11i/2.6 Implement Oracle Workflow

Volume 1 - Student Guide

D13401GC10
Edition 1.0
February 2002
D34084

ORACLE

Author

Clara Jaeckel

Technical Contributors and Reviewers

Mark Craig, Bill Curtis, Leta Davis, Kevin Hudson, Adam Laro-Bashford, Helena Martinovic

This book was published using: **oracle*tutor***

Table of Contents

Preface

Profile

Before You Begin This Course

Before you begin this course, you should have the following qualifications:

- Thorough knowledge of Oracle8*i* or Oracle9*i* Database Server and Oracle9*i* Application Server technology
- If you plan to use Oracle Workflow with Oracle E-Business Suite, thorough knowledge of Oracle E-Business Suite
- Working knowledge of XML

Prerequisites

- *Introduction to Oracle9i: SQL*
- *Oracle9i: Program with PL/SQL*

How This Course Is Organized

11i/2.6 Implement Oracle Workflow is an instructor-led course featuring lecture and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills introduced.

Related Publications

Oracle Publications

Title	Part Number
Oracle Workflow Guide Release 2.6.2	*A95625*

Additional Publications

- System release bulletins
- Installation and user's guides
- *read.me* files
- *Oracle Magazine*

Typographic Conventions

Typographic Conventions in Text

Convention	Element	Example
Bold italic	Glossary term (if there is a glossary)	The ***algorithm*** inserts the new key.
Caps and lowercase	Buttons, check boxes, triggers, windows	Click the Executable button. Select the Can't Delete Card check box. Assign a When-Validate-Item trigger to the ORD block. Open the Master Schedule window.
Courier new, case sensitive (default is lowercase)	Code output, directory names, filenames, passwords, pathnames, URLs, user input, usernames	Code output: `debug.set ('I", 300);` Directory: `bin` (DOS), `$FMHOME` (UNIX) Filename: Locate the `init.ora` file. Password: User `tiger` as your password. Pathname: Open `c:\my_docs\projects` URL: Go to `http://www.oracle.com` User input: Enter `300` Username: Log on as `scott`
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address (*but* Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	Do *not* save changes to the database. For further information, see *Oracle7 Server SQL Language Reference Manual*. Enter `user_id@us.oracle.com`, where *user_id* is the name of the user.
Quotation marks	Interface elements with long names that have only initial caps; lesson and chapter titles in cross-references	Select "Include a reusable module component" and click Finish. This subject is covered in Unit II, Lesson 3, "Working with Objects."
Uppercase	SQL column names, commands, functions, schemas, table names	Use the SELECT command to view information stored in the LAST_NAME column of the EMP table.

Convention	Element	Example
Arrow	Menu paths	Select File—> Save.

Brackets	Key names	Press [Enter].
Commas	Key sequences	Press and release keys one at a time: [Alternate], [F], [D]
Plus signs	Key combinations	Press and hold these keys simultaneously: [Ctrl]+[Alt]+[Del]

Typographic Conventions in Code

Convention	Element	Example
Caps and lowercase	Oracle Forms triggers	`When-Validate-Item`
Lowercase	Column names, table names	`SELECT last_name FROM s_emp;`
	Passwords	`DROP USER scott IDENTIFIED BY tiger;`
	PL/SQL objects	`OG_ACTIVATE_LAYER (OG_GET_LAYER ('prod_pie_layer'))`
Lowercase italic	Syntax variables	`CREATE ROLE role`
Uppercase	SQL commands and functions	`SELECT userid FROM emp;`

Typographic Conventions in Navigation Paths

This course uses simplified navigation paths, such as the following example, to direct you through Oracle Applications.

(N) Invoice > Entry > Invoice Batches Summary (M) Query > Find (B) Approve

This simplified path translates to the following:

1. (N) From the Navigator window, select Invoice > Entry > Invoice Batches Summary.
2. (M) From the menu, select Query > Find.
3. (B) Click the Approve button.

Notations :

(N) = Navigator

(M) = Menu

(T) = Tab

(I) = Icon

(H) = Hyperlink

(B) = Button

Typographical Conventions in Help System Paths

This course uses a “navigation path” convention to represent actions you perform to find pertinent information in the Oracle Applications Help System.

The following help navigation path, for example—

(Help) General Ledger > Journals > Enter Journals

—represents the following sequence of actions:

1. In the navigation frame of the help system window, expand the General Ledger entry.
2. Under the General Ledger entry, expand Journals.
3. Under Journals, select Enter Journals.
4. Review the Enter Journals topic that appears in the document frame of the help system window.

Getting Help

Oracle Applications provides you with a complete online help facility.

Whenever you need assistance, simply choose an item from the Help menu to pinpoint the type of information you want.

To display help for a current window:

1. Choose Window Help from the Help menu, click the Help button on the toolbar, or hold down the Control key and type 'h'.

   A web browser window appears, containing search and navigation frames on the left, and a frame that displays help documents on the right.

   The document frame provides information on the window containing the cursor. The navigation frame displays the top-level topics for your responsibility, arranged in a tree control.

2. If the document frame contains a list of topics associated with the window, click on a topic of interest to display more detailed information.
3. You can navigate to other topics of interest in the help system, or choose Close from your web browser's File menu to close help.

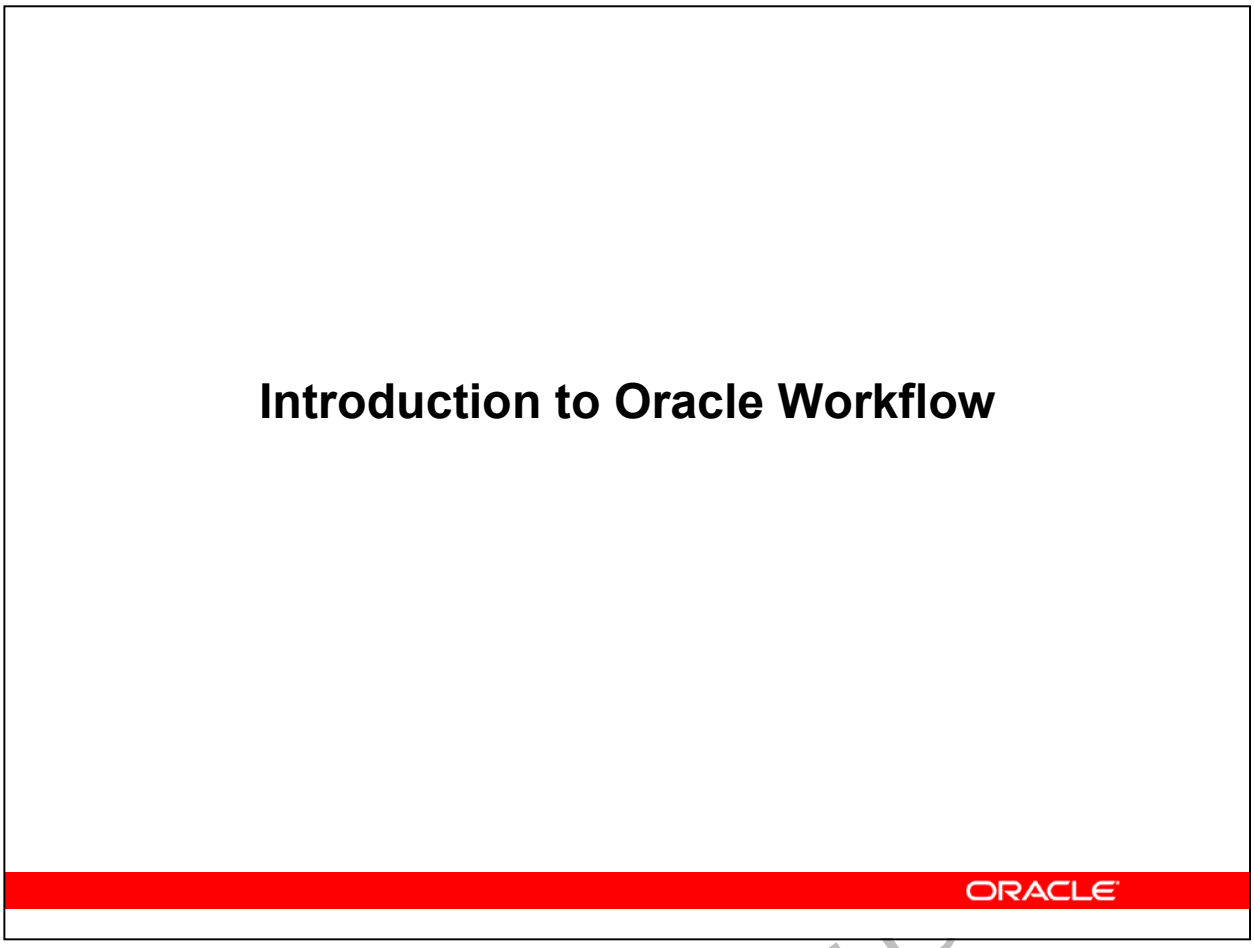
Searching for Help

You can perform a search to find the Oracle Applications help information you want. Simply enter your query in the text field located in the top-left frame of the browser window when viewing help, then click the adjacent Find button.

A list of titles, ranked by relevance and linked to the documents in question, is returned from your search in the right-hand document frame. Click on whichever title seems to best answer your needs to display the complete document in this frame. If the document doesn't fully answer your questions, use your browser's Back button to return to the list of titles and try another.

Introduction to Oracle Workflow

Chapter 1



Objectives

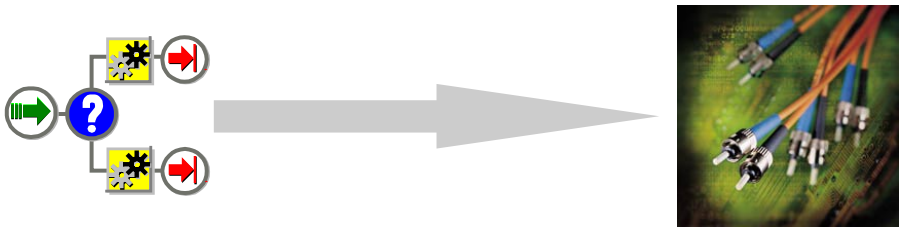
After completing this lesson, you should be able to do the following:

- **Explain the benefits of Oracle Workflow.**
- **Discuss the concept of a workflow process.**
- **Discuss the concept of business events.**

ORACLE

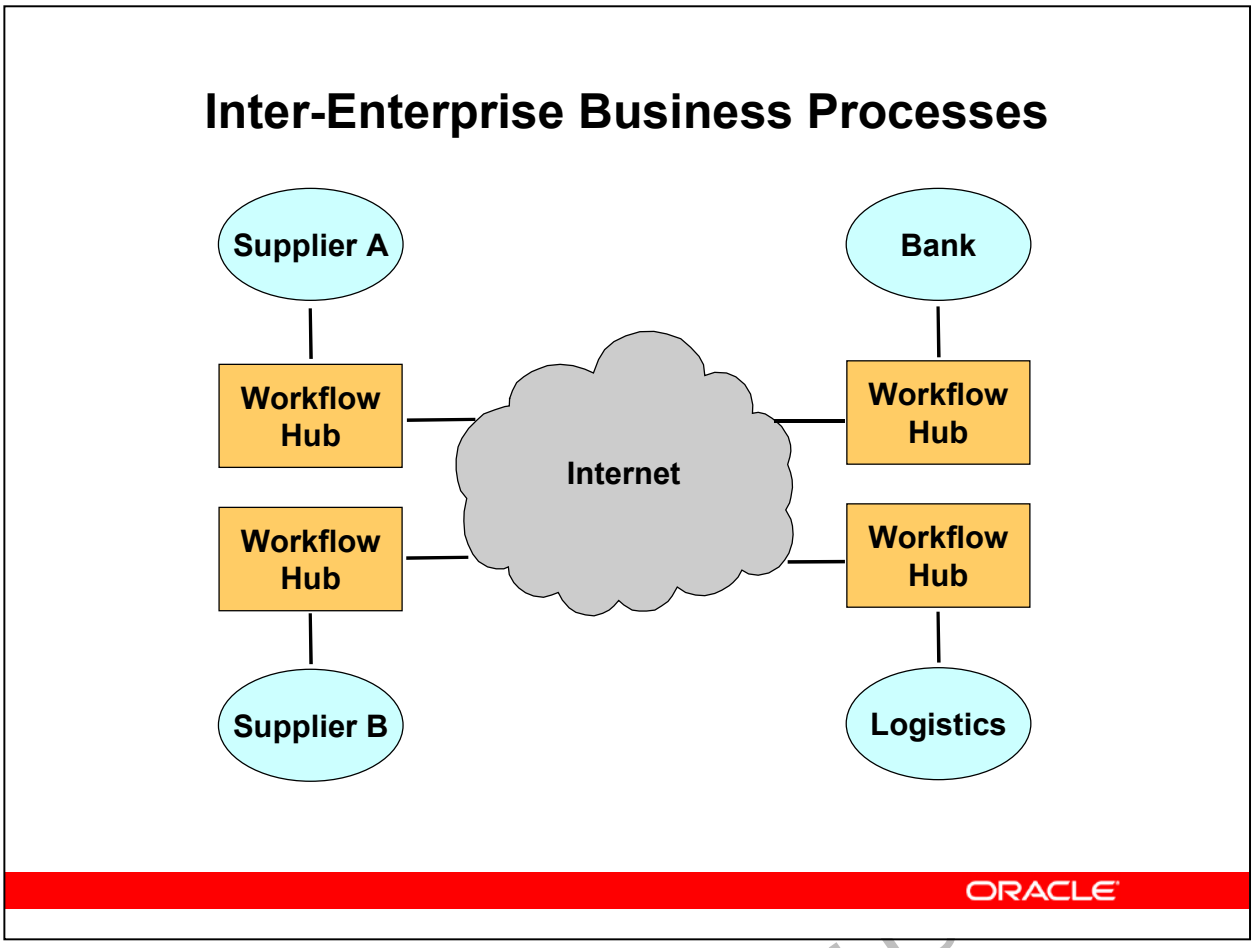
Enabling E-Business

Streamlined business processes play a critical role in the transformation to e-business. Oracle Workflow delivers a complete business process definition, automation, and integration solution.



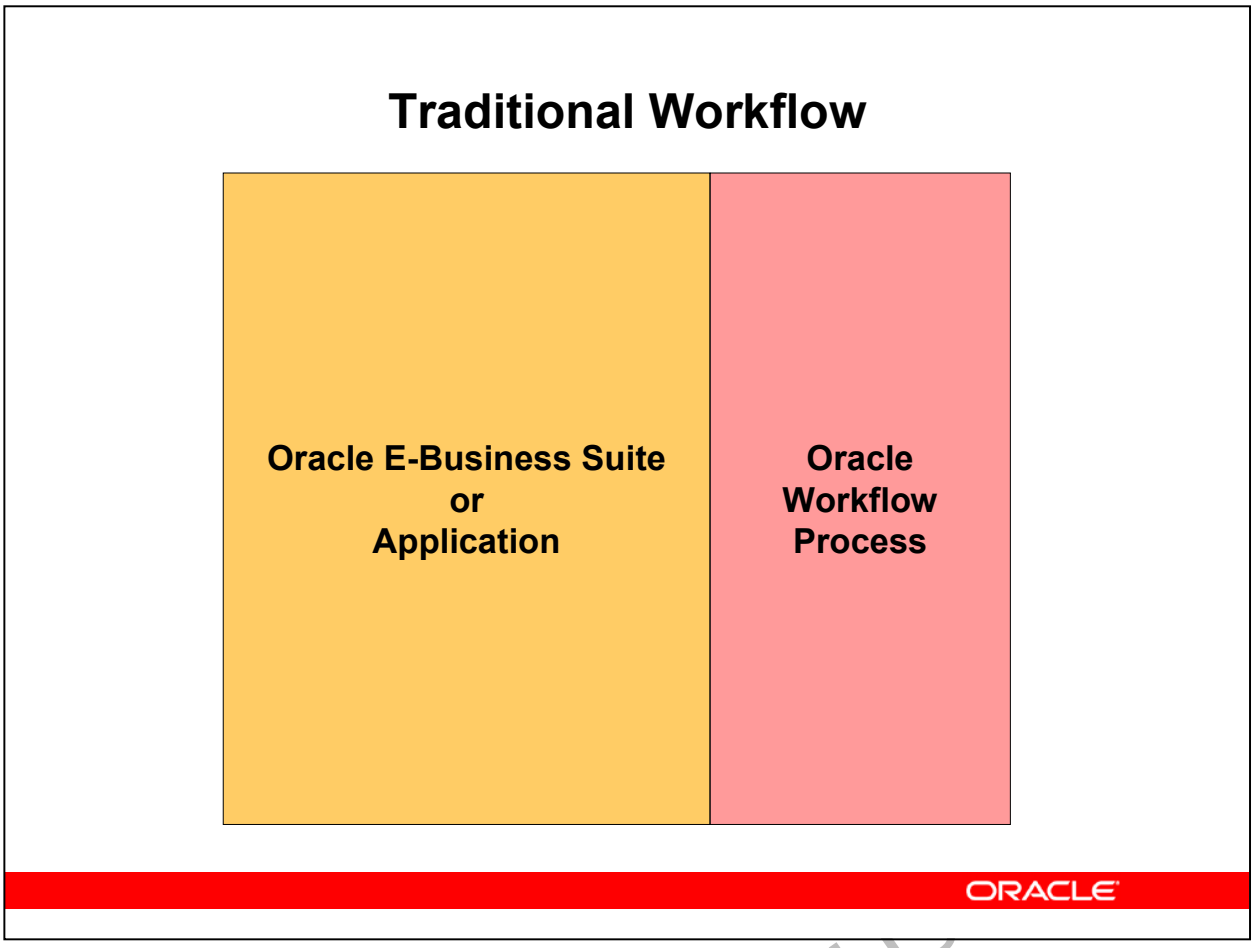
Oracle Workflow: The wiring for e-business

ORACLE



Inter-Enterprise Business Processes

In e-business, different enterprises need to communicate with each other over the Internet. Oracle Workflow with the Business Event System can model business processes that span all the enterprises involved in an end-to-end process.

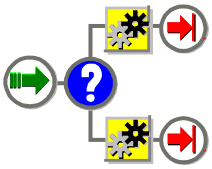


Traditional Workflow

Traditional applications-based workflow processes are launched from a business application through APIs hard-coded within the application. These processes model the business rules in the individual local application and are made up of activities executed by the Workflow Engine only in that application's system. For example, the modeling of an approval hierarchy is a common use of Oracle Workflow in this scenario.

Workflow-Driven Business Processes

- **Oracle Workflow automates and streamlines business processes contained within and between enterprises.**
- **For example, you can use workflow processes to:**
  - **Add personalized trading partner rules**
  - **Validate self-service transactions**
  - **Achieve closed loop business intelligence**
  - **Approve standard business documents**
  - **Step through daily transaction flows**
  - **Integrate with trading partner systems**

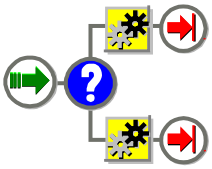


ORACLE

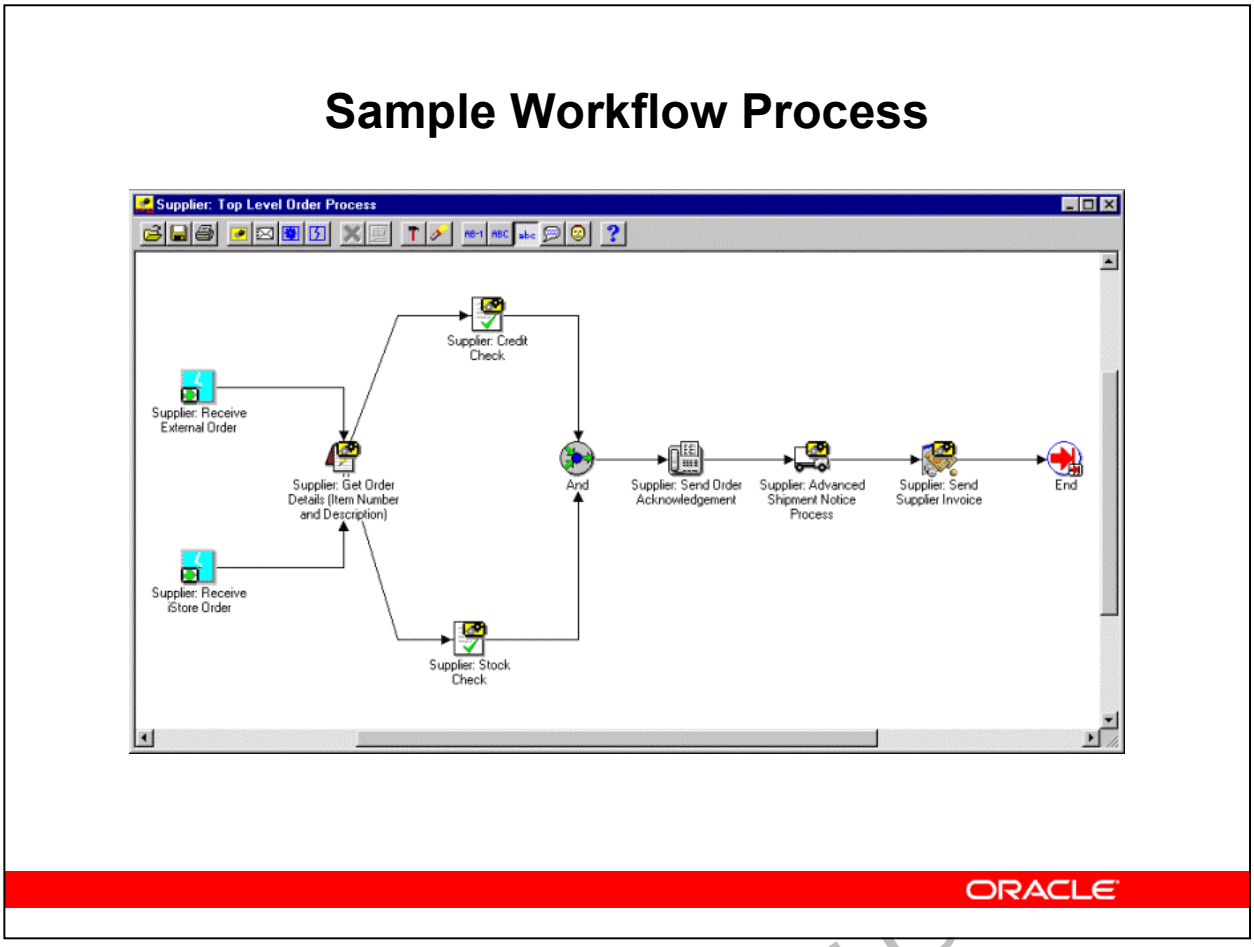
Workflow-Driven Business Processes

Oracle Workflow lets you focus on managing the business process, not individual transactions.

- **Define and implement your business policies**
- **Streamline the entire process**
- **Route information**
- **Capture exceptions and take action**
- **Build continuous improvements directly into the process definition**
- **Adapt your processes as your business changes**



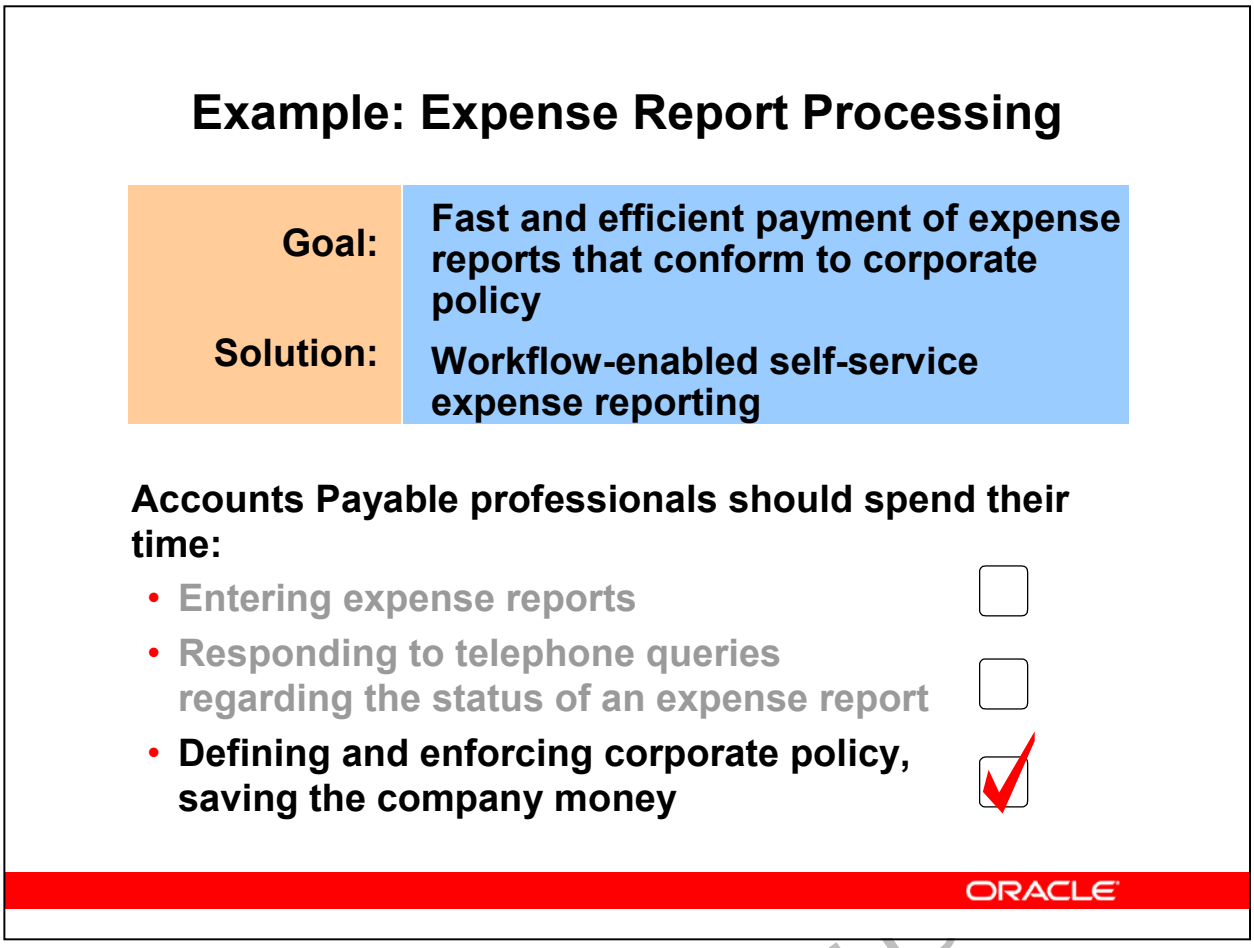
ORACLE



Sample Workflow Process

A workflow process consists of a sequence of activities that together make up a business flow, expressing your organization's policies and rules. The activities can include significant business events, automated functions, notifications to users, or subprocesses.

This example shows a sample order processing workflow process that includes business events. The example comes from the Event System Demonstration workflow which is available with the standalone version of Oracle Workflow.



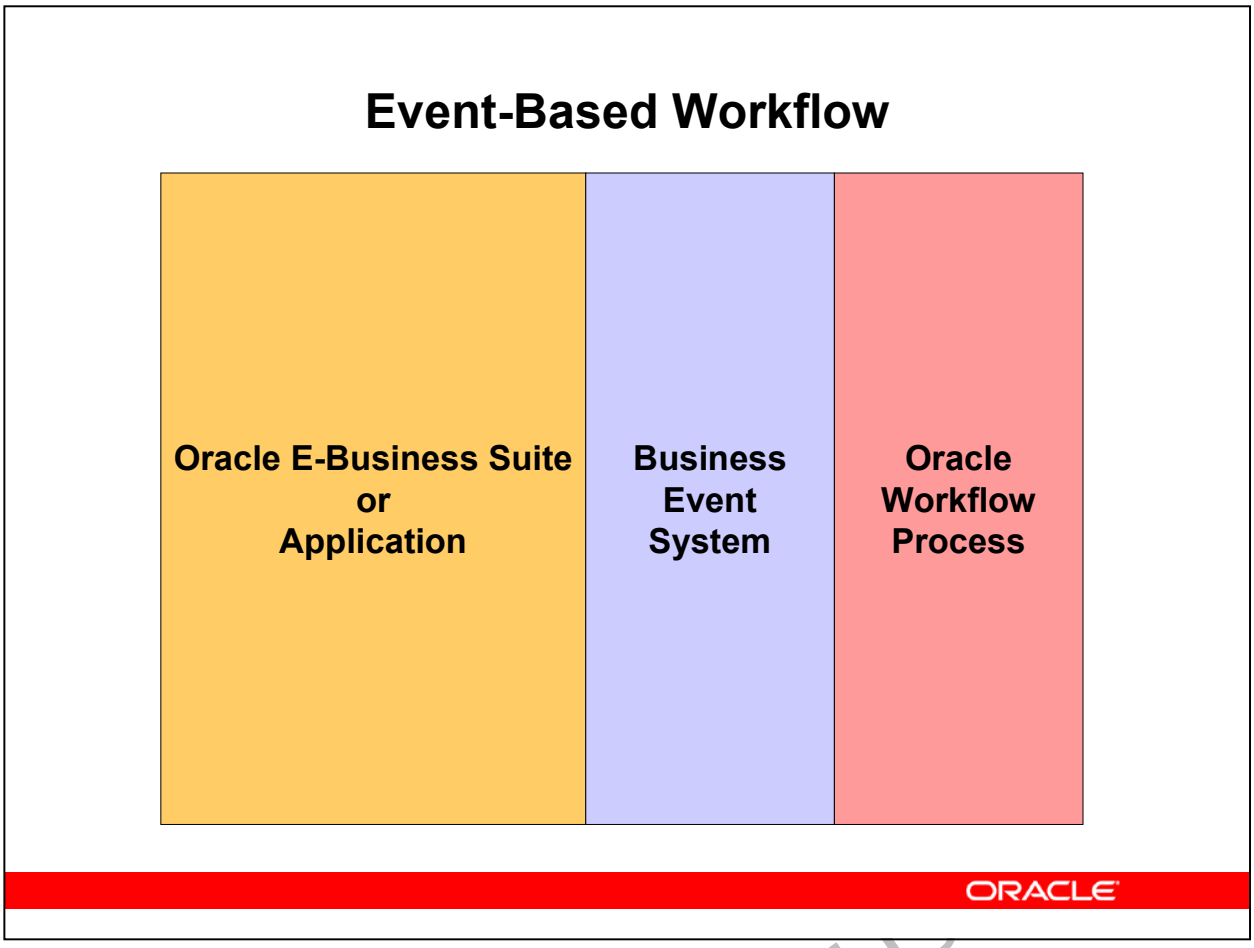
Example: Expense Report Processing

Oracle Workflow can help save time in expense report processing by reducing repetitive data entry tasks and by providing self-service monitoring capabilities.

Example: New Hire Processing

Oracle Workflow can help save time in new hire processing by automating approval hierarchies as well as automatically delivering notifications and reminders of work to be done.

Event-Based Workflow



Event-Based Workflow

With the Business Event System, Oracle Workflow supports both traditional applications-based workflows and event-based integration workflows.

For e-business, there is a requirement to integrate with external systems, such as sending a document to a business-to-business exchange, or other systems external to the local application. Oracle Workflow supports e-business integration workflows by allowing business analysts and developers to model business processes spanning different systems using a graphical drag-and-drop designer - the Workflow Builder - and run those processes using the Workflow Engine and the Business Event System. This support allows Oracle Workflow customers to deal with business objects in comprehensive e-business integration flows, with minimal intrusion into the core application.

The Business Event System and the Workflow Engine can function independently of each other. However, you can achieve the most powerful and flexible processing by using the Business Event System and the Workflow Engine together to execute cross-system processes for e-business integration.

