value.

23. As it is.

24. DOM serialized to string, then convert to Boolean. (See note 20.)

25. DOM parsing to a XML document element (not simply a node, which means when it is dumped back to a String, it will have XML header similar to the following: `<?xml version=”1.0” encoding=”UTF-8”?>)`. An ILLEGAL_CONVERSION warning WorkflowException is thrown if DOM parsing cannot occur.

26. As it is.

27. DOM serialized to string, then convert to XML node. (See note 25.)

Defining Start

Every workflow has at least one start shape that indicates the beginning of the workflow. The first node after the start will be the first activated node of the workflow, which can be a task, decision, or event.

Specifying Multiple Starts

You can specify more than one start shape, if more than one simultaneous path through the workflow is to be started. For example, when you want two streams of work to start at the same time within a single workflow, you can define two starts within the workflow.

These two streams of work can be started simultaneously, if they have the same triggering event and both are marked as manual starts or timed starts set to the same time. Otherwise, the two streams can be started separately with each start being an alternate way of starting the workflow.

The benefit of being able to specify more than one start is visual access to both streams of work within the same workflow, and sharing the same variables with both workflows. You also have the flexibility to take a portion of a workflow and start it in a different manner or at a different time within the workflow.

Note: If no Start node is specified, no task activation can occur in the workflow.

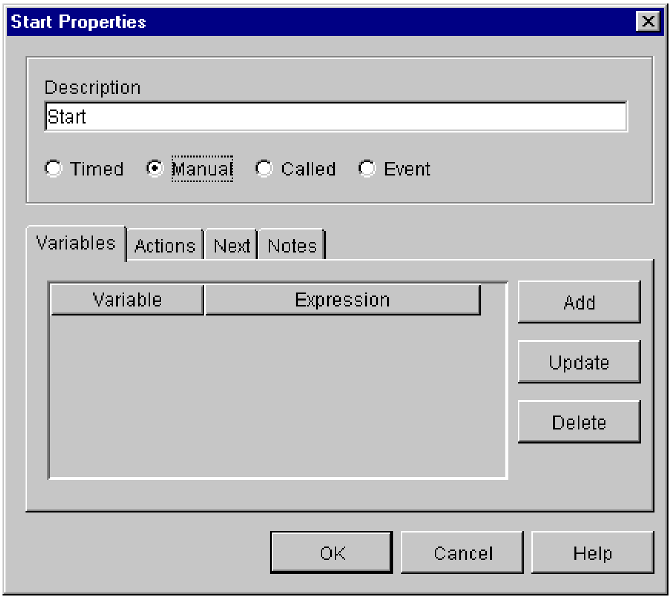
You specify the triggering properties of a workflow in the start node. If a workflow contains multiple starts, each start can have its own triggering method and start independently.

Defining a Start Node

To define a Start shape:

1. Display or add the Start shape as described in “Adding Workflow Components to a Workflow Diagram” on page 6-2.
2. Double-click the Start shape to display the Start Properties dialog box.

Figure 6-3 Start Properties Dialog Box



Start Properties

Description

Start

○ Timed ● Manual ○ Called ○ Event

Note: If a plug-in is defined for the Start node, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, “Working with Plug-Ins.”

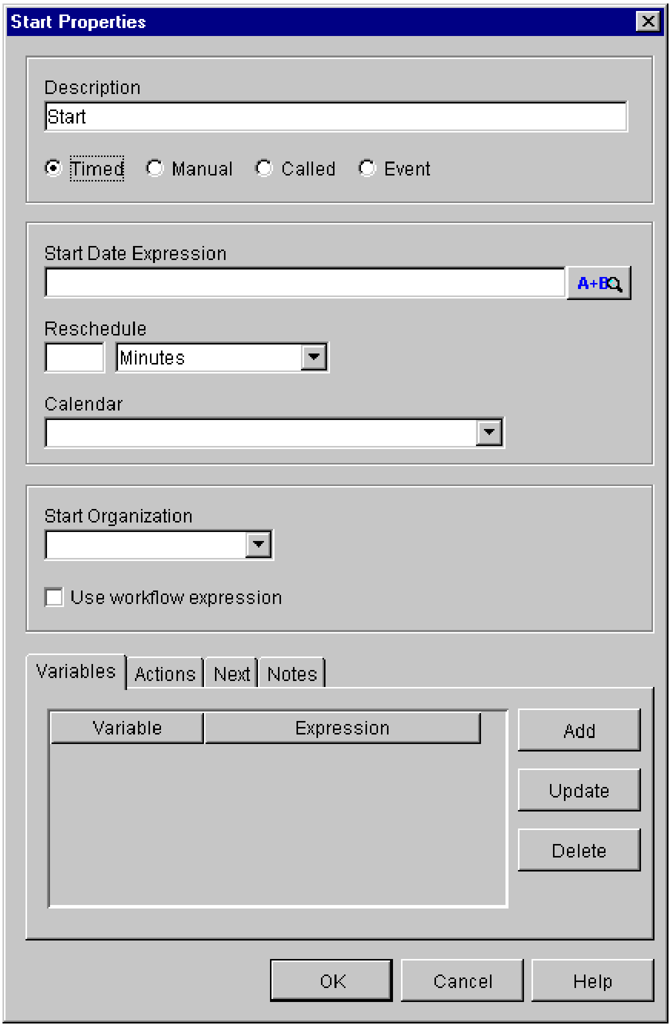
3. The Description field by default contains the name Start. Modify this name as needed to create a unique, identifiable name.

4. Select a workflow triggering method:

   - Event — the workflow starts when it receives an XML document. If you choose this triggering option, complete the procedure in “Defining a Start Node Using an Event Trigger” on page 6-14.

   - Timed — the workflow runs as a time-scheduled job, starting on the date and time you define for the timed option properties.

Figure 6-4 Timed Option



Start Properties

Description: Start

● Timed ○ Manual ○ Called ○ Event

Start Date Expression:

Reschedule: Minutes

Calendar:

Start Organization:

☐ Use workflow expression

Start Date Expression — enter the expression that defines the start date for the workflow. The expression is evaluated at run time, and becomes the start date. You must use a date function, such as Date or DateAdd, in this expression. For details, see "Using Expression Components" on page 8-6.

Reschedule — reschedule the timed workflow. If you define this option, the workflow will be rescheduled according to the incremental values entered here.

Calendar — select the business calendar that is used to evaluate the start date.

Start Organization — select a Start Organization from the drop-down list containing those organizations already defined within WebLogic Process Integrator. When the start node is an event or timed type, you must specify the organization for which the workflow instance will be started because you can start the same workflow template definition for different organizations.

If you want to define the Start Organization using an expression, select the Workflow Expression option. The Expression Builder is displayed. Use it to enter a string that specifies the organization, or an expression that will be evaluated at run time and become the name of an organization.

Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

- Manual — the workflow is started manually by an end user from the Worklist client application. This type of start requires that the end user know the specific business conditions that determine when to start the workflow. You might create a manual start when you cannot create a trigger that captures all required conditions.
- Called — the workflow is started by a call from another workflow using the Start Workflow action. For details, see "Start Workflow" on page 7-25.

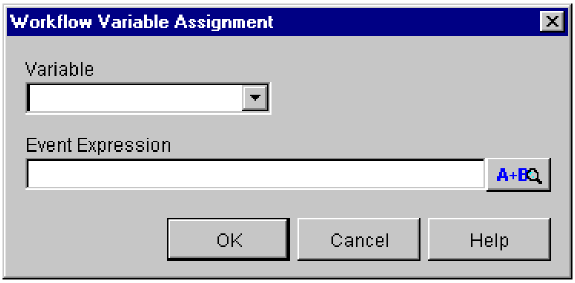
5. Complete or view the fields on the tabs:

- Variables — specify the variables that should be initialized when the workflow starts.

  For example, you might have a counter in your workflow that you wish to take the value of 1 at the start of the workflow.You can assign a variable specifying that the counter takes the value of 1 upon initialization of the start node.

Clicking Add displays the Workflow Variable Assignment dialog box. The variables in the pull-down list are those defined in the Variables Properties dialog box. (See "Understanding Variables" on page 6-3.) The XML expression can be a string that you want to use as the value of the variable, or it can be an expression that is evaluated at run time to become the value of the variable.

Figure 6-5 Workflow Variable Assignment Dialog Box



- Next — displays a list of all task, decision, join, and done nodes in the current workflow. An start can have one or more successors. Select the node(s) that will serve as the next activated node(s) in the workflow when the start node is initiated.
- Actions — displays the list of all actions to be performed when the start node is initiated. Click Add to display the list of available actions. For further details about selecting actions, see Chapter 7, "Defining Actions."
- Notes — a free form text field for recording notes about the start.

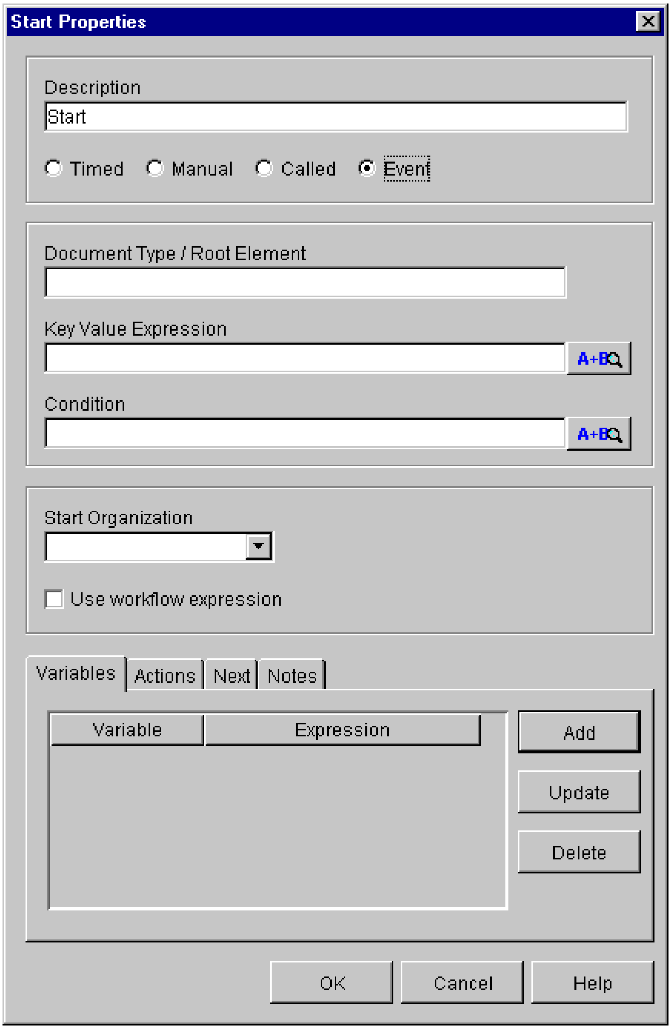
Defining a Start Node Using an Event Trigger

A workflow can be started by responding to an event. An event is an asynchronous notification from another workflow or from an external source, such as another application. An event notification takes the form of an XML message. For further information about events and how they are triggered, see Appendix A, "Understanding Event-Driven Processing."

To define an event trigger to start a workflow:

1. In the Start Properties dialog box, select the Event option.

Figure 6-6 Start Properties Dialog Box



Start Properties

Description: Start

○ Timed ○ Manual ○ Called ● Event

Document Type / Root Element:

Key Value Expression:

Condition:

Start Organization:

□ Use workflow expression

2. In the Document Type/Root Element field, enter the DOCTYPE or root element in the XML message that triggers the start of the workflow.

The DOCTYPE or root element you specify here must match the DOCTYPE or root element in the incoming XML message before BEA WebLogic Process Integrator starts the workflow.

3. In the Key Value Expression field, optionally define a key value for the XML message. You can enter a string that is the key value, or an expression that is evaluated at run time to produce the key value.

   The key value you specify here must match an element in the incoming XML message before BEA WebLogic Process Integrator starts the workflow.

   You also need to define the expression that locates the key value in the incoming XML message, so BEA WebLogic Process Integrator can compare it against the key value you specify in this field. For details, see “Defining an Event Key” on page A-4.

4. In the Condition field, optionally define a condition that needs to be evaluated before the workflow can start. If the condition is true, BEA WebLogic Process Integrator starts the workflow.

5. Select a Start Organization from the drop-down list of organizations already defined within WebLogic Process Integrator. When the Start node is the Event type, you must specify the organization for which the workflow instance is started because you can start the same workflow template definition for different organizations.

6. If you want to define the Start Organization using an expression, select the Workflow Expression option. The Expression Builder is displayed. Use the Expression Builder to enter an expression that is evaluated at run time and becomes the name of an organization. The expression can be either a case-sensitive string that specifies the organization, or an expression using information passed in the XML message.

   Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

7. The XML message that starts the workflow can be parsed to extract information to be used in the initialization of workflow variables. Select the Variables tab to specify the variables that should be initialized when the workflow starts.

   Click Add to display the Workflow Variable Assignment dialog box. The variables in the pull-down list are those defined in the Variables Properties dialog box. (See “Understanding Variables” on page 6-3.) The XML expression can be the name of an element in the XML message the value of which you want to assign to the selected variable, or it can be an XPath function that extracts a value from the XML message and assigns it to the selected variable.

8. Select the Next tab to display a list of all task, decision, join, and done nodes in the current workflow. Select a node (or nodes) in the list to be activated next in the workflow when the Start node is instantiated.

9. Select the Actions tab to display a list of all actions to be performed when the Start node is initiated. Click Add to display a the list of available actions. For details about defining an action, see Chapter 7, “Defining Actions.”

10. Click OK.

Defining Dones

A done shape represents the point within the workflow where the process completes. Once a done node is reached in the workflow, the running instance of the workflow is marked done, regardless of whether all the done nodes have been reached or not.

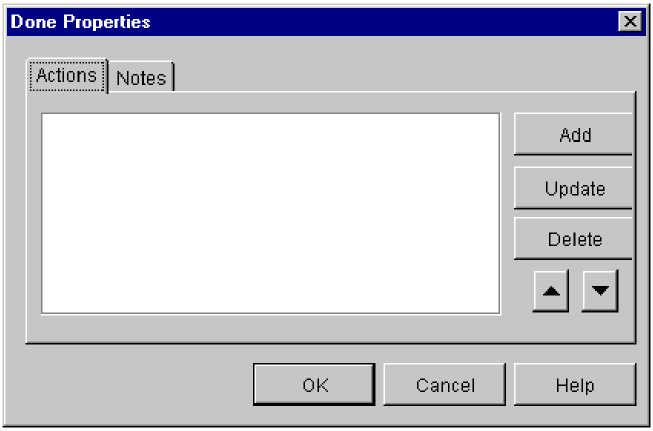
You can specify any number of done shapes in a workflow. If a workflow can exit from various points, you have the option of placing separate done nodes wherever needed.

To define a done:

1. Add or view a done shape as described in “Adding Workflow Components to a Workflow Diagram” on page 6-2.

2. Double-click the done shape or right-click it in the folder tree and choose Properties to display the Done Properties dialog box.

Figure 6-7 Done Properties Dialog Box



Note: If a plug-in is defined for the Done node, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, “Working with Plug-Ins.”

3. Complete the following:

   - Actions — displays the list of all actions to be performed when the Done node is reached. Click Add to display the list of available actions. Refer to Chapter 7, “Defining Actions,” for further details on selecting actions.
   - Notes — a free form text field for recording notes about the done.

Defining Decisions

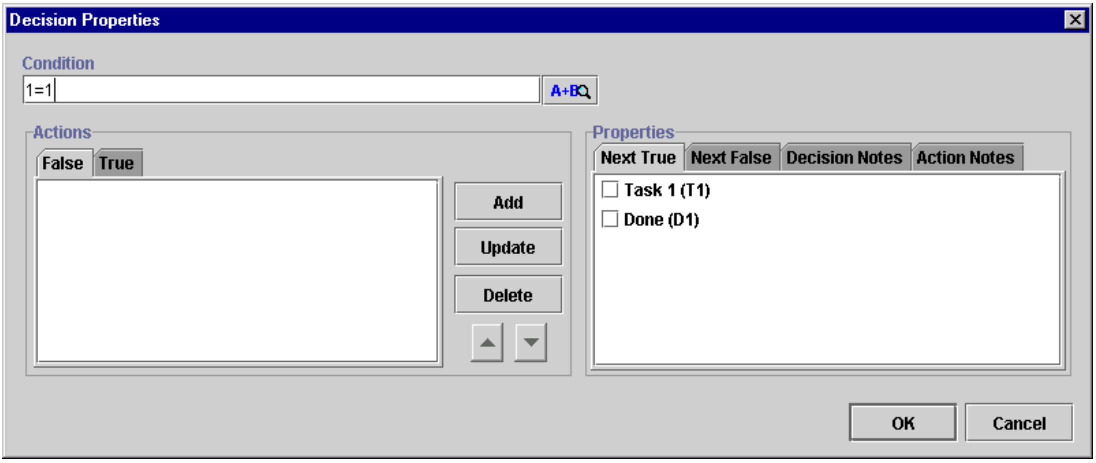
The workflow can use any number of decision shapes. Each decision shape contains a condition that is evaluated when a transition to that decision node occurs. The result is either True or False, with subsequent flow of control passed to the subsequent task, decision, event, and join or done node(s). Additionally, it is possible to specify actions to be executed on both a True and a False evaluation of the condition. (If the condition is verified, it is True. If it is not, it is False.)

To define a decision:

1. Add or view a Decision shape as described in "Adding Workflow Components to a Workflow Diagram" on page 6-2.

2. Double-click the Decision shape or right-click it in the folder tree and choose Properties to display the Decision Properties dialog box.

Figure 6-8 Decision Properties Dialog Box



Condition: 1=1

☐ Task 1 (T1)

☐ Done (D1)

3. Complete the following fields:

   - Condition — specify the conditional expression to evaluate at run time. When the decision is activated in a running workflow (that is, a transition is made to the decision shape), this conditional expression is evaluated.

   Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

   - Actions — a tabbed display showing the list of actions to be executed if the specified condition is evaluated and the result is either False or True. The tab that appears foremost indicates which of the two lists (False or True) is currently displayed. See Chapter 7, “Defining Actions,” for a description of how to specify actions.

False — actions defined in this tab are executed if the condition evaluates to false (is not verified).

True — actions defined in this tab are executed if the condition evaluates to true (is verified).

4. Provide or view information in the tabs of the Properties section of the dialog box.

   - Next True — a list of all nodes in the current workflow. On a True evaluation of the specified Decision condition, the indicated item(s) will be the next activated in the flow. If you have defined a sequence to the shapes on the workflow diagram using connector lines, a check mark is displayed next to any successor shapes. Place a check mark beside any other shape you want to execute after a True evaluation of this decision. A connector line is automatically drawn on the workflow diagram. You can also create the True connection from the decision shape to a subsequent node by clicking the connector on the drawing area toolbar, clicking in the drawing area to drop.

   - Next False — a list of all nodes in the current workflow. On a False evaluation of the specified decision condition, the indicated item(s) will be the next activated in the flow. If you have defined a sequence to the shapes on the workflow diagram using connector lines, a check mark is displayed next to any successor shapes. Place a check mark beside any other shape you want to execute after a False evaluation of this decision. A connector line is automatically drawn on the workflow diagram. The False connection from the decision shape to a subsequent node can also be created by clicking the connector on the drawing area toolbar, clicking in the drawing area to drop it, and then moving its end points to make the connections.

   - Decision Notes — a free-form text field that can be used to make comments or notes for the Decision shape.

   - Action Notes — a read-only text field showing the defined notes for the action selected on the left-hand side of the dialog box.

Defining Tasks

A task node represents a discrete business activity that is performed either automatically when a workflow runs, or manually by a user using the Worklist client application.

Each task node progresses through four different states: created, activated, executed, and marked done. Progress through these states is not automatic. Each state is activated by some event. For each state, you can specify a list of actions that are performed when the task reaches that state.

The following table shows each task state, and when the actions specified in each state are executed:

Table 6-2 Task States

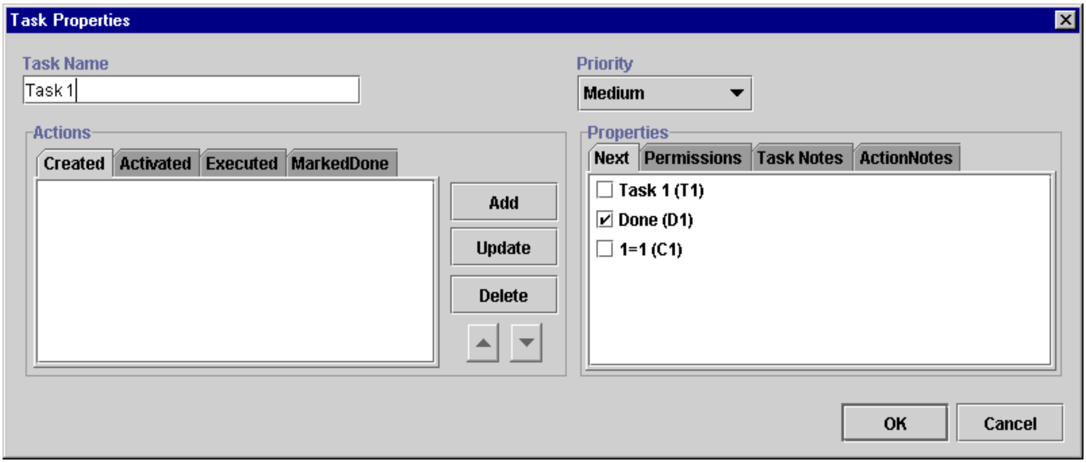
Task states	Actions specified in this state are executed . . .
Created	When a new workflow instance is created
Activated	When the previous node has completed its processing The task remains in Activated state until the task is marked done.
Executed	■ When a user manually executes the task using the Worklist application ■ Using the Execute Task action ■ Programmatically, using the BEA WebLogic Process Integrator APIs
Marked Done	■ Using the Mark Task as Done Action ■ Programmatically using the BEA WebLogic Process Integrator APIs

To define a task:

1. Display or add a task as described in “Adding Workflow Components to a Workflow Diagram” on page 6-2.

2. Double-click the task shape or right-click the task in the folder tree and choose Properties to display the Task Properties dialog box.

Figure 6-9 Task Properties Dialog Box



Task Properties

Task Name: Task 1

Priority: Medium

Properties — Next:

□ Task 1 (T1)
☑ Done (D1)
□ 1=1 (C1)

3. Complete the following fields:

   - Task Name — enter a meaningful name, such as ConfirmOrder. This value is used primarily in the Worklist application, but it is visible from the various BEA WebLogic Process Integrator monitoring reports. The system automatically assigns a unique temporary name to new tasks (in the form of `Task n`, where n is a number).

   - Priority — select Low, Medium, or High priority. This field assigns a priority level to the task in the Worklist application. The user can then perform tasks by priority level. The default value is Medium.

   - Actions — tabbed display of four separate states for the task: Created, Activated, Executed, MarkedDone. You can specify actions to be performed in each task state. For details about specifying actions, see Chapter 7, “Defining Actions.”

4. View or provide information on the following tabs:

   - Next — a list of all nodes in the current workflow. When this task is marked done, the selected item(s) will be the next activated in the flow. If you have defined a sequence to the shapes on the workflow diagram using connector lines, a check mark (✓) is displayed next to any successor shapes. Place a check mark beside any other shape you want to execute after this task is complete. A connector line is automatically drawn on the workflow diagram. You can also create the connection from the task shape to a subsequent shape by dragging the Next connector from the workflow toolbar, dropping it on the workflow drawing and then moving its end points to make the connections.

- Permissions — for tasks assigned to a user, specify what the user can do when handling the task. These selections appear in the Worklist client application. Five different permissions are available for each task:

  Mark done without executing — allows users to mark the task as having been done without actually executing it.

  Re-execute if marked done — allows users to re-execute the task when it is already marked as done.

  Unmark if marked done — allows users to change the status of the task back to not done when the task is marked as done. (If a user marks a completed task as not done, the task status is set to Active; it does not, however, reverse the effects of any actions that have been executed.)

  Modifiable — allows users to change the properties of an existing task from the worklist.

  Re-assignable — allows users to reassign the task to another user.

- Task Notes — a free-form text field where you can make comments or notes for the task shape.
- Action Notes — a read-only text field showing the defined notes for the action selected on the left-hand side of the dialog box.

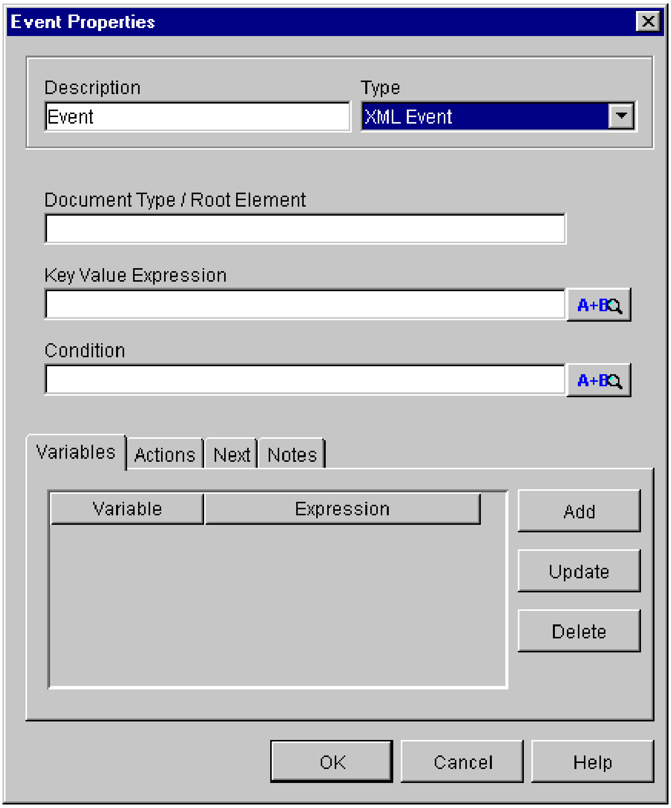
Defining Events

An event node represents a wait state that is activated by the receipt of an XML message. The XML message can come from another workflow or from an external source, such as another application. For further information about events and how they are triggered, see Appendix A, “Understanding Event-Driven Processing.”

To define an event:

1. Display or add a task as described in “Adding Workflow Components to a Workflow Diagram” on page 6-2.
2. Double-click the event shape or right-click it in the folder tree and choose the Properties command to display the Event Properties dialog box.

Figure 6-10 Event Properties



Event Properties

Description: Event

Type: XML Event

Document Type / Root Element

Key Value Expression

Condition

Note: If a plug-in is defined for the Event node, this properties dialog box contains a selection to access the functionality provided by the plug-in. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. In the Description field, enter the name of the event.

4. From the Type drop-down list, select the type of event. For an XML message, select XML Event.

5. In the Document Type/Root Element field, enter the DOCTYPE or root element in the XML message that will trigger the event.

   The DOCTYPE or root element you specify here must match the DOCTYPE or root element in the incoming XML message before BEA WebLogic Process Integrator will trigger the event.

6. In the Key Value Expression field, optionally define a key value for the XML message. You can enter a string that is the key value, or an expression that will be evaluated at run time to produce the key value.

   The key value you specify here must match an element in the incoming XML message before BEA WebLogic Process Integrator will trigger the event.

   You also need to define the expression that locates the key value in the incoming XML message, so BEA WebLogic Process Integrator can compare it against the key value you specify in this field. For details, see “Defining an Event Key” on page A-4.

7. In the Condition field, optionally define a condition that needs to be evaluated before the workflow can start. If the condition is true, BEA WebLogic Process Integrator will trigger the event.

8. The XML message that triggers the workflow can be parsed to extract information to be used in the initialization of workflow variables. Select the Variables tab to specify the variables that should be initialized when the event is triggered.

   Click Add to display the Workflow Variable Assignment dialog box. The variables in the pull-down list are those defined in the Variables Properties dialog box. (See “Understanding Variables” on page 6-3.) The XML expression can be the name of an element in the XML document the value of which you want to assign to the selected variable, or it can be an XPath function that extracts a value from the XML document and assigns it to the selected variable.

9. Select the Actions tab to display a list of all actions to be performed when the Event node is initiated. Click Add to display the list of available actions. For details about defining an action, see Chapter 7, “Defining Actions.”

10. Select the Next tab to display a list of all task, decision, join, and done nodes in the current workflow. Select a node (or nodes) in the list to be activated next in the workflow when the Event node is instantiated.

11. Click OK.

Using Joins (And/Or)

The possibility of multiple paths being active in the same workflow gives rise to the issue of path joining. There are two types of joining: AND joining and OR joining.

How Joins Work

AND joining melds multiple separate paths of control back into a single path, synchronizing them such that control does not continue until all paths converge. This type of join derives its name from the fact that control continues when the AND of all inputs is true. That is, when Path 1 and Path 2 and Path 3 ... etc. are all present, then control proceeds.

OR joining allows multiple path threads to enter and traverse down the same line of control within the workflow. In this case, the threads do not meld; the nodes along this part of the flow are executed multiple times, once for each thread control passing through it. This type of joining derives its name from the fact that control passes to the next node when the OR of its inputs is true. That is, if Path 1 or Path 2 or Path 3 ... etc. is present, then control proceeds.

An OR join can only be activated once. This avoids the multiple occurrence of actions and events succeeding the OR, which can happen by the completion of the various predecessors of the join. However, this single activation prevents a flow from looping when an OR join is defined, as the flow stops the second time at the OR join.

Use AND and OR joins to link task, decision, and event shapes and thus control the transition coming to the join and flowing from it.

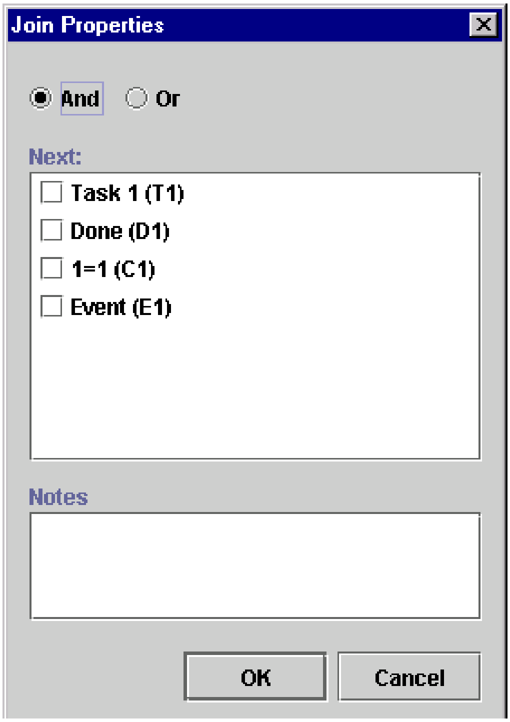
Defining a Join

To define a join:

1. Display or add a join as described in “Adding Workflow Components to a Workflow Diagram” on page 6-2.

2. Double-click the join shape or right-click it in the folder tree and choose Properties to display the Join Properties dialog box.

Figure 6-11 Join Properties Dialog Box



● And ○ Or

Next:

□ Task 1 (T1)
□ Done (D1)
□ 1=1 (C1)
□ Event (E1)

Notes

3. Complete or view the fields as follows:

   - Specify whether the join is an AND or OR.
   - Next — displays a list of all nodes in the current workflow. A join can have one or more successors. Select items in this list to activate them in the flow after the join is satisfied.
   - Join Notes — a free form text field to record any specific notes or comments about the join.

CHAPTER

7 Defining Actions

The following sections introduce and explain how to use BEA WebLogic Process Integrator actions:

- Working with Actions
- Adding or Updating an Action
- Deleting an Action
- Changing the Sequence of Actions
- Using Task Actions
- Using Workflow Actions
- Using Integration Actions
- Using Exception Handling Actions
- Using Miscellaneous Actions

Working with Actions

An action is a basic unit of work that is performed by BEA WebLogic Process Integrator. An action can be as simple as assigning a task to a user or as complicated as sending XML messages or invoking Enterprise JavaBean (EJB) methods. Actions are, in a sense, the primitive building blocks of a workflow because they define the actual behavior of the workflow.

Action Categories

Actions are organized into the following categories.

Type of action	Used to . . .	Actions you can specify
Task (See "Using Task Actions" on page 7-7)	Specify and control tasks	■ Mark Task as Done ■ Unmark Task Done ■ Set Task Due Date ■ Assign Task to User ■ Assign Task to Role ■ Assign Task Using Routing Table ■ Unassign Task ■ Set Task Comment ■ Set Task Priority ■ Execute Task
Workflow (See "Using Workflow Actions" on page 7-23)	Manage an entire workflow	■ Start Workflow ■ Mark Workflow as Done ■ Abort Workflow ■ Set Workflow Variable ■ Set Workflow Comment

Type of action	Used to . . .	Actions you can specify
Integration (See "Using Integration Actions" on page 7-32)	Integrate the workflow with: ■ External software components, such as EJBs and Java classes ■ Applications, such as the Worklist client application or executable programs ■ WebLogic Server functionality, such as the XSL transformation engine	■ Perform Business Operation ■ Post XML Event ■ Call Program ■ Send XML to Client ■ XSL Transform
Exception Handling (See "Using Exception Handling Actions" on page 9-7)	Manage an exception handler	■ Set Workflow Exception Handler ■ Invoke Exception Handler ■ Exit Exception Handler
Miscellaneous (See "Using Miscellaneous Actions" on page 7-62)	Perform additional actions not covered in the other action categories	■ No Operation ■ Evaluate Condition ■ Timed Event ■ Cancel Workflow Event ■ Send E-Mail Message ■ Make Audit Entry

Synchronous and Asynchronous Actions

Most actions as synchronous, which means that the workflow does not proceed to the next action until the current action has completed.

