



# BEA WebLogic Process Integrator

## Studio User Guide

**BEA WebLogic Process Integrator Release 1.2**  
**Document Edition 1.2**  
**December 2000**

## Copyright

Copyright © 2000 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

All other product names may be trademarks of the respective companies with which they are associated.

### BEA WebLogic Process Integrator Studio User Guide

| Document Edition | Part Number | Date          | Software Version |
|------------------|-------------|---------------|------------------|
| 1.2              | N/A         | December 2000 | 1.2              |

---

# Contents

## About This Document

|                                 |      |
|---------------------------------|------|
| What You Need to Know .....     | xii  |
| e-docs Web Site .....           | xii  |
| How to Print the Document.....  | xiii |
| Related Information.....        | xiii |
| Contact Us! .....               | xiv  |
| Documentation Conventions ..... | xv   |

## 1. WebLogic Process Integrator Overview

|                                                   |      |
|---------------------------------------------------|------|
| Architecture Overview .....                       | 1-2  |
| Understanding the Workflow Model.....             | 1-4  |
| Overview of the Workflow Model .....              | 1-4  |
| Workflow Definitions .....                        | 1-5  |
| Workflow Components .....                         | 1-6  |
| Nodes .....                                       | 1-6  |
| Variables .....                                   | 1-7  |
| Exception Handlers .....                          | 1-7  |
| Actions .....                                     | 1-7  |
| Workflow Model Diagram .....                      | 1-8  |
| Administering Data .....                          | 1-9  |
| Monitoring Workflows.....                         | 1-10 |
| Starting WebLogic Process Integrator Studio ..... | 1-11 |
| Navigating WebLogic Process Integrator .....      | 1-13 |
| Menu Options.....                                 | 1-13 |
| Folder Tree Display.....                          | 1-15 |
| Workflow Drawing Area.....                        | 1-19 |
| Exiting WebLogic Process Integrator Studio .....  | 1-22 |

---

## 2. Administering Data within WebLogic Process Integrator

|                                                     |      |
|-----------------------------------------------------|------|
| Security .....                                      | 2-2  |
| Maintaining Organizations .....                     | 2-3  |
| Adding an Organization .....                        | 2-3  |
| Updating an Organization.....                       | 2-6  |
| Deleting an Organization.....                       | 2-7  |
| Maintaining Users.....                              | 2-8  |
| Adding a User to the Current Organization.....      | 2-10 |
| Creating a User .....                               | 2-12 |
| Deleting a User .....                               | 2-13 |
| Updating User Properties .....                      | 2-14 |
| Removing a User from the Current Organization ..... | 2-14 |
| Maintaining Roles.....                              | 2-15 |
| Creating a Role .....                               | 2-16 |
| Updating a Role.....                                | 2-17 |
| Deleting a Role .....                               | 2-18 |
| Administering Task Routing .....                    | 2-19 |
| Adding a Task Routing Specification.....            | 2-20 |
| Updating a Task Routing Specification.....          | 2-21 |
| Deleting a Task Routing Specification.....          | 2-22 |
| Refreshing the Task List.....                       | 2-22 |
| Administering Business Calendars.....               | 2-23 |
| Creating a Calendar .....                           | 2-24 |
| Updating a Calendar .....                           | 2-30 |
| Deleting a Calendar .....                           | 2-30 |
| Understanding Business Operations .....             | 2-31 |
| Using the Business Operations Dialog Box .....      | 2-32 |
| Defining a Business Operation.....                  | 2-33 |
| Defining Business Operations for Java Classes ..... | 2-35 |
| Defining Business Operations for Session EJBs.....  | 2-38 |
| Defining Business Operations for Entity EJBs .....  | 2-40 |

## 3. Defining and Maintaining Workflows

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| Understanding Workflow Templates and Workflow Template Definitions ..... | 3-2 |
| Creating a New Workflow Template .....                                   | 3-3 |

---

|                                                             |      |
|-------------------------------------------------------------|------|
| Creating a New Workflow Template Definition.....            | 3-4  |
| Specifying Workflow Template Definition Properties .....    | 3-6  |
| Opening Workflow Template Definitions.....                  | 3-11 |
| Saving and Closing Workflow Template Definitions .....      | 3-13 |
| Printing Workflow Template Definitions.....                 | 3-13 |
| Copying Workflow Template Definitions.....                  | 3-17 |
| Copying Entire Workflow Template Definitions.....           | 3-17 |
| Copying Nodes Within Workflow Template Definitions .....    | 3-18 |
| Copying Actions Within or Between Workflow Node Shapes..... | 3-19 |
| Copying Actions Within Workflow Nodes.....                  | 3-19 |
| Copying Actions Between Workflow Nodes.....                 | 3-21 |
| Deleting Workflow Template Definitions.....                 | 3-23 |
| Deleting Workflow Templates .....                           | 3-24 |
| Understanding Effective and Expiry Dates .....              | 3-25 |
| Exporting Workflow Template Definitions.....                | 3-26 |
| Importing Workflow Template Definitions.....                | 3-27 |
| Displaying Workflow Instances .....                         | 3-29 |

## 4. Working with Workflow Components

|                                                        |      |
|--------------------------------------------------------|------|
| Adding Workflow Components to a Workflow Diagram ..... | 4-2  |
| Deleting Components from a Workflow Diagram .....      | 4-2  |
| Defining Variables.....                                | 4-3  |
| Defining Variables .....                               | 4-3  |
| Updating Variables.....                                | 4-5  |
| Locating Variables in the Workflow .....               | 4-5  |
| Variable Conversion Rules.....                         | 4-6  |
| Defining Start .....                                   | 4-8  |
| Specifying Multiple Starts.....                        | 4-8  |
| Defining a Start Node.....                             | 4-9  |
| Defining Dones.....                                    | 4-15 |
| Defining Decisions .....                               | 4-16 |
| Defining Tasks.....                                    | 4-19 |
| Defining Events .....                                  | 4-22 |
| Using Joins (And/Or) .....                             | 4-23 |
| How Joins Work.....                                    | 4-23 |

---

|                       |      |
|-----------------------|------|
| Defining a Join ..... | 4-24 |
|-----------------------|------|

## 5. Using Actions

|                                        |      |
|----------------------------------------|------|
| Working with Actions .....             | 5-2  |
| Action Categories .....                | 5-2  |
| Action Placement and Subactions .....  | 5-2  |
| Adding or Updating an Action .....     | 5-3  |
| Deleting an Action .....               | 5-5  |
| Changing the Sequence of Actions.....  | 5-5  |
| Using Task Actions .....               | 5-6  |
| Mark Task as Done.....                 | 5-7  |
| Unmark Task Done.....                  | 5-8  |
| Set Task Due Date .....                | 5-9  |
| Assign Task to User.....               | 5-11 |
| Assign Task to Role.....               | 5-13 |
| Assign Task Using Routing Table.....   | 5-14 |
| Unassign Task .....                    | 5-17 |
| Set Task Comment .....                 | 5-18 |
| Set Task Priority .....                | 5-20 |
| Execute Task .....                     | 5-21 |
| Using Workflow Actions.....            | 5-22 |
| Mark Workflow as Done .....            | 5-23 |
| Abort Workflow .....                   | 5-24 |
| Start Workflow .....                   | 5-25 |
| Set Workflow Variable.....             | 5-27 |
| Set Workflow Comment.....              | 5-30 |
| Using Integration Actions.....         | 5-31 |
| Post XML Event .....                   | 5-31 |
| Send XML to Client .....               | 5-31 |
| Call Program.....                      | 5-32 |
| Perform Business Operation.....        | 5-33 |
| Using Exception Handling Actions ..... | 5-36 |
| Using Miscellaneous Actions .....      | 5-36 |
| No Operation .....                     | 5-36 |
| Send E-Mail Message.....               | 5-37 |

|                             |      |
|-----------------------------|------|
| Evaluate Condition .....    | 5-40 |
| Timed Event .....           | 5-42 |
| Cancel Workflow Event ..... | 5-46 |
| Make Audit Entry .....      | 5-46 |

## 6. Using Expressions and Conditions

|                                                                      |      |
|----------------------------------------------------------------------|------|
| Workflow Expression Functionality .....                              | 6-2  |
| Generate a String That Uniquely Identifies a Workflow Instance ..... | 6-2  |
| Determine Whether a Particular Event Should Trigger a Workflow ..... | 6-3  |
| Using the Expression Builder .....                                   | 6-3  |
| Using Expression Components .....                                    | 6-6  |
| Functions .....                                                      | 6-6  |
| Abs .....                                                            | 6-6  |
| CurrentUser .....                                                    | 6-6  |
| Date .....                                                           | 6-6  |
| DateAdd .....                                                        | 6-7  |
| DateToString .....                                                   | 6-8  |
| StringLen .....                                                      | 6-10 |
| StringToDate .....                                                   | 6-10 |
| SubString .....                                                      | 6-10 |
| TaskAttribute .....                                                  | 6-11 |
| ToInteger .....                                                      | 6-11 |
| ToString .....                                                       | 6-12 |
| WorkflowAttribute .....                                              | 6-12 |
| WorkflowVariable .....                                               | 6-13 |
| XPath .....                                                          | 6-14 |
| Date Function Format .....                                           | 6-15 |
| Pattern Letter Definitions .....                                     | 6-16 |
| Format Guidelines .....                                              | 6-18 |
| Examples of Time Patterns .....                                      | 6-18 |
| Operators .....                                                      | 6-19 |
| Literals .....                                                       | 6-21 |
| Variables .....                                                      | 6-22 |

---

## 7. Handling Workflow Exceptions

|                                                                      |      |
|----------------------------------------------------------------------|------|
| Workflow Exception Handling Overview .....                           | 7-2  |
| Understanding the WebLogic Process Integrator Transaction Model..... | 7-4  |
| Defining Exception Handlers .....                                    | 7-7  |
| Using Exception Handling Actions .....                               | 7-10 |
| Invoke Exception Handler .....                                       | 7-10 |
| Set Workflow Exception Handler .....                                 | 7-10 |
| Exit Exception Handler .....                                         | 7-12 |
| WebLogic Process Integrator Error Messages .....                     | 7-13 |

## 8. Monitoring Workflows

|                                              |      |
|----------------------------------------------|------|
| Using the Workflow Monitoring Features ..... | 8-2  |
| User Interface .....                         | 8-2  |
| Workflow Instance Status.....                | 8-3  |
| Viewing Workflow Status .....                | 8-5  |
| Viewing Workflow Variables.....              | 8-11 |
| Deleting Workflows .....                     | 8-13 |
| Using Workload Reports .....                 | 8-14 |
| Compiling Workload Report Information .....  | 8-15 |
| Viewing Workload Reports .....               | 8-16 |
| Using Statistics Reports.....                | 8-17 |
| Compiling Statistics Report Information..... | 8-17 |
| Viewing Statistics Reports .....             | 8-19 |

## 9. Understanding WebLogic Process Integrator Events and XML Documents

|                                                                         |      |
|-------------------------------------------------------------------------|------|
| XML Overview .....                                                      | 9-2  |
| XML Elements and Attributes.....                                        | 9-2  |
| Document Type Definition (DTD).....                                     | 9-3  |
| Processing Events and Handling XML Documents .....                      | 9-4  |
| Defining a Workflow Event Trigger.....                                  | 9-5  |
| Workflow Examples.....                                                  | 9-10 |
| Using the XML Notification: The WebLogic Process Integrator Event ..... | 9-14 |
| Setting Up a Key Value for Each XML Root.....                           | 9-19 |
| Using XML Actions .....                                                 | 9-20 |

---

|                                                      |      |
|------------------------------------------------------|------|
| Post XML Event .....                                 | 9-20 |
| Send XML to Client .....                             | 9-27 |
| XML Root Element Names .....                         | 9-31 |
| Default Settings for Send XML to Client Action ..... | 9-31 |
| Dialog Box Prompt .....                              | 9-32 |
| Display Message-Box .....                            | 9-34 |
| Invoke Exception Handler .....                       | 9-38 |
| Understanding Workflow Interaction .....             | 9-40 |

## Index



---

# About This Document

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

- Chapter 1, “WebLogic Process Integrator Overview,” describes the WebLogic Process Integrator architecture and workflow model. It also introduces you to the WebLogic Process Integrator user interface and provides instructions for starting and exiting WebLogic Process Integrator.
- Chapter 2, “Administering Data within WebLogic Process Integrator,” explains how to add, update, and delete organizations, users, roles, and tasks. It also describes task routing administration, calendar administration, and business operations.
- Chapter 3, “Defining and Maintaining Workflows,” explains how to create, define, open, save, close, copy, and delete WebLogic Process Integrator workflows.
- Chapter 4, “Working with Workflow Components,” describes workflow components, including variables, starts, done, decisions, tasks, events, and joins.
- Chapter 5, “Using Actions,” describes actions, including task actions, workflow actions, integration actions, and miscellaneous actions.
- Chapter 6, “Using Expressions and Conditions,” describes the workflow Expression Builder and its components.
- Chapter 7, “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 8, “Monitoring Workflows,” describes the workflow monitoring facility, including its user interface, workflow statuses, workflow variables, workflow attachments, workflow reports, and more.

- 
- Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents,” provides an overview of XML (extensible markup language), event processing, XML document handling within WebLogic Process Integrator, and more.

## 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 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 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 WebLogic Process Integrator documents contain information that is relevant to using this product.

- *BEA WebLogic Process Integrator Tutorial*
- *BEA WebLogic Process Integrator Worklist Guide*
- *BEA WebLogic Process Integrator Installation and Configuration Guide*
- *BEA WebLogic Process Integrator Javadoc*
- *BEA WebLogic Process Integrator Release Notes*

---

# Contact Us!

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

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

If you have any questions about this version of BEA WebLogic Process Integrator, or if you have problems installing and running BEA WebLogic Process Integrator, contact BEA Customer Support through BEA WebSupport at **[www.beasys.com](http://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.                                                                                                                                                                                                                                                                                                                                         |
| <i>italics</i>               | 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.<br><i>Examples:</i><br><code>#include &lt;iostream.h&gt; void main ( ) the pointer psz<br/>chmod u+w *<br/>\tux\data\ap<br/>.doc<br/>tux.doc<br/>BITMAP<br/>float</code> |
| <i>monospace italic text</i> | Identifies variables in code.<br><i>Example:</i><br><code>String <i>expr</i></code>                                                                                                                                                                                                                                                                                                                    |
| UPPERCASE TEXT               | Indicates device names, environment variables, and logical operators.<br><i>Examples:</i><br><code>LPT1<br/>SIGNON<br/>OR</code>                                                                                                                                                                                                                                                                       |
| { }                          | 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.<br><i>Example:</i><br><code>buildobjclient [-v] [-o name ] [-f file-list]...<br/>[-l file-list]...</code>                                                                                                                                                                                                                                                                             |
|            | Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.                                                                                                                                                                                                                                                                                                                                                                                 |
| ...        | Indicates one of the following in a command line: <ul style="list-style-type: none"><li>■ That an argument can be repeated several times in a command line..</li><li>■ That the statement omits additional optional arguments.</li><li>■ That you can enter additional parameters, values, or other information.</li></ul> The ellipsis itself should never be typed.<br><i>Example:</i><br><code>buildobjclient [-v] [-o name ] [-f file-list]...<br/>[-l file-list]...</code> |
| .          | Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.                                                                                                                                                                                                                                                                                                                                                  |

# 1 WebLogic Process Integrator Overview

The WebLogic Process Integrator product is a user modeling and workflow environment that combines the tasks of administering data and defining, executing, and monitoring workflows.

The following sections provide an overview of WebLogic Process Integrator:

- Architecture Overview
- Understanding the Workflow Model
- Administering Data
- Monitoring Workflows
- Starting WebLogic Process Integrator Studio
- Navigating WebLogic Process Integrator
- Exiting WebLogic Process Integrator Studio

# Architecture Overview

The Workflow Management Coalition defines a workflow management system as one that “...defines, manages, and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic.”

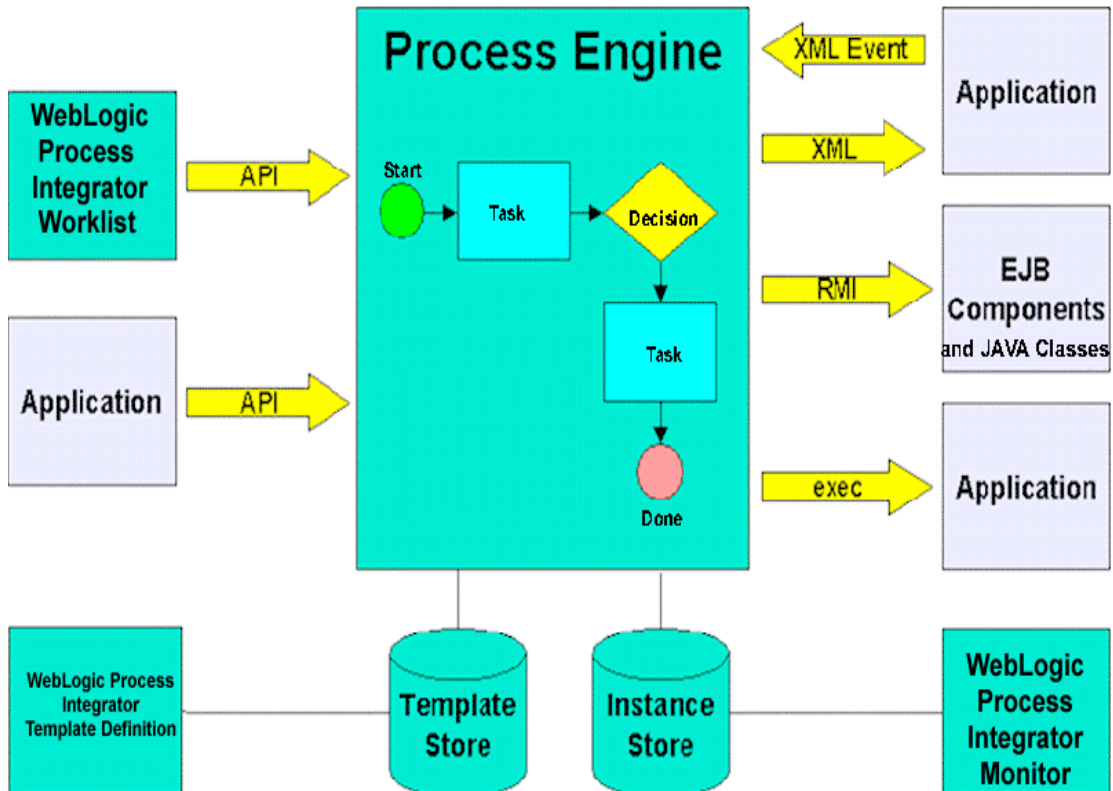
Such a system automates a business process by merging the sequence of activities and invoking the appropriate resources required by the various activities or steps. Individual business processes may have life cycles ranging from minutes to days (or longer), depending on the complexity and duration of the various constituent activities. To achieve this, a workflow management system must provide support in three broad areas:

- Workflow definition and execution — capturing the definition of the business process (workflow), managing the execution of workflow processes in an operational environment, and sequencing the various activities to be performed.
- Workflow monitoring — monitoring the status of workflow processes and dynamically configuring the run-time controller.
- Data administration — managing organizations, users, and roles; rerouting tasks when needed; and maintaining business workflow calendars.

WebLogic Process Integrator is a Java 2 Enterprise Edition (J2EE) session and entity bean-based implementation of a workflow management system. At the center of the WebLogic Process Integrator architecture is the Process Engine, which serves as the run-time controller and is responsible for keeping track of workflow instances and managing the execution of workflows.

WebLogic Process Integrator has the following architecture.

**Figure 1-1 WebLogic Process Integrator Architecture**



# Understanding the Workflow Model

The following sections describe the concepts and terminology associated with the workflow model:

- Overview of the Workflow Model
- Workflow Definitions
- Workflow Components
- Workflow Model Diagram

## Overview of the Workflow Model

It is important to understand the following key terms:

- A workflow is a business process (for example, order processing). Workflow automation is the automation of a business process, in whole or in part, during which information of any type is passed to the right participant at the right time according to a set of intelligent business rules that allow computers to perform most of the work while humans only have to deal with exceptions.
- A workflow template is a folder (or a container) in WebLogic Process Integrator Studio. This workflow template represents a workflow and is given a meaningful workflow name, such as Order Processing or Billing. The workflow template aggregates various definitions (or “versions”) of its implementation; these are referred to as workflow template definitions. Further, a workflow template is responsible for controlling which organizations can use the “contained” workflow template definitions.
- A workflow template definition is a definition (or “version”) of the workflow, distinguished by its effective and expiry dates. At run time, WebLogic Process Integrator starts an instance (or session) of a workflow template definition, selecting the most effective (or current and active) definition.

You design a workflow template by using the drawing area of WebLogic Process Integrator Studio to build a business process flowchart. WebLogic Process Integrator Studio is the WebLogic Process Integrator client application that you use to define and monitor workflows. Workflows are designed from the perspective of business analysts, incorporating varied and multiple business activities and transactions involved in business processes.

In WebLogic Process Integrator, one workflow template can be used to generate multiple workflow instances. When a workflow is started, a copy of the workflow template definition is saved and used to generate information in the workflow, such as what user to assign to the first task, what actions are to be executed, when the task is completed, and so on.

Workflow template definition within an organization is usually the responsibility of an analyst, system administrator, or authorized user.

## **Workflow Definitions**

Workflows are defined for and assigned to organizations within WebLogic Process Integrator. An organization can be an entire company, or an organization can represent a division, region, or location within a company. For each organization, users and roles (groups of users sharing common areas of responsibility) are created. An organization can have multiple workflow templates assigned to it.

When a workflow is started, a workflow template definition is instantiated and an instance is created. This instance holds information pertaining to the state of the workflow (running process).

Workflow template definitions are stored as XML documents. XML is a system for defining, validating, and sharing document formats. Like HTML, XML is a markup language derived from SGML (standard generalized markup language). However, unlike HTML, which was created to allow cross-platform formatting of information for display, XML is best suited for organizing data.

# Workflow Components

The following sections briefly describe workflow components.

## Nodes

Nodes are key components within workflows. Workflow nodes are graphically represented as “shapes” in the workflow drawing area. A workflow node can be a start, task, event, join, decision, or done. Nodes are described briefly in the following list. For more information, refer to Chapter 4, “Working with Workflow Components.”

- Start — the beginning of a workflow; initiated by a predefined, timed, manually initiated, or called event.
- Task — core process activities of a workflow; tasks evolve through various task events as the workflow progresses (created, executed, activated, done).
- Event — a notification node that is triggered by an XML message.
- Join — AND (all nodes linked by it must be satisfied before the successor of the join can be activated) or OR (only one of the linked nodes must be satisfied before the successor of the join can be activated).
- Decision — contains a condition that must be evaluated before the succeeding node can be initiated. The condition evaluates to either True or False. (If the condition is verified, it is True. If it is not, it is False.) Depending on the outcome of the evaluation, the workflow can follow different paths.
- Done — workflow completion.

## **Variables**

Workflows also contain variables, which hold run-time values and data obtained often from external sources. Variables are used to control the flow of business processes. For more information, refer to Chapter 6, “Using Expressions and Conditions.”

## **Exception Handlers**

Workflows contain system- and custom-defined workflow exception handlers, which enable you to generate, trap, and respond to internally and externally generated exception conditions. 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. For more information, refer to Chapter 7, “Handling Workflow Exceptions.”

## **Actions**

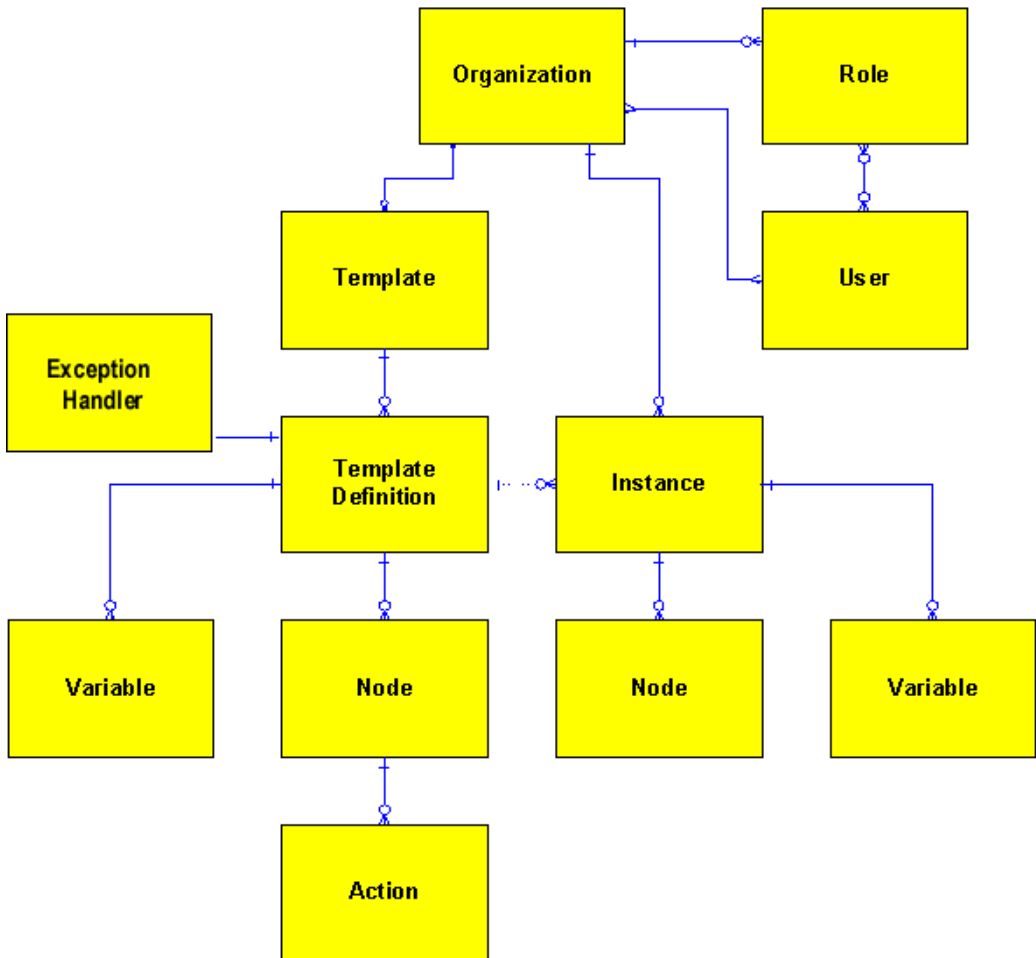
Workflows also contain actions (units of work), which can be as simple as assigning a task to a user or as complicated as sending or receiving XML messages or invoking Enterprise JavaBean (EJB) methods. Actions are, in a sense, the primitive building blocks of a workflow in that they define the actual behavior of the workflow. Actions are integrated into a workflow by specifying them in start nodes, task nodes, decision nodes, done nodes, and events. For more information, refer to Chapter 5, “Using Actions.”

WebLogic Process Integrator provides a standard set of actions, but this set is extendible through use of Enterprise JavaBean (EJB) and Java Class methods. (For more information, refer to “Understanding Business Operations” in Chapter 2, “Administering Data within WebLogic Process Integrator.”)

## Workflow Model Diagram

The following diagram illustrates the WebLogic Process Integrator workflow model.

**Figure 1-2 WebLogic Process Integrator Workflow Model**



1. The workflow model begins at the organization level, where users and roles are created to perform workflow tasks.
2. Workflow templates are created for and attached to the appropriate organization. An organization can have multiple workflow templates.
3. For each workflow template, you define workflow template definitions, which are different “versions” of the workflow template.
4. In defining workflow template definitions, you define workflow nodes, workflow variables, workflow exception handlers, and actions within workflow nodes.
5. When a workflow template definition is instantiated (placed in to run time), an instance of that workflow is created; multiple instances of the same workflow template definition can be running at the same time.

## Administering Data

Users administer components of the WebLogic Process Integrator server and database through the data and server administration facility. More specifically, you can perform the following tasks:

- Add, update, and remove organizations, users and roles.
- Assign users to and remove them from roles.
- Reroute tasks from one user to another for a specified period of time.
- Create calendars that are used in workflow template definitions.

User, role, and organization information within WebLogic Process Integrator is maintained in BEA WebLogic Server™ security realms.

# Monitoring Workflows

WebLogic Process Integrator provides workflow monitoring in the form of two reports: workload and statistics. These can be accessed in the folder tree of the main Studio window. Use the WebLogic Process Integrator monitoring facility to monitor and modify workflows. More specifically, you can perform the following tasks:

- Display the status of workflow instances.
- Modify the tasks associated with a workflow instance. For example, reassign tasks or force work to be redone.
- Graphically display the workload status of the system, showing the number of tasks categorized by status (such as 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.

Workflow monitoring enables you to gather system workload information and make informed decisions based on available data, thus managing your resources more efficiently.

# Starting WebLogic Process Integrator Studio

To start WebLogic Process Integrator Studio:

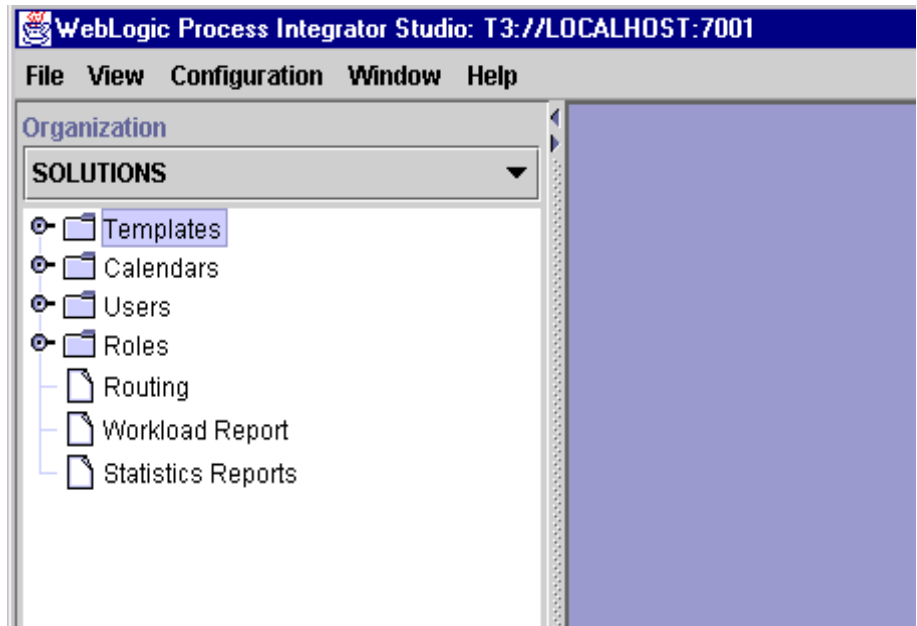
1. Select Start > Programs > WebLogic Process Integrator Studio to display the WebLogic Process Integrator Logon dialog box.

**Figure 1-3 Logon to WebLogic Process Integrator Dialog Box**



2. Enter your user ID, password, and the name of the WebLogic Process Integrator server to which you will connect, and click OK.
3. Click OK to display the main window of WebLogic Process Integrator.

**Figure 1-4 WebLogic Process Integrator Main Window**



# Navigating WebLogic Process Integrator

This section describes the components and tasks of the WebLogic Process Integrator Studio user interface. WebLogic Process Integrator Studio is a client application that allows for the definition and monitoring of workflows.

## Menu Options

The File menu has the following options:

- Logon — logs on to the WebLogic Process Integrator server
- Logoff — logs off from WebLogic Process Integrator server
- Exit — logs off from WebLogic Process Integrator server and exit WebLogic Process Integrator Studio.

The View menu has the following options:

- Refresh — updates the information if changes are made to the database by other client applications
- Use color on flowcharts — allows the user to view the flowcharts in color or in black and white
- Sync selection with tree — synchronizes 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 — choices are Metal, CDE/Motif, and Windows.

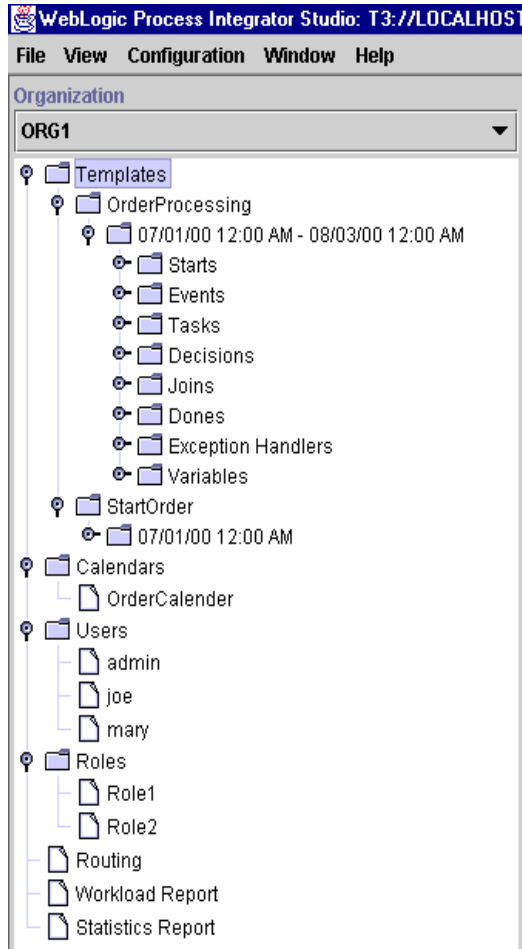
The Configuration menu has the following options:

- Organization — defines organizations, representing different business entities, geographical locations or any other distinction relevant to the particular business of the company
- Business Operations — defines business operations. For more information, see “Understanding Business Operations” in Chapter 2, “Administering Data within WebLogic Process Integrator.”
- Events — defines event key expressions.

## Folder Tree Display

The WebLogic Process Integrator Studio interface contains a folder tree display, which follows standard explorer-type conventions.

**Figure 1-5 WebLogic Process Integrator Folder Tree Display**



**Note:** Displayed information pertains only to the current organization. Select the organization from the Organization drop-down list at the top of the screen.

Double-click one of the following items to expand it and view details about the item:

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

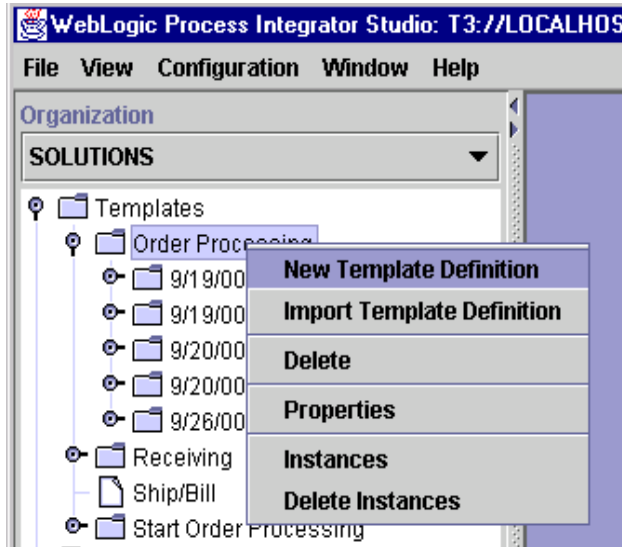
For example, double-click the Templates folder to display a list of workflow templates. Double-click a workflow template to display a list of all workflow template definitions. Expanding a particular workflow template definition displays folders containing the tasks, decisions, events, joins, starts, done, 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 following options:

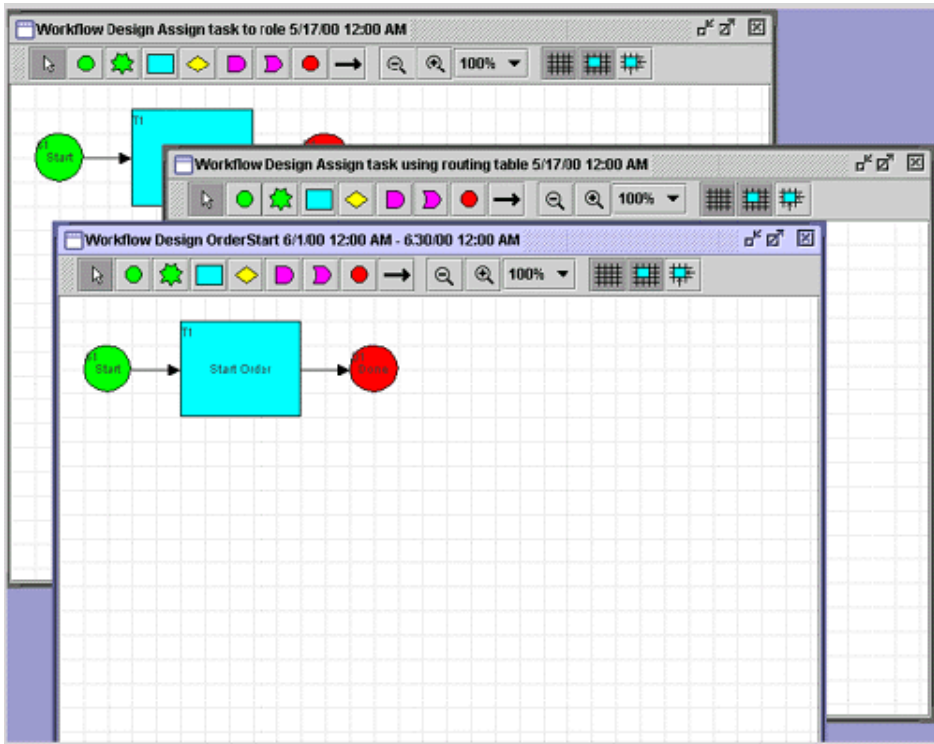
- New Template Definition
- Import Template Definition
- Delete
- Properties
- Instances
- Delete Instances

**Figure 1-6 Workflow Template Menu Options**



WebLogic Process Integrator allows you to have multiple workflow diagrams open simultaneously. It is possible to view more than one workflow and toggle back and forth between them.