Subscription-Based Processing

In the Oracle Workflow Business Event System:

- **Business events in applications trigger event subscriptions in Oracle Workflow.**
- **Subscriptions can launch workflow processes or perform other processing.**
- **Multiple subscriptions can be defined to perform different processing for the same event.**
- **Subscriptions can be enabled, modified, or disabled as necessary without intruding into applications.**

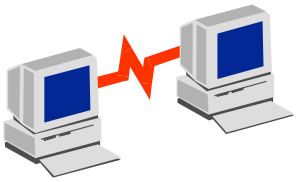
ORACLE

Subscription-Based Processing

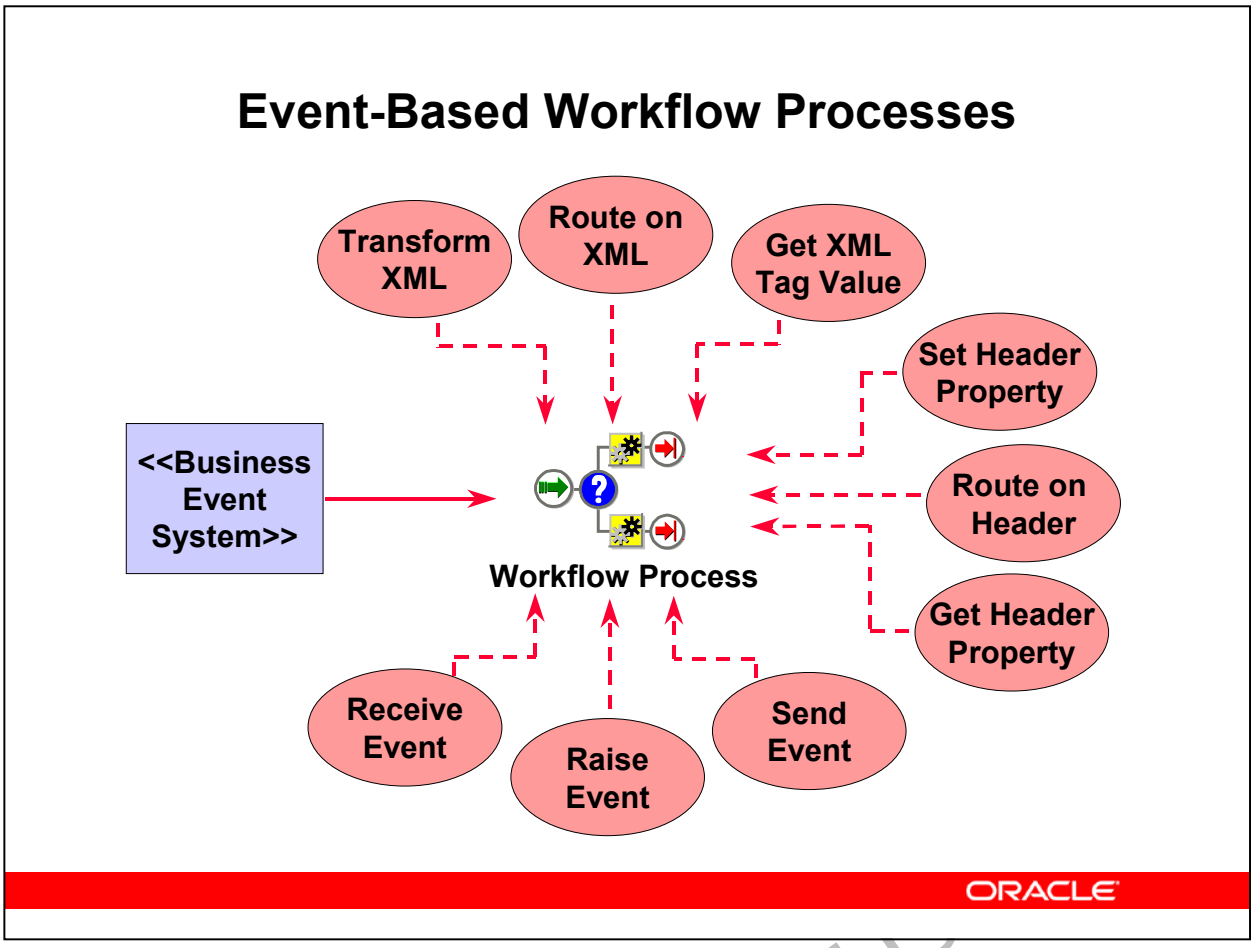
The Business Event System provides increased flexibility through subscription-based processing: you raise a business event from an application, but specify the processing to perform for that event as a subscription in Oracle Workflow. For example, you can launch a workflow process when an event is raised by specifying that process in a subscription to the event. You can also define multiple subscriptions to the same event to perform additional processing for different purposes, without intruding any further on the core application.

System Integration with Oracle Workflow

- **E-business accelerates the demand for system integration.**
- **Communication is required between systems both within and beyond the enterprise.**
- **Oracle Workflow supports e-business integration workflows through the Business Event System.**
- **Business event-based workflows allow modeling of cross-system processes, enabling business process-based integration.**



ORACLE

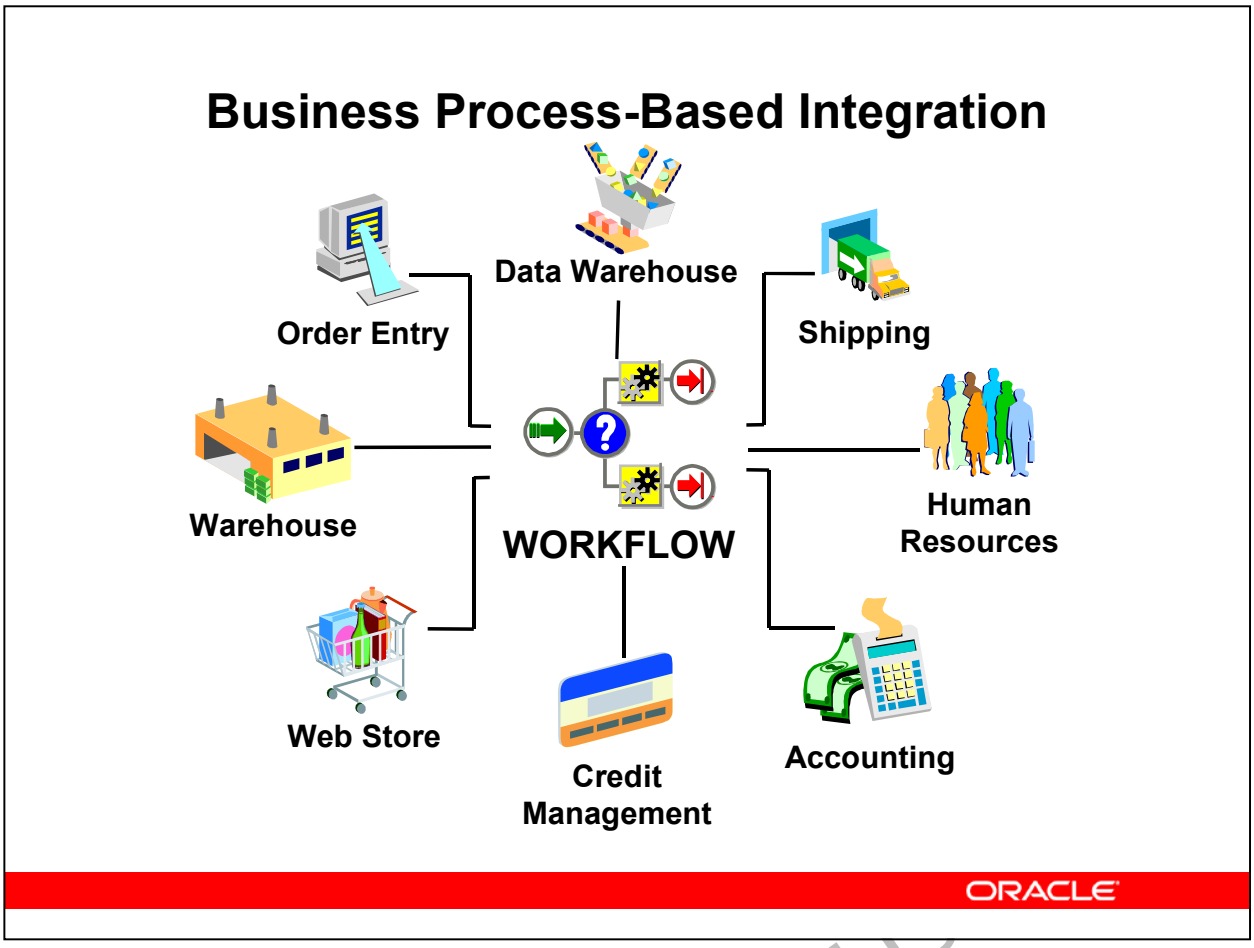


Event-Based Workflow Processes

Event-based workflow processes control and route objects between applications according to business rules. These workflow processes support:

- Receiving business events to launch or continue processes
- Raising new business events
- Sending business event messages for inter-system communication
- Accessing and routing on header properties of event messages
- Accessing and routing on XML content within event messages
  (**Note:** This functionality is currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.)

By letting you model processes across different systems, event-based workflows enable business process-based integration.



Business Process-Based Integration

Business process-based integration is model-driven.

- Business rules are expressed in a process model.
- These rules define the policy for each end-to-end process.
- The process model can encompass applications both within and beyond the enterprise.

Business process-based integration provides:

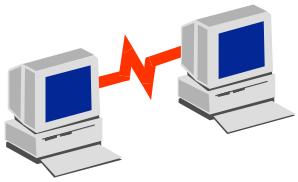
- A global, enterprise-level view of business objects
- Business process automation

Oracle Workflow supports business process-based integration through the Business Event System.

Supported System Integration Types

The Business Event System supports integration in which applications are loosely coupled through asynchronous messaging.

- **Point-to-point system integration—"Hardwired" communication between specified systems**
- **Messaging hub system integration—Intersystem communication routed through a central hub for more complex integration scenarios**
- **Distributed applications messaging—Master/copy replication of data for distributed applications**



ORACLE

Supported System Integration Types

The types of system integration supported by Oracle Workflow are message-based. By supporting the communication of messages between systems, Oracle Workflow lets you define processing across different systems encompassing both your own enterprise and your business partners. The power of this cross-system processing, together with the flexibility provided by subscription-based processing, enables you to use Oracle Workflow for e-business integration.

Designing Applications for Change

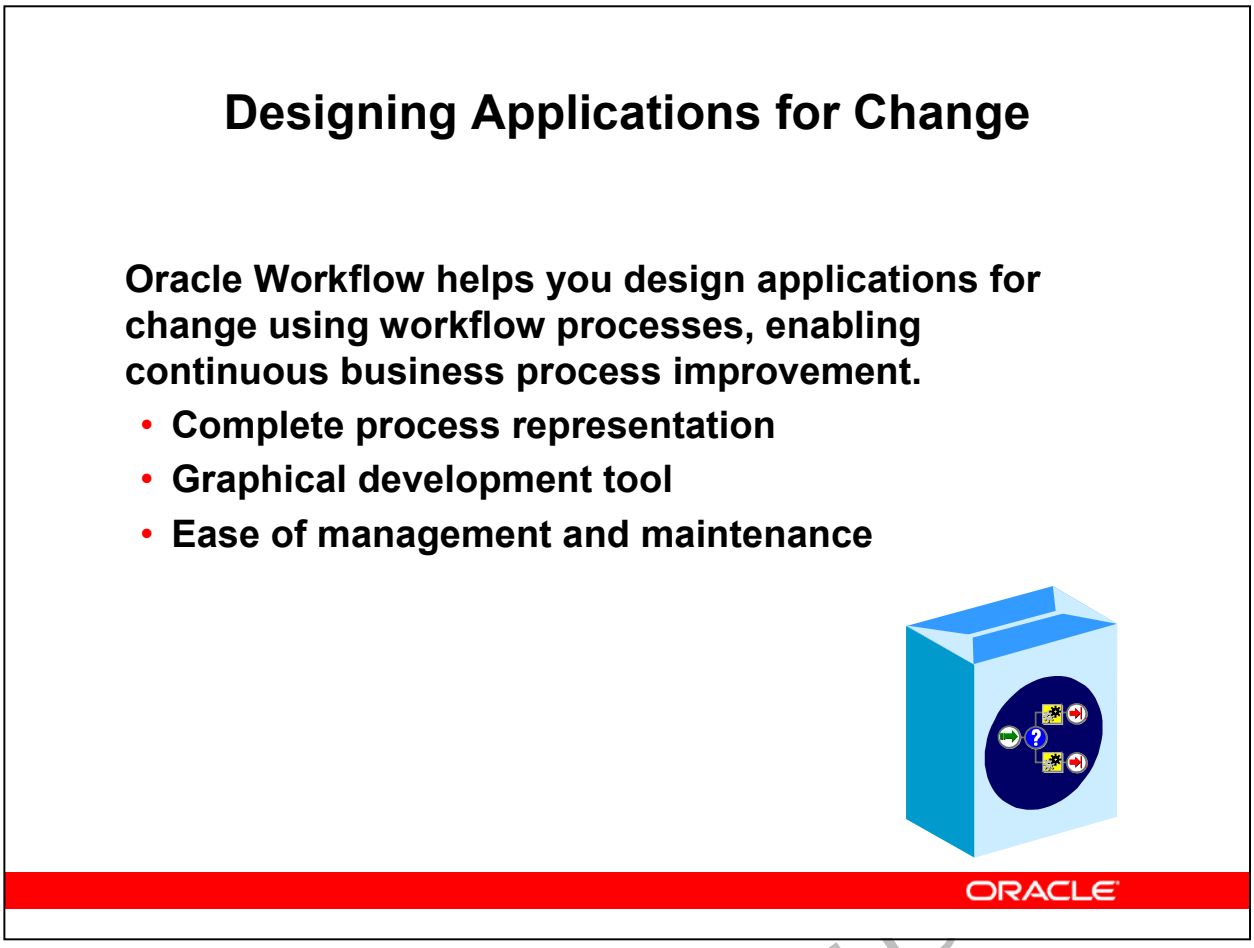
- **Business processes change over time.**
- **Good design in an e-business environment requires:**
  - **Dynamic processes sympathetic to change**
  - **No artificial constraints on business processes**
  - **The ability to modify business processes without changing code**
  - **A visual overview of business processes**

ORACLE

Designing Applications for Change

The benefits of good design include:

- Reduced cost of ownership through diminished development costs
- Ease of management and maintenance
- Visual documentation of business processes



Designing Business Processes for Change

Complete process representation

- A workflow process can cross organizational and company boundaries to represent an end-to-end flow.
- You can provide alternatives within a process to accommodate different situations.
- You can build management metrics and performance goals into a process.

Graphical development tool

- The Oracle Workflow Builder separates business process definition and modeling from code development.
- Workflow diagrams provide a visual overview of your processes.
- You can easily modify a process definition.

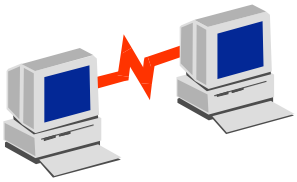
Ease of management and maintenance

- Oracle Workflow lets you analyze time and costs for entire business processes.
- You can refine your process definitions according to your analysis to streamline them and reduce time and costs.
- You can easily implement a modified process.

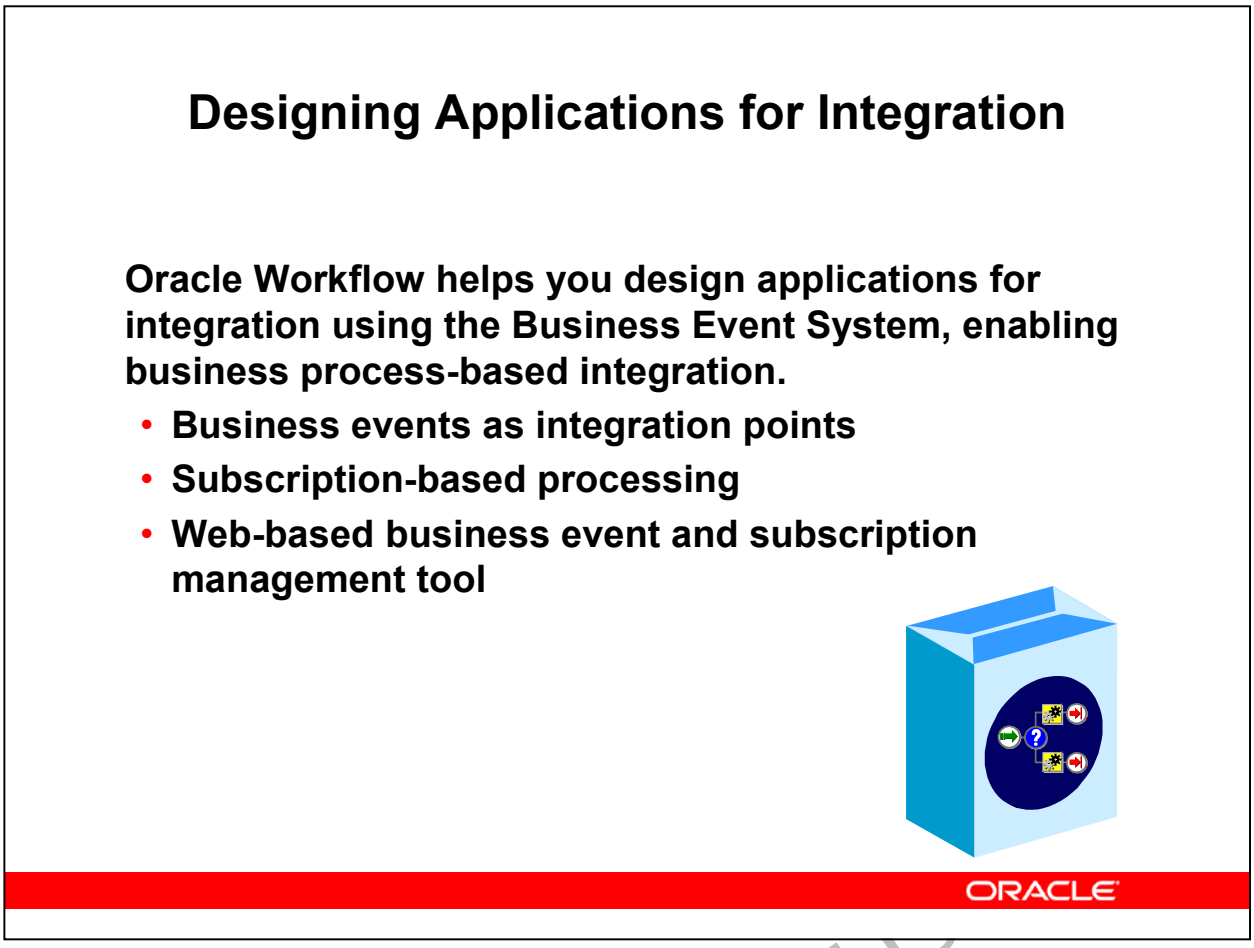
- Oracle Workflow empowers process participants by giving them access to review the progress and current status of their processes.
- Users can find the answers to many common questions themselves using these monitoring capabilities.

Designing Applications for Integration

- System integration is increasingly required for e-business.
- Good design in an e-business environment requires:
  - Availability of integration points in applications for immediate or future use
  - Noninvasive configuration
  - The ability to modify integration processing without changing code



ORACLE



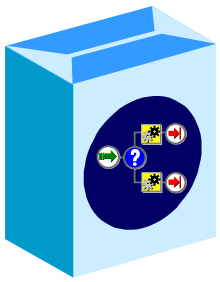
Designing Applications for Integration

- Business events as integration points
  - Applications can raise business events at any point where further processing or integration might be required.
  - You can define the processing to be triggered by an event immediately after defining the event or at any later time.
- Subscription-based processing
  - Use subscriptions to specify the processing you want to perform for a business event.
  - You can define multiple subscriptions to the same event to perform additional processing for different purposes.
- Web-based business event and subscription management tool
  - The Event Manager separates event subscription definition from code development.
  - You can use the Event Manager web pages to define, update, or delete event subscriptions without intruding on the core application.

Oracle Workflow Availability

Oracle Workflow is available in two versions:

- **Standalone**
  - **With the Oracle Database Server (both Standard Edition and Enterprise Edition)**
  - **With the Oracle9i Application Server**
- **Embedded in Oracle E-Business Suite**
  - **Self-service applications**
  - **Professional applications**



ORACLE

Oracle Workflow Availability

A standalone version of Oracle Workflow is available with the Oracle database and Oracle9*i* Application Server. The standalone version of Oracle Workflow is leveraged by products such as JDeveloper, Oracle Warehouse Builder, and Oracle9*i*AS InterConnect, and is embedded in third party products.

Oracle Workflow is also available embedded in the Oracle E-Business Suite to enforce a common set of business rules. In the Oracle E-Business Suite, Oracle Workflow is incorporated in applications including Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Human Resources Management Systems (HRMS). Oracle Workflow is leveraged by both professional applications, which are typically Forms-based applications for power users, and self-service applications, which are typically HTML-based applications for more casual users.

Summary

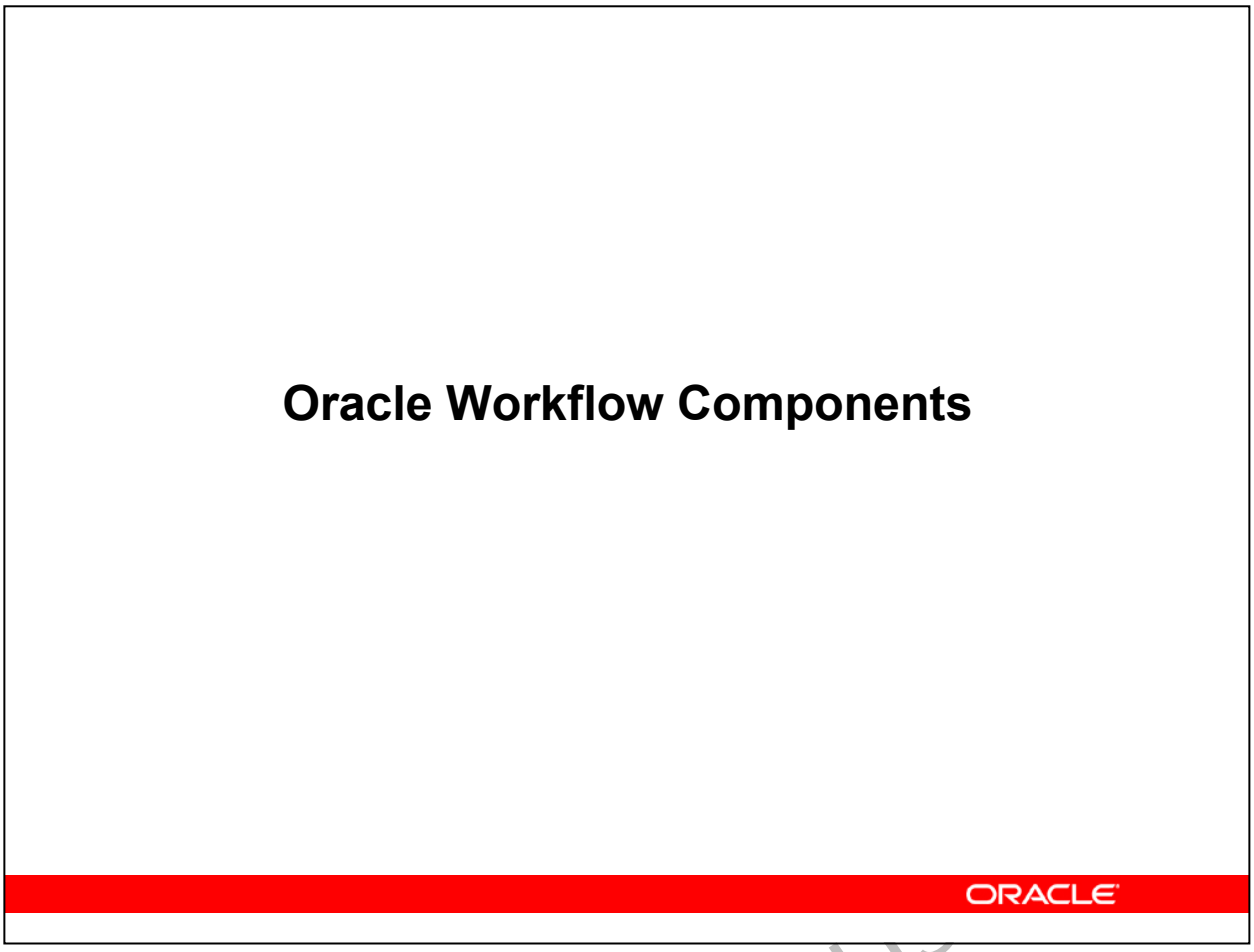
In this lesson, you should have learned how to:

- **Explain the benefits of Oracle Workflow.**
- **Discuss the concept of a workflow process.**
- **Discuss the concept of business events.**

ORACLE

Oracle Workflow Components

Chapter 2

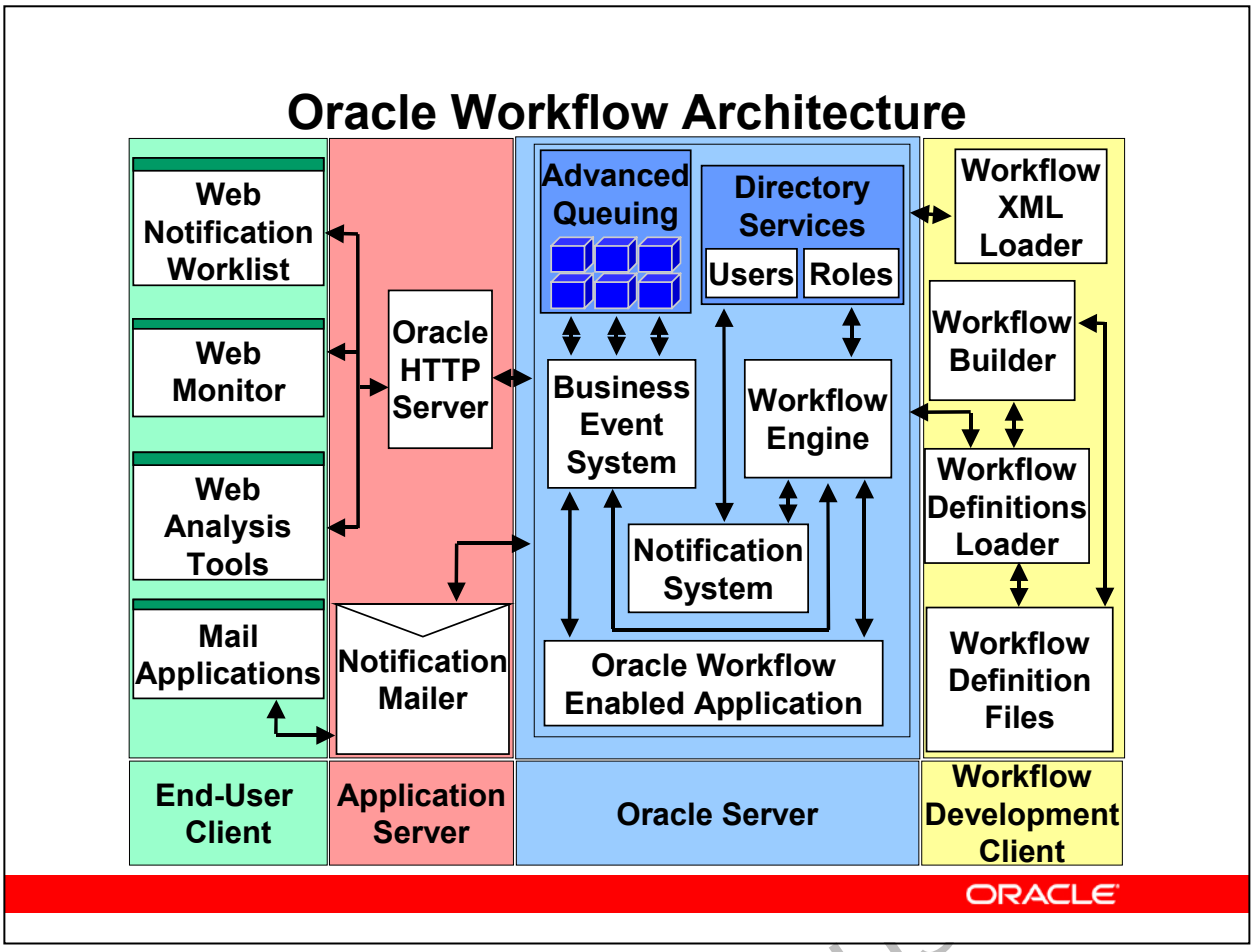


Objectives

After completing this lesson, you should be able to do the following:

- **Describe the architecture and components of Oracle Workflow.**
- **Discuss how the Business Event System communicates events between systems.**
- **Describe how the Workflow Engine executes workflow processes.**

ORACLE



Oracle Workflow Architecture

Workflow Development Client

The development client is a PC running Windows 95, Windows 98, Windows 2000, or Windows NT 4.0 or higher. This platform is used to create and modify Oracle Workflow process definitions.

Oracle Server

The Oracle Server platform is the Oracle RDBMS. This platform hosts the business application integrated with Oracle Workflow, the Workflow Engine, Business Event System, Notification System, and directory services.

Application Server

The application server is the environment outside of the RDBMS. This environment includes ancillary services such as the Oracle9*i* Application Server (Oracle9*i*AS) as the web server, and the Notification Mailer.

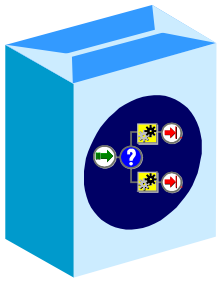
End-User Client

The end-user client is the workstation or PC that an end user uses to perform daily tasks. This client includes browser support for reviewing and responding to notifications in the

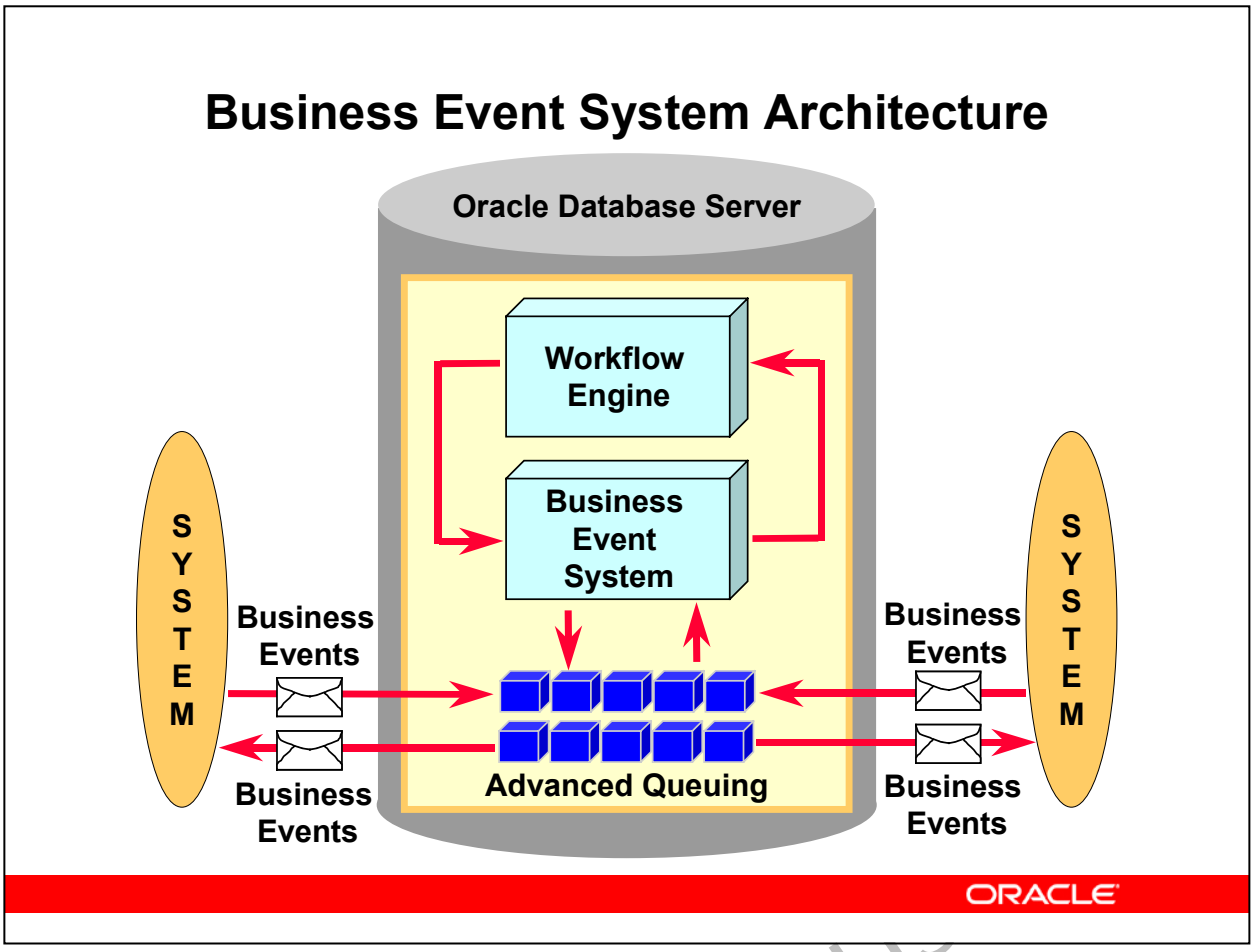
Notification Worklist, the Oracle Workflow Monitor, and web analysis tools, as well as mail applications for reviewing and responding to notifications by e-mail.

Oracle Workflow Components

- Business Event System
- Workflow Engine
- Oracle Workflow Builder
- Notification System
- Notification Worklist
- Directory Services
- Workflow Monitor
- Workflow Definitions Loader
- Workflow XML Loader



ORACLE



Business Event System Architecture

The Business Event System is an application service that communicates business events between systems. Oracle Workflow with the Business Event System can act as a system integration messaging hub that relays business event messages among systems.

The Business Event System leverages Oracle Advanced Queuing (AQ) to send messages from one system to another.

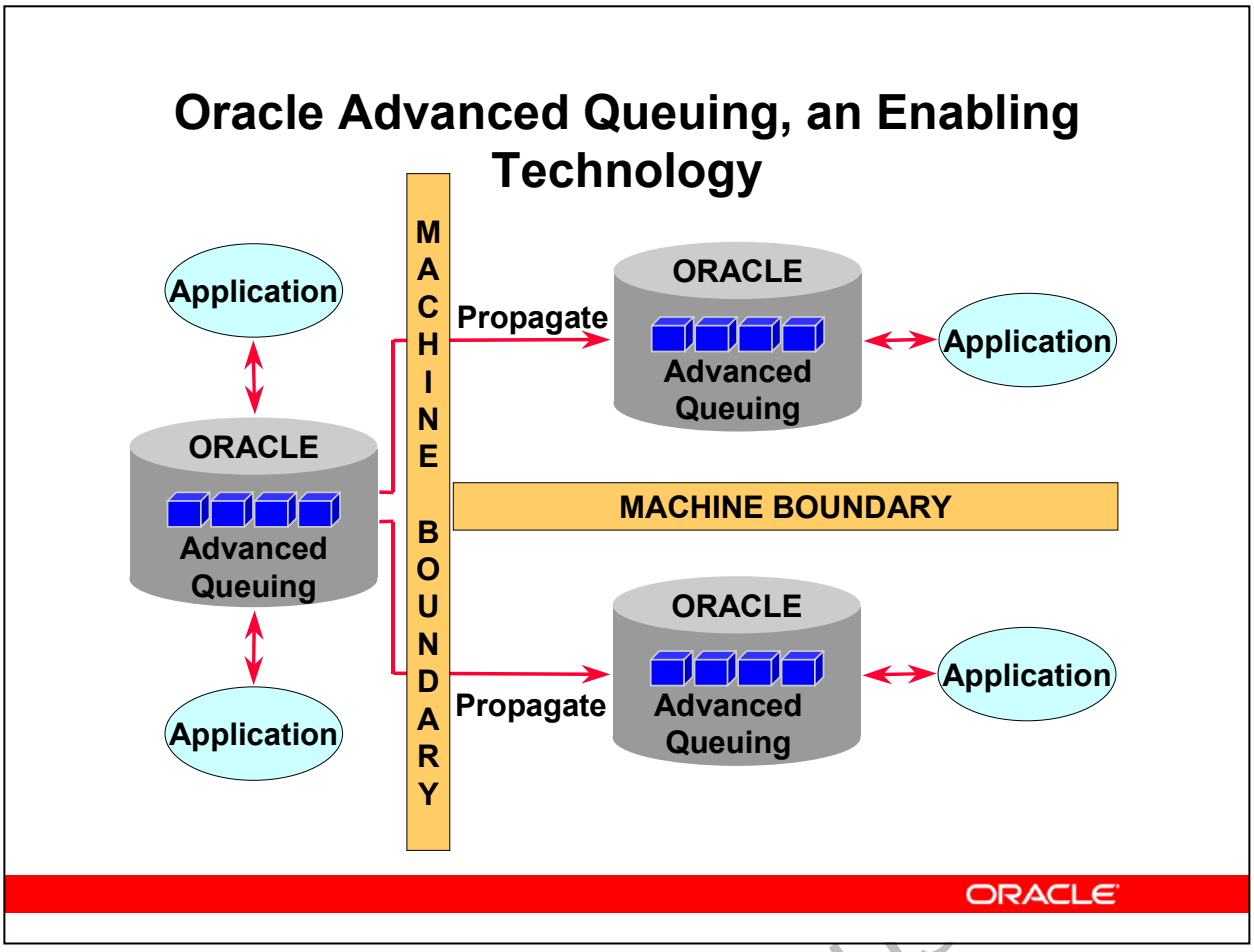
For the greatest flexibility in routing and processing business events, you can model your business process logic in powerful cross-system workflow processes that are executed by the Workflow Engine. However, the Business Event System can also function independently of the Workflow Engine.