Some actions, however, are asynchronous, which means that the workflow proceeds to the next action without waiting for the current action to complete. The following are asynchronous actions:

- Send XML to Client
- Start Workflow
- Timed Event
- Set Task Due Date

There are, however, a couple of exceptions to the asynchronous behavior of these actions.

If you add subactions to the Send XML to Client action (as described in the following section), this action waits for a reply from the client application before it executes the subactions.

Similarly, if you add subactions to the Start Workflow action, this action waits until the called workflow completes before it executes the subactions.

Action Placement and Subactions

You can place actions in any node in a workflow, except a Join node. The most common place is in a Task node.

You can also place actions within other actions. Such actions are executed when the conditions specific to that action are met. The following actions can have subactions defined within them:

- Evaluate Condition — add actions based on a condition in the False/True tab.
- Timed Event — add actions to be triggered in the Triggered tab.
- Send XML to Client — add callback actions that are performed when the workflow receives a reply from the client application.
- Start Workflow — add actions to be performed when the called workflow is completed.
- Set Task Due Date — add overdue actions. Set a due time on a task so that when the due time passes, the specified subactions are performed.

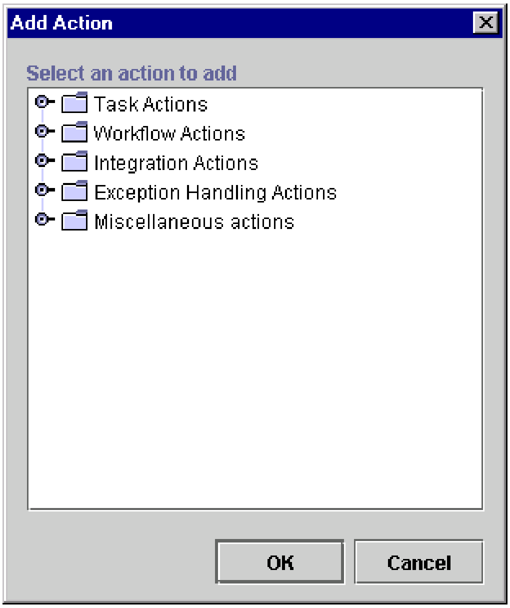
Adding or Updating an Action

You add and update actions in the Task Properties, Decision Properties, Event Properties, and Start Properties dialog boxes. Each dialog box has an area specific to action maintenance.

To add or update an action:

1. Double-click the Task, Event, Decision, Start, or Done node for which you want to add or update an action. The Properties dialog box for the node appears.
2. In the properties dialog box, select the appropriate action tab, and click Add or Update to display the Add Action or Update Action dialog box.

Figure 7-1 Add Action Dialog Box



Note: If a plug-in is defined for actions, this dialog box contains the new action. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. Double-click an action type folder to expand it, select an action, and click OK.

A dialog box specific to the selected action is displayed.

4. Complete the fields in the dialog box and click OK.

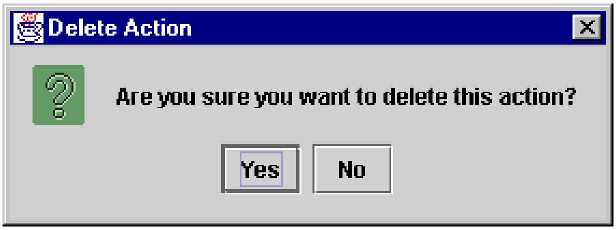
   The action is updated or added to the Properties dialog box for the node you selected.

Deleting an Action

To delete an action:

1. In the workflow diagram, double-click the shape that contains the action you want to delete.
2. Select the desired action in the Actions section of the Properties dialog box, and click Delete. A confirmation dialog box is displayed to prevent accidental deletions.

Figure 7-2 Delete Action Dialog Box



Delete Action

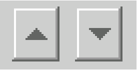
Are you sure you want to delete this action?

When an action is deleted, all references to the action are removed. The action is removed from all instances of the workflow, including those currently running.

Changing the Sequence of Actions

Actions are displayed in the sequence that they will be executed. To re sequence actions, select an action from the list in the Properties dialog box and press the up or down arrow to move its position in the list.

Figure 7-3 Up and Down Arrows



Using Task Actions

This section defines and describes how to use the following task actions to specify and control tasks:

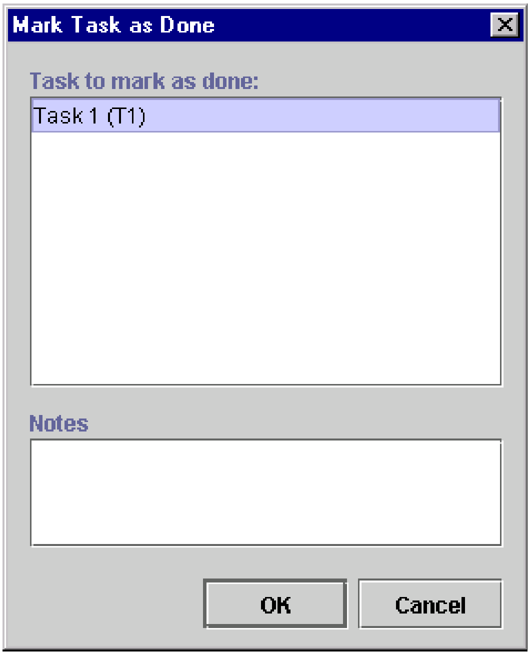
- Mark Task as Done
- Unmark Task Done
- Set Task Due Date
- Assign Task to User
- Assign Task to Role
- Assign Task Using Routing Table
- Unassign Task
- Set Task Comment
- Set Task Priority
- Execute Task

Mark Task as Done

Use this action to explicitly mark a task as done. This action activates the MarkedDone state of a task., and performs all the actions listed in the MarkedDone tab of the Task Properties dialog box. For details about task states, see "Defining Tasks" on page 6-21.

When a task is marked as done, the task is no longer listed as an Active task in the Worklist application. By marking a task done, the task is prevented from being executed again, unless the task properties allow it. When a task is marked as done, all successor nodes are activated (as defined in the Task Properties dialog box).

Figure 7-4 Mark Task as Done Dialog Box


Mark Task as Done

Task to mark as done:

Task 1 (T1)

View or enter information in the Mark Task as Done dialog box.

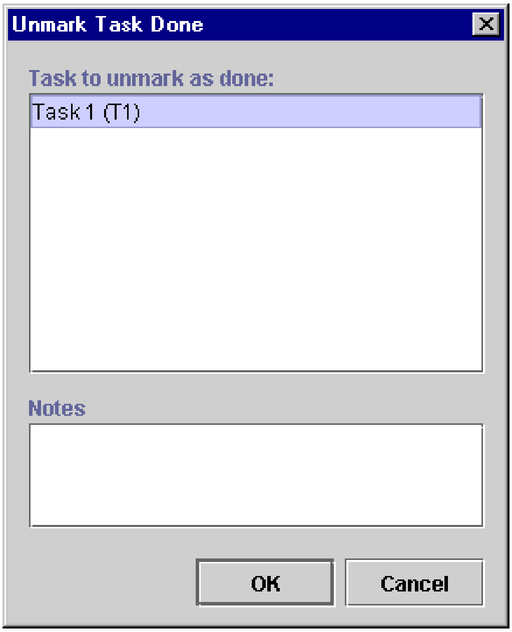
- Task to Mark as Done — displays all tasks in the workflow. Select the task that is to be marked done when the action is executed. Only one task at a time can be marked done.
- Notes — a free-form text box for notes on the action.

Unmark Task Done

This action marks the task as not done. Use this action when a condition arises that prevents the task from continuing in a successfully completed state. A task that is assigned this action becomes Active on the Worklist and can be executed again.

Note: A task that is marked not done does not have its activated state triggered again. To trigger the activated state for the task, the workflow must initiate the task again. For details about task states, see "Defining Tasks" on page 6-21.

Figure 7-5 Unmark Task Done Dialog Box



Unmark Task Done

Task to unmark as done:

Task 1 (T1)

Notes

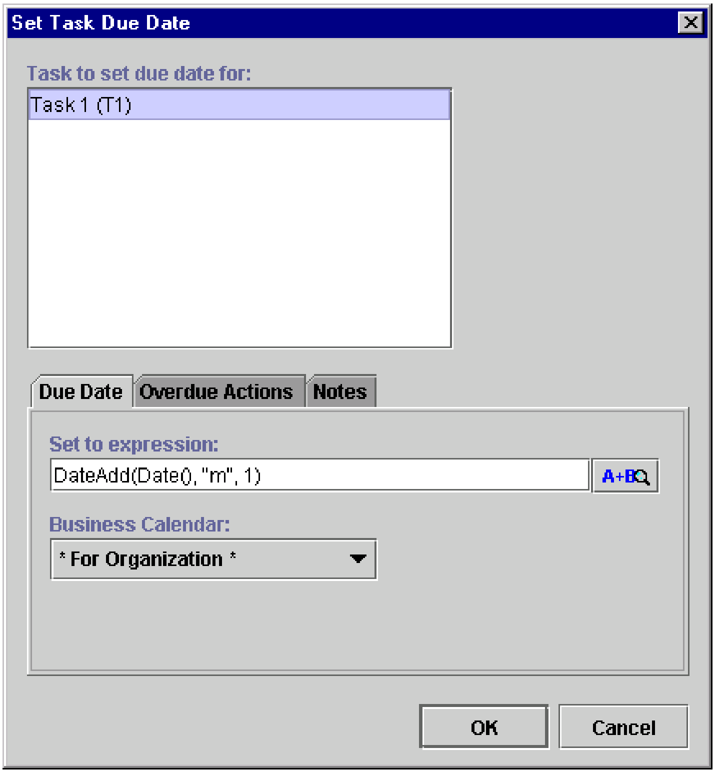
View or enter information in the Unmark Task Done dialog box.

- Task to Unmark as Done — displays all the tasks in the workflow. Select the task that is to be unmarked as done when the action is executed. Only one task at a time can be selected.
- Notes — a free-form text box for notes about the action.

Set Task Due Date

This action sets the due date for a specified task by the task must be executed. You can express the due date in minutes, hours, days, weeks, or months, or it can be the result of an expression. You can also specify subactions that are performed when the task becomes overdue.

Figure 7-6 Set Task Due Date Dialog Box



Set Task Due Date

Task to set due date for:

Task 1 (T1)

Set to expression:

DateAdd(Date(), "m", 1)

Business Calendar:

* For Organization *

View or enter information in the Set Task Due Date dialog box:

- Task to set due date for — displays all the tasks in the workflow. Select the task that is to have its due date set when the action is executed. Only one task at a time can be selected.
- Due Date — enter the due date for the task. The due date can be based on the evaluation of an expression. The due date for the task is displayed in the Worklist application.
  - Set to expression — set the due date based on an expression. In the expression, use the DateAdd function with business hours and business days.

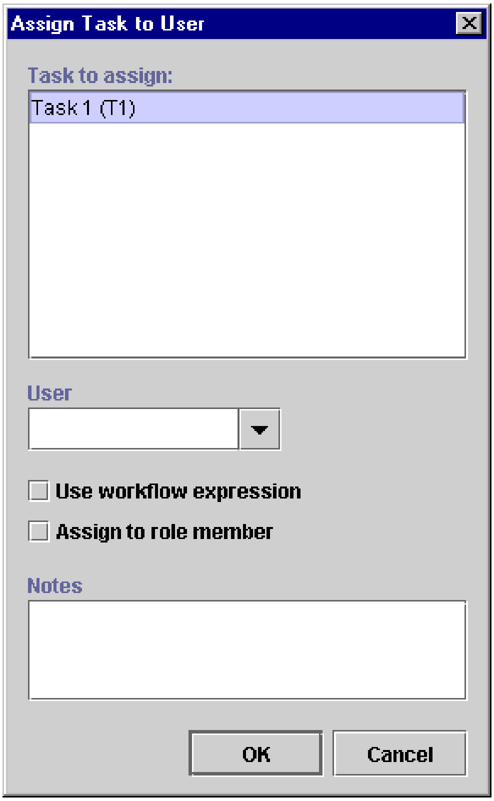
  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

  - Business Calendar — if the due date is set using business hours or business days, a business calendar can be specified. This calendar defaults to the calendar specified for the organization, but can be changed to any other calendar. The specified calendar will be used in calculating the actual due time. Click the Business Calendar drop-down list to display a list of defined business calendars along with two additional choices: For Organization and For Assignee.
- Overdue Actions — a set of subactions to be performed when the due date is reached. This is optional.
- Notes — a free-form text box for notes about the action.

Assign Task to User

Use this action to assign a specified task to an individual user or to a user in a role. The user or role can be specified as a constant or taken from a workflow variable at run time. In the case of a user in a role, the system attempts to perform simple workload balancing in its selection of the assignee; the system determines which user defined to the role has the least number of pending tasks, then assigns the task to that user. Assigning a task to a user causes a notification to be sent to the user. The user views the task from the Worklist application.

Figure 7-7 Assign Task to User Dialog Box



Task to assign: Task 1 (T1)

User

□ Use workflow expression

□ Assign to role member

Notes

View or enter information in the Assign Task to User dialog box.

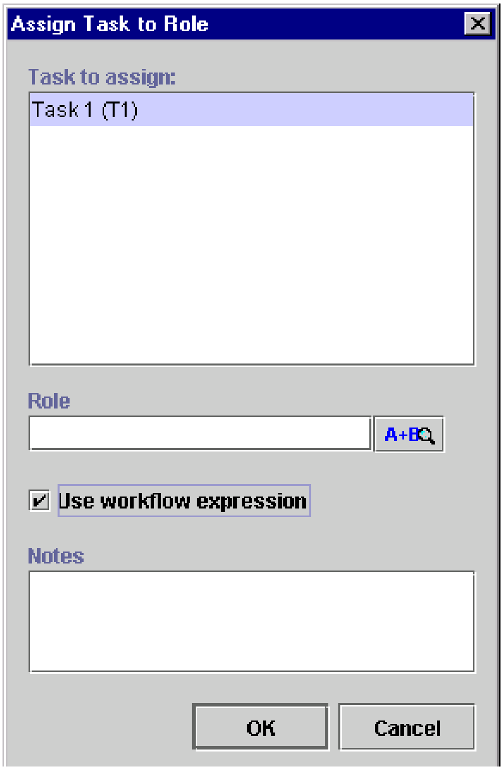
- Task to assign — displays all tasks in the workflow. Select the task that is to be assigned when the action is executed. Only one task at a time can be selected.
- User — drop-down list from which you select the user to assign to, the role from which a user is to be chosen, or an expression to reference the user or role, depending on the other dialog settings.
- Use workflow expression — Indicates whether the new assignee name is a constant or is taken from the value of a workflow expression at run time. If selected, a workflow expression must be constructed.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."
- Assign to role member — indicates whether the User control specifies a user or a role. If checked, the User drop-down list shows a list of roles. When the action is executed, the system attempts to perform workload balancing by assigning the task to the role member who has the least number of assigned tasks at that time. If not checked, the User drop-down list shows a list of individual users.
- Notes — a free-form text box for notes about the action.

Assign Task to Role

Use this action to assign a task to a role, either directly to a constant role or by taking the value of a workflow expression at run time. It is not always desirable to assign a task to a specific user. When you assign a task to a role, any user that is defined as a member of the role has access to the worklist of the role. When a task is assigned to a role, there is no workflow notification of a new task. Assign a task to a role when the task can be performed by anyone who is a member of that role and has a certain skill level.

Figure 7-8 Assign Task to Role Dialog Box



Assign Task to Role

Task to assign:

Task 1 (T1)

Role

☑ Use workflow expression

Notes

View or enter information in the Assign Task to Role dialog box.

- Task to assign — displays all tasks in the workflow. Select the task that is to be assigned when the action is executed. Only one task at a time can be selected.
- Role — from the drop-down list, select the role to assign to the task or a workflow expression reference to the role, depending on the Use workflow expression setting.
- Use workflow expression — indicates whether the role is a constant or is taken from the value of a workflow expression at run time. If selected, a workflow expression must be constructed, and the A+B button appears to the right of Role.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

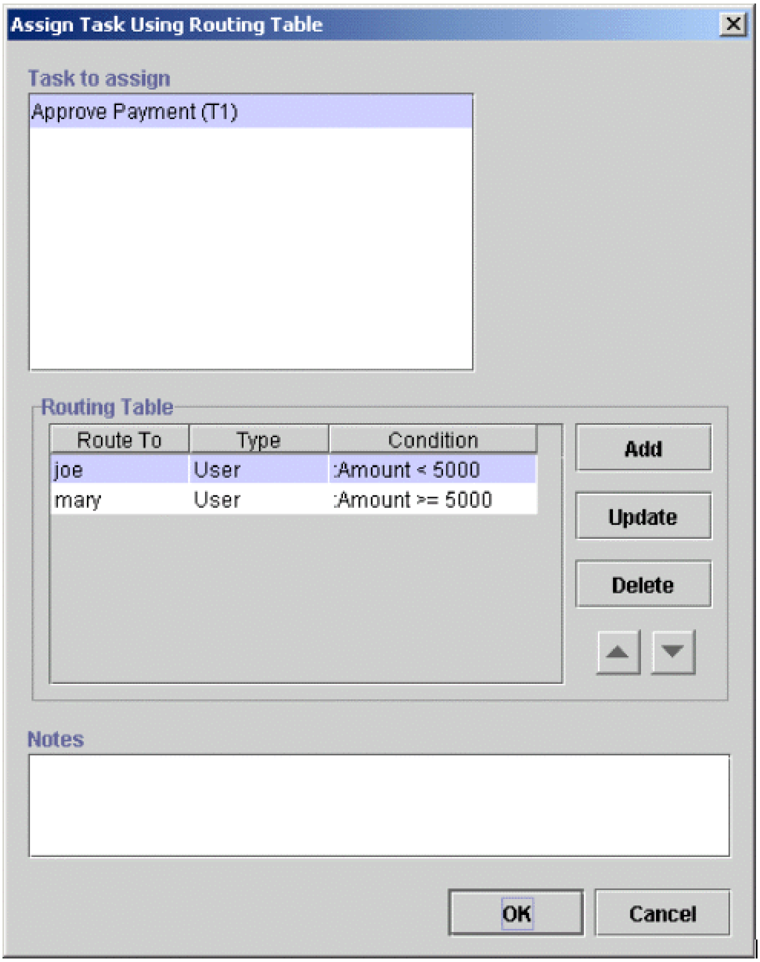
- Notes — free-form text box for notes about the action.

Assign Task Using Routing Table

Assign a task to a user or role depending on a set of conditions. A routing table consists of a sequence of one or more routing conditions. A routing condition specifies a potential assignee (user or role) for the task, together with a specification of the conditions under which the assignment should be made. When the action is executed at run time, the system evaluates each condition in sequence until it encounters one that yields a True result. If such a condition is encountered, the task is assigned to the corresponding user or role, and any subsequent conditions are ignored.

Note: If no routing condition is satisfied, a run-time exception occurs.

Figure 7-9 Assign Task Using Routing Table Dialog Box



Task to assign: Approve Payment (T1)

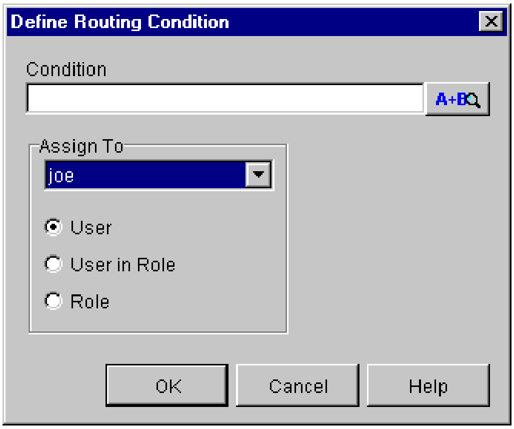
Route To	Type	Condition
joe	User	:Amount < 5000
mary	User	:Amount >= 5000

View or enter information in the Assign Task Using Routing Table dialog box.

- Task to Assign — displays all tasks in the workflow. Select the task that is to be assigned when the action is executed. Only one task at a time can be selected.
- Routing Table — table of conditions and assignees. Routing information can be added, updated, or deleted. Additionally, the sequence of conditions can be changed.
- Notes — enter notes about the action.

Clicking Add or Update displays the Define Routing Condition dialog box.

Figure 7-10 Define Routing Condition Dialog Box



Assign To
joe

● User
○ User in Role
○ Role

The Define Routing Condition dialog box contains the following:

- Condition — must be a valid workflow conditional expression that yields a logical result (True or False).

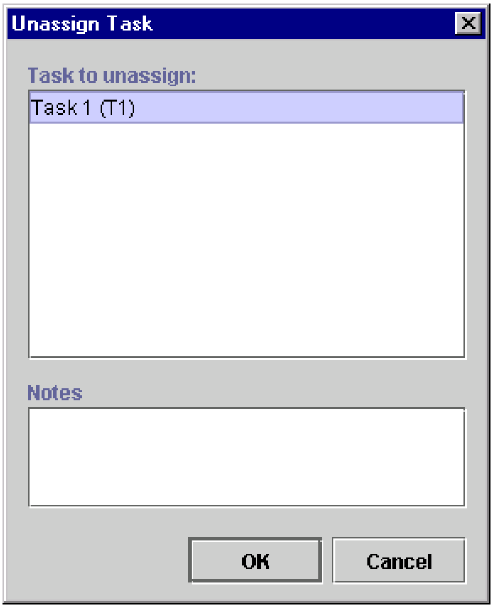
  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

- Assign To — a list showing all possible assignees: User, User in Role, and Role. Select the required assignee from the list.

Unassign Task

Remove the current task assignment. The task is no longer assigned to the user or role.

Figure 7-11 Unassign Task Dialog Box



Task 1 (T1)

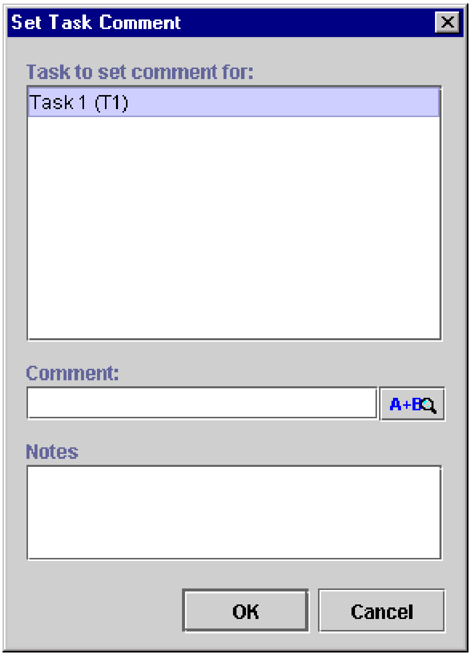
View or enter information in the Unassign Task dialog box.

- Task to unassign — displays all tasks in the workflow. Select the task that is to be unassigned when the action is executed. Only one task at a time can be selected.
- Notes — a free-form text box for notes about the action.

Set Task Comment

Display a free-form text message to the user at run time when the action is executed. The text is typically explanatory or requests the user to perform manual action(s) for which the workflow does not provide further automated support. When the action is executed, the system displays a text message in the comment column of the Worklist application.

Figure 7-12 Set Task Comment Dialog Box



View or enter information in the Set Task Comment dialog box.

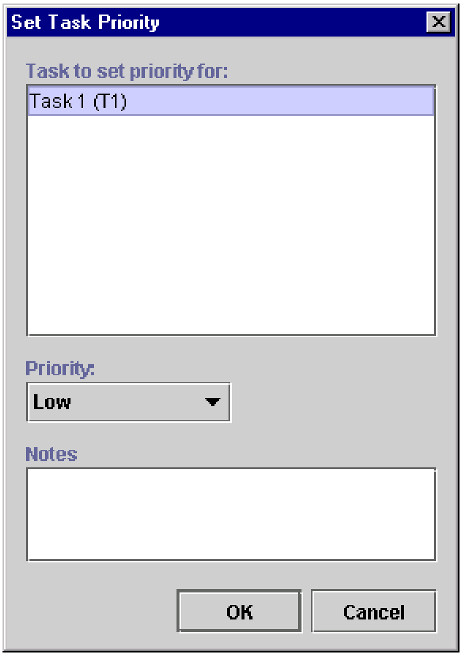
- Task to set comment for — displays all the tasks in the workflow. Select the task that is to have its comment set when the action is executed. Only one task at a time can be selected.
- Comment — a valid workflow expression to be evaluated at run time for the task comment.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”
- Notes — a free-form text box for notes about the action.

Set Task Priority

Set a task priority to Low, Medium, or High.

Figure 7-13 Set Task Priority Dialog Box



Task to set priority for:
Task 1 (T1)

Priority:
Low

Notes

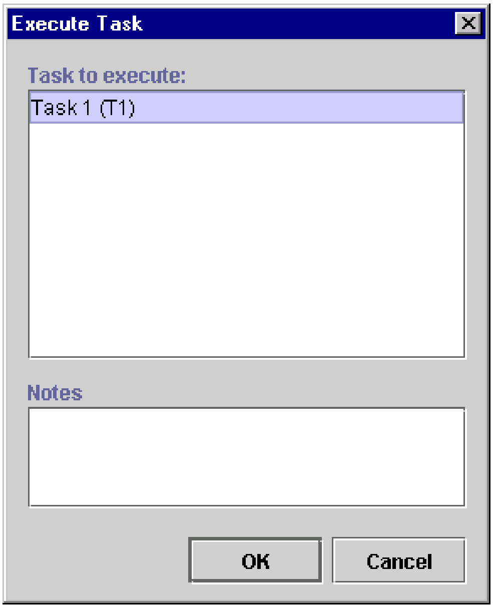
View or enter information in the Set Task Priority dialog box.

- Task to set priority for — displays all tasks in the workflow. Select the task to which you will assign a priority.
- Priority — select the task priority. Choices are Low, Medium, and High.
- Notes — free-form text box for notes about the action.

Execute Task

Use this action to execute a task explicitly. This action activates the executed state of a task, and performs all the actions listed in the Executed tab of the Task Properties dialog box. For details about task states, see “Defining Tasks” on page 6-21.

Figure 7-14 Execute Task Dialog Box



View or enter information in the Execute Task dialog box.

- Task to execute — Displays all tasks in the workflow. Select the task to execute.
- Notes — Free-form text box for notes about the action.

Using Workflow Actions

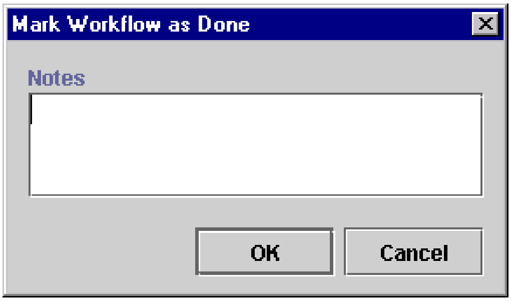
This section defines and describes how to use the following workflow actions to manage workflows:

- Mark Workflow as Done
- Abort Workflow
- Start Workflow
- Set Workflow Variable
- Set Workflow Comment

Mark Workflow as Done

This action marks the workflow as done and serves the same purpose as a workflow reaching a Done node.

Figure 7-15 Mark Workflow as Done

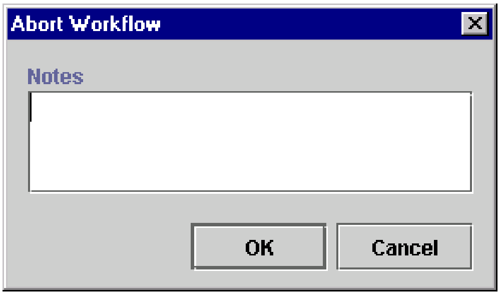


Enter notes about the action in the free-form text box, and click OK.

Abort Workflow

Abort the current workflow. Abort Workflow permanently stops a workflow that is currently in process. This action enables the system to handle a situation in which there is no reason to continue a given workflow. The BEA WebLogic Process Integrator Workflow Monitoring facility differentiates between a workflow that has aborted upon the execution of this action and a workflow that has completed by reaching a Done node. A workflow that is aborted will be deleted from the server.

Figure 7-16 Abort Workflow Dialog Box



View or enter information in the Abort Workflow dialog box.

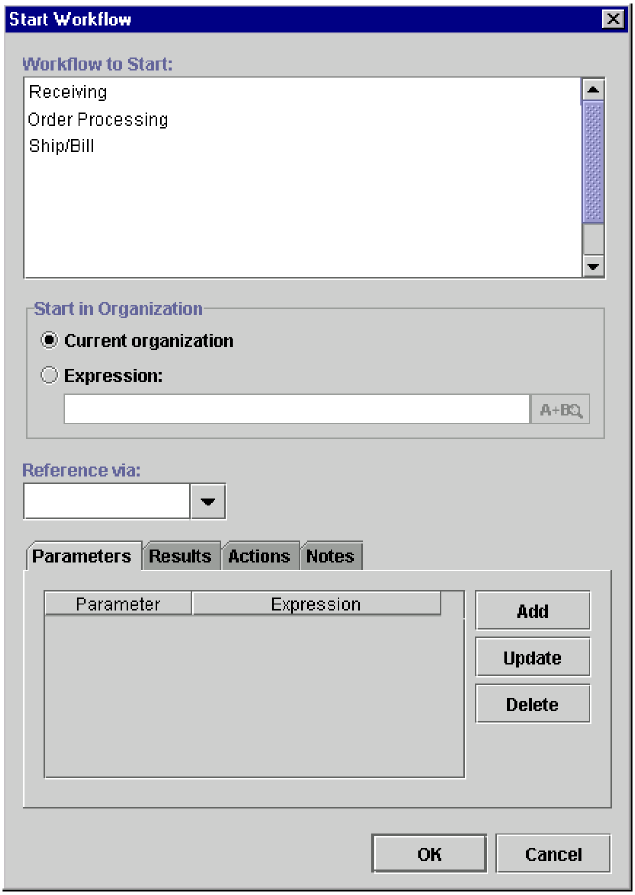
- Notes — a free-form text box for notes about the workflow abortion.

Start Workflow

Use this action to start another workflow from the current one. The workflow being started is sometimes referred to as a *subworkflow* or a *called* workflow. The workflow that starts the subworkflow is called the *calling* workflow.

The started workflow can be tracked with a variable reference, and when it completes, you can perform an optional set of subactions.

Figure 7-17 Start Workflow Dialog Box


Start Workflow

Workflow to Start:

Receiving

Order Processing

Ship/Bill

Start in Organization

● Current organization

○ Expression:

Reference via:

View or enter information in the Start Workflow dialog box.

- Workflow to Start — displays all workflow templates defined in the system. Select the workflow template to start when the action is executed. Only one workflow template definition within that workflow template can be started at a time, this being the most effective of the active workflow template definitions.
- Start in Organization — the organization for which the specified workflow should be triggered and executed.
    - Current organization — starts the workflow for the current organization (the organization selected from the Organization drop-down list on the BEA WebLogic Process Integrator main window).
    - Expression — starts the workflow for an organization taken from the value of a workflow expression. If selected, a workflow expression must be constructed.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- Reference via — list of workflow variables available for the current workflow. When you select a workflow, the instance ID of the called subworkflow is assigned automatically to the selected variable. The instance ID is a unique identifier for the called subworkflow.

  The calling workflow can then use the instance ID in expressions using the WorkflowVariable function to retrieve variables from the called subworkflow. For details about using the instance ID in the WorkflowVariable function, see “Functions” on page 8-6.

Use the following tabs to select additional information for the action:

- Parameters — displays variables that were defined as input parameters. (See “Defining Variables” on page 6-3.) The calling workflow can set the initial values for these parameters, and pass them as input to the called subworkflow.

- Results — displays variables that were defined as output parameters (See "Defining Variables" on page 6-3.) The called subworkflow can put result values in the parameters, and return them as output to the calling workflow.
- Actions — specify optional subactions to be performed when the called workflow is completed. It can be used to mark the current task as done.
- Notes — a free-form text box for notes about the action.

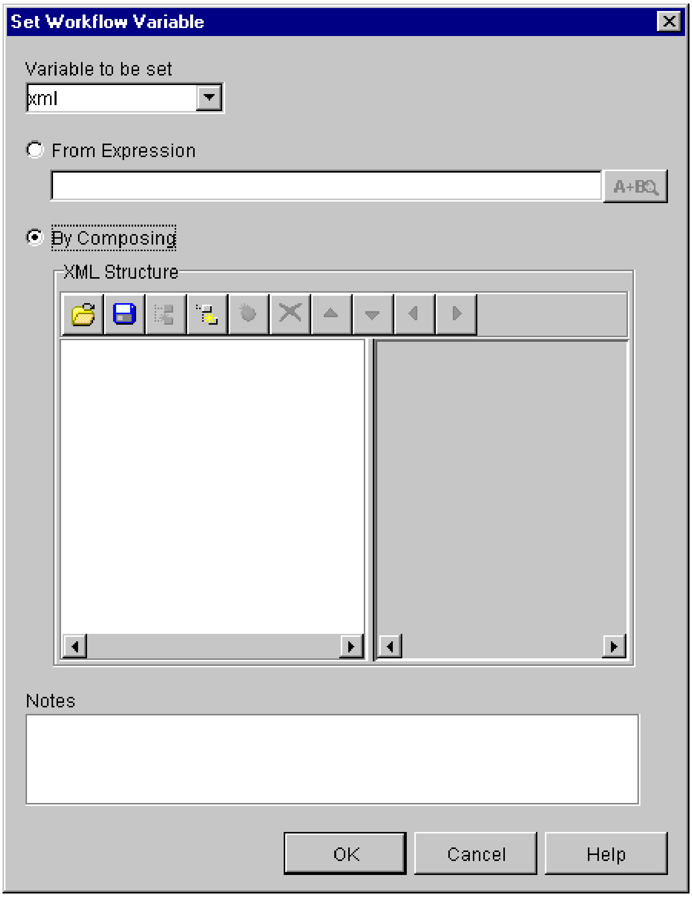
Set Workflow Variable

Use this action to assign a value to an existing workflow variable you defined using the Variable Properties dialog box. For details about defining variables, see "Defining Variables" on page 6-3.