**Figure 1-7 Multiple Workflow Diagrams**



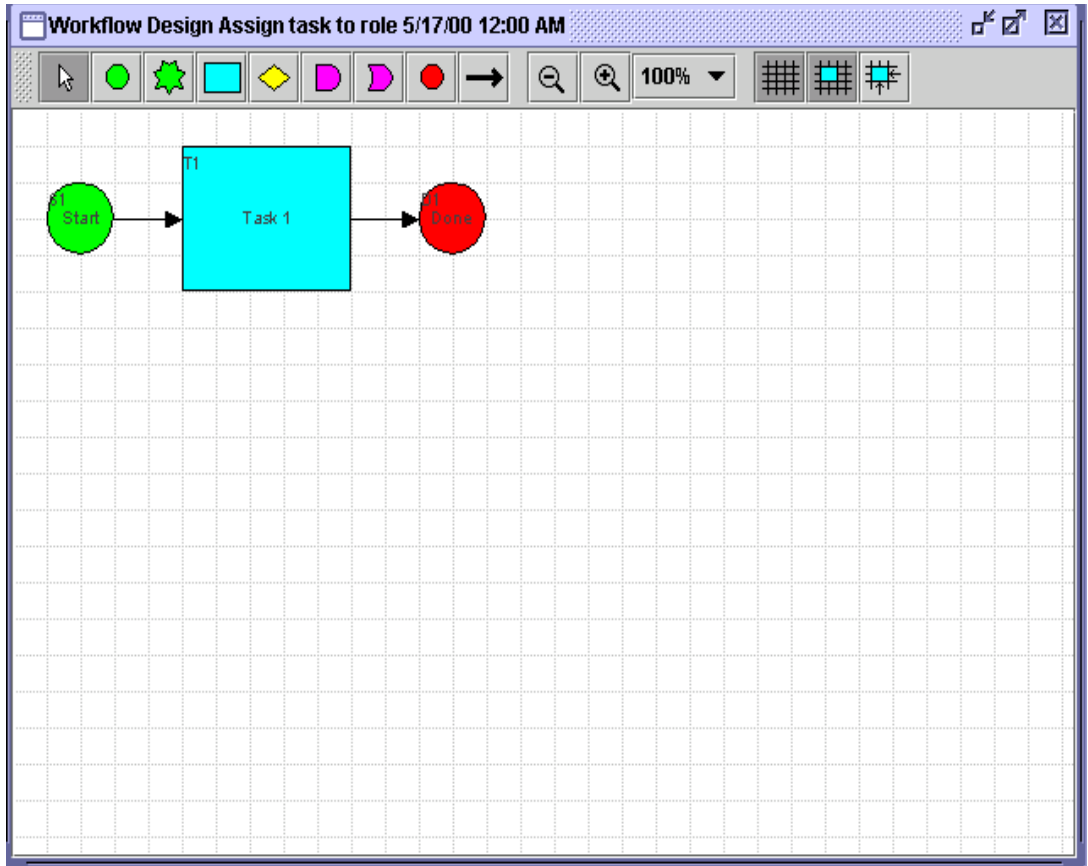
## Workflow Drawing Area

Use the shapes (representing the various nodes or components of a workflow) on the workflow toolbar to draw a workflow. The shapes somewhat simplify the complexity involved in designing workflows. Workflow shapes and connectors can be selected and dropped onto the workflow drawing area, allowing you to modify workflows as needed.

To open the workflow drawing area, shown in the following figure, create a new workflow template definition or open an existing workflow template definition. Refer to Chapter 3, “Defining and Maintaining Workflows,” for detailed instructions on performing these tasks.

To place a shape in the drawing area, click the shape on the toolbar, place your cursor on the drawing area, and click again to drop the shape onto the drawing area. To move a shape within the diagram, click and drag the shape with your mouse.

**Figure 1-8 Workflow Drawing Area**



The following table describes the various node shapes that WebLogic Process Integrator provides for workflow template definition.

**Table 1-1 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 with an AND condition.                                                                                                                            |
|  | Or         | Allows joining of one or more task, decision, or event 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. |

**Table 1-1 Workflow Template Definition Nodes**

| Symbol                                                                            | Shape Name | Purpose                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Connector  | <p>Used to connect workflow nodes. The arrow directs you to the subsequent task in the flow.</p> <p><b>Note:</b> 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.</p> |

Each decision contains a condition. This condition, if verified, evaluates to True; if it is not verified, it evaluates to False. In a workflow, a decision is followed by one or more nodes, which are dependent on either a True or False condition. Depending on whether the condition results evaluate to True or False, one or more subsequent nodes are instantiated.

Having more than one possible outcome from a decision is not mandatory, since actions can be set to False or True within a decision; also, a workflow can be aborted upon a True or False, in which case there will be no True or False connector resulting from the decision.

## Exiting WebLogic Process Integrator Studio

To exit WebLogic Process Integrator 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. Select 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 WebLogic Process Integrator Studio without having saved your changes, a dialog box prompts you to do so.

# 2 Administering Data within WebLogic Process Integrator

The following sections describe the data administration tasks within WebLogic Process Integrator:

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

# Security

User, role, and organization information within WebLogic Process Integrator is maintained in BEA WebLogic Server™ security realms. Creating a new user, role, or organization within WebLogic Process Integrator means adding those entities to the WLS security realm. Therefore, the functions of creating and deleting users, roles, and organizations are accessible only if a manageable realm is used and detected by WebLogic Process Integrator.

When using a non-manageable security realm, such as the WebLogic properties file, only certain updates can be performed to the data kept within the WebLogic Process Integrator database. For example, the business calendar attached to a user, role, or organization can be changed. However, in order to add a new user, role, or organization, or to add a user to a role or an organization, the addition must be made directly within the security realm and not through use of the WebLogic Process Integrator user interface. WebLogic Process Integrator cannot persist information to the security realm.

In the sections that follow, the following functions can be performed only when using a manageable realm:

- Create New User, Role, or Organization
- Delete User, Role, or Organization
- Add a User to an Organization
- Remove a User from an Organization
- Add a User to a Role
- Remove a User from a Role

# 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.

The Organization drop-down list is located on the main menu of the 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

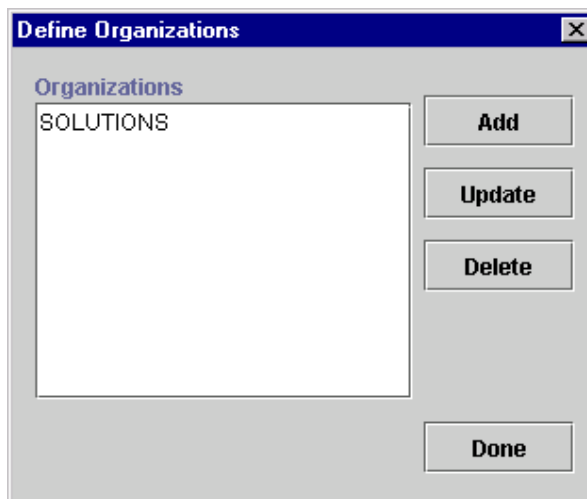
The Add Organization facility adds a group, representing the organization, to the security realm and also adds the organization to the WebLogic Process Integrator database.

This facility is available from the WebLogic Process Integrator Studio application only if it is operating within a manageable security realm.

To add an organization:

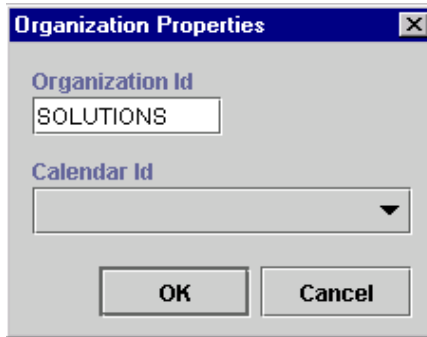
1. From the Configuration menu, choose Organizations to display the Define Organizations dialog box.

**Figure 2-1 Define Organizations Dialog Box**



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

**Figure 2-2 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. See “Administering Business Calendars” for details on this feature.
5. Click OK.

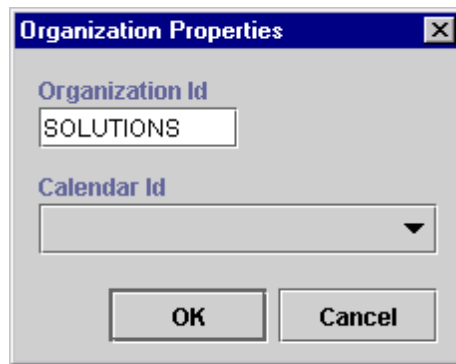
# Updating an Organization

The Update Organization facility allows you to update the business calendar of an existing organization; it is accessible to you regardless of whether WebLogic Process Integrator Studio operating within a manageable realm.

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 the Update button to display the Organization Properties dialog box.

**Figure 2-3 Organization Properties Dialog Box**



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

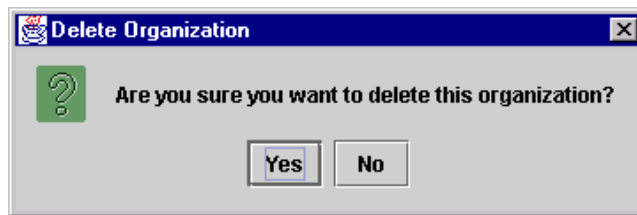
## Deleting an Organization

The Delete Organization facility allows you to remove an organization from the security realm as well as from the WebLogic Process Integrator database. This facility is only available to you if WebLogic Process Integrator Studio is operating within a manageable realm.

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 the Delete button to display the Delete Organization dialog box.

**Figure 2-4 Delete Organization Dialog Box**



4. Click Yes to confirm or No to cancel.

# Maintaining Users

Create, update, and delete users with the Users feature. You can also add or remove users from organizations.

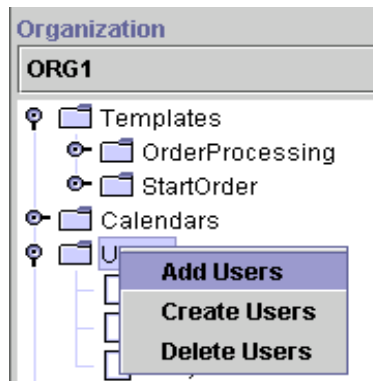
A user is an individual defined in WebLogic Process Integrator who has permissions to perform tasks. A role is an area of responsibility, ability, or authorization level that is shared by a particular group of individuals.

The User folder is located in the 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:

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

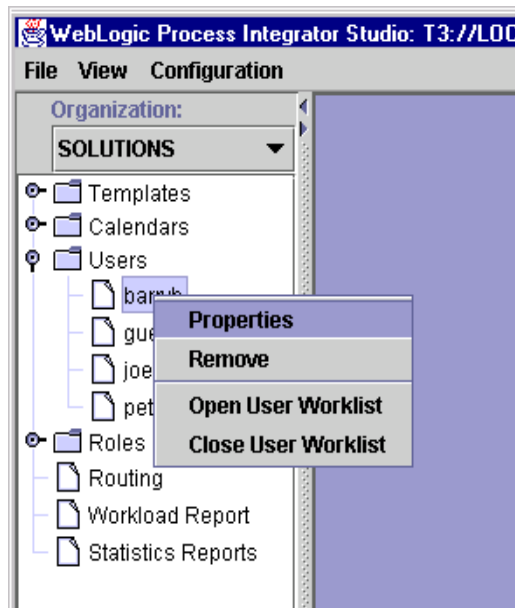
**Figure 2-5 User Folder Menu Options**



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** — 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. (See “Using Workload Reports” in Chapter 8, “Monitoring Workflows.”)
- **Close User Worklist** — closes the list of tasks assigned to the user.

**Figure 2-6 Users Menu Options**



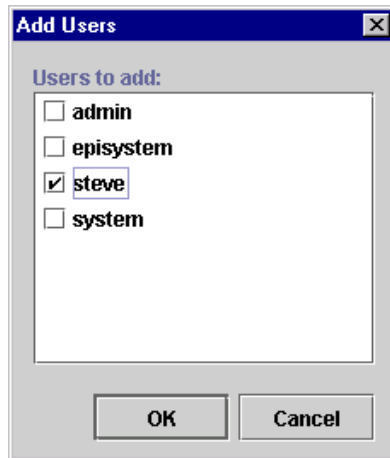
# Adding a User to the Current Organization

The Add User facility adds a user that is already defined in the WebLogic Server (WLS) 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.

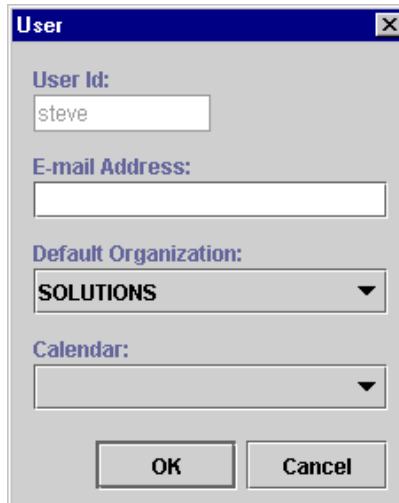
To add a user to the current WebLogic Process Integrator organization:

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

**Figure 2-7 Add Users Dialog Box**



2. A list of valid users is displayed. (Valid users are those users registered in WLS). 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 2-8 User Dialog Box**A screenshot of a Windows-style dialog box titled "User". It contains four labeled input fields: "User Id:" with a text box containing "steve", "E-mail Address:" with an empty text box, "Default Organization:" with a dropdown menu showing "SOLUTIONS", and "Calendar:" with an empty dropdown menu. At the bottom are "OK" and "Cancel" buttons.

User

User Id:  
steve

E-mail Address:

Default Organization:  
SOLUTIONS

Calendar:

OK Cancel

The User dialog box contains the following fields:

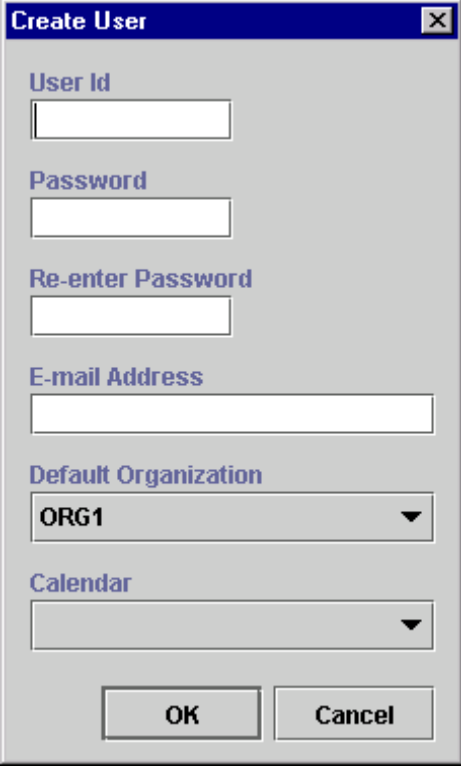
- User Id — uniquely defines the user. The user enters this ID to log in to WebLogic Process Integrator applications. In the User dialog box, this field is already populated with the name of the user selected in Step 2.
  - E-mail Address — enter the e-mail address of the user.
  - Default Organization — assign one default organization to the user.
  - Calendar — select the appropriate business calendar to assign to the user. See “Administering Business Calendars” for details on this feature.
4. Click OK to save the new user information, or click Cancel to cancel the operation.

# Creating a User

The Create User option is available only if the security realm is a manageable one. To create a new 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 2-9 Create User Dialog Box**

The image shows a 'Create User' dialog box with a blue title bar and a close button. It contains several input fields and dropdown menus. The fields are labeled 'User Id', 'Password', 'Re-enter Password', and 'E-mail Address'. The 'Default Organization' dropdown menu is set to 'ORG1'. The 'Calendar' dropdown menu is empty. At the bottom are 'OK' and 'Cancel' buttons.

**Create User**

**User Id**

**Password**

**Re-enter Password**

**E-mail Address**

**Default Organization**  
ORG1 ▼

**Calendar**  
▼

OK Cancel

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 WebLogic Process Integrator client applications.
  - Password and Re-enter Password — enter a password for the user.

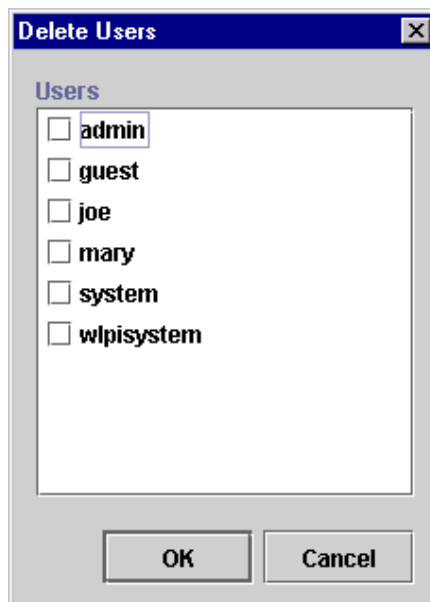
- E-mail Address — optionally enter the e-mail address of the user.
- Default Organization — select a default organization for the user.
- Calendar — optionally select a business calendar for the user.

## 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 2-10 Delete Users Dialog Box**



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:

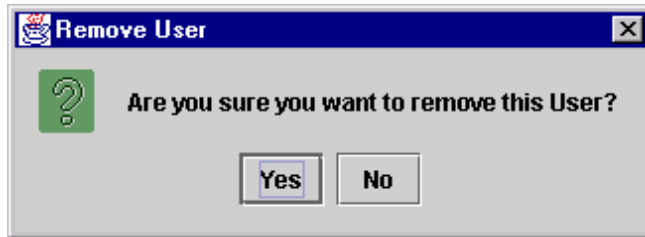
1. Right-click a user and choose Properties to display the User dialog box.
2. Make the necessary changes to the E-mail Address, Default Organization, and Calendar.
3. Click OK to save the update the user information, or click Cancel to cancel the operation.

# Removing a User from the Current Organization

The Remove User facility removes a user that is already defined in the WebLogic Server (WLS) security realm from the current organization. It does not remove the user from the WLS 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 2-11 Remove User Dialog Box**

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

## Maintaining Roles

Use the Roles feature to create, update, and delete roles. A role is a common area of responsibility, ability, or authorization level that is shared by a group of individuals. A single user can belong to one or more roles.

Roles are displayed within each organization. Selecting an organization from the Organizations drop-down list in the WebLogic Process Integrator main window and double-clicking the Roles folder displays the roles belonging to that particular organization. Double-clicking on 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 (WLS) 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. (See “Using Workload Reports” in Chapter 8, “Monitoring Workflows.”)
- Close Role Worklist — closes the list of tasks assigned to the role.

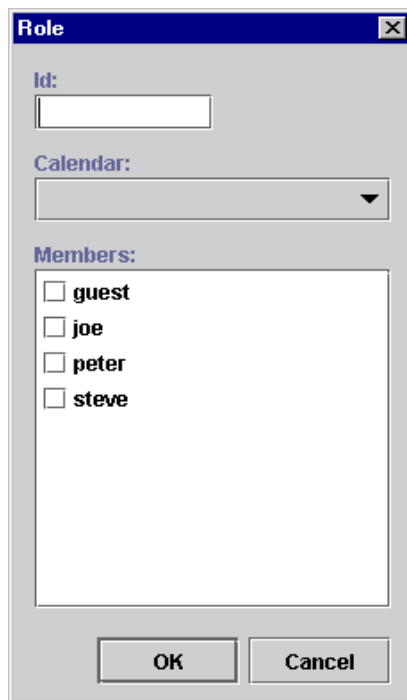
# Creating a Role

The Create Role facility allows you to create a new role in the WebLogic Server (WLS) security realm and is accessible only if WebLogic Process Integrator Studio is operating with 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 2-12 Role Dialog Box**

The image shows a 'Role' dialog box with a blue title bar and a close button. It contains three main sections: 'Id:' with a text input field; 'Calendar:' with a drop-down menu; and 'Members:' with a list of four users: 'guest', 'joe', 'peter', and 'steve', each preceded by an unchecked checkbox. At the bottom are 'OK' and 'Cancel' buttons.

| Role                 |                                                                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Id:</b>           | <input type="text"/>                                                                                                                             |
| <b>Calendar:</b>     | <div>▼</div>                                                                                                                                     |
| <b>Members:</b>      | <div><input type="checkbox"/> guest<br/><input type="checkbox"/> joe<br/><input type="checkbox"/> peter<br/><input type="checkbox"/> steve</div> |
| <div>OK Cancel</div> |                                                                                                                                                  |

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. See “Administering Business Calendars” for details on this feature.
4. 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.

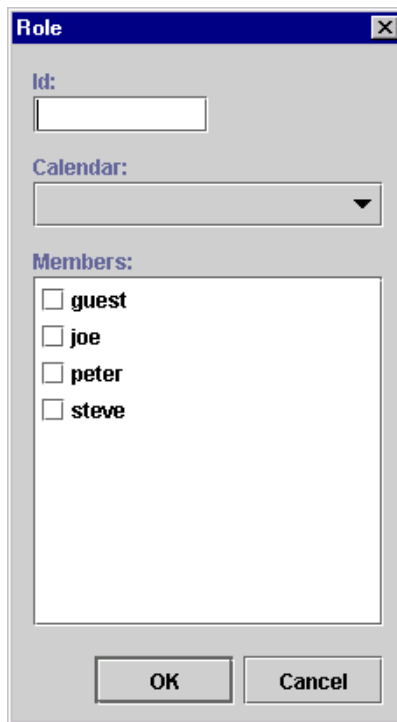
5. Click OK to save the new role. Click Cancel to cancel the operation.

## 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 2-13 Role Dialog Box**

The image shows a 'Role' dialog box with a blue title bar and a close button. It contains three sections: 'Id:' with a text input field, 'Calendar:' with a dropdown menu, and 'Members:' with a list of four members: 'guest', 'joe', 'peter', and 'steve', each preceded by an unchecked checkbox. At the bottom are 'OK' and 'Cancel' buttons.

**Role**

**Id:**

**Calendar:**

**Members:**

- ☐ guest
- ☐ joe
- ☐ peter
- ☐ steve

**OK** **Cancel**

2. Make changes as needed to the calendar and role members.
3. Click OK to save the update. Click Cancel to cancel the operation.

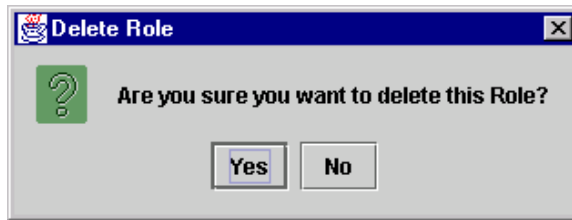
# Deleting a Role

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 2-14 Delete Role Dialog Box**



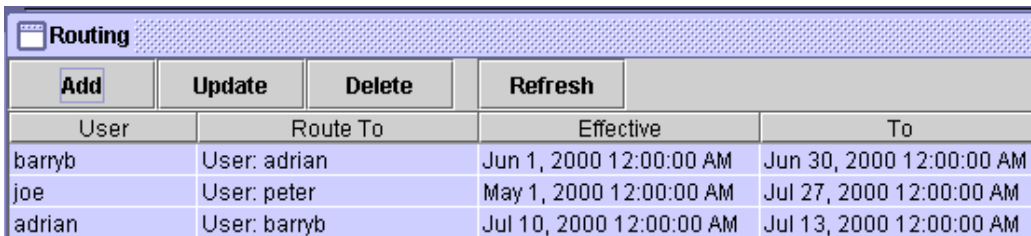
2. To delete the role, click Yes. To cancel the delete, click No.

# 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 2-15 Routing Dialog Box**

The screenshot shows a 'Routing' dialog box with a title bar and a table. The title bar has a 'Routing' label and a close button. The table has four columns: 'User', 'Route To', 'Effective', and 'To'. There are three rows of data. The first row shows 'barryb' routing to 'User: adrian' from 'Jun 1, 2000 12:00:00 AM' to 'Jun 30, 2000 12:00:00 AM'. The second row shows 'joe' routing to 'User: peter' from 'May 1, 2000 12:00:00 AM' to 'Jul 27, 2000 12:00:00 AM'. The third row shows 'adrian' routing to 'User: barryb' from 'Jul 10, 2000 12:00:00 AM' to 'Jul 13, 2000 12:00:00 AM'. Above the table are buttons for 'Add', 'Update', 'Delete', and 'Refresh'.

| 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 Task Routing Specification

To reroute a task:

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

**Figure 2-16 Reroute Tasks Dialog Box**

**Reroute Tasks**

**Route From:**  
barryb

**Route To:**  
adrian

☒ User  
☐ User In Role  
☐ Role

**Effective:**

| 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:**

| 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  |     |     |

OK Cancel

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 on 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 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.")
- Users (See "Adding a User to the Current Organization.")
- Roles (See "Creating a Role.")
- Time-related actions:
  - Set Task Due Date action (See "Set Task Due Date" in Chapter 5, "Using Actions.")
  - Timed Event action (See "Timed Event" in Chapter 5, "Using Actions.")

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 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 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 2-17 Calendar Properties Dialog Box**

**Properties**

Name:

Time Zone: **ACT**

**Rules**

**Period**

**From**

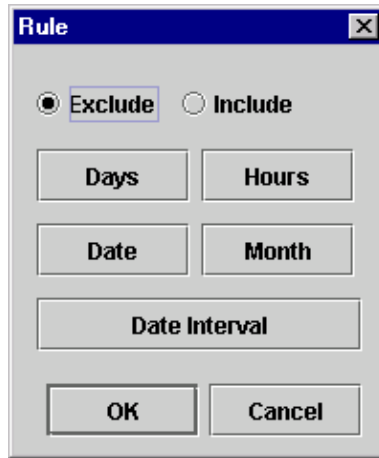
| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
|     | 1   | 2   | 3   | 4   | 5   | 6   |
| 7   | 8   | 9   | 10  | 11  | 12  | 13  |
| 14  | 15  | 16  | 17  | 18  | 19  | 20  |
| 21  | 22  | 23  | 24  | 25  | 26  | 27  |
| 28  | 29  | 30  | 31  |     |     |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

**To**

| 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 the Add button. This displays the Rule dialog box.

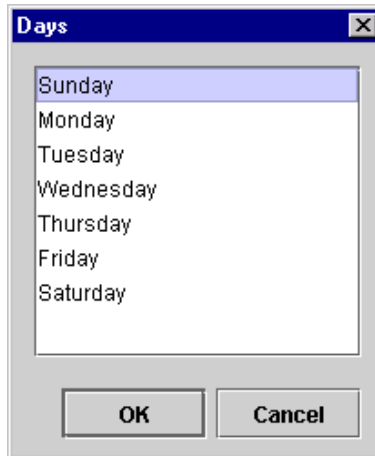
**Figure 2-18 Rule Dialog Box**



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.

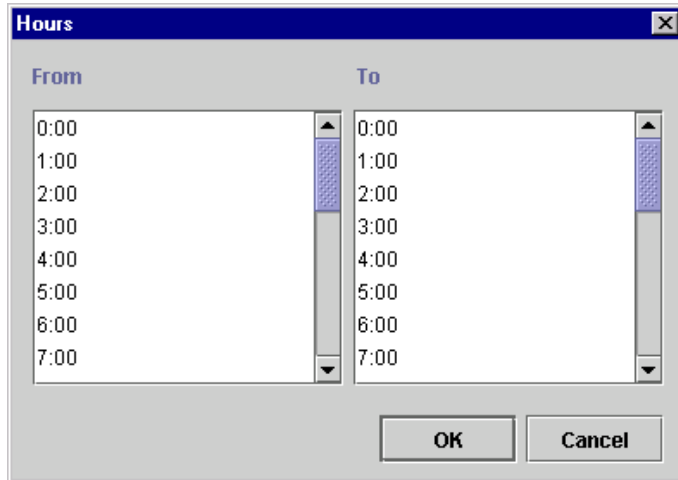
6. Define calendar rules using the following buttons:
- Days — displays the Days dialog box. Excludes or includes certain calendar days. (Hold the CTRL key down to select more than one day of the week from the Days dialog box.)

**Figure 2-19 Days Dialog Box**



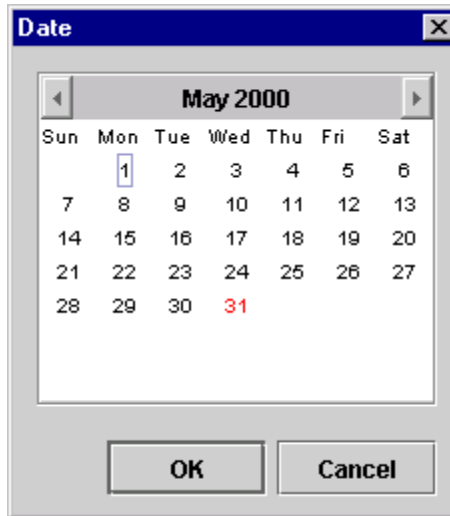
- Hours — displays the Days dialog box. Excludes or includes certain hours of the day. Select a From and To value.

**Figure 2-20 Hours Dialog Box**



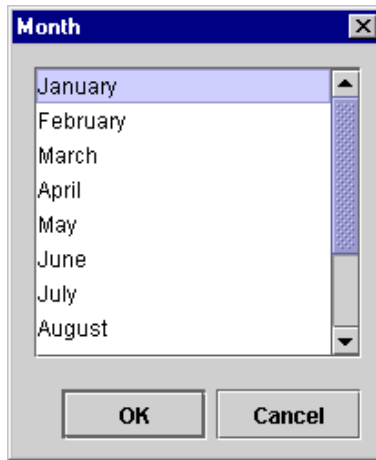
- Date — displays the Date dialog box. Excludes or includes a certain date.

**Figure 2-21 Date Dialog Box**



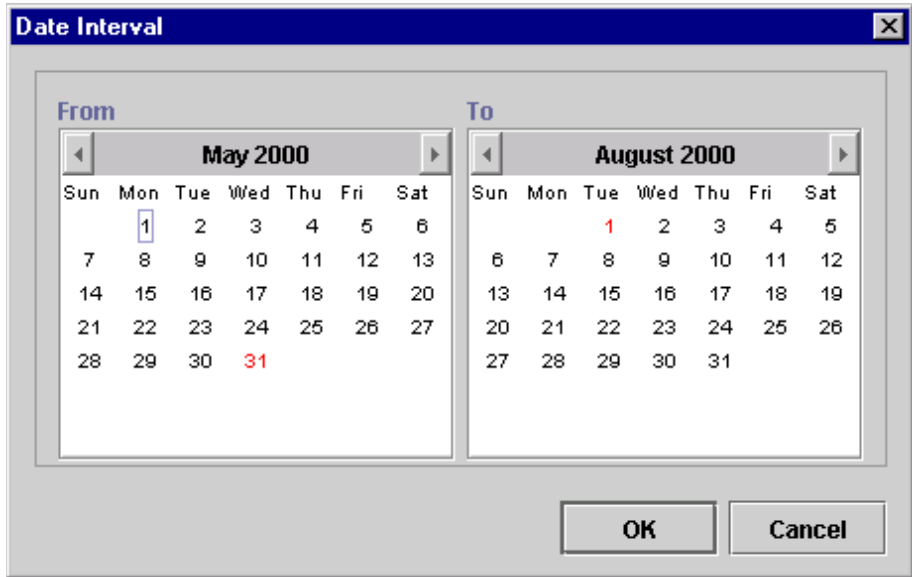
- Month — displays the Month dialog box. Excludes or includes certain month(s) of the year. (Hold the CTRL key down while clicking with your mouse to select more than one month of the year from the Month dialog box.)

**Figure 2-22 Month Dialog Box**



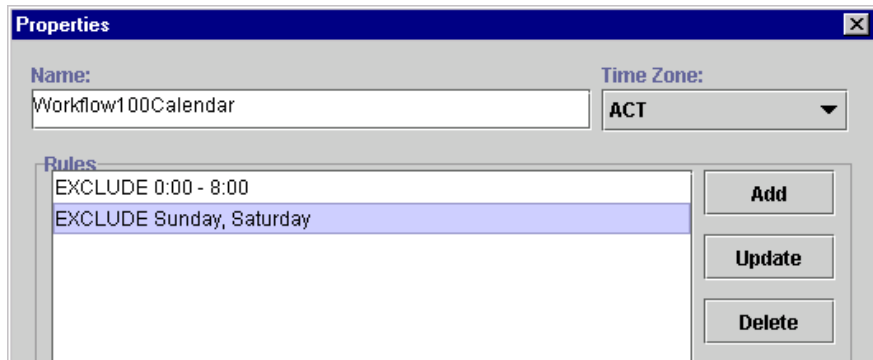
- Date Interval — displays the Date Interval dialog box. Enables you to specify a specific date interval to be excluded or included.

**Figure 2-23 Date Interval Dialog Box**



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

**Figure 2-24 Calendar Rules**



**Note:** You can update or delete a calendar rule by highlighting the rule in the Rules portion of the Calendar Properties dialog box, clicking the Update or Delete button, 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 the OK button to save your changes, or click Cancel to terminate the operation.

## Updating a Calendar

To update an existing calendar:

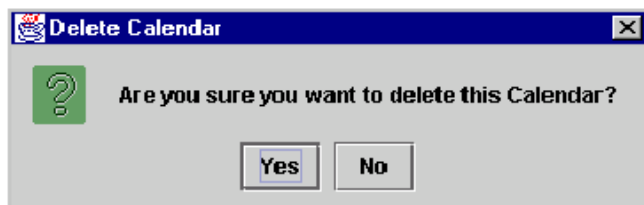
1. In the folder tree, right-click the calendar you want to update, and choose Properties.
2. Make the necessary changes to the business days and hours.
3. Click OK to save the changes. Click Cancel to cancel the operation.

## 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 2-25 Delete Calendar Dialog Box**



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

# Understanding Business Operations

The WebLogic Process Integrator Business Operations facility enables you to display all Enterprise JavaBeans (EJBs) or Java classes that are registered in WebLogic Server and their remote methods and parameters. Using this Business Operations facility, you provide a descriptive name for EJB/Java class method and its parameters that you wish to deploy, hence providing non-technical users with an understanding of the business meaning behind the bean and its methods.

In WebLogic Process Integrator, a business operation represents a method call on an EJB or Java Class instance. This method call is added to a workflow in WebLogic Process Integrator through use of the Perform Business Operation action. The invocation results of this action can be assigned to the workflow variables.

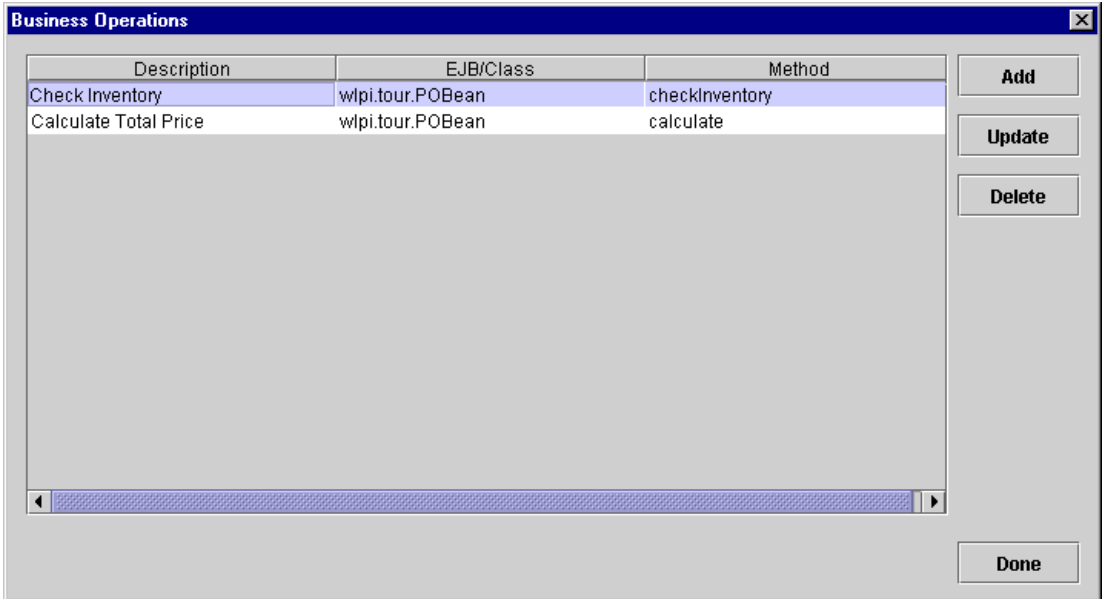
EJBs represent a component architecture for developing and deploying object-oriented, distributed, enterprise-level applications. The Business Operations feature, in effect, allows for the creation of customized actions, which create the connection between WebLogic Process Integrator and end-user applications and components.