Business Event System Components

The Business Event System includes:

- **The Event Manager—Lets you register:**
  - **Business events**
  - **Systems**
  - **Named communication agents within systems**
  - **Subscriptions to events that are significant to your systems**
- **Workflow Engine event activities—Let you model business events within workflow processes**

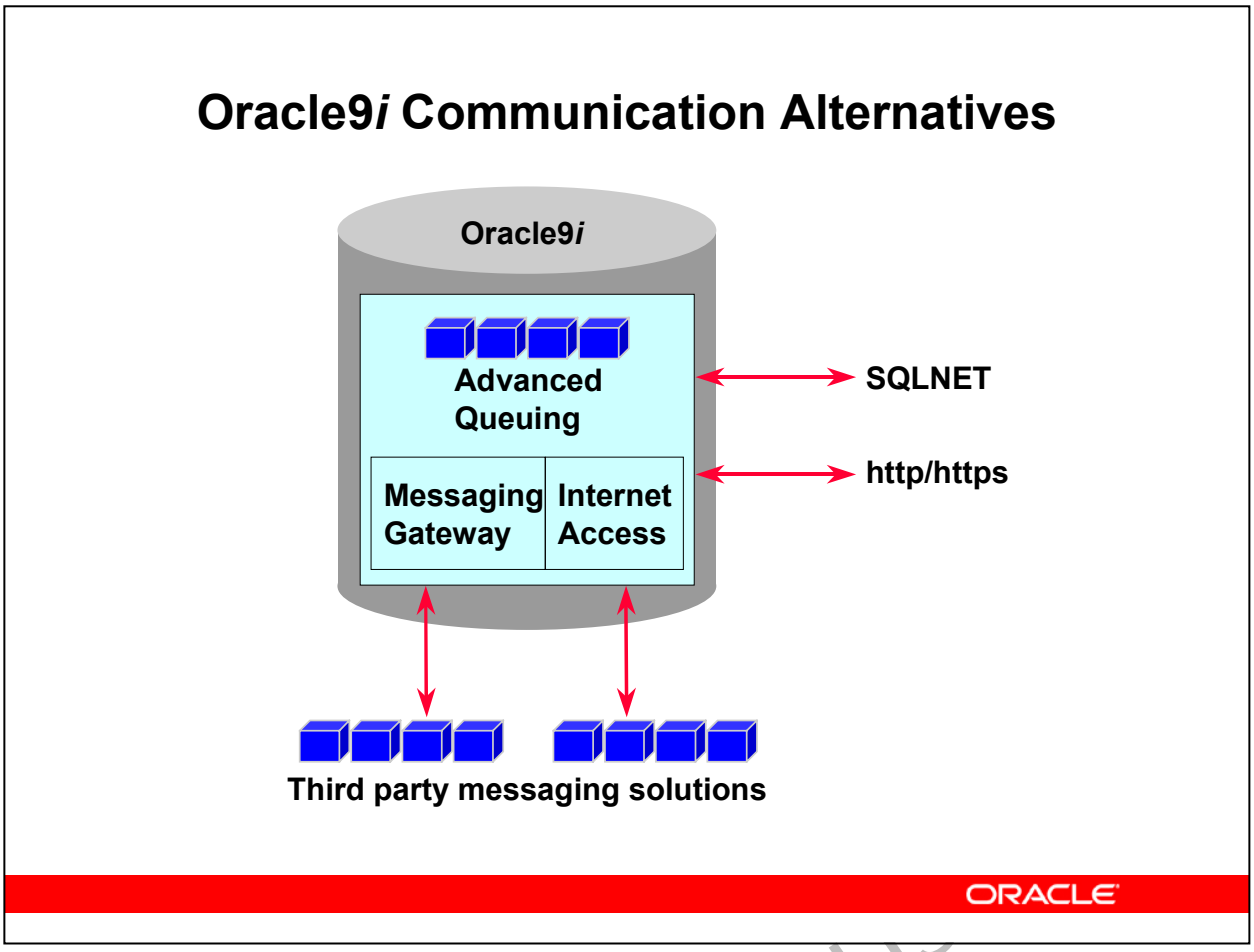
ORACLE



Oracle Advanced Queuing, an Enabling Technology

Oracle Advanced Queuing (AQ) allows queue-to-queue propagation across machine boundaries. AQ is a feature of Oracle9*i* that provides database-integrated message queuing functionality, leveraging the functions of the Oracle database to store messages persistently and transmit them using various transport protocols.

For more information, refer to the *Oracle9i Application Developer's Guide - Advanced Queuing*.



Oracle9i Communication Alternatives

In Oracle9*i*, you can use Oracle Advanced Queuing (AQ) for communication by SQLNET, HTTP, and HTTPS protocols, and for integration with third party messaging solutions.

You can use Oracle Net Services (formerly Net8) to propagate messages by the SQLNET protocol. Oracle Advanced Queuing's Internet access functionality lets you perform AQ operations over the Internet by using AQ's Internet Data Access Presentation (IDAP) for messages and transmitting the messages over the Internet using transport protocols such as HTTP or HTTPS.

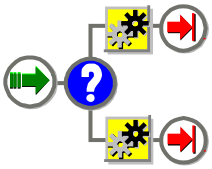
Messaging Gateway is a feature of AQ that enables communication between applications based on non-Oracle messaging systems and AQ. Standard AQ functionality provides propagation between two AQ queues; Messaging Gateway extends that propagation to legacy applications based on non-Oracle messaging systems.

For more information, refer to the *Oracle9i Application Developer's Guide - Advanced Queuing*.

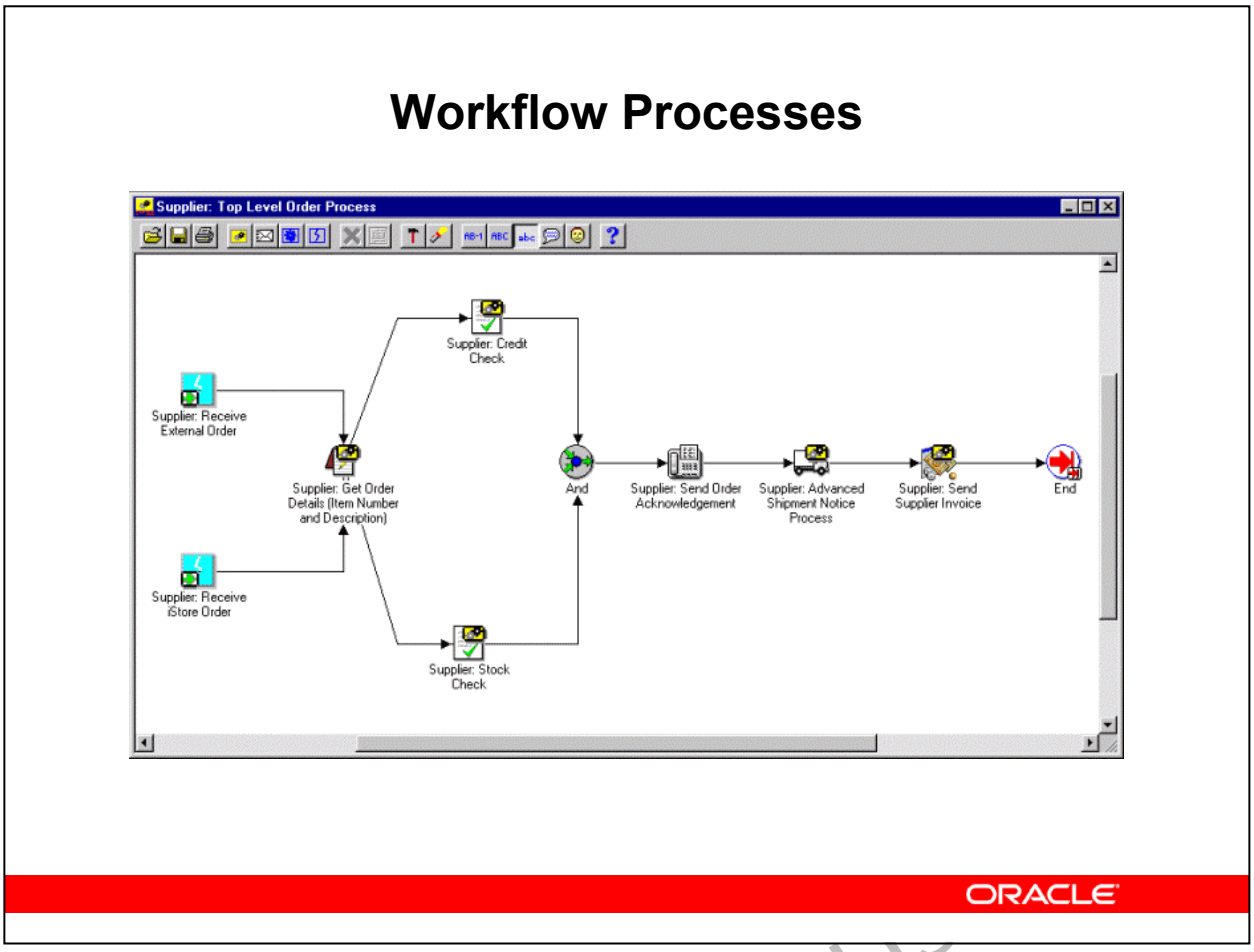
Workflow Engine

The Workflow Engine:

- **Is embedded in the Oracle database server**
- **Uses the process definition created with Oracle Workflow Builder to coordinate the routing of activities for the process**
- **Monitors the state of each activity in a workflow process**
- **Signals any changes in the workflow state using calls to PL/SQL or Java APIs**
- **Guarantees consistency between the application and the workflow state because of Oracle Database Server transactional integrity**



ORACLE



Workflow Processes

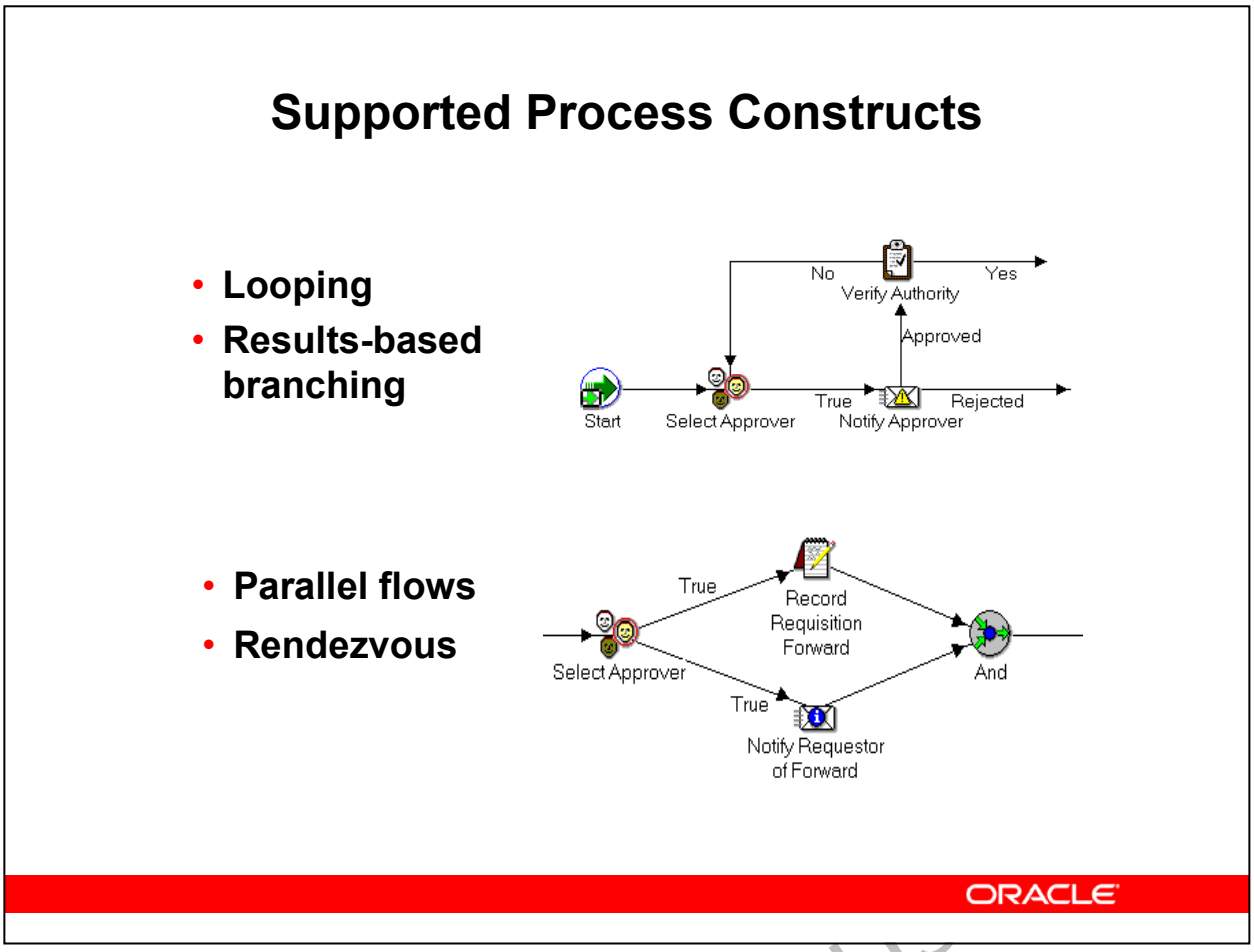
- A workflow process definition must be saved to the same database as the Workflow Engine.
- A process definition is composed of activities and the transitions between them.
- A completed application transaction or event can initiate a workflow process by raising an event or by calling a series of Workflow Engine APIs.
- The Workflow Engine locates the 'Start' activity in the process definition.
- The Workflow Engine drives through the process, performing all automated steps such as function activities and Raise and Send event activities, until an asynchronous activity such as a notification, Receive event activity, or blocking activity occurs.
  - The Workflow Engine calls the Notification System to deliver a notification message to an appropriate role. Once a user of that role completes the notification response, the Workflow Engine continues to drive through the remaining activities in the process.
  - If a blocking activity is encountered, the Workflow Engine waits for an external program to complete and call the appropriate Workflow Engine API before proceeding to the next activity.

- If a Receive event activity is encountered, the Workflow Engine waits to receive the event from the Business Event System before proceeding to the next activity.

• The process completes when the Workflow Engine encounters an 'End' activity.

Example: Order Processing

This example shows a workflow process that includes business events. The example comes from the Event System Demonstration workflow which is available with the standalone version of Oracle Workflow.

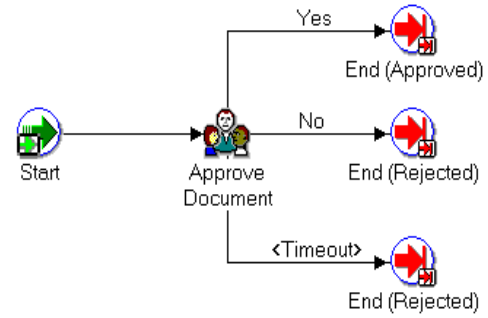


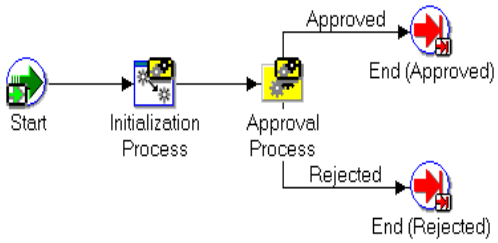
Supported Process Constructs

The Workflow Engine supports sophisticated workflow rules to model your business logic.

Supported Process Constructs

- **Voting**
- **Timeouts**
  - **Escalation**
  - **Automatic forwarding**
- **Subprocesses (unlimited hierarchy)**

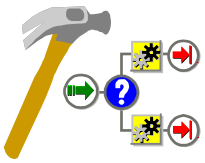




ORACLE

Oracle Workflow Builder

- **Oracle Workflow Builder is the development tool for Oracle Workflow.**
  - **Lets you graphically define and customize workflow definitions**
  - **Requires a PC running Windows 95, 98, 2000 or Windows NT**
- **You can save workflow definitions to a database or a flat file.**
- **Oracle Workflow Builder consists of two parts:**
  - **Navigator Tree**
  - **Process Diagram**



ORACLE

Oracle Workflow Builder

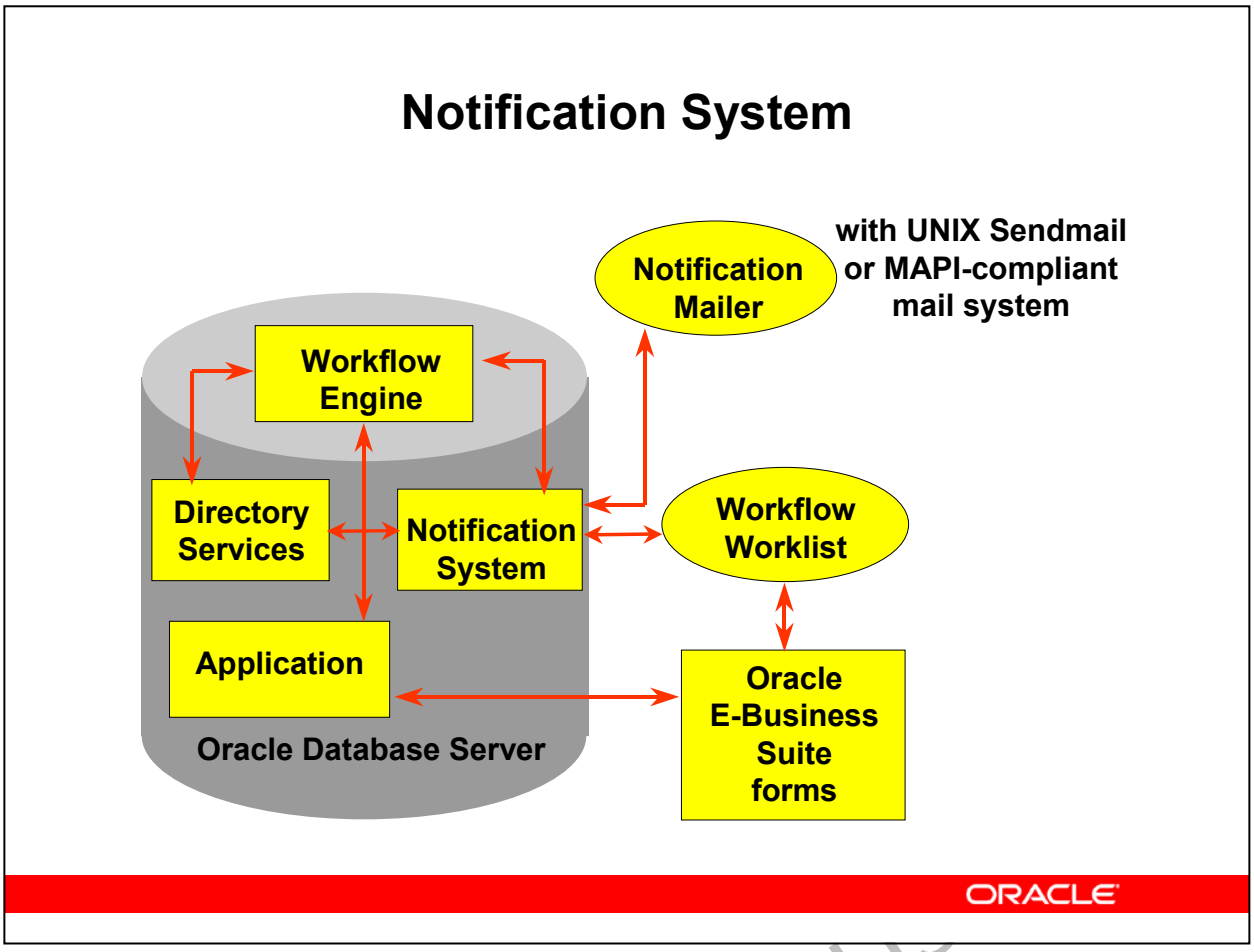
Saving workflow definitions as flat files on the local file system enables designers to back up their work and use source control.

Navigator Tree

The navigator provides a tree structure for the workflow definition, with the highest level being the data store. Next is the item type, which is a grouping of workflow objects into a high level category. The lower levels are the workflow objects themselves, such as attributes, processes, notifications, functions, events, messages, and lookup types. All these objects are organized into their respective categories.

Process Diagram

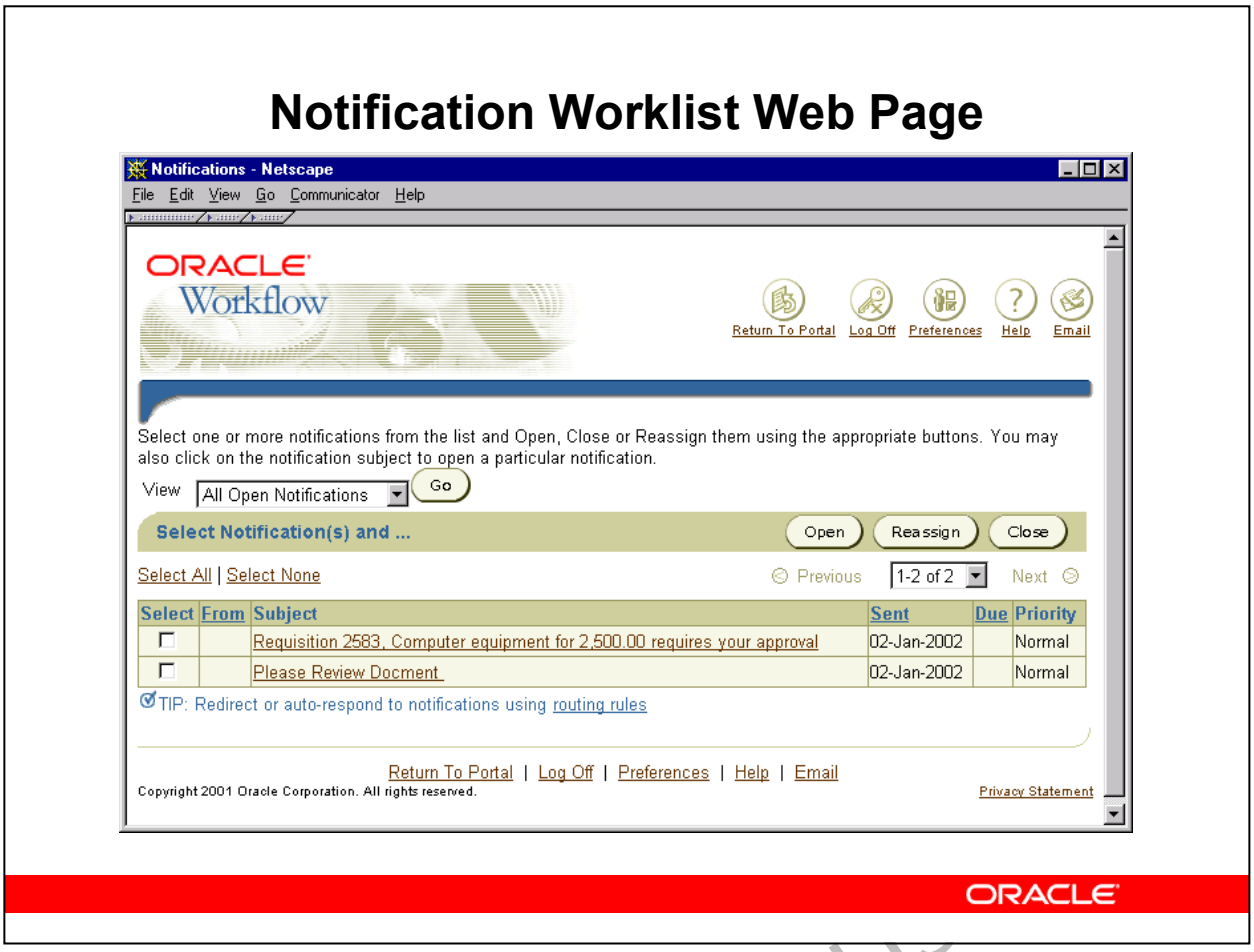
The diagram is made up of icons representing workflow objects. A diagram is built by dragging the objects from the navigator window and dropping them into the process diagram window. This method is called bottom-up design. You can also create new objects as you design the diagram and complete the definitions of those objects later. This method is called top-down design.



Notification System

The Notification System:

- Routes notifications to a role, which can be a single user or group of users
- Enables users to receive and respond to notifications using an e-mail application or Web browser of choice
- Allows any users with access to the Internet to be included in a workflow process
- Provides access to the Notification Worklist from Oracle E-Business Suite
- Enables users to drill down to any URL or Oracle E-Business Suite form to respond to a notification from the Notification Worklist



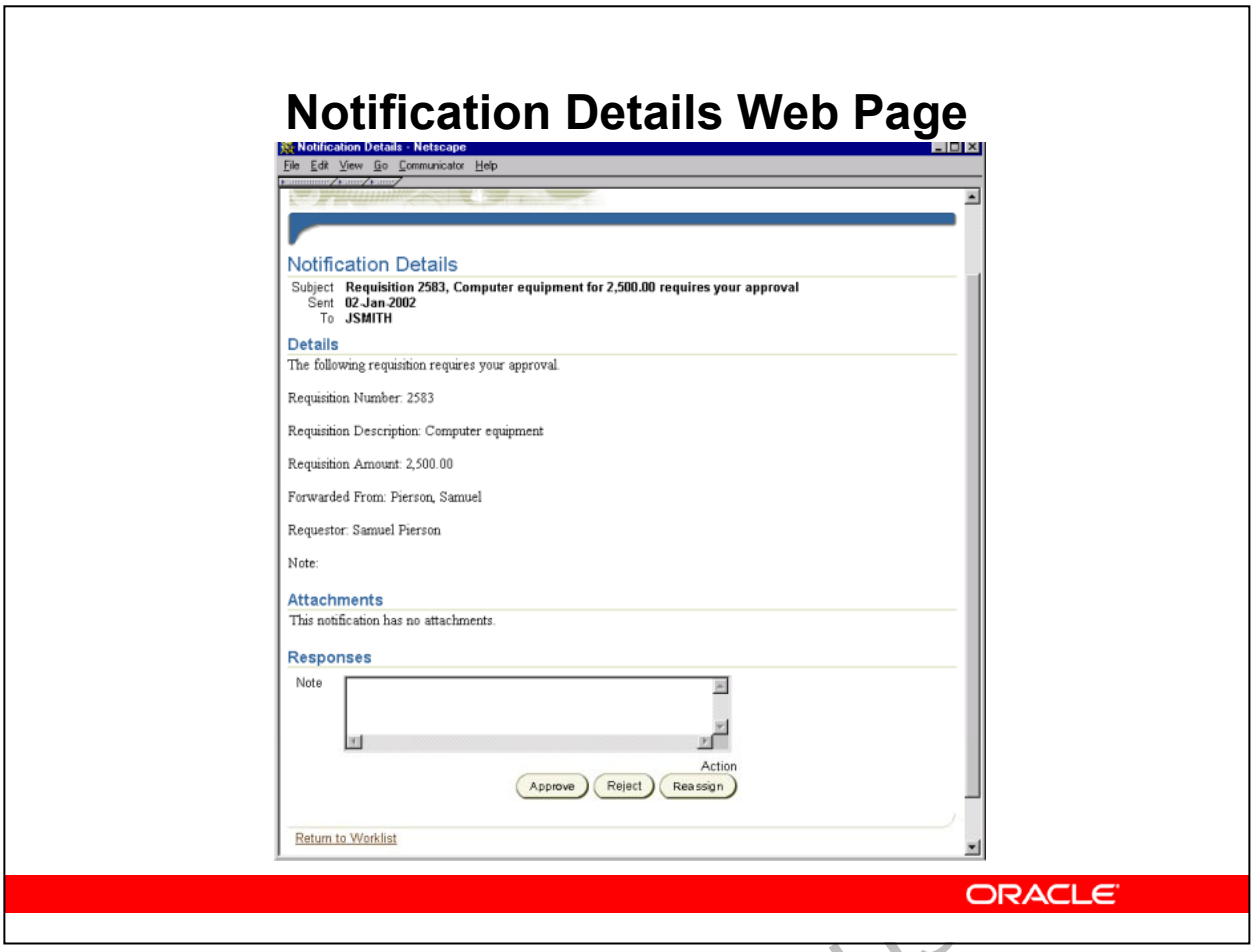
Notification Worklist

The Notification Worklist is a feature that provides a list of open notifications for a particular user. The Worklist is viewable through the Oracle Workflow Web Agent. You can also configure the Notification Mailer to send an e-mail summarizing the outstanding notifications. From the Worklist, a user can view the notifications as well as respond to those that require a response.

New Self-Service Web Page Format

The Oracle Workflow web pages are being converted to the new format for Oracle self-service web applications. Depending on your version of Oracle Workflow and which patches you have applied, you may see Oracle Workflow web pages in the previous format or in the new format. Currently, the Notifications Worklist and the Notification Rules web pages are available in the new format for the version of Oracle Workflow embedded in Oracle E-Business Suite.

Notification Details Web Page

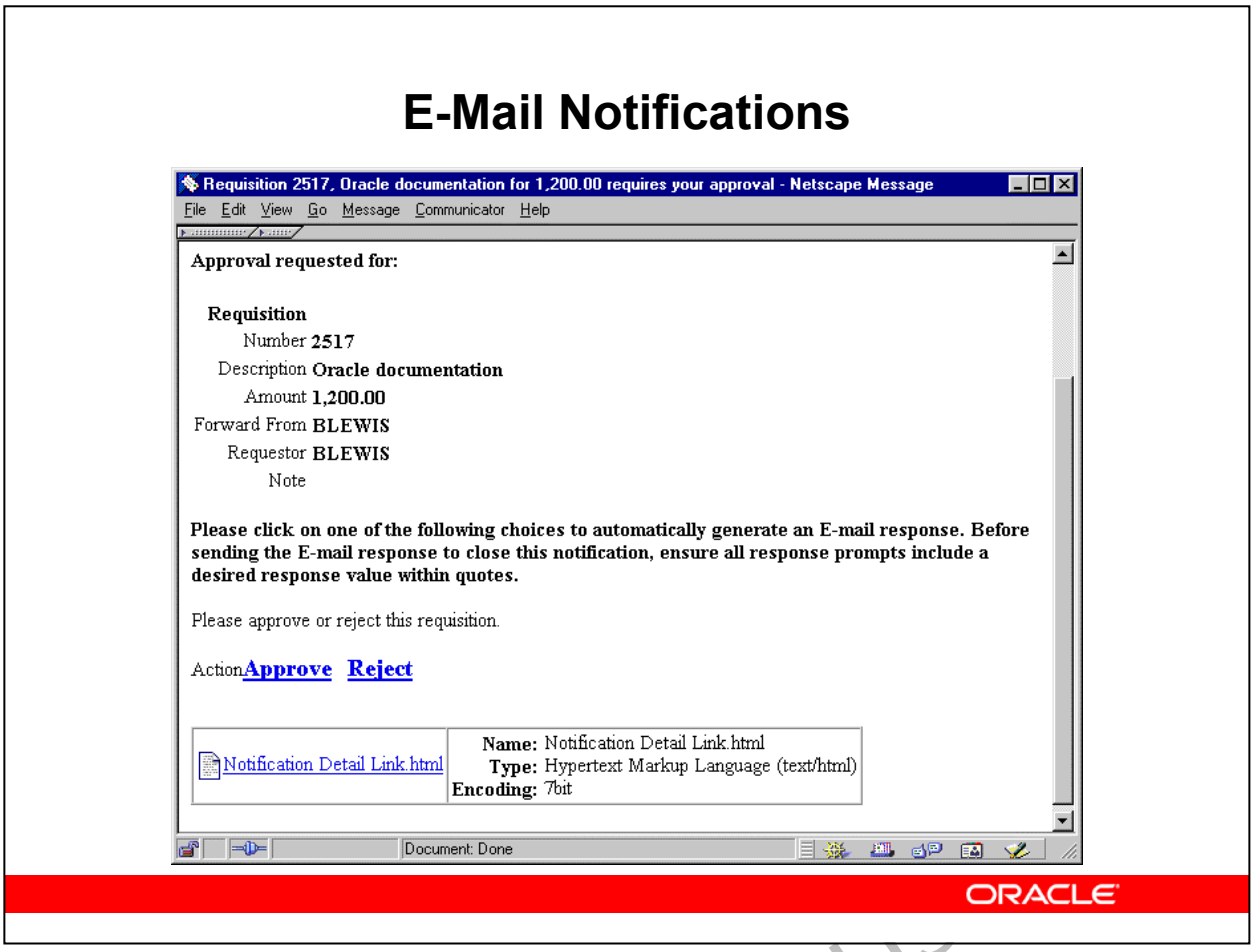


Viewing Notifications from a Web Page

The Notification System has an Oracle Web Agent interface that dynamically generates each notification for a user.