The value you assign to the variable can be an expression or a literal. This action evaluates the expression or literal, and assigns the result to a workflow variable when the action is executed.

If you assign a value to an XML type variable, you can use this action to compose an XML document and store the content in the XML type variable. You can then send the XML type variable as an internal or external XML message using the Post XML Event action (see "Post XML Event" on page 7-32).

Figure 7-18 Set Workflow Variable Dialog Box



Variable to be set: xml

The Set Workflow Variable dialog box contains the following:

- Variable to be set — choose an existing, defined variable to which you will assign a value, or type one in. (For details, see “Defining Variables” on page 6-3.)
- From Expression — set the value for the variable by constructing an expression or by entering a literal in double quotes (for example, a string type variable).

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- By Composing — set the value for an XML or string type variable by composing an XML document containing elements and attributes.

  Note: An XML document that is stored in a string type variable will first be converted to a string.

  To compose an XML document, perform the actions listed in the following table. The XML document you define is stored in the workflow template definition.

To . . .	Perform This Action
Retrieve an existing XML document for editing	Click the Import button to open the XML Finder dialog box. Use the XML Finder to retrieve an existing XML document that you can edit. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.
Add the first element of the XML message	Click the Add Child button.
Change the default element name	Click the element and wait, or triple-click the element. The element name becomes an entry field. Change the element name as appropriate, then press Enter. To cancel the change, press Esc.

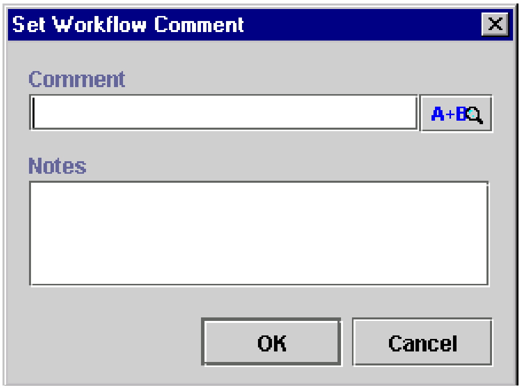
To . . .	**Perform This Action**
Add a sub-element to an element	Select the element, then click the Add Child button.
Add an element at the same level as another element	Select the element, then click the Add Sibling button.
Add an attribute to an element	Select the element, then click the Add Attribute button.
Change the default attribute name	Select the attribute, then double-click the attribute. The attribute name becomes an entry field. Change the attribute name as appropriate, then press Enter. To cancel the change, press Esc.
Change the order of the elements	Select the element you want to move, and use the arrow keys as follows: ■ To move the element up or down a level, click the Move Left or Move Right arrow. ■ To move the element up or down within a level, click the Move Up or Move Down arrow
Change the content of a leaf element or attribute	In the right panel, double-click the line next to the element or attribute for which you want to change the content. The line becomes an entry field. Type the content. To cancel the change, press Esc. **Note:** The syntax for the content of a leaf element or an attribute must follow the same rules that are used for workflow expressions. For example, string literals must be enclosed in single or double quotes. If the syntax is incorrect, a red X appears next to the entry. For details about defining workflow expressions, see Chapter 8, “Using Expressions and Conditions.”
Delete an element or attribute	In the left panel, select the element or attribute you want to delete, and click the Delete button.
Save the XML message	Click the Export button to open the XML Finder dialog box. Use the XML Finder to save the XML message. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.

- Notes — a free-form text box for notes about the action.

Set Workflow Comment

Set the comment for a workflow instance. Use this action to specify a valid workflow expression to be displayed to the user at run time when the action is executed. The text is typically informative or requests the user to perform manual action(s) for which the workflow does not provide further automated support.

Figure 7-19 Set Workflow Comment Dialog Box



View or enter information in the Set Workflow Comment dialog box.

- Comment — a valid workflow expression, evaluated at run time.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- Notes — a free-form text box for notes about the workflow comment.

Using Integration Actions

To integrate a workflow with software components (such as EJBs, Java classes, and executable programs), external applications (such as the Worklist client application or an executable program), and WebLogic Server functionality (such as the XSL transformation engine), BEA WebLogic Process Integrator provides the following actions:

- Post XML Event
- Send XML to Client
- Call Program
- Perform Business Operation
- XSL Transform

The following sections describe how to use these actions.

Post XML Event

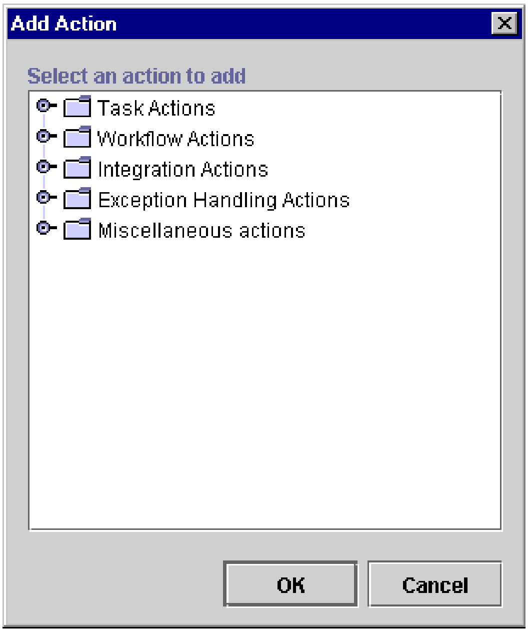
This action sends an XML message to a specified destination to trigger an event. This action can trigger two kinds of events:

- An internal event in BEA WebLogic Process Integrator. As an internal event, the Post XML Event action can:
  - Trigger an Event node in the current workflow or another one. For details about triggering events, see “Defining Events” on page 6-24.
  - Start another workflow for which the Start node is triggered by an event. For details about event triggers in Start nodes, see “Defining a Start Node Using an Event Trigger” on page 6-14.
- An external event in a JMS topic or JMS queue. As an external event, the Post XML Event can communicate with an external application that subscribes to the JMS topic you specify, or with a specific application that is the JMS queue receiver.

To post an XML event:

1. Access the Add Action dialog box. This dialog box can be accessed from the various Properties dialog boxes in BEA WebLogic Process Integrator. For example, in the workflow drawing area, right-click a task and choose Properties. The Task Properties dialog box is displayed.
2. In the Task Properties dialog box, click Add to display the Add Action dialog box.

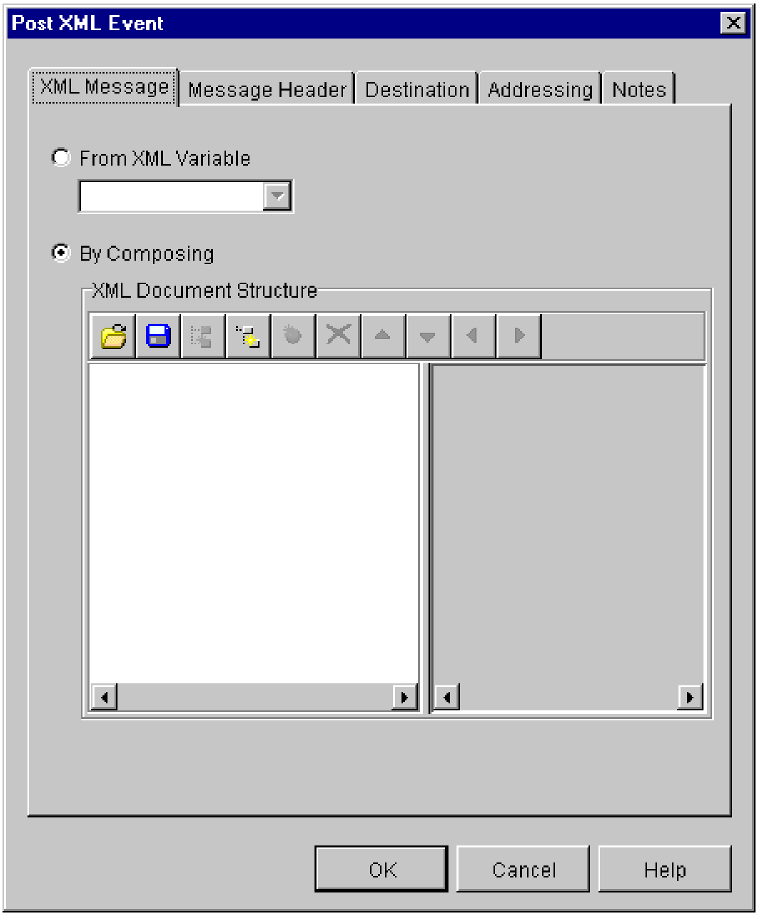
Figure 7-20 Add Action Dialog Box



Note: If a plug-in is defined for actions, this dialog box contains the new action. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. Under Integration Actions, select Post XML Event and click OK to display the Post XML Event dialog box.

Figure 7-21 Post XML Event Dialog Box



4. Select the XML Message tab to define the XML message you want this action to send. The XML message you define is stored in the workflow template definition.

   If the XML document is stored in an XML type variable, select the variable from the From XML Variable drop-down list. For details about XML type variables, see “Understanding Variables” on page 6-3.

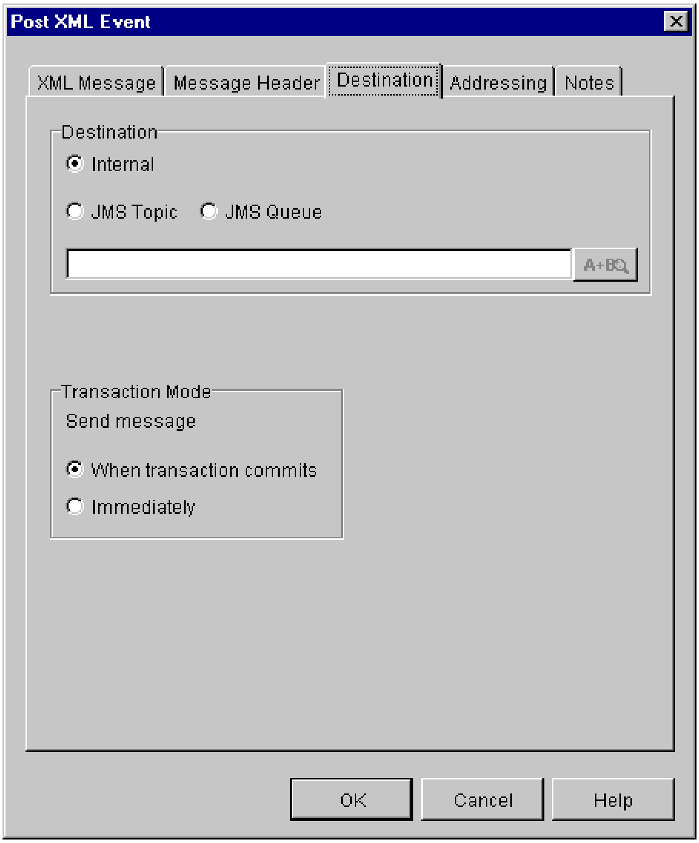
If you want to compose the XML message, you can do so by performing the following actions:

To . . .	**Perform This Action**
Retrieve an existing XML document for editing	Click the Import button to open the XML Finder dialog box. Use the XML Finder to retrieve an existing XML document that you can edit. For details about the XML Finder, see "Locating XML Entities Using the XML Finder" on page C-2.
Add the first element of the XML message	Click the Add Child button.
Change the default element name	Click the element and wait, or triple-click the element. The element name becomes an entry field. Change the element name as appropriate, then press Enter. To cancel the change, press Esc.
Add a sub-element to an element	Select the element, then click the Add Child button.
Add an element at the same level as another element	Select the element, then click the Add Sibling button.
Add an attribute to an element	Select the element, then click the Add Attribute button.
Change the default attribute name	Select the attribute, then double-click the attribute. The attribute name becomes an entry field. Change the attribute name as appropriate, then press Enter. To cancel the change, press Esc.
Change the order of the elements	Select the element you want to move, and use the arrow keys as follows: ■ To move the element up or down a level, click the Move Left or Move Right arrow. ■ To move the element up or down within a level, click the Move Up or Move Down arrow

To . . .	**Perform This Action**
Change the content of a leaf element or attribute	In the right panel, double-click the line next to the element or attribute for which you want to change the content. The line becomes an entry field. Type the content. To cancel the change, press Esc. **Note:** The syntax for the content of a leaf element or an attribute must follow the same rules that are used for workflow expressions. For example, string literals must be enclosed in single or double quotes. If the syntax is incorrect, a red X appears next to the entry. For details about defining workflow expressions, see Chapter 8, "Using Expressions and Conditions."
Delete an element or attribute	In the left panel, select the element or attribute you want to delete, and click the Delete button.
Save the XML message	Click the Export button to open the XML Finder dialog box. Use the XML Finder to save the XML message. For details about the XML Finder, see "Locating XML Entities Using the XML Finder" on page C-2.

5. Select the Destination tab to specify where you want to post the XML message.

Figure 7-22 Destination Tab



Destination

● Internal

○ JMS Topic ○ JMS Queue

Transaction Mode

Send message

● When transaction commits

○ Immediately

If the destination is internal, the XML message is sent to an internal JMS queue maintained by BEA WebLogic Process Integrator. The JNDI name of the internal JMS queue is com.bea.wlpi.EventQueue.

If the destination is external, specify the JNDI name of the JMS topic or JMS queue to which you want to post the message.

For an external destination, you can enter the JNDI name of the JMS topic or JMS queue directly into the expression field, or you can define an expression that is evaluated at run time to produce the JNDI name.

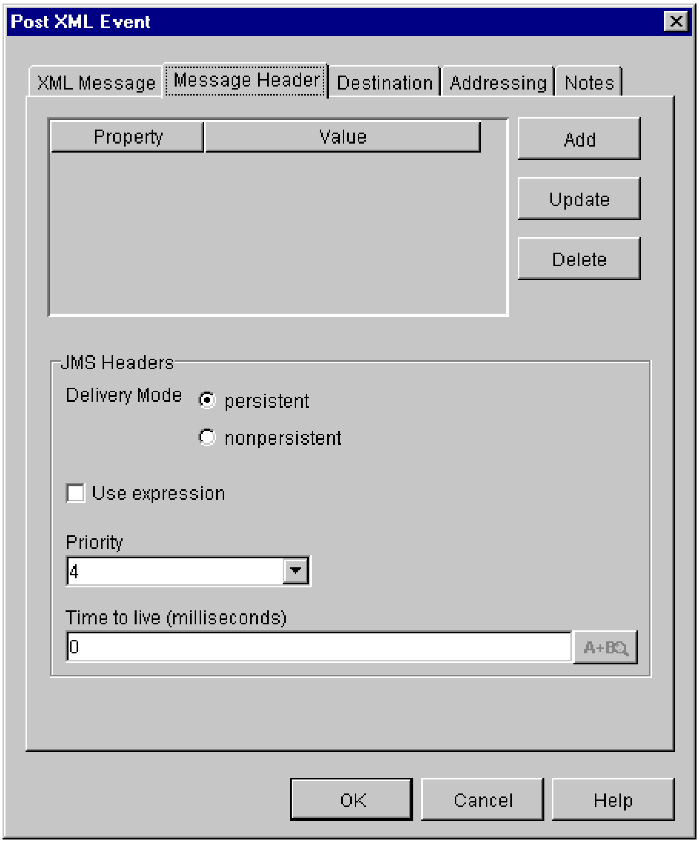
Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

If the XML message is part of a transaction, you can send the message immediately, or only if the transaction completes successfully. Select one of the following:

- When transaction commits — ensures that the message is sent only if the transaction completes successfully and a commit is issued. If the transaction is unsuccessful and is rolled back, the message is not sent.
- Immediately — sends the message immediately without waiting for the transaction to complete.

6. Optionally, select the Message Header tab to specify any JMS message properties you want to add to the message, and to set values for standard JMS headers.

Figure 7-23 Message Header Tab



Delivery Mode ● persistent ○ nonpersistent

□ Use expression

Priority: 4

Time to live (milliseconds): 0

JMS messages contain a standard set of header fields that are always transmitted with the message. JMS messages also include property fields in which you can specify application-specific information.

For example, you can use the property field to specify information in the message that is not appropriate for the body of the message. The receiving workflow then uses the EventAttribute function in an event key expression to retrieve the information you specified in the property field. For details about the EventAttribute function, see “Using Expression Components” on page 8-6.

JMS message properties are name-value pairs. The name can be almost any string that is a valid identifier in the Java language. The value can be any one of the following types: boolean, byte, short, int, long, float, double, or string.

To define JMS message properties, click Add to display the Set JMS Properties dialog box. In the dialog box, specify the following:

a. In the JMS Property field, specify the property name.

b. In the JMS Property Value field, enter the property value. Enter the value directly or enter an expression that is evaluated at run time to produce the value.

c. Click OK.

You can also specify values for the following standard JMS headers:

- Delivery Mode — specifies one of two modes: persistent and nonpersistent.

  Persistent messages are not lost even if the JMS server fails. The message is delivered again after the server recovers.

  Nonpersistent messages can be lost if the JMS server fails. The message is not delivered again after the server recovers.

- Priority — assigns a priority level to the message. Levels 0-4 are normal priority. Levels 5-9 are expedited priority. Messages with an expedited priority are delivered ahead of messages with a normal priority.

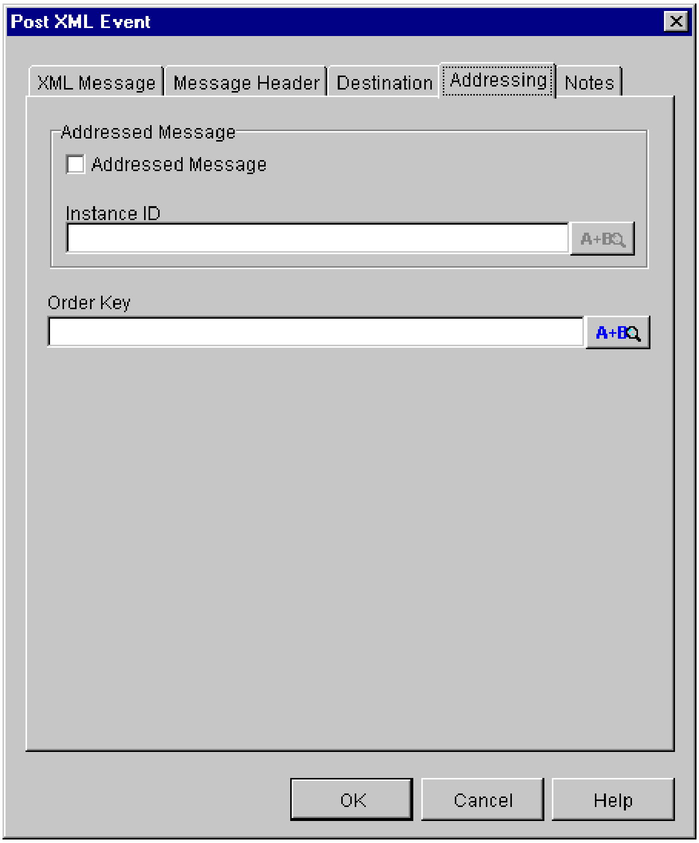
  You can select a value from the drop-down list, or you can define an expression that is evaluated at run time to produce the value. To use the Expression Builder to define an expression, select the Use Expression check box first.

- Time to live (milliseconds) — specifies how long the message is available before it expires. For example, if you want the message to be available for 1 hour, you specify a value of 3600000 (1000 milliseconds x 60 seconds x 60 minutes). The default value of 0 (zero) indicates that the message will not expire.

  You can enter the value directly, or you can define an expression that is evaluated at run time to produce the value. To use the Expression Builder to define an expression, select the Use Expression check box first.

7. Optionally, select the Addressing tab to indicate that you want to persist an XML message for one or more workflow instances, and to specify an order key.

Figure 7-24 Addressing Tab



Specify the following:

- Addressed message — indicates that you want to persist a message for one or more workflow instances. If a workflow instance with the ID you specify is not activated when you send the message, BEA WebLogic Process Integrator persists the message until a workflow that matches the ID is activated.

Note: If you specify a Time to live in the Message Header area, the message persists only for the time you specified.

- Instance ID — enter the ID of the workflow for which you want to persist the message, or enter an expression that is evaluated at run time to produce the ID.

If you want to persist the message for more than one workflow instance, separate the IDs with commas.

- Order key — guarantees that if more than one event is posted at the same time, the events are processed by the same event listener in the order in which they are received.

  An order key must be an integer value, and the value must be the same for each event that you want processed in the order in which it is received. For example, if two events are posted at the same time, and you want them processed in the order in which they are received, enter the same order key value, such as the integer value of 8, for each event.

  Enter the order key directly, or enter an expression that is evaluated at run time to produce the order key.

8. Select the Notes tab to enter any notes about this action.

9. Click OK.

Send XML to Client

Use this action to communicate between a workflow and a client application by sending an XML document to the client. The client application must be programmed to identify the XML document, perform the appropriate action, and return an XML document in response to the workflow.

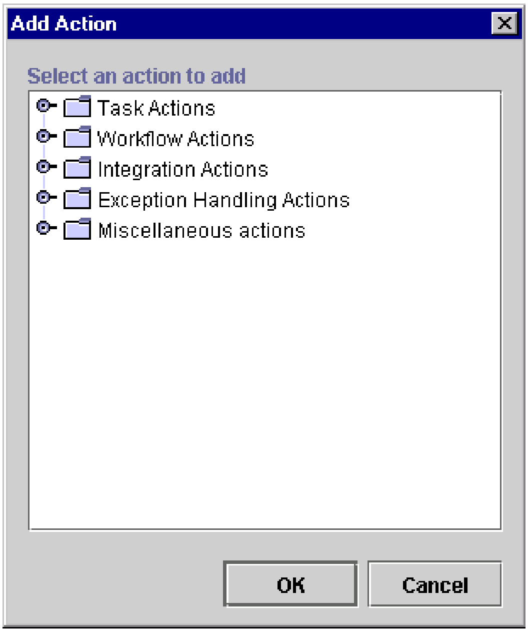
For examples of using this action with the Worklist client application, see “Using the Send XML to Client Action with the Worklist Application” on page 7-47.

To send an XML document to a client:

1. Access the Add Action dialog box. This dialog box can be accessed from the various Properties dialog boxes in BEA WebLogic Process Integrator. For example, in the workflow drawing area, right-click a task and choose Properties. This displays the Task Properties dialog box.

2. In the Task Properties dialog box, click Add to display the Add Action dialog box.

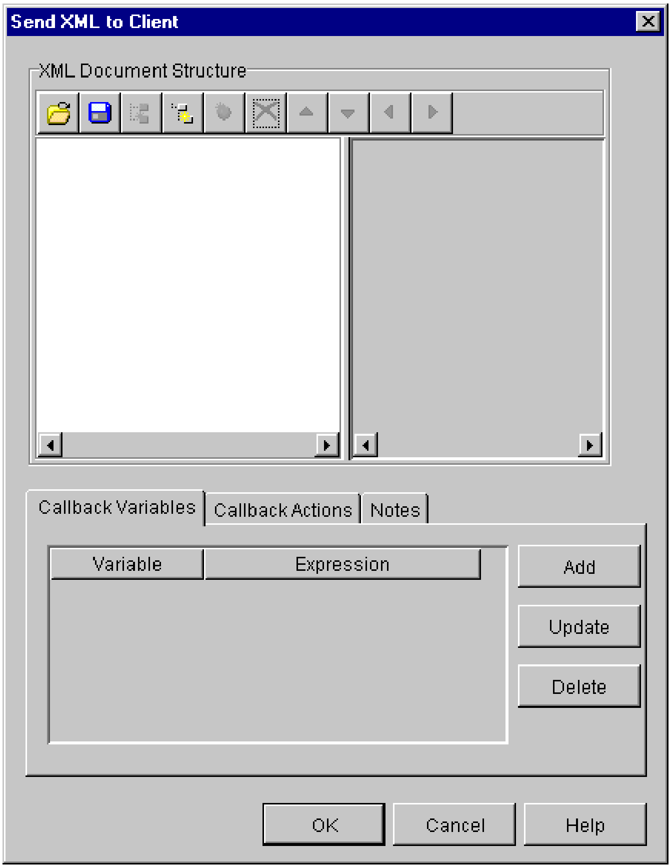
Figure 7-25 Add Action Dialog Box



Note: If a plug-in is defined for actions, this dialog box contains the new action. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. Under Integration Actions, select Send XML to Client, and click OK to display the Send XML to Client dialog box.

Figure 7-26 Send XML to Client Dialog Box



4. Define the XML document you want to send by performing the actions listed in the following table. The XML document you define is stored in the workflow template definition.

To . . .	**Perform This Action**
Retrieve an existing XML document for editing	Click the Import button to open the XML Finder dialog box. Use the XML Finder to retrieve an existing XML document that you can edit. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.
Add the first element of the XML message	Click the Add Child button
Change the default element name	Click the element and wait, or triple-click the element. The element name becomes an entry field. Change the element name as appropriate, then press Enter. To cancel the change, press Esc.
Add a sub-element to an element	Select the element, then click the Add Child button.
Add an element at the same level as another element	Select the element, then click the Add Sibling button.
Add an attribute to an element	Select the element, then click the Add Attribute button.
Change the default attribute name	Select the attribute, then double-click the attribute. The attribute name becomes an entry field. Change the attribute name as appropriate, then press Enter. To cancel the change, press Esc.
Change the order of the elements	Select the element you want to move, and use the arrow keys as follows: ■ To move the element up or down a level, click the Move Left or Move Right arrow. ■ To move the element up or down within a level, click the Move Up or Move Down arrow

To . . .	Perform This Action
Change the content of a leaf element or attribute	In the right panel, double-click the line next to the element or attribute for which you want to change the content. The line becomes an entry field. Type the content. To cancel the change, press Esc. **Note:** The syntax for the content of a leaf element or an attribute must follow the same rules that are used for workflow expressions. For example, string literals must be enclosed in single or double quotes. If the syntax is incorrect, a red X appears next to the entry. For details about defining workflow expressions, see Chapter 8, “Using Expressions and Conditions.”
Delete an element or attribute	In the left panel, select the element or attribute you want to delete, and click the Delete button.
Save the XML message	Click the Export button to open the XML Finder dialog box. Use the XML Finder to save the XML message. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.

Note: The system-generated `actionid` element and its value are used at run time to identify the Send XML to Client action. The `actionid` element must be the first child element of the root element. Do not move, delete or edit the `actionid` element, and do not add any subelements to it.

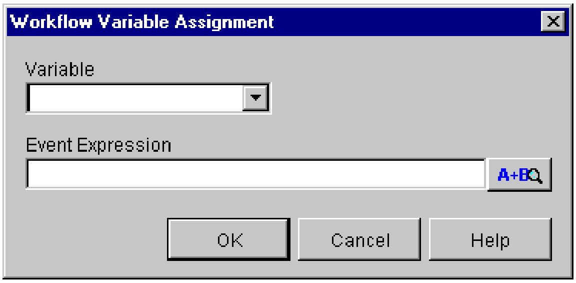
If you change, move, or delete the `actionid` element, a warning message is displayed when you click OK in the Send XML to Client dialog box. The Studio then resets the `actionid` element and its value to the system-generated value and location.

5. Use the Callback Variables tab to specify the variables that will receive the response from the client. Specify a variable for each anticipated response from the client.

   Clicking Add displays the Workflow Variable Assignment dialog box. The variables in the pull-down list are those defined in the Variables Properties dialog box. (For details, see “Defining Variables” on page 6-3.) The XML expression can be the name of an element in the XML message returned by the client the value of which you want to assign to the selected variable, or it is an

XPath function that extracts a value from the XML message from the client, and then assigns that value to the selected variable.

Figure 7-27 Workflow Variable Assignment Dialog Box



6. Use the Callback Actions tab to specify any actions that should be performed when the reply is received from the client.

   Clicking Add opens the Add Action dialog box where you can select the actions you want to specify. Typically, you would specify the Mark Task as Done action. The Mark Task as Done action indicates that the client has returned a response, and control can return to the Send XML to Client action.

7. Use the Notes area to add free-form text notes about this action.

8. Click OK.

Using the Send XML to Client Action with the Worklist Application

The Worklist client application is designed to perform some action when it receives an XML document that conforms to a predefined DTD file. The predefined DTD files are shipped with BEA WebLogic Process Integrator, and are in the following directory where BEA WebLogic Process Integrator is installed:

```
...docs\javadocs\com\bea\wlpi\common\doc-files
```

There are four pairs of predefined DTD files. Each pair has one DTD file for a request to the Worklist client, and another DTD file for a response from the Worklist client. The root element of the DTD file indicates the action the Worklist client performs when it receives a request XML document.

The following table lists the predefined DTD files, and the actions performed by the Worklist application when it receives an XML document that conforms to one of the request DTD files.

Table 7-1 Predefined DTD Files

Request DTD	Root Element	Action Performed by Worklist Client	Response DTD
`ClientMsgBoxReq.dtd`	message-box	Displays a message dialog box in which the user responds to a message or query. Reply = ok/yes/no/cancel	`ClientMsgBoxResp.dtd`
`ClientSetVarsReq.dtd`	set-variables	Displays a prompt dialog box with entry fields in which the user enters some values. Reply = name/value pairs	`ClientSetVarsResp.dtd`
`ClientCallPgmReq.dtd`	call-program	Executes a program on the client machine.	`ClientCallPgmResp.dtd`
`ClientCallAddInReq.dtd`	call-addin	Invokes a custom extension to the Worklist client application.	`ClientCallAddInResp.dtd`

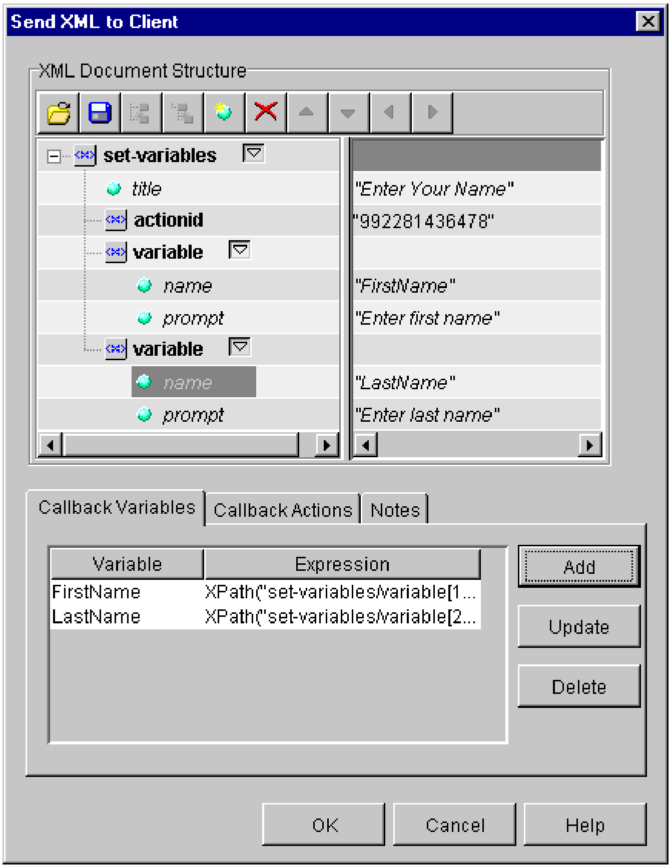
This following sections show you how to define XML documents that cause the Worklist client application to display two types of dialog boxes:

- Prompt dialog box where a user can enter values
- Message dialog box where a user responds to a message

Defining a Prompt Dialog Box

To define a prompt dialog box, the entries in the Send XML to Client dialog box should look like those shown in the following figure.

Figure 7-28 Send XML to Client Dialog Box



set-variables	
title	*"Enter Your Name"*
actionid	"992281436478"
variable	
name	*"FirstName"*
prompt	*"Enter first name"*
variable	
name	*"LastName"*
prompt	*"Enter last name"*

Variable	Expression
FirstName	XPath("set-variables/variable[1...
LastName	XPath("set-variables/variable[2...

The XML document structure shown in the dialog box conforms to the ClientSetVarsReq.dtd file. The contents of the DTD file are shown below.

Listing 7-1 Contents of the ClientSetVarsReq.dtd File

```
<!ELEMENT set-variables (actionid, variable+)>
<!ATTLIST set-variables title CDATA #IMPLIED>
<!ELEMENT actionid (#PCDATA)>
<!ELEMENT variable (#PCDATA)>
<!ATTLIST variable name NMTOKEN #REQUIRED
                   prompt NMTOKEN #IMPLIED>
```

To define the prompt dialog box:

1. Enter the following name for the root element of the XML document: set-variables (all in lowercase).

   Note: The system-generated actionid element and its value are used at run time to identify the Send XML to Client action. The actionid element must be the first child element of the root element. Do not move, delete or edit the actionid element, and do not add any subelements to it.

   If you change, move, or delete the actionid element, a warning message is displayed when you click OK in the Send XML to Client dialog box. The Studio then resets the actionid element and its value to the system-generated value and location.

2. Define title as an attribute of the root element

3. The title attribute takes the title value of the dialog box that is displayed by the Worklist client. Enter the following value for the title attribute: Enter Your Name.

4. Define two variable subelements, each with the following two attributes:

   Name

   Prompt

5. For the first Prompt attribute, specify the following value: Enter first name.

6. For the second Prompt attribute, specify the following value: Enter last name.

7. For the first `Name` attribute, specify a variable that will hold the first name the user enters in the prompt dialog box, such as `FirstName`.

8. For the second `Name` attribute, specify a variable that will hold the last name the user enters in the prompt dialog box, such as `LastName`.