In WebLogic Process Integrator, workflows can instantiate serializable Java objects by invoking their constructors as business operations and can store the resulting object reference in a variable. Such serializable Java objects are persisted as part of a workflow instance. Additionally, calls to stateful session EJBs are supported; this requires support of non-persistent workflow variables. The content of such variable types are cleared at transaction boundaries. (See “Understanding the WebLogic Process Integrator Transaction Model” in Chapter 7, “Handling Workflow Exceptions.”)

# Using the Business Operations Dialog Box

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

**Figure 2-26 Business Operations Dialog Box**



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

- Description — description assigned to the business operation. See “Defining a Business Operation” for more information.
- EJB/Class — Enterprise JavaBean (EJB) or Java class to be invoked.
- Method — associated EJB method.
- 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.

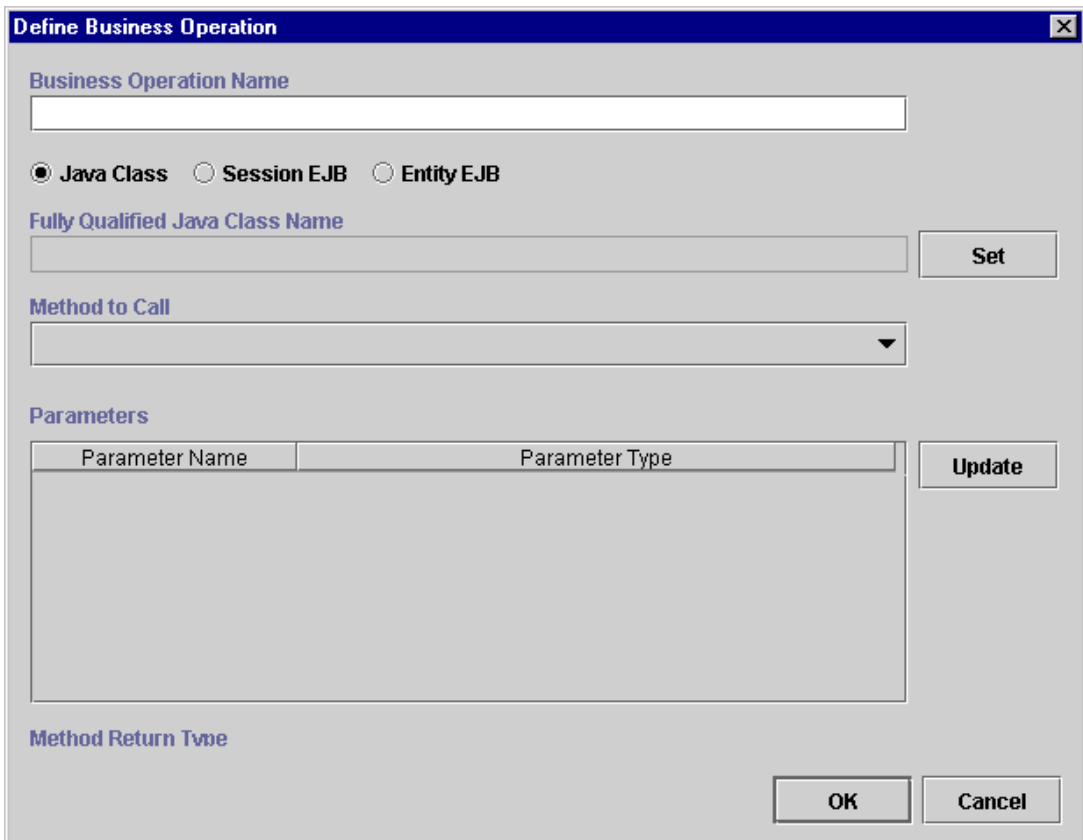
- Delete — deletes the highlighted business operation.

## Defining a Business Operation

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

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

**Figure 2-27 Define Business Operation Dialog Box**



The dialog box is titled "Define Business Operation" and contains the following fields and controls:

- Business Operation Name:** A text input field.
- Radio Buttons:** Three radio buttons labeled "Java Class" (selected), "Session EJB", and "Entity EJB".
- Fully Qualified Java Class Name:** A text input field with a "Set" button to its right.
- Method to Call:** A dropdown menu.
- Parameters:** A table with two columns: "Parameter Name" and "Parameter Type". An "Update" button is to the right of the table.
- Method Return Type:** A text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

| Parameter Name | Parameter Type |
|----------------|----------------|
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |
|                |                |

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

Next, you will select the type of business operation that you are defining. The choices are:

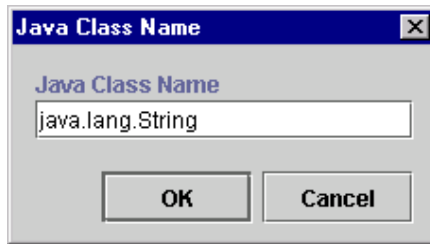
- **Java Class** — used in a business operation that wraps a call to a standard Java class that is accessible on the server. In order for a Java class to be wrapped to the server, it must be a serializable class. Java classes are persisted through the workflow instance.
- **Session EJB** — used in a business operation that wraps a session-type EJB on the server. A session EJB is a server-side process EJB that acts as an interface between the server and the client. Although Session EJBs can be placed in workflow variables, they are not persisted with workflow instances. Instead they exist only for the life of the transaction in which they are created.
- **Entity EJB** — used in a business operation that wraps an entity-type EJB on the server. An entity EJB represents an internal state of the WebLogic Process Integrator engine; the client accesses that state indirectly through Session EJBs. Entity EJBs are persisted through the workflow instance.

## Defining Business Operations for Java Classes

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

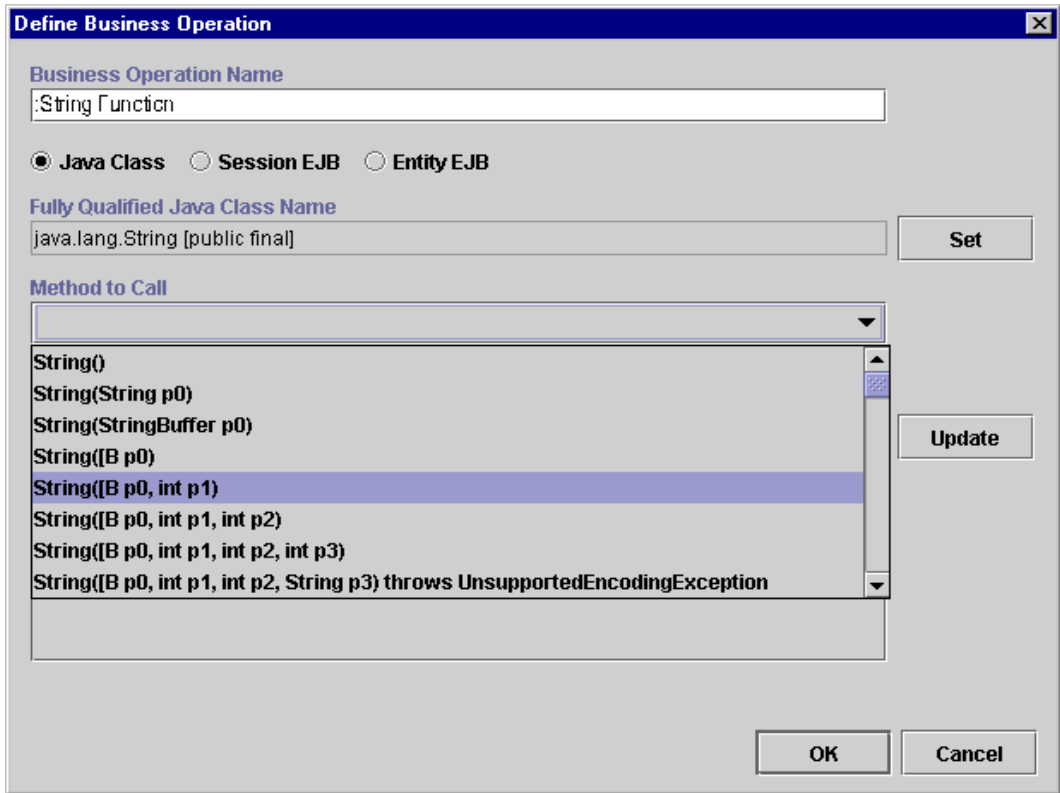
1. Select the Java Class radio button.
2. Click the Set button to display the Java Class Name dialog box.

**Figure 2-28 Java Class Name Dialog Box**



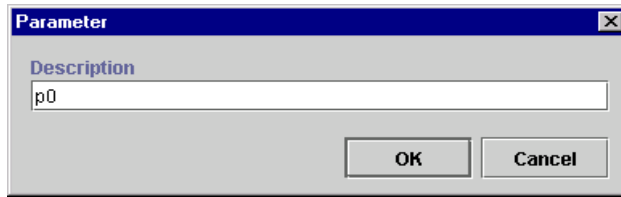
3. Enter a fully qualified Java class name (for example, `java.lang.String`), and click OK. The Java class name is set in the Fully Qualified Java Class Name field of the Define Business Operation dialog box.
4. Use the Method to Call drop-down list to select a method to call upon invocation of the business 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 2-29 Method to Call Drop-down List



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 the Update button to display the Parameter dialog box.

**Figure 2-30 Parameter Dialog Box**



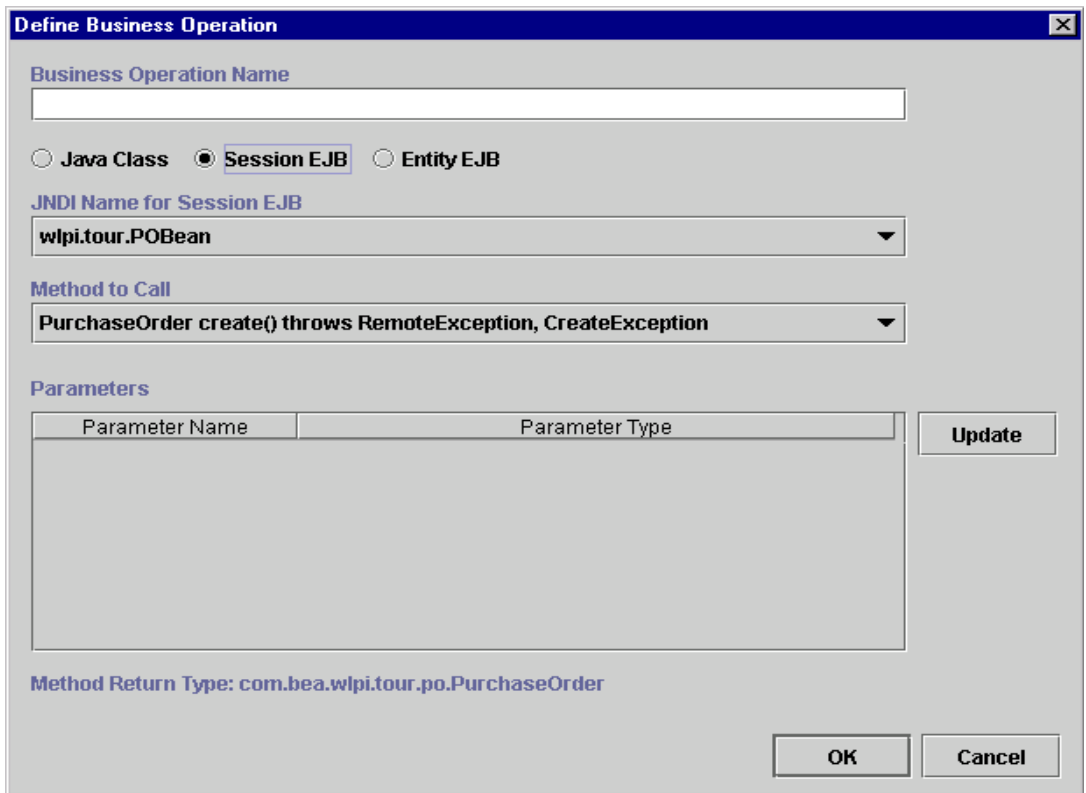
6. Change the parameter name in the Description field, and click OK.

### 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 2-31 Define Business Operation: Session EJB**



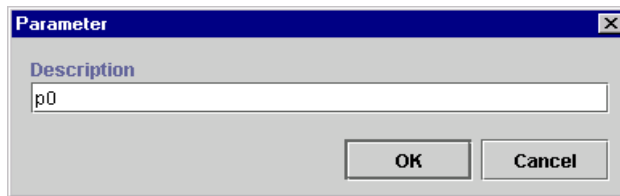
The dialog box titled "Define Business Operation" contains the following fields and controls:

- Business Operation Name:** A text input field.
- Radio Buttons:** Three options are available: ☐ Java Class, ☒ Session EJB, and ☐ Entity EJB.
- JNDI Name for Session EJB:** A dropdown menu showing "wlpi.tour.POBean".
- Method to Call:** A dropdown menu showing "PurchaseOrder create() throws RemoteException, CreateException".
- Parameters:** A table with two columns: "Parameter Name" and "Parameter Type". The table is currently empty.
- Update:** A button located to the right of the Parameters table.
- Method Return Type:** A text label showing "com.bea.wlpi.tour.po.PurchaseOrder".
- OK and Cancel:** Buttons at the bottom right of the dialog.

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 2-32 Parameter Dialog Box**



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 2-33 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 |

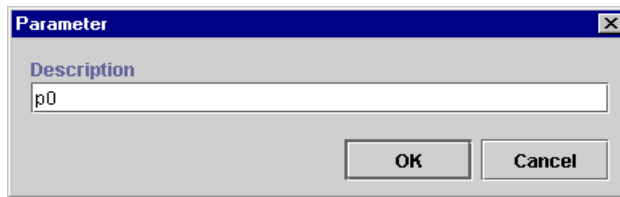
**Method Return Type:** com.bea.wlpi.server.reroute.Reroute

**Update** **OK** **Cancel**

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 the Update button to display the Parameter dialog box.

**Figure 2-34 Parameter Dialog Box**



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” in Chapter 5, “Using Actions.”



# 3 Defining and Maintaining Workflows

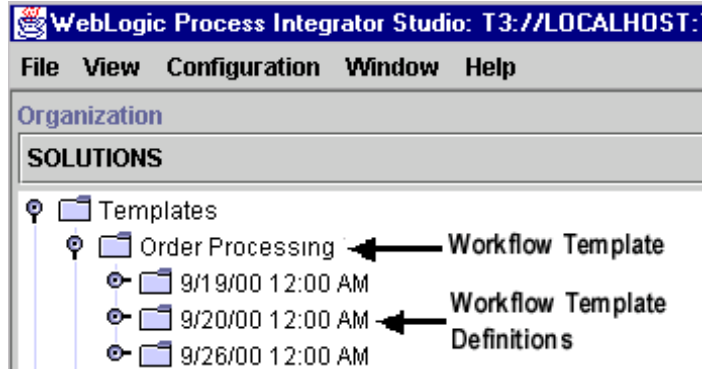
The following sections introduce you to WebLogic Process Integrator workflow templates and workflow template definitions. It also explains how to perform a number of workflow template definition and maintenance tasks.

- 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 Workflow Template Definitions
- Importing Workflow Template Definitions
- Displaying Workflow Instances

# Understanding Workflow Templates and Workflow Template Definitions

A workflow template is, in essence, a folder or a container for WebLogic Process Integrator workflow template definitions. Each workflow template can hold one or more WebLogic Process Integrator workflow template definitions. WebLogic Process Integrator workflow template definitions are identified in the folder tree by an Effective and Expiry date. (See “Understanding Effective and Expiry Dates.”)

**Figure 3-1 Workflow Templates and Workflow Template Definitions**

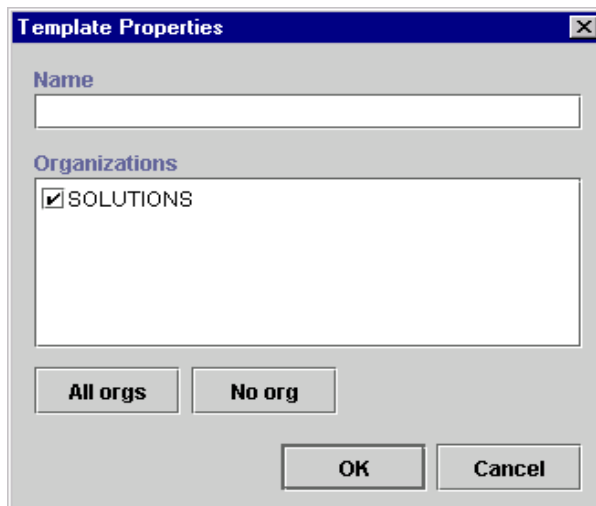


## Creating a New Workflow Template

To create a new workflow template:

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

**Figure 3-2 Template Properties Dialog Box**



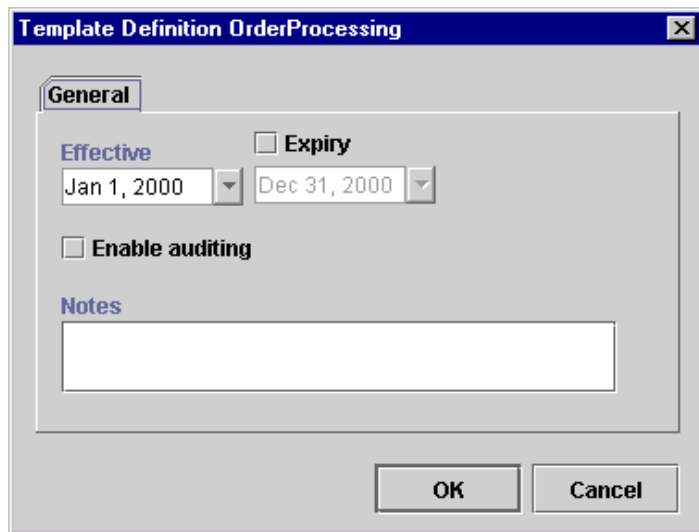
3. In the Name field, enter a meaningful name for the workflow template. All workflow template definitions defined within this workflow template use this name for identification, once they are instantiated (placed into run time) and become workflow instances. The name must be unique.
4. In the Organizations 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 organization, click No Org.
6. Click OK. The new workflow template appears under Templates in the folder tree of the 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 3-3** Template Definition Dialog Box



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

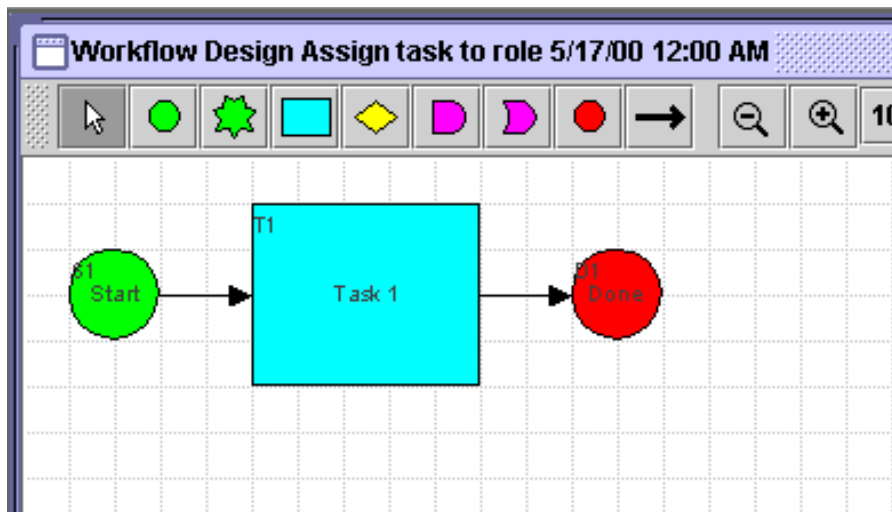
- Enable auditing — allows you to send audit and error messages to the WebLogic Process Integrator audit topic: `wlpi.AuditTopic`. The JMS topic for audit entries defaults to `wlpi.AuditTopic`. This can be changed in the WebLogic properties file.
- Notes — free-form optional field where general comments 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 WebLogic Process Integrator.

**Figure 3-4 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. (See “Creating a New Workflow Template Definition.”)
- Exception Handlers — assigns system and custom-defined workflow exception handlers to the workflow template definition. (See Chapter 7, “Handling Workflow Exceptions.”)

Figure 3-5 Template Definition Dialog Box: General Tab

The screenshot shows a dialog box titled "Template Definition OrderProcessing" with a close button (X) in the top right corner. It has two tabs: "General" (selected) and "Exception Handlers". The "General" tab contains the following fields and controls:

- Id**: A text input field with a search icon (A+B) to its right.
- Active**: An unchecked checkbox.
- Effective**: A date dropdown menu showing "Jul 1, 2000".
- Expiry**: A checked checkbox next to a date dropdown menu showing "Aug 3, 2000".
- Enable auditing**: An unchecked checkbox.
- Notes**: A large text area for notes.
- Last changed on**: A date/time dropdown menu showing "Nov 1, 2000 11:09:00 AM".
- Last changed by**: A text input field containing the name "mary".

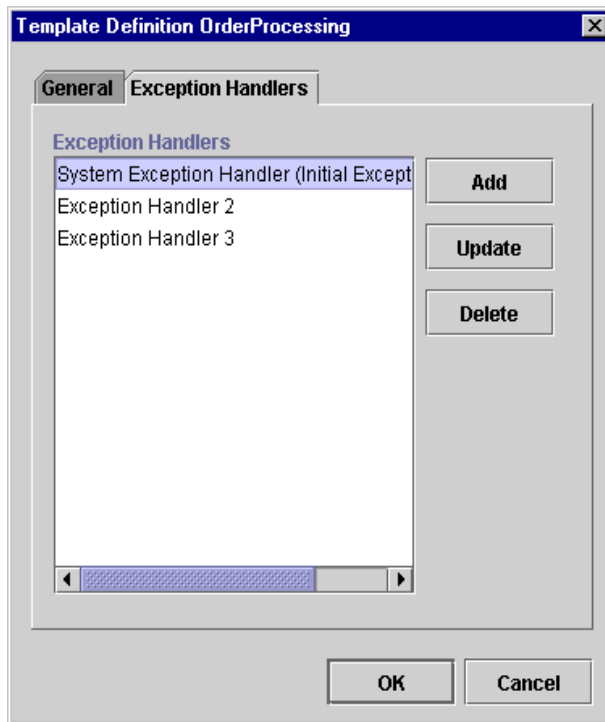
At the bottom right of the dialog are "OK" and "Cancel" buttons.

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 6, “Using Expressions and Conditions.”

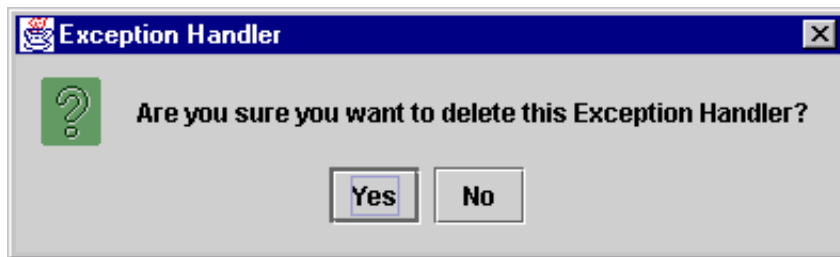
- **Active** — if selected, indicates that the workflow template definition is active and available for processing. A new workflow template definition defaults to the inactive status and must be activated before it can be processed. Often workflow template definitions that are currently in development 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.
- **Last Changed On** — read-only field showing the date that the workflow template definition was last updated.
- **Last Changed By** — read-only field showing the userID of the last person to change the workflow template definition.

**Figure 3-6 Template Definition Dialog Box: Exception Handlers Tab**



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.

**Figure 3-7** Delete Exception Handler



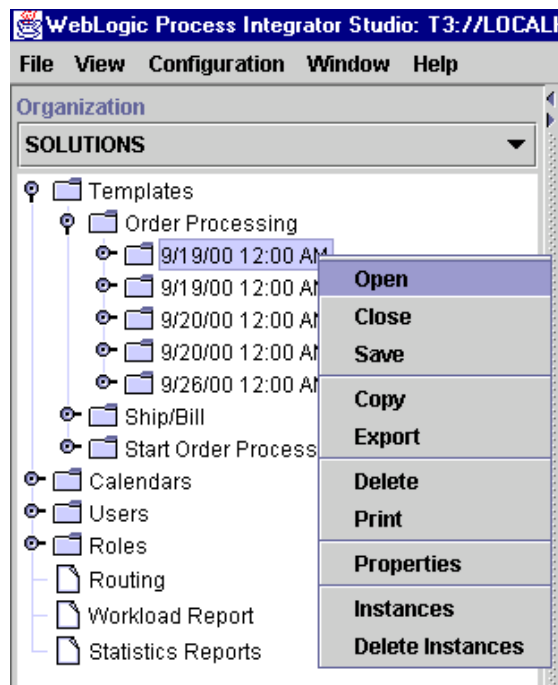
**Note:** For detailed instructions on using the Exception Handler Properties dialog box, see Chapter 7, “Handling Workflow Exceptions.”

# Opening Workflow Template Definitions

To open an existing workflow template definition:

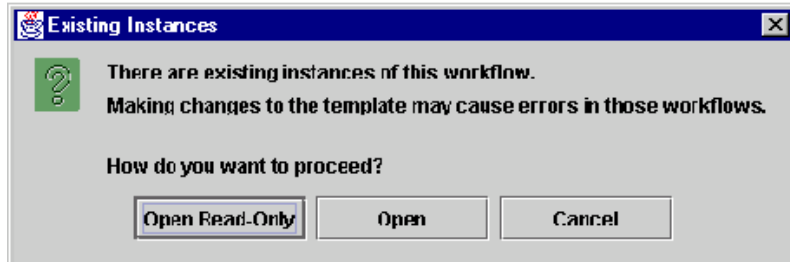
1. Right-click the workflow template definition in the folder tree, and choose Open from the pop-up menu.

**Figure 3-8 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 3-9 Existing Instances Dialog Box



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 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 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.

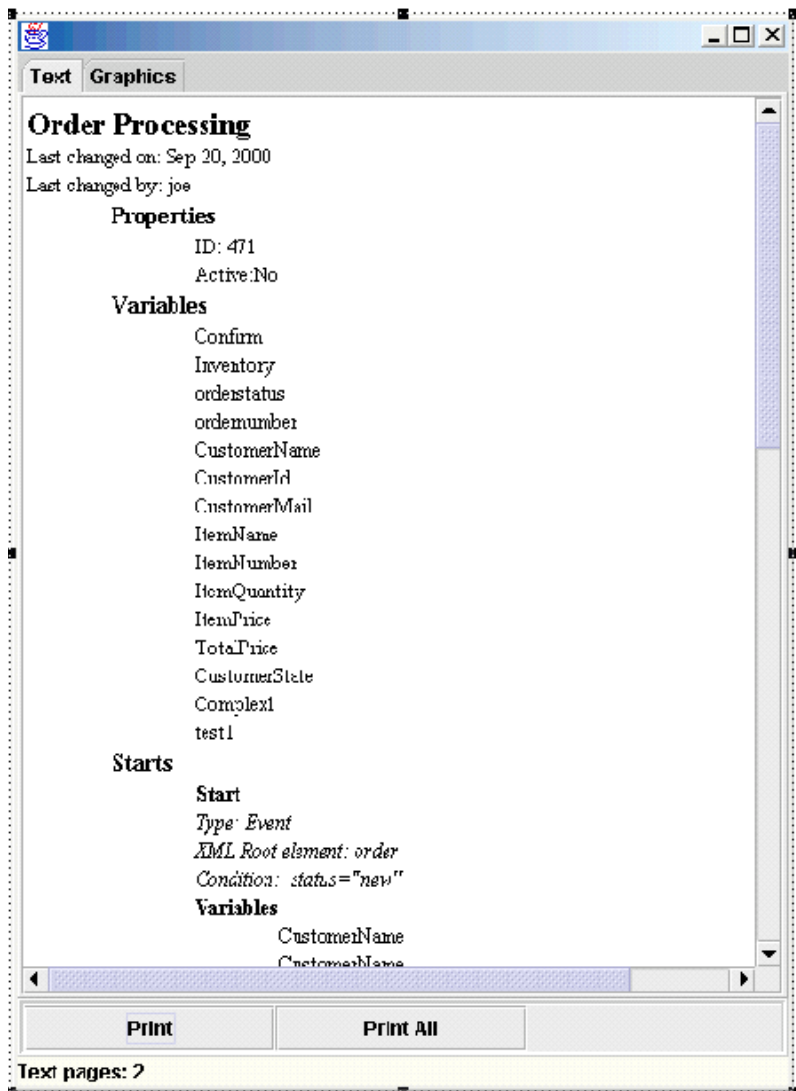
# Printing Workflow Template Definitions

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.

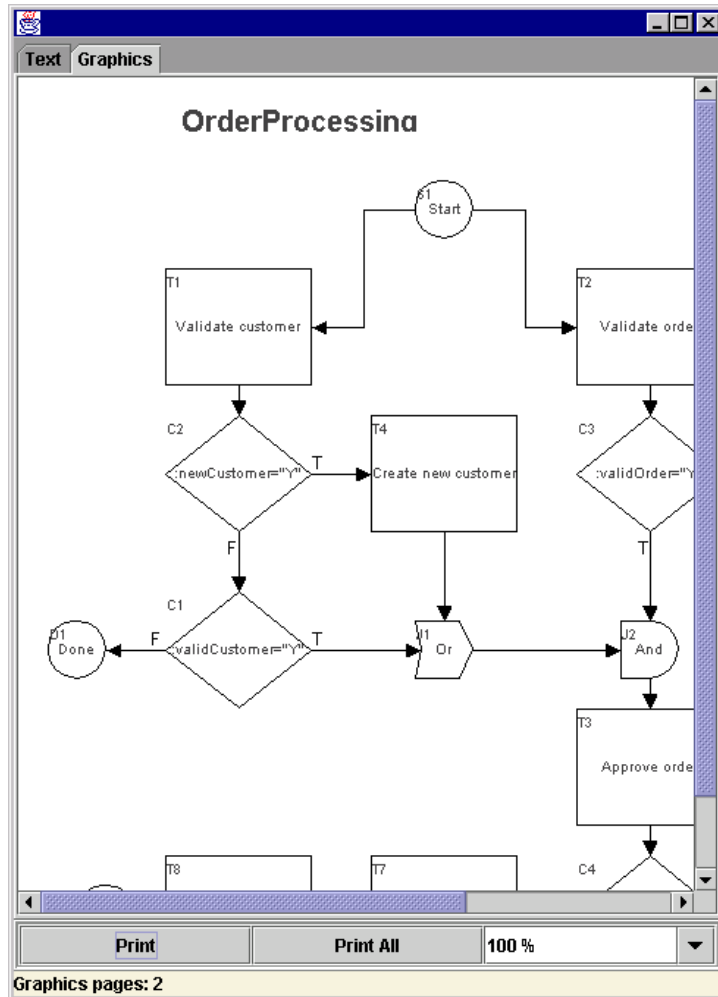
The Text tab contains information relevant to each workflow node, including action and node notes, as well as details on actions within tasks, events, decisions and done nodes, including subactions within actions.

Figure 3-10 Print Workflow Text



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

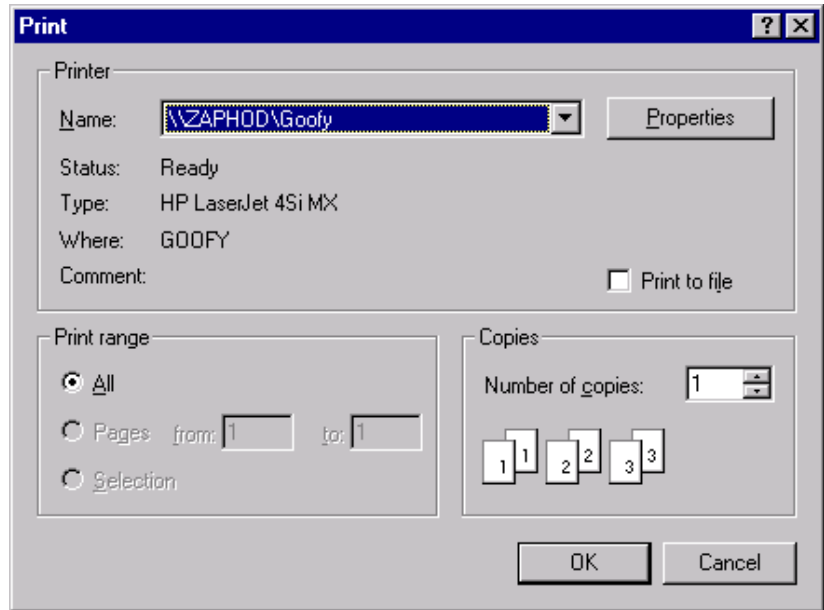
**Figure 3-11 Print Workflow Graphics**



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.

The Print dialog box is displayed. Select the printer name and click OK.

**Figure 3-12 Print Dialog Box**



# Copying Workflow Template Definitions

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, 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” 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**

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 copy actions saves you time.

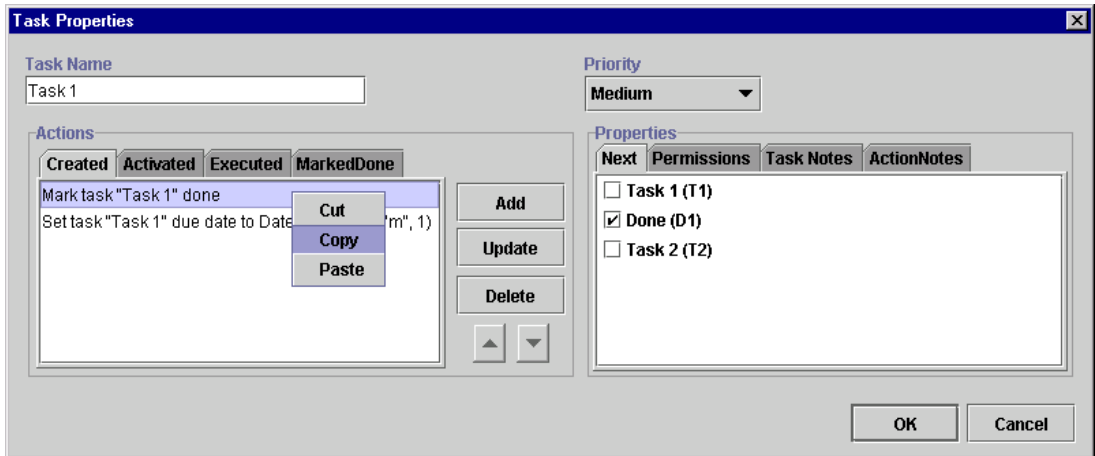
### **Copying Actions Within Workflow Nodes**

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 3-13 Copy Action**



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.