To view notifications in the Notification Details web page, you must have Oracle HTTP Server installed as the web server for Oracle Workflow, and you must have a Web browser that supports Frames and JavaScript.

The Notification Details web page can display extended HTML message formats and can include links to URLs or Oracle E-Business Suite forms that let users research and make decisions through online inquiry.



E-Mail Notifications

- The Notification System interfaces with the Notification Mailer program to send e-mail notifications to users and roles. Users can reply to e-mail notifications using their e-mail client.
- There are two versions of the Notification Mailer program to integrate with UNIX Sendmail or Windows NT MAPI-compliant mail applications.
- E-mail notifications can be delivered to users on other mail systems if the appropriate UNIX gateway software is also installed.

Directory Services

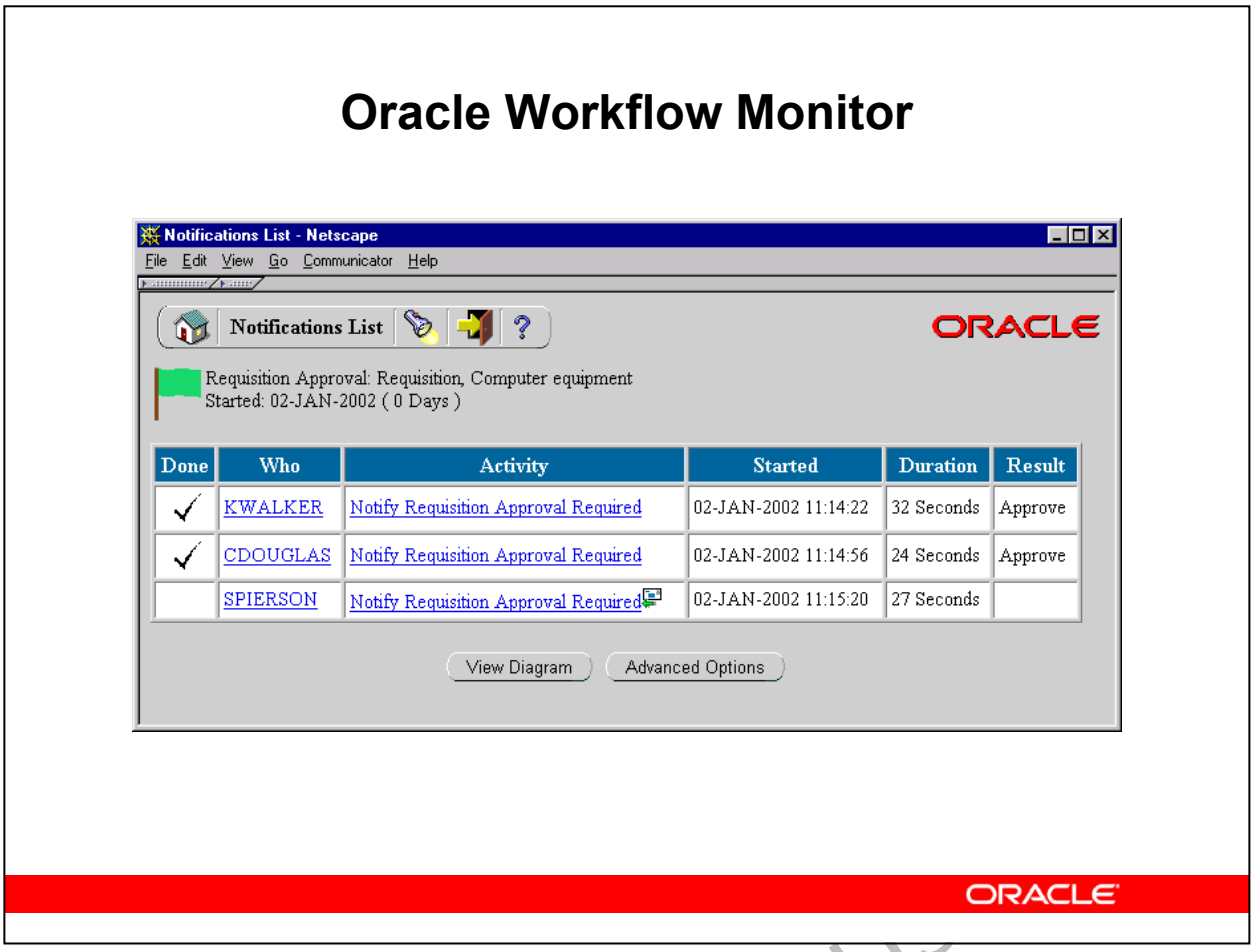
The directory service for Oracle Workflow is implemented as a set of views that are mapped across the user tables of the underlying application.

- **WF_USERS: Contains information on user names, display names, notification preferences and e-mail addresses**
- **WF_ROLES: Contains information on the roles of which users can be members**
- **WF_USER_ROLES: Contains information on the association of users with roles**

ORACLE

Directory Services

Users can be associated with more than one role, and a role may contain more than one user. The Workflow Engine and Notification System use the directory service to determine who should receive notifications and in what format. Notifications can be delivered to an individual user or to all members of a particular role.

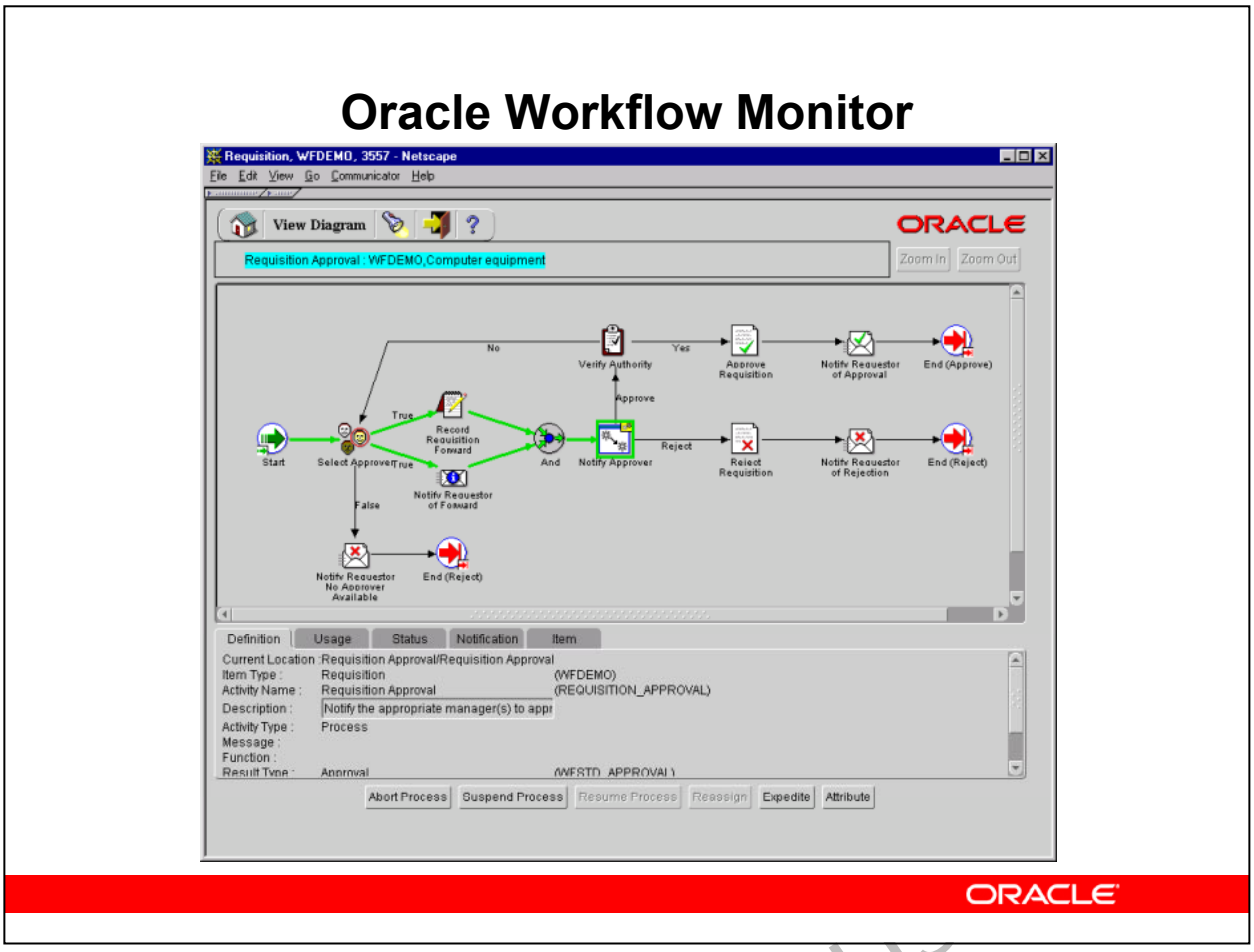


Workflow Monitor

The Workflow Monitor Notifications List:

- Displays all the current notifications that have been sent for a process that require a response
- Shows the decision makers in the process as well as the current owner of any outstanding notifications

Note: The Find Processes web page lets you search for the process instance you want to view in the Workflow Monitor Notifications List.



Workflow Monitor

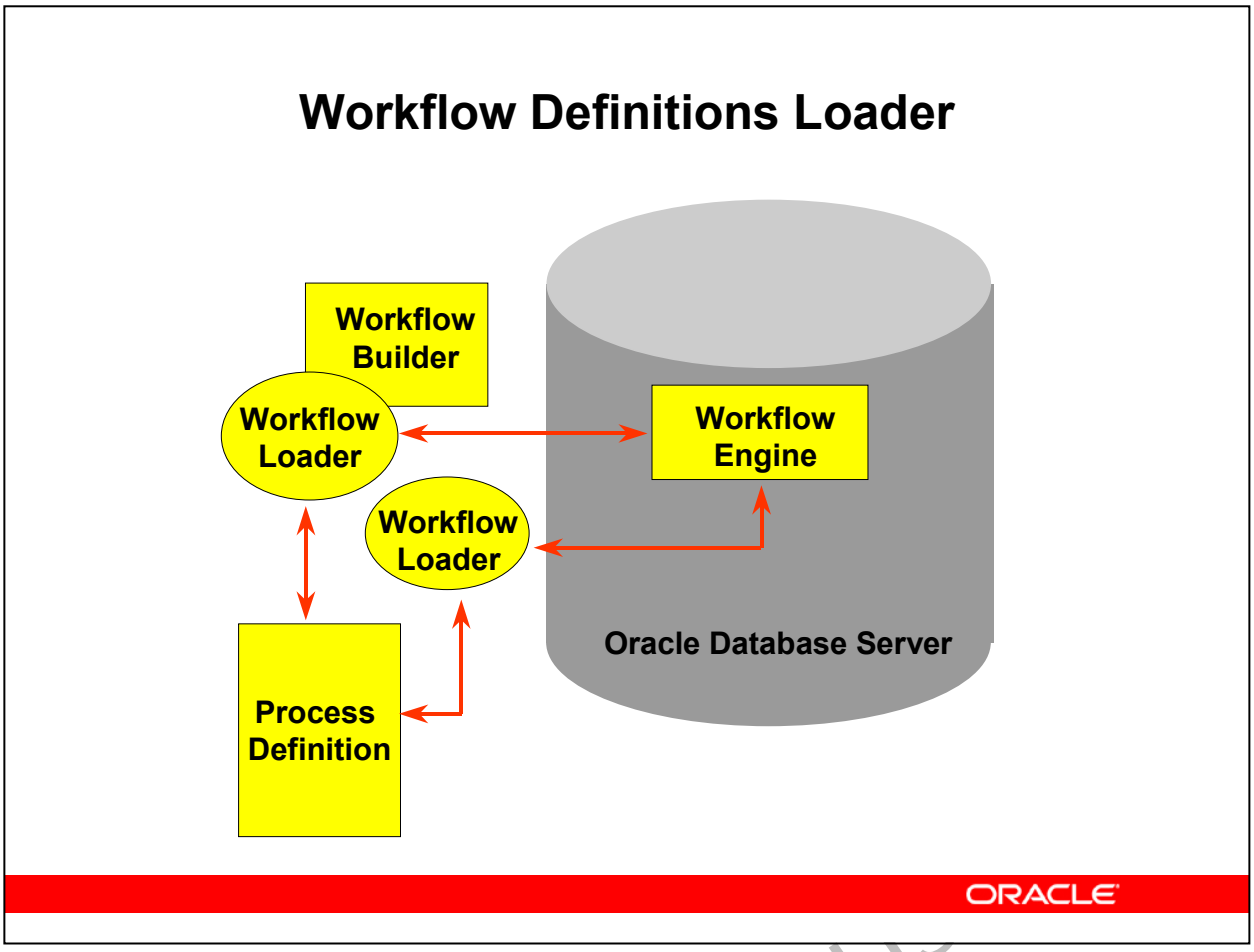
The Workflow Monitor diagram is a Java applet that displays status information for a single workflow process instance. The Workflow Monitor diagram:

- Graphically depicts the status of a workflow process instance in its upper frame
- Displays detailed information about individual activities, as well as about the process as a whole, in the lower tabbed frame
- Runs in USER mode to display end user information or ADMIN mode to display administrator information and functionality

Note: The Find Processes web page lets you search for the process instance you want to view in the Workflow Monitor.

To view a process in the Workflow Monitor, you must have Oracle HTTP Server installed as the web server for Oracle Workflow, and you must have a web browser that supports Java Development Kit (JDK) Version 1.1.8 or higher and Abstract Windowing Toolkit (AWT), such as Netscape Communicator version 4.76 or a higher version of 4.7x, or Microsoft Internet Explorer version 5.0x or 5.5.

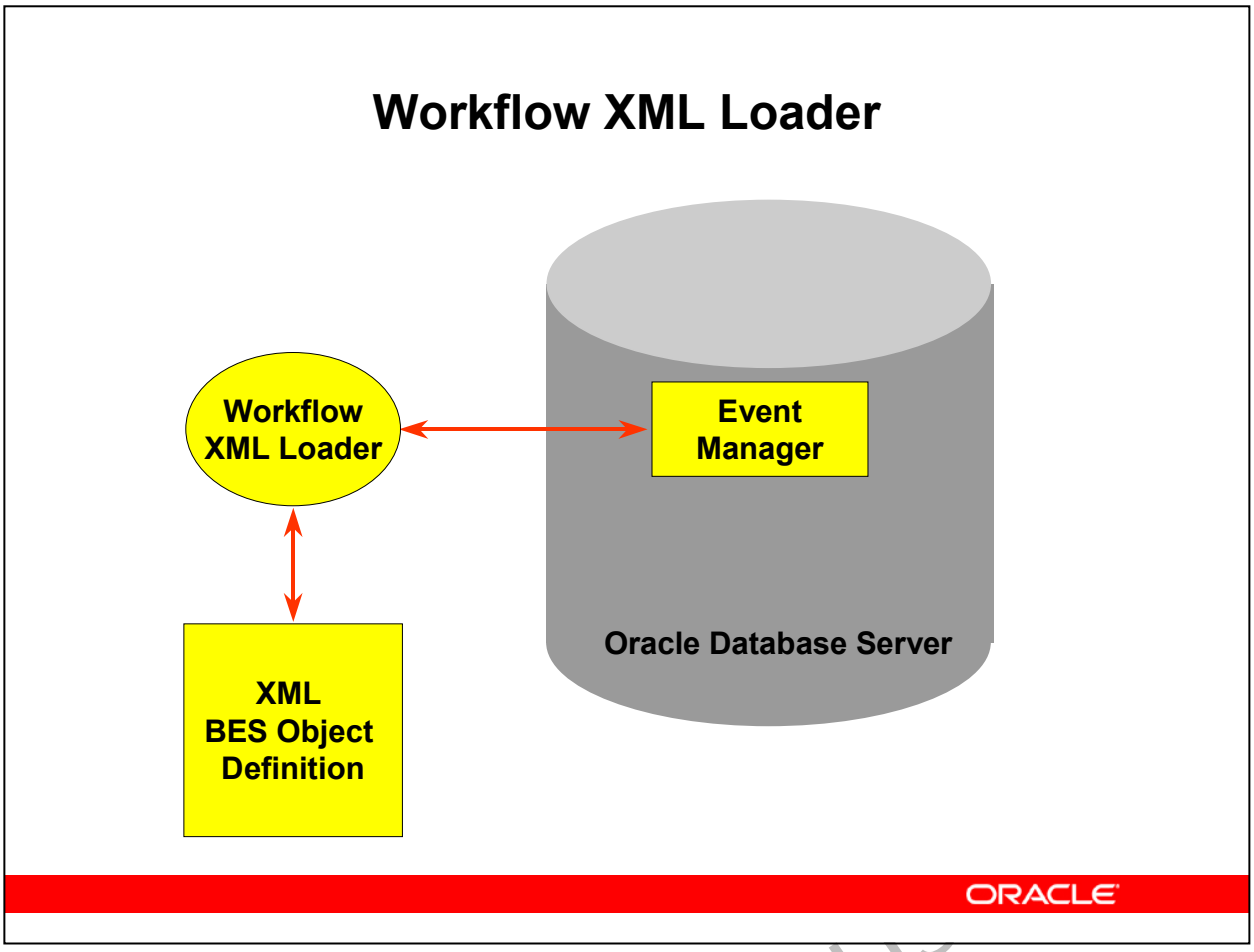
Workflow Definitions Loader



Loading Workflow Definitions

The Workflow Definitions Loader is a utility program that lets you transfer process definitions between a database and a flat file. The Workflow Definitions Loader:

- Runs on the server machine
- Lets you upgrade a database with new versions of process definitions or upload existing process definitions after a database upgrade
- Is also integrated into Oracle Workflow Builder
- Allows process definitions to be source-controlled as flat files



Workflow XML Loader

The Workflow XML Loader is a utility program that lets you transfer XML definitions for Business Event System objects between a database and a flat file.

- When you download Business Event System object definitions from a database, Oracle Workflow saves the definitions as an XML file.
- When you upload object definitions to a database, Oracle Workflow loads the definitions from the source XML file into the Business Event System tables in the database, creating new definitions or updating existing definitions as necessary.

Demonstration

This demonstration shows you how to:

- Open a Workflow process in Oracle Workflow Builder
- Save a Workflow process to a database
- Start a Workflow process from the Launch Processes web page
- Follow a Workflow process through to completion
- Make changes to a Workflow process

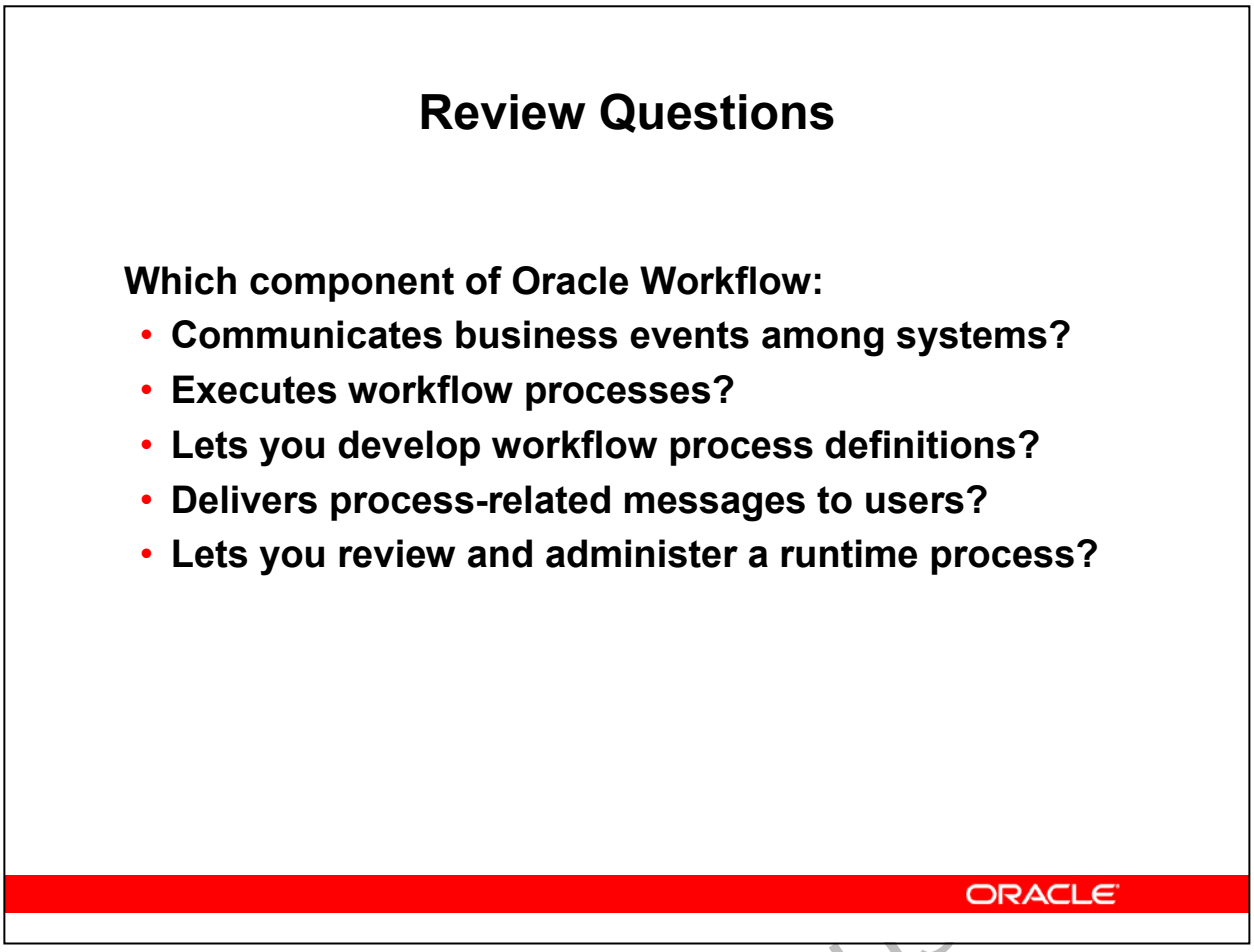
ORACLE

Review Questions

Which component of Oracle Workflow:

- **Communicates business events among systems?**
- **Executes workflow processes?**
- **Lets you develop workflow process definitions?**
- **Delivers process-related messages to users?**
- **Lets you review and administer a runtime process?**

ORACLE



Review Questions and Solutions

Which component of Oracle Workflow:

- Communicates business events among systems? **Business Event System**
- Executes workflow processes? **Workflow Engine**
- Lets you develop workflow process definitions? **Oracle Workflow Builder**
- Delivers process-related messages to users? **Notification System**
- Lets you review and administer a runtime process? **Workflow Monitor**

Summary

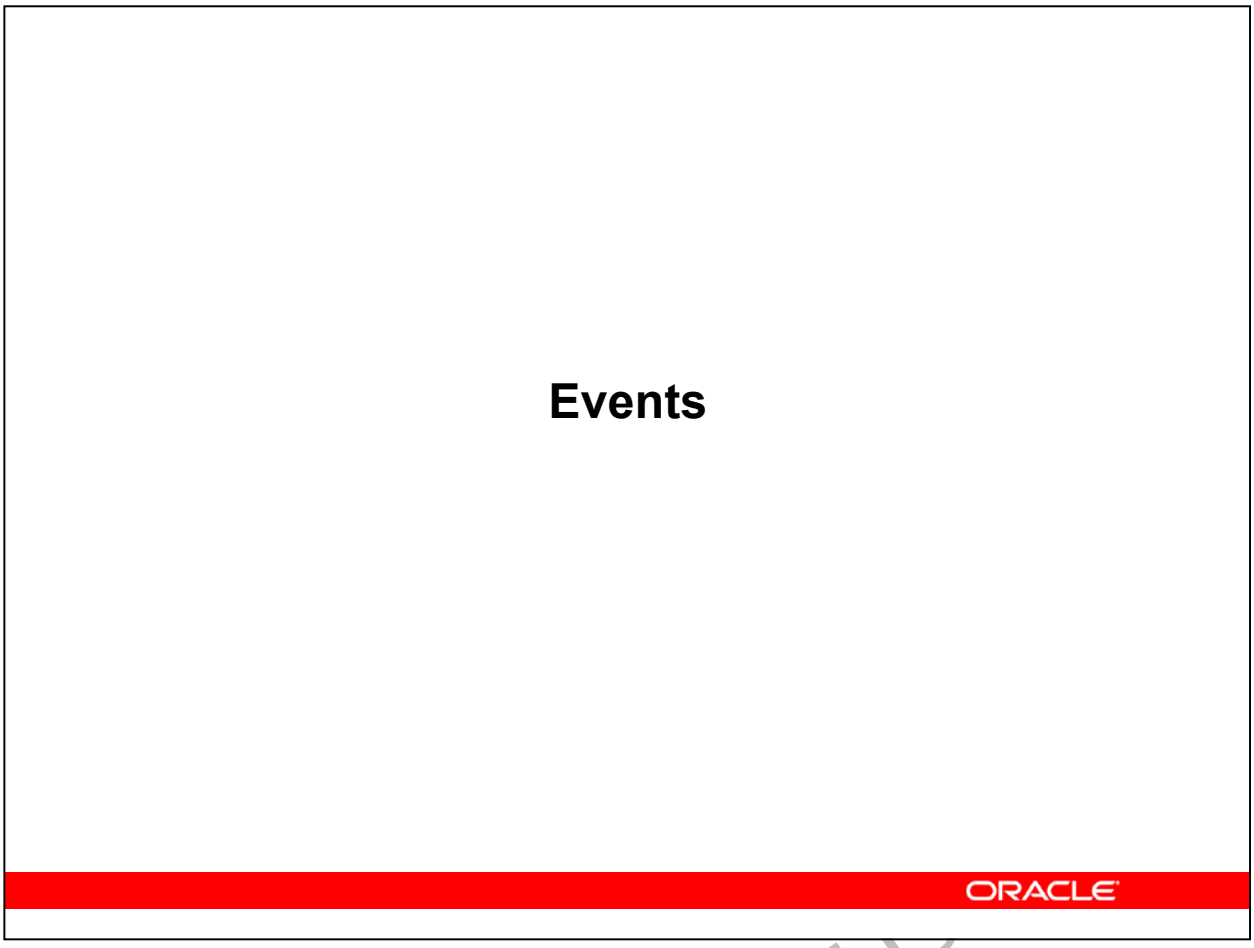
In this lesson, you should have learned how to:

- **Describe the architecture and components of Oracle Workflow.**
- **Discuss how the Business Event System communicates events between systems.**
- **Describe how the Workflow Engine executes workflow processes.**

ORACLE

Events

Chapter 3

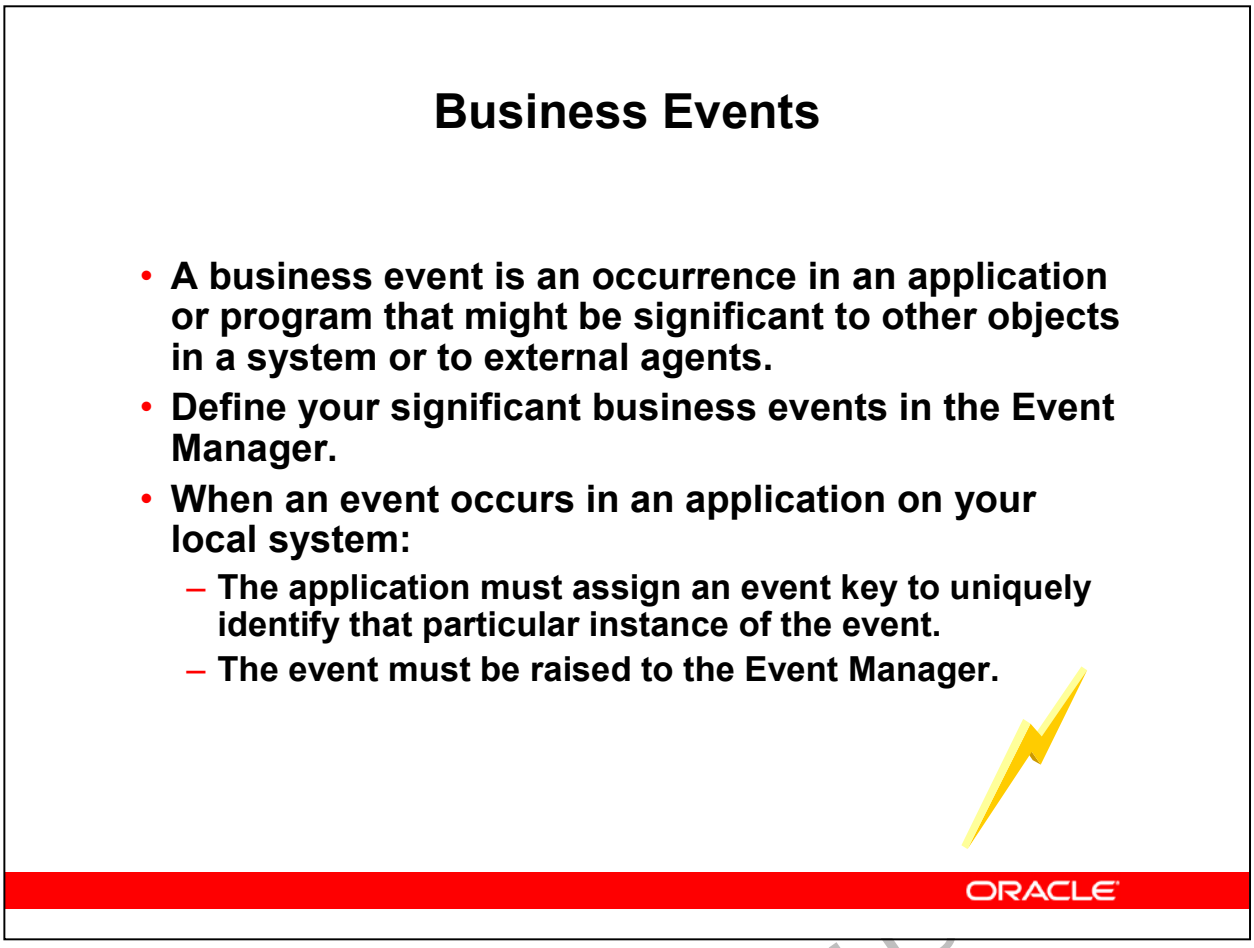


Objectives

After completing this lesson, you should be able to do the following:

- **Define events.**
- **Describe the abstract datatypes used by the Business Event System.**
- **Define event groups.**
- **Raise events.**
- **Use predefined Oracle Workflow events.**

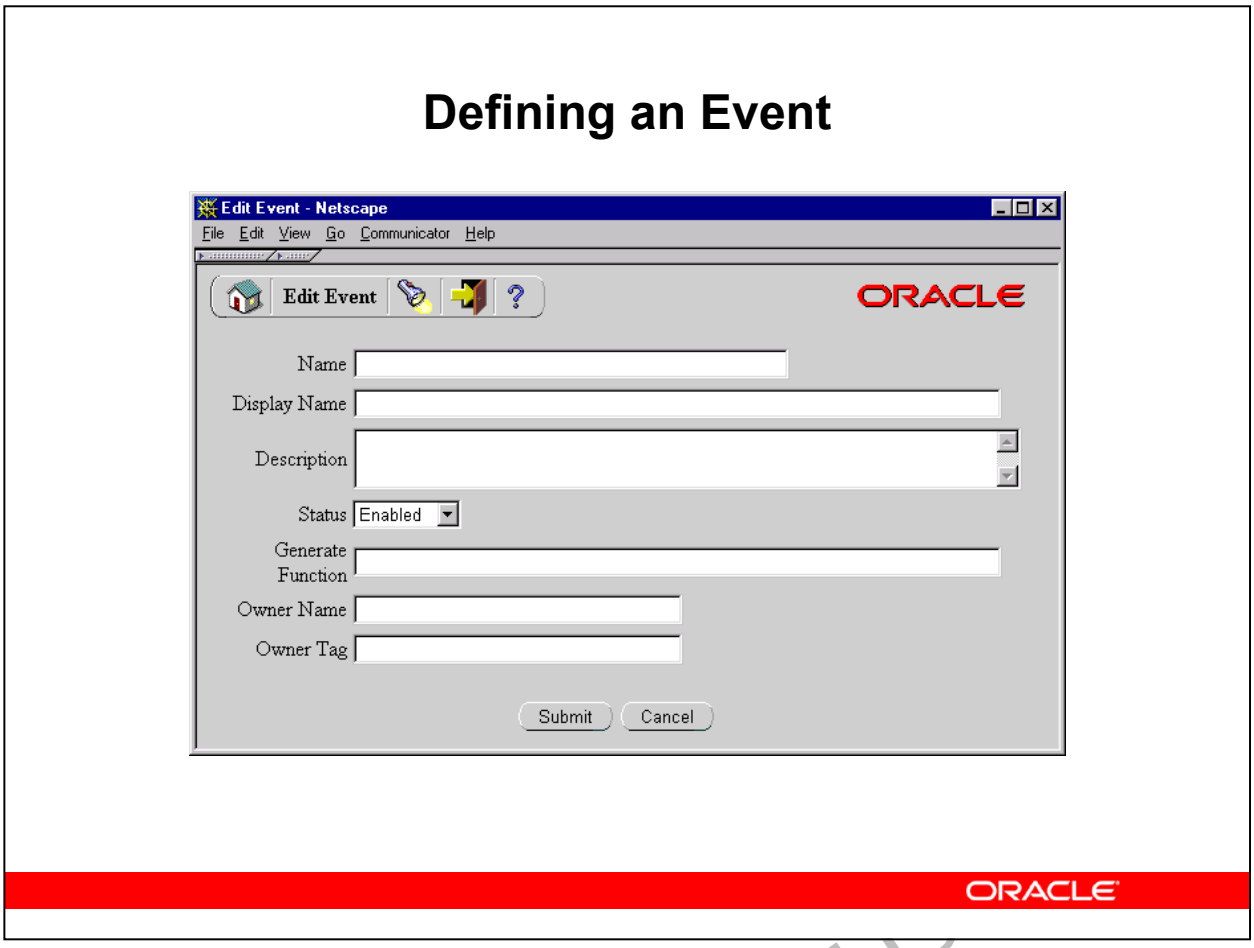
ORACLE



Business Events

For instance, the creation of a purchase order is an example of a business event in a purchasing application. For an event related to a business document, the event key could be the document identifier, such as the purchase order number.