9. To complete the dialog box, you need to specify the Callback variables that will hold the first and last name the user enters, and the XPath statements that will extract the first and last name from the response XML document the Worklist client returns to the workflow. The response XML document conforms to the `ClientSetVarsResp.dtd` file.

   The following table shows the Callback variables and the XPath statements you can use. You can create the XPath statements by typing them manually, or you can use the XPath wizard to create them from the `ClientSetVarsResp.dtd` file.

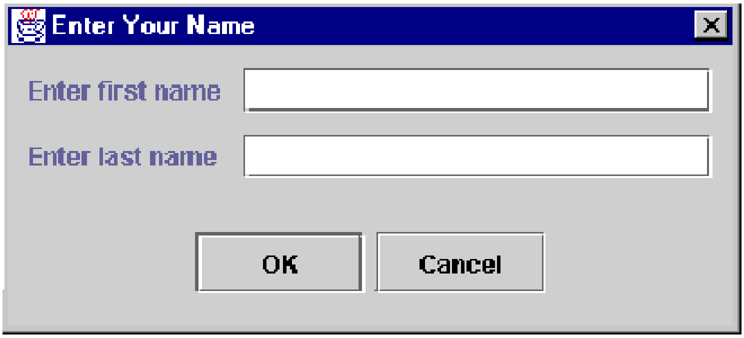
   Note: To access the XPath wizard, click Add in the Send XML to Client dialog box. From the Workflow Variable Assignment dialog box, click the A+B button to open the Expression Builder, then click XPath Wizard. For details about using the XPath wizard, see “Creating XPath Expressions and Functions Using the XPath Wizard” in Chapter 8, “Using Expressions and Conditions.”

Table 7-2 Callback Variables and XPath Statements

Callback Variable	XPath Statement
`FirstName`	`XPath(“/set-variables/variable[1]/text()”)`
`LastName`	`XPath(“/set-variables/variable[2]/text()”)`

When the the Worklist client receives an XML document with this structure, it displays the following dialog box to the user.

Figure 7-29 Enter Your Name Dialog Box

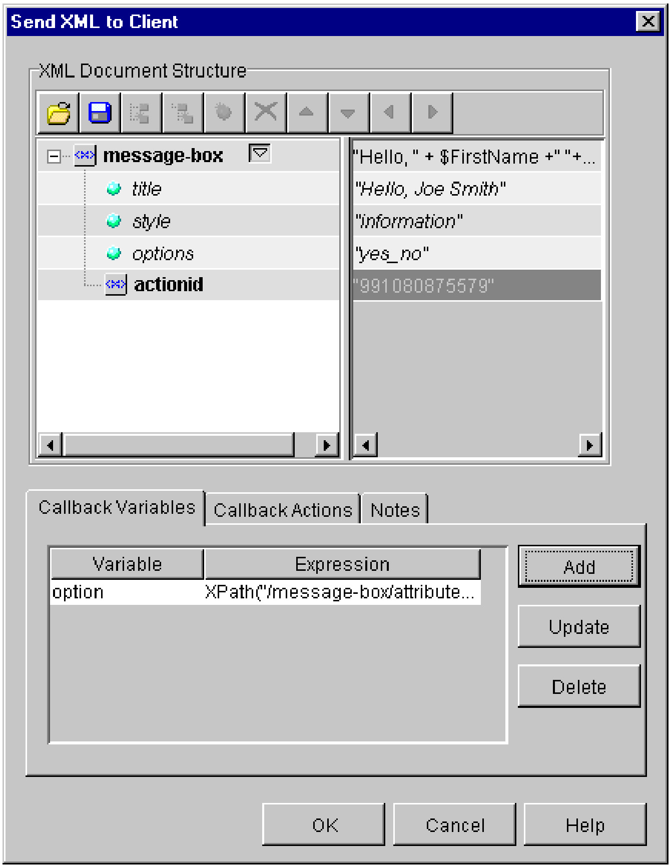


After the user enters the values for the first name and last name, the Worklist client returns the values in an XML document that conforms to the `ClientSetVarsResp.dtd` file. The XPath statements you defined are used by the workflow to extract the first and last name from the XML document, and to put those values into the Callback variables. The workflow can then use the values in the Callback variables for further processing.

Defining a Message Dialog Box

To define a prompt dialog box, the entries in the Send XML to Client dialog box should look like those shown in the following figure.

Figure 7-30 Send XML to Client Dialog Box



message-box	"Hello, " + $FirstName +" "+...
title	"Hello, Joe Smith"
style	"information"
options	"yes_no"
actionid	"991080875579"

Variable	Expression
option	XPath("/message-box/attribute...

The XML document structure shown in the dialog box conforms to the `ClientMsgBoxReq.dtd` file. The contents of the DTD file are shown below.

Listing 7-2 Contents of the ClientMsgBoxReq.dtd File

```
<!ELEMENT message-box (#PCDATA, actionid)>
<!ATTLIST message-box title CDATA #IMPLIED
      style (plain|information|question|warning|error) "plain"
      options (ok|ok_cancel|yes_no|yes_no_cancel) "ok">
<!ELEMENT actionid (#PCDATA)>
```

To define the message box:

1. Enter the following name for the root element of the XML document: `message-box` (all in lowercase, include the hyphen).

   Note: The system-generated `actionid` element and its value are used at run time to identify the Send XML to Client action. The `actionid` element must be the first child element of the root element. Do not move, delete or edit the `actionid` element, and do not add any subelements to it.

   If you change, move, or delete the `actionid` element, a warning message is displayed when you click OK in the Send XML to Client dialog box. The Studio then resets the `actionid` element and its value to the system-generated value and location.

2. Define `title` as an attribute of the root element.

3. The `title` takes the title value of the message-box that is displayed by the Worklist client. Enter the following value for the `title` attribute: `Hello, Joe Smith`.

4. Define `style` as an attribute of the root element, with a value of `information`.

5. Define `options` as an attribute of the root element, with a value of `yes_no`.

6. To complete the dialog box, you need to specify the Callback variable that will hold the response the user selects, and the XPath statements that will extract the value from the response XML document the Worklist client returns to the workflow. The response XML document conforms to the `ClientMsgBoxResp.dtd` file.

The following table shows the Callback variable and the XPath statement you can use. You can create the XPath statement by typing it manually, or you can use the XPath wizard to create them from the `ClientMsgBoxResp.dtd` file.

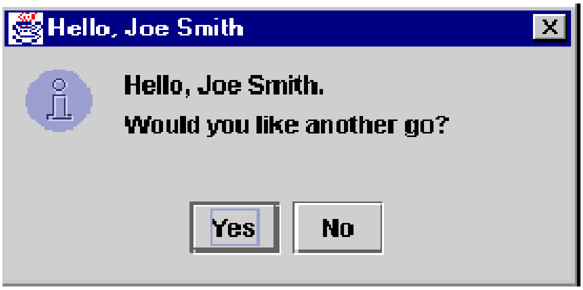
Note: To access the XPath wizard, click Add in the Send XML to Client dialog box. From the Workflow Variable Assignment dialog box, click the A+B button to open the Expression Builder, then click XPath Wizard. For details about using the XPath wizard, see “Creating XPath Expressions and Functions Using the XPath Wizard” in Chapter 8, “Using Expressions and Conditions.”

Table 7-3 Callback Variable and XPath Statement

Callback Variable	XPath Statement
`option`	`XPath(“message-box/attribute::option”)`

When the the Worklist client receives an XML document with this structure, it displays the following dialog box to the user.

Figure 7-31 Example Message-Box



Hello, Joe Smith

Hello, Joe Smith.

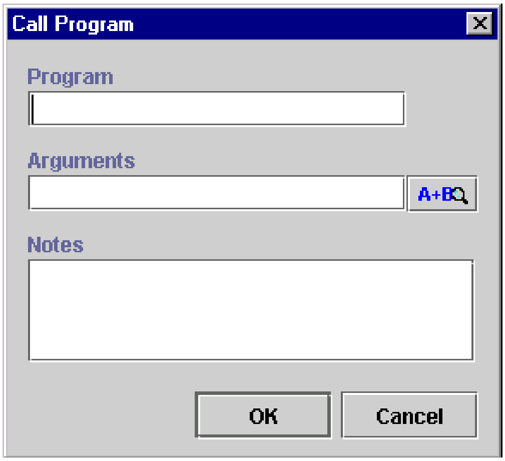
Would you like another go?

After the user responds, the Worklist client returns the answer in an XML document that conforms to the `ClientMsgBoxResp.dtd` file. The XPath statement you defined is use by the workflow to extract the user’s answer from the XML document, and to put the value into the Callback variable. The workflow can then use the value in the Callback variable for further processing.

Call Program

Use this action to call an executable program on the workflow server.

Figure 7-32 Call Program Dialog Box



View or enter information in the Call Program dialog box.

- Program — the fully qualified path to an executable program to execute at run time.
- Arguments — a valid workflow expression to be evaluated at run time and passed to the called program.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- Notes — a free-form text box for notes about the action.

This action is executed asynchronously, meaning that any actions following this action in the action list are executed immediately, without waiting for the called program to complete.

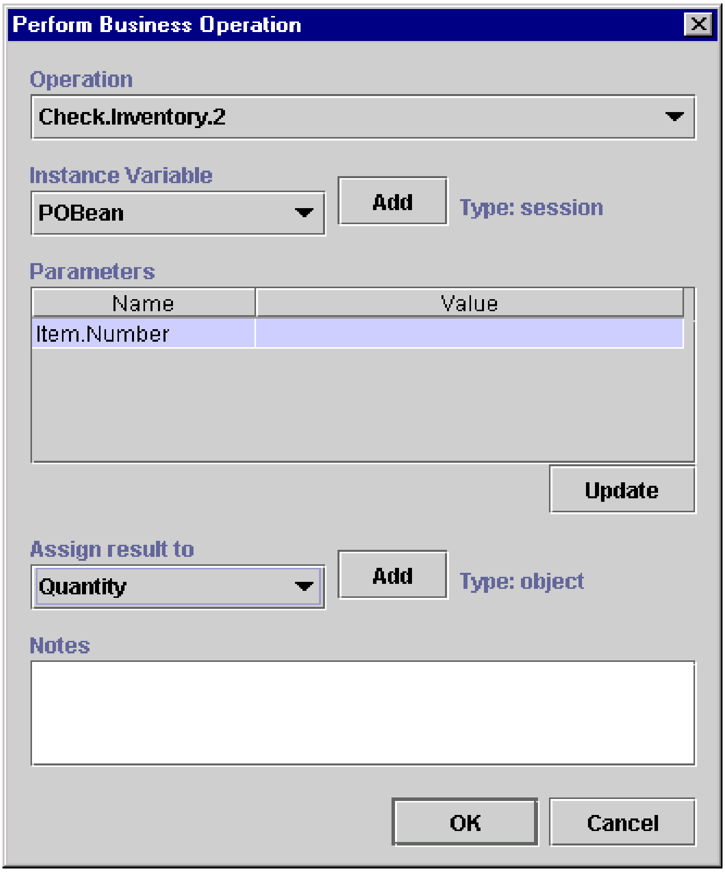
Perform Business Operation

Use this action to invoke a software component, such as an EJB or Java class, to perform a business activity.

The business operation you want to invoke must already be defined. Additionally, any Java Object, Session EJB, and Entity EJB variables referenced by the business operation must also be defined. Ideally, technical personnel with detailed knowledge about the software components define the business operation, and they can assist you in making the correct selections for this action.

For details about defining business operations, see “Understanding Business Operations” on page 4-35. For details about defining variables, see “Defining Variables” on page 6-3.

Figure 7-33 Perform Business Operation Dialog Box



Perform Business Operation

Operation: Check.Inventory.2

Instance Variable: POBean

Type: session

Parameters

Name	Value
Item.Number	

Assign result to: Quantity

Type: object

Notes

View or enter the following information in the Perform Business Operation dialog box.

- Operation — select the predefined business operation to invoke.
- Instance Variable — select the variable that contains the reference to the Java object or EJB that the business operation invokes. This option is available if the business operation invokes one of the following types of methods:
  - EJB remote method
  - Java class non-static, non-constructor method

  For these types of methods, the instance variable drop-down list displays the following types of variables: Java Object, Session EJB, and Entity EJB.
- Add — create a new instance variable not shown in the list using the Variable Properties dialog box. For details about the Variable Properties dialog box, see "Defining Variables" on page 6-3.
- Parameters — the business operation you select might have parameters that require values. If it does, you must enter values for the parameters. To add a parameter value to a listed parameter, highlight it, click Update, and use the invoked Expression Builder dialog box to define this value. (See "Using the Expression Builder" on page 8-3.)
- Assign Result To — assign the business operation result (for example, a calculated average) to a workflow variable. The drop-down list shows the available variables. When the business operation is invoked through the Perform Business Operation action, the method call assigns the result of its call to this variable.
- Add — create a new variable not shown in the list using the Variable Properties dialog box. For details about the Variable Properties dialog box, see "Defining Variables" on page 6-3.
- Notes — a free-form text box for notes about the action.

XSL Transform

XSLT is a specialized XML language that defines rules for translating an XML document into another XML or non-XML document. An XSL template document specifies which elements in the input XML document are to be transformed, and how they are to be transformed.

The XSL Transform action provides a way for you to specify an input XML document that is to be transformed, the XSL template document that specifies the details of the transformation, and a output variable that will contain the transformed document.

The actual transformation occurs at run time, and is performed by the XSL transformation engine bundled with WebLogic Server.

The input document can contain workflow expressions. If it does, BEA WebLogic Process Integrator resolves the expressions, and replaces the expression with the result before the transformation occurs.

Similarly, the XSL template document can contain references to workflow variables. If it does, BEA WebLogic Process Integrator resolves the references, and replaces the reference with the appropriate value before the transformation occurs.

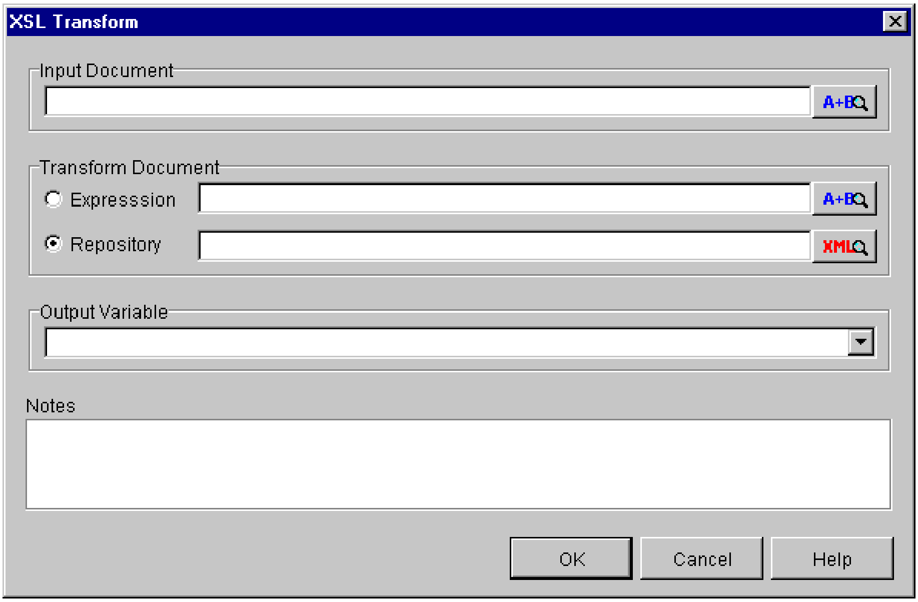
To define the XSL Transform action:

1. Access the Add Action dialog box. This dialog box can be accessed from the various Properties dialog boxes in BEA WebLogic Process Integrator. For example, in the workflow drawing area, right-click a task and choose Properties. This displays the Task Properties dialog box.

2. In the Task Properties dialog box, click Add to display the Add Action dialog box.

3. Under Integration Actions, select XSL Transform, and click OK to display the XSL Transform dialog box.

Figure 7-34 XSL Transform Dialog Box



4. In the Input Document field, enter an expression that represents the XML document you want to transform. The expression is evaluated at run time to obtain the XML document.

   Note: Expressions are built using WebLogic Process Integrator expression syntax. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

5. In one of the Transform Document fields, specify the XSL template document that is used to transform the input document. Specify the XSL template document in one of following ways:

   - Expression — represents the XSL template document. The expression is evaluated at run time to obtain the XSL template document. Enter the expression directly, or use the Expression Builder to construct the expression.

- Repository — contains the name of an XSL template document stored in the XML repository. Enter the name of the document directly, or click the XML button to display the XML Finder dialog box. The name of the XSL template document is stored in the workflow template definition.

  The XML Finder dialog box helps you find XML objects stored in the XML repository, the local file system, or at a URL. For more information about using the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.

6. From the Output Variable drop-down list, select the variable that will contain the transformed XML document. The variable must be an XML type variable. You can also type the name of a variable. If it does not exist, you will be prompted to create it. For details about defining variables, see “Understanding Variables” on page 6-3.
7. In the Notes field, enter any notes about this action.
8. Click OK.

Using Exception Handling Actions

For instructions on using exception handling actions, see “Using Exception Handling Actions” on page 9-7.

Using Miscellaneous Actions

This section defines and describes how to use the following miscellaneous actions:

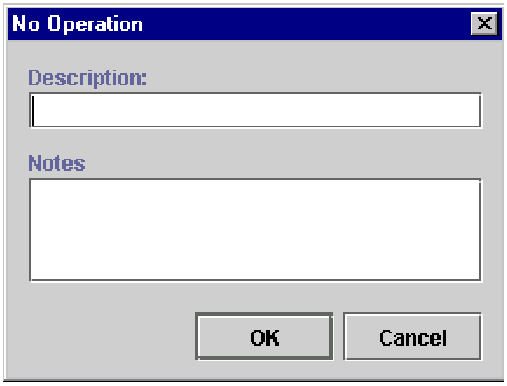
- No Operation
- Send E-Mail Message
- Evaluate Condition
- Timed Event

- Cancel Workflow Event
- Make Audit Entry

No Operation

The No Operation action has no function; it simply acts as a placeholder to remind an analyst to add an action at a later time.

Figure 7-35 No Operation Dialog Box



View or enter information in the No Operation dialog box.

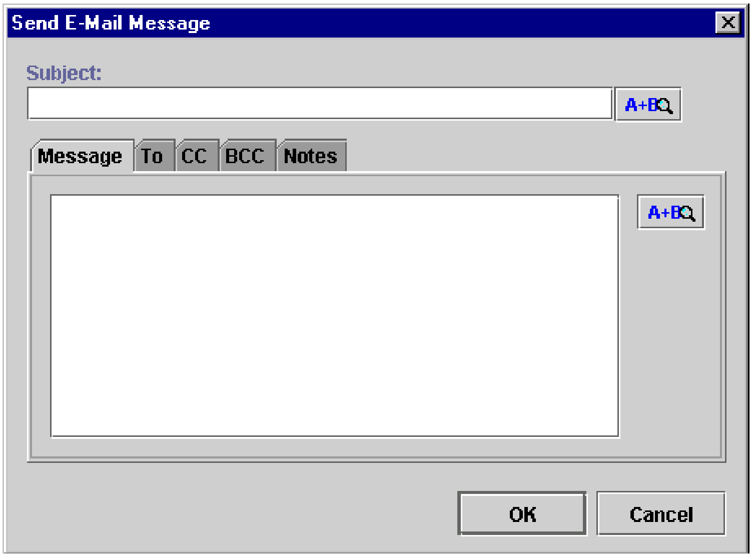
- Description — enter a descriptive name for the No Operation action (preferably a name that will remind you to add an action at a later time).
- Notes — a free-form text box for notes about the action.

Send E-Mail Message

Send an e-mail message to a user of the BEA WebLogic Process Integrator system or even to an outside party. Internet standard SMTP protocol is used to transmit the message.

Note: This action requires configuration in the config.xml file. For details, see *Installing and Configuring BEA WebLogic Process Integrator*.

Figure 7-36 Send E-Mail Message Dialog Box



View or enter information in the Send E-Mail Message dialog box.

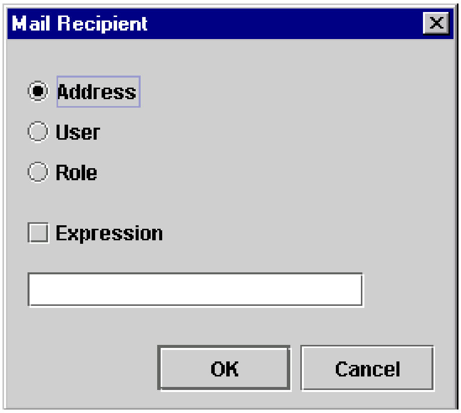
- Subject — a valid workflow expression or a literal entered in double quotes that will be evaluated at run time for the subject text of the e-mail.
- Message — a valid workflow expression or a literal entered in double quotes that will be evaluated at run time for the message text of the e-mail.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- TO, CC, and BCC — determine how the e-mail should be addressed.
- Notes — a free-form text box for notes about the action.

Click OK to display the Mail Recipient dialog box.

Figure 7-37 Mail Recipient Dialog Box



● Address
○ User
○ Role

□ Expression

- Address — if selected, an edit field displays the e-mail address.
- User — if selected, a drop-down list displays a list of all the users. The e-mail address is obtained from the user's profile.
- Role — if selected, a drop-down list displays a list of all roles. The e-mail message will be sent to all users assigned to that role.
- Expression — the e-mail address is derived from an expression value.

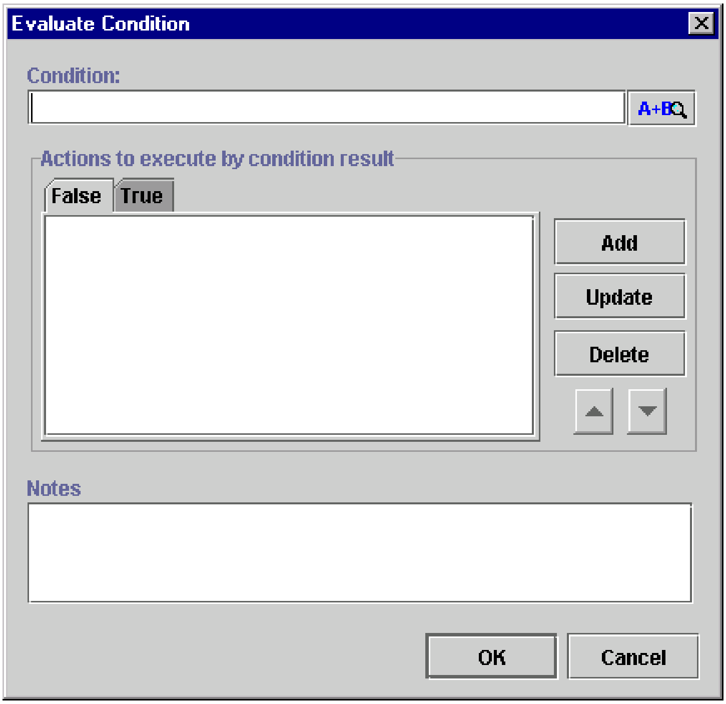
Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- Notes — a free-form text box for notes about the action.

Evaluate Condition

Evaluate a conditional expression at run time and perform alternative sequences of subactions depending on the result. If the condition evaluates to True, one sequence of actions is performed, and if it is False, a different sequence of actions is performed instead. For example, a manager may be authorized to make purchases with a total value that is less than a specified amount, requiring higher approval if the amount exceeds this threshold. When a manager starts a purchase requisition workflow, the workflow can use an Evaluate Condition action to check whether this threshold would be exceeded, and to assign the task of approving the requisition either to the authorized person or to the manager.

Figure 7-38 Evaluate Condition Dialog Box



View or enter information in the Evaluate Condition dialog box.

- Condition — a valid workflow condition, evaluated at run time to determine which set of subactions to perform.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

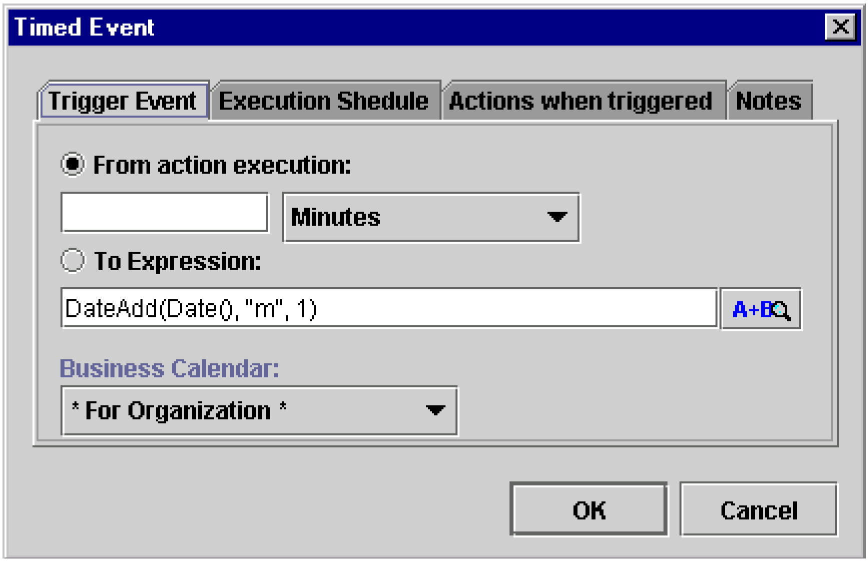
- Actions to execute by condition result — a tabbed display showing two sets of subactions. Either the True or the False set of subactions will be executed, depending on the result of the condition evaluation.

- Notes — a free-form text box for notes about the action.

Timed Event

Create a timed even that will be triggered at an exact time and date according to a specified execution schedule.

Figure 7-39 Timed Event: Trigger Event Tab



● From action execution:

Minutes

○ To Expression:

DateAdd(Date(), "m", 1)

Business Calendar:

* For Organization *

View or enter information in the Trigger Event tab.

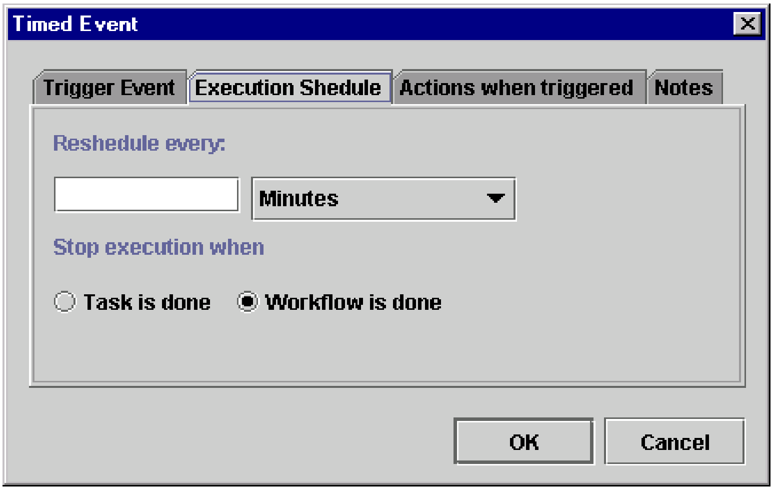
- From action execution — perform the trigger upon action execution, based on a specified unit of time. Use the drop-down list to the right to select the unit of time, and in the field to the left, enter a number to be associated with the unit of time. For example, if you choose “4 Minutes,” the event will be triggered four minutes after the preceding event is executed.
- To Expression — timed event will be triggered according to a date expression.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, "Using Expressions and Conditions."

- Business Calendar — if you select the interval Business Hours or Business Days, the business calendar is used.
  - For Organization — use the business calendar assigned to the organization for which the workflow template definition instance will be executed.
  - For Assignee — use the calendar that belongs to the user or role owning the task that precedes the event to be triggered.

The Execution Schedule tab of the Timed Event dialog box allows you to specify that the event will be repeated according to a specified unit of time until the task is marked done or the workflow template definition is done.

Figure 7-40 Timed Event: Execution Schedule Tab



Reshedule every: Minutes

Stop execution when

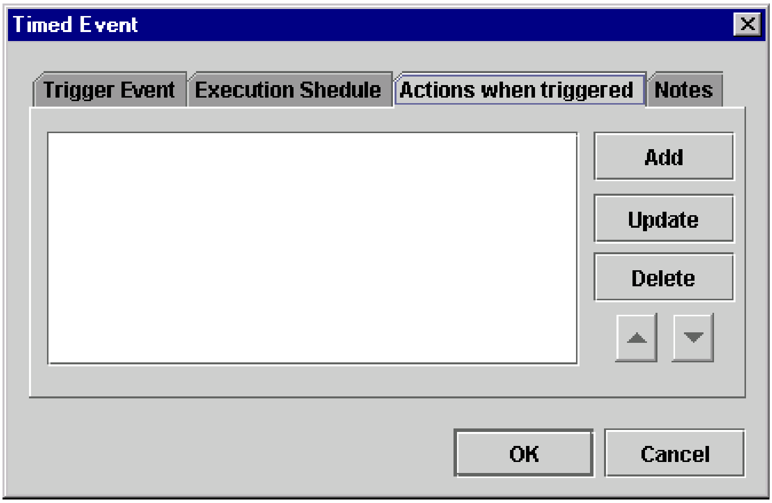
○ Task is done ● Workflow is done

View or enter information in the Execution Schedule tab.

- Reschedule every — repeat the trigger upon task execution, based on a specified unit of time. Use the drop-down list to the right to select the unit of time, and in the field to the left, enter a number to be associated with the unit of time. For example, if you choose "4 Minutes," the event will be triggered every four minutes until the task is marked done or the workflow template definition is done.

- Stop execution when — specify when the trigger should stop, either when the current task is completed or when the entire workflow is completed.
  - Task is done — the trigger will stop when the preceding task in which this action is specified is marked as done.
  - Workflow is done — the trigger will stop when the entire workflow is completed.

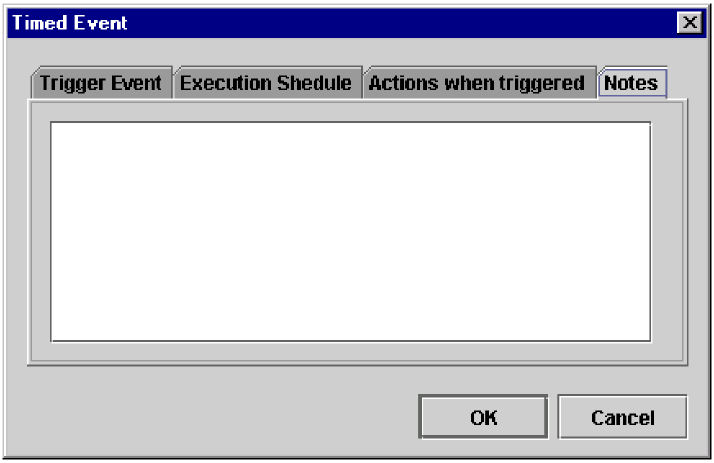
Figure 7-41 Timed Event: Actions When Triggered Tab



View or enter information in the Actions when triggered tab.

- Add — invokes the Add Action dialog box. Select the action(s) to be triggered by the timed event.
- Update — updates the selected action to be triggered by the timed event. Highlight the action to be updated to invoke the relevant Action dialog box, where you can make updates to the action.
- Delete — deletes the selected action from the timed event.

Figure 7-42 Timed Event: Notes Tab

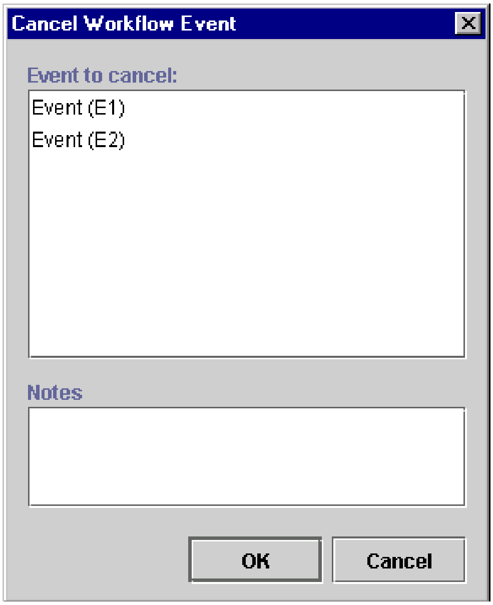


Use the Notes tab of the Timed Event dialog box to make notes about the action.

Cancel Workflow Event

This action cancels an event that has been defined within a workflow.

Figure 7-43 Cancel Workflow Event Dialog Box



View or enter information in the Cancel Workflow Event dialog box.

- Event to Cancel — displays all events defined in the workflow. Highlight the workflow event to be canceled. (Hold down the CTRL key to highlight more than one event.) Click OK to proceed with the cancellation.
- Notes — a free-form text box for notes about the action.

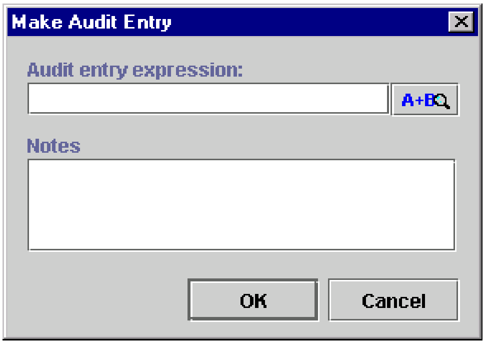
Make Audit Entry

This action allows you to specify an additional audit entry expression. This audit entry will be published to a Java Message Service (JMS) topic for further retrieval. The audit entry can be an expression built with functions and workflow variables, but it can also be a string of text.

Note: The JMS topic is specified in the Template Definition Properties dialog box. For further information, refer to “Creating a New Workflow Template Definition” on page 5-4.

BEA WebLogic Process Integrator provides a default audit facility whereby all major user interactions and changes are published to the JMS topic. The Make Audit Entry action allows you to add an additional specific audit entry; such an audit entry might be used, for example, to track the value of a variable.