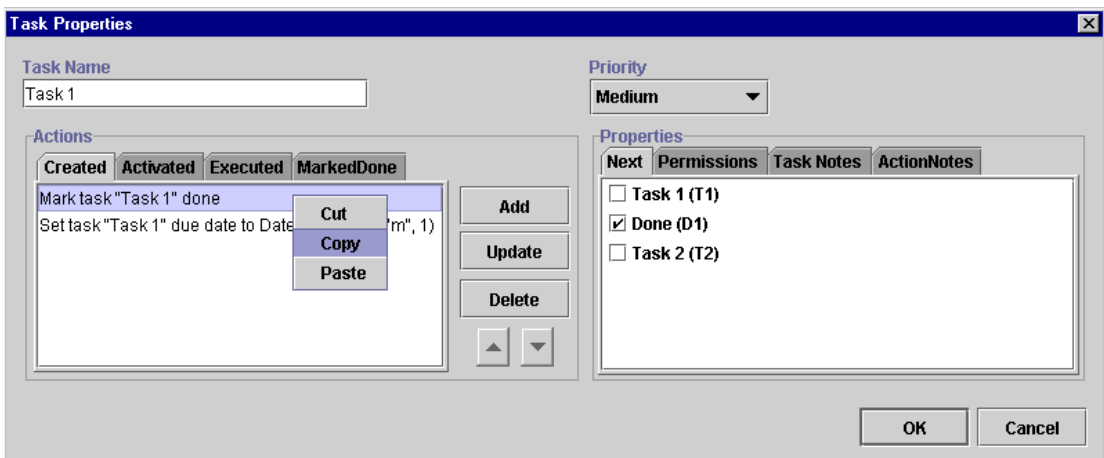
## Copying Actions Between Workflow Nodes

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 3-14 Copy Action**



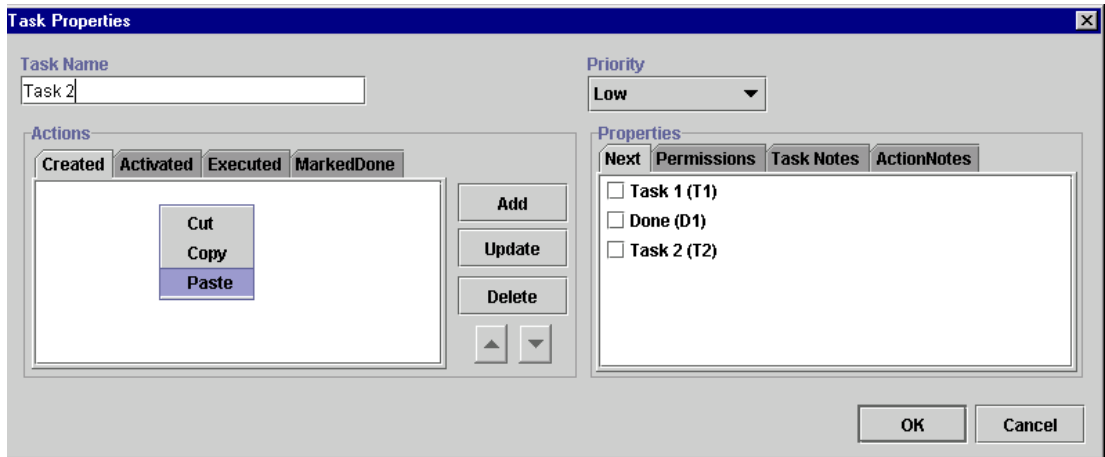
5. From the drawing area, right-click the node shape to which you will paste the defined action.

### 3 *Defining and Maintaining Workflows*

---

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 3-15 Paste Action**



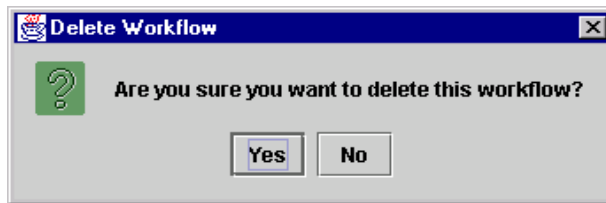
9. Depending on the pasted action, a dialog box appears that corresponds to the action.
10. Make changes, if needed, to the action definition, and Click OK. The action is pasted into the Actions portion of the window.

# 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 3-16 Delete Workflow Dialog Box**



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

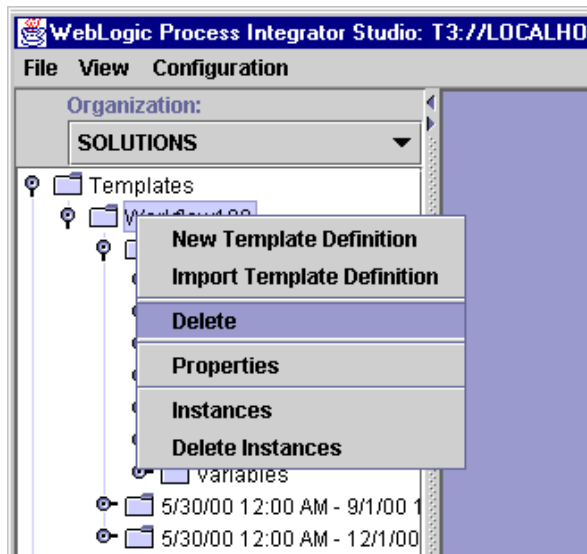
- All instances of that workflow template definition are deleted (including all tasks regardless of their current status).
- All history relating to that workflow template definition is deleted and is therefore no longer available in statistical reports.
- Workload reports that refer specifically to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

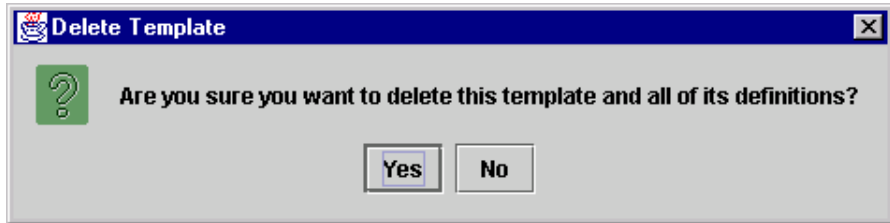
1. Right-click on the workflow template name in the folder tree.

**Figure 3-17 Delete Workflow Template**



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

**Figure 3-18 Delete Template Dialog Box**



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 can not 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.

## Exporting Workflow Template Definitions

You can export a workflow template definition from WebLogic Process Integrator to any drive mapped in your computer. You can, for example, move a workflow template definition from WebLogic Process Integrator Studio to another system. An exported workflow template definition is saved as an XML document.

To export a workflow template definition, right-click it in the folder tree, then choose Export from the pop-up menu. In the Save dialog box, select the drive where you will save the exported workflow template definition.

**Figure 3-19** Save Dialog Box



# Importing Workflow Template Definitions

You can import a workflow template definition (that was previously exported) into WebLogic Process Integrator to create a new workflow template definition.

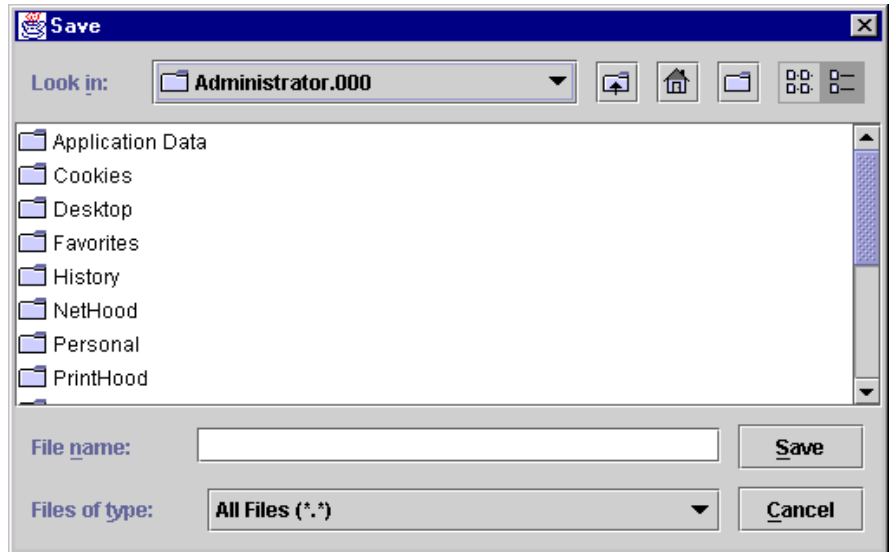
**Note:** Imported workflow template definitions are always marked as “inactive.” Before an imported workflow template definition can be instantiated, you must change the definition to “active” in the Template Definition dialog box. Refer to “Creating a New Workflow Template Definition.”

To import a workflow template definition, proceed as follows:

1. In the WebLogic Process Integrator folder tree, right-click the workflow template into which you will import the workflow template definition.
2. From the pop-up menu, select Import Template Definition to display the Save dialog box.
3. In the Save dialog box, select the current location of the exported workflow template definition file on your hard drive, and click Save.

**Note:** You cannot import a workflow template definition under a different name from that it contained when it was exported. Otherwise, you will receive a warning message.

**Figure 3-20** Save Dialog Box



4. After the file is read, 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 actions must be redefined when the actions are imported in a workflow.

# Displaying Workflow Instances

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.

To display the instances of a workflow template definition for monitoring purposes, right-click the workflow template folder and choose Instances from the pop-up menu. Refer to Chapter 8, “Monitoring Workflows,” for further details.



# 4 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
- Defining 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” in Chapter 1, “WebLogic Process Integrator Overview.”

# 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.

# Defining 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.

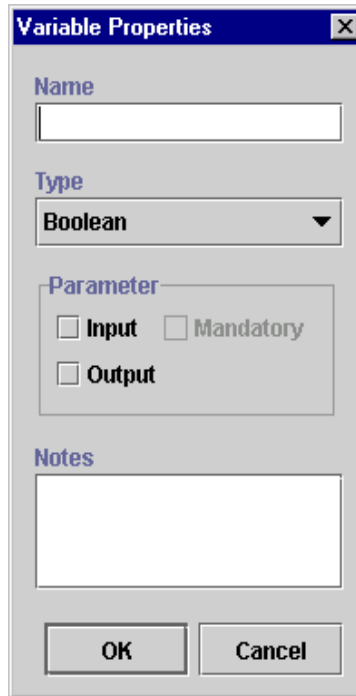
Each workflow template definition can have a set of variables associated with it. A variable is typically used to store application-specific information required by the workflow at run time. Variables are created and assigned values largely to “control” the logical path through a workflow instance; the same workflow template definition is instantiated multiple times and can be traversed in different ways if the flow contains decision nodes, which test workflow variable values to branch to either the next True or next False within the workflow.

## Defining Variables

To define a variable:

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

**Figure 4-1 Variable Properties Dialog Box**



The image shows a 'Variable Properties' dialog box with a blue title bar and a close button. It contains four sections: 'Name' with a text input field; 'Type' with a dropdown menu currently set to 'Boolean'; 'Parameter' with three checkboxes for 'Input', 'Mandatory', and 'Output'; and 'Notes' with a large text area. At the bottom are 'OK' and 'Cancel' buttons.

2. Complete the following fields:
  - Name — enter a meaningful name for the variable, such as ItemNumber, in the Name field.
  - Type — select the variable type: Boolean, Date, Double, Entity EJB, Integer, Java Object, Session EJB, String, or XML.
  - Parameter — input or output. For Input, choose whether the parameter is a mandatory one. The parameter is defined for workflow variables that have a called start node. These parameters are passed to (Input) and returned from (Output) a sub-workflow.
  - Notes — a free-form text field that can be used to make notes for the variable.

## Updating Variables

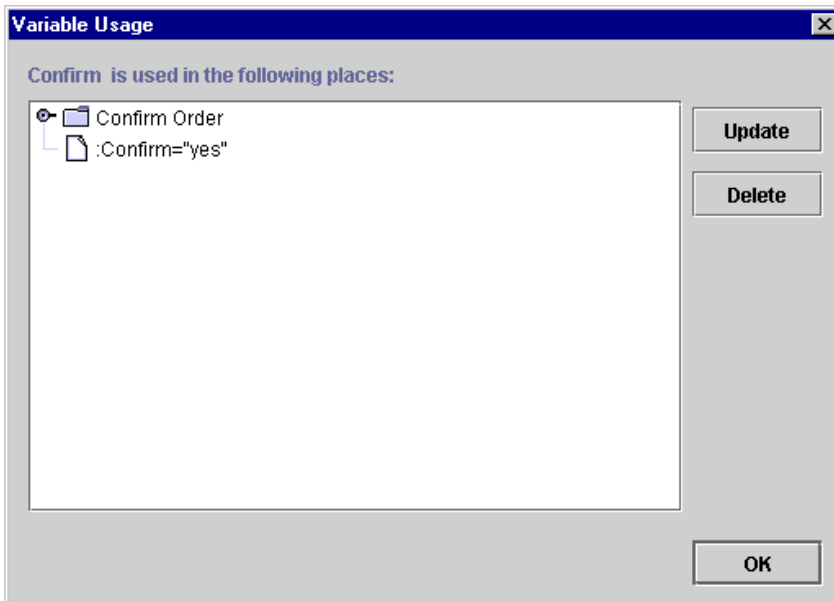
To update an existing variable:

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 4-2 Variable Usage Dialog Box**



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.

**Table 4-1 Variable Conversions**

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

**Note:** NS means Not Supported.

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 at simultaneously, if they have the same triggering event and are both 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 to the ability to specify more than one start is visual access to both streams of work within the same workflow and these streams of work can share the same variables. You have the flexibility to take a portion of a workflow and starting it in a different manner or at a different time within the workflow.

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

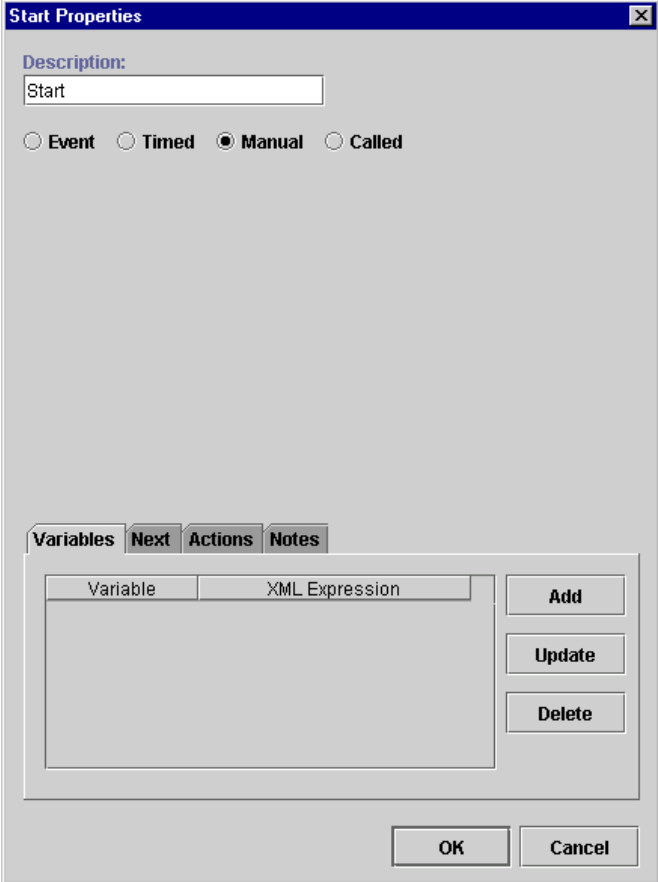
You specify the triggering properties of a workflow in the start node. If a workflow contains multiple starts, each start can have its own triggering method and start independently.

## 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.”
2. Double-click the Start shape to display the Start Properties dialog box.

**Figure 4-3 Start Properties Dialog Box**



The dialog box is titled "Start Properties" and has a close button (X) in the top right corner. It contains a "Description:" label followed by a text field containing the word "Start". Below this are four radio buttons: "Event", "Timed", "Manual" (which is selected), and "Called". At the bottom, there are four tabs: "Variables", "Next", "Actions", and "Notes". The "Variables" tab is active, showing a table with two columns: "Variable" and "XML Expression". To the right of the table are three buttons: "Add", "Update", and "Delete". At the bottom right of the dialog are "OK" and "Cancel" buttons.

| Variable | XML Expression |
|----------|----------------|
|----------|----------------|

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 — if you choose this triggering option, complete the procedure in “Defining a Workflow Event Trigger” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”
  - Timed — if chosen, the following fields appear. Define an exact date and time when the workflow will start.

Figure 4-4 Timed Option

**Start Properties**

**Description**  
Start

☐ Event ☒ **Timed** ☐ Manual ☐ Called

**Start Date Expression**  
 [A+BQ](#)

**Reschedule**  
 **Minutes** ▼

**Calendar**  
 ▼

**Start Organization Expression**  
 [A+BQ](#)

**Variables** | **Next** | **Actions** | **Notes**

| Variable     | XML Expression |
|--------------|----------------|
| CustomerName | customer.name  |

**Add**  
**Update**  
**Delete**

**OK** **Cancel**

**Start Date Expression** — enter the expression that defines the start date for the workflow; you must use a date function, such as Date or DateAdd, in this expression. Refer to “Using Expression Components” in Chapter 6, “Using Expressions and Conditions.”

**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 this to enter an expression that will be evaluated at run time and become the name of an organization. The expression can be either a string that specifies the organization (case sensitive) or an expression using information passed in the XML document.

**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 6, “Using Expressions and Conditions.”

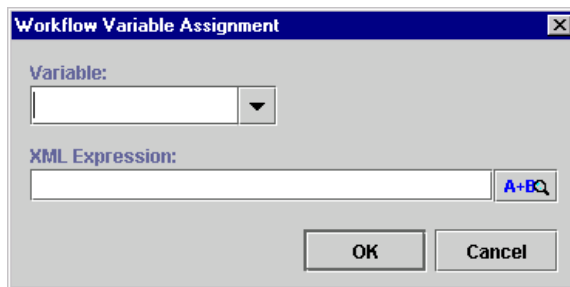
- Manual — if chosen, the workflow must be started manually by an end user from the WebLogic Process Integrator worklist. This type of start requires that the end user know the specific business conditions that determine when to start the workflow. You might create a manual start when you cannot create a trigger that captures all required conditions.
- Called — if chosen, the workflow is called by another workflow using the Start Workflow action. Refer to “Start Workflow” in Chapter 5, “Using Actions,” for more information on this action.

5. Complete or view the fields on the tabs:

- Variables — assign workflow variables to be initialized at the start of your workflow. 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 following dialog box where each variable can be assigned an XML Expression. The variables in the pull-down list are those defined in the Variables Properties dialog box. (See “Defining Variables.”)

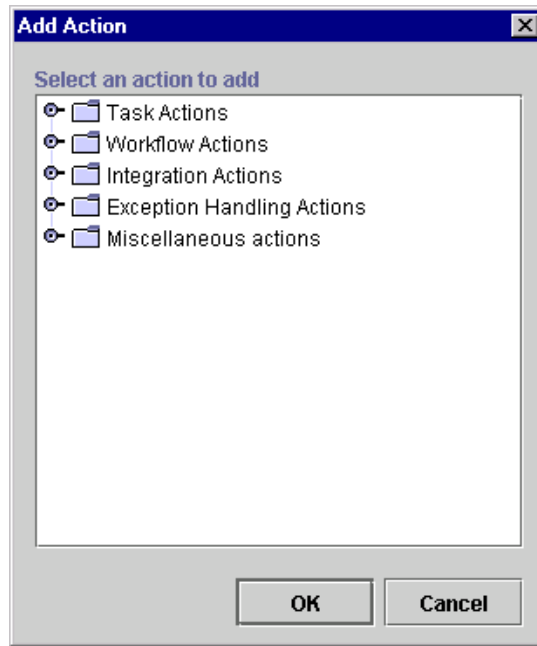
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “Using Expressions and Conditions.”

**Figure 4-5 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. Refer to Chapter 5, “Using Actions,” for further details on selecting actions.

**Figure 4-6 Add Action Dialog Box**



- Notes — a free form text field for recording notes about the start.

# Defining Dones

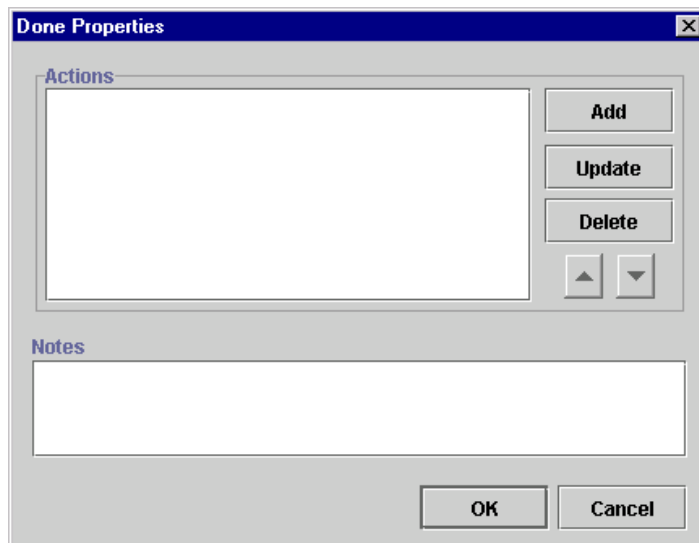
A done shape represents the point within the workflow where the process completes. Once a done node is reached in the workflow, the running instance of the workflow is marked done, regardless of whether all the done nodes have been reached or not.

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.”
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 4-7 Done Properties Dialog Box**



3. Complete or view the fields:

- **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 5, “Using Actions,” for further details on selecting actions.
- **Notes** — a free form text field for recording notes about the done.

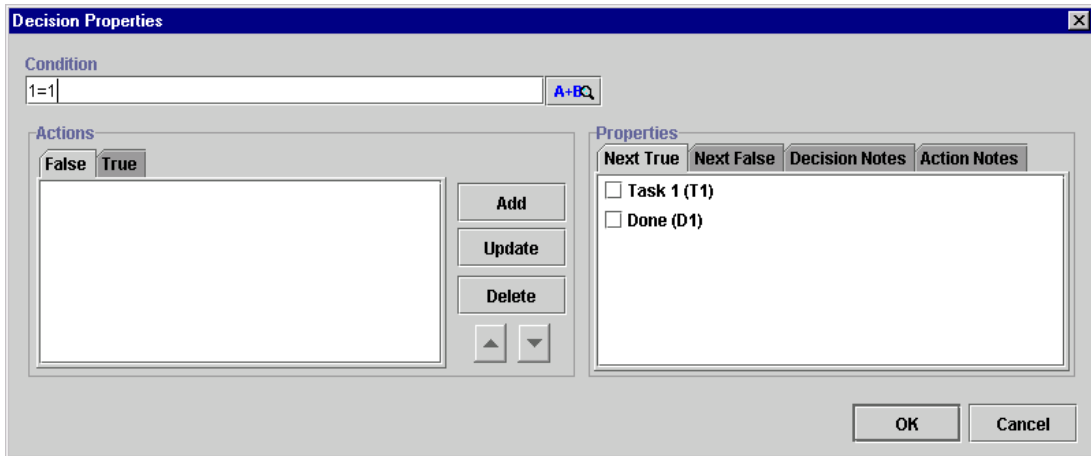
# Defining Decisions

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.”
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 4-8 Decision Properties Dialog Box



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 6, “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 5, “Using 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.

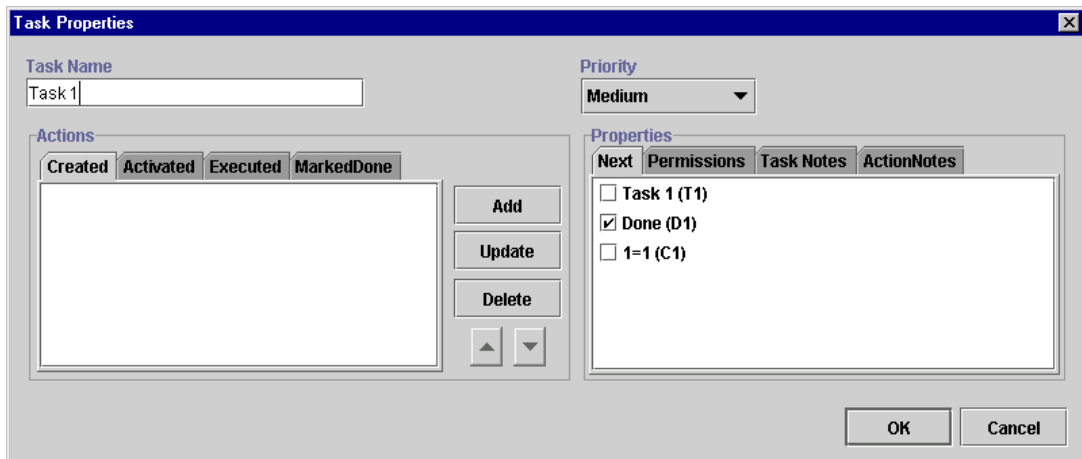
# Defining Tasks

A workflow contains a series of tasks, each of which represents a discrete activity to be performed automatically by the WebLogic Process Integrator server or by a workflow user. The task behavior is determined by various task properties and by the task actions that define the processing to be performed as certain task events occur.

To define a task:

1. Display or add a task as described in “Adding Workflow Components to a Workflow Diagram.”
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 4-9 Task Properties Dialog Box**



3. Complete the following fields:
  - **Task Name** — enter a meaningful name, such as ConfirmOrder. This value is used primarily in the worklist, but it is visible from the various 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 is displayed in the worklist and can be used to sort the worklist by priority. The initial default value is Medium.
- **Actions** — tabbed display of four separate lists of actions for the task, corresponding to the task events. See Chapter 5, “Using Actions,” for a description of how to specify actions. A task responds to an event at run time by executing the associated actions; actions defined within these task events are guaranteed to occur in sequence. The four task events are as follows:

**Created** — the workflow is started and all tasks in the workflow are created.

**Activated** — the workflow diagram indicates a task transition has occurred, usually as a result of another task being marked as done or a decision being evaluated. The task is available to be performed.

**Executed** — an event has occurred that causes this task to be executed. This usually occurs when a user selects a task in the worklist and executes it, often by double-clicking the task, or when an Execute Task action is performed for that task.

**Marked Done** — the task has been completed. This event occurs usually in response to the execution of a Mark Task as Done action.

To display the list of actions for a different event, select the appropriate tab.

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** — specifies how a task instance can be modified or processed once the workflow is started. Select a permission to activate it. The permissions can be selected in any combination. Five different permissions are maintained 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

A workflow event shape represents a notification node. The workflow waits for an XML message to trigger the event. Upon that trigger, actions defined within the event can be executed and/or workflow variables can be set.

To define an event:

1. Display or add a task as described in “Adding Workflow Components to a Workflow Diagram.”
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 4-10 Event Properties**

**Event Properties**

**Description:**  
Event

**Root Element:**

**Key Value Expression:**  
A+BQ

**Condition:**  
A+BQ

**Variables** **Next** **Actions** **Notes**

| Variable | XML Expression |
|----------|----------------|
|          |                |

Add  
Update  
Delete

OK Cancel

3. To define the event, follow the procedure in “Using the XML Notification: The WebLogic Process Integrator Event” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”

## 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. An AND join will reset itself immediately after it is activated, in order to allow a looping process.

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.

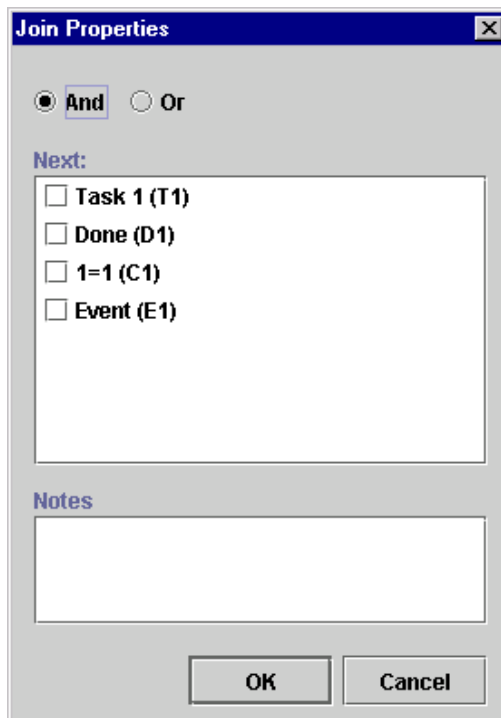
Use AND and OR joins to link task, decision, and event shapes and thus control the transition coming to the join and flowing from it.

# Defining a Join

To define a join:

1. Display or add a join as described in “Adding Workflow Components to a Workflow Diagram.”
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 4-11 Join Properties Dialog Box**



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.



# 5 Using Actions

The following sections introduce and explain how to use 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 WebLogic Process Integrator. Numerous actions are shipped with 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 in that they define the actual behavior of the workflow.

## Action Categories

Actions are organized into the following categories:

- Task Actions (See “Using Task Actions.”)
- Workflow Actions (See “Using Workflow Actions.”)
- Integration Actions (See “Using Integration Actions.”)
- Error Handling Actions (See “Using Exception Handling Actions” in Chapter 7, “Handling Workflow Exceptions.”)
- Miscellaneous Actions (See “Using Miscellaneous Actions.”)

## Action Placement and Subactions

You can place actions in a number of places in a workflow. The most common place is in a Task. Actions can also be placed in decisions, starts, events, and dones. The action is executed when the node is activated. 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.

- Start Workflow — add actions to be performed upon starting another workflow.
- Set Task Due Date — add overdue actions; set a due time on a task so that when the due time passes, other actions will be performed.

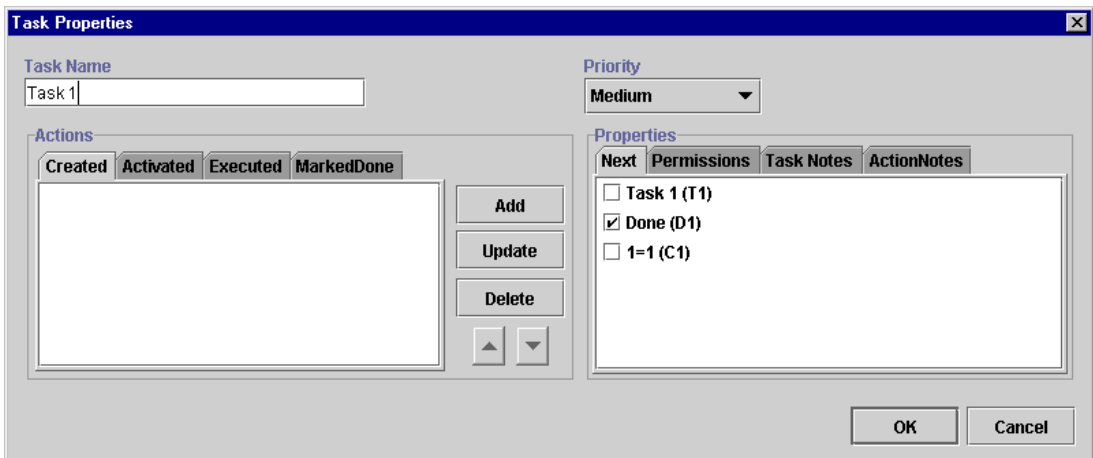
## Adding or Updating an Action

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:

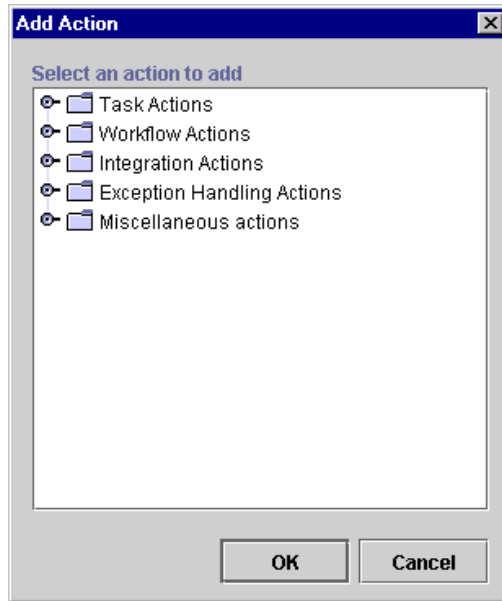
1. Double-click the Task, Event, Decision, Start, or Done for which you want to add or update an action. The Properties dialog box appears.

**Figure 5-1 Maintaining Actions**



2. In the properties dialog box, select (activate) the appropriate action tab and click Add or Update to display the Add Action or Update Action dialog box.

**Figure 5-2 Add Action Dialog Box**



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.

For details on completing the information in specific action dialog boxes, see the sections “Using Task Actions,” “Using Workflow Actions,” “Using Integration Actions,” “Using Exception Handling Actions,” and “Using Miscellaneous Actions.”

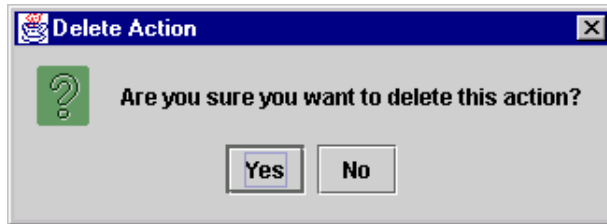
The action is updated or added to the Activated actions tab in the Properties dialog box.

# 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 the Delete button. A confirmation dialog box is displayed to prevent accidental deletions.

**Figure 5-3 Delete Action Dialog Box**



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 5-4 Up and Down Arrows**



# Using Task Actions

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

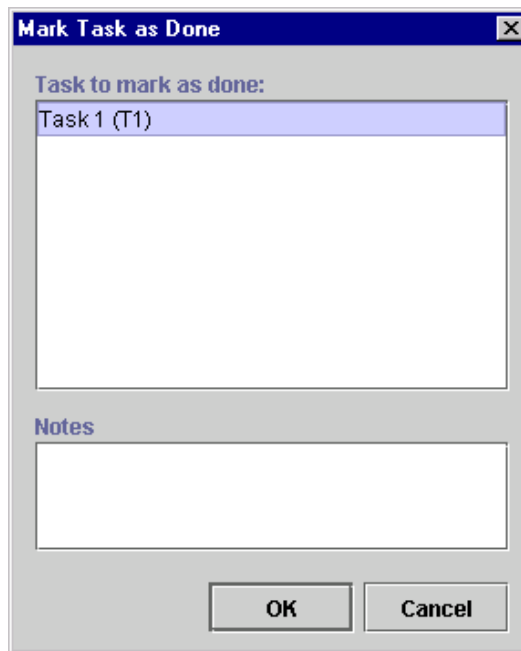
- Mark Task as Done
- Unmark Task Done
- Set Task Due Date
- Assign Task to User
- Assign Task to Role
- Assign Task Using Routing Table
- Unassign Task
- Set Task Comment
- Set Task Priority
- Execute Task

## Mark Task as Done

Use this action to set the completion date of a specified task to the current time and mark it done. Executing this action performs all actions found in the MarkedDone tab of the Task Properties dialog box.

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 5-5 Mark Task as Done Dialog Box**



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.

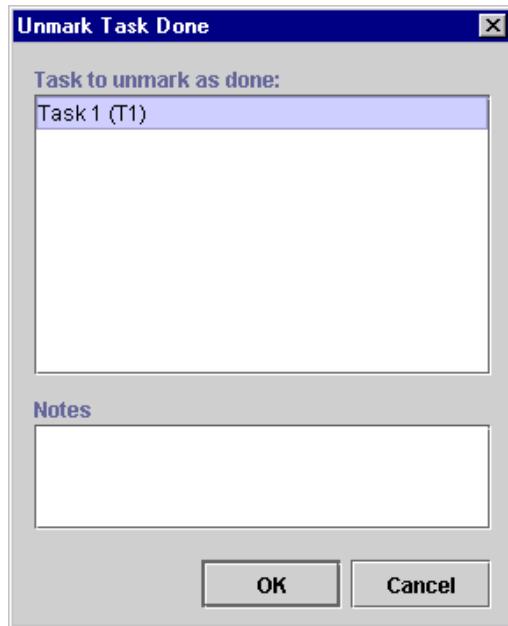
## Unmark Task Done

This action clears the completion date of a specified task and marks the task as not done.

Use this action when a condition arises that causes the task to no longer be in a successfully completed state. A task that is assigned this action becomes Active on the Worklist and can be executed again.

Note that a task that is unmarked done will not have its activated event triggered again. To trigger the activated event, there must be a transition to that task.

**Figure 5-6 Unmark Task Done Dialog Box**



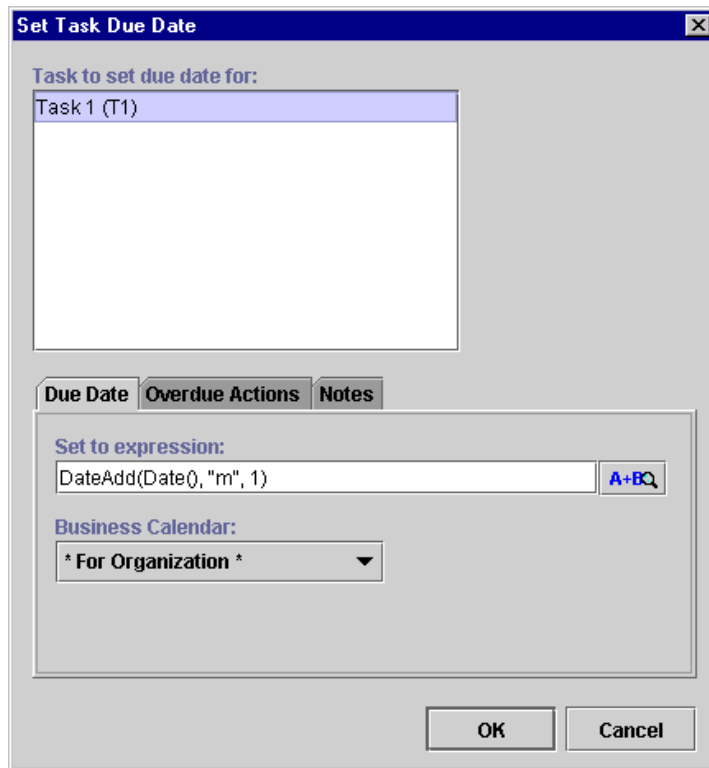
View or enter information in the Unmark Task Done dialog box.