Defining an Event



To Define an Event:

1. Use a web browser to connect to the Events web page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_html.listevents

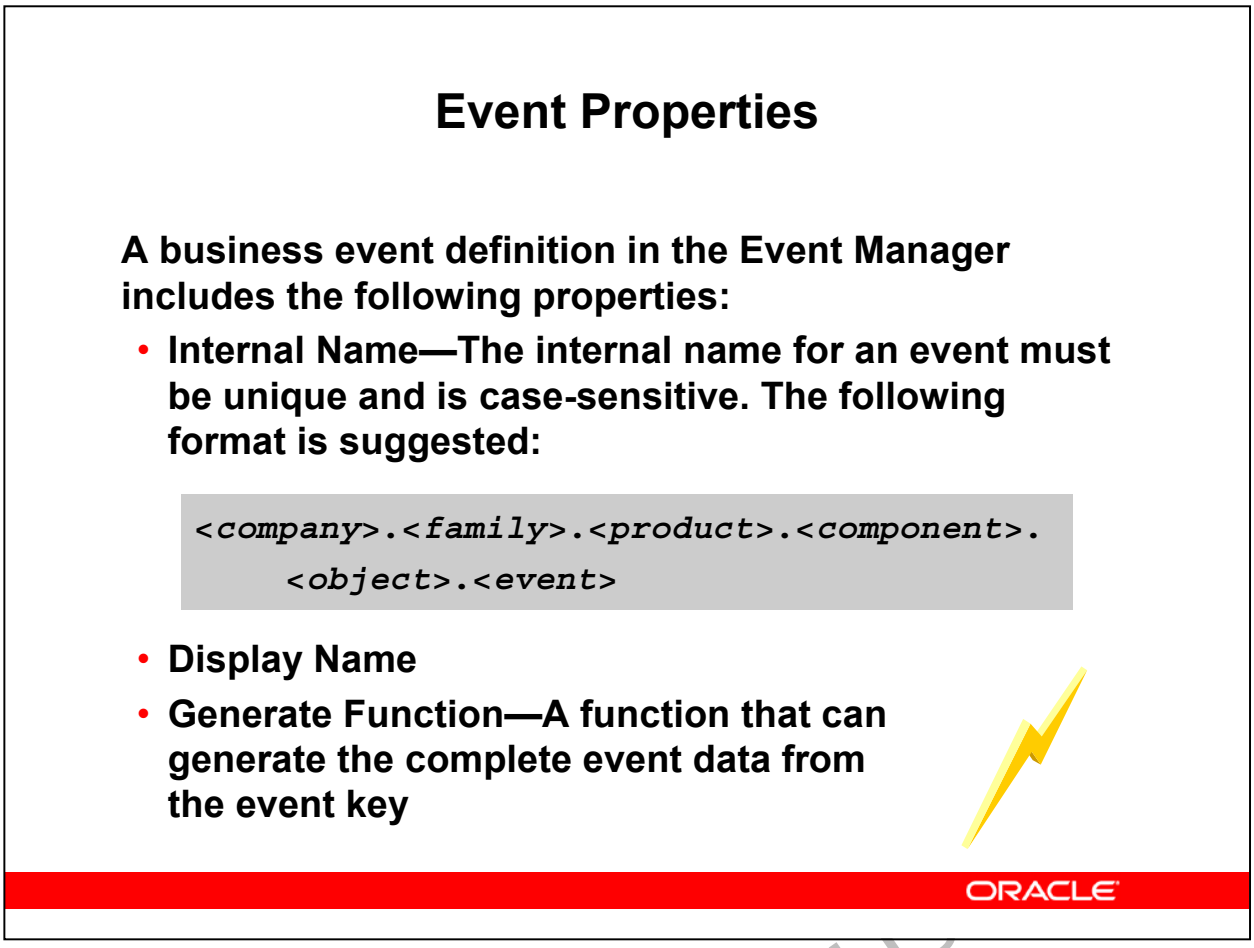
   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.You can also access the Events web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Events/Event Groups option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.
2. The Events page appears, displaying a list of existing events.

   Choose the Add Event button to open the Edit Event page.
3. Enter the internal name of the event in the Name field.
4. Enter a display name for the event.

5. Enter an optional description for the event.
6. In the Status field, select Enabled or Disabled as the event status. If you disable an event, it still remains in the Events list for reference, but you cannot use the event in active subscriptions.
7. If you are defining an event that occurs on your local system, enter the Generate function for the event.
8. If you use a program to create event definitions automatically, the Event Manager displays owner information set by that program in the Owner Name and Owner Tag fields. You can use the Edit Event page to update this information manually if necessary.
9. Choose the Submit button to save the event.

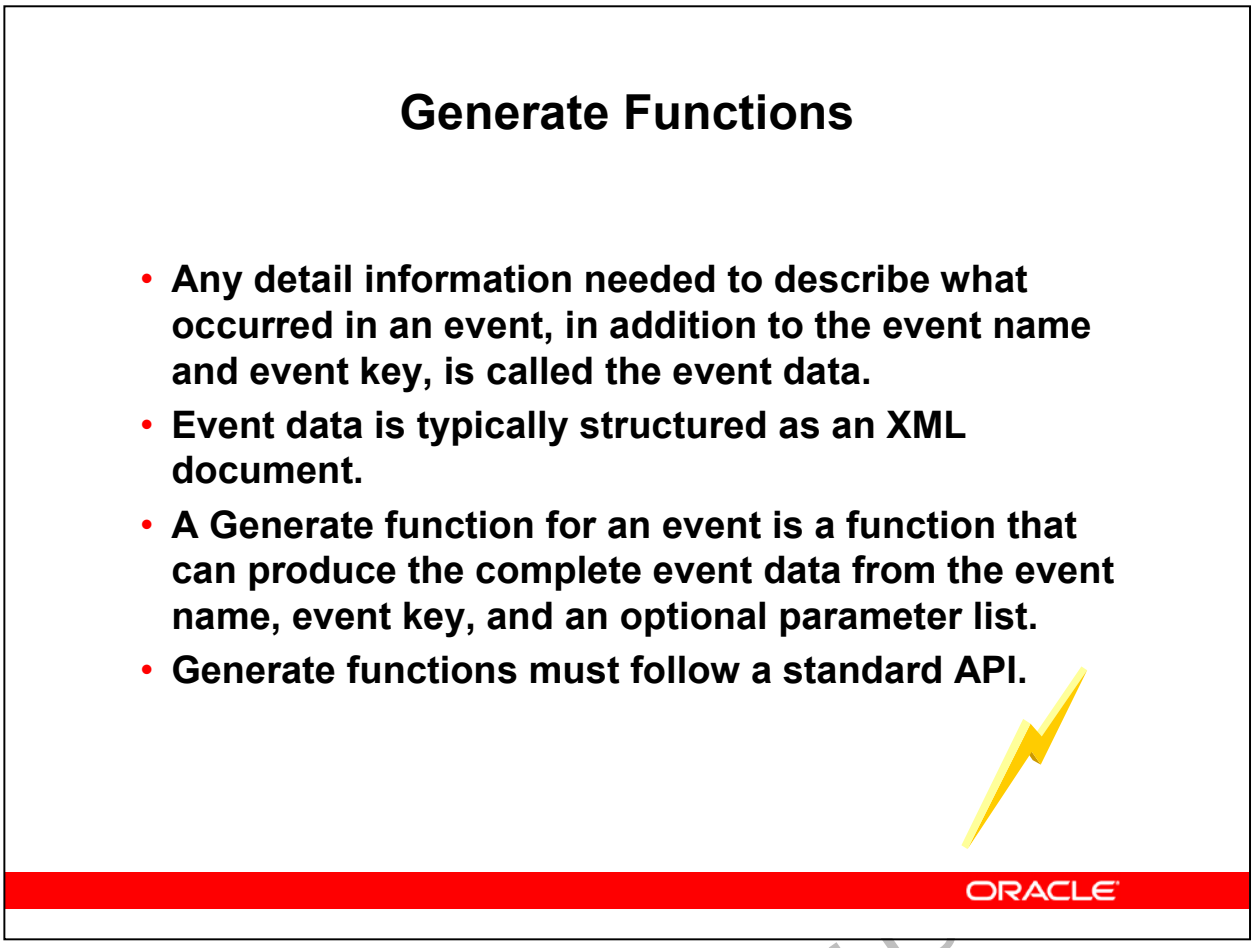


Event Properties

Some examples of event names are:

- oracle.apps.wf.event.event.create
- oracle.apps.wf.event.event.delete
- oracle.apps.wf.event.subscription.create

These events are predefined events within the Business Event System. For more information, refer to the Predefined Workflow Events chapter in the *Oracle Workflow Guide*.



Generate Functions

- The event data is stored as a character large object (CLOB).
- The application where an event occurs can include the event data when raising the event to the Event Manager.
- If the application will not provide the event data, you should specify a Generate function for the event.
- The Event Manager checks each subscription before executing it to determine whether the subscription requires the event data. If the event data is required but is not already provided, the Event Manager calls the Generate function for the event to produce the event data.
- If the event data is required but no Generate function is defined for the event, Oracle Workflow creates a default set of event data using the event name and event key.

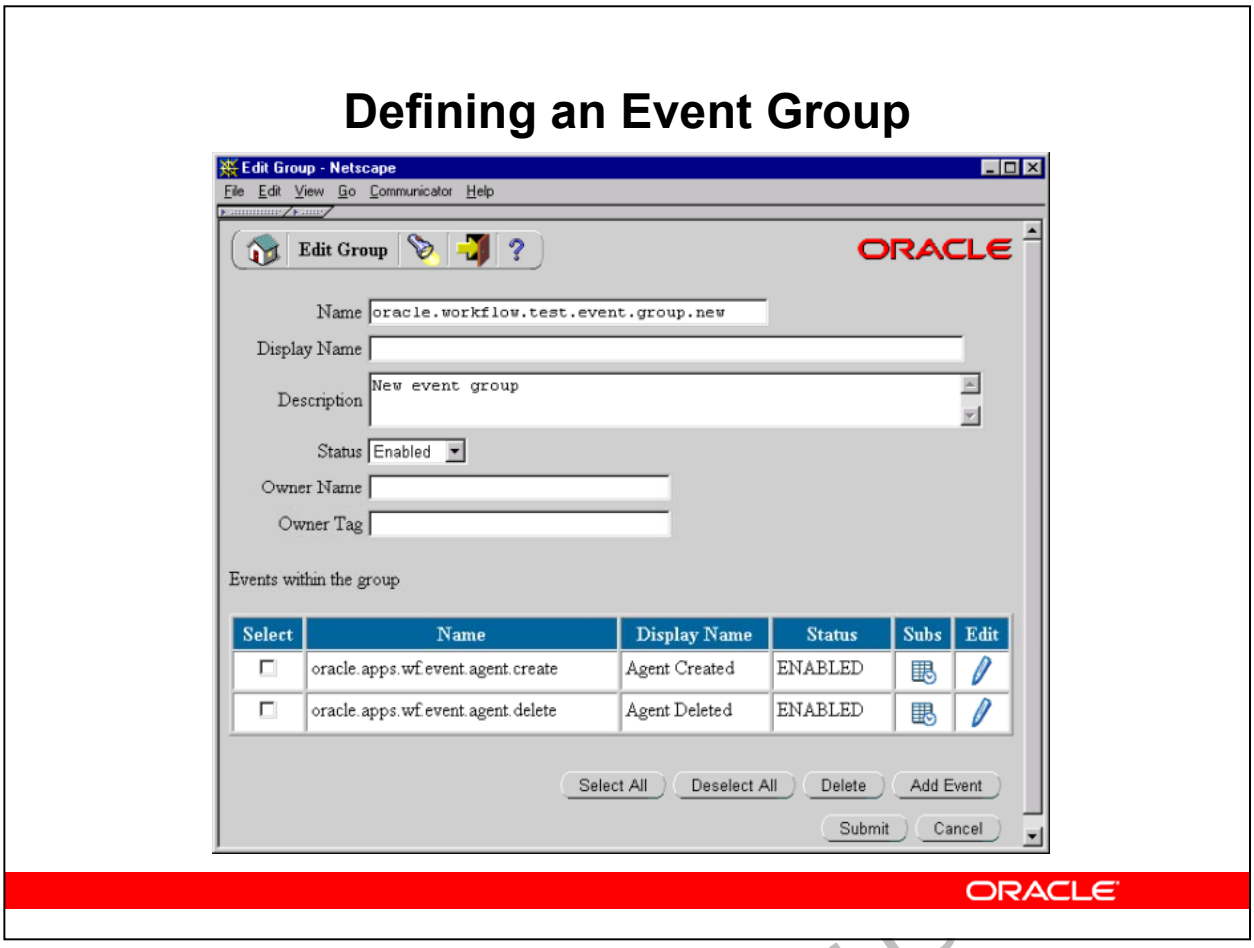
Event Groups

- **Event groups let you associate any events you want with each other and reference them as a group in event subscriptions.**
- **An event group is a type of event composed of a set of individual member events.**
- **Once you have defined an event group, you can register a subscription to the group rather than having to create separate subscriptions for each individual event within it.**
- **The subscription will be executed whenever any one of the group's member events occurs.**

ORACLE

Event Groups

- The internal names of event groups should follow the same format as the names of individual events.
- Event groups cannot be used to raise events. You must raise each event individually.
- An event group can contain only individual events as its members. It cannot contain another group.



To Define an Event Group:

1. Use a web browser to connect to the Events web page.

   For standalone Oracle Workflow, connect to the following URL: <*webagent*>/wf_event_html.listevents

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

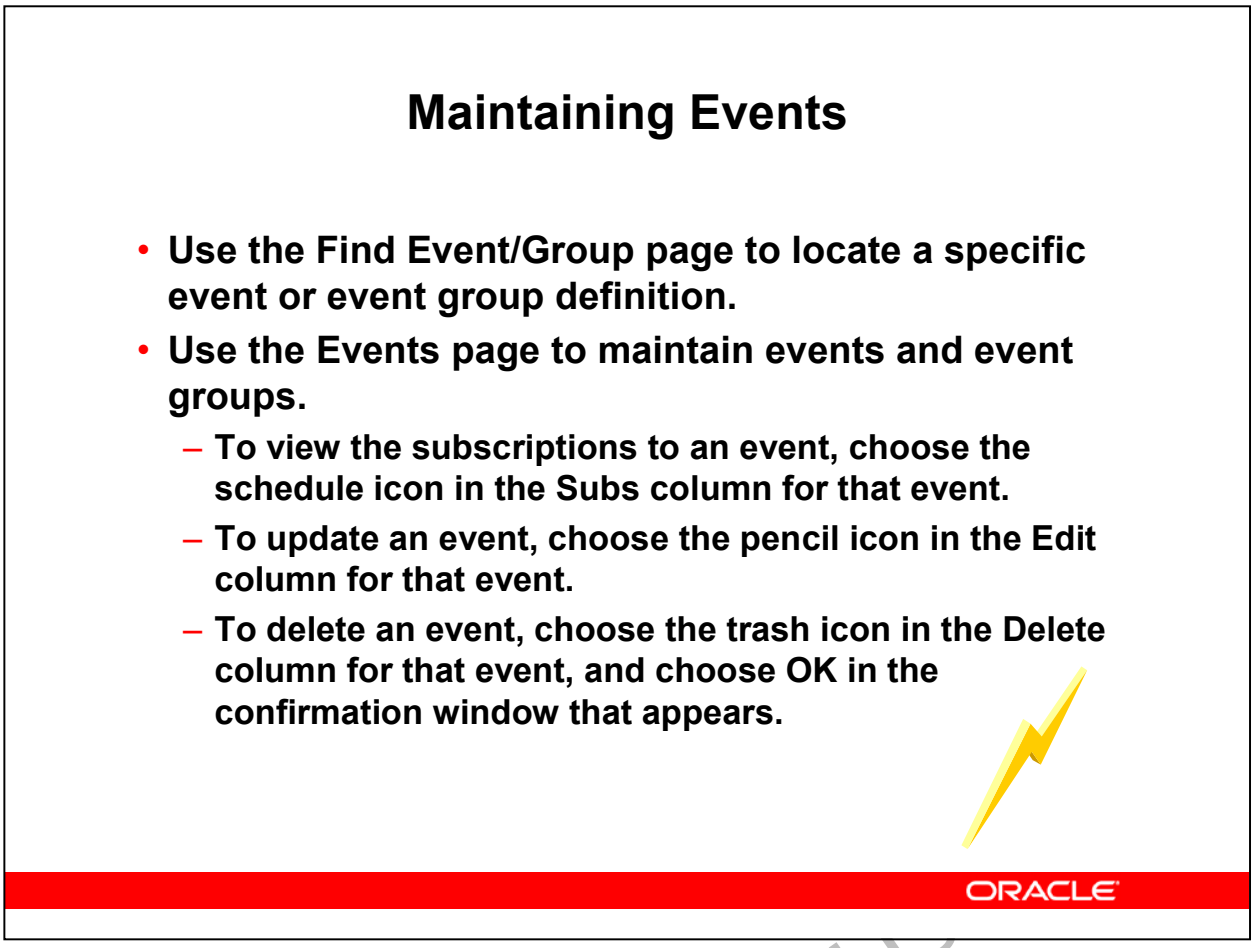
   You can also access the Events web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Events/Event Groups option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.
2. The Events page appears, displaying a list of existing events. Choose the Add Group button to open the Edit Group page.
3. Enter the internal name of the event group in the Name field.

4. Enter a display name for the event group.
5. Enter an optional description for the event group.
6. In the Status field, select Enabled or Disabled as the event group status. If you disable an event group, it still remains in the Events list for reference, but you cannot use the event group in active subscriptions.
7. If you use a program to create event definitions automatically, the Event Manager displays owner information set by that program in the Owner Name and Owner Tag fields. You can use the Edit Group page to update this information manually if necessary.
8. Choose the Submit button to save the event group.

   After you save the event group definition, the Edit Group page displays the list of member events for that group.
9. To add a member event to the group, choose the Add Event button.
10. In the Add to Group page that appears, enter search criteria to locate the event you want to add. The search criteria are:
    - Name—enter the internal name of the event you want to add.
    - Display Name—enter the display name of the event you want to add.
    - Status—choose Enabled or Disabled as the status of the event you want to add. Choose Any to search for events of any status.
11. Choose the Go button. The Add to Group page displays a list of events that match your search criteria.
12. Select the event or events that you want to add to your event group.
13. Choose the Add button to add the selected events to your event group. The Edit Group page appears, displaying the updated list of event group members.
14. Choose the Submit button to save the event group definition.
15. If you want to remove a member event from the group, select the event or events you want to delete in the Edit Group page.
16. Choose the Delete button to remove the selected events from your event group. The Edit Group page displays the updated list of event group members.



Maintaining Events

Finding Events

To access the Find Event/Group web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.findevent

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find Event/Group web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Event/Groups option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

In the Find Event/Group page, you can search for events by entering the following criteria:

- Name
- Display Name

- Status
- Type

Modifying Events

To access the Events web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.listevents

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Events web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Events/Event Groups option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

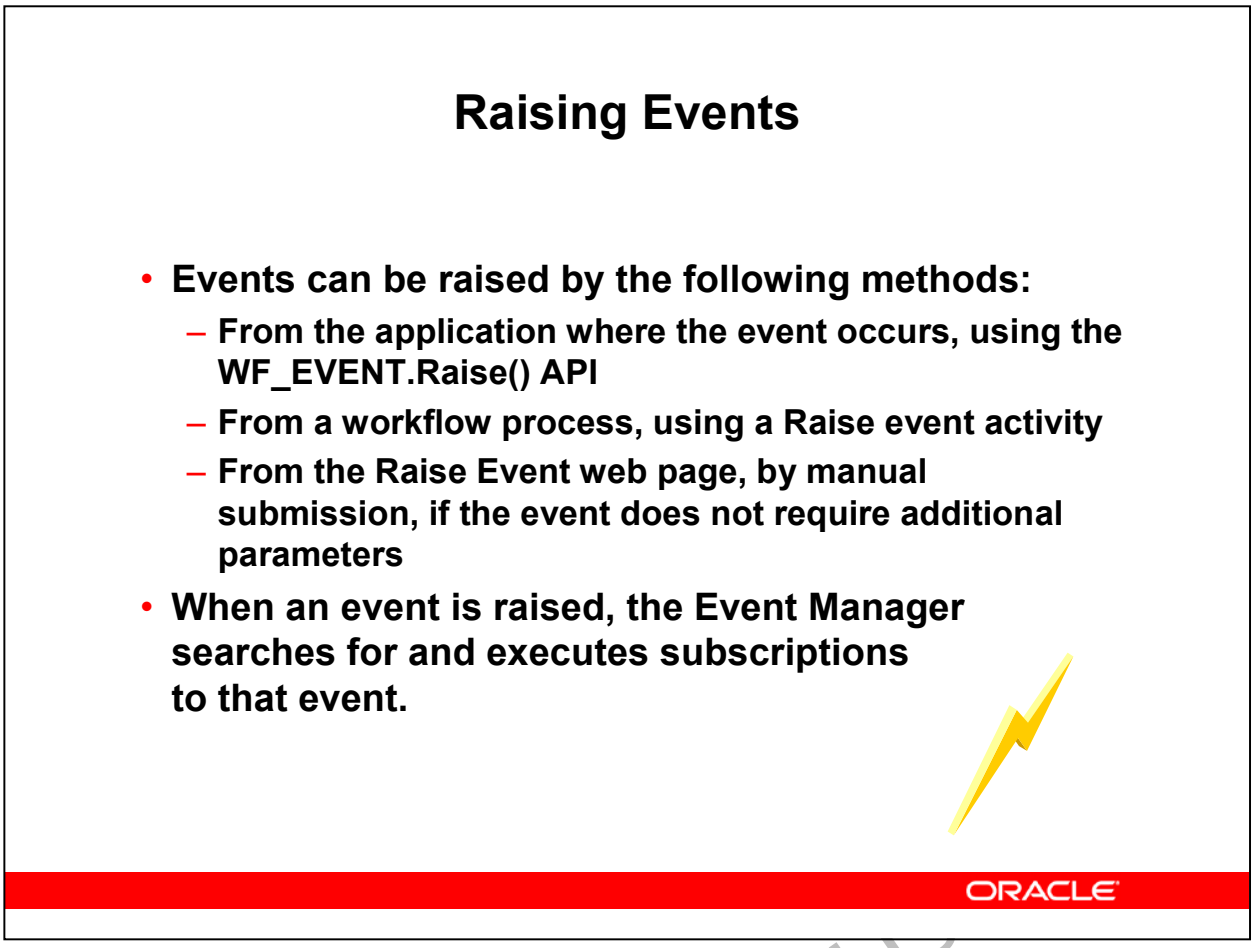
Subscriptions to an Event

For events that do not have any subscriptions yet, a blank schedule icon appears. For events that do have subscriptions referencing them, a full schedule icon appears.

You can begin defining a new subscription on the event by choosing the Add Subscription button in the Event Subscriptions page. The Edit Subscription page appears with the event name automatically entered in the Event Filter field.

Deleting Events

You can only delete events that do not have any subscriptions referencing them and that are not members of any event group.



Raising Events

To raise an event, you must provide the event name and an event key to identify the instance of the event. You can optionally provide a CLOB containing the event data.

Also, if you use the WF_EVENT.Raise API, you can optionally provide a list of additional header parameters for the event in the WF_PARAMETER_LIST_T format. If you use a Raise event activity, you can define additional header parameters as activity attributes for the activity. If the event is sent to a workflow process, any parameters included in the parameter list of the event message are set as item attribute values for the workflow process.

Note: You cannot specify additional parameters using the Raise Event page.

When an event is raised, various event header properties, the event parameters, and the event data are stored together in a special datatype structure to form the event message. The event message is the format in which Oracle Workflow communicates events among systems.

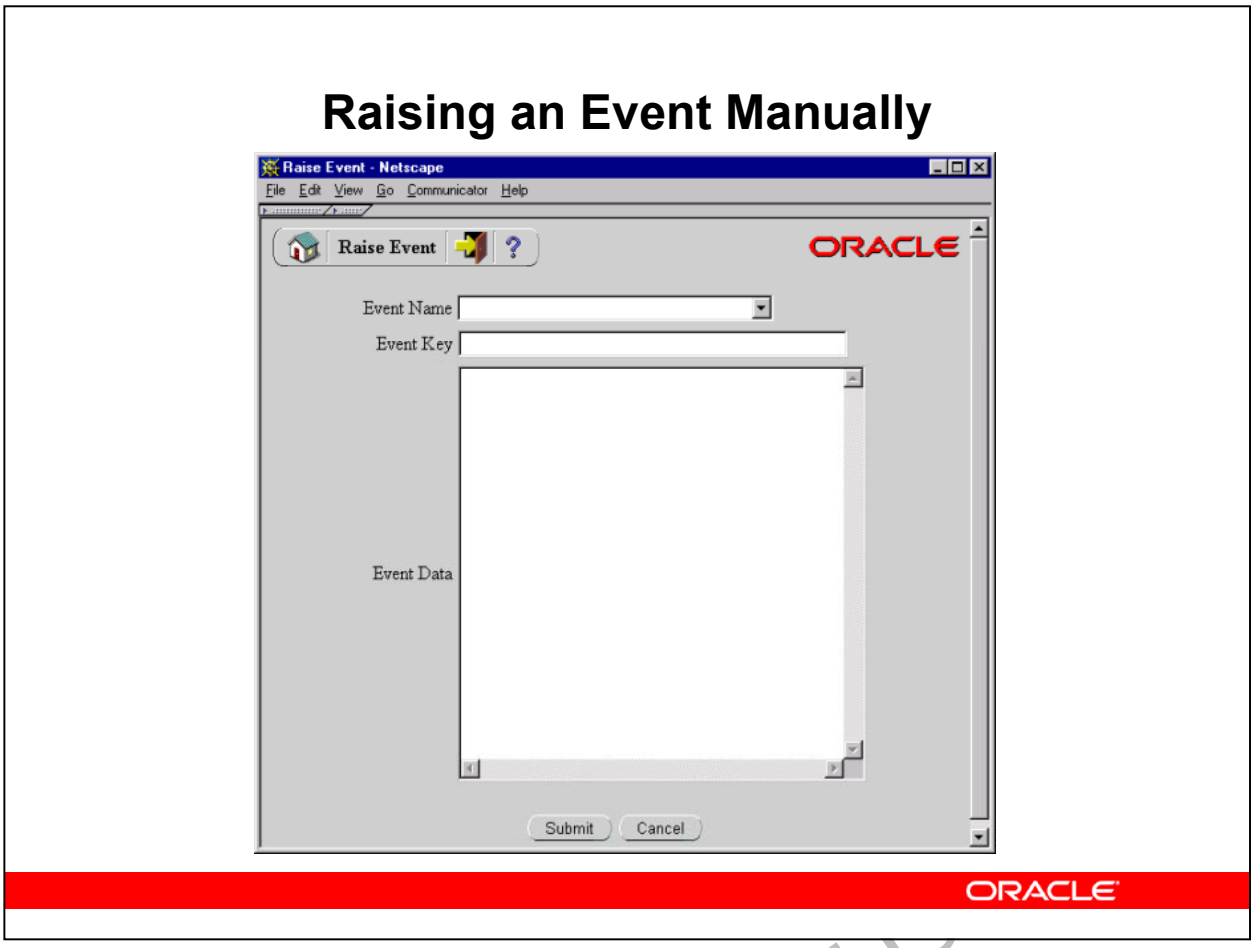
Example

The following example code shows how to use the WF_EVENT.Raise API to raise an event from an application.

```
declare
  l_xmldocument varchar2(32000);
  l_eventdata clob;
  l_parameter_list wf_parameter_list_t;
  l_message varchar2(10);

begin

/*
** If the complete event data is easily available,
** we can optionally test if any subscriptions to
** this event require it (rule data = Message).
*/

  l_message := wf_event.test('<EVENT_NAME>');

/*
** If we do require the complete event data, and we
** have the data now, set it; else we can just rely
** on the Event Generate Function callback code.
** Then Raise the Event with the required
** parameters.
*/

 if l_message = 'MESSAGE' then
   if l_xmldocument is not null then
     dbms_lob.createtemporary(l_eventdata, FALSE,
       DBMS_LOB.CALL);
     dbms_lob.write(l_eventdata, length(l_xmldocument), 1,
       l_xmldocument);
     -- Raise the Event with the message
     wf_event.raise(
       p_event_name => '<EVENT_NAME>',
       p_event_key  => '<EVENT_KEY>',
       p_event_data => l_eventdata,
       p_parameters => l_parameter_list);
   else
     -- Raise the Event without the message
     wf_event.raise(
       p_event_name => '<EVENT_NAME>',
```

```
        p_event_key  => '<EVENT_KEY>',
        p_parameters => l_parameter_list);
    end if;
  elsif
    l_message = 'KEY' then
      -- Raise the Event
      wf_event.raise(
        p_event_name => <EVENT_NAME>,
        p_event_key  => <EVENT_KEY>,
        p_parameters => l_parameter_list);
  end if;

/*
** Up to your own custom code to commit the
** transaction
*/

  commit;

/*
** Up to your own custom code to handle any major
** exceptions
*/

exception
when others then
null;
end;
```

Raising an Event Manually



To Raise an Event:

1. Use a web browser to connect to the Raise Event page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_html.entereventdetails

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the Raise Event web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Raise Business Event option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.
2. The Raise Event page appears.
3. In the Event Name field, select the event that you want to raise.
4. Enter an event key that uniquely identifies this instance of the event.

5. Optionally enter event data to describe what occurred in the event.

   Note: The maximum length of the data you can enter in the Event Data field is 32 kilobytes. If the event data exceeds 32 Kb, you should assign a Generate function in the event definition to generate the event data, rather than entering the data directly in the Event Data field.

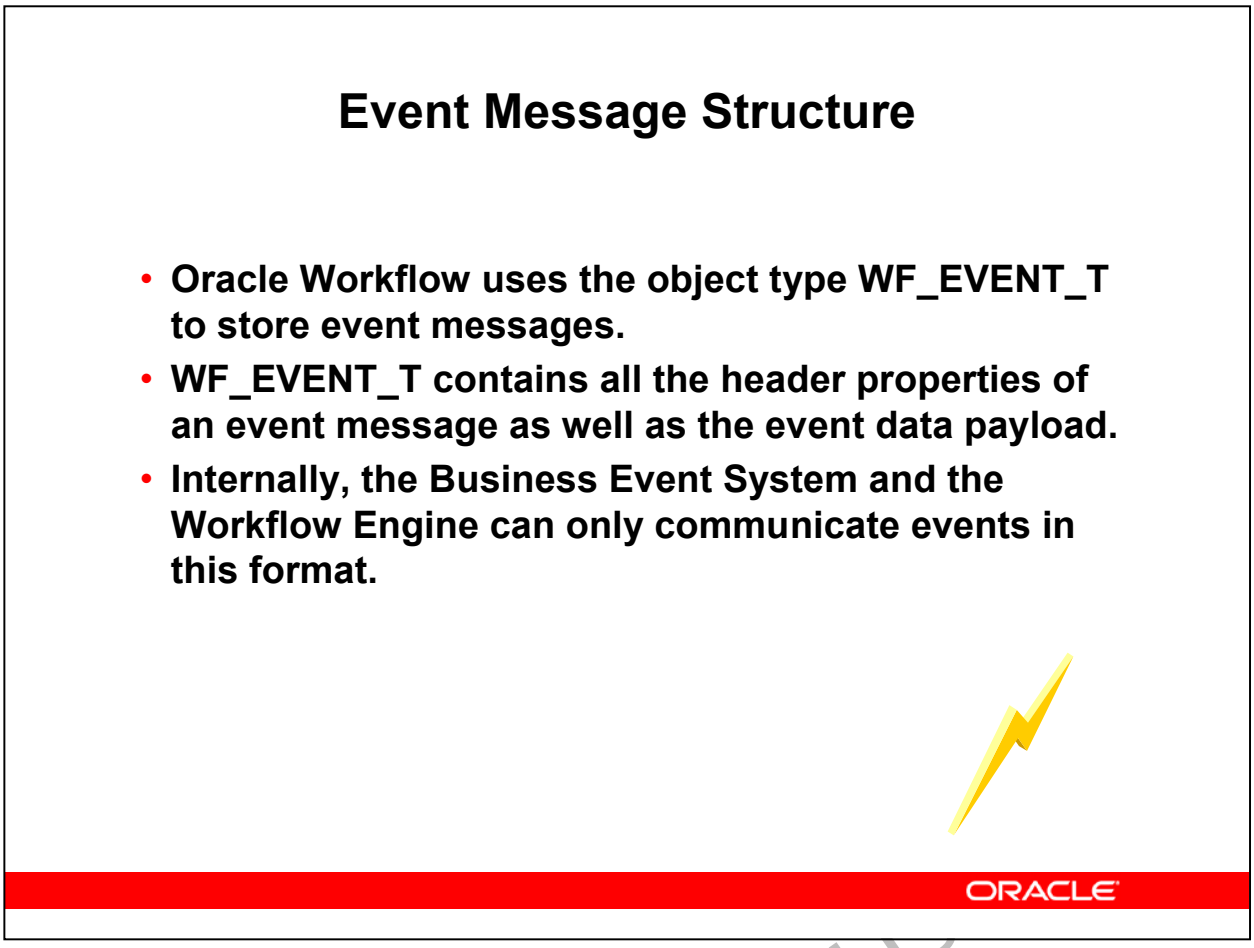
6. Choose the Submit button to raise the event to the Event Manager. Oracle Workflow raises the event and displays a confirmation message with the event name and event key. Choose the OK button.