Figure 7-44 Make Audit Entry Dialog Box



The Make Audit Entry dialog box contains the following:

- Audit entry expression — enter an audit entry expression to be set out to the audit listener for further retrieval at run time.

  Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

- Notes — a free-form text box for notes about the action.

CHAPTER

8 Using Expressions and Conditions

The following sections explain how to use the BEA WebLogic Process Integrator Expression Builder and expression components:

- Workflow Expression Functionality
- Using the Expression Builder
- Using Expression Components
- Creating XPath Expressions and Functions Using the XPath Wizard

Workflow Expression Functionality

A workflow expression is an algebraic expression that defines a calculation that the BEA WebLogic Process Integrator system performs at run time. You build workflow expressions using the WebLogic Process Integrator expression syntax, which is comparable to that of Structured Query Language (SQL). You also use XPath function syntax to extract values from XML documents. The WebLogic Process Integrator expression syntax makes it easy to manipulate strings, test for relationships and conditions, and perform arithmetic calculations involving workflow variables, database fields, and literal values (constants).

The result of an expression may be a string, a string representation of an integer, double, date/time value, or either of the Boolean (logical) values True and False. Expressions that yield a Boolean result are variously referred to as conditional expressions or conditions.

You can use workflow expressions, as described in the following sections.

Generate a String that Uniquely Identifies a Workflow Instance

Consider a workflow named "Registration of Invoice," concerned with the process of registering invoices. Instances of the workflow can be uniquely identified by appending the invoice number and date to this workflow name. The following expression yields a string that, when appended to the workflow name, provides an informative description of the workflow instance:

```
"#" + :’Invoice #’ + ", dated " + :’Invoice Date’
```

where `’Invoice #’` and `’Invoice Date’` are the names of workflow fields.

This expression might yield the following workflow instance name:

```
"Registration of Invoice # 12345, dated May 16, 2000"
```

Determine Whether a Particular Event Should Trigger a Workflow

Consider a customer credit review workflow that should be initiated if the current balance due exceeds 75 percent of the customer's credit limit. The workflow is triggered by the arrival of an XML document indicating that a customer record in the underlying Accounts Receivable application has changed.

If the XML document "customer" contains elements called `balance_due` and `credit_limit_amount`, the expression required to detect this condition in the XML document is:

```
ToInteger(XPath("/customer/balance_due/text()")) >
ToInteger(XPath("/customer/_credit_limit_amount/text()"))
* .75
```

Using the Expression Builder

The Expression Builder enables you to build expressions to be executed in BEA WebLogic Process Integrator at run time. Expressions are made up of functions, operators, literals, and variables. These expression components are discussed in "Using Expression Components" on page 8-6.

The Properties dialog boxes for workflow components have an Expression Builder button, shown in the following figure. You click the button to display the Expression Builder dialog box.

Figure 8-1 Expression Builder Button

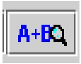
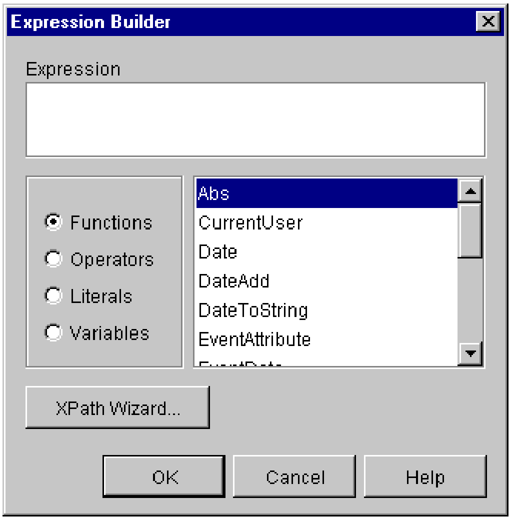


Figure 8-2 Expression Builder Dialog Box



Note: If a plug-in is defined for functions that are part of expressions, this dialog box contains the new function. For more information about plug-ins, see Chapter 3, “Working with Plug-Ins.”

To build an expression, either type it directly in the Expression field or use the functions, operators, literals, and variables options to build the expression. Selecting a component from the scrollable list to the right and clicking OK places the appropriate expression text in the Expression field.

To create XPath expressions, you can either type them yourself, or you can use the XPath wizard. To open the XPath wizard, clickXPath Wizard. For details about using the wizard, see “Creating XPath Expressions and Functions Using the XPath Wizard” on page 8-22.

If the expression is valid, the message "Expression is valid" appears in the Expression Builder dialog box; otherwise, an error message is presented in a dialog box. Note that validation does not check the validity of workflow variable referenced by the expression, nor does it check for the correct number of parameters being passed to a function, nor does it attempt to do type checking. If there is a type mismatch in the expression (for example, performing arithmetic on a non-numeric string), a run-time error results when the expression is evaluated.

If you enter an invalid expression, the system displays a message that attempts to explain why the expression is invalid. The following table lists the messages that may be encountered, together with possible causes.

Table 8-1 Invalid Expression Messages

Message	Possible Cause(s)
Invalid <operator> operand	One or both of the operand(s) to an AND, OR, XOR, NOT,*,/,%,+,-,<,<=,=,<>,>=,> operator was not valid. For example, attempted arithmetic on a non-numeric string: "Name: " * 25.4.
Mismatched operand types in <operator> comparison	Attempting to compare values of different types. For example, comparing a date to a number: {97/09/30} <= 56.9.
NumberFormatException	A string representation of a number did not have the required format for the type of number. For example: 1.23ZX.
Invalid function name	The expression calls an undefined function. For example: "Name: " + somefunc(:var). somefunc is not a valid function name.
Incomplete escape sequence	The last character in a quoted string was a back slash character that was not followed with one of the following characters: r, n, f, t, \, ’, ",0.
Invalid escape sequence	A quoted string contained a back slash character that was not followed with one of the following characters: r, n, f, t, \, ’, ",0.
Fatal error	An internal error occurred in the expression evaluator. The expression probably contains a syntax error.
Unclosed string	A string literal did not have a matching quote at the end. In the following example, a double quote is not matched by a single quote: ‘This is an unclosed string'.
Illegal character	The expression contains a character that violates the syntax of the expression language. In the following example, ! is the invalid character: 7 * :wks + " days" !.
Error: unmatched input	The evaluator could not interpret the expression. The expression probably contains a syntax error.

Table 8-1 Invalid Expression Messages

Message	Possible Cause(s)
Unrecoverable syntax error	The expression contains an invalid token. In this example, name is invalid because the + operator was omitted: "Name: " name. The Expression should have been: "Name: " + name.

Using Expression Components

Expressions are made up of functions, operators, literals, and variables.

Functions

WebLogic Process Integrator provides the following functions, which can be used by the workflow developer to construct conditions.

Note: If a plug-in has been defined for functions that are part of expressions, you can select the new functions in addition to the following ones. For more information about plug-ins, see Chapter 3, “Working with Plug-Ins.”

Abs

Returns the absolute value of the expression.

`Abs(`*`expression`*`)`

`expression` is a workflow expression for which the absolute value is to be calculated.

CurrentUser

Returns the ID of the user currently executing a task.

```
CurrentUser
```

Date

Returns current system date.

```
Date()
```

DateAdd

Performs date arithmetic.

`DateAdd(`*`date`*`, `*`interval`*`, `*`number`*` [, buscal])`

`date` is the base date.

`interval` is the unit of time to use. Possible values are listed in the following table.

Table 8-2 Interval Values

Interval	Description
`S`	Seconds
`m`	Minutes
`H`	Hours
`D`	Days
`W`	Weeks
`M`	Months
`BH`	Business hours
`BD`	Business days

The interval is not case sensitive except for "m" and "M".

number is the number of units to add (or subtract).

buscal is the name of the business calendar to be used for the business hours and business days calculation. If no calendar is specified, the default calendar is used.

DateToString

Converts a date to a string.

```
dateToString(date, format)
```

date is the date to be converted to the string value.

format is a string specifying the format of the string. Possible values are listed in the following table.

Table 8-3 Format Values

Format	Description
`yyyy`	Year for example: `2000`
`MM`	Numeric month of the year for example, January = `01`, February = `02`, and so on
`dd`	Numeric day of the month for example: `02` or `28`
`DD`	3- digit day of the year based on the total number of days (365) in a year for example: October 18, 2000 = `292` (the 292nd day of the year)
`hh`	Hours in non-military time for example: 1 pm = `01`, 2 pm = `02`, and so on
`HH`	Hours in military time for example: 1 pm = `13`, 2 pm = `14`, and so on
`mm`	Minutes for example: `02` minutes

Table 8-3 Format Values

Format	Description
`ss`	Seconds for example: `35` seconds
`SSS`	Milliseconds for example: `370` milliseconds

Note: Format is case sensitive.

The following separator characters are valid:

- - (dash)
- / (slash)
- : (colon)
- . (period)
- spaces

For example, given the following date and time: October 18, 2000, 2:30 pm, 35.370 seconds, the format result can be entered as follows:

`DateToString(Date(), “yyyy-MM-dd HH:mm:ss.SSS”)` resulting in `2000-10-18 14:30:35.370`

`DateToString(Date(), “yyyy/MM-DD hh:mm:ss.SSS”)` resulting in `2000/10-292 02:30:35.370`

EventAttribute

Obtain event properties from an incoming EventData object, such as a JMS message property.

```
EventAttribute(attribute)
```

`attribute` is the value to extract from the JMS properties.

For example, you can set JMS message property values using the Post XML Event dialog box, and then send the values to another workflow. The receiving workflow then retrieves the property information using the EventAttribute function in an event key expression.

For details about using the Post XML Event action, see "Post XML Event" on page 7-32. For details about defining an event key expression, see "Defining an Event Key" on page A-4.

EventData

Retrieve the actual message content of the incoming JMS message. The content could be an XML document.

```
EventData()
```

StringLen

Provides the string length.

```
stringlen(string)
```

`string` is the string for which a length will be provided.

StringToDate

Converts a string to a date.

```
stringToDate(string, format)
```

`string` is the string to be converted to the date value.

`format` is a string specifying the format of the string.

SubString

Extracts a substring from a string.

`SubString(`*`string`*`, `*`start`*` [,`*`length`*`])`

`string` is the string from which the substring is to be extracted.

`start` is the starting position in the string. (The first position is 0.)

`length` is the length of the substring. This is an optional parameter. If it is not included, the substring extracted will be the rest of the string.

TaskAttribute

Provides the task attribute.

`TaskAttribute(attribute [,taskname])`

`attribute` is the task attribute.

The following table identifies the construction for inserting system maintained workflow information into an expression.

Table 8-4 TaskAttribute Expressions

Information	Expression
Task instance ID	`TaskAttribute(“TaskId”)`
User ID of assignee	`TaskAttribute(“Assignee”)`
Priority set by application	`TaskAttribute(“Priority”)`
Task overdue date	`TaskAttribute(“Due”)`
Name of task	`TaskAttribute(“Name”)`
Task start date/time	`TaskAttribute(“Started”)`
Completed date/time	`TaskAttribute(“Completed”)`
Application comment	`TaskAttribute(“Comment”)`

ToInteger

Converts a string value to an integer.

`ToInteger(expression)`

`expression` is the string (enclosed in double quotes) to be converted to integer.

ToString

Converts any data type value to a string.

`ToString(expression)`

`expression` is the integer to be converted to string.

WorkflowAttribute

Provides the workflow attribute.

`attribute` is the workflow attribute.

The following table identifies the construction for inserting system maintained workflow information into an expression.

Table 8-5 WorkflowAttribute Expressions

Information	Expression
Workflow instance ID	`WorkflowAttribute(“InstanceId”)`
Workflow template ID	`WorkflowAttribute(“TemplateId”)`
Template definition ID	`WorkflowAttribute(“TemplateDefinitionId”)`
Workflow initiator	`WorkflowAttribute(“Initiator”)`
Parent workflow ID	`WorkflowAttribute(“ParentId”)`
Workflow name	`WorkflowAttribute(“Name”)`
Start date/time	`WorkflowAttribute(“Started”)`
Completed date/time	`WorkflowAttribute(“Completed”)`
ID set by application	`WorkflowAttribute(“Id”)`
Application comment	`WorkflowAttribute(“Comment”)`
Name of Java exception class that raised the error	`WorkflowAttribute(“ErrorType”)`

Table 8-5 WorkflowAttribute Expressions

Information	Expression
Message number of the error being handled by the exception handler	`WorkflowAttribute(“ErrorNumber”)`

One of four values:
Error_System = “1”
Error_Workflow = “2”
Warning_Workflow = “3”
Error_Custom = “4”
Note: Value is set to Error_Custom when a workflow invokes an exception handler via the Invoke Error Handler action or via the API.
	`WorkflowAttribute(“ErrorSeverity”)`

Message text of the error being handled by the exception handler	`WorkflowAttribute(“ErrorText”)`

Note: For a list of WebLogic Process Integrator error messages, see “WebLogic Process Integrator Error Messages” on page 9-14.

WorkflowVariable

Returns the value of a workflow variable for a particular workflow instance.

```
workflowVariable(instanceid, variable)
```

`instanceid` is the workflow instance ID. For details, see "Start Workflow" on page 7-25.

`variable` is the workflow variable.

XPath

Note: To create XPath expressions, you can either type them yourself, or you can use the XPath Wizard. For details about using the wizard, see "Creating XPath Expressions and Functions Using the XPath Wizard" on page 8-22.

```
XPath(xpathstring, xmldocument)
```

`xpathstring` is the XPath expression

`xmldocument` is the name of the variable that contains the XML document on which the expression is evaluated. This parameter can either be of XML type or string type. This parameter is optional. If it is not specified, the XML document is assumed to be the incoming XML event.

`XPath` is a language for addressing parts of an XML document. It provides basic facilities for the manipulation of strings, numbers, and Booleans. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document. To obtain the official specifications of the XPath language, refer to the following Internet site:

http://www.w3.org/TR/xpath.html

To obtain a quick reference guide to XPath notations and functions, refer to the following Internet site:

http://www.mulberrytech.com/quickref/XSLTquickref.pdf

The following are examples of the most common uses of XPath for retrieving text values from nodes, attribute values, selecting subtrees, selecting nodes by attribute value and so on from an XML document. Consider the XML document in the following listing:

Listing 8-1 Common Uses of XPath

```
<?xml version="1.0"?>
<a>
  <b name="bill">This is the first value</b>
  <c>
    <d id="d1">This is the second value</d>
    <d id="d2">This is the third value</d>
    <d id="d3">This is the fourth value</d>
    <d id="d4">This is the fifth value</d>
  </c>
</a>
```

To select the textual value of the `<b>` element (i.e., "This is the first value"):

```
XPath("/a/b/text()")
```

To select the textual value of the `<d>` element whose ID attribute has the value "`d3`" (i.e., "This is the fourth value"):

```
XPath("/a/c/d[@id=\"d3\"]/text()")
```

To select the textual value of the second `<d>` element (i.e., "This is the third value"):

```
XPath("/a/c/d[2]/text()")
```

To select the value of the name attribute of the `<b>` element (i.e., "`bill`"):

```
XPath("/a/b/@name")
```

To select the entire `<c>` subtree:

```
<c>
    <d id="d1">This is the second value</d>
    <d id="d2">This is the third value</d>
    <d id="d3">This is the fourth value</d>
    <d id="d4">This is the fifth value</d>
</c>)
XPath("/a/c")
```

Date Function Format

To specify the date format, use a *time pattern* string to describe dates and times. For example, the following time pattern string:

```
"yyyy.MM.dd G ’at’ hh:mm:ss z"
```

results in the following formatting:

```
2000.07.31 AD at 13:10:35 PDT
```

This section describes pattern letters, provides formatting guidelines, and shows examples.

Pattern Letter Definitions

In this pattern, all ASCII letters are reserved as pattern letters, which are defined as the following.

Listing 8-2 Pattern Letter Definitions

```
Symbol   Meaning               Presentation     Example
------   -------               ------------     -------
 G        era designator        (Text)           AD
 y        year                  (Number)         1996
 M        month in year         (Text & Number)  July & 07
 d        day in month          (Number)         10
 h        hour in am/pm (1~12)  (Number)         12
 H        hour in day (0~23)    (Number)         0
 m        minute in hour        (Number)         30
 s        second in minute      (Number)         55
 S        millisecond           (Number)         978
 E        day in week           (Text)           Tuesday
 D        day in year           (Number)         189
```

F	day of week in month	(Number)	2 (2nd Wed in July)
w	week in year	(Number)	27
W	week in month	(Number)	2
a	am/pm marker	(Text)	PM
k	hour in day (1~24)	(Number)	24
K	hour in am/pm (0~11)	(Number)	0
z	time zone	(Text)	Pacific Standard Time
’	escape for text		
’’	single quote		’

Format Guidelines

The count of pattern letters determine the format.

- (Text) — for 4 or more pattern letters, use the full form. For less than 4, use the short or abbreviated form, if one exists.
- (Number) — the minimum number of digits. Shorter numbers are zero-padded to this amount. Year is handled specially; that is, if the count of ‘y’ is 2, the year will be truncated to 2 digits.
- (Text & Number) — for 3 or more pattern letters, use text; otherwise, use a number.

Any characters in the pattern that are not in the ranges of `[’a’..’z’]` and `[’A’..’Z’]` will be treated as quoted text. For instance, characters like `’:’`, `’.’`, `’ ’`, `’#’` and `’@’` will appear in the resulting time text even though they are not within single quotes.

A pattern containing any invalid pattern letter will result in a thrown exception during formatting or parsing.

Examples of Time Patterns

Listing 8-3 Examples Using the U.S. Locale

```
Format Pattern                         Result
--------------                         -------
"yyyy.MM.dd G ’at’ hh:mm:ss z"    ->>  1996.07.10 AD at 15:08:56 PDT
"EEE, MMM d, ’’yy"                ->>  Wed, July 10, ’96
"h:mm a"                          ->>  12:08 PM
"hh ’o’’’’clock’ a, zzzz"         ->>  12 o’clock PM,Pacific Daylight Time
"K:mm a, z"                       ->>  0:00 PM, PST
"yyyyy.MMMMM.dd GGG hh:mm aaa"    ->>  1996.July.10 AD 12:08 PM
```

Operators

The following table describes the available operator values.

Table 8-6 Operator Values

Operator	Actual Value	Syntax	Operand(s)	Result
AND	AND	expr1 AND expr2	Logical	True if expr1 and expr2 are both True; otherwise False
OR	OR	expr1 OR expr2	Logical	True if either expr1 or both expr1 and expr2 are True; otherwise False
XOR	XOR	expr1 XOR expr2	Logical	True if either but not both expr1 or expr2 are True; otherwise False
NOT	NOT	NOT expr	Logical	True if expr is False; otherwise False
(parentheses)	()	(expr)	Any expression	Evaluates expr first
Multiply	*	expr1 * expr2	Numeric	Numeric product
Divide	/	expr1 / expr2	Numeric	Numeric quotient
Modulo	%	expr1 % expr2	Numeric	Numeric modulus (remainder when expr1 divided by expr2)
Plus	+	+ expr expr1 + expr2	Numeric String or numeric	Unary plus String concatenation or numeric addition
Minus	-	- expr expr1 – expr2	Numeric Numeric	Unary minus (negates expr) Subtraction
Less than	<	expr1 < expr2	String or numeric	Logical True if expr1 is less than expr2; otherwise False
Less than or equal to	<=	expr1 <= expr2	String or numeric	Logical True if expr1 is less than or equal to expr2; otherwise False

Table 8-6 Operator Values

Operator	Actual Value	Syntax	Operand(s)	Result
Equal to	=	expr1 = expr2	String or numeric	Logical True if expr1 is equal to expr2; otherwise False
Not equal to	<>	expr1 <> expr2	String or numeric	Logical True if expr1 is not equal to expr2; otherwise False
Greater than	>	expr1 > expr2	String or numeric	Logical True if expr1 is greater than expr2; otherwise False
Greater than or equal to	>=	expr1 >= expr2	String or numeric	Logical True if expr1 is greater than or equal to expr2; otherwise False

Logical operators AND, OR, XOR, and NOT are valid only when the Expression Builder is constructing a logical condition (for example, a conditional action).

Literals

Expressions can contain literal values, or constants. The following table describes the available literals.

Table 8-7 Literal Values

Literal Type	Syntax	Description
String	“string” or ‘string’	A character string enclosed in single or double quotes. Use the following escape sequences to embed special characters in the string: \r carriage return character \n line feed character \’ single quote character \” double quote character \f form-feed character \t tab character \\ back slash character \0 ASCII Null character
Integer	[+|-] digits	A 32-bit signed integer in the range -2,147,483,647 to +2,147,483,648 (approximately 9 digits or precision)
Double	[+|-]digits[.[digits]]	A 64-bit IEEE double in the range -2^{53} x 10^{104} to + 2^{53} x 10^{104} (approximately 15 digits of precision)
Date/Time	{yyyy/mm/dd hh:nn:ss}	A date and time

Variables

An expression can contain references to any workflow variable defined in the associated workflow and to any system-defined variable.

A workflow variable reference can use any one of the following syntaxes:

- `:variable`
- `:’variable’`
- `:"variable"`
- `$variable`
- `$'variable'`
- `$"variable"`

It is better to use the $ notation for consistency with the XPath function.

Creating XPath Expressions and Functions Using the XPath Wizard

The XPath wizard provides a graphical interface that you can use to define, edit, and test XPath location expressions and XPath functions. Specifically, you can:

- Create XPath location expressions and functions for sample XML, DTD, and schema documents
- Test the XPath location expressions and functions you create by applying them to XML documents
- View all the XPath location expressions and functions you create during a Studio session

The following sections provide details about these tasks. You can also view information about using the XPath wizard by clicking the Show Information button in the wizard menu bar.

Creating XPath Location Expressions

Using the XPath wizard, you can create XPath location expressions that target:

- Element content
- Attribute content
- Nodes

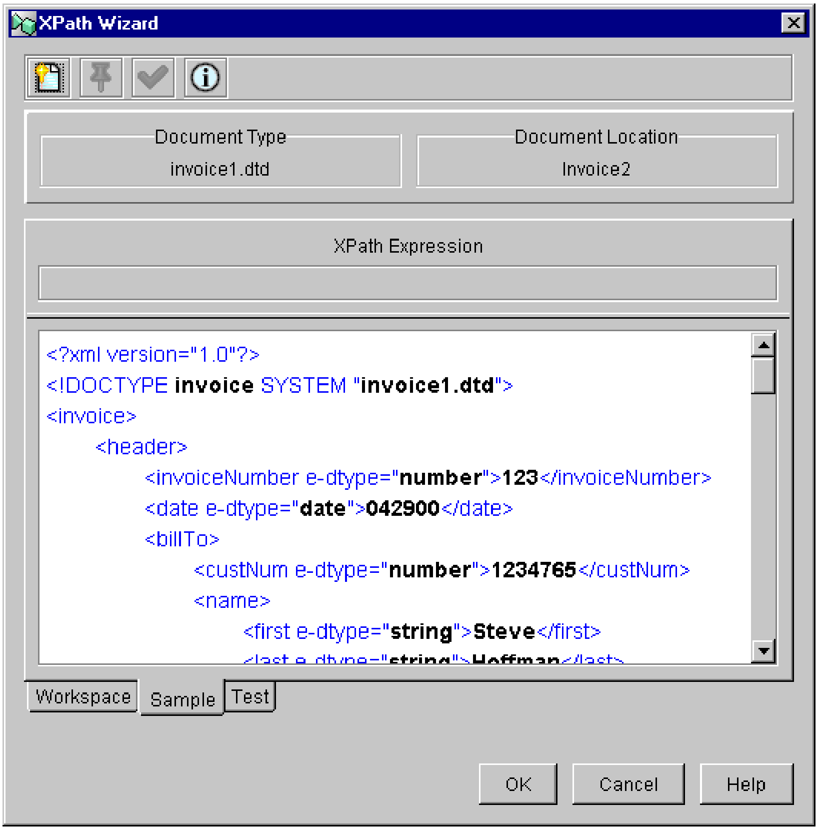
To create an XPath location expression:

1. Select the Sample tab to open the Sample area where you can create XPath expressions.
2. Click the Open button to open the XML Finder dialog box. Use the XML Finder dialog box to retrieve an XML, DTD, or schema (XSD) document that you can use the create the XPath expression. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.

   If you open a DTD or XSD document, the wizard generates a sample XML document from the DTD or schema you selected. You use the generated XML document to create the XPath expression.

   After you retrieve the document using the XML Finder, the content of the document appears in the XPath wizard.

Figure 8-3 Sample XML Document



Document Type: invoice1.dtd

Document Location: Invoice2

XPath Expression:

```
<?xml version="1.0"?>
<!DOCTYPE invoice SYSTEM "invoice1.dtd">
<invoice>
    <header>
        <invoiceNumber e-dtype="number">123</invoiceNumber>
        <date e-dtype="date">042900</date>
        <billTo>
            <custNum e-dtype="number">1234765</custNum>
            <name>
                <first e-dtype="string">Steve</first>
                <last e-dtype="string">Hoffman</last>
```

The Document Type field shows you the type of the currently open document, and the Document Location field shows you the path of the document.

3. Select the markup or content for an element, attribute, or node that you want to be the target of the XPath expression.

   The wizard generates an XPath expression based on your selection, and displays it in the XPath Expression field.

4. Put the XPath expression you created in the Workspace by clicking the Pin button

   The Workspace provides a convenient area where you can hold all the XPath expressions you create during a session of the Studio.

Every time you create an XPath expression, you can pin it to the Workspace. This way, you can create multiple expressions for a single XML document, and then test them later to ensure they return the correct result.

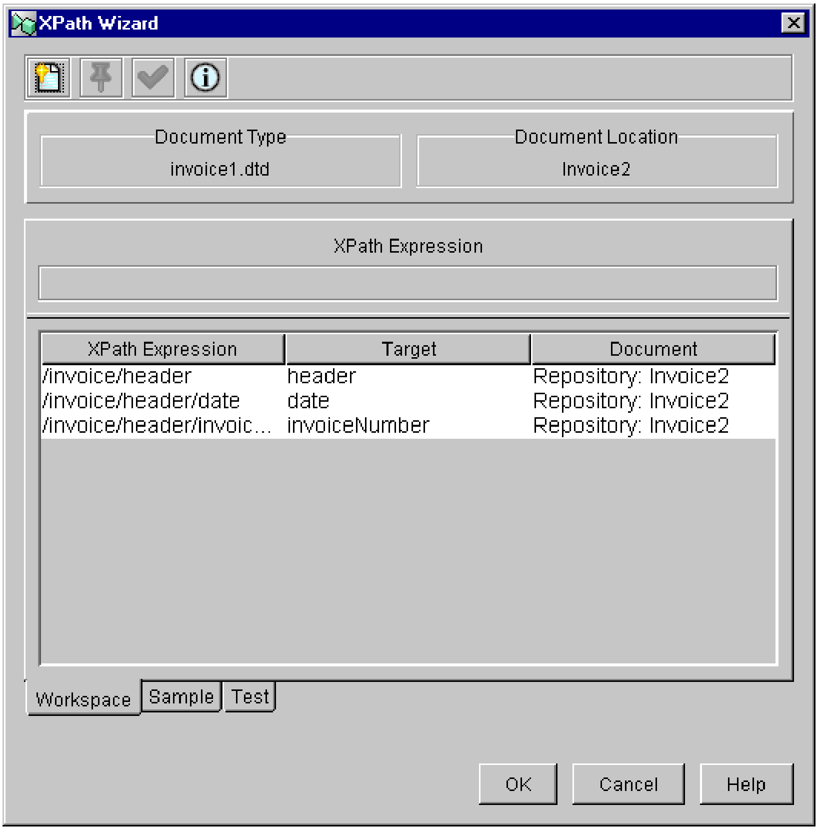
5. To view the expressions you create, select the Workspace tab. For details, see “Viewing XPath Location Expressions” on page 8-25.
6. To test the expressions you create, select the Test tab. For details, see “Testing XPath Location Expressions” on page 8-27.
7. Click OK to return the expression to the Expression Builder.

Viewing XPath Location Expressions

After you create your XPath location expressions and pin them to the Workspace, you can view them in the Workspace.

To view the XPath expressions you created, select the Workspace tab.

Figure 8-4 XPath Expressions in the Workspace



XPath Expression	Target	Document
/invoice/header	header	Repository: Invoice2
/invoice/header/date	date	Repository: Invoice2
/invoice/header/invoic...	invoiceNumber	Repository: Invoice2

The Workspace shows you the following information:

- XPath Expression — lists the expressions you create during a session of the Studio.
- Target — lists the element or attribute in the XML document that is the target of the XPath expression
- Document — lists the source document from which the XPath expression was created

In the Workspace, you can select an expression to test. For details, see “Testing XPath Location Expressions” on page 8-27.

Testing XPath Location Expressions

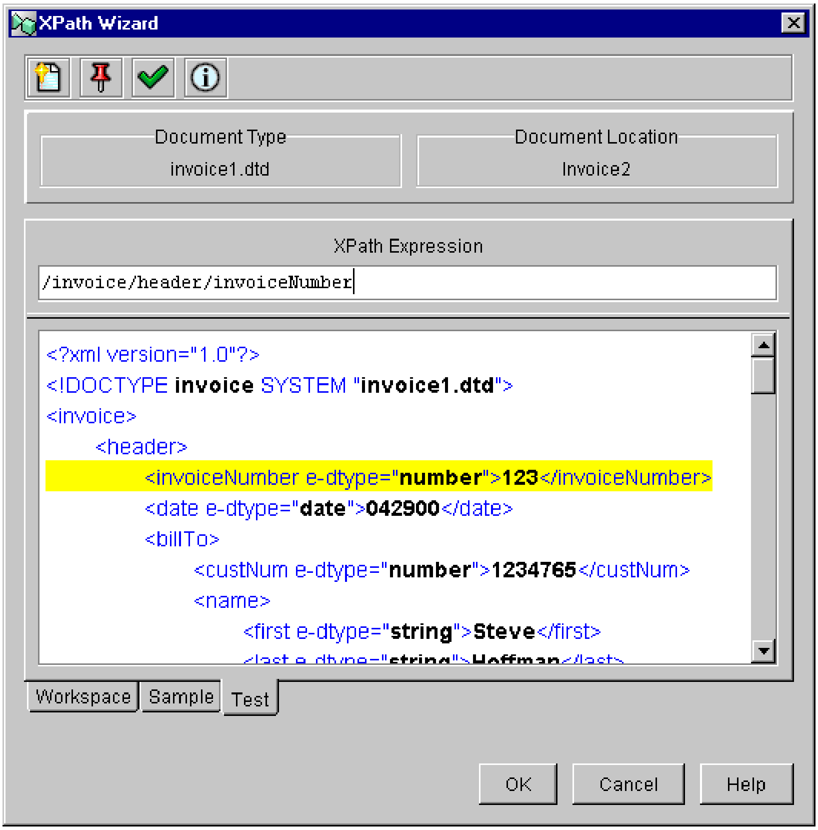
After you create your XPath location expressions, you can test them against a target XML document.

To test an XPath expression:

1. In the Workspace, select the expression you want to test.
2. Select the Test tab.

   The Test area shows you the XML document with the target of the XPath expression highlighted.

Figure 8-5 Target of the XPath Expression



Document Type: invoice1.dtd

Document Location: Invoice2

XPath Expression: /invoice/header/invoiceNumber

```
<?xml version="1.0"?>
<!DOCTYPE invoice SYSTEM "invoice1.dtd">
<invoice>
    <header>
        <invoiceNumber e-dtype="number">123</invoiceNumber>
        <date e-dtype="date">042900</date>
        <billTo>
            <custNum e-dtype="number">1234765</custNum>
            <name>
                <first e-dtype="string">Steve</first>
                <last e-dtype="string">Hoffman</last>
```

If you need to, you can modify the XPath expression by editing it in the XPath Expression field.

To test the expression again, click the Checkmark button. The new result of the expression is highlighted in the XML document.

Creating and Testing XPath Functions

The XPath function library provides functions that you can use to perform operations on the function arguments. Using the XPath wizard, you can create the following types of XPath functions:

- String functions that perform operations on the string contents of nodes and attributes, such as accessing, extracting, and combining
- Boolean functions that perform Boolean tests or operations on their arguments, and that return a value based on the test or operation
- Number functions that return numeric results, such as a count of nodes in an document, or the sum of numeric content
- Node set functions that return information about individual nodes or node sets

To create an XPath function:

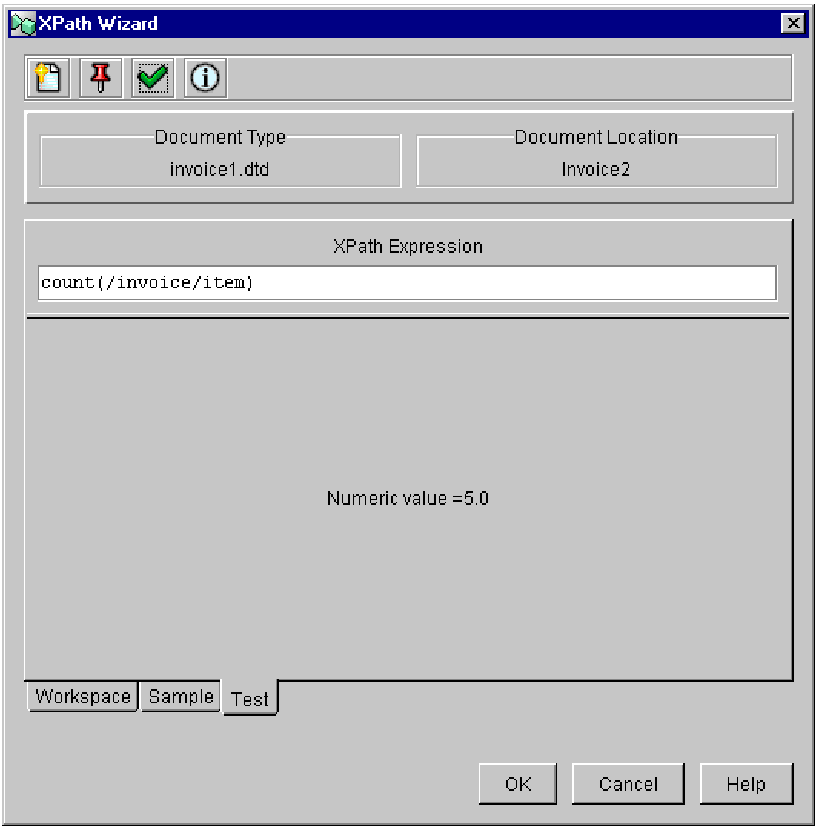
1. Select the Sample tab to open the Sample area where you can create XPath functions.
2. Click the Open button to open the XML Finder dialog box. Use the XML Finder dialog box to retrieve an XML, DTD, or schema (XSD) document for which you want to create the XPath function. For details about the XML Finder, see "Locating XML Entities Using the XML Finder" on page C-2.
3. Select the markup or content for an element, attribute, or node that you want to be part of the XPath function.
4. Select the Test tab.
5. Complete the XPath function in the XPath Expression field.