- 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 to an interval (expressed in minutes, hours, days, weeks, or months) from when the action is executed, or based on an expression. Subactions can also be specified to be performed when the task becomes overdue.

**Figure 5-7 Set Task Due Date Dialog Box**



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 — the task due date can be set to occur from a point in time after action execution, or based on the evaluation of an expression.
  - Set to expression — set the due date based on an expression.

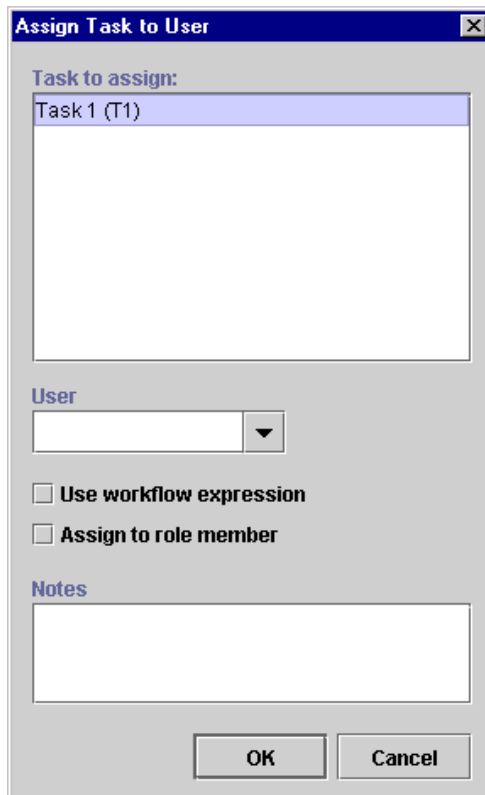
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “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 5-8 Assign Task to User Dialog Box**



The dialog box is titled "Assign Task to User" and contains the following elements:

- Task to assign:** A list box containing "Task 1 (T1)".
- User:** A text field with a dropdown arrow.
- ☐ **Use workflow expression**
- ☐ **Assign to role member**
- Notes:** A text area.
- OK** and **Cancel** buttons.

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 6, “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 tasks 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 5-9 Assign Task to Role Dialog Box**



The dialog box is titled "Assign Task to Role" and contains the following elements:

- Task to assign:** A list box containing "Task 1 (T1)".
- Role:** A text input field with a search icon (A+B) to its right.
- ☒ **Use workflow expression**
- Notes:** A large text area for additional information.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

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.

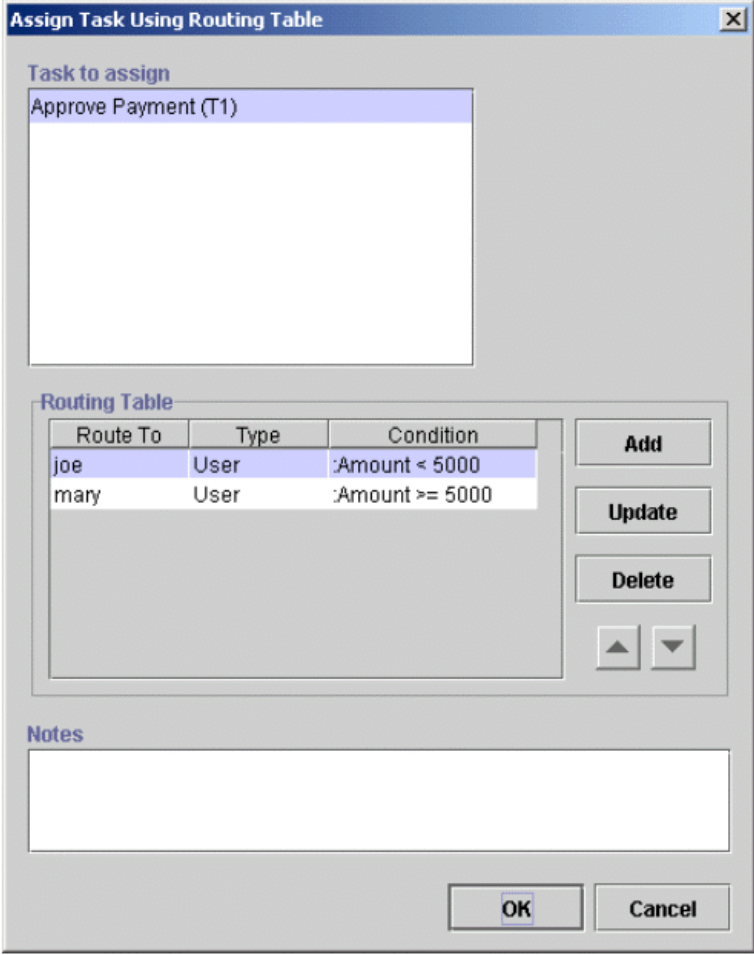
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “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 5-10 Assign Task Using Routing Table Dialog Box**

The dialog box is titled "Assign Task Using Routing Table". It contains three main sections: "Task to assign", "Routing Table", and "Notes".

**Task to assign**

A list box containing the task "Approve Payment (T1)".

**Routing Table**

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

Buttons: Add, Update, Delete, and two arrow buttons (up and down).

**Notes**

A text area for entering notes.

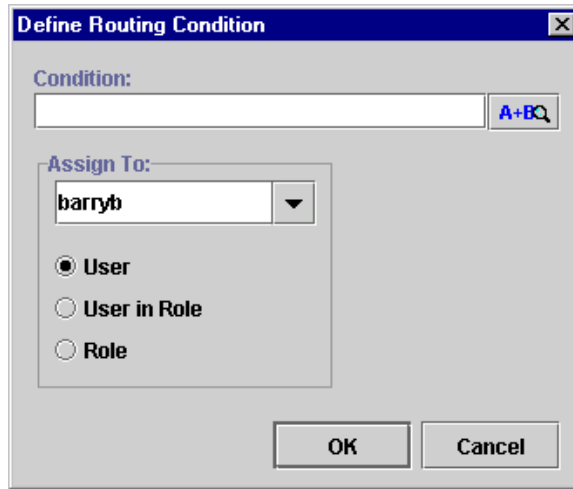
Buttons: OK, Cancel.

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 5-11 Define Routing Condition Dialog Box**



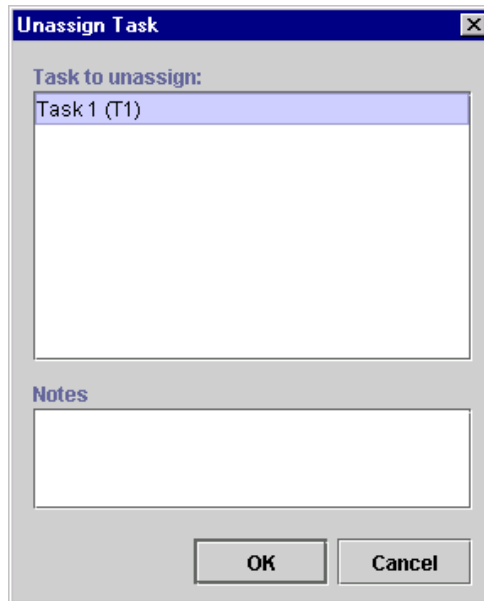
The Define Routing Condition dialog box contains the following:

- **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 6, “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 5-12 Unassign Task Dialog Box**



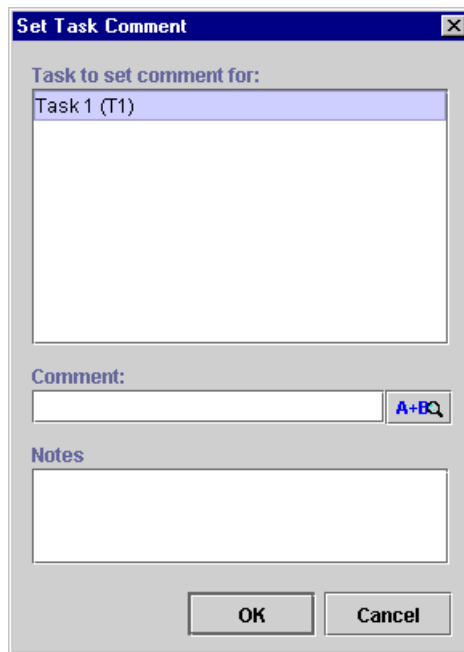
View or enter information in the Unassign Task dialog box.

- **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 5-13 Set Task Comment Dialog Box**



The dialog box is titled "Set Task Comment" and contains the following elements:

- Task to set comment for:** A list box containing "Task 1 (T1)".
- Comment:** A text input field with a blue "A+B" icon to its right.
- Notes:** A larger text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

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.

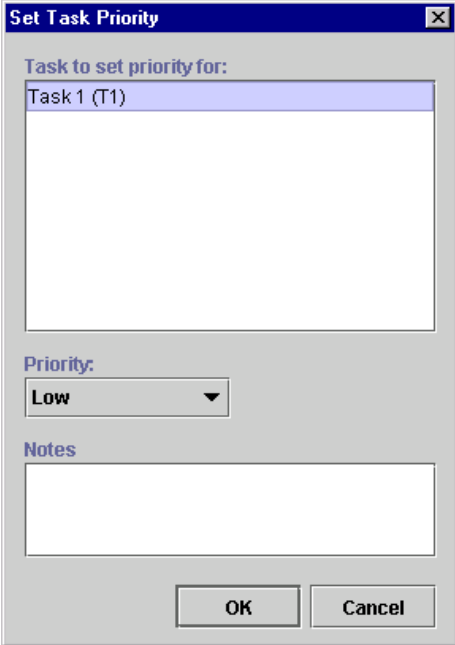
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “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 5-14 Set Task Priority Dialog Box**

The image shows a 'Set Task Priority' dialog box. It has a title bar with the text 'Set Task Priority' and a close button. Inside the dialog, there is a section labeled 'Task to set priority for:' followed by a list box containing 'Task 1 (T1)'. Below this is a section labeled 'Priority:' with a dropdown menu currently set to 'Low'. At the bottom is a section labeled 'Notes' with a large text area. At the very bottom are 'OK' and 'Cancel' buttons.

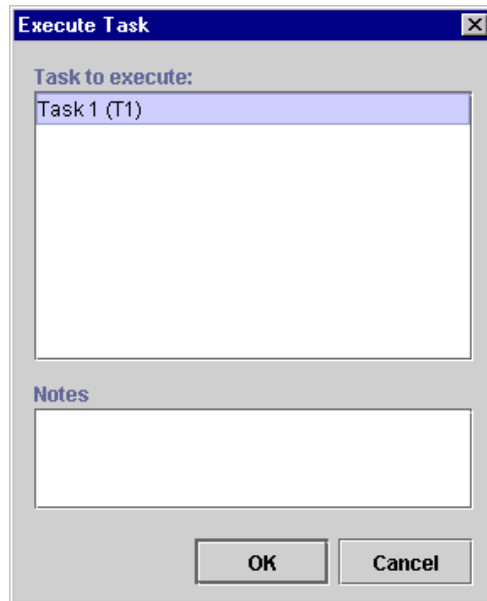
View or enter information in the Set Task Priority dialog box.

- 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

Execute a selected task.

**Figure 5-15** 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

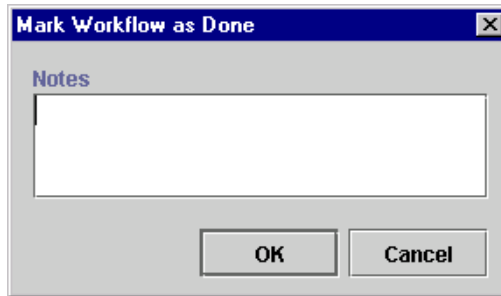
This section defines and describes how to use the following workflow actions:

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

## Mark Workflow as Done

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

**Figure 5-16 Mark Workflow as Done**

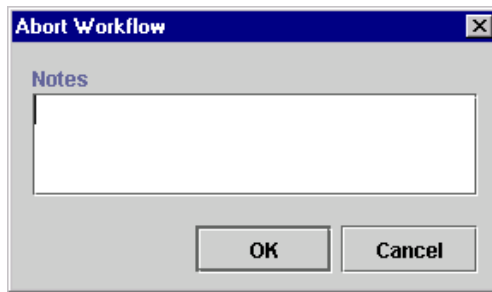


Enter notes about the action in the free-form text box, and click OK.

## 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 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 5-17** Abort Workflow Dialog Box



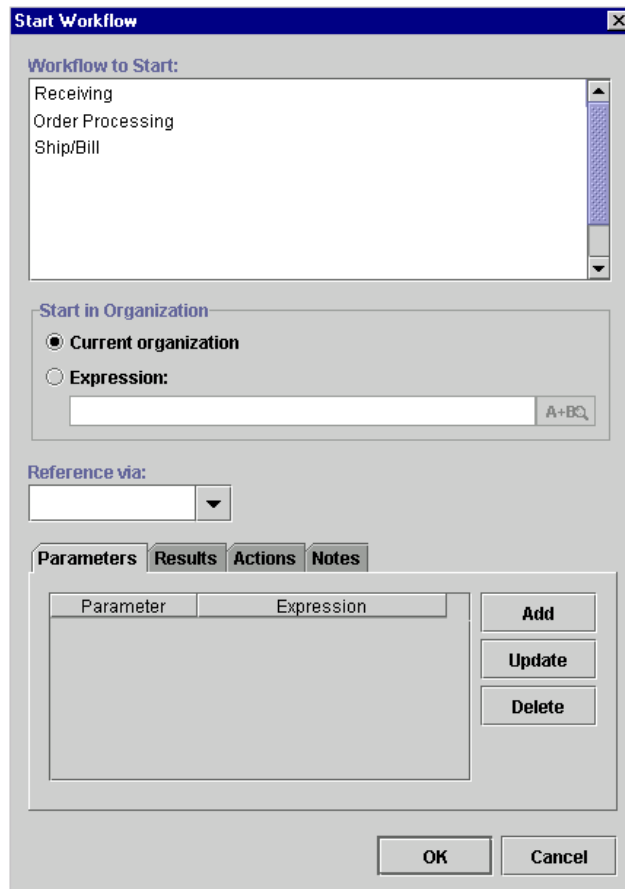
View or enter information in the Abort Workflow dialog box.

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

## Start Workflow

This action supports subworkflows; variable values can be transferred from a parent workflow to a child workflow. You trigger a subworkflow (or child workflow) from a parent workflow. The started workflow can be tracked with a variable reference, and when it completes, you can perform an optional set of subactions.

**Figure 5-18 Start Workflow Dialog Box**



The **Start Workflow** dialog box is used to configure a workflow action. It features a list of workflows to start, options for the starting organization, a reference variable, and a table for parameters.

**Workflow to Start:**

- Receiving
- Order Processing
- Ship/Bill

**Start in Organization:**

- ☒ Current organization
- ☐ Expression:  A+BQ

**Reference via:**

▼

**Parameters** | Results | Actions | Notes

| Parameter | Expression |
|-----------|------------|
|-----------|------------|

**Add**  
**Update**  
**Delete**

**OK** **Cancel**

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 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 6, “Using Expressions and Conditions.”

- **Reference via** — list of workflow variables available for the current workflow. By selecting a variable, you assign the workflow ID of the started workflow to that variable. This variable can be used in expressions from within the calling workflow in order to retrieve variables from the newly started workflow. For example, `NewWF.Name` retrieves the value of the Name variable from the new workflow referenced via the `NewWF` variable.

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

- **Parameters** — list of variables within the target workflow, allowing the initial values to be set from an explicitly defined variable in the calling workflow.
- **Results** — displays variables for the selected workflow that have been declared as “output” at definition time. You can save values from these variable in the variables defined for this workflow.

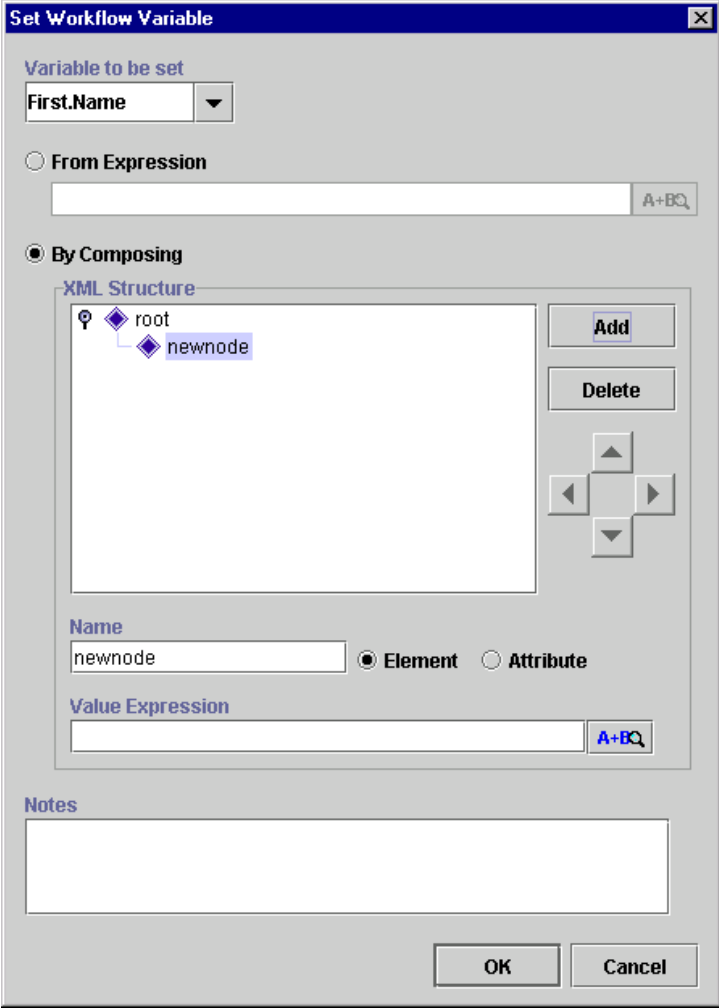
- Actions — specify optional subactions to be performed when the called workflow is completed. It can be used to mark the current task as done.
- Notes — a free-form text box for notes about the action.

## **Set Workflow Variable**

This action allows you to assign a value to a workflow variable, which can be an expression evaluated at run time or a literal. This action evaluates the expression or literal and assigns the result to a workflow variable when the action is executed.

In addition, this action allows you to compose an XML document and store the content in an XML type variable. XML type variables can be sent as internal or external XML messages using the action Post XML Event. For more information on this action, refer to “Using XML Actions” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”

**Figure 5-19 Set Variable Dialog Box**



The dialog box is titled "Set Workflow Variable" and contains the following sections:

- Variable to be set:** A dropdown menu showing "First.Name".
- From Expression:** An unselected radio button with an empty text field and an "A+BQ" button.
- By Composing:** A selected radio button.
  - XML Structure:** A tree view showing a "root" node with a child "newnode" node. To the right are "Add" and "Delete" buttons, and a set of four directional arrow buttons.
  - Name:** A text field containing "newnode".
  - Element/Attribute:** Two radio buttons; "Element" is selected.
  - Value Expression:** An empty text field with an "A+BQ" button.
- Notes:** A large empty text area.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

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.
- From Expression — define the variable by constructing a expression or 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 6, “Using Expressions and Conditions.”

- By Composing — set the XML or string type variable to store the XML document by creating an XML structure containing elements and attributes, as follows:
  - a. Select either the Element or Attribute radio button, depending on what you are adding to the XML document structure.
  - b. In the Name field, enter the name of the element or attribute, and click Add to add it to the XML document.
  - c. Use the directional arrows to the right to move the element or attribute forward or backwards, up or down, depending on where you need to position it in the structure. Moving an element or attribute forward will make it a subelement of the previous main node.

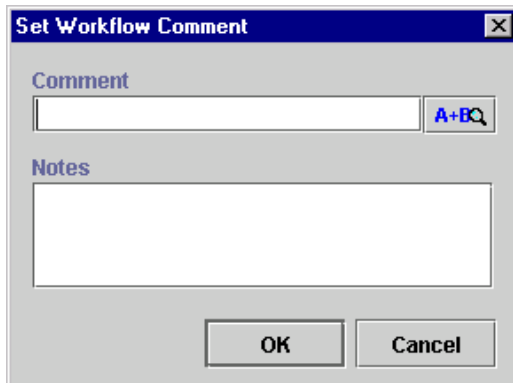
**Note:** An XML document that is stored in a string variable will first be converted to a string.

- Value Expression — use the Value Expression field to define the value of an element or attribute. The value can be either a string, typed in quotes, or an expression containing workflow variables and functions.
- 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 5-20 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 6, “Using Expressions and Conditions.”

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

# Using Integration Actions

This section defines and describes how to use the following integration actions:

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

## Post XML Event

For a description of this action and how to use it, see “Post XML Event” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”

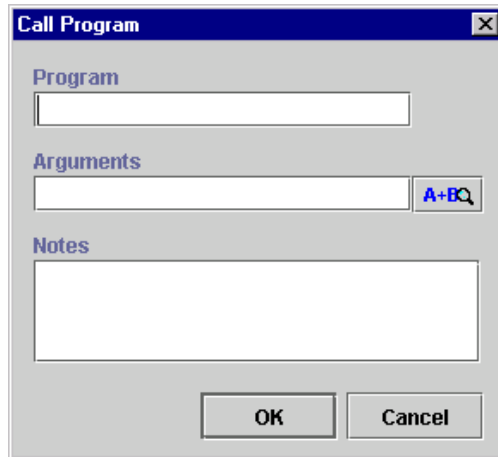
## Send XML to Client

For a description of this action and how to use it, see “Send XML to Client” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”

## Call Program

Use this action to call an executable program on the workflow server.

**Figure 5-21** 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 6, “Using Expressions and Conditions.”

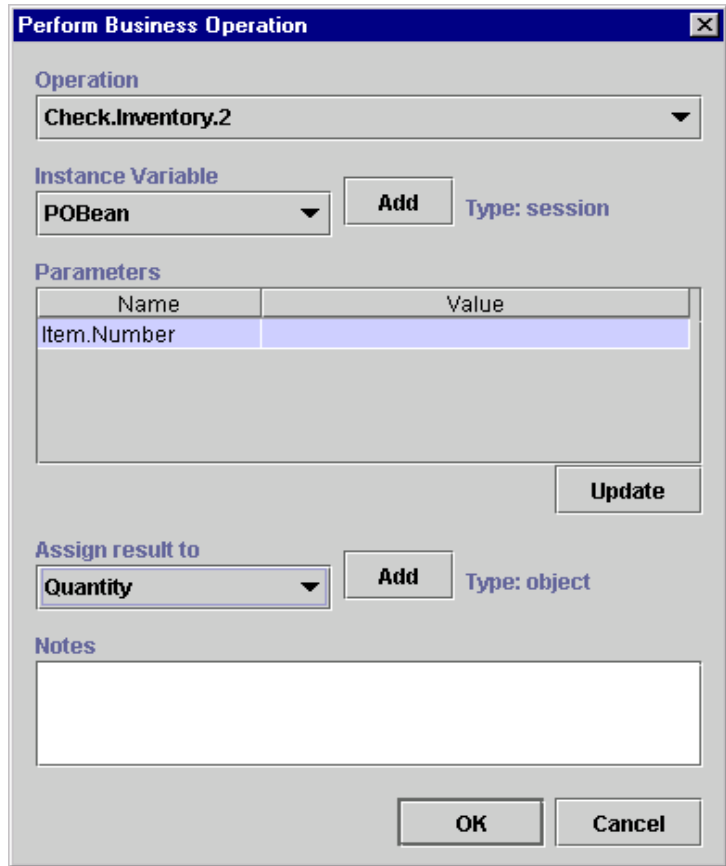
- **Notes** — a free-form text box for notes about the action.

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**

Allows the invocation of a predefined business operation on an Enterprise JavaBean (EJB) or Java class. The results of such an invocation can be assigned to workflow variables. The Perform Business Operation action is linked to a WebLogic Process Integrator function called Business Operations (accessed through the Configuration menu). See “Understanding Business Operations” in Chapter 2, “Administering Data within WebLogic Process Integrator,” for more information.

**Figure 5-22 Perform Business Operation Dialog Box**



The dialog box is titled "Perform Business Operation" and contains the following sections:

- Operation:** A dropdown menu showing "Check.Inventory.2".
- Instance Variable:** A dropdown menu showing "POBean" with an "Add" button and the text "Type: session".
- Parameters:** A table with two columns: "Name" and "Value". The first row has "Item.Number" in the "Name" column and is highlighted. Below the table is an "Update" button.
- Assign result to:** A dropdown menu showing "Quantity" with an "Add" button and the text "Type: object".
- Notes:** A large empty text area.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

| Name        | Value |
|-------------|-------|
| Item.Number |       |

View or enter information in the Perform Business Operation dialog box.

- **Operation** — select the predefined business operation to be invoked. Business operations in WebLogic Process Integrator are Enterprise JavaBean (EJB) or Java based. These operations are defined using the Define Business Operation dialog box, discussed in the section entitled “Understanding Business Operations” in Chapter 2, “Administering Data within WebLogic Process Integrator.”
- **Instance Variable** — select the instance variable, which is the name of the variable containing the Java object or EJB on which WebLogic Process Integrator should invoke the business operation. The Instance Variable option is activated if the business operation represents one of the following:
  - An EJB remote method
  - A Java class non-static, non-constructor method

In the above cases, the instance variable drop-down list displays defined Java object, Session EJB, and Entity EJB type variables.

- **Add** — create a new instance variable by accessing the Variable Properties dialog box. (See “Defining Variables” in Chapter 4, “Working with Workflow Components.”)
- **Parameters** — certain business operations require parameter input. You must enter values for the parameters; the business operation is called based on these actual values. To add a parameter value to a listed parameter, highlight it, click the Update button, and use the invoked Expression Builder dialog box to define this value. (See “Using the Expression Builder” in Chapter 6, “Using Expressions and Conditions.”)
- **Assign Result To** — assign the business operation result (for example, a calculated average) to a workflow, selected from the drop-down list. 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 to assign the result of the call to by accessing the Variable Properties dialog box. (See “Defining Variables” in Chapter 4, “Working with Workflow Components.”)
- **Notes** — a free-form text box for notes about the action.

## Using Exception Handling Actions

For instructions on using exception handling actions, see “Using Exception Handling Actions” in Chapter 7, “Handling Workflow Exceptions.”

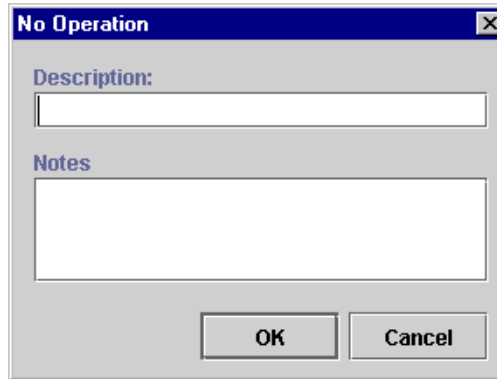
## Using Miscellaneous Actions

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 5-23 No Operation Dialog Box**

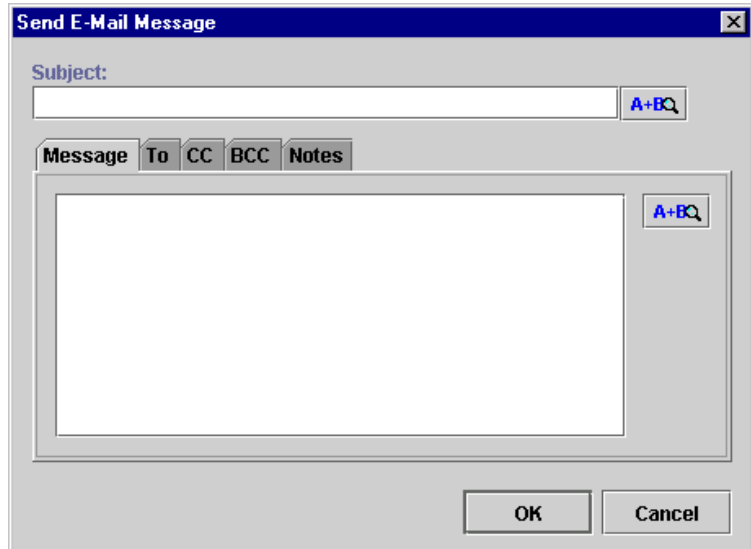
View or enter information in the No Operation dialog box.

- **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 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 `weblogic.properties` file and EJB deployment. Refer to Steps 2 and 3 in “Customizing and Starting the Server” in Chapter 4, “Configuring and Starting WebLogic Process Integrator,” of the *BEA WebLogic Process Integrator Installation and Configuration Guide*.

**Figure 5-24 Send E-Mail Message Dialog Box**

View or enter information in the Send E-Mail Message dialog box.

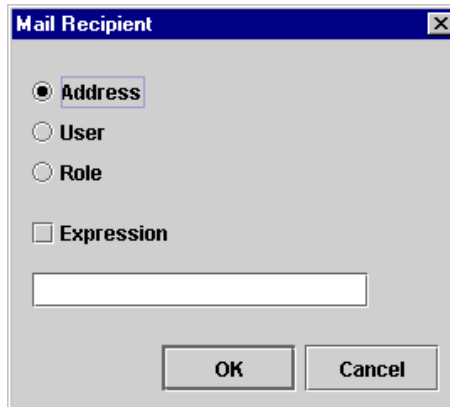
- **Subject** — a valid workflow expression or a literal entered in double quotes that will be evaluated at run time for the subject text of the e-mail.
- **Message** — a valid workflow expression or a literal entered in double quotes that will be evaluated at run time for the message text of the e-mail.

**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 6, “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 5-25 Mail Recipient Dialog Box



- 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.

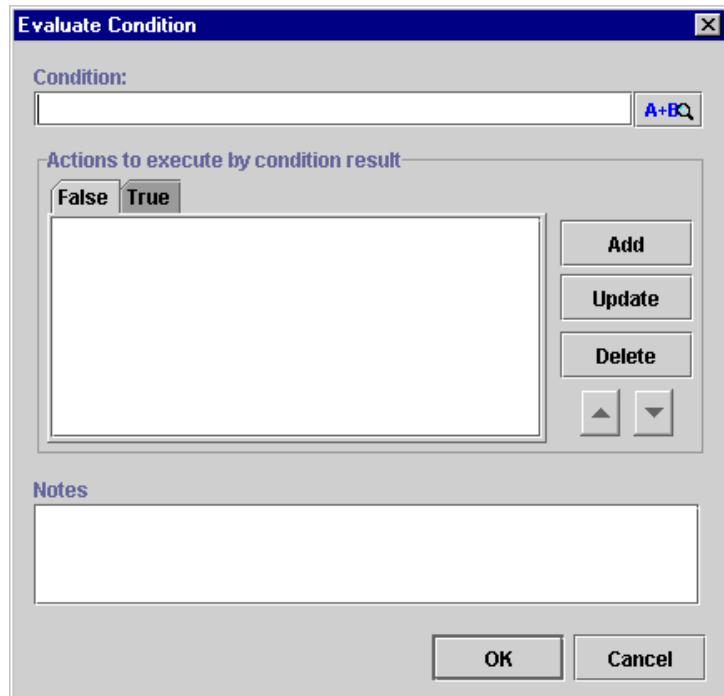
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “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 5-26 Evaluate Condition Dialog Box**



The dialog box is titled "Evaluate Condition" and contains the following elements:

- Condition:** A text input field with a search icon (A+BQ) to its right.
- Actions to execute by condition result:** A section with two tabs, "False" and "True". Below the tabs is a large empty rectangular area for listing actions. To the right of this area are three buttons: "Add", "Update", and "Delete", followed by two arrow buttons (up and down).
- Notes:** A text input field at the bottom of the dialog.
- Buttons:** "OK" and "Cancel" buttons at the bottom right of the dialog.

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 6, “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 5-27 Timed Event: Trigger Event Tab**

The screenshot shows a dialog box titled "Timed Event" with four tabs: "Trigger Event", "Execution Schedule", "Actions when triggered", and "Notes". The "Trigger Event" tab is selected. It contains two radio buttons: "From action execution:" (selected) and "To Expression:". Below "From action execution:" is a text input field and a dropdown menu set to "Minutes". Below "To Expression:" is a text input field containing the expression "DateAdd(Date(), \"m\", 1)" and a button labeled "A+B". Below these is a section labeled "Business Calendar:" with a dropdown menu set to "\* For Organization \*". At the bottom right are "OK" and "Cancel" buttons.

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.

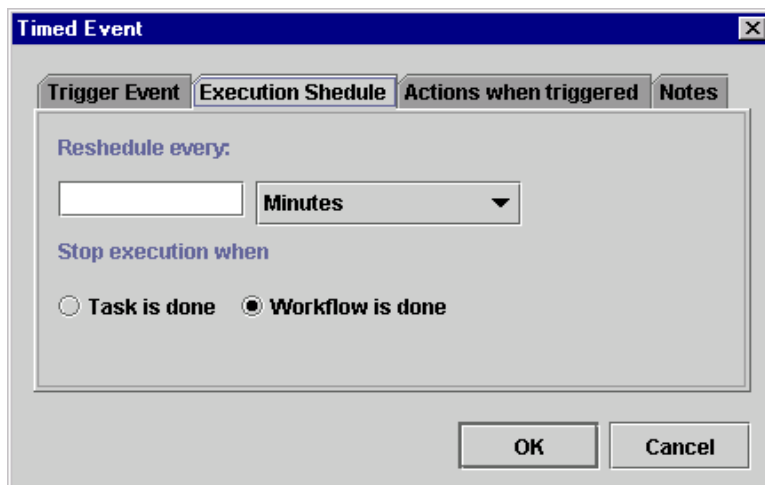
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “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 5-28 Timed Event: Execution Schedule Tab**

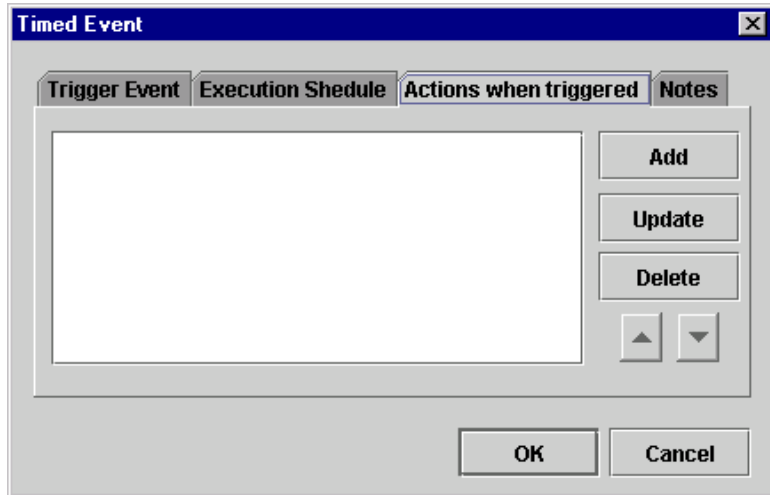


View or enter information in the Execution Schedule tab.

- **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.

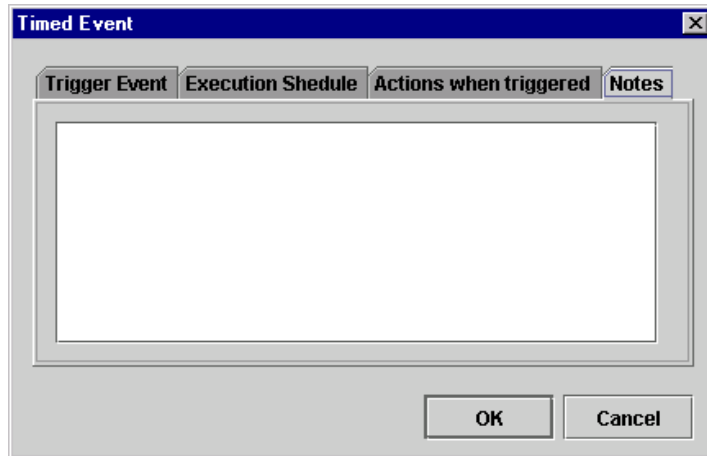
**Figure 5-29 Timed Event: Actions When Triggered Tab**



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 5-30 Timed Event: Notes Tab**

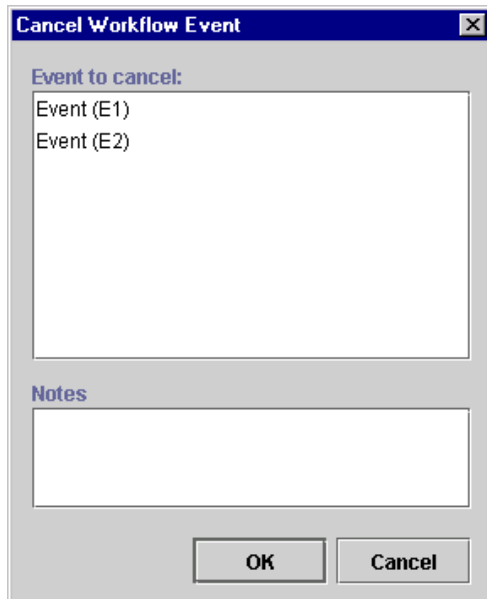


Use the Notes tab of the Timed Event dialog box to make notes about the action.