Business Event System Datatypes

Oracle Workflow uses the following abstract datatypes to model the structure and behavior of Business Event System data.

- **Event message structure: WF_EVENT_T**
- **Agent structure: WF_AGENT_T**
- **Parameter list structure: WF_PARAMETER_LIST_T**
- **Parameter structure: WF_PARAMETER_T**

ORACLE



WF_EVENT_T Attributes

Attribute Name: PRIORITY

Datatype: NUMBER

Description: The priority with which the message recipient should dequeue the message. A smaller number indicates a higher priority.

Attribute Name: SEND_DATE

Datatype: DATE

Description: The date and time when the message is available for dequeuing. The send date can be set to the system date to indicate that the message is immediately available for dequeuing, or to a future date to indicate future availability.

Attribute Name: RECEIVE_DATE

Datatype: DATE

Description: The date and time when the message is dequeued by an agent listener.

Attribute Name: CORRELATION_ID

Datatype: VARCHAR2(240)

Description: A correlation identifier that associates this message with other messages. This attribute is initially blank but can be set by a function. If a value is set for the correlation ID, then that value is used as the item key if the event is sent to a workflow process.

Attribute Name: PARAMETER_LIST

Datatype: WF_PARAMETER_ LIST_T

Description: A list of additional parameter name and value pairs.

Attribute Name: EVENT_NAME

Datatype: VARCHAR2(240)

Description: The internal name of the event.

Attribute Name: EVENT_KEY

Datatype: VARCHAR2(240)

Description: The string that uniquely identifies the instance of the event.

Attribute Name: EVENT_DATA

Datatype: CLOB

Description: A set of additional details describing what occurred in the event. The event data can be structured as an XML document. A Generate function can be defined for the event to produce the event data.

Attribute Name: FROM_AGENT

Datatype: WF_AGENT_T

Description: The agent from which the event is sent. For locally raised events, this attribute is initially null.

Attribute Name: TO_AGENT

Datatype: WF_AGENT_T

Description: The agent to which the event should be sent (the message recipient).

Attribute Name: ERROR_ SUBSCRIPTION

Datatype: RAW(16)

Description: If an error occurs while processing this event, this is the subscription that was being executed when the error was encountered.

Attribute Name: ERROR_MESSAGE

Datatype: VARCHAR2(4000)

Description: An error message that the Event Manager generates if an error occurs while processing this event.

Attribute Name: ERROR_STACK

Datatype: VARCHAR2(4000)

Description: An error stack of arguments that the Event Manager generates if an error occurs while processing this event. The error stack provides context information to help you locate the source of an error.

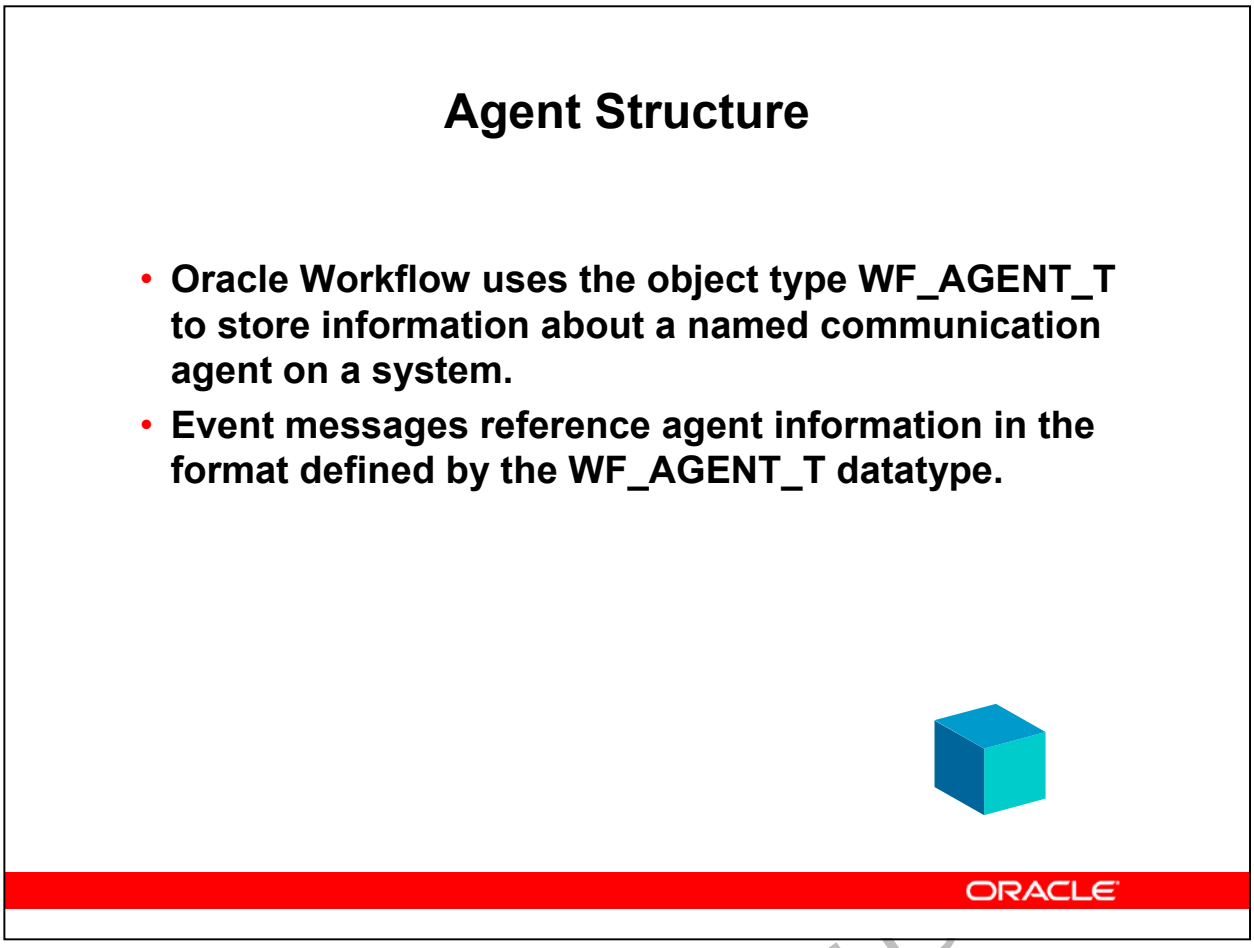
The WF_EVENT_T object type also includes the following methods, which you can use to retrieve and set the values of its attributes.

Note: You must call the Initialize method before you can perform any further manipulation on a new WF_EVENT_T object.

- Initialize
- getPriority
- getSendDate
- getReceiveDate
- getCorrelationID
- getParameterList
- getEventName
- getEventKey
- getEventData
- getFromAgent
- getToAgent
- getErrorSubscription
- getErrorMessage
- getErrorStack
- setPriority
- setSendDate
- setReceiveDate
- setCorrelationID
- setParameterList
- setEventName
- setEventKey
- setEventData
- setFromAgent
- setToAgent
- setErrorSubscription
- setErrorMessage
- setErrorStack

- Content
- Address
- AddParameterToList
- GetValueForParameter

For more information, refer to the Oracle Workflow APIs chapter in the *Oracle Workflow Guide*.



WF_AGENT_T Attributes

Attribute Name: NAME
Datatype: VARCHAR2(30)
Description: The name of the agent.

Attribute Name: SYSTEM
Datatype: VARCHAR2(30)
Description: The system where the agent is located.

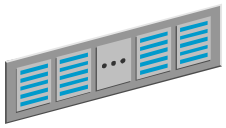
The WF_AGENT_T object type also includes the following methods, which you can use to retrieve and set the values of its attributes.

- getName
- getSystem
- setName
- setSystem

For more information, refer to the Oracle Workflow APIs chapter in the *Oracle Workflow Guide*.

Parameter List Structure

- **Oracle Workflow uses the named varying array WF_PARAMETER_LIST_T to store a list of parameters in a form that can be included in an event message.**
- **WF_PARAMETER_LIST_T allows custom values to be added to the WF_EVENT_T event message object.**
- **The WF_PARAMETER_LIST_T datatype can include up to 100 parameter name and value pairs.**



ORACLE

WF_PARAMETER_LIST_T

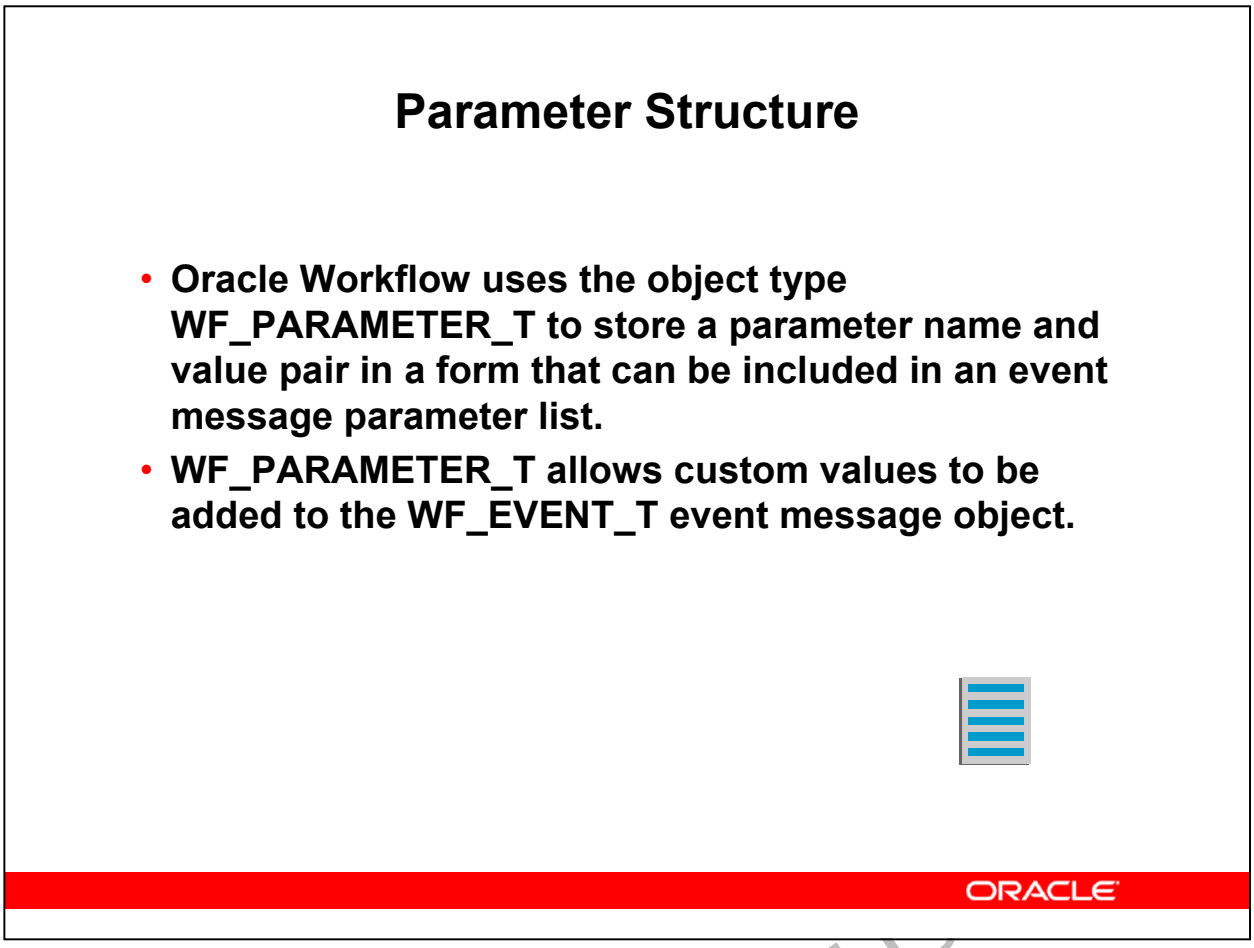
Maximum size: 100

Element datatype: WF_PARAMETER_T

When WF_PARAMETER_LIST_T is used within a WF_EVENT_T structure, you can use the following WF_EVENT_T methods to add and retrieve parameters in the list.

- AddParameterToList
- GetValueForParameter

When an event message that includes a parameter list is sent to a workflow process, the parameters in the list are created as item attributes for the workflow process.



WF_PARAMETER_T Attributes

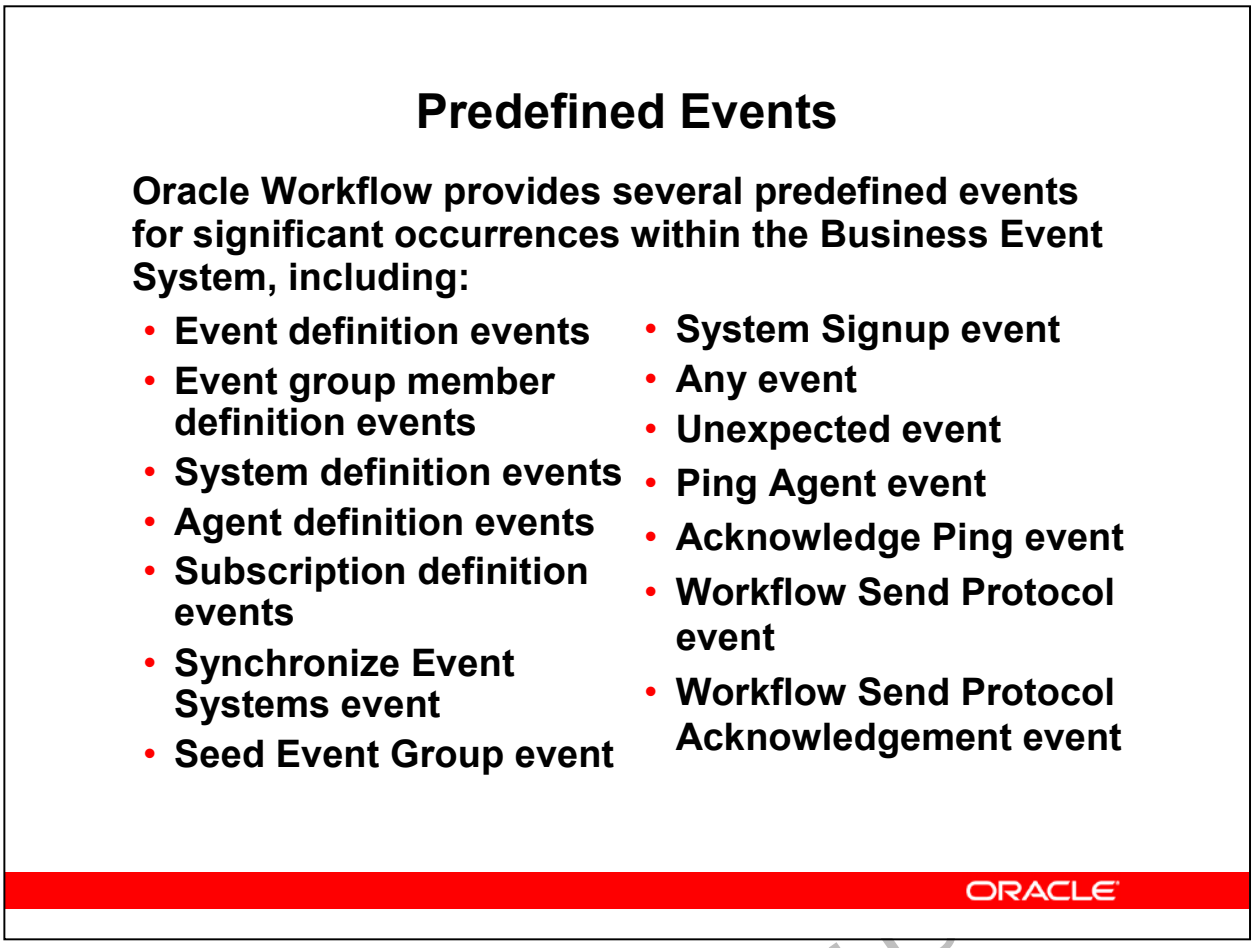
Attribute Name: NAME
Datatype: VARCHAR2(30)
Description: The parameter name.

Attribute Name: VALUE
Datatype: VARCHAR2(2000)
Description: The parameter value.

The WF_PARAMETER_T object type also includes the following methods, which you can use to retrieve and set the values of its attributes.

- getName
- getValue
- setName
- setValue

For more information, refer to the Oracle Workflow APIs chapter in the *Oracle Workflow Guide*.



Predefined Events

You can define subscriptions to the predefined Workflow events for replication, validation, or other purposes. Some predefined events are referenced by default subscriptions that are created automatically when you install Oracle Workflow.

Business Event System Object Definition Events

These predefined events are raised whenever a Business Event System object definition is created, updated, or deleted. These events are used for replication of Business Event System data from one system to another.

Individual Event or Event Group Definition Events

- Event Created
- Event Updated
- Event Deleted

Event Group Member Definition Events

- Event Group Creation
- Event Group Updated
- Event Group Deleted

System Definition Events

- System Created
- System Updated
- System Deleted

Agent Definition Events

- Agent Created
- Agent Updated
- Agent Deleted

Event Subscription Definition Events

- Subscription Created
- Subscription Updated
- Subscription Deleted

Synchronize Event Systems Event

This event is used to synchronize the Event Manager data on the local system with another system.

Seed Event Group

This event group contains events used for automatic replication of Business Event System objects from one system to another. The group includes all the event, event group, system, agent, and subscription definition events, as well as the Synchronize Event Systems event.

System Signup Event

This event is used to sign up a destination system for receiving event messages from the source system.

Any Event

This event is raised implicitly when any other event is raised locally or received from an external source.

Unexpected Event

The Event Manager executes subscriptions to this event when another event is raised locally or received from an external source, but no subscription to that other event exists.

Ping Agent Events

- Ping Agent—This event is used in the Workflow Agent Ping/Acknowledge process to ping inbound agents.
- Acknowledge Ping—This event is used in the Workflow Agent Ping/Acknowledge process to acknowledge receipt of a Ping Agent event message.

Workflow Send Protocol Events

- Workflow Send Protocol—This event is used with the Workflow Send Protocol process to send the event message to an agent.
- Workflow Send Protocol Acknowledgement—This event is used with the Workflow Send Protocol process to acknowledge receipt of an event message.

For more information, refer to the Predefined Workflow Events chapter in the *Oracle Workflow Guide*.

Practice - Defining an Event

Overview

In this practice you will create an event definition in the Event Manager. After completing the practice, you should be able to see your event listed in the Events page.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.

- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.

- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility that includes Event Manager functionality. The username you use to log in should have this responsibility assigned to it.

Tasks

1. Define an event in the Event Manager.

   Solution:

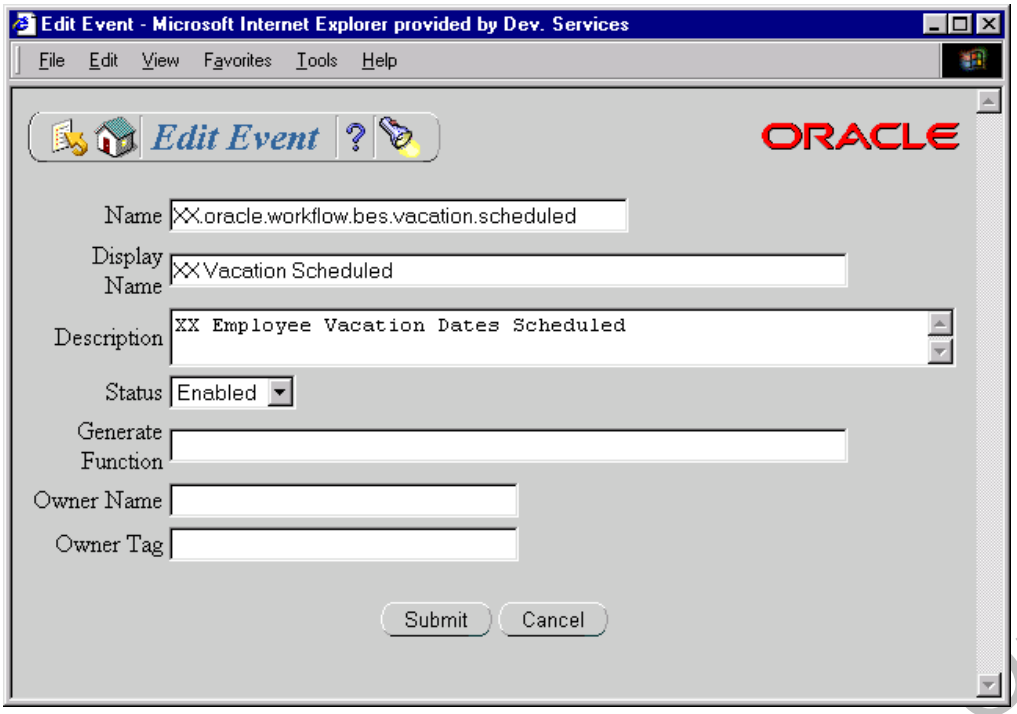
   1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

   2. Choose the Events link in standalone Oracle Workflow, or the Add Events/Event Groups link in Oracle Workflow embedded in Oracle E-Business Suite.

   3. In the Events page, choose the Add Event button to open the Edit Event page.

   4. In the Name field, enter *XX*.oracle.workflow.bes.vacation.scheduled as the internal name of the event.

5. In the Display Name field, enter *XX* Vacation Scheduled.
6. In the Description field, enter *XX* Employee Vacation Dates Scheduled.
7. In the Status field, select Enabled.
8. Leave the Generate Function, Owner Name, and Owner Tag fields blank.



9. Choose the Submit button to save the event.

Practice - Raising an Event

Overview

In this practice, you will raise an event manually from the Raise Event page. The event you raise, oracle.apps.wf.event.wf.send, should trigger a predefined subscription that launches a workflow process. The focus of this practice is on using the Raise Event page; however, you can optionally review the workflow process in the Workflow Monitor to confirm that the event was raised successfully.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.

- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.

- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility that includes Event Manager functionality. The username you use to log in should have this responsibility assigned to it.

Tasks

1. Raise the oracle.apps.wf.event.wf.send event from the Raise Event page.

   Solution:

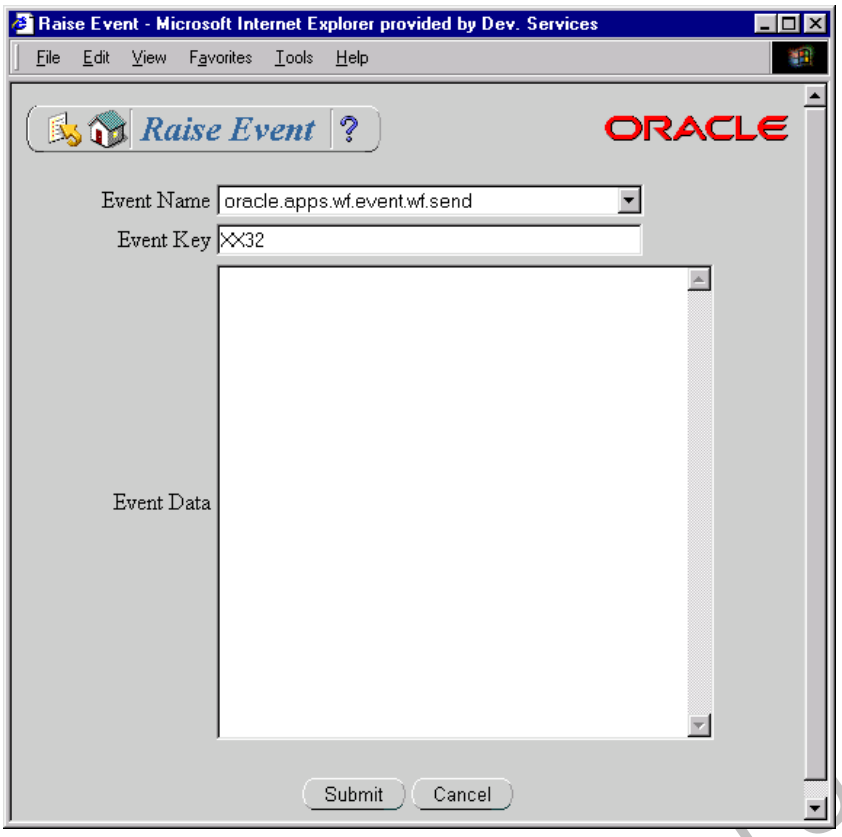
   1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

   2. Choose the Raise Event link in standalone Oracle Workflow, or the Raise Business Event link in Oracle Workflow embedded in Oracle E-Business Suite.

3. In the Event Name field, select oracle.apps.wf.event.wf.send. This is the predefined Workflow Send Protocol event used in the Workflow Send Protocol process.

4. In the Event Key field, enter a unique event key such as *XX*32.

5. For the purposes of this practice, since you will not perform further processing on this event, you do not need to enter any event data. Leave the Event Data field blank.



6. Choose the Submit button to raise the event.

7. Choose OK in the confirmation window.

   When the oracle.apps.wf.event.wf.send event is raised on the local system, it triggers a predefined subscription that sends the event to the Workflow Send Protocol process. However, since you did not specify a recipient for the event message in the subscription, the process will simply complete without performing any further processing.

8. You can optionally review the status of the process in the Workflow Monitor by choosing the Find Processes link from the Workflow home page and searching for the process with the Workflow Send Protocol item type and with your event key as the item key.

Summary

In this lesson, you should have learned how to:

- **Define events.**
- **Describe the abstract datatypes used by the Business Event System.**
- **Define event groups.**
- **Raise events.**
- **Use predefined Oracle Workflow events.**

ORACLE

Systems and Agents

Chapter 4

Objectives

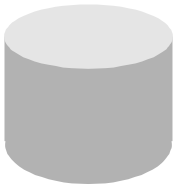
After completing this lesson, you should be able to do the following:

- **Define systems.**
- **Define communication agents on systems.**
- **Associate queues and queue handlers with agents.**
- **Review event messages on local agents.**
- **Sign up systems with each other for event message propagation.**

ORACLE

Systems

- A system is a logically isolated software environment such as a host machine or database instance.
- Each system to or from which you will communicate events must be defined in the Event Manager.
  - Local system
  - External Workflow-enabled systems
  - External non-Workflow systems



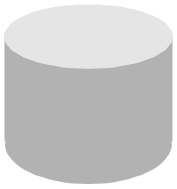
ORACLE

Systems

Systems that will communicate events with each other must store each other's system definitions in order to address event messages to each other.

Local System

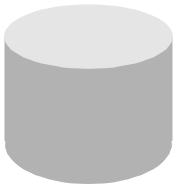
- When you install Oracle Workflow in a database, that database is automatically defined as a system in the Event Manager.
- The system name is set to the database global name.
- This system is automatically set as the local system in the Global Workflow Preferences page.



ORACLE

External Systems

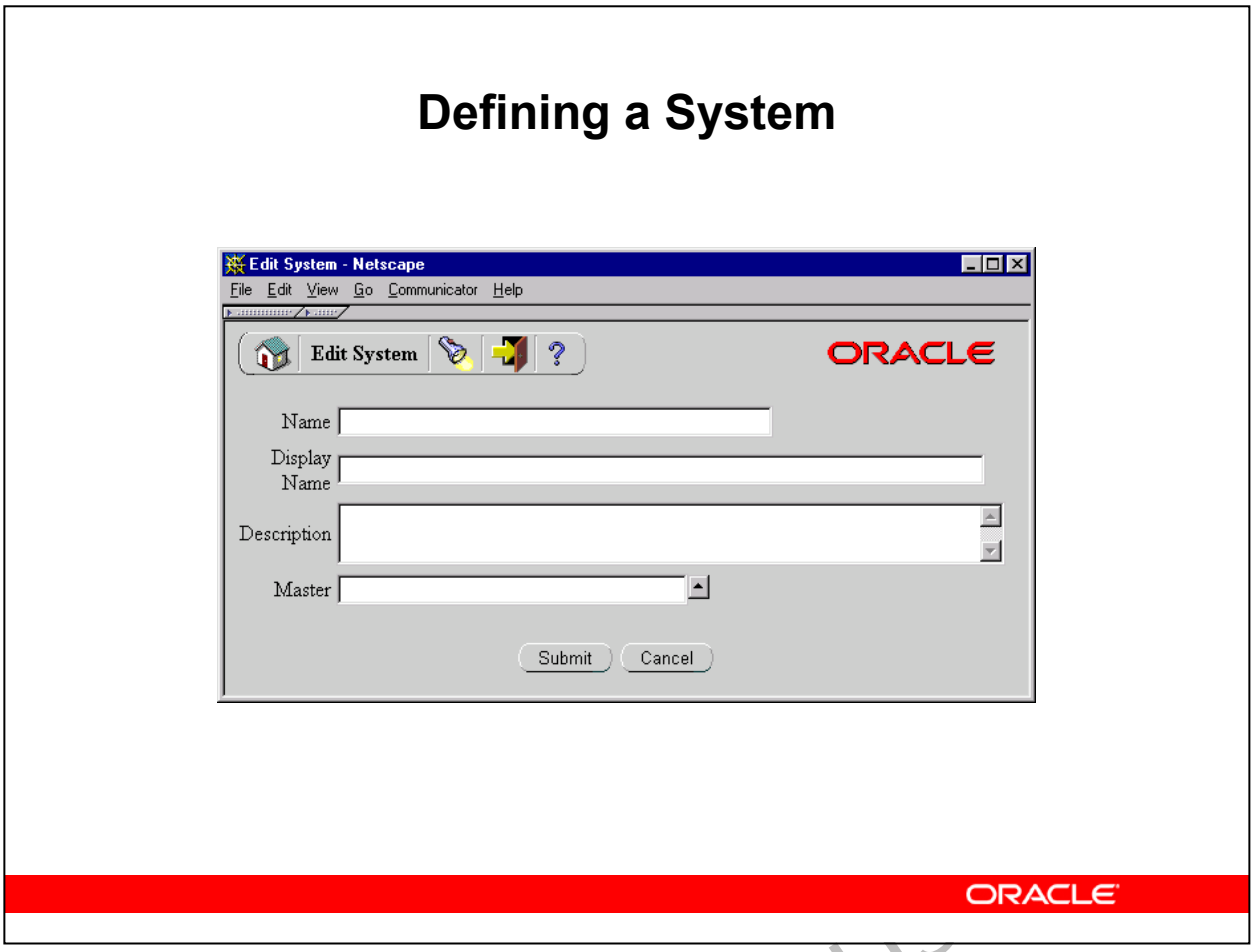
- **Workflow-enabled: For communication with another Workflow-enabled system, use the system signup procedure to automatically copy the system definition for the other system into the Event Manager of your local system.**
- **Non-Workflow: For communication with a system that does not have Oracle Workflow installed, use the Edit System page to manually create a system definition for the other system in the Event Manager of your local system.**



ORACLE

External Systems

A non-Workflow system can be either an Oracle system that does not have Oracle Workflow installed or a non-Oracle system.



To Define a System:

1. Use a web browser to connect to the Systems web page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_ html.listsystems

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the Systems web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

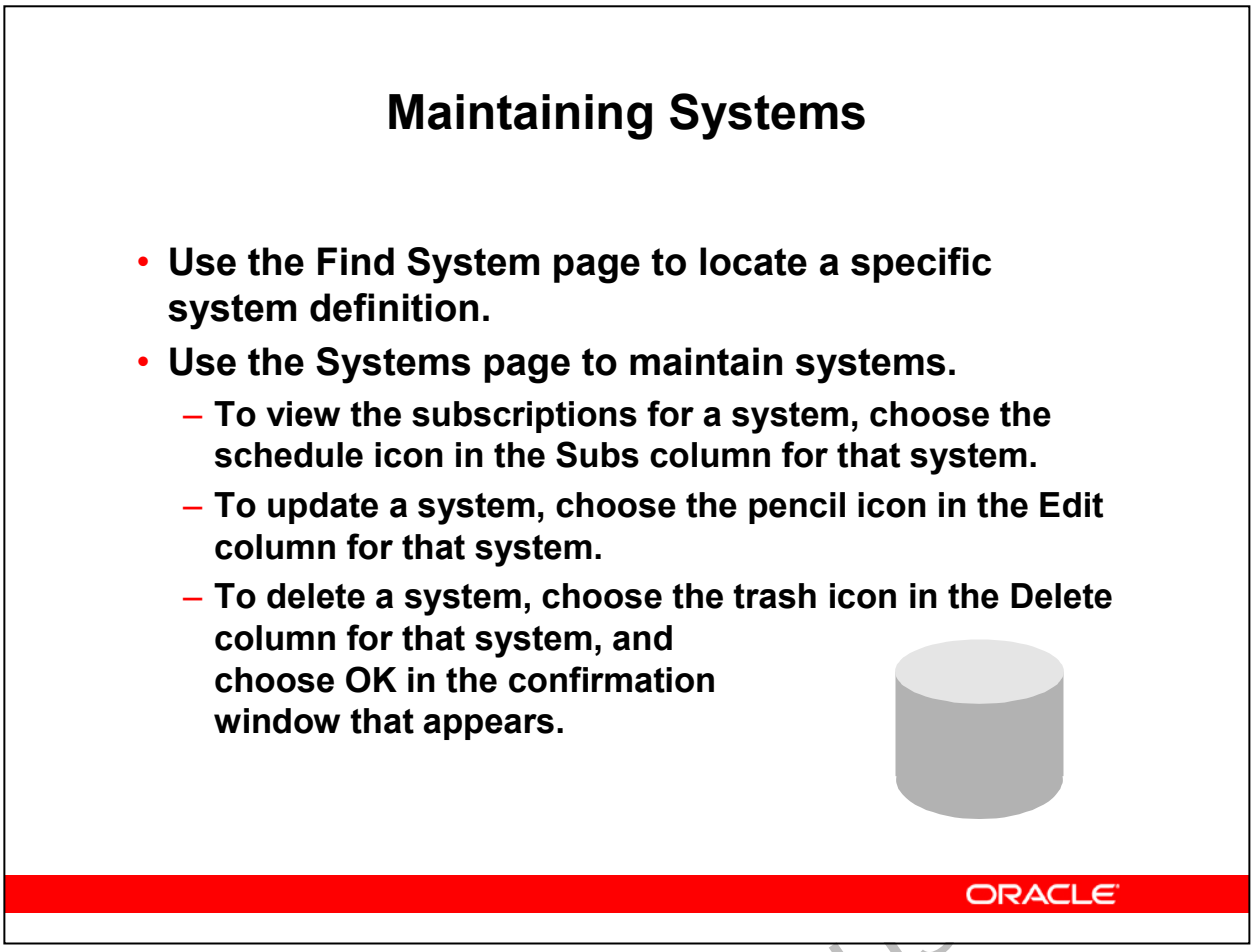
   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Systems option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.
2. The Systems page appears, displaying a list of existing systems. An asterisk marks the local system.