Alternatively, after you open the document for which you want to create the XPath function, you can select the Test tab and enter the entire XPath function in the XPath Expression field.

To test the function:

1. Click the Checkmark button. The return value of the function is displayed as a literal in the Test area.

Figure 8-6 Result of an XPath Function



count(/invoice/item)

Numeric value =5.0

2. If necessary, modify the XPath function by editing it in the XPath Expression field.

3. To test the function again, click the Checkmark button. The new return value of the function is displayed in the Test area.

To view all the functions you create and pin during a Studio session, select the Workspace tab.

To return the function to the Expression Builder, click OK.

Using XPath Expressions and Functions From the Expression Builder

If you have an XPath location expression or function in the Expression Builder, you can pass it to the XPath wizard, and then use all the functions in the wizard to edit and test the expression or function.

To pass an expression or function from the Expression Builder to the XPath wizard:

1. In the Expression Builder, display the XPath expression or function you want to pass to the wizard.
2. Click XPath Wizard.

   The XPath expression or function is passed to the XPath wizard, and appears as the first item in the XPath Expression list in the Workspace.

CHAPTER

9 Handling Workflow Exceptions

The following sections explain how to handle internally and externally generated workflow exception conditions:

- Workflow Exception Handling Overview
- Defining Exception Handlers
- Using Exception Handling Actions
- WebLogic Process Integrator Error Messages

Workflow Exception Handling Overview

The workflow exception handling facility enables you to generate, trap, and respond to internally and externally generated exception conditions at run time. Exceptions may be abnormal conditions that you define within the workflow, or they may be typical run-time server exceptions that you trap and respond to accordingly.

Exception handling within WebLogic Process Integrator is performed at the workflow level, rather than at the task level. All workflow template definitions have at least one exception handler, the system exception handler. The system exception handler, is by default, the initial exception handler and is invoked whenever an exception occurs. The system exception handler responds to exceptions by marking the active transaction for rollback only and rethrowing the exception to the client.

The initial exception handler is the handler that is active as soon as WebLogic Process Integrator attempts to create a workflow instance from the workflow template definition. The workflow template definition can change its active exception handler throughout the course of the workflow by using the Set Exception Handler action. Once the workflow invokes the Set Exception Handler action, the initial exception handler becomes inconsequential. The concept of the "initial exception handler" is required, since exceptions can occur prior to a workflow invoking the Set Exception Handler action. For example, exceptions can occur during the initialization of variables in a Start node.

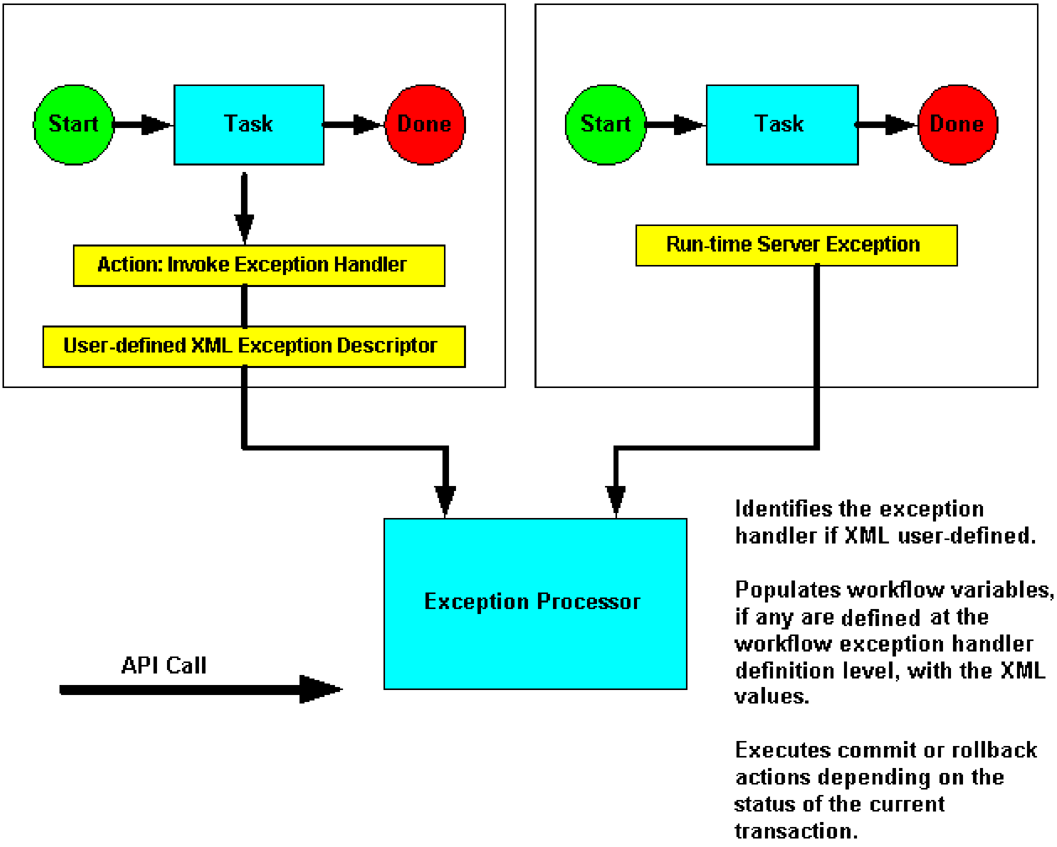
The workflow exception handling facility allows you to define customized exception handlers to act as the initial exception handler and specify actions to be executed in response to exceptions. You set such exception handlers using the Set Workflow Exception Handler action; an exception handler, when set, persists through the workflow instance until the next Set Workflow Exception Handler action is reached within that workflow or until the workflow completes (whichever occurs first).

At any point in its life cycle, a workflow instance has a “current” exception handler. You can change the current exception handler at run-time using the Set Workflow Exception Handler action. You invoke custom exception handlers through use of the Invoke Exception Handler action. Similarly, applications such as custom worklist interfaces can invoke a user-defined exception handler through the following API method: `Worklist.invokeWorkflowErrorHandler`. Further, you can define an XML document within the Invoke Exception Handler action, allowing workflow variables to be populated by values from this XML document.

Exception handlers have commit and rollback processing paths. You specify certain actions within a workflow to be executed in each of these paths. When an exception occurs, if the currently active transaction is marked for rollback only, the exception handler executes the rollback processing path for the transaction. If the transaction is not marked for rollback only, the exception handler executes its commit path.

The following figure illustrates the workflow exception handling process within WebLogic Process Integrator.

Figure 9-1 Workflow Exception Handling


Start
Task
Done
Action: Invoke Exception Handler
User-defined XML Exception Descriptor
Start
Task
Done
Run-time Server Exception
Exception Processor
API Call
Identifies the exception handler if XML user-defined.
Populates workflow variables, if any are defined at the workflow exception handler definition level, with the XML values.
Executes commit or rollback actions depending on the status of the current transaction.

Defining Exception Handlers

Use the exception handler definition facility to define custom exception handlers to act as the initial exception handler and specify actions to be executed in response to exceptions. If you do not define any exception handlers, the system exception handler is, by default, used. The system exception handler responds to exceptions by rolling back the transaction and rethrowing the exception.

The initial exception handler is the handler that is active as soon as WebLogic Process Integrator attempts to create a workflow instance from the workflow template definition. The workflow template definition can change its active exception handler throughout the course of the workflow by using the Set Exception Handler action.

Note: Once defined, an exception handler cannot be deleted if it is referenced by either the Invoke Exception Handler action or the Set Workflow Exception Handler action. See "Using Exception Handling Actions" on page 9-7.

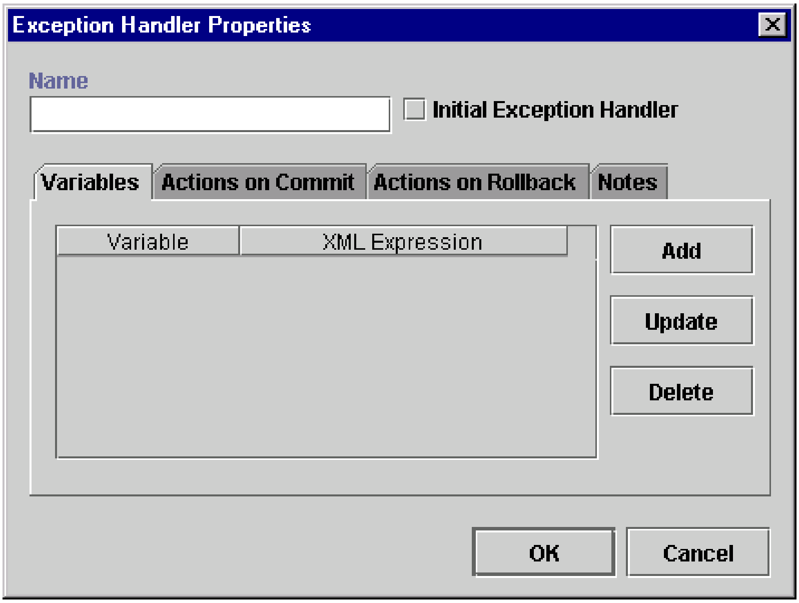
To define an exception handler for a workflow template definition within WebLogic Process Integrator, proceed as follows:

1. In the folder tree, right-click the Exception Handlers folder (located within the selected workflow template definition folder), and select Create Exception Handler to display the Exception Handler Properties dialog box.

   Note: You can also define exception handlers by accessing the Exception Handlers tab of the Template Definition Properties dialog box. (See "Specifying Workflow Template Definition Properties" on page 5-6.)

   The Exception Handler Properties dialog box allows you to define actions for commit and rollback exception situations. You can also define workflow variables to be populated by values from the XML message defined within the Invoke Exception Handler action.

Figure 9-2 Exception Handler Properties Dialog Box



2. Enter a meaningful and easily identifiable name for the new exception handler in the Name field.

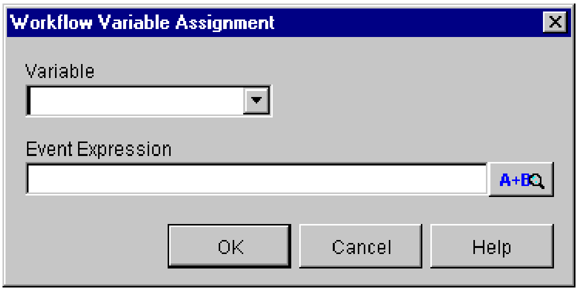
3. Optionally select the Initial Exception Handler checkbox to mark the exception handler as the default initial exception handler in the workflow. When a workflow encounters an exception, this is the first exception handler that is called.

4. In the Variables tab, assign workflow variables to be populated by values from XML message defined within the Invoke Exception Handler action or the API.

5. Click Add to display the Workflow Variable Assignment dialog box and assign XML expressions to the variables assigned to the workflow. The variables in the pull-down list are those defined in the Variables Properties dialog box. (See “Defining Variables” on page 6-3.)

Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

Figure 9-3 Workflow Variable Assignment Dialog Box



6. In the Actions on Commit tab, assign actions to be performed if the current active transaction is not marked for rollback only.

7. In the Actions on Rollback tab, assign actions to be performed if the currently active transaction is programmatically marked for rollback only. The workflow is rolled back to the beginning of the currently active transaction. Only the following actions are available: Post XML Event, Call Program, Perform Business Operation, Exit Exception Handler, No Operation, and Make Audit Entry.

   Note: Certain Enterprise Java Beans (EJBs) can either mark a transaction for rollback only by calling a UserTransaction method, or they can throw an unchecked exception across a container boundary. In either of these two cases, WebLogic Server (WLS) rolls back the transaction.

8. In the Notes tab, enter notes pertaining to the exception handler and its assigned actions.

Using Exception Handling Actions

This section defines and describes how to use the following exception handling actions:

- Invoke Exception Handler
- Set Workflow Exception Handler
- Exit Exception Handler

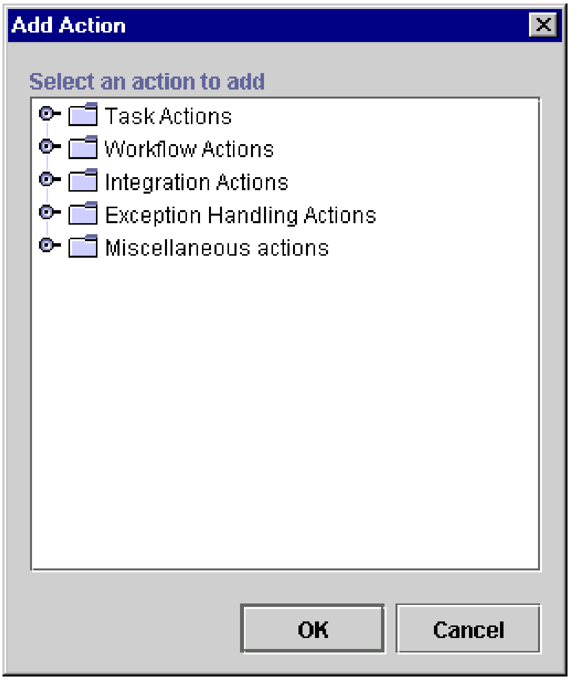
Invoke Exception Handler

Use this action to allow the invocation of a specific exception handler within the workflow. Upon execution of the Invoke Exception Handler action, WebLogic Process Integrator sends the user-defined XML document to the exception processor and invokes the specified exception handler.

To invoke an exception handler, proceed as follows:

1. Access the Add Action dialog box. This dialog box can be accessed from the various Properties dialog boxes in BEA WebLogic Process Integrator. For example, from the workflow drawing area, right-click a task and choose Properties. This displays the Task Properties dialog box.
2. Click Add to display the Add Action dialog box.

Figure 9-4 Add Action Dialog Box

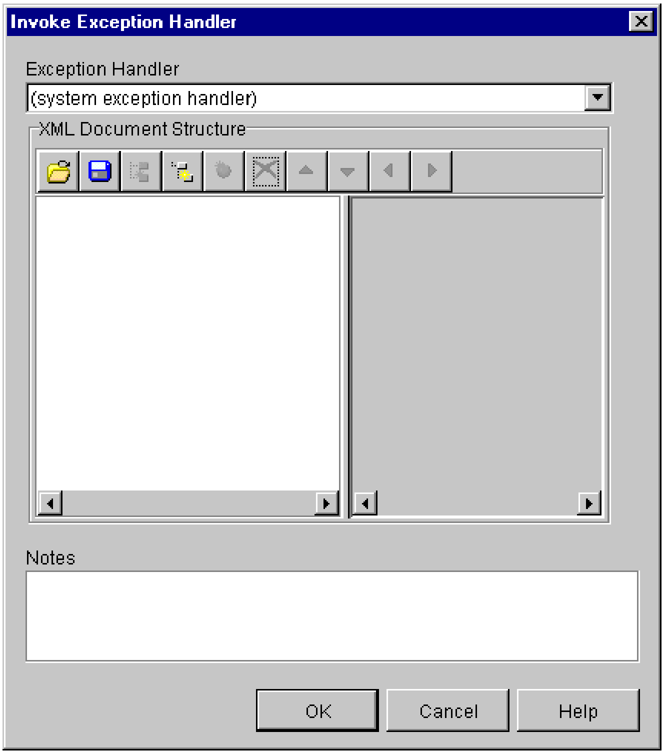


Note: If a plug-in is defined for actions, this dialog box contains the new action. For more information about plug-ins, see Chapter 3, "Working with Plug-Ins."

3. Under Exception Handling Actions, select Invoke Exception Handler and click OK to display the Invoke Exception Handler dialog box.

   Note: You cannot invoke an exception handler from within another exception handler. Therefore, if you access the Add Action dialog box from within an exception handler, the Invoke Exception Handler action is not available.

Figure 9-5 Invoke Exception Handler Dialog Box



Exception Handler: (system exception handler)

4. The Invoke Exception Handler dialog box contains the following:

   - Exception Handler — specify the exception handler to invoke. If no custom-defined exception handlers exist, the system exception handler, by default is used and invoked whenever an exception occurs. The system exception handler is available for use at all times, even if custom handlers are defined.
   - XML Document Structure — create an XML document with elements, subelements, and attributes. Values can be entered for each element or attribute of the XML document. Creating an XML document within this action is optional. This action can be used just to invoke an exception handler and not necessarily send information through XML.

     To compose the XML document, perform the actions listed in the following table. The XML document you define is stored in the workflow template definition.

To . . .	Perform This Action
Retrieve an existing XML document for editing	Click the Import button to open the XML Finder dialog box. Use the XML Finder to retrieve an existing XML document that you can edit. For details about the XML Finder, see "Locating XML Entities Using the XML Finder" on page C-2.
Add the first element of the XML message	Click the Add Child button
Change the default element name	Click the element and wait, or triple-click the element. The element name becomes an entry field. Change the element name as appropriate, then press Enter. To cancel the change, press Esc.
Add a sub-element to an element	Select the element, then click the Add Child button.
Add an element at the same level as another element	Select the element, then click the Add Sibling button.
Add an attribute to an element	Select the element, then click the Add Attribute button.

To . . .	**Perform This Action**
Change the default attribute name	Select the attribute, then double-click the attribute. The attribute name becomes an entry field. Change the attribute name as appropriate, then press Enter. To cancel the change, press Esc.
Change the order of the elements	Select the element you want to move, and use the arrow keys as follows: ■ To move the element up or down a level, click the Move Left or Move Right arrow. ■ To move the element up or down within a level, click the Move Up or Move Down arrow
Change the content of a leaf element or attribute	In the right panel, double-click the line next to the element or attribute for which you want to change the content. The line becomes an entry field. Type the content. To cancel the change, press Esc. **Note:** The syntax for the content of a leaf element or an attribute must follow the same rules that are used for workflow expressions. For example, string literals must be enclosed in single or double quotes. If the syntax is incorrect, a red X appears next to the entry. For details about defining workflow expressions, see Chapter 8, “Using Expressions and Conditions.”
Delete an element or attribute	In the left panel, select the element or attribute you want to delete, and click the Delete button.
Save the XML message	Click the Export button to open the XML Finder dialog box. Use the XML Finder to save the XML message. For details about the XML Finder, see “Locating XML Entities Using the XML Finder” on page C-2.

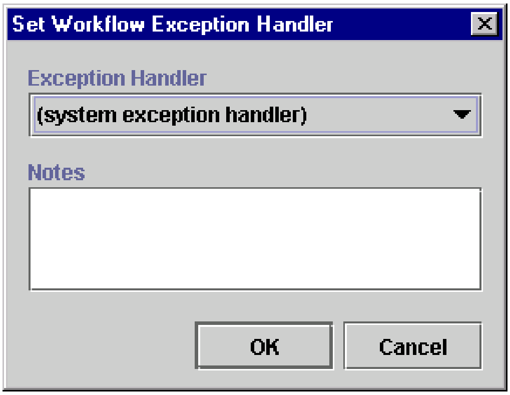
- Notes — a free-form text box for notes about the action.

Set Workflow Exception Handle

Use this action to make a specified exception handler the active exception handler for a workflow template definition. If this action is not used in the workflow template definition and no other exception handler is defined and marked as the initial exception handler, the system exception handler, by default, is used and invoked whenever an exception occurs. The system exception handler responds to exceptions by rolling back the transaction and rethrowing the exception.

The exception handler you set for the workflow persists throughout the instance of the workflow until a subsequent Set Workflow Exception Handler action is executed within the same workflow.

Figure 9-6 Set Workflow Exception Handler



Set Workflow Exception Handler

Exception Handler: (system exception handler)

Notes:

View or enter information in the Set Workflow Exception Handler dialog box.

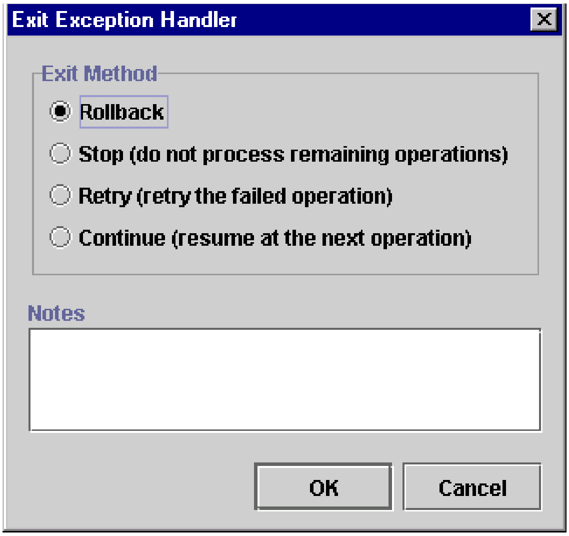
- Exception Handler — Displays a list of exception handlers to choose from. If you have not defined an exception handler, the list only displays the system exception handler.
- Notes — Free-form text box for notes on the action.

Exit Exception Handler

Use this action to exit an exception handler using one of four methods: rollback, stop, retry, or continue.

Note: This action is only displayed in the Add Actions dialog box when this dialog box is accessed from the Actions on Commit and Actions on Rollback tabs of the Exception Handler Properties dialog box. See "Defining Exception Handlers" on page 9-4.

Figure 9-7 Exit Exception Handler



● Rollback
○ Stop (do not process remaining operations)
○ Retry (retry the failed operation)
○ Continue (resume at the next operation)

View or enter information in the Exit Exception Handler dialog box.

- Exit Method — Select one of the following exception handler exit methods:
  - Rollback — Marks the user transaction for rollback only, returning the workflow to the state in which it existed prior to the start of the transaction. The exception is propagated to the client (if any) as a WorkflowException.

  Note: In the Rollback path, the only exit option available in the Exit Exception Handler action is Rollback. When a transaction is rolled back, there are no other options other than rollback.

Certain Enterprise Java Beans (EJBs) can either mark a transaction for rollback only by calling a UserTransaction method, or they can throw an unchecked exception across a container boundary. In either of these two cases, WebLogic Server (WLS) rolls back the transaction.

- Stop — Exits the exception handler and does not execute any follow-on actions.
- Retry — Exits the exception handler and attempts to retry the failed operation. (The failed operation is an action or the setting of a variable)
- Continue — Stops the execution of the exception handler and attempts to continue the execution of the workflow at the next operation. (The next operation is the next action or the next variable to be set.)

WebLogic Process Integrator Error Messages

The following is a list of error messages that may be encountered in WebLogic Process Integrator and handled by the exception handler.

To see a list of Workflow Attribute values, see Chapter 8, “Using Expressions and Conditions.” Four error-related values allow you to obtain context about the cause of an exception. These attributes allow the exception handler to interrogate the following information:

- The exception class name.
- The WebLogic Process Integrator error number.
- The error message text.
- The severity code (which also tells you if the exception handler was called—via action or API—or was raised as a result of a caught exception).

Table 9-1 WebLogic Process Integrator Error Messages

ErrorNumber	ErrorText
0	Unknown error
1	System error
2	Workflow error
3	Workflow warning
4	Nested exception is:
5	The server was unable to complete your request.
6	The server was unable to complete your request.
7	This workflow cannot be modified, because it is complete.
8	This task's properties do not allow it to be marked done.
9	This task's properties do not allow it to be unmarked done.
10	This task's properties do not allow it to be reassigned.
11	This task's properties do not allow it to be modified.
12	Cannot take this task, because it is already done.
13	Cannot assign this task, because it is already done.
14	Cannot execute this task, because it is already done.
15	Cannot execute this task, because it is inactive.
16	Cannot execute this task, because the workflow is complete.
17	Cannot execute this task, because it is not assigned to you.
18	Source and destination of reroute must be different.
19	The effective date must be on or before the expiry date.
20	Source user is already rerouted during the period specified.
21	Specified reroute would create a circular reference.

ErrorNumber	ErrorText
22	Task names must be unique within a workflow.
23	"{0}" is already in use as an organization name.
24	Role names must be unique within an organization.
25	The specified user "{0}" is already defined.
26	Report names must be unique.
27	Variable names must be unique.
28	Workflow template names must be unique.
29	Workload graph names must be unique.
30	"{0}" is already used as business calendar name.
31	User "{0}" is currently logged on and cannot be deleted.
32	Cannot delete user "{0}", because there are tasks assigned to the user. Please reassign all tasks first.
33	You do not have permission to maintain users.
34	You do not have permission to maintain roles.
35	You do not have permission to maintain organizations.
36	You do not have permission to define workflows.
37	You do not have permission for workflow monitoring.
38	You do not have permission to do task rerouting.
39	You do not have permission to modify the server configuration.
40	The variable "{0}" cannot be deleted, because it is referenced by the ID expression or the trigger definition.
41	The variable "{0}" cannot be deleted, because it is referenced by one or more actions.
42	The variable "{0}" cannot be deleted, because it is referenced by one or more decisions.
43	Variable names cannot be blank.

ErrorNumber	ErrorText
44	Cannot delete role "{0}", because there are tasks assigned to the role. Please reassign all tasks first.
45	The workflow "{0}" cannot be deleted, because it is referenced by one or more actions.
46	The task "{0}" cannot be deleted, because it is referenced by one or more actions.
47	The role "{0}" cannot be deleted, because it is referenced by one or more actions.
48	The user "{0}" cannot be deleted, because it is referenced by one or more actions.
49	This workflow cannot be modified, because it is suspended.
50	Cannot delete a business calendar's template rules.
51	No business calendar ID specified in expression.
52	No XML defined for calendar.
53	The system could not find the specified business calendar "{0}".
54	Business calendar "{0}" has no rules defined for the year {1}.
55	The system could not find the processor for business calendar "{0}".
56	{0}, {1}: unexpected token "{2}" in business calendar rule.
57	{0}, {1}: unexpected character "{2}" in business calendar rule.
58	The timezone identifier "{0}" is not recognized.
59	EJB home environment not set.
61	Unable to get initial context: environment may be invalid; {0}.
62	Unable to load class object for home interface of type "{0}"; {1}.
63	Unable to load class object for remote interface of type "{0}"; {1}.
64	No object is bound to "{0}" in specified context; {1}.
65	The object bound to "{0}" does not implement home interface "{1}".
66	No home method was found on "{0}" matching "{1}"; {2}.
67	No remote method was found on "{0}" matching "{1}"; {2}.

ErrorNumber	ErrorText
68	Unable to load class for method return value of type "{0}"; {1}.
69	No remote object(s) were found.
70	EJB "{0}" method invocation failed; {1}.
71	"{0}" object returned from method invocation not of expected type "{1}".
72	Wrong number of parameters for method: expected {1}, found {2}.
73	Unable to load class (wrapped if primitive) for parameter {0} of type "{1}"; {2}.
74	Unable to load class (unwrapped if primitive) for parameter {0} of type "{1}"; {2}.
75	Value for parameter {0} cannot be cast to required type "{1}".
76	The home method did not return a session bean.
77	Workflow template (ID={0}) not found.
78	Workflow template definition (ID={0}) not found.
79	The system could not find an active, effective template definition to start.
80	XML syntax error at line {0}, column {1}.
81	Unable to load class object for "{0}"; {1}.
82	No matching constructor was found on "{0}"; {1}.
83	Creation of new "{0}" failed; {1}.
84	No method was found on "{0}" matching "{1}"; {2}.
85	"{0}" method invocation failed; {1}.
86	The workflow template is currently locked by {0}.
87	The interval “from” date is invalid.
88	The interval “to” date is invalid.
89	The date "{0}" is invalid.
90	Invalid month name: "{0}".

ErrorNumber	ErrorText
91	Invalid day name: "{0}".
92	The system could not find the specified task instance: "{0}".
93	Mandatory input variable "{0}" not set.
94	The system could not find the specified join instance: "{0}".
95	The system could not find the specified variable instance: "{0}".
96	The system could not find the specified organization: "{0}".
97	The system could not find the specified role: "{0}".
98	The system could not find the specified user: "{0}".
99	User "{0}" does not belong to organization "{1}".
100	The system could not add user "{0}" to organization "{1}".
101	The system could not add user "{0}" to role "{1}".
102	The system could not remove user "{0}" from organization "{1}".
103	The system could not remove user "{0}" from role "{1}".
104	The specified user "{0}" does not belong to the organization within which the role "{1}" is defined.
105	No security realm has been installed.
106	The installed security realm "{0}" is not listable.
107	The security realm "{0}" is not manageable.
108	Unable to connect to database.
109	The role "{0}" is empty.
110	Fixup error: missing reference "{0}".
111	Invalid interval unit: "{0}".
112	Illegal type conversion: from "{0}" to "{1}".
113	Cannot instantiate a workflow, because the template definition is inactive.

ErrorNumber	ErrorText
114	The system could not find the target task "{0}".
115	The system could not find the target event "{0}".
116	The system could not find the target action "{0}".
117	The system could not find the specified business operation "{0}".
118	Error calling program: "{0}" with arguments "{1}".
119	User "{0}" does not have an e-mail address.
120	User "{0}" in role "{1}" does not have an e-mail address.
121	The routing table failed to identify a suitable assignee.
122	No workflow organization defined for user "{0}".
123	Wrong start date expression: "{0}".\n"(1)".
124	The system could not find the error handler "{0}".
125	An exception occurred during error handler processing.
126	An error handler exceeded the maximum number of retries allowed.
127	The application or workflow instance invoked an error handler.
128	The system could not find the specified workflow instance: {0}.
129	An error occurred when parsing an XML document.
130	The definition was created with a later of version of the product. Upgrade your WebLogic Process Integrator Studio to version {0} or later.

CHAPTER

10 Monitoring Workflows

The following sections discuss workflow monitoring:

- Using the Workflow Monitoring Features
- Viewing Workflow Status
- Viewing Workflow Variables
- Deleting Workflow Instances
- Using Workload Reports
- Using Statistics Reports

Using the Workflow Monitoring Features

Use the BEA WebLogic Process Integrator workflow monitoring features to monitor and modify workflows, gather information, and make informed decisions based on available data. You can perform the following tasks:

- Display the status of workflow instances.
- Modify the tasks associated with a workflow instance, for example, reassigning tasks or forcing work to be redone.
- Graphically display the workload status of the system, showing the number of tasks categorized by status (pending, inactive, done, overdue) and user or role.
- View historical statistical data associated with workflows in order to determine bottlenecks and inefficiencies in the business processes.
- View a user or role worklist in order to dynamically maintain workflows.

User Interface

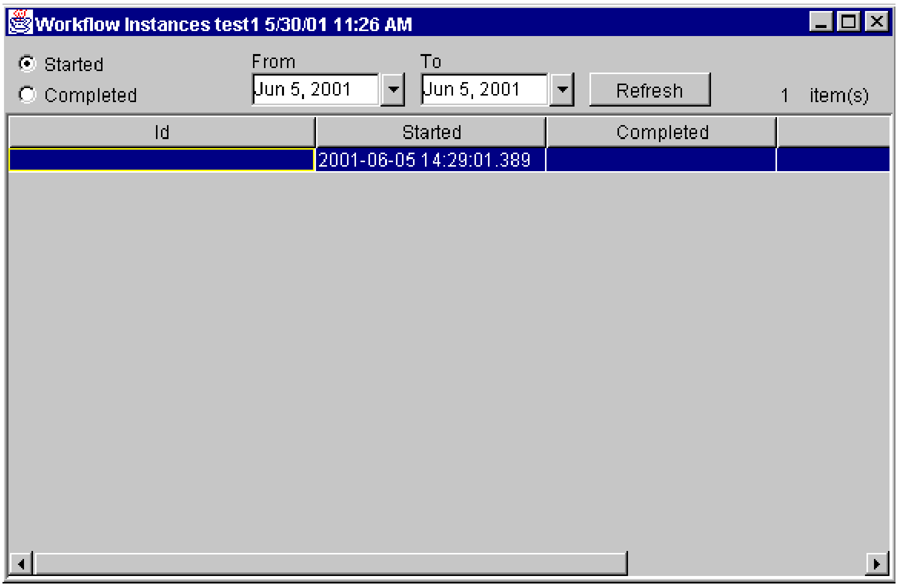
The following WebLogic Process Integrator folder tree items are relevant to the monitoring features:

- Template — shows a list of defined workflow templates. Select a particular workflow template definition from the folder tree, right-click, and choose Instances to display all workflow instances associated with the definition that have actually been executed for the specified organization.
- Workload Report — allows you to create graphical reports based on workflow, task, user or role, and task status
- Statistics Reports — allows you to generate statistical reports based on workflow, task, user or role, and date
- Users and Roles — displays the user and role worklists. Right-click the user or role and choose Open User/Role Worklist.