## Cancel Workflow Event

This action cancels an event that has been defined within a workflow.

**Figure 5-31 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.

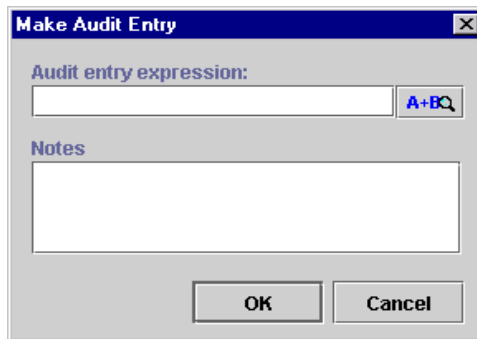
## Make Audit Entry

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” in Chapter 3, “Defining and Maintaining Workflows.”

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 5-32 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 6, “Using Expressions and Conditions.”

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



# 6 Using Expressions and Conditions

The following sections explain how to use the WebLogic Process Integrator Expression Builder and expression components:

- Workflow Expression Functionality
- Using the Expression Builder
- Using Expression Components

# Workflow Expression Functionality

A workflow expression is an algebraic expression that defines a calculation that the 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

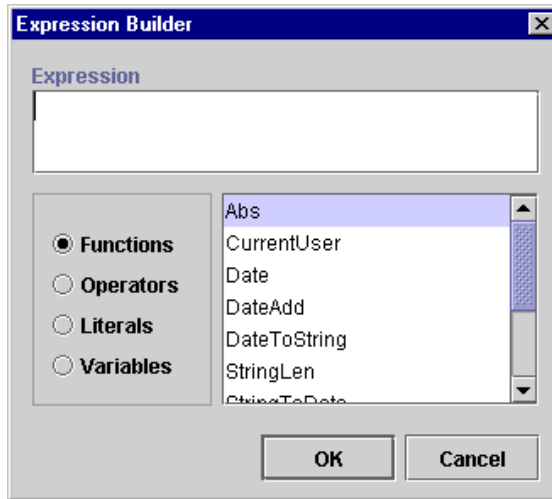
The Expression Builder enables you to build expressions to be executed in WebLogic Process Integrator at run time. Expressions are made up of functions, operators, literals, and variables. These expression components are discussed in "Using Expression Components."

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 6-1 Expression Builder Button**



**Figure 6-2 Expression Builder Dialog Box**



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.

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 6-1 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. |

**Table 6-1 Invalid Expression Messages**

| <b>Message</b>                                    | <b>Possible Cause(s)</b>                                                                                                                                                                      |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                                                            |
| Unrecoverable syntax error                        | <p>The expression contains an invalid token. In this example, name is invalid because the + operator was omitted: "Name: " name.</p> <p>The Expression should have been: "Name: " + name.</p> |

# 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.

### 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:

**Table 6-2 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:

**Table 6-3 Format Values**

| Format | Description                                                                                                                                   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| yyyy   | Year<br>for example: 2000                                                                                                                     |
| MM     | Numeric month of the year<br>for example, January = 01, February = 02, and so on                                                              |
| dd     | Numeric day of the month<br>for example: 02 or 28                                                                                             |
| DD     | 3- digit day of the year based on the total number of days (365) in a year<br>for example: October 18, 2000 = 292 (the 292nd day of the year) |
| hh     | Hours in non-military time<br>for example: 1 pm = 01, 2 pm = 02, and so on                                                                    |
| HH     | Hours in military time<br>for example: 1 pm = 13, 2 pm = 14, and so on                                                                        |
| mm     | Minutes<br>for example: 02 minutes                                                                                                            |
| ss     | Seconds<br>for example: 35 seconds                                                                                                            |
| SSS    | Milliseconds<br>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
```

## 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 6-4 TaskAttribute Expressions**

| Information                 | Expression                              |
|-----------------------------|-----------------------------------------|
| Task instance ID            | <code>TaskAttribute("TaskId")</code>    |
| User ID of assignee         | <code>TaskAttribute("Assignee")</code>  |
| Priority set by application | <code>TaskAttribute("Priority")</code>  |
| Task overdue date           | <code>TaskAttribute("Due")</code>       |
| Name of task                | <code>TaskAttribute("Name")</code>      |
| Task start date/time        | <code>TaskAttribute("Started")</code>   |
| Completed date/time         | <code>TaskAttribute("Completed")</code> |
| Application comment         | <code>TaskAttribute("Comment")</code>   |

## 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 6-5 WorkflowAttribute Expressions**

| Information            | Expression                                             |
|------------------------|--------------------------------------------------------|
| Workflow instance ID   | <code>WorkflowAttribute("InstanceId")</code>           |
| Workflow template ID   | <code>WorkflowAttribute("TemplateId")</code>           |
| Template definition ID | <code>WorkflowAttribute("TemplateDefinitionId")</code> |
| Workflow initiator     | <code>WorkflowAttribute("Initiator")</code>            |
| Parent workflow ID     | <code>WorkflowAttribute("ParentId")</code>             |
| Workflow name          | <code>WorkflowAttribute("Name")</code>                 |
| Start date/time        | <code>WorkflowAttribute("Started")</code>              |
| Completed date/time    | <code>WorkflowAttribute("Completed")</code>            |

**Table 6-5 WorkflowAttribute Expressions**

| Information                                                                                                                                | Expression                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| ID set by application                                                                                                                      | <code>WorkflowAttribute("Id")</code>            |
| Application comment                                                                                                                        | <code>WorkflowAttribute("Comment")</code>       |
| Name of Java exception class that raised the error                                                                                         | <code>WorkflowAttribute("ErrorType")</code>     |
| Message number of the error being handled by the exception handler                                                                         | <code>WorkflowAttribute("ErrorNumber")</code>   |
| One of four values:<br>Error_System = "1"<br>Error_Workflow = "2"<br>Warning_Workflow = "3"<br>Error_Custom = "4"                          | <code>WorkflowAttribute("ErrorSeverity")</code> |
| <b>Note:</b> Value is set to Error_Custom when a workflow invokes an exception handler via the Invoke Error Handler action or via the API. |                                                 |
| Message text of the error being handled by the exception handler                                                                           | <code>WorkflowAttribute("ErrorText")</code>     |

**Note:** See “WebLogic Process Integrator Error Messages” in Chapter 7, “Handling Workflow Exceptions,” for a list of WebLogic Process Integrator error messages.

## WorkflowVariable

Returns the value of a workflow variable for a particular workflow instance.

`workflowVariable(instanceid, variable)`

*instanceid* is the workflow instance ID.

*variable* is the workflow variable.

## XPath

`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 6-1 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 6 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       |

## 6 *Using Expressions and Conditions*

---

|    |                      |          |                       |
|----|----------------------|----------|-----------------------|
| 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 7 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 6-8 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          | Numeric           | Unary plus                                                             |
|                       |              | expr1 + expr2   | String or numeric | String concatenation or numeric addition                               |
| Minus                 | -            | - expr          | Numeric           | Unary minus (negates expr)                                             |
|                       |              | expr1 – expr2   | Numeric           | 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 6-8 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 6-9   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:<br><br>\r carriage return character<br>\n line feed character<br>\' single quote character<br>\" double quote character<br>\f form-feed character<br>\t tab character<br>\\ back slash character<br>\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 $-2^{53} \times 10^{104}$ to $+2^{53} \times 10^{104}$ (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.



# 7 Handling Workflow Exceptions

The following sections explain how to handle internally and externally generated workflow exception conditions:

- Workflow Exception Handling Overview
- Understanding the WebLogic Process Integrator Transaction Model
- 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.

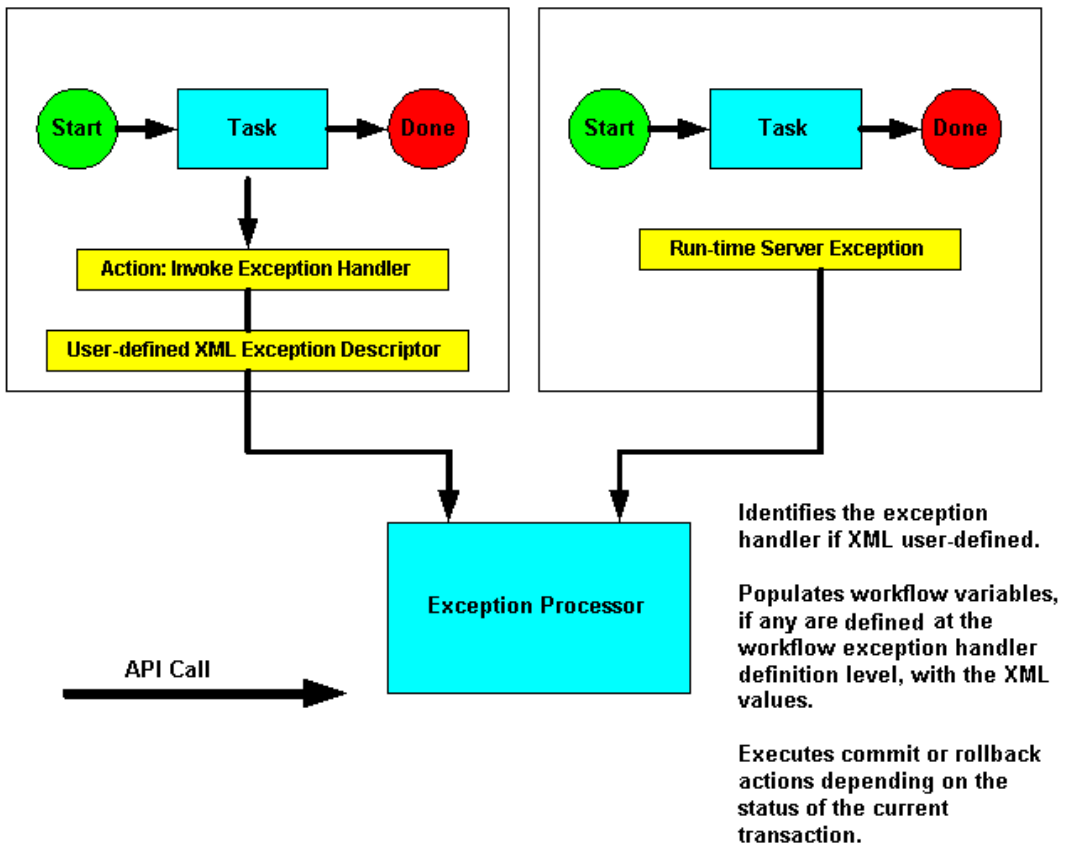
The workflow exception handling facility allows you to define customized exception handlers to act as the initial exception handler and specify actions to be executed in response to exceptions. You set such exception handlers using the Set Workflow Exception Handler action; an exception handler, when 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: `WorkflowProcessor.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 7-1 Workflow Exception Handling**



# Understanding the WebLogic Process Integrator Transaction Model

All Enterprise Java Bean (EJB) methods in the public API (with the one exception of the Audit EJB) use container-managed transactions with the `TX_REQUIRED` transaction attribute. The WebLogic Process Integrator Worklist application executes user commands by calling a method on the Worklist EJB within a discrete transaction. If another application or EJB calls a WebLogic Process Integrator API method, the transactional behavior depends on whether the caller has a transaction context. If there is a transaction context, the WebLogic Process Integrator EJB uses it; otherwise, the EJB container automatically creates a new one in which to execute the call.

The workflow template definition determines what WebLogic Process Integrator does when processing such API calls. For example, when instantiating a workflow, WebLogic Process Integrator first executes the Created actions for all tasks in the workflow. It then processes Start node(s) as follows:

- In the case of an event or time triggered workflow, the system activates the Start node corresponding to the event
- In the case of a manually started or called workflow (sub-workflow), it activates all manual or called Start nodes respectively.

Upon activation, a Start node first initializes variables, then activates its associated actions. Next, it activates any successor nodes. The activation of these successor nodes proceeds according to the rules for each node type (decision, event, task, and so on). For example, a Decision node first evaluates its condition, executes the appropriate list of true or false actions, and finally activates its true or false successor nodes.

Similarly, a Task node has a list of Activated actions, which it executes in sequence. If one of these actions marks a task as complete, that task executes its Done actions before activating the successor nodes. If an action marks the workflow as complete or aborts the workflow, no further processing occurs.

If an exception occurs, WebLogic Process Integrator invokes the current exception handler (by default, the System exception handler). The exception handler can perform compensating actions and determines how to proceed afterwards.

Possible outcomes are:

- Rollback the User Transaction and rethrow the exception.
- Stop processing and commit return to the caller.
- Retry the failed operation.
- Ignore the error and continue processing.

Likewise, when you work with tasks in the WebLogic Process Integrator Worklist application, the EJB container creates an encompassing transaction in which to execute each command. Such commands include:

- Executing a task
- Marking a task as done
- Marking a task as not done
- Reassigning a task to another user or role
- ...

Activation events propagate outwards along the paths emanating from the point of origin, according to the built-in processing rules for each node type. When a Start Workflow action instantiates a sub-workflow, the instantiation and unitization of the sub-workflow follows the pattern described above, but it is encompassed by the outer transaction in which the action itself is executing.

When WebLogic Process Integrator catches or throws an exception, the transaction commit/rollback behavior depends upon where the exception occurred and on what the active exception handler does. If the user transaction has not already been set for rollback only and the exception is recoverable (in other words, it is a workflow warning) and occurred while processing an action or setting an event variable, the system allows the exception handler to decide the outcome. Otherwise, the system sets the user transaction for rollback only, invokes the active exception handler then rethrows the exception.

If an application enlists a WebLogic Process Integrator EJB in a global transaction and this EJB throws an exception that does not result in the user transaction being set for rollback only, the behavior is application defined. Similarly, even if such a WebLogic Process Integrator call succeeds, there is no guarantee that the transaction will eventually be committed by the external application (or EJB). An exception or other error condition may subsequently occur, resulting in the entire transaction being rolled back.

If the calling application creates the transaction context, then the calling application has control over committing and rolling back the transaction. However, if the container creates the transaction, then it commits the transaction when it successfully gets control back from WebLogic Process Integrator. WebLogic Process Integrator returns control to the container when it has no more work to do in processing the request.

In cases involving workflow processing (such as task execution, workflow instantiation, event or timer processing), WebLogic Process Integrator returns control when there are no more actions to execute synchronously (when the workflow instance enters a quiescent state, waiting for the next request).

# 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 Error Handler action or the Set Workflow Error Handler action. See “Using Exception Handling Actions.”

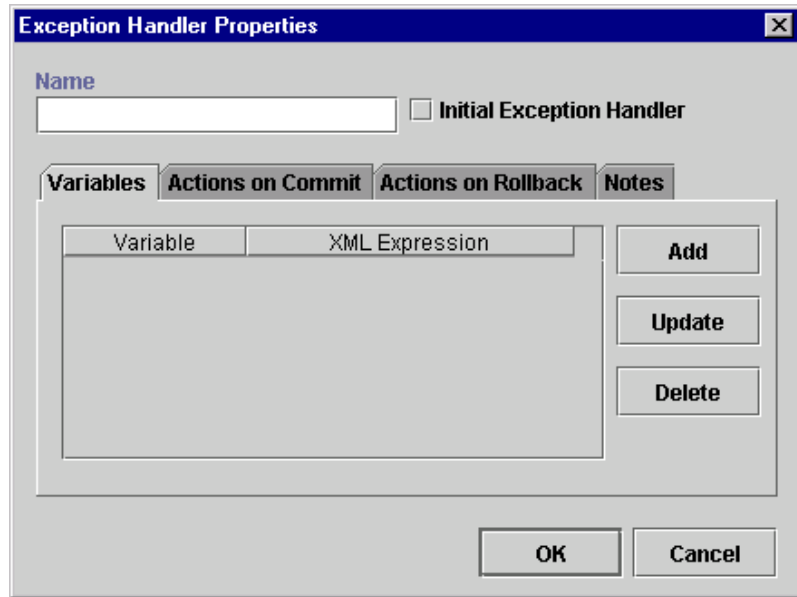
To define an exception handler for a workflow template definition within WebLogic Process Integrator, proceed as follows:

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” in Chapter 3, “Defining and Maintaining Workflows.”)

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.

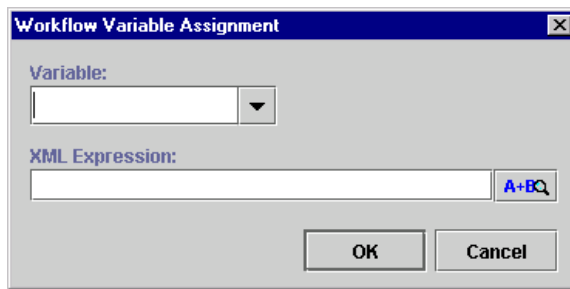
**Figure 7-2 Exception Handler Properties Dialog Box**



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” in Chapter 4, “Working with Workflow Components.”)

**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 6, “Using Expressions and Conditions.”

**Figure 7-3 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.

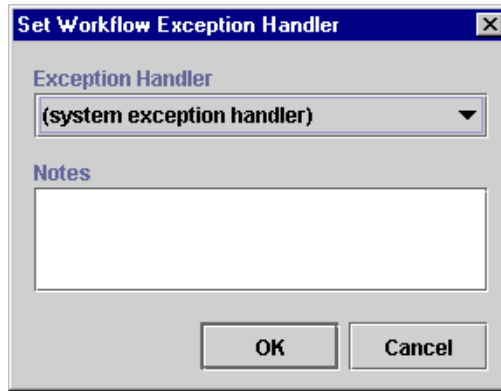
For instructions on using the Invoke Exception Handler action, see “Using XML Actions” in Chapter 9, “Understanding WebLogic Process Integrator Events and XML Documents.”

## Set Workflow Exception Handler

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 7-4 Set Workflow Exception Handler**



View or enter information in the Set Workflow Exception Handler dialog box.

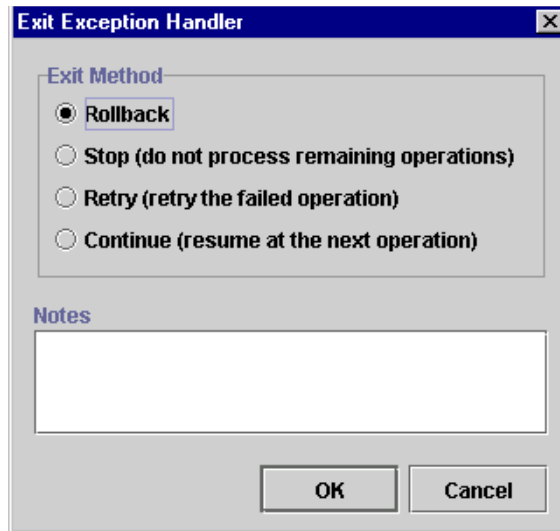
- **Exception Handler** — Displays a list of exception handlers to choose from. If you have not defined an exception handler, the list only displays the system exception handler.
- **Notes** — Free-form text box for notes on the action.

## 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.”

**Figure 7-5** Exit Exception Handler



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 error handler and does not execute any follow-on actions.
- **Retry** — Exits the error 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.

Refer to Table 6-5 in Chapter 6, “Using Expressions and Conditions,” to obtain a list of Workflow Attribute values. 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 7-1 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.                                                                                |

## 7 *Handling Workflow Exceptions*

---

| 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}".                                                          |

## 7 *Handling Workflow Exceptions*

---

| 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.                          |

| <b>ErrorNumber</b> | <b>ErrorText</b>                                                                                                                            |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 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. |



# 8 Monitoring Workflows

The following sections discuss workflow monitoring.

- Using the Workflow Monitoring Features
- Viewing Workflow Status
- Viewing Workflow Variables
- Deleting Workflows
- Using Workload Reports
- Using Statistics Reports

# Using the Workflow Monitoring Features

Use the 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

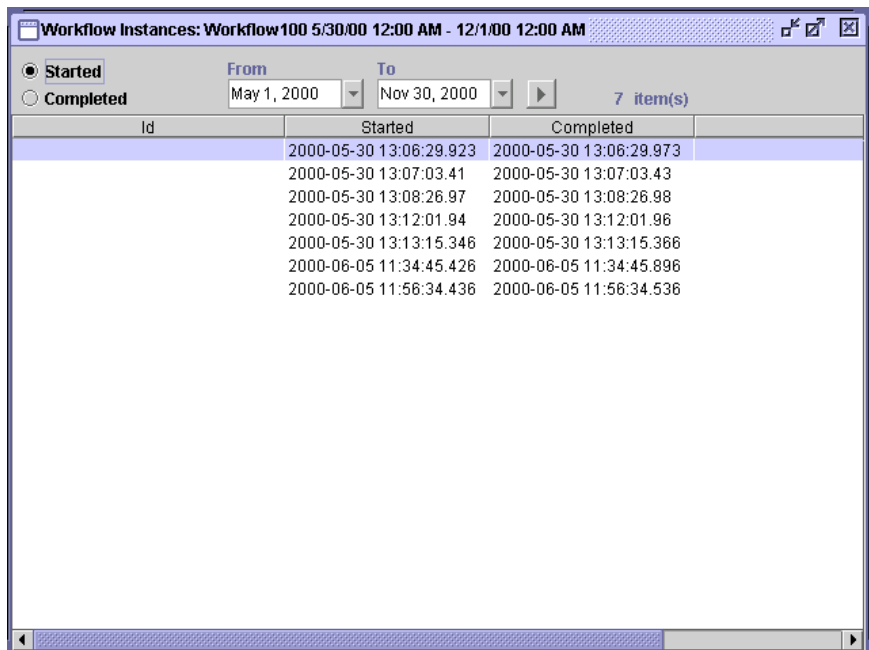
The following WebLogic Process Integrator folder tree items are relevant to the monitoring features:

- **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 8-1 Workflow Instances Dialog Box**



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 8-2 Workflow Instance From/To Dates**

The screenshot shows a dialog box for selecting date ranges. At the top, there are two date pickers labeled 'From' and 'To', both set to 'May 16, 2000'. Below these is a calendar for 'May 2000'. The calendar grid shows days of the week (Sun to Sat) and dates (1 to 31). The date '16' is highlighted in red. At the bottom right of the calendar, there is a 'Time' field set to '12:00 AM'.

For each workflow shown, the following information is displayed:

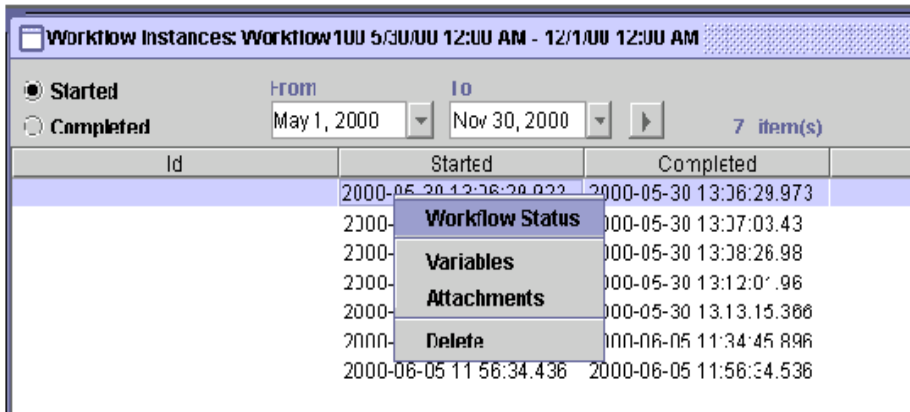
- ID — workflow template definition ID. This is maintained automatically by the workflow engine from the expression specified in the properties at workflow template definition time.
- Started — date the workflow was instantiated.
- Completed — completion date of the workflow. If it is not completed, this column will be blank.

For performance reasons, the 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 8-3 Workflow Status Menu Option**



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 8-1 Workflow Status Icons**

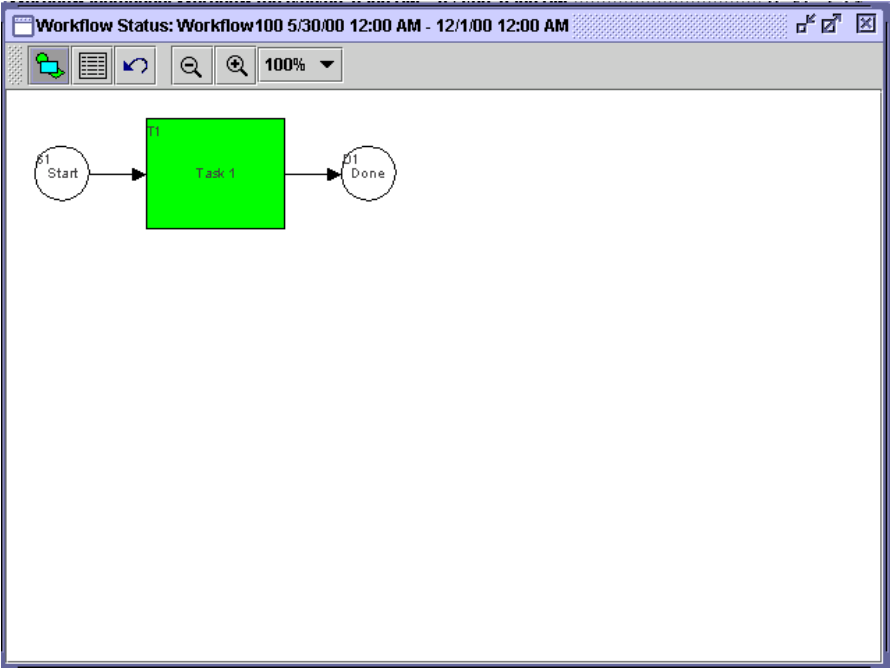
| Symbol                                                                              | Shape Name               | Purpose                                              |
|-------------------------------------------------------------------------------------|--------------------------|------------------------------------------------------|
|  | Graphical Representation | Displays a graphical representation of the workflow. |
|  | List View                | Displays a list view of the workflow.                |

**Table 8-1   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).

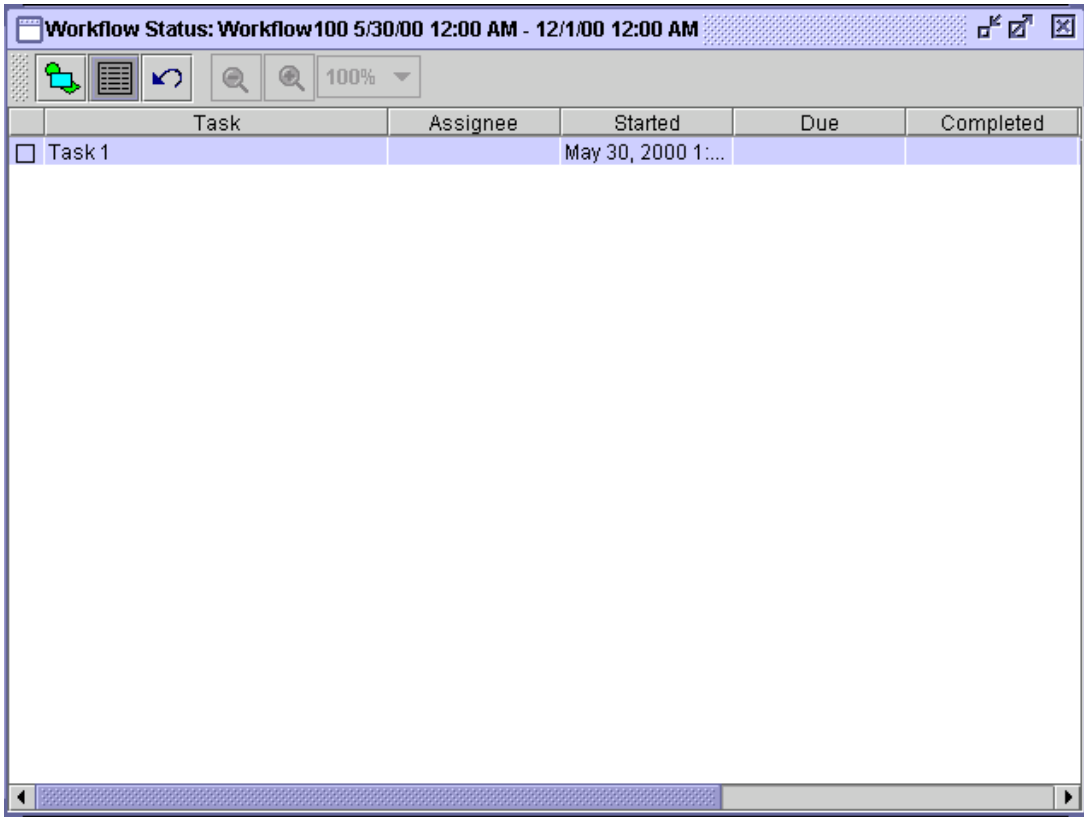
**Figure 8-4   Graphical Representation of Workflow**



In the graphical representation, the active tasks are green, the executed tasks are gray, and the inactive tasks are white.

Clicking the list view icon displays a list view of the workflow and its associated tasks.

**Figure 8-5 List View of Workflow**



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 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 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.

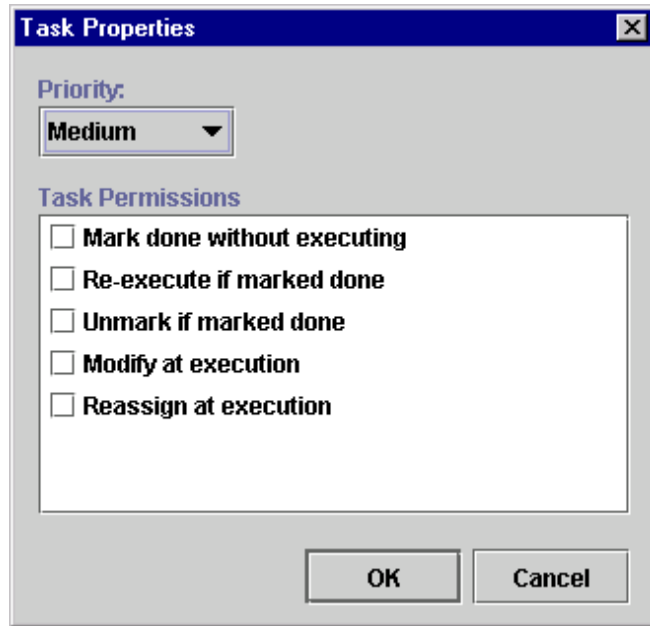
**Figure 8-6 Reassign Task Dialog Box**

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 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 8-7 Task Properties Dialog Box**



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

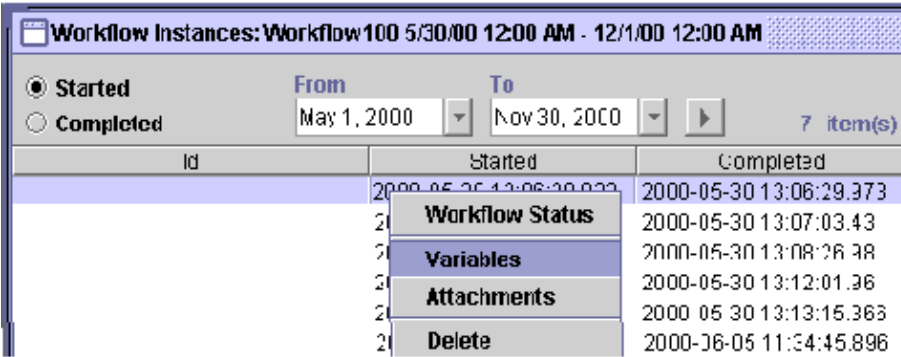
the Worklist application. If this is not selected, then none of the task permissions can be altered.

- 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

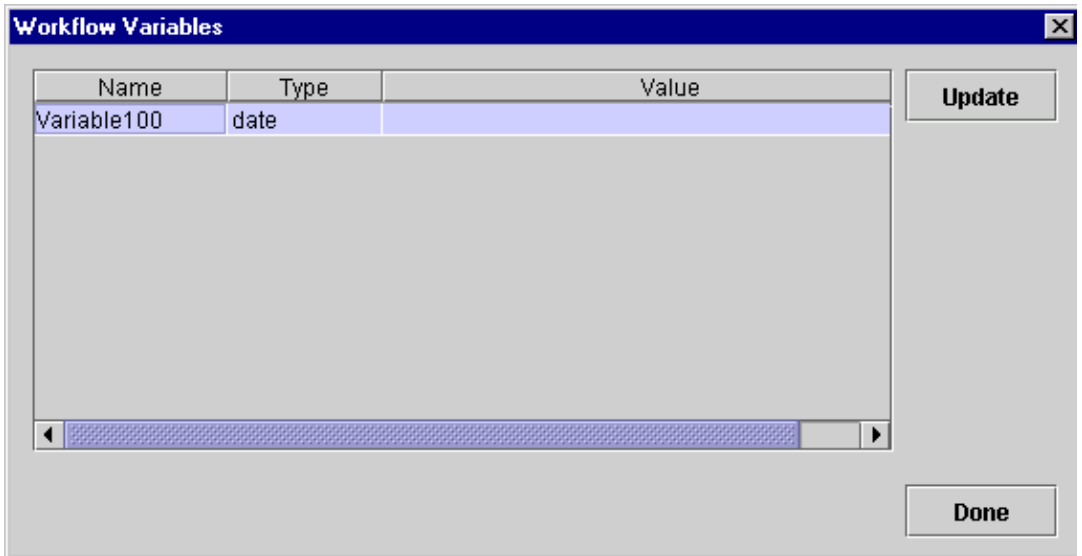
The second option on the pop-up menu, which is displayed by right-clicking the instance, is the workflow variables. This command is used to display a list of all workflow instance variables.

Figure 8-8 Variables Menu Option



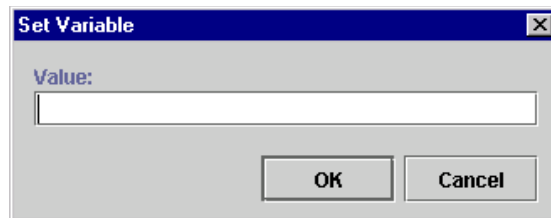
When selected, the Variables menu option displays the Workflow Variables dialog box, which lists the variables of the selected workflow instance.

**Figure 8-9 Workflow Variables Dialog Box**



The Workflow Variables dialog box displays each variable in a list, showing its name and its current value. Click the Update command or double-click an item in the list to display the Set Variable dialog box showing the current value. You can change this value.

**Figure 8-10 Set Variable Dialog Box**

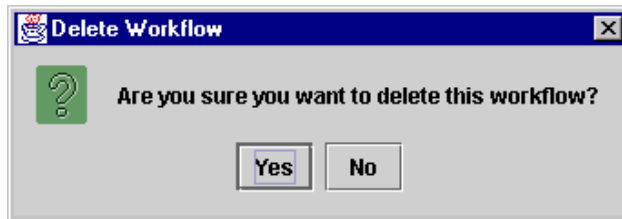


When you save the change, a message is sent to the WebLogic Process Integrator server to reset the variable. Because workflow logic often depends on the value of variables, use caution when manually changing the value of a variable.

# Deleting Workflows

The Delete menu option allows you to delete a workflow instance. Selecting this menu option displays the Delete Workflow dialog box.

**Figure 8-11 Delete Workflow Dialog Box**

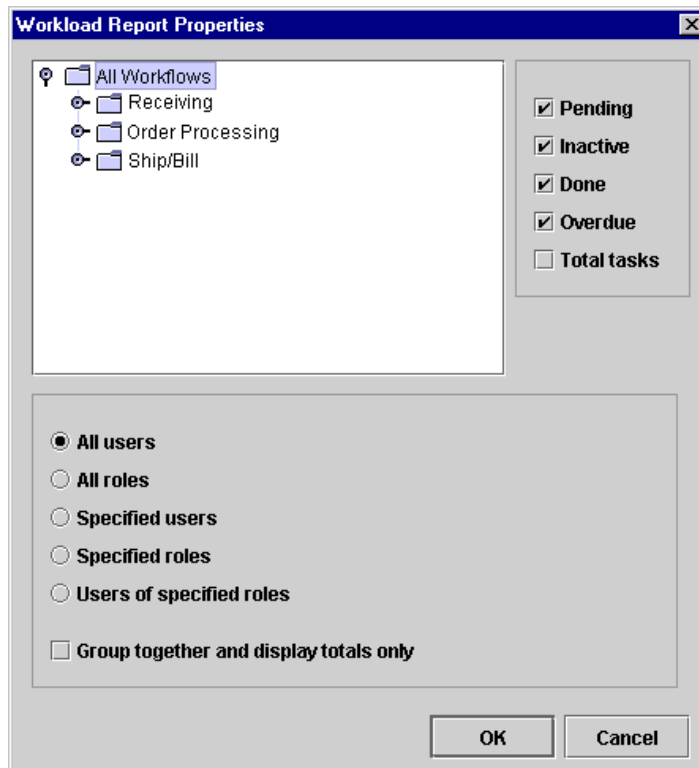


A positive confirmation sends a message to the WebLogic Process Integrator server prompting the deletion of the selected workflow instance(s).

# Using Workload Reports

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 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 8-12 Workload Report Properties Dialog Box**



## Compiling Workload Report Information

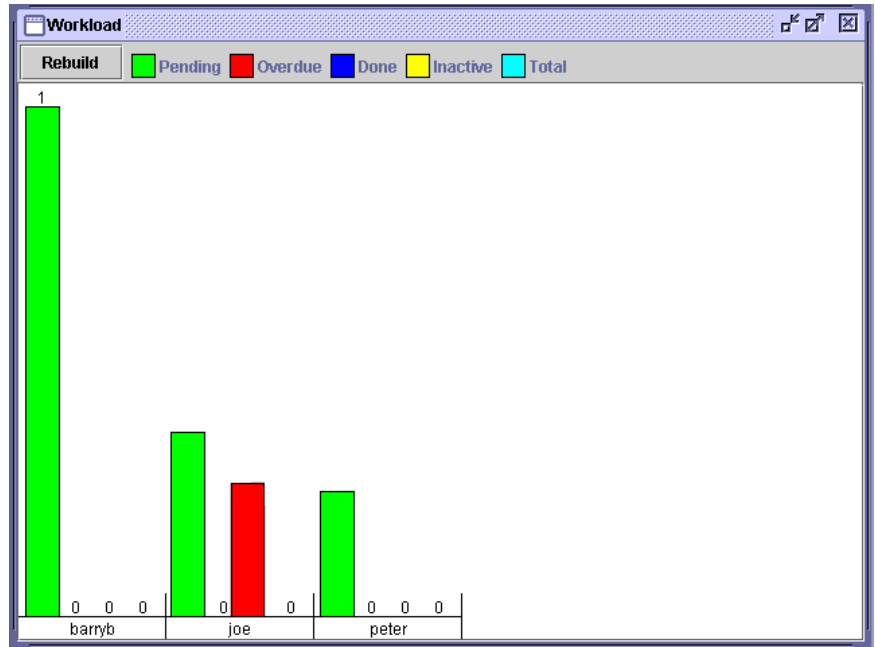
The workload report will display information according to the selections you make in the Workload Report Properties dialog box:

- 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 8-13 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.

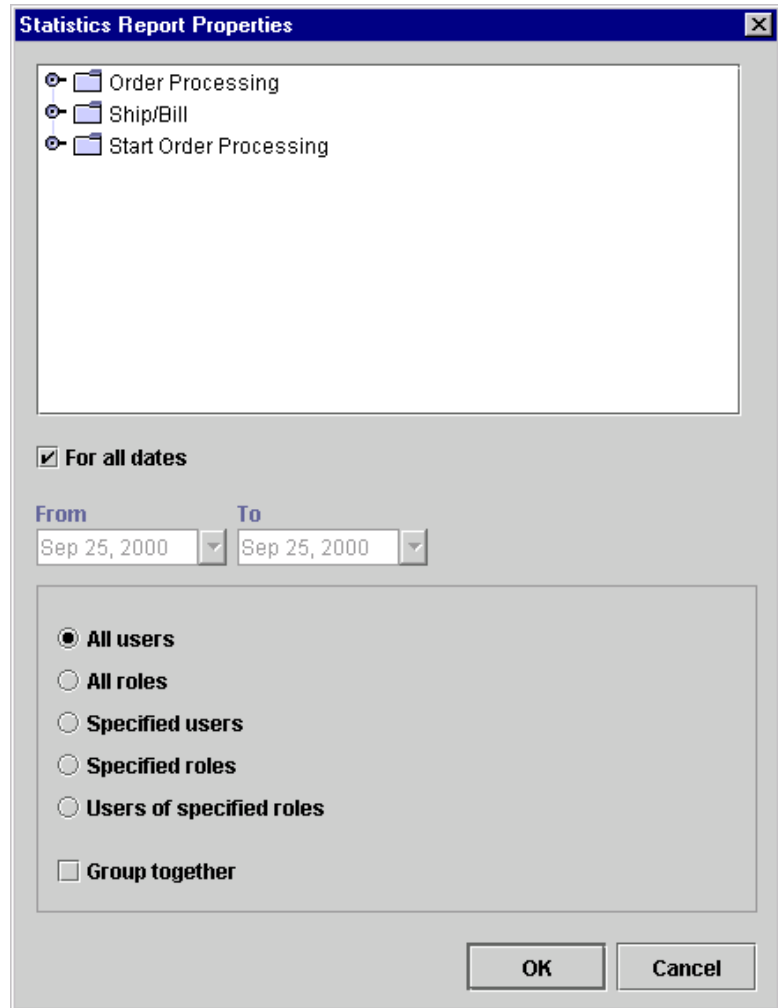
# Using Statistics Reports

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 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 8-14 Statistics Report Properties Dialog Box**



- List of workflow templates — only tasks within the selected workflow template will be included 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 workload information for each user.
- All roles — displays workload information for each role.
- Specified users — displays workload information for each selected user.
- Specified roles — displays workload information for each selected role.
- 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 — 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. The top of the panel on the right has a drop-down list to select the organization to run the report on. Below this is a set of three option buttons. The selected option determines the unit of time used in the report. This can be minutes, hours, or days. The arrow is used to regenerate the report when the units have been changed.

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.

- Min — minimum amount of time to spend performing this task.
- Max — maximum amount of time to spend performing this task.

Because the report definition is stored separately from the results, the statistics report can be run at any time.

# 9 Understanding WebLogic Process Integrator Events and XML Documents

The following sections discuss XML (extensible markup language) events and XML document handling within WebLogic Process Integrator:

- XML Overview
- Processing Events and Handling XML Documents
- Defining a Workflow Event Trigger
- Using the XML Notification: The WebLogic Process Integrator Event
- Using XML Actions
- Understanding Workflow Interaction

# XML Overview

XML is a system for defining, validating, and sharing document formats. Like HTML, XML is a markup language derived from SGML (Standard Generalized Markup Language). However, unlike HTML, which was created to allow cross-platform formatting of information for display, XML is best suited for organizing data. While HTML specifies how a document should be displayed, it does not describe what type of information is contained in the document or how it is organized. XML fills this void, allowing document authors to organize information in a standard manner.

The development of XML is a public project headed by the World Wide Web Consortium and is not owned by any one company. In a recommendation on February 10, 1998, the World Wide Web Consortium (W3C) accepted version 1.0 of the XML specification. However, XML is still evolving. Specifics on the XML standard and syntax can be found at the W3C site (<http://www.w3.org/xml>).

XML allows information exchange across different and potentially incompatible systems and platforms. Because XML is a text based, non-binary format and uses syntax rather than binary markers to organize data, it can be deployed on just about any platform.

Although XML specific editors are now available, because XML is just text, any editor or program that can generate text files can be used to create XML documents.

XML is license-free, platform independent, and well-supported. It is rapidly being adopted by software companies as a standard means of sending and receiving data.

## XML Elements and Attributes

XML uses tags to distinguish document structures and uses attributes to encode extra document information.

XML tags usually surround an identified object in the data stream. A start-tag and an end-tag, together with the data enclosed by them, constitute an element. For example:

```
<OrderType>New</OrderType>
```

The element `OrderType` is assigned the value “New”.

XML elements can contain other, embedded elements, and the entire document must be enclosed by a single document element. The following example shows an XML document with a root element `Order` and three subelements: `OrderId`, `OrderType`, and `WorkflowStatus`.

```
<Order>
  <OrderId>1</OrderId>
  <OrderType>New</OrderType>
  <WorkflowStatus>New</WorkflowStatus>
</Order>
```

It is possible for an element to hold information about its content beyond just its name. This “information about information,” or metadata, is stored in an attribute. When a document type definition (DTD) is not in use, the attribute value is simply considered to be a unit of text and no type distinction is made. But when a DTD is in use, more control can be exerted over the range of values allowed for each attribute. An attribute is associated with a particular element by the DTD, and is assigned an attribute type.

## Document Type Definition (DTD)

It is possible to predefine which elements are allowed within other elements. An optional document type definition (DTD) contains rules for each element allowed within a specific class of documents so that the placement of elements and their attributes can be controlled and validated.

The DTD establishes formal document structure rules. It defines the elements that can be used and the relationships between them. In a way, the DTD defines a document structure tree.

Some XML editors can read the DTD and use it to build the document model. The editor can then identify errors in the XML document if the data contained in the document does not conform to the DTD structure.

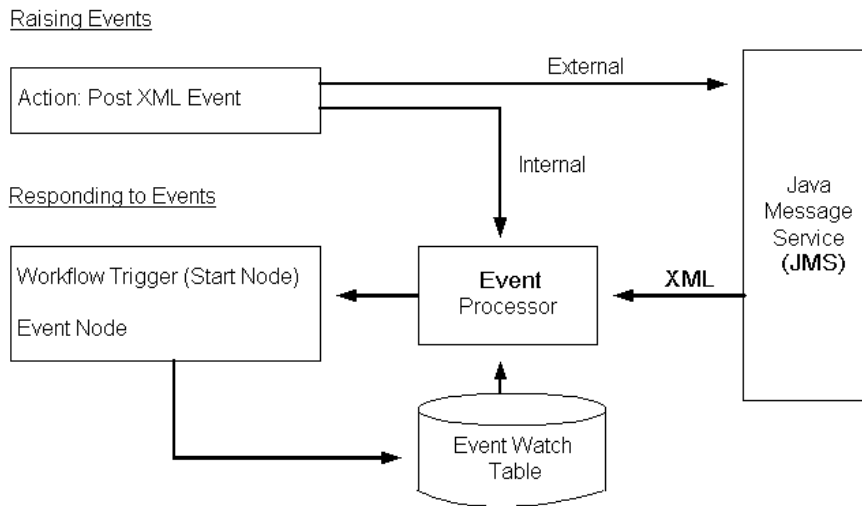
The WebLogic Process Integrator DTDs are provided on the documentation CD at:

```
Drive:/docs/wlpi/v1_2/classdocs/com/bea/wlpi/common/doc-files
```

# Processing Events and Handling XML Documents

The following diagram illustrates the XML event processing scheme within the WebLogic Process Integrator server.

**Figure 9-1 XML Event Processing Scheme**



Notifications to the WebLogic Process Integrator server and automatic workflow triggering are event driven. You can set up workflow events to wait for a notification and perform actions or start workflows upon the event being triggered. You define events in XML format by using the Start node for workflow triggering. Event definitions are then registered automatically in the Event Watch table.

Events are triggered and processed upon receipt of XML messages that match the criteria stored in the event definitions. These XML documents can either be internal to the WebLogic Process Integrator environment (raised by the Post XML Event action within tasks of workflows), or they can be received from applications or other external sources. In the latter case, the XML documents are received from the Java Message Service (JMS) and placed in the event handler request queue for processing.

The Post XML Event action can be used to raise external events (for processing by other applications) by posting the XML message to a Java Message Service (JMS) topic.

## Defining a Workflow Event Trigger

A workflow can be triggered automatically, in which case it responds to a particular action or condition taking place. Custom triggers such as database events can be defined in order to start workflows. The workflow event trigger is already defined in WebLogic Process Integrator and is delivered as part of the standard software. When this trigger is set up, the workflow is responding to an incoming XML document.

A workflow event trigger is defined in the Start Properties dialog box. See the section entitled “Defining Start” in Chapter 4, “Working with Workflow Components,” for instructions on accessing this dialog box. The Start dialog box allows you to specify the root element of the XML field, the condition under which the trigger should activate, and the variable assignments, if any.

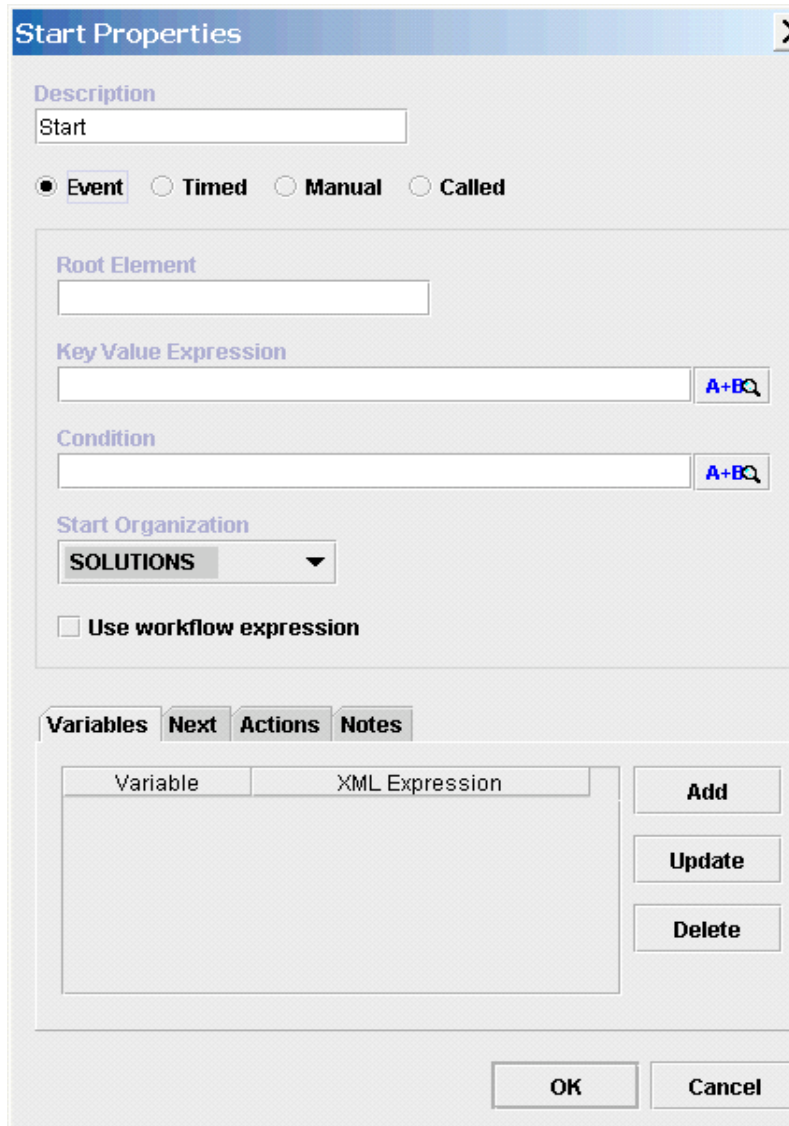
To define a workflow event trigger:

1. In the Start Properties dialog box, select the Event option.
2. In the Root Element field, enter the XML document root element to serve as the event trigger.
3. In the Key Value Expression field, define a key value for the XML document. For more information, see “Setting Up a Key Value for Each XML Root.”
4. In the Condition field, define the condition under which the event is to be triggered. The workflow will be triggered by the XML document, provided that this defined condition is verified.
5. Select a Start Organization from the drop-down list containing those 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 will be 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 this to enter an expression that will be evaluated at run time and become the name of an organization. The expression can be either a string that specifies the organization (case sensitive) or an expression using information passed in the XML document.

**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 6, “Using Expressions and Conditions.”

Figure 9-2 Start Properties Dialog Box



The **Start Properties** dialog box is used to configure the start of a workflow. It includes fields for a description, event type selection, root element, key value expression, condition, start organization, and a section for workflow variables.

**Description**  
Start

☒ **Event**   ☐ **Timed**   ☐ **Manual**   ☐ **Called**

**Root Element**  
[Empty text box]

**Key Value Expression**  
[Empty text box] **A+B**

**Condition**  
[Empty text box] **A+B**

**Start Organization**  
SOLUTIONS ▼

☐ **Use workflow expression**

**Variables**   **Next**   **Actions**   **Notes**

| Variable           | XML Expression |
|--------------------|----------------|
| [Empty table body] |                |

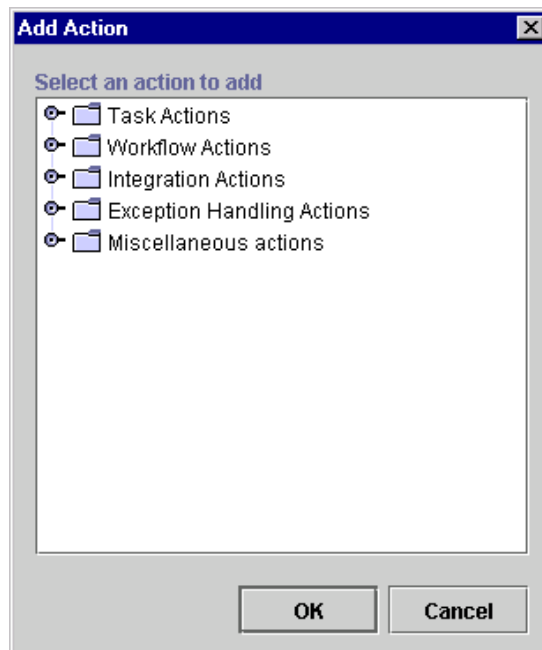
**Add**  
**Update**  
**Delete**

**OK**   **Cancel**

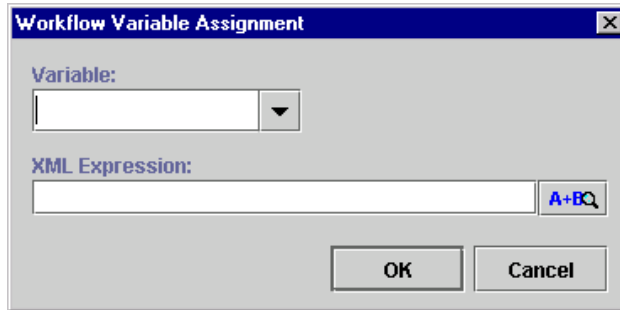
7. Select the **Variables** tab to assign values to workflow variables from the information contained in the XML document. Click **Add** to display the Workflow Variable Assignment dialog box, where each variable can be assigned an XML expression.

- 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 6, “Using Expressions and Conditions.”
8. Select the Next tab to display a list of all task, decision, join, and done nodes in the current workflow. An event can have one or more successors. 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. Refer to Chapter 5, “Using Actions,” for more details.

**Figure 9-3 Add Action Dialog Box**



**Figure 9-4 Workflow Variable Assignment Dialog Box**



10. Use the Variable drop-down list to select an existing variable or enter a new variable.
11. Use the XML Expression field to define the XML Expression for this 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 6, "Using Expressions and Conditions."

12. Click OK to confirm your changes.
13. Click OK in the Start Properties dialog box to confirm your changes.

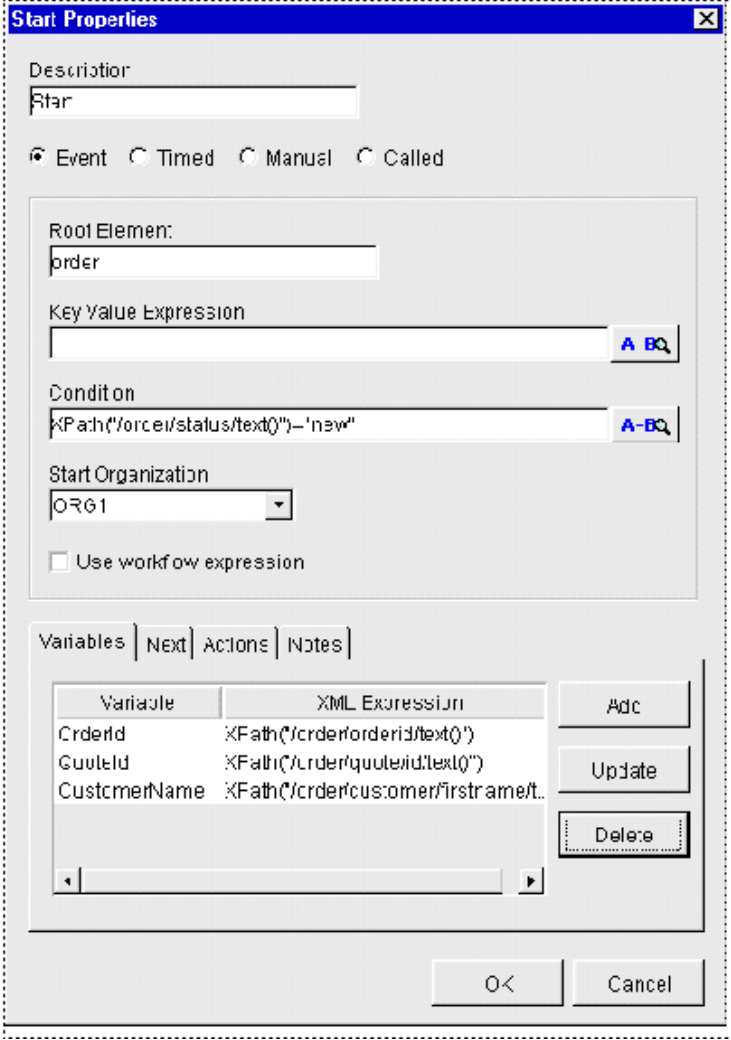
# Workflow Examples

The following example shows a workflow that is triggered by the XML document containing the root element `order` if the `workflowstatus` element of the XML document is “new”.

The following workflow variables are populated with the values contained in the XML document:

- `OrderId`
- `QuoteId`
- `CustomerName`

Figure 9-5 Start Properties Dialog Box



The dialog box is titled "Start Properties" and contains the following fields and controls:

- Description:** A text field containing the word "Star".
- Event Type:** Four radio buttons labeled "Event", "Timed", "Manual", and "Called". The "Event" button is selected.
- Root Element:** A text field containing the word "order".
- Key Value Expression:** An empty text field with a blue "A-BQ" button to its right.
- Condition:** A text field containing the XPath expression `<XPath("/order/status/text()")='new'` with a blue "A-BQ" button to its right.
- Start Organization:** A dropdown menu currently showing "ORG1".
- Use workflow expression:** An unchecked checkbox.
- Navigation:** Four tabs labeled "Variables", "Next", "Actions", and "Notes". The "Variables" tab is active.
- Variables Table:**

| Variable     | XML Expression                                         |
|--------------|--------------------------------------------------------|
| Orderid      | <code>&lt;XPath("/order/orderid/text()")</code>        |
| Quoteid      | <code>&lt;XPath("/order/quoteid/text()")</code>        |
| CustomerName | <code>&lt;XPath("/order/customer/firstname/t...</code> |
- Buttons:** "Add", "Update", and "Delete" buttons are located to the right of the variables table. At the bottom right are "OK" and "Cancel" buttons.

The corresponding XML document is as follows:

### **Listing 9-1 XML Document**

---