   Choose the Add System button to open the Edit System page.
3. Enter the internal name of the system in the Name field.

Note: The internal name must be all-uppercase and should not include any single or double quotation marks (' or ") or spaces.

4. Enter a display name for the system.
5. Enter an optional description for the system.
6. Optionally select a master system from which you want this system to receive Event Manager object definition updates. Click on the Master field's up arrow icon to display a list of systems from which to choose.
7. Choose the Submit button to save the system.



Maintaining Systems

Finding Systems

To access the Find System web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.findsystem

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find System web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Systems option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

In the Find System page, you can search for systems by entering the following criteria:

- Name
- Display Name

- Master

Modifying Systems

To access the Systems web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.listsystems

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Systems web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Systems option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

Subscriptions to a System

For systems that do not have any subscriptions yet, a blank schedule icon appears. For systems that do have subscriptions referencing them, a full schedule icon appears.

You can begin defining a new subscription for the system by choosing the Add Subscription button in the Event Subscriptions page. The Edit Subscription page appears with the system name automatically entered in the System field.

Deleting Systems

You can only delete systems that do not have any agents defined on them or any subscriptions referencing them.

Agents

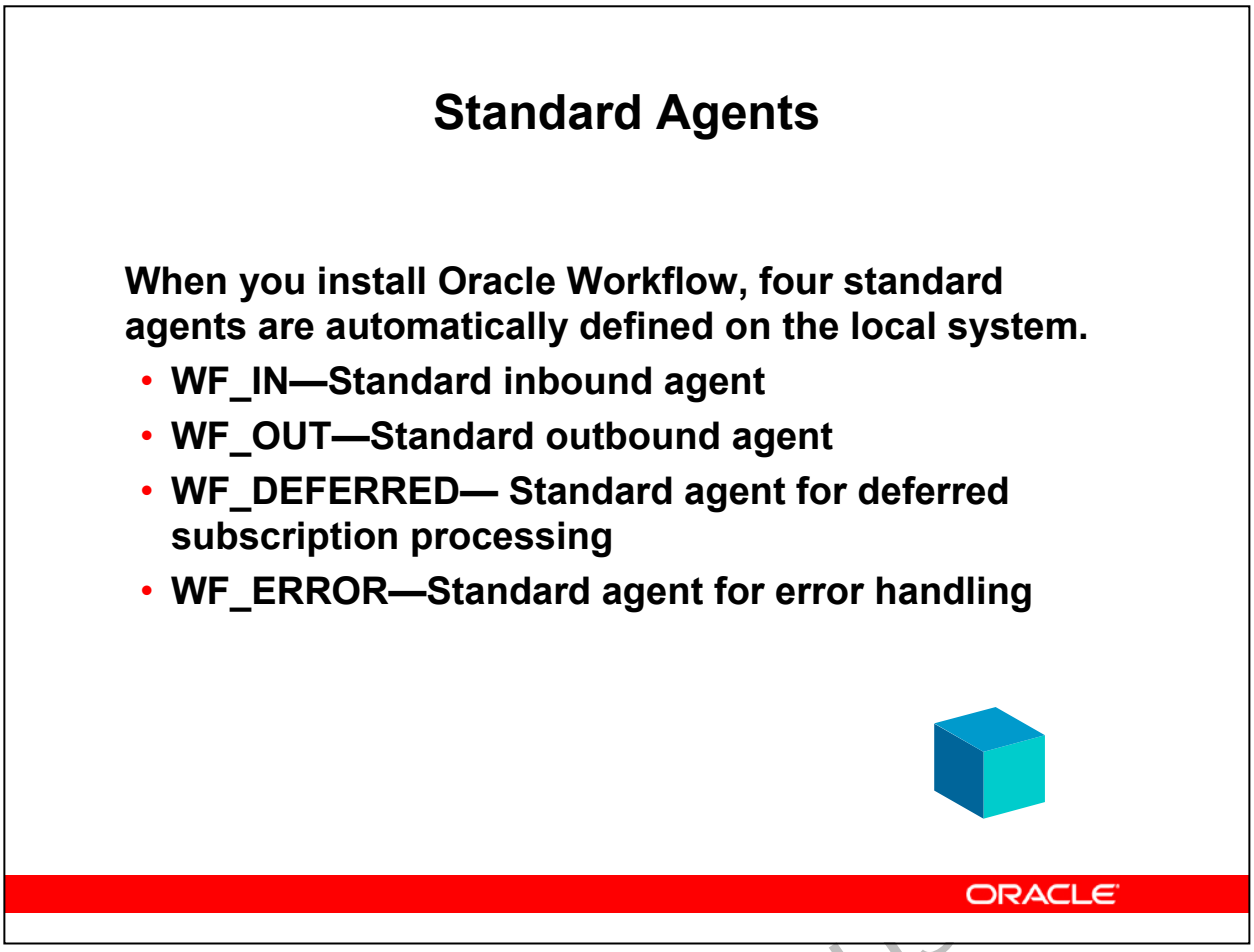
- An agent is a named point of communication within a system.
- A single system can have several different agents representing different communication alternatives, such as different protocols or propagation frequencies.
- Each agent that you will use to communicate events must be defined in the Event Manager.

ORACLE

Agents

- Communication within and between systems is accomplished by sending a message from one agent to another.
- Each agent on a Workflow-enabled system is associated with an AQ queue.
- The Business Event System interacts with the agent by enqueuing or dequeuing event messages on its queue.

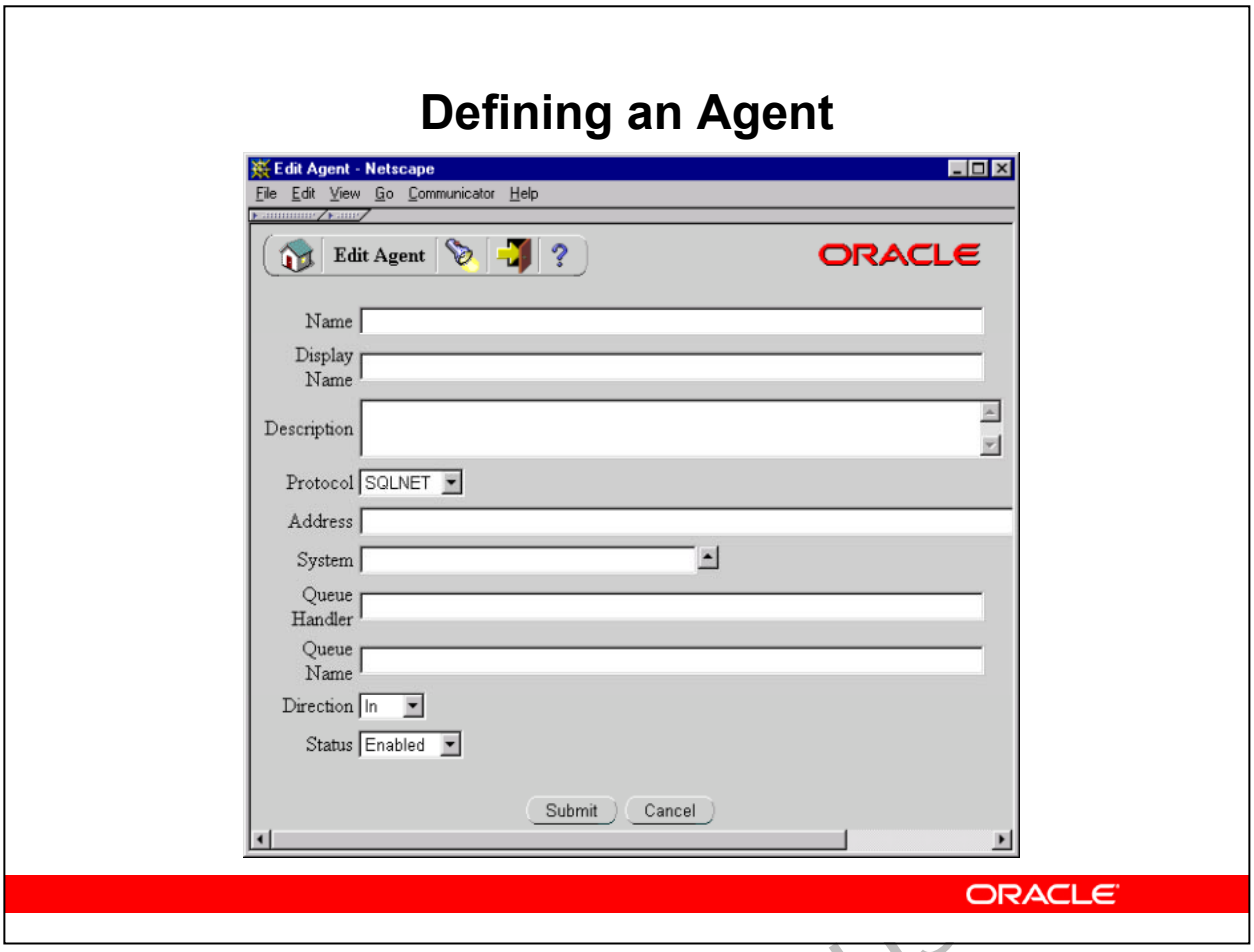
ORACLE



Standard Agents

Oracle Workflow provides WF_IN and WF_OUT as default agents that you can use to receive and send events without needing to define any custom agents. You can optionally define additional inbound and outbound agents to expand your event processing. For example, you can define custom agents for:

- Propagation of event messages by different protocols or with different frequencies
- Increased scalability
- Additional levels of service—for instance, if you have five outbound agents, and one agent stops functioning, you can still send messages from the other four agents



To Define an Agent:

1. Use a web browser to connect to the Agents web page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_ html.listagents

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the Agents web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Agents option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.

2. The Agents page appears, displaying a list of existing agents grouped by the system where they are located.

   Choose the Add Agent button to open the Edit Agent page.

3. Enter the internal name of the agent in the Name field. The agent's internal name must be unique within the agent's system.

   Note: The internal name must be all-uppercase and should not include any single or double quotation marks (' or ") or spaces.
4. Enter a display name for the agent.
5. Enter an optional description for the agent.
6. Select the message communication protocol that the agent supports.
7. If the agent supports inbound communication to its system, enter the address for the agent. The format of the address depends on the protocol you select.

   For agents that use the SQLNET protocol, the address must be in the following format to enable AQ propagation:

   <schema>.<queue>@<database link>

   <schema> represents the schema that owns the queue, *<queue>* represents the queue name, and *<database link>* represents the database link to the instance where the queue is located.
8. Enter the system in which the agent is defined. Click on the System field's up arrow icon to display a list of systems from which to choose.
9. Enter the queue handler for the agent.

   Note: You must enter the queue handler name in all uppercase.
10. Enter the name of the queue that the local system uses to interact with the agent. Since only the local system refers to this queue name, the queue name should be within the scope of this system, without requiring a database link. Use the following format to specify the queue name:

    <schema>.<queue>

    <schema> represents the schema that owns the queue, and *<queue>* represents the queue name.

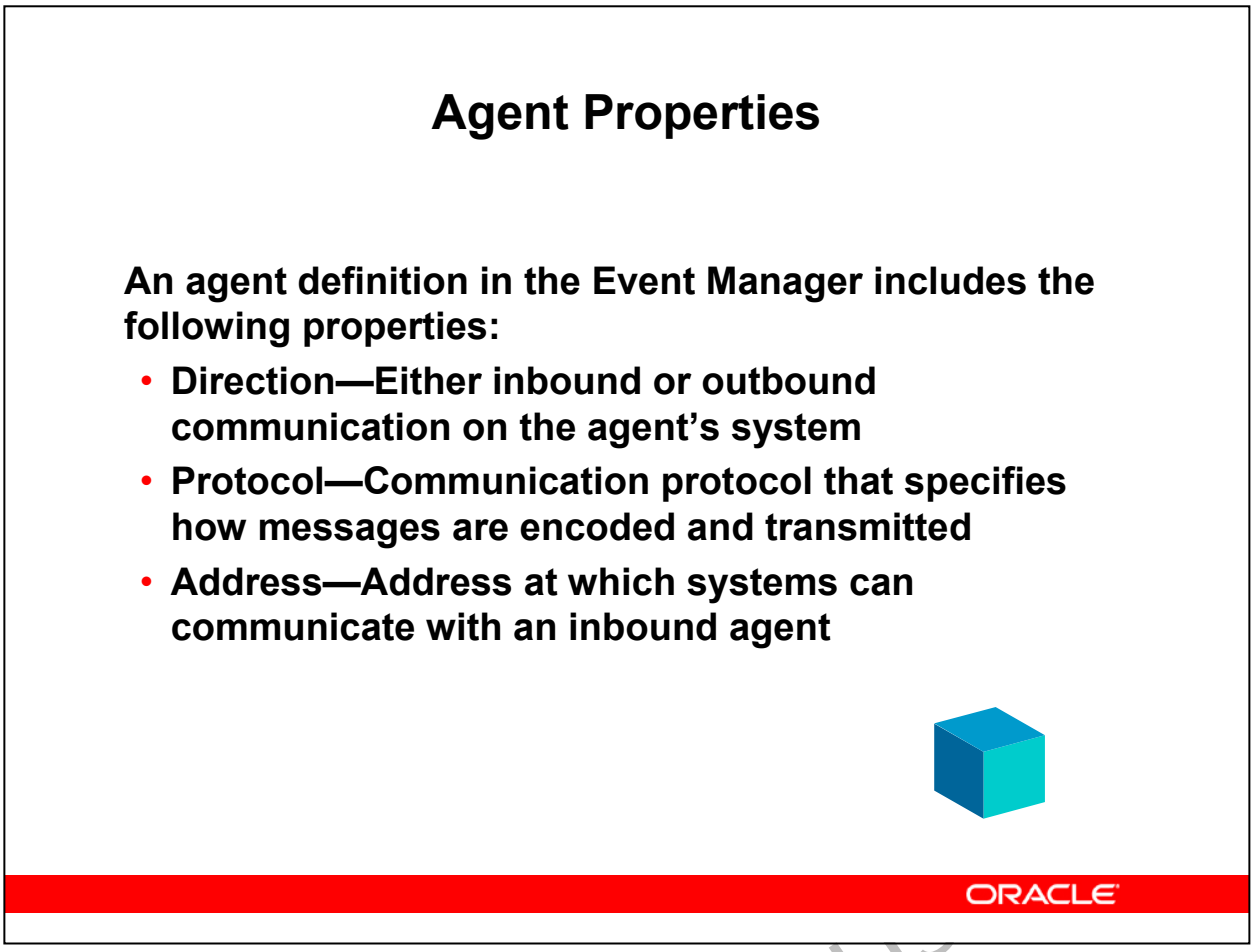
    Note: You must enter the queue name in all uppercase.
11. In the Direction field, select In for an agent that supports inbound communication to its system, or select Out for an agent that supports outbound communication from its system.
12. In the Status field, select Enabled or Disabled as the agent status. If you disable an agent, it still remains in the Agents list for reference, but you cannot use the agent in active subscriptions.
13. Choose the Submit button to save the agent.

Agent Properties

An agent definition in the Event Manager includes the following properties:

- **Internal Name—Must be unique within the agent's system**
- **Display Name**
- **System**

ORACLE



Agent Properties

Direction

An agent can support only one direction of communication on its system:

- In—The agent receives messages in a specific protocol and presents them to the system in a standard format.
- Out—The agent accepts messages from the system in a standard format and sends them using the specified protocol.

Protocol

For a message to be successfully communicated, the sending and receiving agents must use the same protocol.

You can use Oracle Advanced Queuing (AQ) to perform the propagation of messages by the SQLNET protocol which it supports, or you can use AQ's Internet access functionality to perform AQ operations over the Internet using transport protocols such as HTTP and HTTPS. You can also use the Messaging Gateway and Internet access features of AQ for integration with third party messaging solutions, or implement an external service to propagate messages by a different protocol.

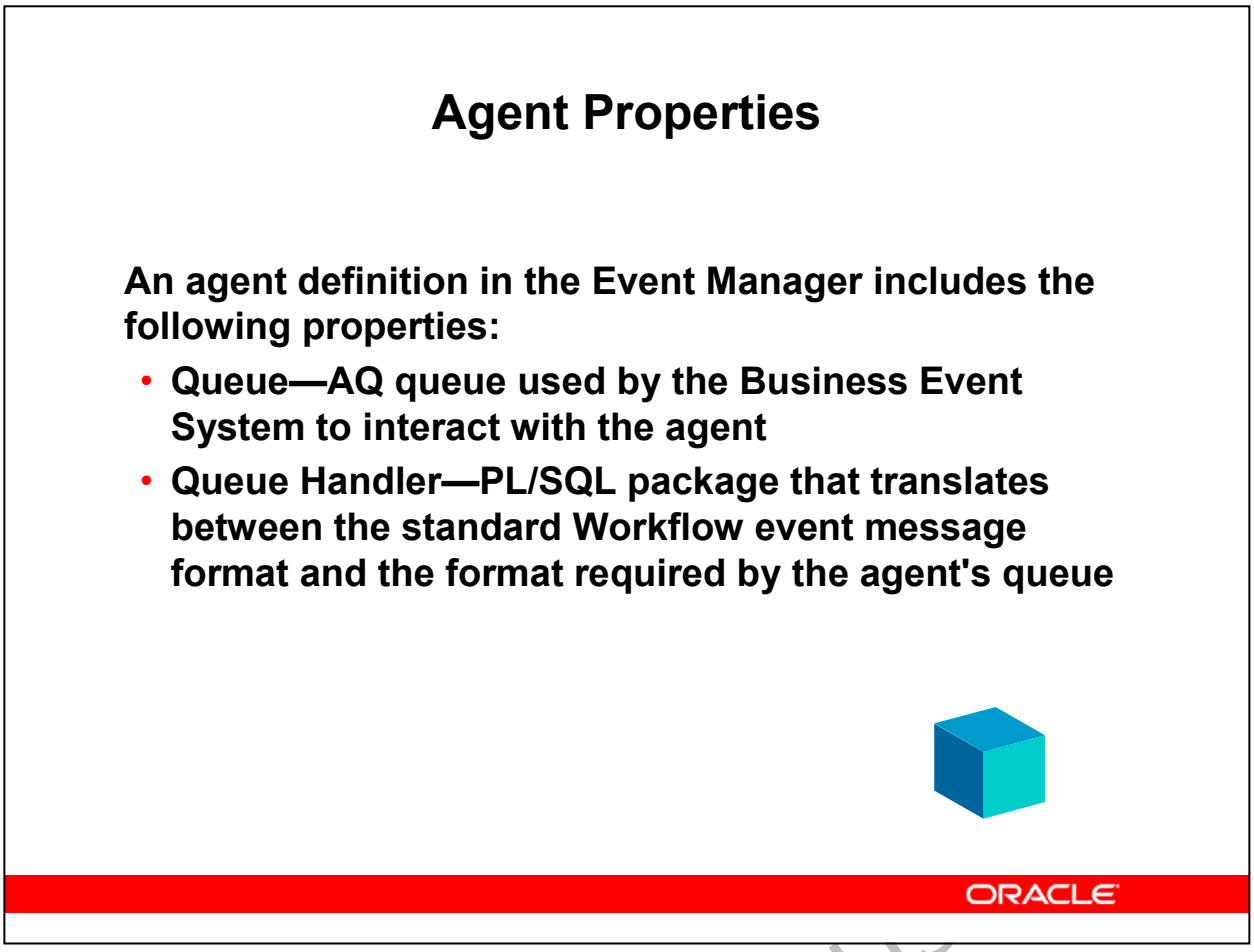
Address

The address format for an inbound agent depends on the agent's protocol. For agents that use the SQLNET protocol, the address must be in the following format:

<schema>.<queue>@<database link>

In this format, *<schema>* represents the schema that owns the queue, *<queue>* represents the name of the queue associated with the agent, and *<database link>* represents the name of the database link to the instance where the queue is located.

Note: You must enter the database link name exactly as the name was specified when the database link was created. The names of the database links that you want to use for the Business Event System should be fully qualified with the domain names.



Agent Properties

Queue

Each agent on a Workflow-enabled system should be associated with an AQ queue.

- Outbound agent—To send messages, the system enqueues the messages on the queue and sets the recipient addresses.
- Inbound agent— To receive messages, the system runs a queue listener on the queue.

Specify the queue using the following format:

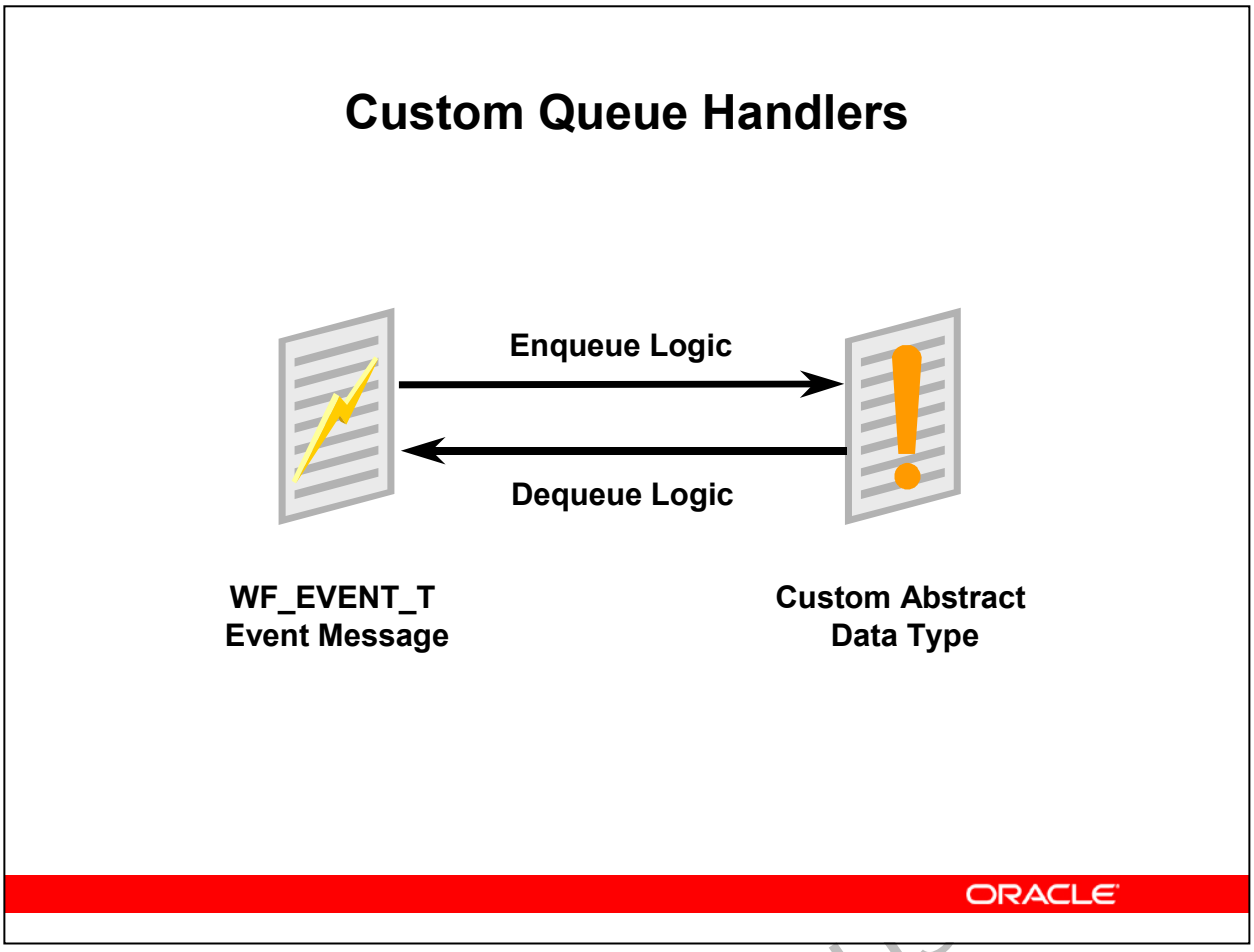
<*schema*>.<*queue*>

<*schema*> represents the schema that owns the queue and <*queue*> represents the queue name.

Queue Handler

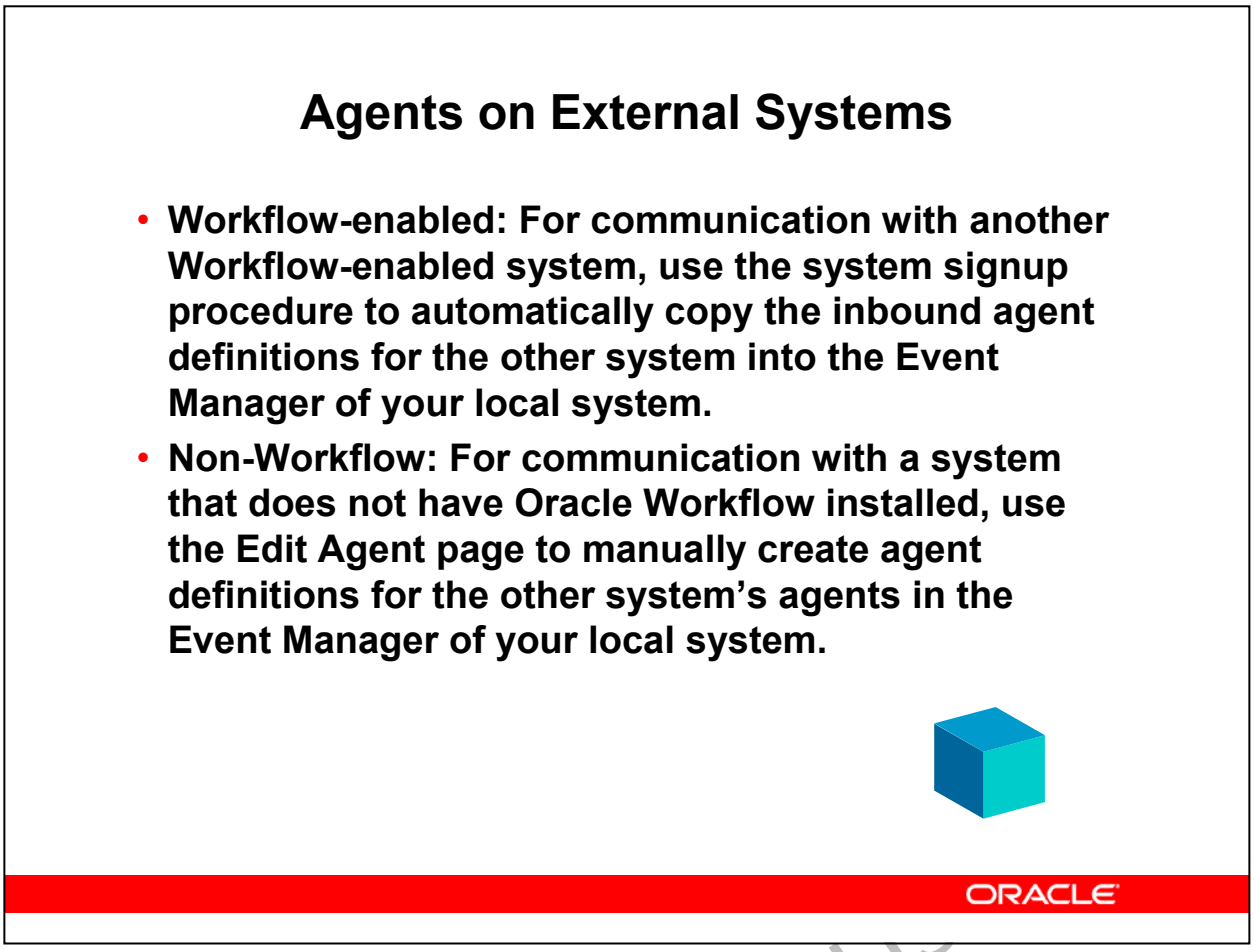
Event messages within the Business Event System are encoded in a standard format defined by the datatype WF_EVENT_T. You must assign each agent a queue handler to enqueue and dequeue messages on the agent's queue, translating between the standard Workflow format and the format required by the queue.

Oracle Workflow provides a standard queue handler named WF_EVENT_QH which you can use for normal Business Event System processing with queues using SQLNET propagation and the payload type WF_EVENT_T. You can also create your own custom queue handler.



Custom Queue Handlers

A custom queue handler must translate between the standard WF_EVENT_T event message structure and your custom Abstract Data Type (ADT). Queue handler packages must include an enqueue API and a dequeue API, which must both follow a standard API format.



Agents on External Systems

Systems that will communicate events with each other must store each other's inbound agent definitions in order to address event messages to each other.

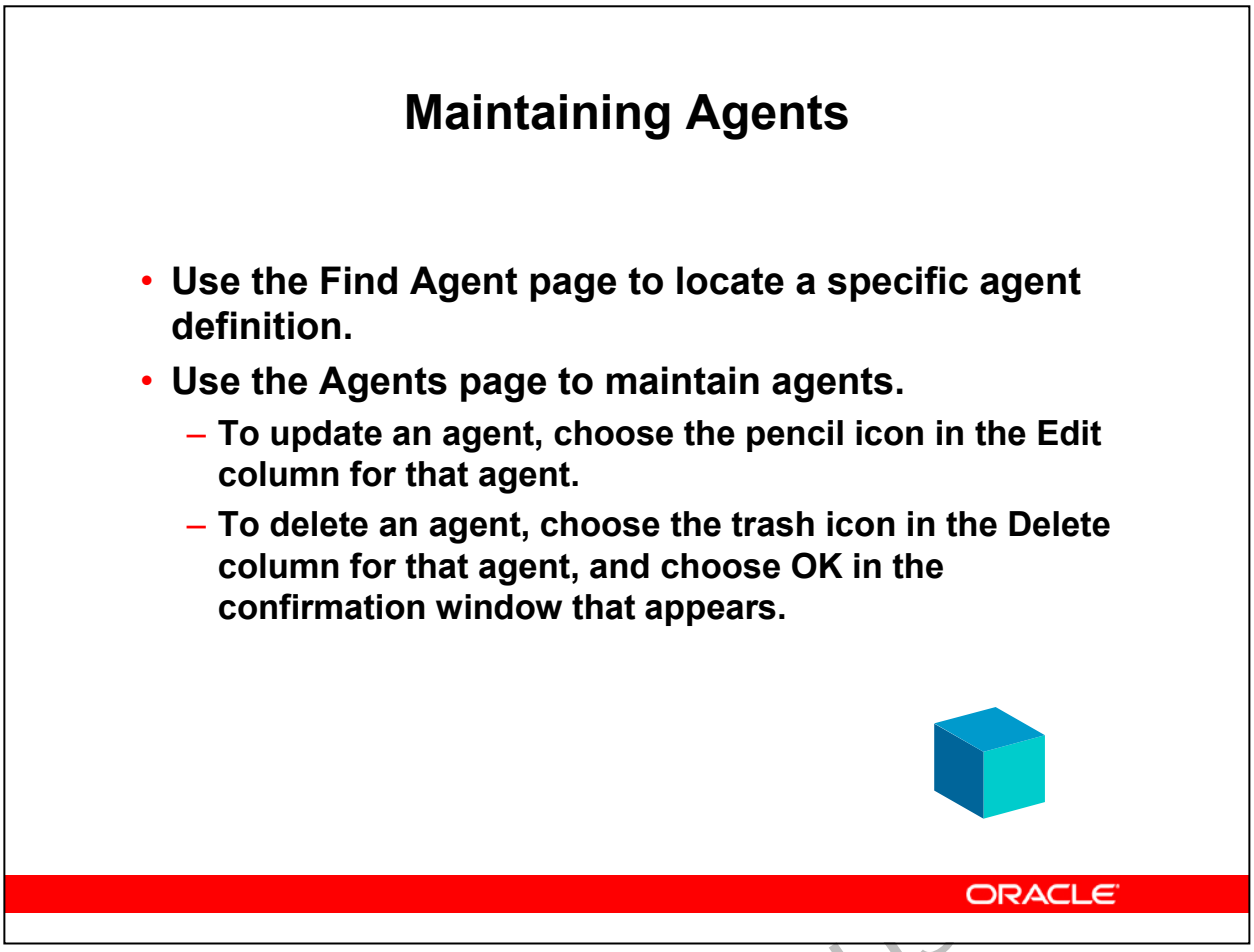
If your local Workflow-enabled system will communicate with a non-Workflow system, the non-Workflow system must provide its own external propagation agents to handle Business Event System event messages.

- An inbound agent on a non-Workflow system must be able to dequeue event messages from a Business Event System outbound queue and process the contents of those messages. The inbound agent must have an address at which systems can communicate with the agent.
- An outbound agent on a non-Workflow system must be able to enqueue event messages in the appropriate format on a Business Event System inbound queue.

You must manually define the inbound agents for the external system in the Event Manager of your local system. You can optionally define the external system's outbound agents as well.

- Before defining agents for a non-Workflow system, define the system itself using the Edit System page.