Workflow Instance Status

A workflow instance is a session of a workflow template definition that has been placed into run-time. To monitor the status of workflow instances, select a workflow template definition within a workflow template in the folder tree, right-click, and choose Instances from the pop-up menu. This displays the Workflow Instance dialog box, which contains status information for the instances of the selected workflow template definition.

Figure 10-1 Workflow Instances Dialog Box



Workflow Instances test1 5/30/01 11:26 AM

● Started
○ Completed

From: Jun 5, 2001
To: Jun 5, 2001

1 item(s)

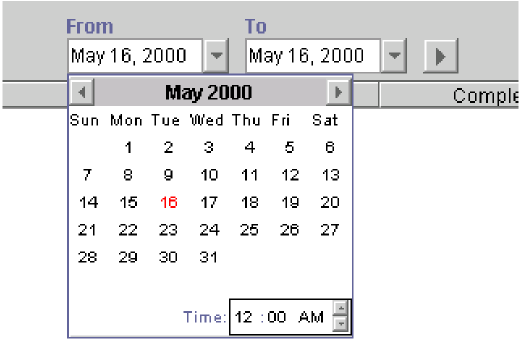
Id	Started	Completed	
	2001-06-05 14:29:01.389		

The Workflow Instances dialog box contains the following:

- Started — if selected, this option finds all workflow instances with a start date in the specified date range. This option is mutually exclusive of the Completed option.
- Completed — if selected, this option finds all workflow instances with a completed date in the specified date range. This option is mutually exclusive of the Started option.

- From — workflow instances started or completed (depending on the selected option) on or after this date will be shown. Select the date from the drop-down list.
- To — workflow instances started or completed (depending on the selected option) on or before this date will be shown. Select the date from the list.

Figure 10-2 Workflow Instance From/To Dates



From: May 16, 2000

To: May 16, 2000

May 2000

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Time: 12 :00 AM

For each workflow shown, the following information is displayed:

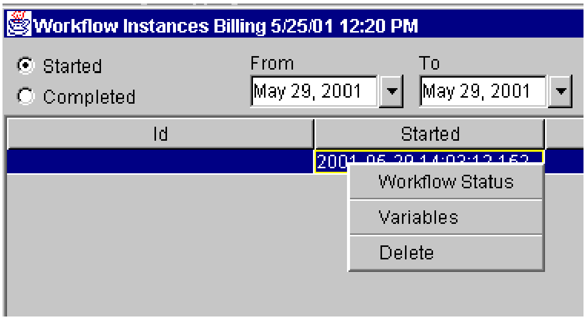
- ID — workflow template definition ID. This is maintained automatically by the workflow engine from the expression specified in the properties at workflow template definition time.
- Started — date the workflow was instantiated.
- Completed — completion date of the workflow. If it is not completed, this column will be blank.

For performance reasons, the BEA WebLogic Process Integrator server only returns 100 items at a time. If there are additional items, a button (+) appears which, when pressed, will retrieve the next 100 items. When no additional items remain, the button is no longer shown.

Viewing Workflow Status

Right-clicking or double-clicking a particular workflow template definition instance in the list of instances in the Workflow Instances dialog box displays a menu. Choose the first item on the menu to display the overall workflow status for a particular instance.

Figure 10-3 Workflow Status Menu Option



Workflow Instances Billing 5/25/01 12:20 PM

● Started
○ Completed

From: May 29, 2001
To: May 29, 2001

Id	Started
	2001-05-29 14:03:13.162

The Workflow Status command displays a window showing either the graphical representation of the workflow with the active task highlighted or a list view of all tasks. The following two icons upper in the right hand side of the dialog box to allow toggling between the graphical representation and the list view.

Table 10-1 Workflow Status Icons

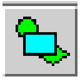
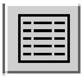
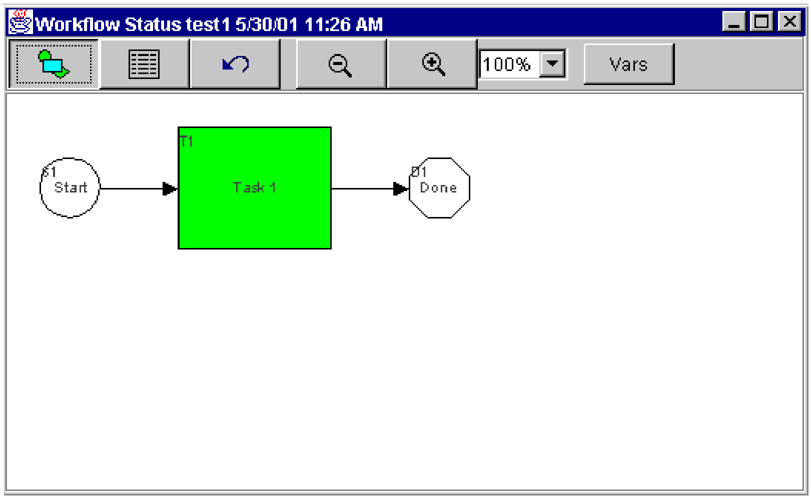
Symbol	Shape Name	Purpose
	Graphical Representation	Displays a graphical representation of the workflow.
	List View	Displays a list view of the workflow.

Table 10-1 Workflow Status Icons

Symbol	Shape Name	Purpose
↶	Refresh	Refreshes the workflow view.

Clicking the graphical representation icon displays a graphical representation of a workflow template definition that has been instantiated (is now an instance of the workflow template).

Figure 10-4 Graphical Representation of Workflow

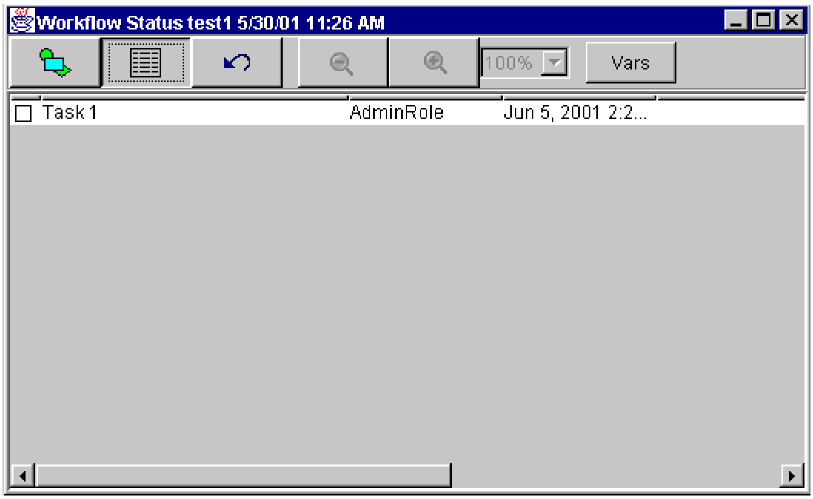


In the graphical representation, the active tasks are green, the executed tasks are gray, and the inactive tasks are white.

ClickingVars displays the Workflow Variables dialog box. For details about this dialog box, see “Viewing Workflow Variables” on page 10-10.

Clicking the list view icon displays a list view of the workflow and its associated tasks.

Figure 10-5 List View of Workflow



□ Task 1 AdminRole Jun 5, 2001 2:2...

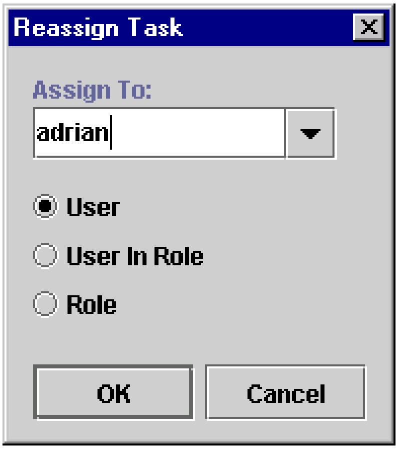
Various properties are displayed for each task within the workflow, depending on the way the list is tailored. Usually, the task name is shown in the list. The first column shown is the task status. A box indicates that the task is pending, or waiting to be executed (it has a start date, but not a completed date). Tasks shown without a box are not yet in a state that allows them to be executed (meaning that they have neither a start date nor a completed date). A box with a check mark in it indicates a task that has already been completed (has a completed date). Furthermore, a pending task is shown with a red box, indicating the task is now overdue (meaning its due date is before the current date).

Right-clicking a task in the list displays a menu with the following options:

- Mark Task as Done — changes a task status to done. When executed, this command sends a message to the BEA WebLogic Process Integrator server to set the task's completed date to the current date, and to then perform the actions defined for the marked done event of the task. Tasks can only be marked done if their properties allow it.
- Unmark Task Done — resets a task status to change it back to pending. When executed, this command sends a message to the BEA WebLogic Process Integrator server to clear the task's completed date. It has no effect on a task that is not complete. Tasks can only be unmarked done if their properties allow it.

- Reassign Task — reassigns a task to another user or role. When executed, this command displays the Reassign Task dialog box, which allows for the selection of a new user or role to assign the task to.

Figure 10-6 Reassign Task Dialog Box



Assign To: adrian

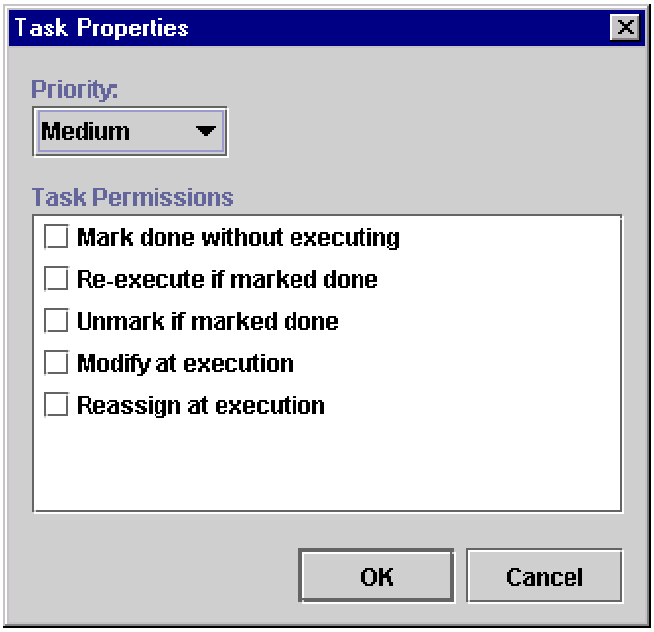
● User
○ User In Role
○ Role

A task can be assigned to a user, a user within a role (determined by the workflow engine so that workload balancing is automatically done), or a role. When a selection is made, a message is sent to the BEA WebLogic Process Integrator server to set the task's assignee to the selected user or role.

Tasks can only be reassigned if their properties allow reassignment of tasks, and the task is not already done.

- Task Properties — display and change properties of the selected task. When executed, the Task Properties dialog box is displayed.

Figure 10-7 Task Properties Dialog Box



Priority: Medium

Task Permissions

- □ Mark done without executing
- □ Re-execute if marked done
- □ Unmark if marked done
- □ Modify at execution
- □ Reassign at execution

The Task Properties dialog box contains the following:

- Priority — select the task priority from the drop-down list: Low, medium, or High.
- Task Permissions — determine what a user can do with a task from the worklist. The permissions are:
  - Mark done without executing — if selected, any end user assigned this task can perform the Mark Task as Done command from the Worklist application. The task's completed date can be set manually by the end user.
  - Re-execute if marked done — if selected, any end user assigned this task can perform the Execute Task command from the Worklist application, even if the task has already been completed.
  - Unmark if marked done — if selected, any end user assigned this task can perform the Unmark Task as Done command on the task from the Worklist application.
  - Modify at execution time — if selected, any end user assigned this task can change the task permissions by selecting Properties from the Task menu of

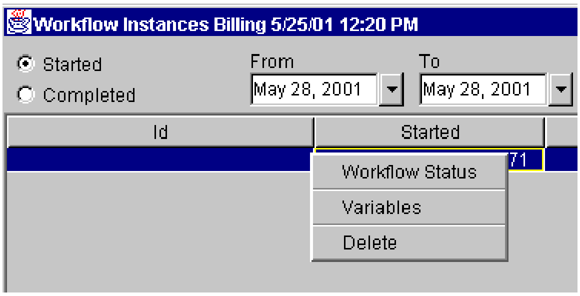
the Worklist application. If this is not selected, then none of the task permissions can be altered.

- Reassign at execution time — if selected, any end user assigned this task can perform the Take and Reassign commands from the Worklist application, meaning they can take the task as their own responsibility or reassign it to another user.

Viewing Workflow Variables

The second option on the pop-up menu, which is displayed by right-clicking the instance, is the workflow variables. This command is used to display a list of all workflow instance variables.

Figure 10-8 Variables Menu Option



Workflow Instances Billing 5/25/01 12:20 PM

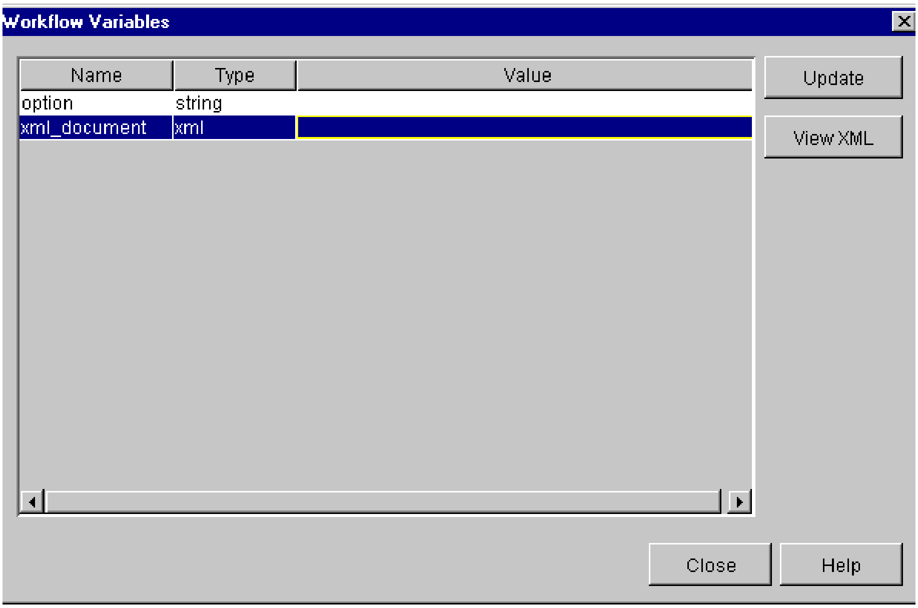
● Started
○ Completed

From: May 28, 2001
To: May 28, 2001

Id	Started
	71

When selected, the Variables menu option displays the Workflow Variables dialog box, which lists the variables of the selected workflow instance.

Figure 10-9 Workflow Variables Dialog Box

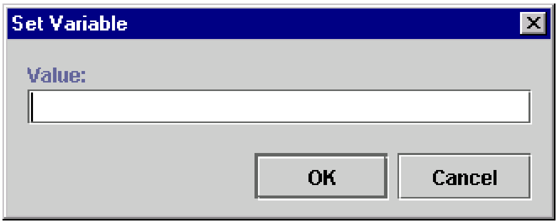


Name	Type	Value
option	string	
xml_document	xml	

The Workflow Variables dialog box displays each variable in a list, showing its name and its current value. Use the buttons in the dialog box as follows:

- Update — select a variable and click Update to display the Set Variable dialog box showing the current value. You can change this value.

Figure 10-10 Set Variable Dialog Box



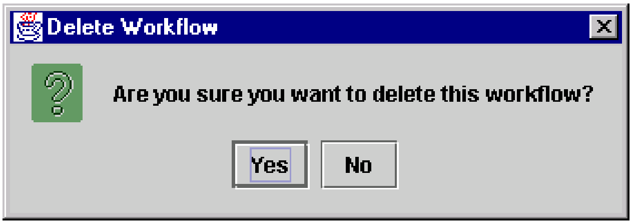
When you save the change, a message is sent to the BEA WebLogic Process Integrator server to reset the variable. Because workflow logic often depends on the value of variables, use caution when manually changing the value of a variable.

- View XML — select an XML-type variable and click View XML to display the XML contents of the variable.

Deleting Workflow Instances

The Delete menu option allows you to delete a workflow instance. Selecting this menu option displays the Delete Workflow dialog box.

Figure 10-11 Delete Workflow Dialog Box



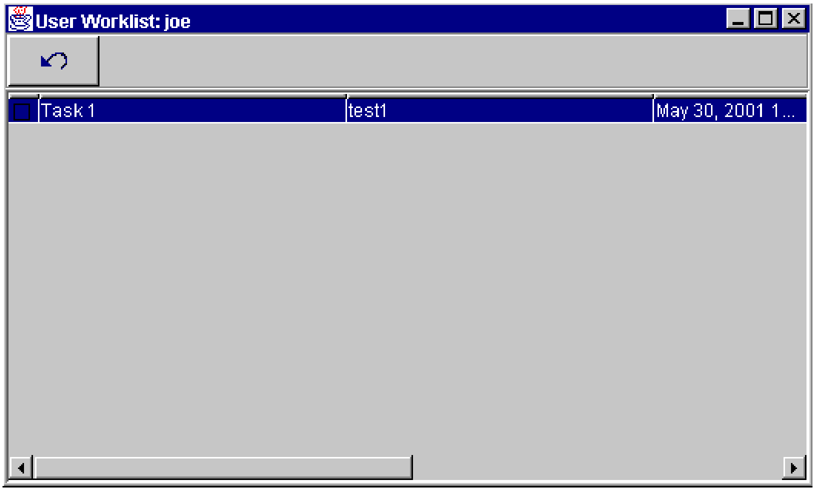
Are you sure you want to delete this workflow?

A positive confirmation sends a message to the BEA WebLogic Process Integrator server prompting the deletion of the selected workflow instance(s).

Viewing User Worklists

In the BEA WebLogic Process Integrator folder tree, right-click a user from the list and select Open User Worklist from the pop-up menu. A list of tasks assigned to the user is displayed in the User Worklist dialog box.

Figure 10-12 User Worklist Dialog Box



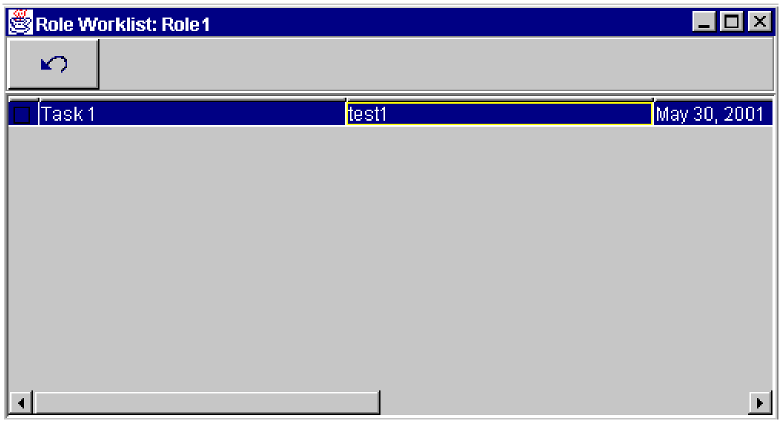
□	Task 1	test1	May 30, 2001 1...

Note: You cannot execute tasks on worklists invoked from within the Studio. To execute tasks from a worklist you must invoke the Worklist client application.

Viewing Role Worklists

In the BEA WebLogic Process Integrator folder tree, right-click a role from the list and select Open Role Worklist from the pop-up menu. A list of tasks assigned to the role is displayed in the Role Worklist dialog box.

Figure 10-13 Role Worklist Dialog Box



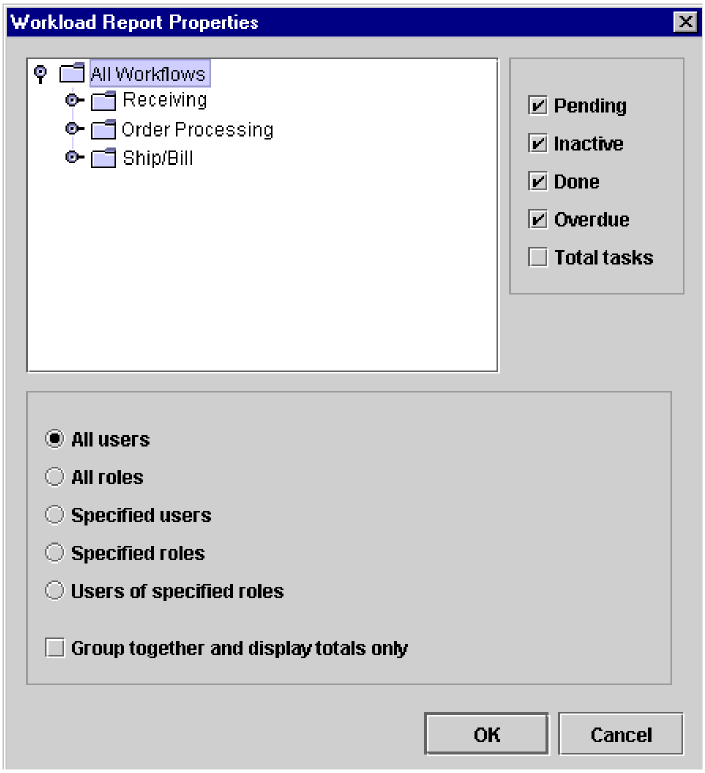
□	Task 1	test1	May 30, 2001

Note: You cannot execute tasks on worklists invoked from within the Studio. To execute tasks from a worklist you must invoke the Worklist client application.

Using Workload Reports

BEA WebLogic Process Integrator allows you to view workload reports, showing the number of tasks, by workflow, task, user or role, and task status. In the BEA WebLogic Process Integrator folder tree, right-click Workload Report, and choose Open from the pop-up menu to display the Workload Report Properties dialog box.

Figure 10-14 Workload Report Properties Dialog Box



☑ Pending

☑ Inactive

☑ Done

☑ Overdue

□ Total tasks

● All users

○ All roles

○ Specified users

○ Specified roles

○ Users of specified roles

□ Group together and display totals only

Compiling Workload Report Information

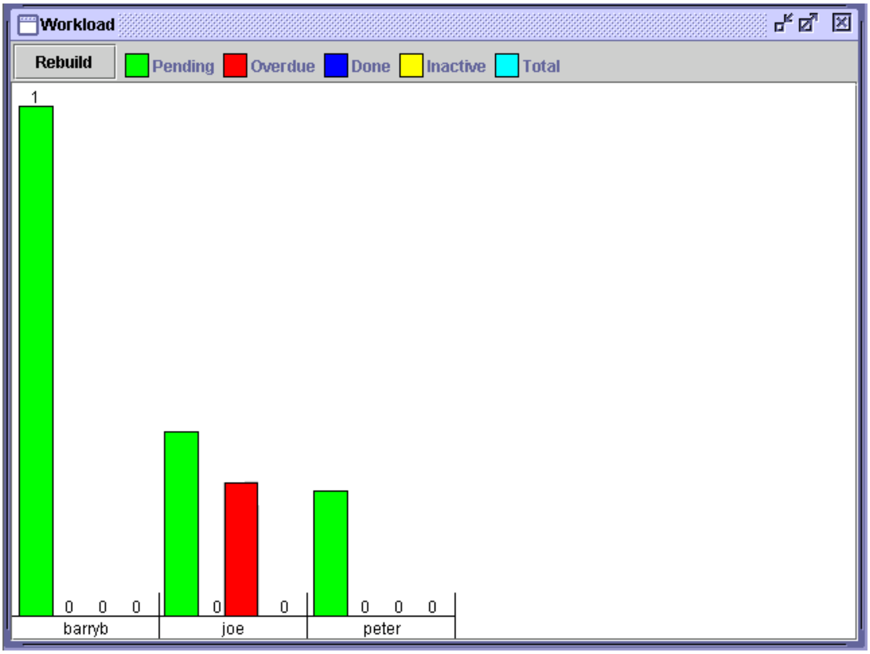
The workload report will display information according to the selections you make in the Workload Report Properties dialog box:

- Workflow Templates folders — select a workflow template to include information pertaining to all workflow template definitions within that workflow template, or expand a workflow template and select only one workflow template definition. Only information within the selected workflow template or workflow template definition will be included in the workload report.
- Pending — displays pending tasks. Pending tasks have been started and not yet completed.
- Inactive — displays inactive tasks. Inactive tasks are those which have not yet been started.
- Done — displays completed tasks.
- Overdue — displays those tasks which are now overdue. Overdue tasks are pending tasks with a due date on or before the current date.
- Total Tasks — displays all tasks, regardless of task state (that is, inactive, pending, and completed tasks are all counted).
- All users — displays workload information for each user.
- All roles — displays workload information for each role.
- Specified users — displays workload information for each the selected users in the displayed users list.
- Specified roles — displays workload information for each of the selected roles in the displayed roles list.
- Users of specified roles — displays workload information for each user who is a member of one of the selected roles in the displayed roles list.
- Group together and display totals only — displays total counts of tasks only for the group identified by the above selected option. If not selected, each individual user or role is shown in the workload report.

Viewing Workload Reports

After you make your workload report selections, click OK to display a graphical representation of the current workload.

Figure 10-15 Workload Report Dialog Box



The contents of the report are broken down by user, role, or totals, depending on the options selected. This display may scroll left and right if there is a lot of information. A bar chart indicates the total number of tasks of various state counted. The legend at the top of the window shows what each color represents, depending on the options selected at design time. For more precise indicators, the actual number of tasks is shown above each bar.

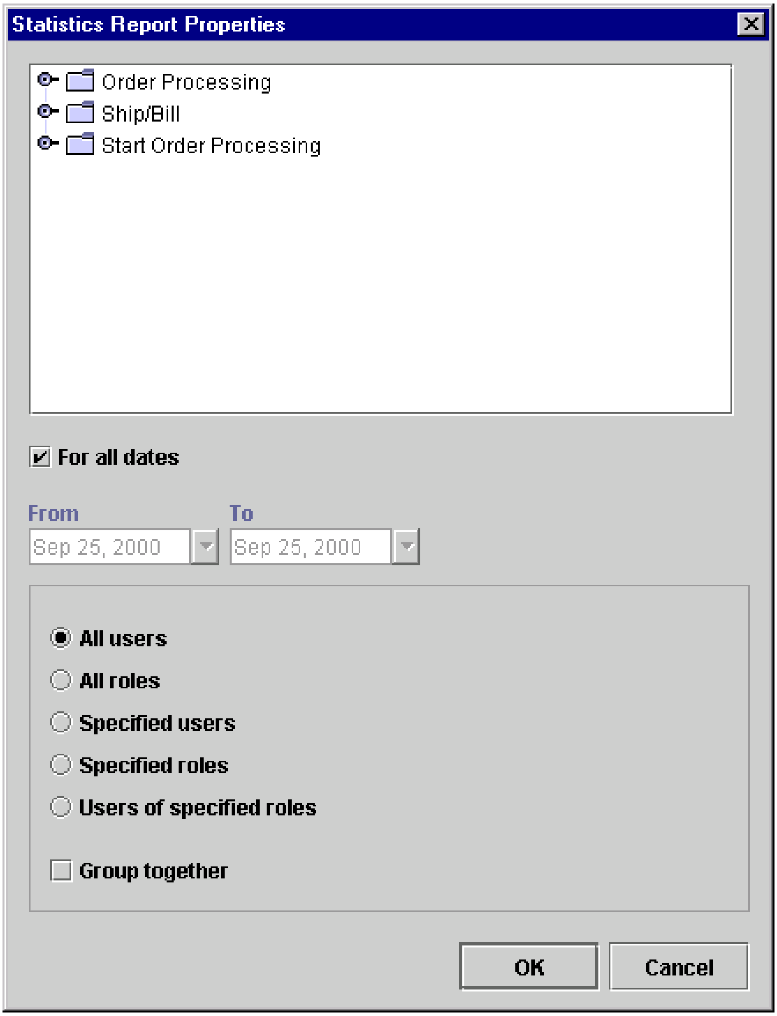
Using Statistics Reports

You can view statistics reports that are based on tasks, users, roles, and so on. A statistics report provides statistical information based on historical data kept for completed workflows. It shows number of tasks completed, total time, average time, minimum and maximum times spent for task. Standard deviation is also shown.

Compiling Statistics Report Information

In the BEA WebLogic Process Integrator folder tree, right-click Statistics Reports and choose Open from the pop-up menu to display the Statistics Report Properties dialog box. The report includes information according to the selections you make in the Statistics Report Properties dialog box.

Figure 10-16 Statistics Report Properties Dialog Box



Statistics Report Properties

- Order Processing
- Ship/Bill
- Start Order Processing

☑ For all dates

From: Sep 25, 2000
To: Sep 25, 2000

● All users
○ All roles
○ Specified users
○ Specified roles
○ Users of specified roles

□ Group together

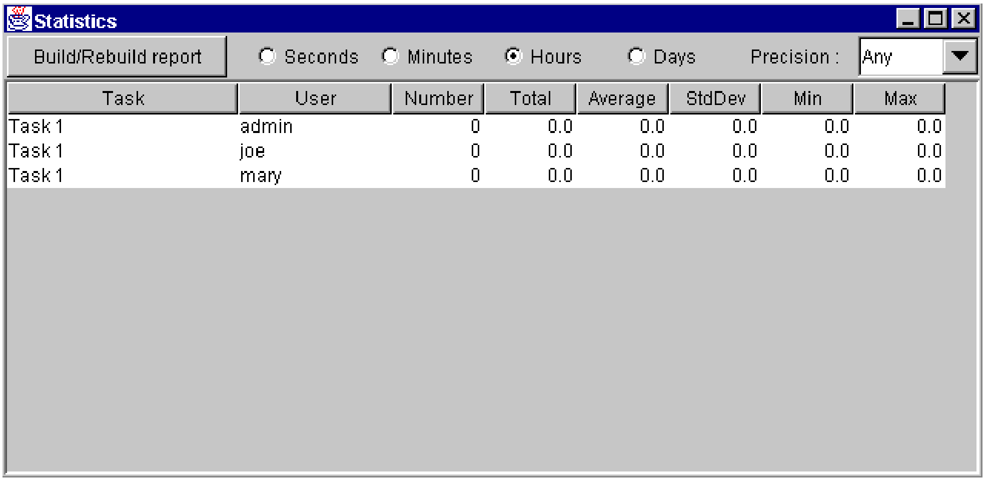
- List of workflow templates — list of available workflow templates. Expand a template to select the workflow template definition to include in the statistics report.
- For all dates — statistical calculations are gathered regardless of date.
- From — all tasks, in completed workflows, completed on or after this date are included in the statistical calculations.

- To — all tasks, in completed workflows, completed on or before this date are included in the statistical calculations.
- All users — displays statistical information for each user.
- All roles — displays statistical information for each role.
- Specified users — displays statistical information for each selected user.
- Specified roles — displays statistical information for each selected role.
- Users of specified roles — displays statistical information for each user who is a member of one of the selected roles in the displayed roles list.
- Group together — the report displays total counts of tasks only for the group identified by the above selected option. If not selected, each individual user or role is shown in the workload report.

Viewing Statistics Reports

When you make your statistics report selections, click OK to display the statistics report.

Figure 10-17 Statistics Report



○ Seconds ○ Minutes ● Hours ○ Days Precision : Any

Task	User	Number	Total	Average	StdDev	Min	Max
Task 1	admin	0	0.0	0.0	0.0	0.0	0.0
Task 1	joe	0	0.0	0.0	0.0	0.0	0.0
Task 1	mary	0	0.0	0.0	0.0	0.0	0.0

The top of the dialog box has several options:

- Build/Rebuild — rebuild the report if you change the units of time or the precision level.
- Seconds, Minutes, Hours, Days — units of time used in the report.
- Precision — drop-down list from which you can select how many decimal places you want in the statistical information.

Several columns are displayed in the report, depending on the options selected at report design time:

- Task — name of the task being measured.
- User — user or role assigned to the task. If the Group together option was selected at report design time, this column is not shown, because the results are totaled for all users or roles.
- Number — total number of tasks completed.
- Total — total amount of time spent on doing the task. This is the time interval between the task start date and completed date.
- Average — average amount of time spent performing this task.
- StdDev — standard deviation from the average.
- Min — minimum amount of time spent performing this task.
- Max — maximum amount of time spent performing this task.

Because the report definition is stored separately from the results, the statistics report can be run at any time.

A Understanding Event-Driven Processing

The following sections explain event-driven processing, and how to trigger events:

- What Is an Event?
- What Is an Event Key?
- What Is an Event Condition?
- Defining an Event Key

What Is an Event?

An event is an asynchronous notification from another workflow or from an external source, such as another application. An event notification takes the form of an XML message. You can define an event to start a workflow, to initialize variables, to activate a node in the workflow, or to execute an action.

To start a workflow using an event, you define an event-type Start node that waits until it receives the appropriate trigger before starting the workflow. Similarly, to initialize variables, to activate a node in the workflow, or to execute an action using the appropriate trigger, you define an Event node. For details, see “Defining a Start Node Using an Event Trigger” on page 6-14 and “Defining Events” on page 6-24.

The event that starts a workflow or triggers an Event node is the receipt of an XML message. The actual trigger is either the document type declaration (DOCTYPE) specified in the prolog of the XML message, or it is the root element of the XML message.

You specify the DOCTYPE or root element with which you want to start the workflow using the Start Properties dialog box. Similarly, you use the Event Properties dialog box to specify the DOCTYPE or root element that triggers an Event node.

BEA WebLogic Process Integrator does not start the workflow or trigger an Event node unless the DOCTYPE or root element in the Start or Event Properties dialog box matches that in the incoming XML message.

In addition to using the DOCTYPE or root element, you can further qualify the start of a workflow or the triggering of an Event node by using an event key and an event condition.

What Is an Event Key?

An event key is a value that must match an element in the incoming XML document before BEA WebLogic Process Integrator starts the workflow or triggers an Event node. The key value can be any element immediately under the root element of the incoming XML message. You specify the key value in the Start or Event properties dialog box.