```
<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>
```

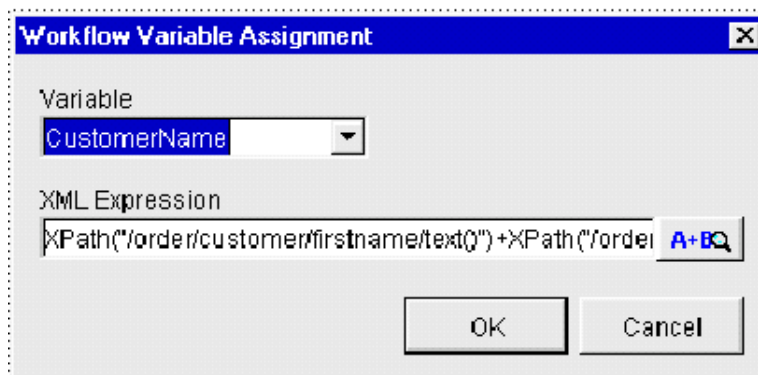
```
</Order>
```

The root element is order.

The XML expression is `XPath("/Order/Customer/FirstName/text()")`  
`+ XPath("/Order/Customer/LastName/text()")`

The customer element has three subelements: `FirstName`, `LastName`, and `Address`. Values from the XML document are extracted using the XPath language.

**Figure 9-6 Workflow Variable Assignment Dialog Box**



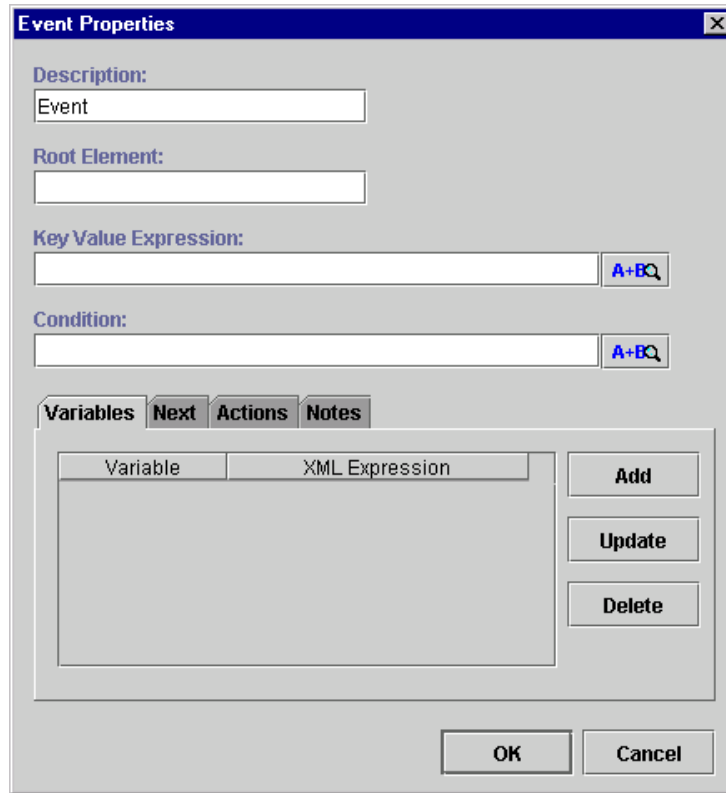
# Using the XML Notification: The WebLogic Process Integrator Event

A workflow event represents a notification node. Any time an event is defined, the workflow waits for an XML message to trigger the event. Upon that trigger, various subactions defined within the event can be executed and/or workflow variables can be set.

To define a WebLogic Process Integrator notification event:

1. Open the Event Properties dialog box as described in “Defining Events” in Chapter 4, “Working with Workflow Components,” for instructions on accessing this dialog box.

Figure 9-7 Event Properties Dialog Box



The dialog box is titled "Event Properties" and contains several input fields and a table. The fields are labeled "Description:", "Root Element:", "Key Value Expression:", and "Condition:". Each field has a text input area and a small "A+B" icon to its right. Below these fields are four tabs: "Variables", "Next", "Actions", and "Notes". The "Variables" tab is selected, showing a table with two columns: "Variable" and "XML Expression". To the right of the table are three buttons: "Add", "Update", and "Delete". At the bottom of the dialog are "OK" and "Cancel" buttons.

| Variable | XML Expression |
|----------|----------------|
|----------|----------------|

2. In the Description field, enter a meaningful name for the event you will define. The name will be displayed in the workflow drawing area.
3. In the Root Element field, enter the XML document root element to serve as the event trigger.

4. In the Key Value Expression field, define a key value for the XML document.

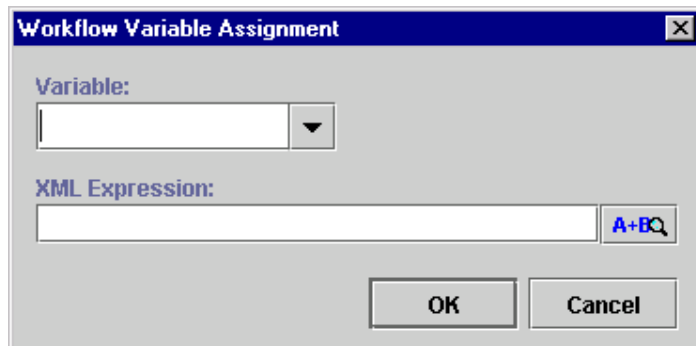
**Note:** It is possible to define a key value for each XML root in order to speed up the search for a specific XML trigger. For instructions, see “Setting Up a Key Value for Each XML Root.”

5. In the Condition field, define the condition under which the event is to be triggered. The workflow will be triggered by the XML document, provided that this defined condition is verified.

6. Use the Variables tab to assign values by way of the XML document. Click Add to display the following dialog box where each variable can be assigned an XML Expression.

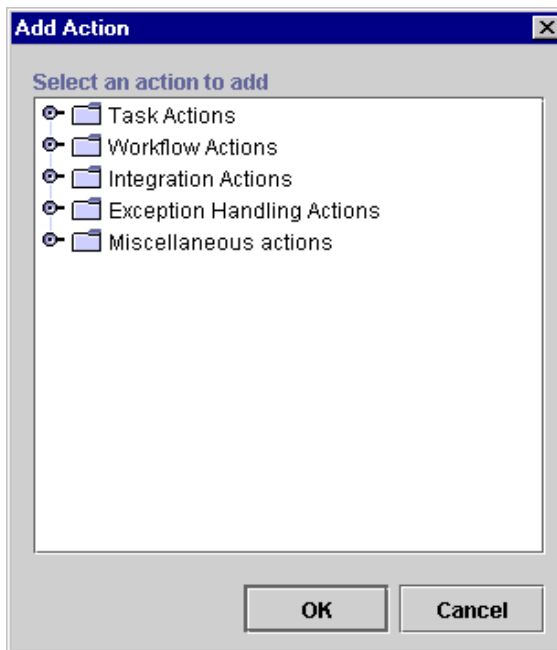
**Note:** Expressions are built using WebLogic Process Integrator expression syntax and typically use the syntax of XPath function to extract values out of XML documents. If you know the syntax of the expression you want to use, you can type the expression directly in the field. Alternatively, click 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 6, “Using Expressions and Conditions.”

**Figure 9-8 Workflow Variable Assignment Dialog Box**



7. Use the Variable drop-down list to select an existing variable or enter a new variable.
8. Use the XML Expression field to define the XML Expression for this 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 6, “Using Expressions and Conditions.”
9. Click OK to confirm your changes.
10. Select the Next tab of the Event Properties dialog box to display a list of all task, decision, join, and done nodes in the current workflow.
11. Select the node to succeed the event. An event can have one or more successors. The indicated items in this list will be the next activated in the flow after the event is triggered.
12. Select the Actions tab to display the list of all actions to be performed upon the triggering of the event.
13. Click Add to display the list of available actions. Refer to “Working with Actions” in Chapter 5, “Using Actions,” for action procedures.

**Figure 9-9 Add Action Dialog Box**



14. Use the Notes tab to record notes about the event.

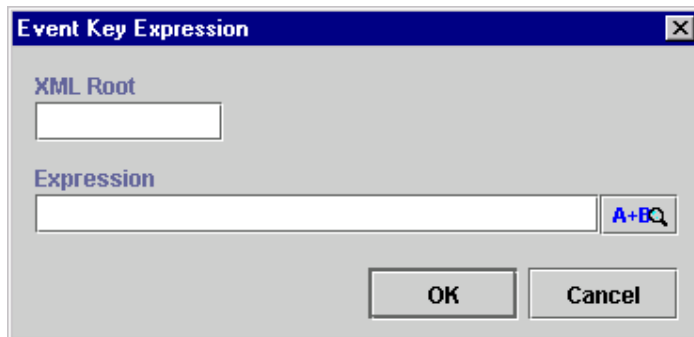
## Setting Up a Key Value for Each XML Root

A key value acts as a filter for locating the correct trigger position to a specific root element. A key value can be set to a variable, for instance `OrderId`. At run time, the XML processor looks for a trigger with the root element `Order` and the specified `OrderId` that will be the value of that variable at the time the workflow is run. This avoids a sequential search of all triggers in order to find the right one.

To set up a key value for each XML root, proceed as follows:

1. From the WebLogic Process Integrator Configuration menu, select Events to display the Event Key Expression dialog box.

**Figure 9-10** Event Key Expressions Dialog Box



2. In the XML Root field, enter the XML document root.
3. In the Expression field, set up a key value for the XML root.

**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 6, "Using Expressions and Conditions."

4. Click OK to confirm your entries.

# Using XML Actions

This section introduces you to the following XML-related actions within WebLogic Process Integrator:

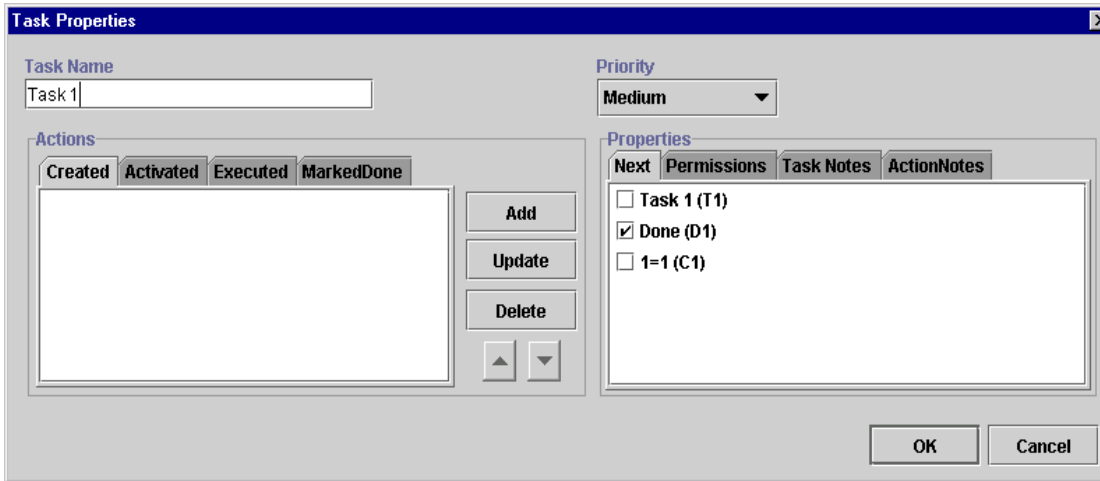
- Post XML Event
- Send XML to Client
- XML Root Element Names

## Post XML Event

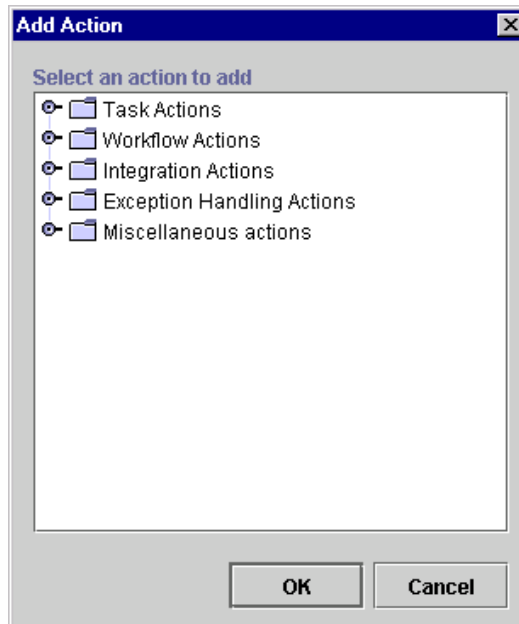
Post an XML event from the current workflow that can be used to trigger another workflow or an event that is contained within this workflow or another workflow. You can post the XML document as either an external document or as internal messages to the WebLogic Process Integrator server.

To post an XML event, proceed as follows:

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

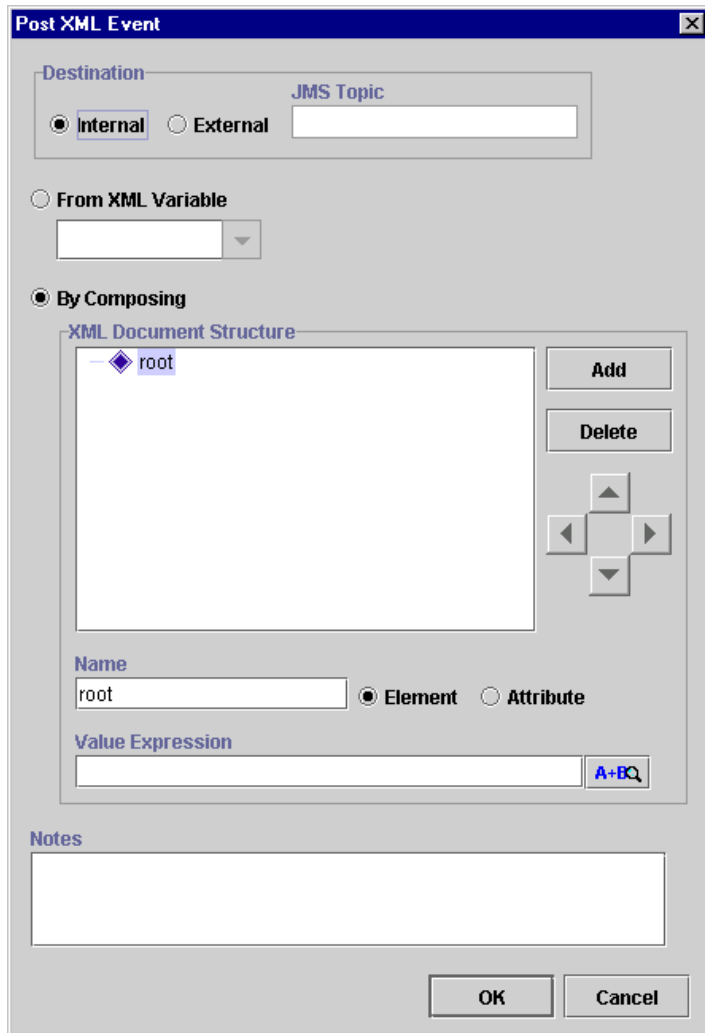
**Figure 9-11 Task Properties Dialog Box**

2. Click Add to display the Add Action dialog box.

**Figure 9-12 Add Action Dialog Box**

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

**Figure 9-13 Post XML Event Dialog Box**



The dialog box is titled "Post XML Event" and contains the following sections:

- Destination:** Includes radio buttons for "Internal" (selected) and "External", and a "JMS Topic" text field.
- From XML Variable:** Includes a radio button and a dropdown menu.
- By Composing:** Includes a radio button (selected) and an "XML Document Structure" section.
  - XML Document Structure:** A tree view showing a root element labeled "root".
  - Buttons:** "Add" and "Delete" buttons are located to the right of the tree view.
  - Navigation:** A set of four arrow buttons (up, down, left, right) is located to the right of the tree view.
- Name:** A text field containing "root".
- Element/Attribute:** Radio buttons for "Element" (selected) and "Attribute".
- Value Expression:** A text field with a "A+B" button to its right.
- Notes:** A large text area for additional notes.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

4. Select the destination where the XML document will be posted; if External is checked, the created XML document is posted on a specified JMS topic. The filed JMS topic is displayed, allowing you to enter the name of the topic. If Internal is checked, the XML message is used to trigger another workflow or an WebLogic Process Integrator event defined within a workflow.
5. If external, the XML document will be posted on a specified JMS topic. The filed JMS topic will open-up for the user to type the in the name of the topic.
6. If the XML document to be posted is stored in a variable of type XML, select this variable from the From XML Variable drop-down list. (The XML type variable is defined in the Set Variable action in “Timed Event” in Chapter 5, “Using Actions.”)
7. If you select an External destination, in the JMS Topic field, enter the name of the JMS topic to which the XML document will be posted. Normally, this is defined for the external client that is listening to receive the message.
8. Use the remaining portion of the Post XML Event dialog box to create an XML structure containing elements and attributes, as follows:
  - a. Select either the Element or Attribute radio button, depending on what you are adding to the XML document structure.
  - b. In the Name field, enter the name of the element or attribute, and click Add to add it to the XML document.
  - c. Use the directional arrows to the right to move the element or attribute forward or backwards, up or down, depending on where you need to position it in the structure. Moving an element or attribute forward will make it a subelement of the previous main node.

**Figure 9-14 Subelement**

**Post XML Event**

**Destination**

☒ Internal ☐ External **JMS Topic**

☐ **From XML Variable**

☒ **By Composing**

**XML Document Structure**

root  
newnode  
newnode

**Add**  
**Delete**

**Name**  
root ☒ **Element** ☐ **Attribute**

**Value Expression**  
 **A+B**

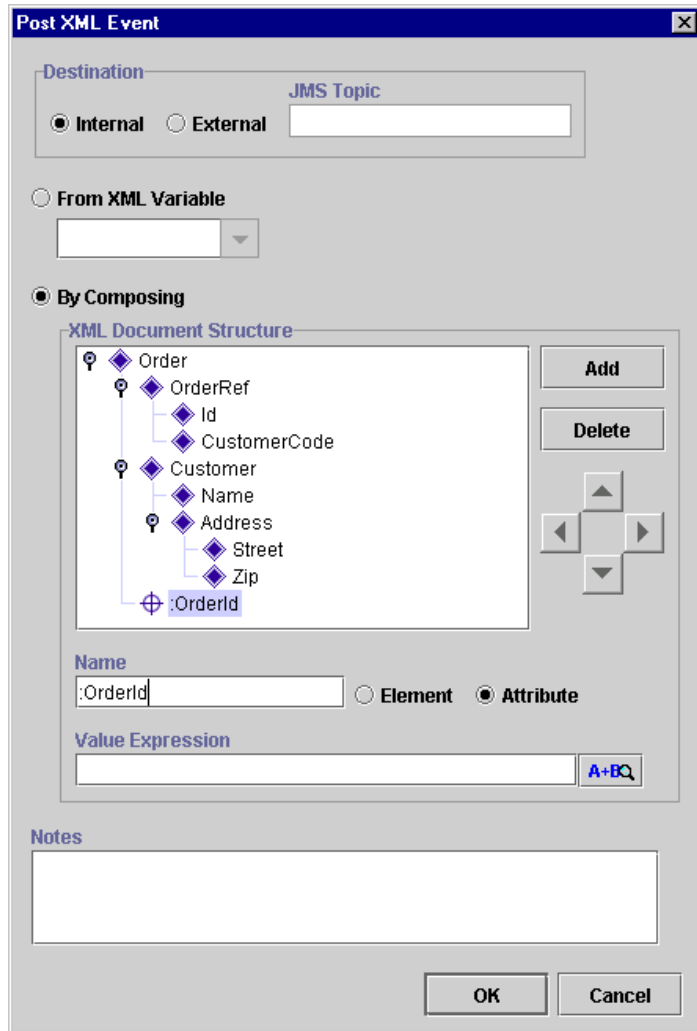
**Notes**

**OK** **Cancel**

- d. Use the Value Expression field to define the value of an element or attribute. The value can be either a string, typed in quotes, or an expression containing workflow variables and functions. For example, in the following figure, the attribute ID is being assigned the value of the workflow variable: OrderId.

**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 6, “Using Expressions and Conditions.”

Figure 9-15 Post XML Event Dialog Box



The dialog box is titled "Post XML Event". It contains several sections:

- Destination:** Includes radio buttons for "Internal" (selected) and "External", and a text field labeled "JMS Topic".
- From XML Variable:** Includes a radio button and a dropdown menu.
- By Composing:** Includes a radio button (selected) and an "XML Document Structure" tree view. The tree shows a hierarchy: Order (root) -> OrderRef -> Id -> CustomerCode -> Customer -> Name -> Address -> Street -> Zip. The "OrderId" node is selected at the bottom.
- Name:** Includes a text field with ":OrderId" and radio buttons for "Element" and "Attribute" (selected).
- Value Expression:** Includes a text field and a button labeled "A+BQ".
- Notes:** Includes a large text area.
- Buttons:** Includes "Add", "Delete", "OK", and "Cancel".

- e. Click OK to post the XML event.

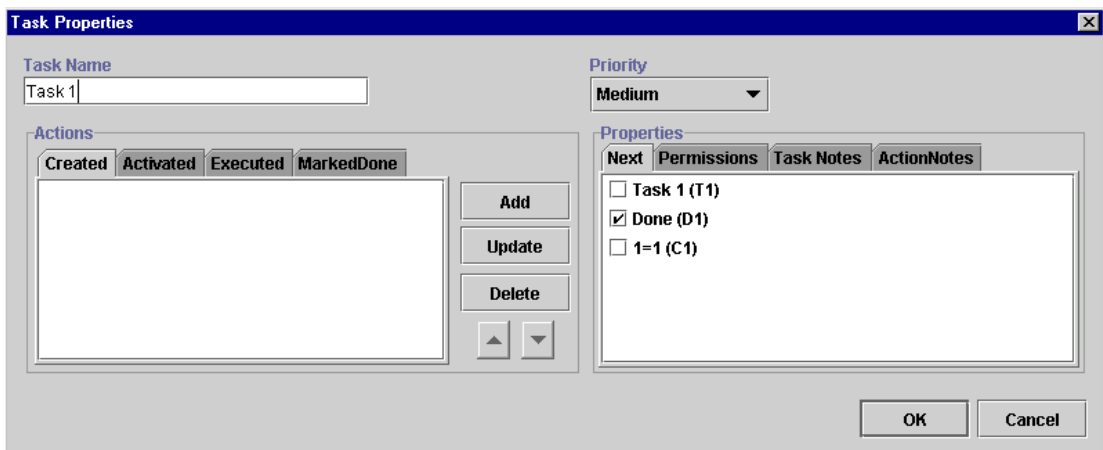
## Send XML to Client

This action allows communication between the WebLogic Process Integrator server and a client program by way of an XML message. The Send XML to Client action expects a response back from the client. The WebLogic Process Integrator Worklist client application is a good example of use for this action.

To send an XML document to a client, proceed as follows:

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

**Figure 9-16 Task Properties Dialog Box**



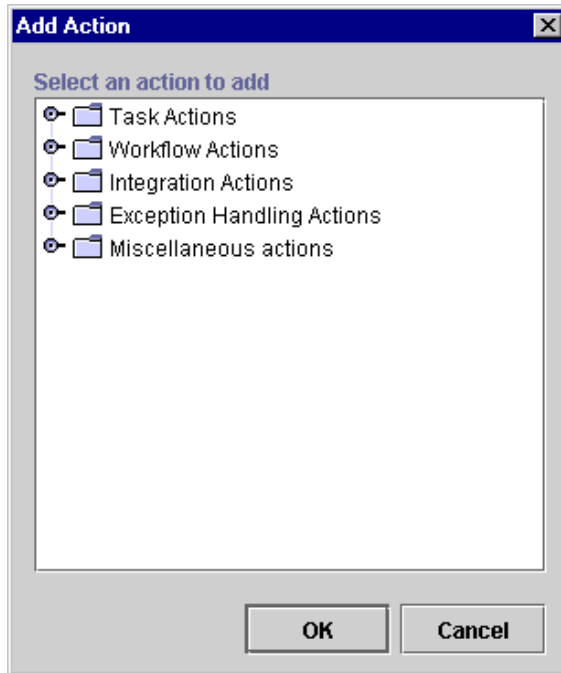
The Task Properties dialog box is a window with a blue title bar and a close button. It contains several sections:

- Task Name:** A text field containing "Task 1".
- Priority:** A dropdown menu set to "Medium".
- Actions:** A section with four tabs: "Created", "Activated", "Executed", and "MarkedDone". Below the tabs is a large empty text area. To the right of the text area are three buttons: "Add", "Update", and "Delete", followed by two arrow buttons (up and down).
- Properties:** A section with four tabs: "Next", "Permissions", "Task Notes", and "ActionNotes". The "Next" tab is selected, showing a list of items with checkboxes:
  - ☐ Task 1 (T1)
  - ☒ Done (D1)
  - ☐ 1=1 (C1)

At the bottom right of the dialog are "OK" and "Cancel" buttons.

2. Click Add to display the Add Action dialog box.

**Figure 9-17 Add Action Dialog Box**



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

- XML Document Structure — create an XML structure with elements, subelements, and attributes. Values can be entered for each element or attribute of the XML document. Refer to “XML Root Element Names.”

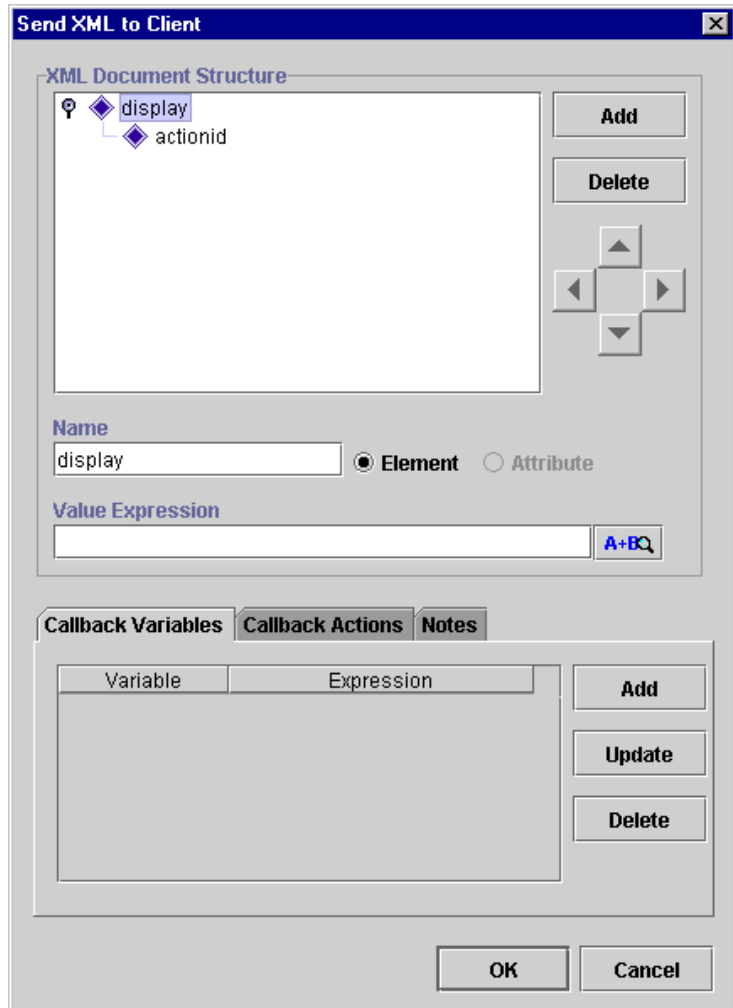
**Note:** The instructions for creating an XML document structure in “Post XML Event” can be applied here.

- Name — name of the XML Element or Attribute.
- Value Expression — expression to be evaluated at run time for the XML event.

**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 6, “Using Expressions and Conditions.”

- **Callback Variables** — retrieves callback information, such as user input or information from a database. Provides a method for placing data into an existing workflow variable.
- **Callback Actions** — relates to the Callback Variables feature. When information is returned (based on the callback variable), tells the system to perform certain actions. Normally, you use the Mark Task as Done action here for synchronous messaging.
- **Notes** — a free-form text box for notes about the action.

Figure 9-18 Send XML to Client Dialog Box



The dialog box is titled "Send XML to Client". It contains an "XML Document Structure" section with a tree view showing a root node "display" with a child node "actionid". To the right of the tree are "Add" and "Delete" buttons, and a set of four directional arrow buttons. Below the tree is a "Name" field containing "display", with radio buttons for "Element" (selected) and "Attribute". Below that is a "Value Expression" field with a search icon labeled "A+B". At the bottom, there are three tabs: "Callback Variables", "Callback Actions", and "Notes". The "Callback Variables" tab is active, showing a table with two columns: "Variable" and "Expression". To the right of the table are "Add", "Update", and "Delete" buttons. At the bottom right of the dialog are "OK" and "Cancel" buttons.

| Variable | Expression |
|----------|------------|
|----------|------------|

## XML Root Element Names

The name of the root element in the XML message format defines the processing performed by the client as follows:

**Table 9-1 XML Root Elements**

| XML Root Element Name | Function Performed by Client                                   |
|-----------------------|----------------------------------------------------------------|
| message-box           | A message-box pop-up is displayed.<br>Reply = ok/yes/no/cancel |
| set-variables         | A query-box pop-up is displayed.<br>Reply = name/value pairs   |
| call-addin            | Invokes a custom extension to the Worklist client application. |
| call-program          | Spawns a process on the client machine.                        |

## Default Settings for Send XML to Client Action

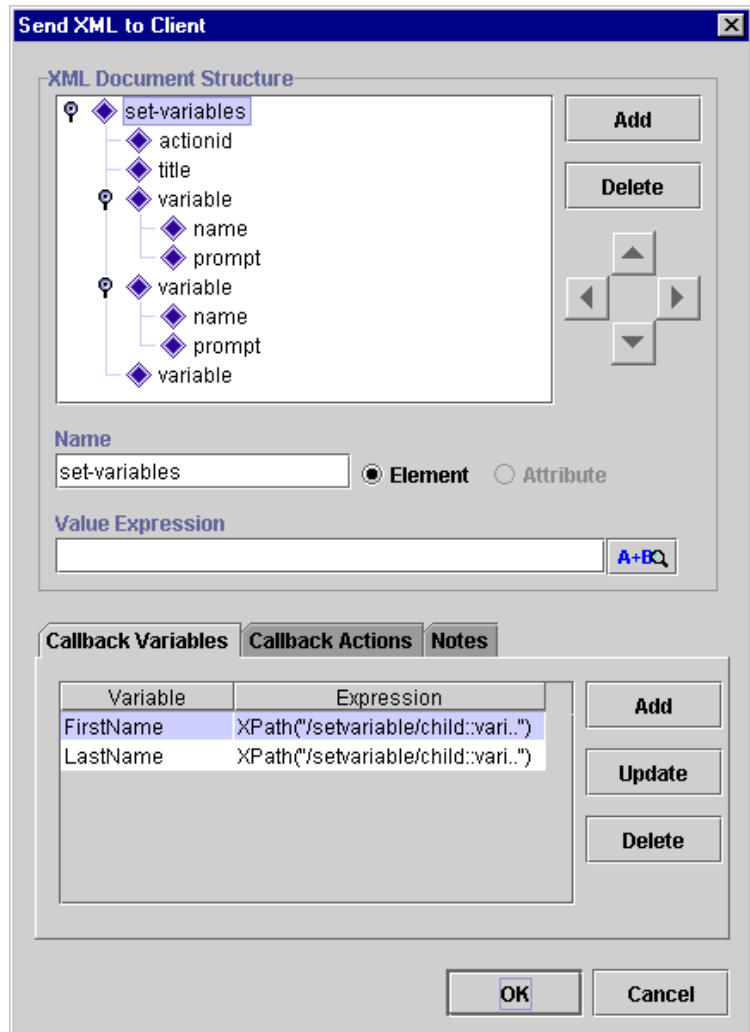
For users who are using the WebLogic Process Integrator Worklist application, you can set up two default settings, which allow you to:

- Prompt for a value upon execution of a task in the WebLogic Process Integrator worklist (displays a prompt dialog box where values can be entered).
- Display a text box upon execution of a task in the WebLogic Process Integrator worklist (displays a text box where values can be entered).

### Dialog Box Prompt

Set up the dialog box prompt default setting shown in the following dialog box and procedure.

**Figure 9-19 Send XML to Client Dialog Box**



To set up the dialog box prompt default setting:

1. Enter the following name for the root element of the XML document:  
`set-variables` (all in lowercase).
2. Set up `title` as an attribute; `title` takes the title value of the dialog box to be displayed.

**Note:** `actionid` cannot be edited.

3. For each variable, create an element root with two attribute as follows:

Variable

Name

Prompt

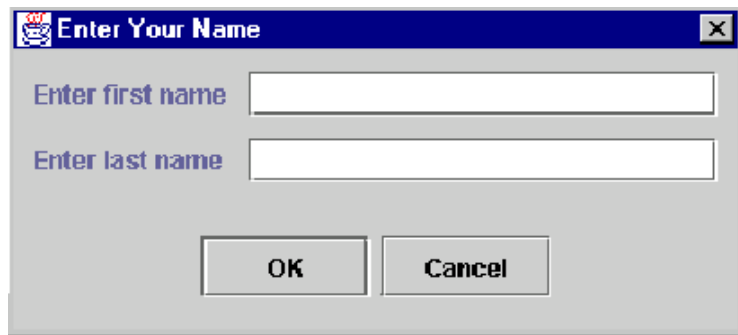
Variable

Name

Prompt

This configuration displays the Enter Your Name dialog box upon execution of a task in the WebLogic Process Integrator Worklist.

**Figure 9-20 Enter Your Name Dialog Box**



After the user enters the values for the first name and last name, the two variables (set up in step 3) are populated by the callback XML document.

### **Display Message-Box**

The default message-box can be set up as one of the following:

- Plain
- Information — provides information to the user.
- Question — asks the user a question.
- Warning — displays a warning to the user.
- Error — indicates an error to the user.

Figure 9-21 Send XML to Client Dialog Box

**Send XML to Client**

**XML Document Structure**

- message-box
  - actionid
  - title
  - style
  - options

**Name**  
 message-box ☒ **Element** ☐ **Attribute**

**Value Expression**  
 Name + " " + :LastName + ".\\nWould you like another go?" [A+BQ](#)

**Callback Variables** **Callback Actions** **Notes**

| Variable | Expression                       |
|----------|----------------------------------|
| option   | XPath("/message-box/attribute... |

**Add** **Update** **Delete**

**OK** **Cancel**

1. Enter the following name for the root element of the XML document:  
message-box (all in lowercase, include the hyphen).
2. Set up title as an attribute; title takes the title value of the message-box to be displayed.

3. Set up `style` as an attribute; selections are plain, information, question, warning, or error.
4. Set up `options` as an attribute; selections are ok, ok\_cancel, yes\_no, yes\_no\_cancel.

To display a message-box, send the following XML:

---

### **Listing 9-2 Message-Box XML**

---

```
<message-box title="title" style="style" options="options">message  
<actionid>959395846210</actionid>  
</message-box>
```

where

```
style ::= { plain | information | question | warning | error }  
options ::= { ok | ok_cancel | yes_no | yes_no_cancel }
```

The message-box returns the following XML:

```
<message-box option="option"></message-box>
```

where

"option" is one of: "ok", "cancel", "yes", "no"

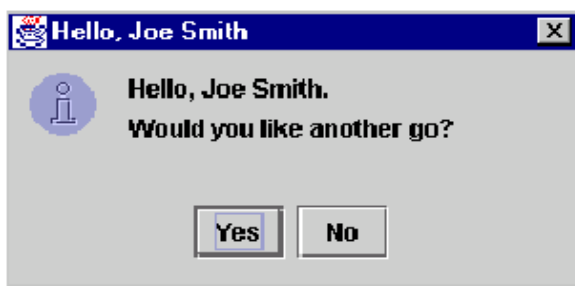
Use the XPath function to extract the value:

```
XPath("/message-box/attribute::option")
```

---

The following is an example of a resulting message-box:

**Figure 9-22 Example Message-Box**



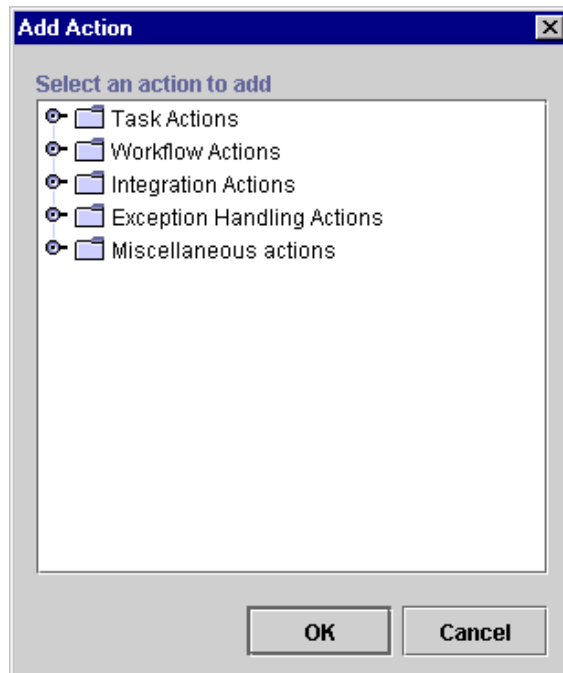
## Invoke Exception Handler

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 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.

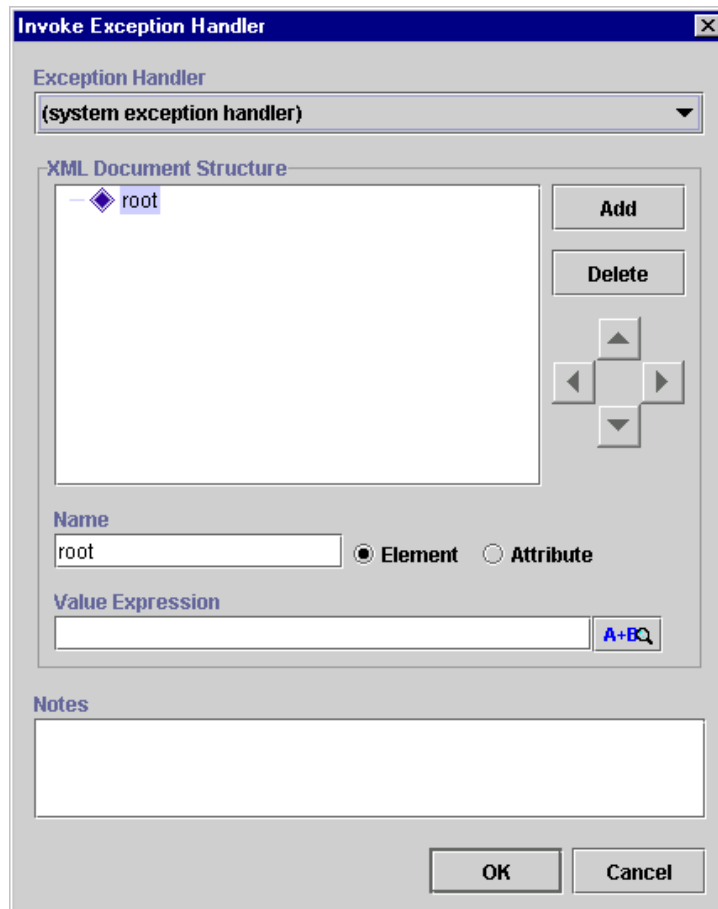
**Figure 9-23 Add Action Dialog Box**



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 9-24 Invoke Exception Handler Dialog Box**



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. Refer to “XML Root Element Names.” 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.

**Note:** The instructions for creating an XML document structure in “Post XML Event” can be applied here.

- **Name** — name of the XML Element or Attribute.
- **Value Expression** — expression to be evaluated at run time for the XML document.

**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 6, “Using Expressions and Conditions.”

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

# Understanding Workflow Interaction

To better understand workflows and how the various components defined within workflows interact, refer to the *WebLogic Process Integrator Tutorial*.

---

# Index

## A

- Abort Workflow action 5-24
- action
  - Abort Workflow action 5-24
  - adding to a start 4-14, 9-8
  - Assign Task to Role action 5-13
  - Assign Task to User action 5-11
  - Assign Task Using Routing Table action 5-14
  - Call Program action 5-32
  - Cancel Workflow Event action 5-46
  - changing the sequence of 5-5
  - decision condition 4-17
  - deleting 5-5
  - Evaluate Condition action 5-40
  - Make Audit Entry action 5-46
  - Mark Task as Done action 5-7
  - No Operation action 5-36
  - Perform Business Operation action 2-41
  - Post XML Event action 5-31
  - Send E-mail Message action 5-37
  - Send XML to Client action 5-31
  - Set Task Comment action 5-18
  - Set Task Due Date action 5-9
  - Set Workflow Comment action 5-30
  - Start Workflow action 5-25
  - Timed Event action 5-42
  - Unassign Task action 5-17
  - Unmark Task Done action 5-8
  - XML actions 9-20
- activated task event 4-20

- activating workflows 3-8
- adding
  - organizations 2-5
  - roles 2-16
- administration
  - organization 2-23
  - role 2-15
  - task routing 2-19
  - user 2-8
- architecture overview 1-2
- Assign Task to Role action 5-13
- Assign Task to User action 5-11
- Assign Task Using Routing Table action 5-14

## B

- business calendar
  - creating 2-24
  - deleting 2-30
  - properties 2-24
  - rules 2-24
  - updating 2-30
- business operations 2-31

## C

- calendar
  - creating 2-24
  - deleting 2-30
  - properties 2-24
  - rules 2-24

---

- updating 2-30
- Call Program action 5-32
- callback actions 9-29
- callback variables 9-29
- called start 4-12
- Cancel Workflow Event action 5-46
- closing
  - workflows 3-13
- components (of workflow) 4-1
- condition 4-15, 6-1
- connector 1-22
- copying workflows 3-17
- created task event 4-20
- creating
  - business calendar 2-24
  - roles 2-16
  - workflow definition 3-4
  - workflow templates 3-3
- customer support contact information xiv

## D

- data administration 1-9
- data type definition (DTD) 9-3
- decision 1-6, 1-21, 4-16
  - actions and conditions 4-17
  - condition 4-15
  - next false 4-18
  - next true 4-18
  - properties 4-15
- deleting
  - actions 5-5
  - business calendar 2-30
  - organizations 2-7
  - roles 2-18
  - users 2-14
  - workflows 8-13
- documentation, where to find it xii
- done 1-6, 1-21, 4-15
- drawing area 1-19
- DTD (data type definition) 9-3

## E

- effective date 3-4, 3-8, 3-25
- EJB (Enterprise JavaBean) 5-2
- enable audit to JMS (Java Message Service)
  - 3-5, 3-8
- Enterprise JavaBean (EJB) 5-2
- error messages 7-13
- Evaluate Condition action 5-40
- event 1-6, 1-21, 4-22
  - properties 4-22
  - trigger definition 9-5
  - XML notification 9-14
- exception handler
  - error messages 7-13
- exception handlers 7-1
- exceptions 7-1
- execution schedule 5-43
- Exit Exception Handler action 7-12
- expiry date 3-4, 3-8, 3-25
- exporting workflows 3-26
- expression 6-1
  - Expression Builder 6-3
- Expression Builder 6-3
  - expression components 6-6
  - invalid expression messages 6-4
- expression components
  - functions 6-6
  - literals 6-21
  - operators 6-19
  - variables 6-22

## F

- functions
  - Abs 6-6
  - Date 6-6
  - DateAdd 6-7
  - DateAdd interval values 6-7, 6-8, 6-11, 6-12
  - DateToString 6-8
  - StringLen 6-10

---

- StringToDate 6-10
- SubString 6-10
- TaskAttribute 6-11
- WorkflowAttribute 6-12
- WorkflowVariable 6-13
- xPath 6-14

## I

- importing workflows 3-27
- instances 3-29
- integration actions
  - Call Program action 5-32
  - Post XML Event action 5-31
  - Send XML to Client action 5-31
- Invoke Exception Handler action 7-10

## J

- Java Message Service (JMS) 3-5, 3-8
  - posting an audit entry to a topic 5-46
- JMS (Java Message Service) 3-5, 3-8
  - posting an audit entry to a topic 5-46
- join 1-6, 1-21, 4-23

## L

- literals 6-21

## M

- maintaining users 2-8
- Make Audit Entry action 5-46
- manual start 4-12
- Mark Task as Done action 5-7
- marked done task event 4-20
- miscellaneous actions
  - Cancel Workflow Event action 5-46
  - Evaluate Condition action 5-40
  - Make Audit Entry action 5-46
  - No Operation action 5-36
  - Send E-mail Message action 5-37

- Timed Event action 5-42
- monitoring
  - deleting workflows 8-13
  - graphical representation of workflow 8-6
  - list view of workflow 8-7
  - reassignment of tasks 8-9
  - statistics reports 8-17
  - workflow status 8-3
  - workflow status icons 8-5
  - workflow variables 8-11
  - workflows 1-10
  - workload reports 8-14

## N

- next false 4-18
- next true 4-18
- No Operation action 5-36

## O

- operator values 6-19
- organization
  - adding 2-5
  - deleting 2-7
  - properties 2-5
  - updating 2-6

## P

- Perform Business Operation action 2-41
- permissions, task
  - mark done without executing 4-20
  - modifiable 4-21
  - re-assignable 4-21
  - re-execute if marked done 4-21
  - unmark if marked done 4-21
- Post XML Event action 5-31, 9-20
- printing 3-13
- printing product documentation xiii

---

## R

- reassignment of tasks 8-9
- refreshing the task list 2-22
- related information xiii
- role 2-15
  - creating 2-16
  - deleting 2-18
  - updating 2-17
- routing condition definition 5-16
- routing of tasks 2-19
  - adding a specification 2-20
  - deleting a specification 2-22
  - refreshing the task list 2-22
  - updating a specification 2-21
- rules, business calendar 2-24

## S

- saving
  - workflows 3-13
- Send E-mail Message action 5-37
- Send XML to Client action 5-31, 9-27
- server administration 1-9
- Set Task Comment action 5-18
- Set Task Due Date action 5-9
- Set Workflow Comment action 5-30
- Set Workflow Exception Handler action 7-10
- SQL (structured query language)
  - expressions 6-2, 7-13
- start 1-6, 1-21, 4-8
  - adding actions to 4-14, 9-8
  - called option 4-12
  - manual option 4-12
  - properties 4-10
  - timed option 4-11
- Start Workflow action 5-25
- starting
  - WebLogic Process Integrator 1-11
- statistics reports 8-2, 8-17
- status icons, monitoring workflows 8-5
- structured query language (SQL)

- expressions 6-2, 7-13

- support
  - technical xiv

## T

- task 1-6, 1-21, 4-19
  - actions 4-20
    - permissions 4-20
    - priority 4-20
    - properties 4-17, 4-19, 8-10
    - reassignment 8-9
- task actions
  - Assign Task to Role action 5-13
  - Assign Task to User action 5-11
  - Assign Task Using Routing Table action 5-14
  - Mark Task as Done action 5-7
  - Set Task Comment action 5-18
  - Set Task Due Date action 5-9
  - Unassign Task action 5-17
  - Unmark Task Done action 5-8
- task event 4-20
- task routing
  - adding a specification 2-20
  - administration 2-19
  - deleting a specification 2-22
  - refreshing the task list 2-22
  - updating a specification 2-21
- Timed Event action 5-42
  - execution schedule 5-43
- timed start 4-11
- triggering
  - actions triggered by Timed Event action 5-44
  - adding actions to trigger in a workflow 4-14, 9-8
  - called option 4-12
  - defining a workflow event trigger 9-5
  - manual option 4-12
  - Timed Event action 5-42

---

- timed option 4-11
- workflow triggering example 9-10

## U

- Unassign Task action 5-17
- Unmark Task Done action 5-8
- updating
  - business calendar 2-30
  - organizations 2-6
  - roles 2-17
  - users 2-14
  - variables 4-5
- user 2-8
  - deleting 2-14
  - menu options 2-9
  - properties 2-14
  - updating 2-14

## V

- variable 1-7, 6-22
  - assignment in a workflow 4-13, 7-9
  - properties 4-4
  - updating 4-5
  - workflow 8-11
- variables menu 8-11

## W

- WebLogic Process Integrator
  - adding an organization 2-3
  - architecture overview 1-2
  - business operations 2-31
  - closing workflows 3-13
  - copying workflows 3-17
  - creating a business calendar 2-24
  - creating a new workflow template 3-3
  - creating workflow definition 3-4
  - data and server administration 1-9
  - decision 4-16

- defining a workflow event trigger 9-5
- defining business calendar rules 2-24
- deleting organizations 2-7
- deleting roles 2-18
- deleting users 2-14
- done 4-15
- error messages 7-13
- event 4-22
- exiting 1-22
- exporting workflows 3-26
- Expression Builder 6-3
- expression components 6-6
- expressions and conditions 6-1
- importing workflows 3-27
- join 4-23
- roles 2-15
- saving workflows 3-13
- start 4-8
- starting 1-11
- statistics reports 8-17
- task 4-19
- task routing administration 2-19
- updating organizations 2-6
- updating roles 2-17
- updating user properties 2-14
- user 2-8
- workflow components 4-1
- workflow drawing area 1-19
- workflow effective and expiry dates 3-25
- workflow instances 3-29
- workflow monitoring 1-10
- workflow properties 3-6
- workflow shapes 1-21
- workflow status 8-3
- workflow variables 8-11
- working with actions 5-1
- workload reports 8-14
- XML events and documents 9-1
- XML notification, WebLogic Process Integrator event 9-14

---

WebLogic Process Integrator event 9-1

workflow

- activating 3-8
- adding actions to 4-14, 9-8
- changing the sequence of actions 5-5
- closing 3-13
- components 4-1
- connector 1-22
- copying 3-17
- creating 3-4
- decision 1-21, 4-16
- decision condition 4-15
- defining a workflow event trigger 9-5
- deleting 8-13
- deleting an action 5-5
- done 1-21, 4-15
- drawing area 1-19
- effective and expiry date 3-25
- effective date 3-5, 3-8
- enable audit to JMS 3-5
- enable audit to JMS topic 3-8
- event 1-21, 4-22
- exception handlers 7-1
- expiry date 3-5, 3-8
- exporting 3-26
- expressions and conditions 6-1
- Id 3-8
- join 1-21, 4-23
- monitoring 1-10
- monitoring status icons 8-5
- name 3-25
- properties 3-6
- reassignment of tasks 8-9
- saving 3-13
- setting up a key value for each XML root 9-19
- start 1-21, 4-8
- statistics reports 8-17
- status 8-3
- task 1-21, 4-19
- task event 4-20

triggering example 9-10

updating variables 4-5

variable assignment 4-13, 7-9

variables 8-11

working with actions 5-1

XML notification, WebLogic Process  
Integrator event 9-14

workflow actions

Abort Workflow action 5-24

Set Workflow Comment action 5-30

Start Workflow action 5-25

workflow definition

printing 3-13

workflow template

creating 3-3

definition 3-6

drawing area 1-19

workload reports 8-2, 8-14

## X

XML

actions 9-20

attributes 9-2

documents

- setting up a key value for each XML  
root 9-19

elements 9-2

events and documents 9-1

notification 9-14

overview 9-2

posting an XML document to the  
WebLogic Process Integrator  
server 9-20

root 9-19

sending an XML document to a client  
9-27

setting up a key value for each XML root  
9-1

xPath 6-14