If you specify an event key, BEA WebLogic Process Integrator checks that the key value you specified in the properties dialog box matches an element in the incoming XML message before it starts a workflow or triggers an Event node.

Using an event key can improve the performance of your workflows. By specifying a key value using the properties dialog box for a Start or Event node, you are, in effect, uniquely identifying that node. When BEA WebLogic Process Integrator receives an XML message, it can quickly locate the appropriate Start or Event node by matching the key value in the incoming XML message to the key value you specified for the Start or Event node.

What Is an Event Condition?

To even further qualify the start of a workflow or the triggering of an Event node, you can specify a condition that must be evaluated. A condition may be an expression, such as Amount > 1000. You specify an event condition in the Start or Event Properties dialog box.

If you specify a condition, BEA WebLogic Process Integrator evaluates the condition when it receives an XML message. If the condition is true, BEA WebLogic Process Integrator starts the workflow or triggers an Event node.

Defining an Event Key

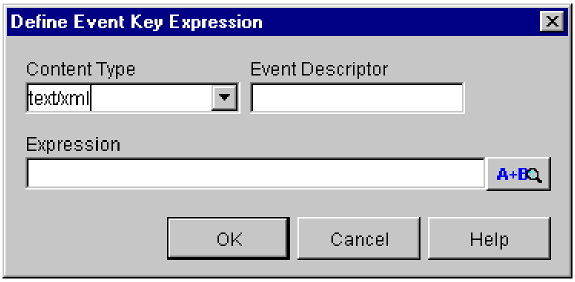
To use an event key, you need to do the following:

- Specify the key value in the Start or Event Properties dialog boxes. For details, see “Defining a Start Node Using an Event Trigger” on page 6-14 and “Defining Events” on page 6-24.
- Define the expression that locates the key value in the incoming XML message, so BEA WebLogic Process Integrator can compare it against the key value you specify in the Start or Event Properties dialog boxes.

To specify an event key:

1. From the BEA WebLogic Process Integrator Configuration menu, select Events to display the Event Key Expressions dialog box.
2. Click Add to display the Define Event Key Expression dialog box.

Figure A-1 Define Event Key Expression Dialog Box



Define Event Key Expression

Content Type: text/xml

3. In the Content Type field, select text/xml for an XML message.
4. In the Event Descriptor field, enter the DOCTYPE or root element you specified in the Start or Event Properties dialog box.
5. In the Expression field, enter the expression that locates the key value in the incoming XML message.

   Note: Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click the A+B button to display the Expression Builder dialog box. The Expression Builder helps you construct an expression, which is made up of functions, operators, literals, and variables. For more information on constructing expressions, see Chapter 8, “Using Expressions and Conditions.”

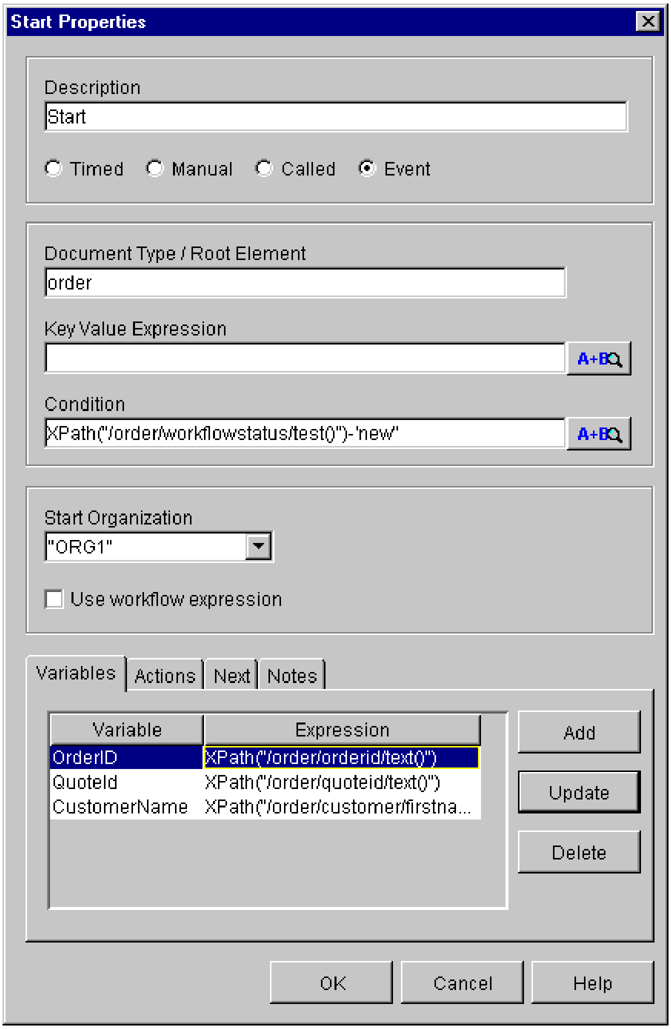
6. Click OK. The event key is stored in an event key table maintained by BEA WebLogic Process Integrator, and is displayed in the Event Key Expressions dialog box.
7. Click Close.

B Example of Starting a Workflow Using an Event

This section presents an example that shows you how to define a Start node that starts a workflow when an event is triggered. The event that starts the workflow is the receipt of an XML document that contains the root element `order`, and a `workflowstatus` element with a value of `new`.

To define the Start node, the entries in the Start Properties dialog box should look like those shown in the following figure.

Figure B-1 Start Properties Dialog Box



Start Properties

Description: Start

○ Timed ○ Manual ○ Called ● Event

Document Type / Root Element: order

Key Value Expression:

Condition: XPath("/order/workflowstatus/test()")-'new'

Start Organization: "ORG1"

□ Use workflow expression

Variable	Expression
OrderID	XPath("/order/orderid/text()")
QuoteId	XPath("/order/quoteid/text()")
CustomerName	XPath("/order/customer/firstna...

In the Start Properties dialog box, specify the following:

- Document Type/Root Element — specify the root element `order`
- Condition — specify the following XPath statement that determines if the `workflowstatus` element of the XML document contains the value new:

  ```
  XPath(“/order/workflowstatus/test()-’new”)
  ```

  If the condition evaluates to true, the workflow starts.
- Variables — specify the XPath statements shown in the following table that are used to populate the workflow variables when the workflow is started with values contained in the XML document:

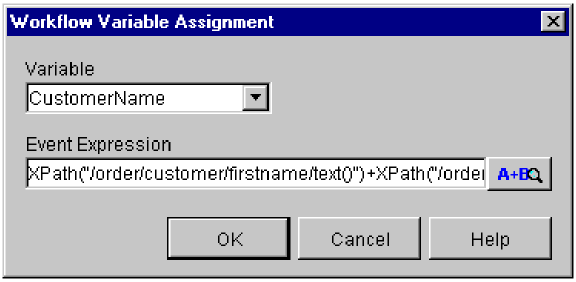
Table B-1 Callback Variables and XPath Statements

Workflow Variable	XPath Statement
`OrderID`	`XPath(“/order/orderid/text()”)`
`QuoteID`	`XPath(“/order/quoteid/text()”)`

`CustomerName`	`XPath("/Order/Customer/FirstName/text()") +`
	`XPath("/Order/Customer/LastName/text()")`

To specify the variables, click Add to open the Workflow Variable Assignment dialog box.

Figure B-2 Workflow Variable Assignment Dialog Box



Workflow Variable Assignment

Variable: CustomerName

Event Expression: XPath("/order/customer/firstname/text()")+XPath("/orde

The XML document that triggers the event-type start is shown in the following listing.

Listing B-1 XML Document That Triggers the Event-Type Start

```
<Order>
  <OrderId>1</OrderId>
  <OrderType>New</OrderType>
  <WorkflowStatus>New</WorkflowStatus>
  <Status />
 <LEC>
  <VendorId>ABC</VendorId>
  </LEC>
<SalesPerson>
  <UserId>Mark</UserId>
  <Email>Mark@ABC</Email>
</SalesPerson>
<Quote>
  <Id>1</Id>
</Quote>
<Customer>
  <FirstName>Joe</FirstName>
  <LastName>Customer</LastName>
 <Address>
  <Street>123 Main Street</Street>
  <City>Boston</City>
  <State>MA</State>
  </Address>
  </Customer>
  </Order>
```

CHAPTER

C Working with XML Entities

The following sections explain how to use the XML Finder to find XML entities, and how to manage XML entities in the XML repository:

- Locating XML Entities Using the XML Finder
- Managing the XML Repository

Locating XML Entities Using the XML Finder

The XML Finder dialog box helps you find XML entities stored in the XML repository, or the local file system, or at a URL. The dialog box also maintains a list of the XML entities you used most recently, so you do not have to search for them if you want to use them again. XML entities can be XML documents, DTD files, schema files, and XSLT template documents. Once you select an XML entity, the XML Finder returns the entity to the Studio.

You can access the XML Finder in a couple of ways:

- From Studio dialog boxes, such as the following: XSL Transform, Send XML to Client, Post XML Event, and the XPath wizard.

  Using this method, you can access XML entities stored in the XML repository, the local file system, or at a URL. The following sections provide details.

- By choosing Tools→Show XML Finder.

  Using this method, you can access XML entities stored in the XML respository only. For details, see “Managing the XML Repository” on page C-7.

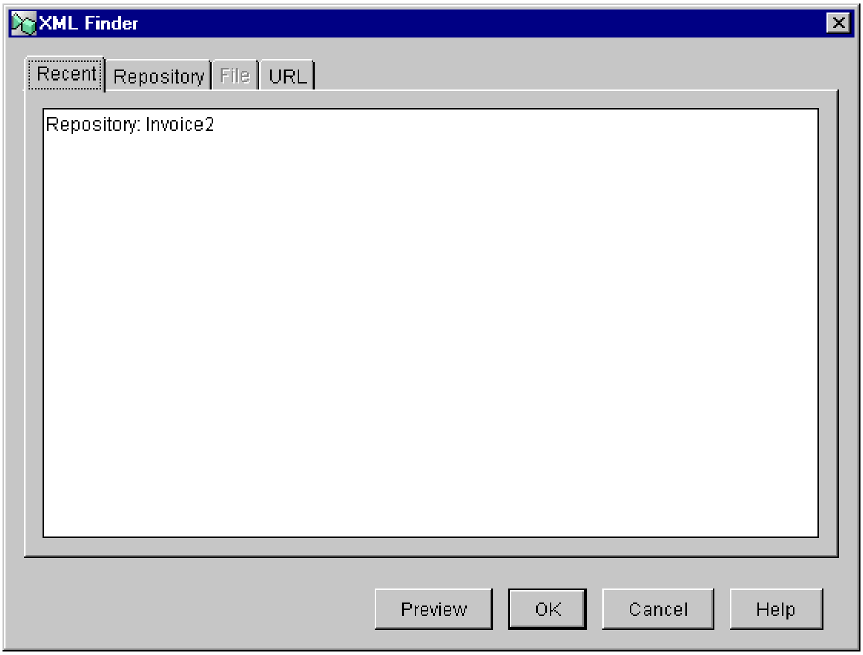
Retrieving the Most Recently Used XML Entities

The XML Finder maintains a list of the XML entities you used most recently on your workstation. You can re-use the entities in the list without having to search for them again.

To retrieve an entity from the most recently used list:

1. In the XML Finder dialog box, select the Recent tab.

Figure C-1 Recently Used XML Objects



Repository: Invoice2

2. Select the entity you want to use.

3. To see the contents of the entity, click Preview.

4. To return the entity to the Studio, click OK.

Retrieving XML Entities from the XML Repository

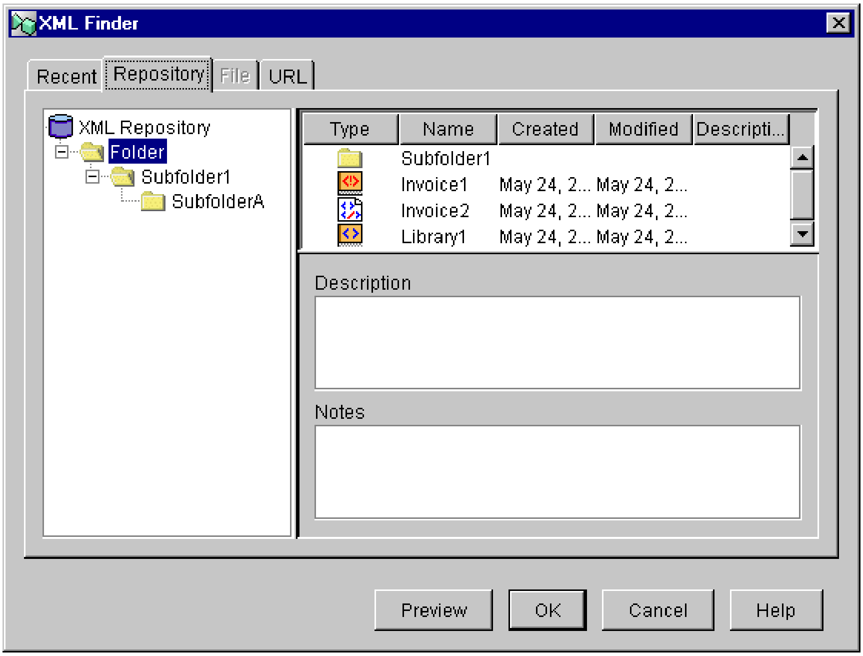
The XML Finder provides a way to obtain XML entities stored in the XML repository maintained by BEA WebLogic Process Integrator. The XML Finder also provides a way to manage the entities in the repository. For more information about the XML repository, see “Managing the XML Repository” on page C-7.

To retrieve an entity from the repository:

1. In the XML Finder dialog box, select the Repository tab.

2. Select the folder from which you want to retrieve an XML entity.

Figure C-2 XML Objects in the Selected Folder



Type	Name	Created	Modified	Descripti...
	Subfolder1			
	Invoice1	May 24, 2...	May 24, 2...	
	Invoice2	May 24, 2...	May 24, 2...	
	Library1	May 24, 2...	May 24, 2...	

The left panel shows you a tree view of the repository with folders and subfolders arranged hierarchically. The top-most panel on the right shows you the entities in the selected folder. The Description field contains a description of the selected folder, and the Notes field contains any notes about the selected folder.

3. In the top-most panel on the right, select the entity you want to retrieve.

4. To see the contents of the entity, click Preview.

5. To return the entity to the Studio, click OK.

Retrieving XML Entities from the Local File System

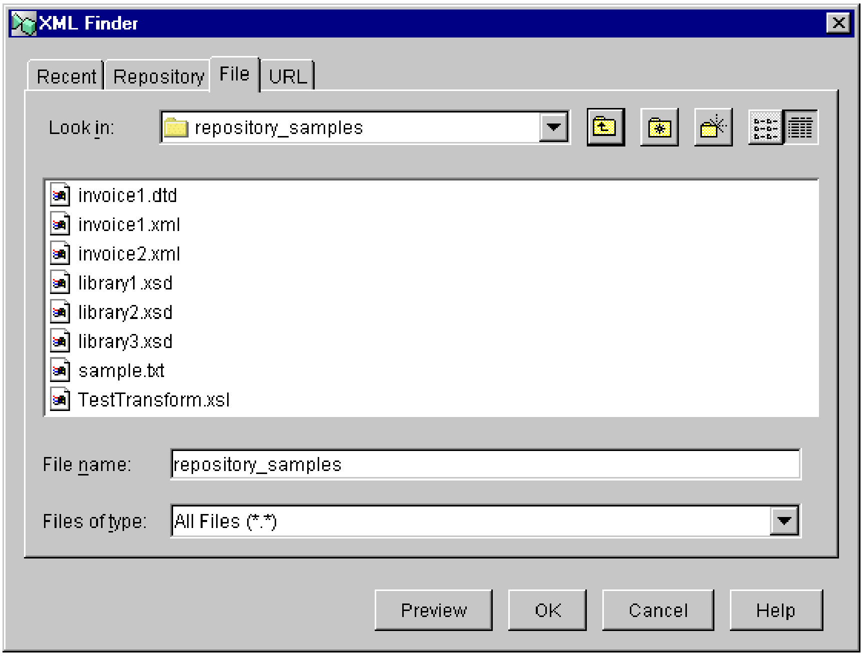
The XML Finder provides a way to retrieve XML entities from the file system on your workstation.

To retrieve an entity from the local file system:

1. In the XML Finder dialog box, select the File tab.

   Note: The File tab is unavailable if you access the XML Finder from a dialog box that you are using to specify an action that takes place at run time. For example, if you access the XML Finder from the XSL Transform dialog box, the File tab is unavailable because the XSL transformation occurs at run time. Specifying an XSL template document located on the local file system is not appropriate in this case because BEA WebLogic Process Integrator can not access the local file system at run time.

Figure C-3 XML Objects in the File Sytem



Look in: repository_samples

invoice1.dtd
invoice1.xml
invoice2.xml
library1.xsd
library2.xsd
library3.xsd
sample.txt
TestTransform.xsl

File name: repository_samples

Files of type: All Files (*.*)

2. In the Look in field, specify the drive and folder from which you want to obtain the XML entity. If necessary, use the buttons to the right of the Look in field.
3. From the file list, select the entity you want to retrieve.
4. To see the contents of the entity, click Preview.
5. To return the entity to the Studio, click OK.

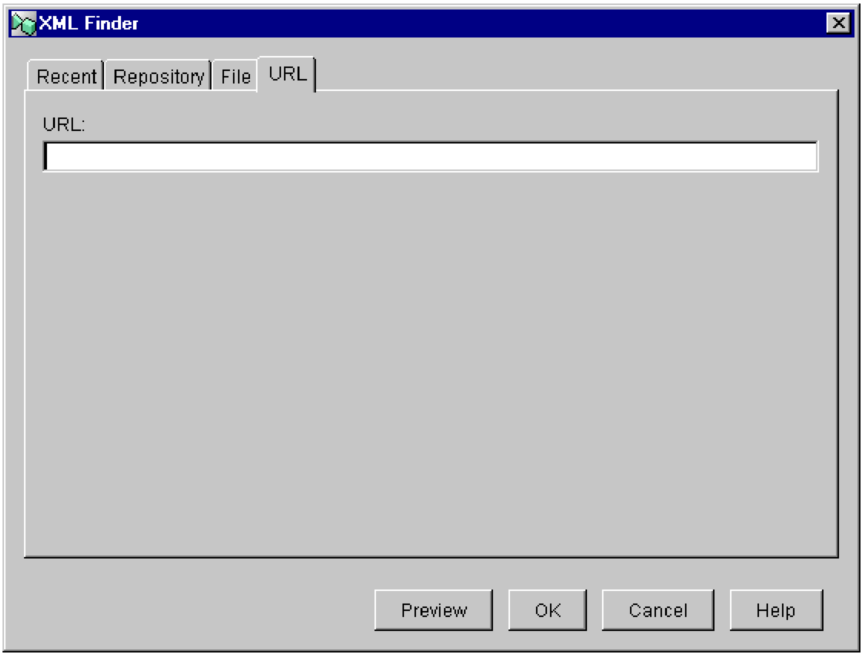
Retrieving XML Entities from a URL

The XML Finder provides a way to retrieve XML entities from a URL.

To retrieve an entity from a URL:

1. In the XML Finder dialog box, select the URL tab.

Figure C-4 URL Tab in the XML Finder Dialog Box



2. In the URL field, enter the complete URL, including the protocol, server, and filename, of the XML entity you want to retrieve.
3. To see the contents of the entity, click Preview.
4. To return the entity to the Studio, click OK.

Managing the XML Repository

The XML repository is maintained by BEA WebLogic Process Integrator, and is used to store XML entities, such as XML documents, DTD files, and XSL template documents.

You can view the contents of the repository, and perform actions on the repository, on folders in the repository, and on XML entities in folders.

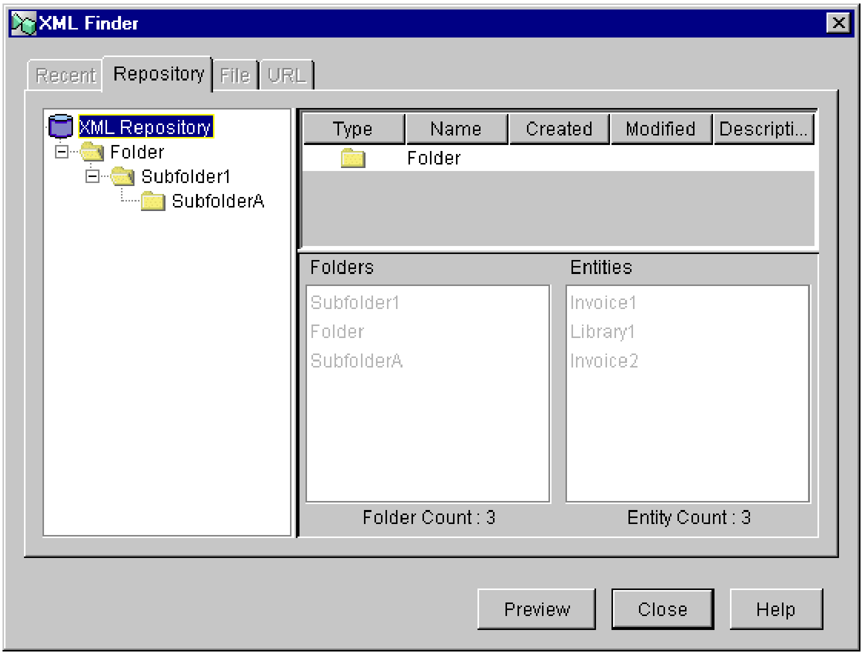
Viewing the Contents of the Repository

You can view the contents of the entire repository, or the contents of individual folders in the repository.

To view the contents of the entire repository,

1. Choose Tools→Show XML Finder. The XML Finder dialog box opens with the Repository tab selected.
2. In the left panel, select XML Repository.

Figure C-5 Contents of the Entire Repository



Type	Name	Created	Modified	Descripti...
🗀	Folder			

Folders

Subfolder1
Folder
SubfolderA

Folder Count : 3

Entities

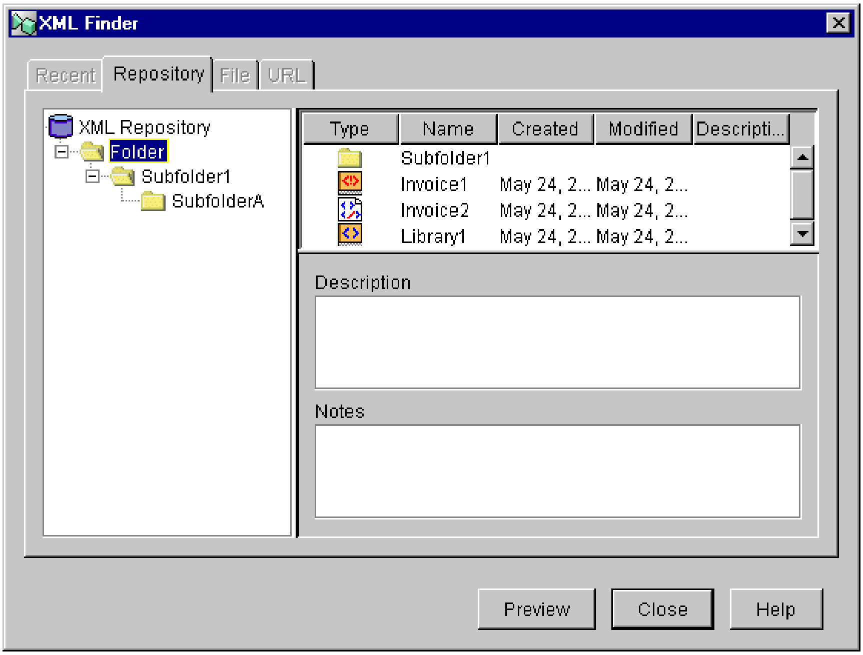
Invoice1
Library1
Invoice2

Entity Count : 3

The left panel shows you a tree view of the repository with folders and subfolders arranged hierarchically. The top-most panel on the right shows you a detailed view of the repository. The Folders field lists all the folders in the repository, including their total number. The Entities field lists all the XML entities in the repository, including their total number.

To view the contents of a folder, select the folder you want to view.

Figure C-6 XML Entities in the Selected Folder



Type	Name	Created	Modified	Descripti...
	Subfolder1			
<!>	Invoice1	May 24, 2...	May 24, 2...	
	Invoice2	May 24, 2...	May 24, 2...	
<>	Library1	May 24, 2...	May 24, 2...	

The left panel shows you a tree view of the repository with folders and subfolders arranged hierarchically. The top-most panel on the right shows you the contents of the selected folder. The Description field contains a description of the selected folder, and the Notes field contains any notes about the selected folder.

The XML repository stores different types of XML entities, each one represented by a symbol, as shown in the following table.

Symbol	XML Entity Type
<!>	DTD file
	MFL file
< >	Schema file

Symbol	XML Entity Type
T	Text file
</>	XML document
a→X	XSLT template document

Working with the Entire Repository

You can perform several different actions at the root level of the repository. To perform actions at the root level of the repository:

1. Choose Tools→Show XML Finder. The XML Finder dialog box opens.
2. Right-click XML Repository in the tree view of the repository. A pop-up menu appears where you can select the following actions.

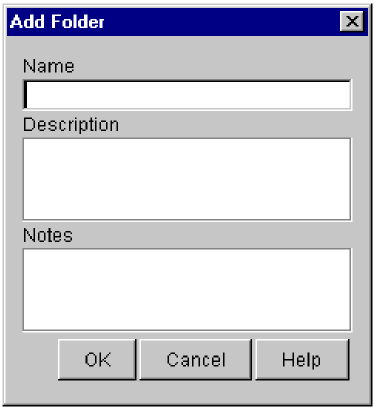
Menu Option	Action
Add Folder	Add a folder to the root of the repository using the Add Folder dialog box. For details, see "Adding a Folder" on page C-11.
Add Entity	Add an XML entity to the repository using the the Add Entity dialog box. For details, see "Adding an XML Entity" on page C-12.
Paste	Paste an XML entity that was cut from another location in the repository into the current selection. This menu option is available only if an XML entity is cut from another location, and is available to be pasted.

You can also export the contents of the repository to a Java Archive (JAR) file, and import the contents of a JAR file into the repository. For details, see "Exporting and Importing Workflow Packages" on page 5-24.

Adding a Folder

Clicking Add Folder in the pop-up menu opens the Add Folder dialog box.

Figure C-7 Add Folder Dialog Box



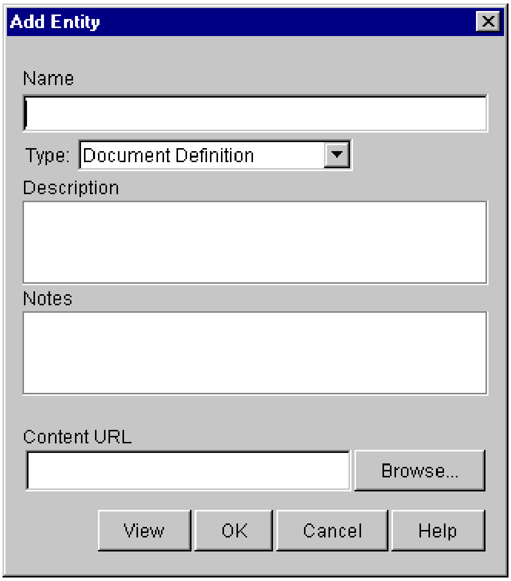
To add a folder:

1. In the Name field, enter the name of the folder.
2. Optionally, enter a description and notes about the folder in the Description and Notes fields, respectively.
3. Click OK.

Adding an XML Entity

Clicking Add Entity in the pop-up menu opens the Add Entity dialog box.

Figure C-8 Add Entity Dialog Box



Type: Document Definition

To add an XML entity:

1. In the Name field, enter a unique name for the entity you are adding.
2. Using the Type drop-down list, select the type of entity you are adding.
3. Optionally, enter a description and notes about the entity in the Description and Notes fields, respectively.
4. In the Content URL field, enter the URL of the entity you are adding, or use Browse to locate the entity on the local file system.
5. Optionally, click View to display the contents of the entity you are adding.
6. Click OK.

Working with Folders

You can perform several different actions on a folder. To perform actions on a folder:

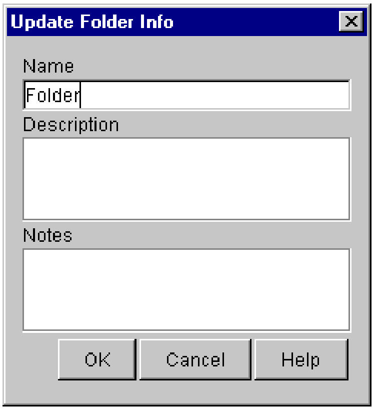
1. Choose Tools→Show XML Finder. The XML Finder dialog box opens.
2. Right-click a folder in the tree view of the repository. A pop-up menu appears where you can select the following actions.

Menu Option	**Action**
Add Folder	Add a subfolder to the selected folder using the Add Folder dialog box. For details, see “Adding a Folder” on page C-11.
Delete Folder	Delete the folder from the repository. A folder may only be deleted when it has no subfolders. A dialog box asks you to confirm the delete action.
Update Folder Info	Change the name, description, and notes about the folder using the Update Folder Info dialog box. For details, see “Updating Folder Information” on page C-13.
Add Entity	Add an XML entity to the folder using the Add Entity dialog box. For details, see “Adding an XML Entity” on page C-12.
Paste	Paste an XML entity that was cut from another location in the repository into the current selection. This menu option is available only if an XML entity is cut from another location, and is available to be pasted.

Updating Folder Information

Clicking Update Folder Info in the pop-up menu opens the Update Folder Info dialog box which displays current information about the selected folder.

Figure C-9 Update Folder Info Dialog Box



Name: Folder

To update folder information:

1. Change the contents of the Name, Description, and Notes fields as necessary.
2. Click OK.

Working with XML Entities

You can perform several different actions on XML entities in folders in the repository. To perform actions on XML entities:

1. Choose Tools→Show XML Finder. The XML Finder dialog box opens, displaying the repository.
2. In the tree view of the repository, select the folder that contains the entities with which you want to work.
3. In the top-most panel on the right, right-click the entity with which you want to work. A pop-up menu appears where you can select the following actions.

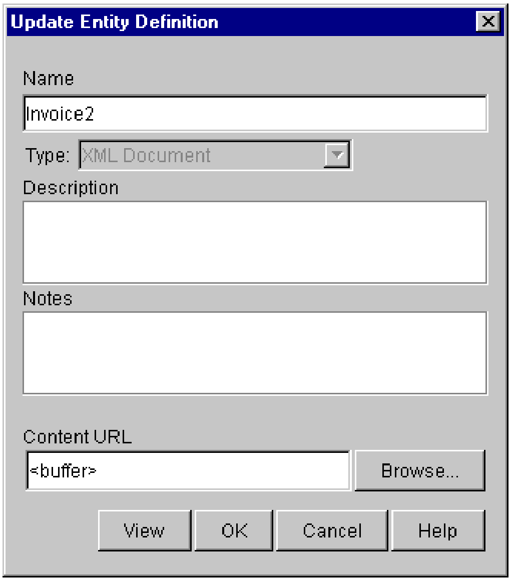
Menu Option	Action
Add Entity	Add an XML entity to the folder using the the Add Entity dialog box. For details, see “Adding an XML Entity” on page C-12.

Menu Option	Action
Export Entity	Export the entity to the local file system using the Save dialog box. For details, see “Exporting an Entity” on page C-16.
Delete Entity	Delete the entity from the folder. A dialog box asks you to confirm the delete action.
Update Entity	Change the name, description, and notes about the entity using the Update Folder Info dialog box. For details, see “Updating Entity Information” on page C-15.
Cut	Cut the entity from the folder. Once the entity is cut, you can paste it into another location in the repository.

Updating Entity Information

Clicking Update Entity in the pop-up menu opens the Update Entity Definition dialog box which displays current information about the selected entity.

Figure C-10 Update Entity Definition Dialog Box



Update Entity Definition

Name: Invoice2

Type: XML Document

Description:

Notes:

Content URL: <buffer>

To update entity information:

1. Change the contents of the Name, Description, and Notes fields as necessary.

   Note: You cannot change the entity content type using the Update Entity Definition dialog box. To change the type of an entity, you must create a new entity with the desired content type. For details, see “Adding an XML Entity” on page C-12.

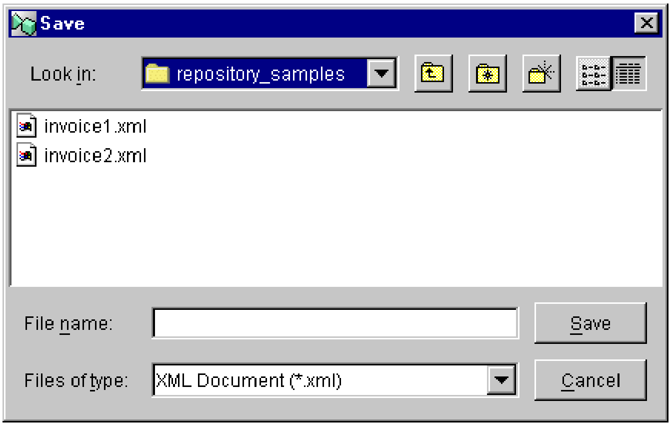
2. In the Content URL field, enter the new URL of the entity, or use Browse to specify the new location of the entity in the local file system.

3. Click OK.

Exporting an Entity

Clicking Export Entity in the pop-up menu opens the Save dialog box.

Figure C-11 Save Dialog Box



To export an entity to the local file system:

1. From the Look in drop-down list, select the folder to which you want to export the entity.

2. In the File name field, specify the name of the file to which you want to export the entity, or select an existing file.

3. Click Save.

Index