- Use the Edit Agent page to define an agent for a non-Workflow system, following the same steps as for any other agent.
  - You must associate the agent with the non-Workflow system to which it belongs and specify the protocol by which you will communicate with the agent.
  - For an inbound agent, you must also specify the address at which you will communicate with the agent.
  - You can leave the queue name and queue handler blank if the agent is not implemented as an Oracle Advanced Queuing queue.



Maintaining Agents

Finding Agents

To access the Find Agent web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.findagent

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find Agent web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Agents option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

In the Find Agent page, you can search for agents by entering the following criteria:

- Name
- Protocol

- Address
- System
- Direction
- Status

Modifying Agents

To access the Agents web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.listagents

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Agents web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Agents option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

Deleting Agents

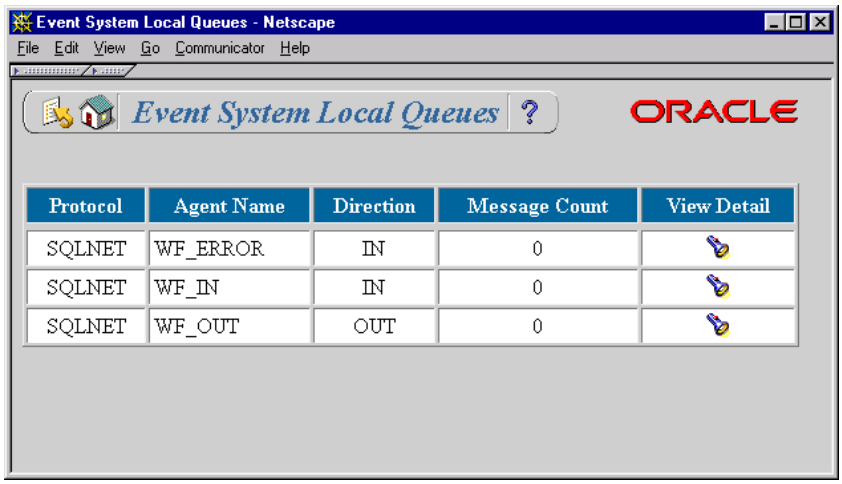
You can only delete agents that do not have any subscriptions referencing them.

Reviewing Event Messages on Local Agents

- **Use the Event System Local Queues page to review the local agents and how many messages are on their queues.**
- **For queues that use the standard WF_EVENT_T datatype as their payload type, you can also review the message details.**

ORACLE

Reviewing Event Messages on Local Agents



ORACLE

To Review Event Messages on Local Agents:

1. Use a web browser to connect to the Event System Local Queues web page. For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_html.eventqueuedisplay

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the Event System Local Queues web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Event Queue Summary option from a Workflow administrator responsibility with Event Manager functionality.

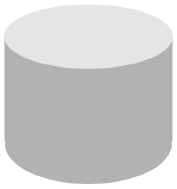
   Note: You must have workflow administrator privileges to access the Event Manager web pages.

2. The Event System Local Queues page appears, displaying a list of the local agents used by the Business Event System.

3. You can review message details for queues that use the standard WF_EVENT_T datatype as their payload type. To review message details, choose the flashlight icon in the View Detail column for a queue.
4. The Find Standard Event Queue Messages page appears. Enter search criteria to locate specific event messages. The search criteria are:
   - Event Name
   - Event Key
   - Status
5. Choose the Go button. The Local Queue Messages page appears, displaying a list of event messages on the queue you selected that match your search criteria.
6. To review the XML document that contains the event data for a message, choose the flashlight icon in the XML Document column for that message.

Signing Up Systems

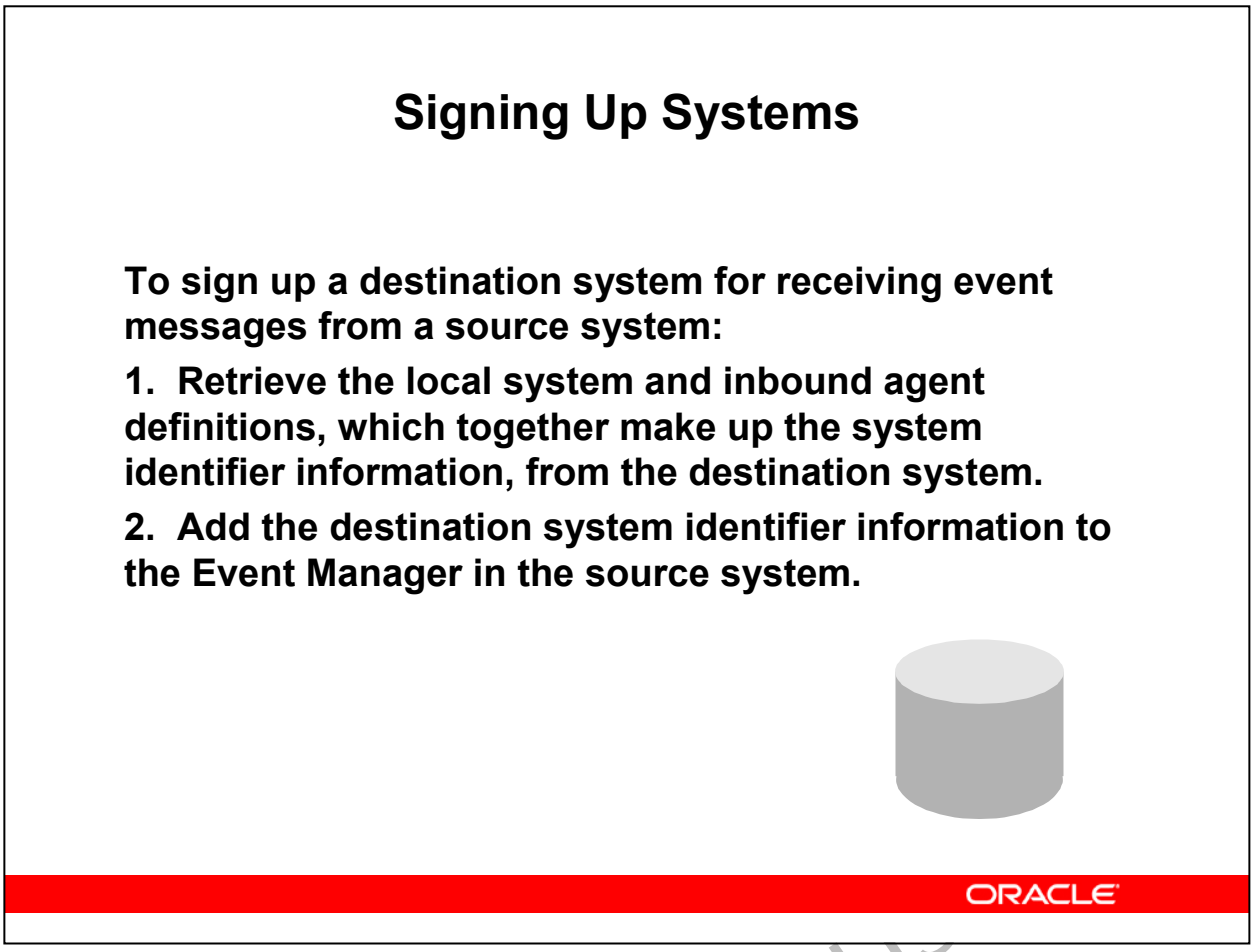
- Before sending events from one system to another, you must sign up the destination system with the source system as a recipient of event messages.
- Signing up a system means defining the destination system and its inbound agents in the Event Manager of the source system.
- When the destination system is signed up, you can address event messages from the source system to the destination agents.



ORACLE

Signing Up Systems

Usually when you integrate two systems, both systems should be signed up with each other, so that each system can both send messages to and receive messages from the other system.



Signing Up Systems

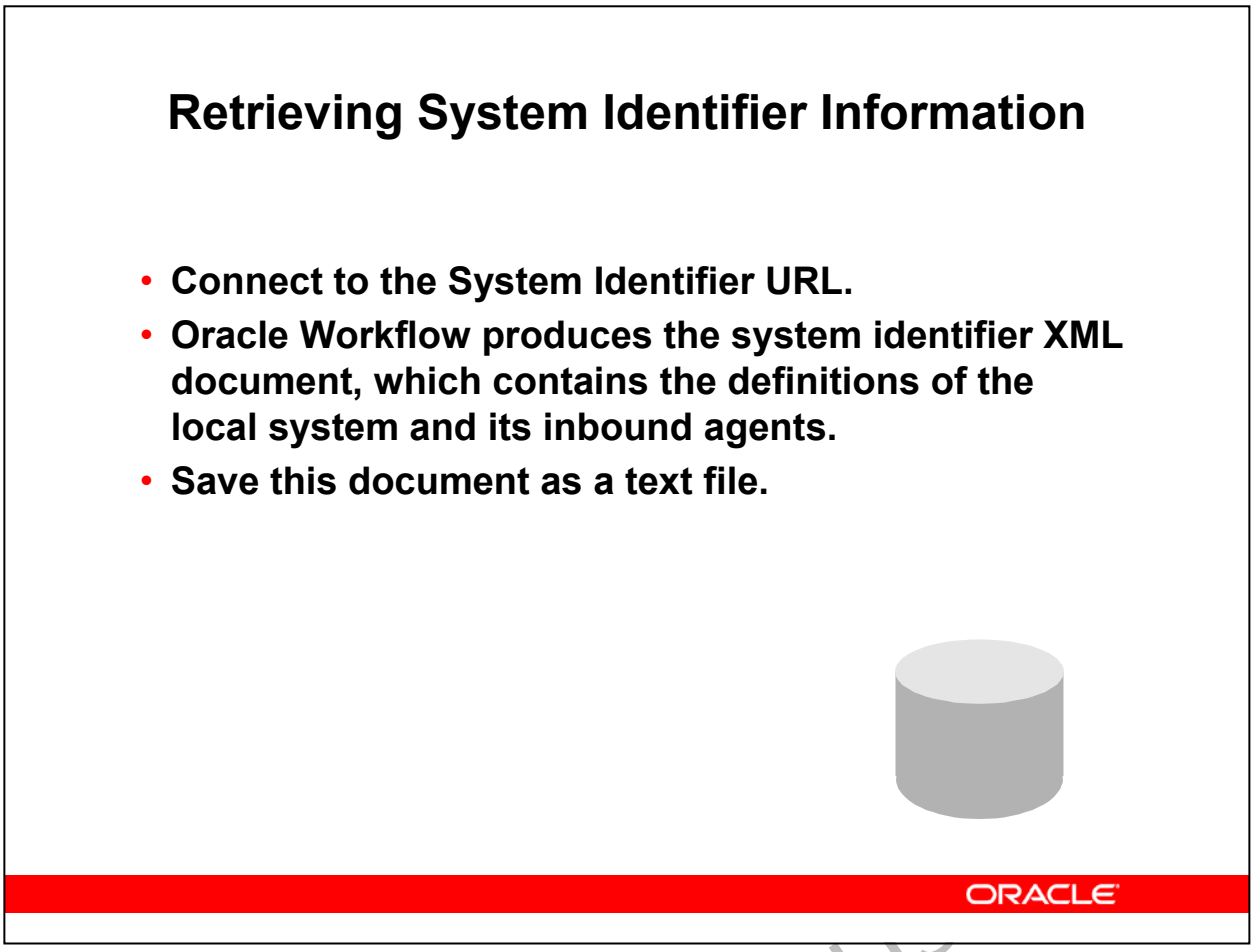
Oracle Workflow provides a procedure that automates the steps to copy the system and inbound agent definitions for one Workflow-enabled system into the Event Manager of another Workflow-enabled system.

- You can use the System Identifier web page on the destination system to generate an XML document containing the system identifier information.
- You can use the System Signup web page on the source system to add the information by raising the System Signup event with the XML document from the destination system as the event data. When the System Signup event is raised on the source system, Oracle Workflow executes a predefined subscription that adds the system identifier information to the Event Manger in that system.

For communication between a Workflow-enabled system and a non-Workflow system, you must perform manual steps to store the required destination system and agent information in the source system.

- To send events from a Workflow-enabled source system to a non-Workflow destination system, you must manually define the non-Workflow system and its agents in the Event Manager of the source system using the Edit System and Edit Agent pages.

- To send events from a non-Workflow source system to a Workflow-enabled destination system, you must store the system and inbound agent information for the Workflow-enabled system in the non-Workflow system according to that external system's requirements. You can optionally use the System Identifier web page on the destination system to generate an XML document containing its local system and inbound agent definitions, but you must manually perform whatever steps are necessary to store that information in the non-Workflow source system.



Retrieving System Identifier Information

Use a web browser to connect to the System Identifier URL. For standalone Oracle Workflow, connect to the following URL:

<*webagent*>/wf_event_html.getsystemidentifier

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

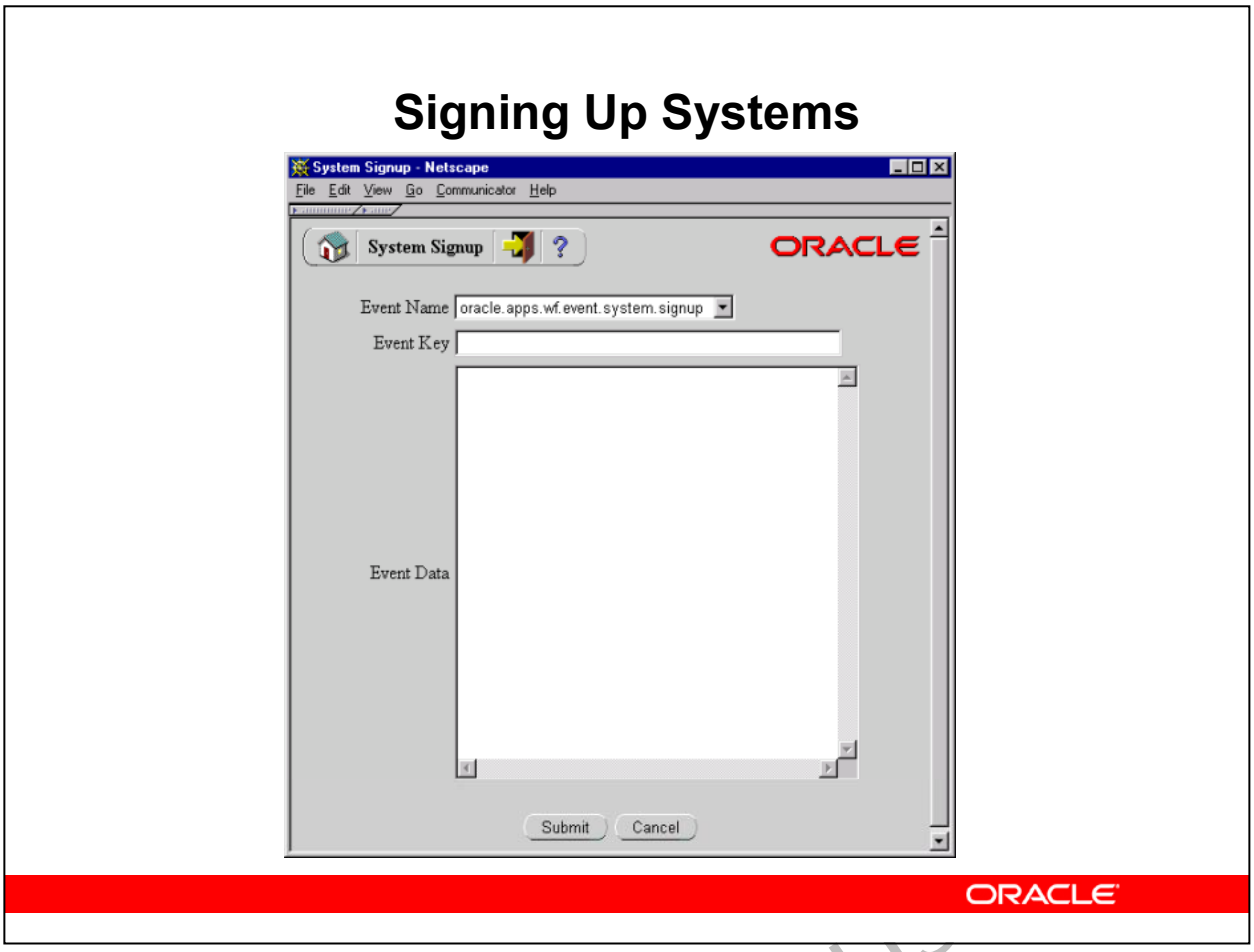
You can also access the System Identifier URL from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Get System Identifier option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

System Identifier XML Document

After you save the system identifier XML document as a text file, you can copy the document and enter it as the event data for the System Signup event when you sign this system up with a source system.



To Sign Up a System:

Note: You can only use this procedure to sign up one Workflow-enabled system with another.

1. Use a web browser to connect to the System Signup web page . For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_html.entereventdetails?p_event_name= oracle.apps.wf.event.system.signup

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the System Signup web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the System Signup option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.

2. In the System Signup page, enter an event key that uniquely identifies this instance of the event.

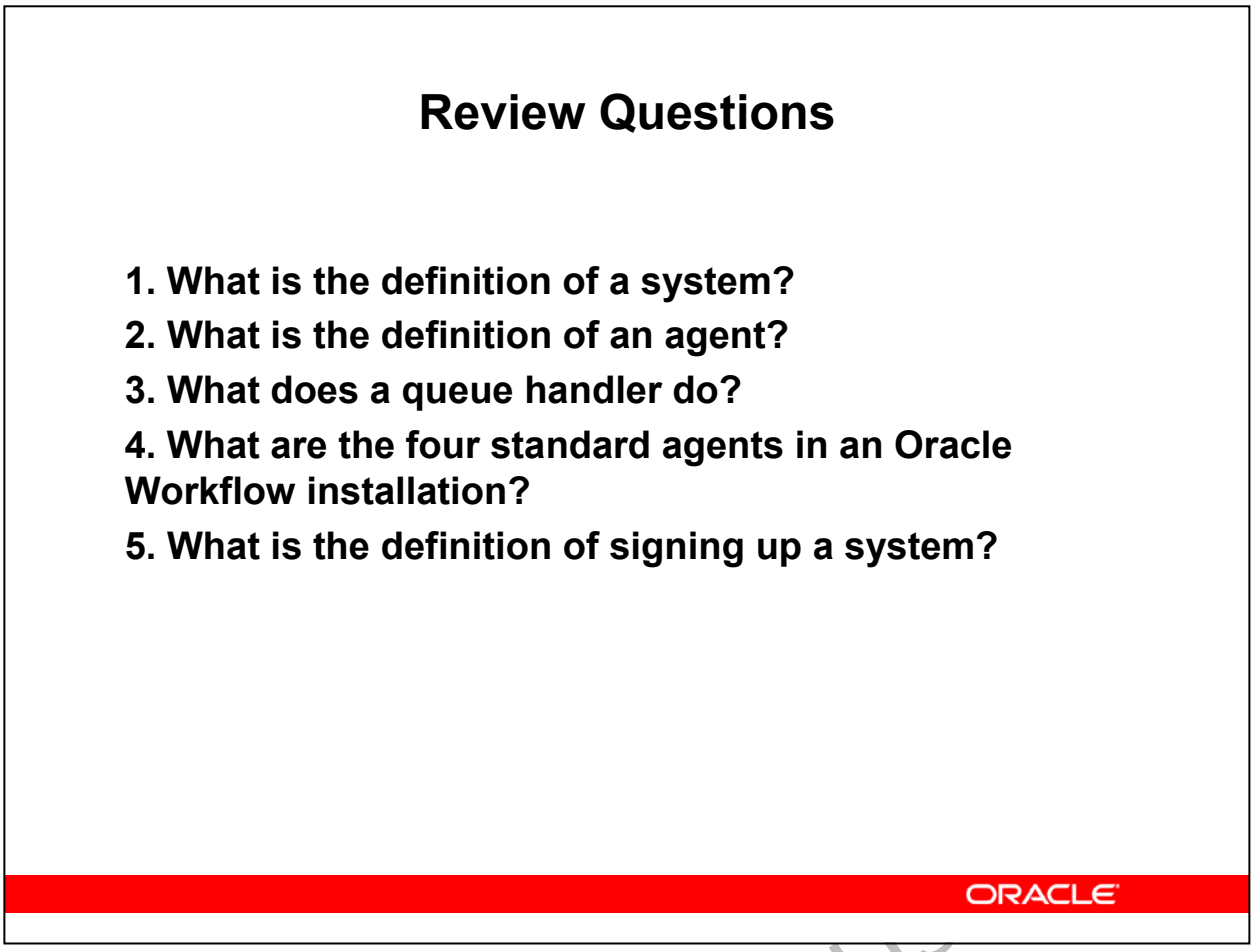
3. Copy the XML document containing the destination system identifier information into the Event Data field.
4. Choose the Submit button to raise the System Signup event to the Event Manager. A confirmation message is displayed.

   When the System Signup event is raised, Oracle Workflow executes a predefined subscription that adds the system identifier information from the event data to the Event Manager.

Review Questions

1. What is the definition of a system?
2. What is the definition of an agent?
3. What does a queue handler do?
4. What are the four standard agents in an Oracle Workflow installation?
5. What is the definition of signing up a system?

ORACLE



Review Questions and Solutions

1. What is the definition of a system?

   A system is a logically isolated software environment such as a host machine or database instance.

2. What is the definition of an agent?

   An agent is a named point of communication within a system.

3. What does a queue handler do?

   A queue handler enqueues and dequeues messages on an agent's queue, translating between the standard Workflow format and the format required by the queue.

4. What are the four standard agents in an Oracle Workflow installation?

   The four standard agents automatically defined on the local system when you install Oracle Workflow are:

   - **WF_IN—Standard inbound agent**
   - **WF_OUT—Standard outbound agent**
   - **WF_DEFERRED—Standard agent for deferred subscription processing**
   - **WF_ERROR—Standard agent for error handling**

5. What is the definition of signing up a system?

Signing up a system means defining the destination system and its inbound agents in the Event Manager of the source system.

Summary

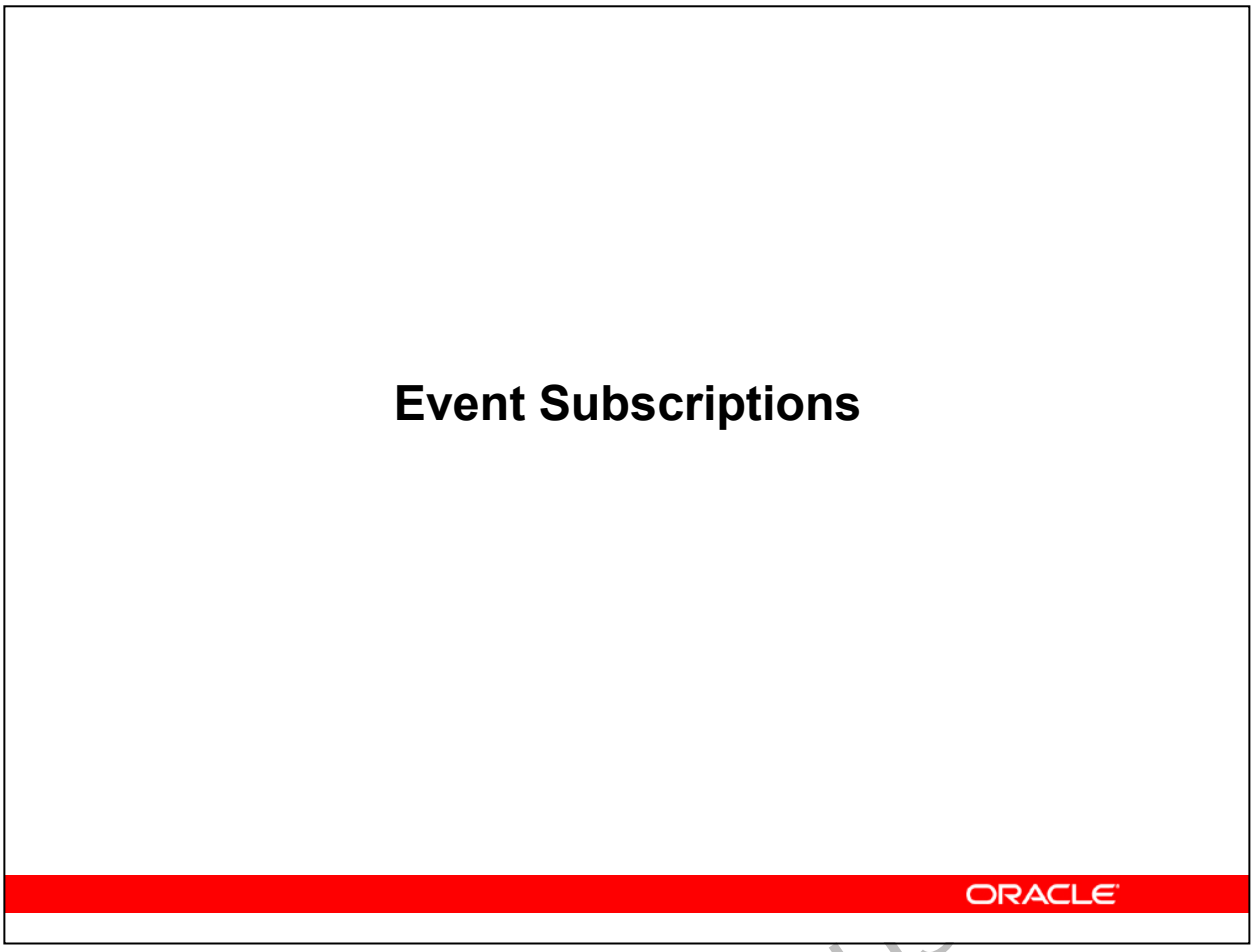
In this lesson, you should have learned how to:

- **Define systems.**
- **Define communication agents on systems.**
- **Associate queues and queue handlers with agents.**
- **Review event messages on local agents.**
- **Sign up systems with each other for event message propagation.**

ORACLE

Event Subscriptions

Chapter 5



Objectives

After completing this lesson, you should be able to do the following:

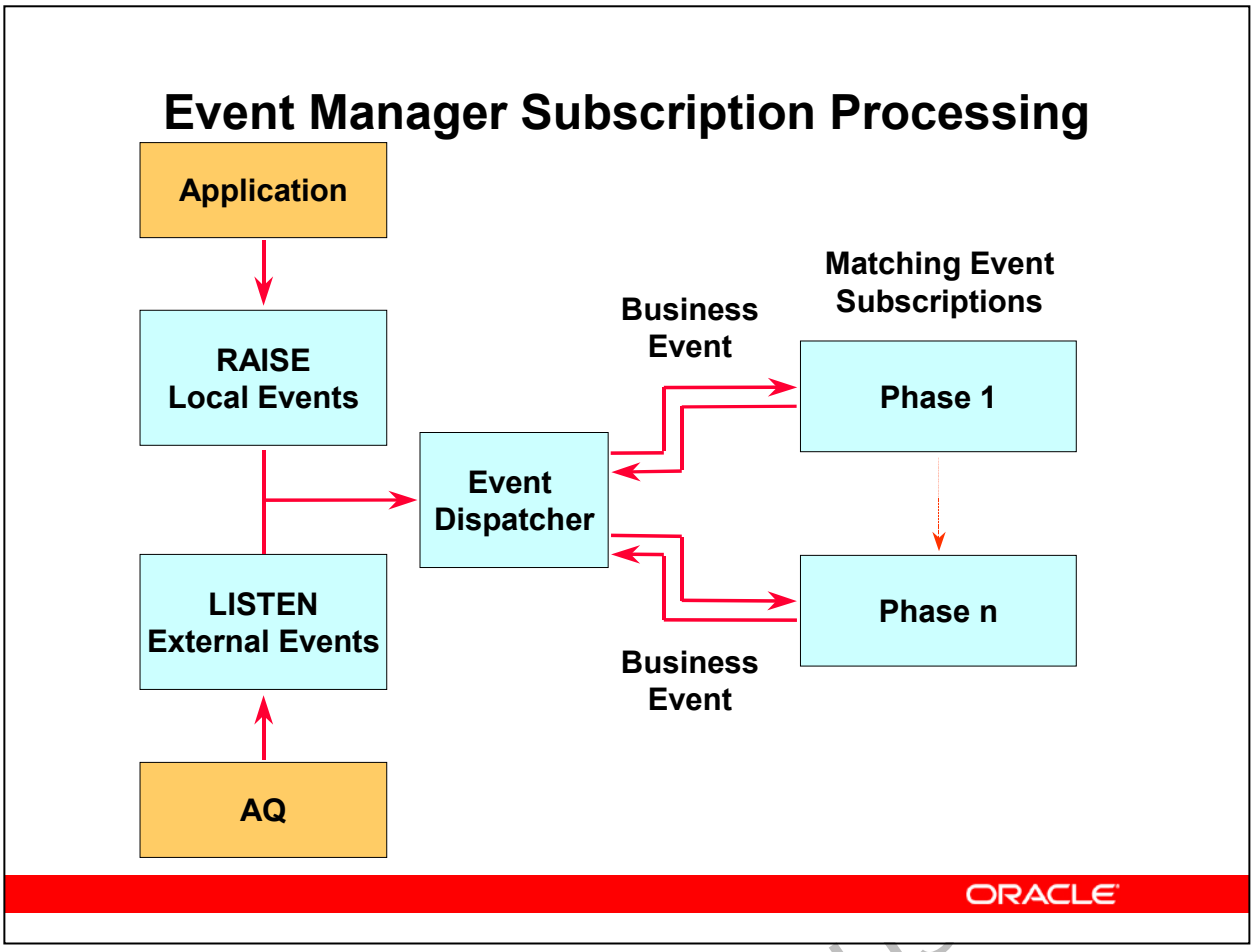
- **Describe the types of processing a subscription can include.**
- **Define subscriptions.**
- **Use predefined Oracle Workflow subscriptions.**

ORACLE

Event Subscriptions

- An event subscription is a registration indicating that a particular event is significant to a system.
- Define subscriptions in the Event Manager to specify the processing that you want to perform when events occur.
- Whenever an event is raised locally or received from an external source, the Event Manager searches for and executes any active subscriptions by the local system to that event or to the Any event.

ORACLE

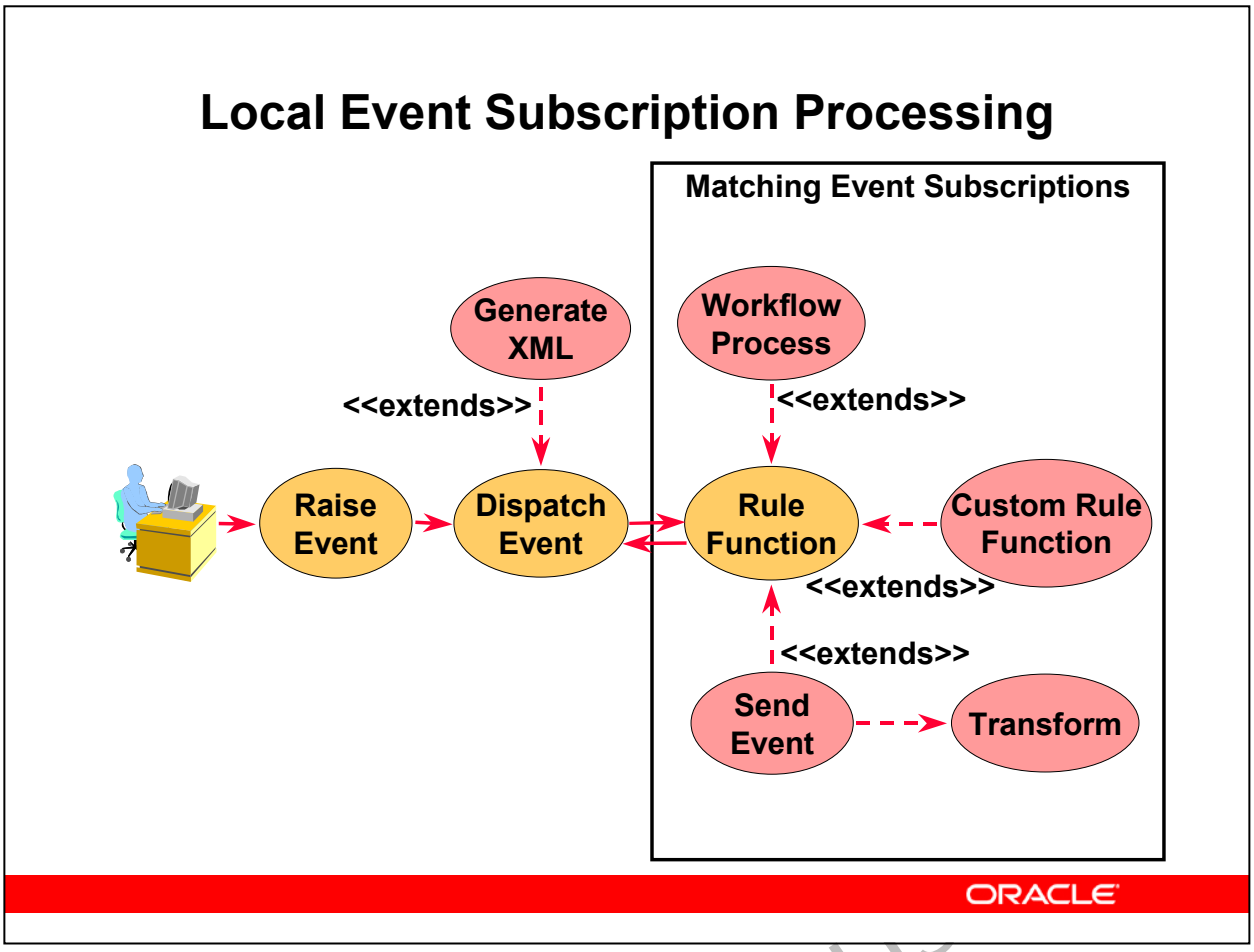


Event Manager Subscription Processing

When an event is raised by a local application, or when an event is received from an external source through Oracle Advanced Queuing (AQ), it is processed by a component of the Event Manager called the Event Dispatcher.

In both cases, the Event Dispatcher searches for and executes any subscriptions to that event. If there are multiple subscription to the same event, a subscription property called the phase value determines the order in which the subscriptions are executed.

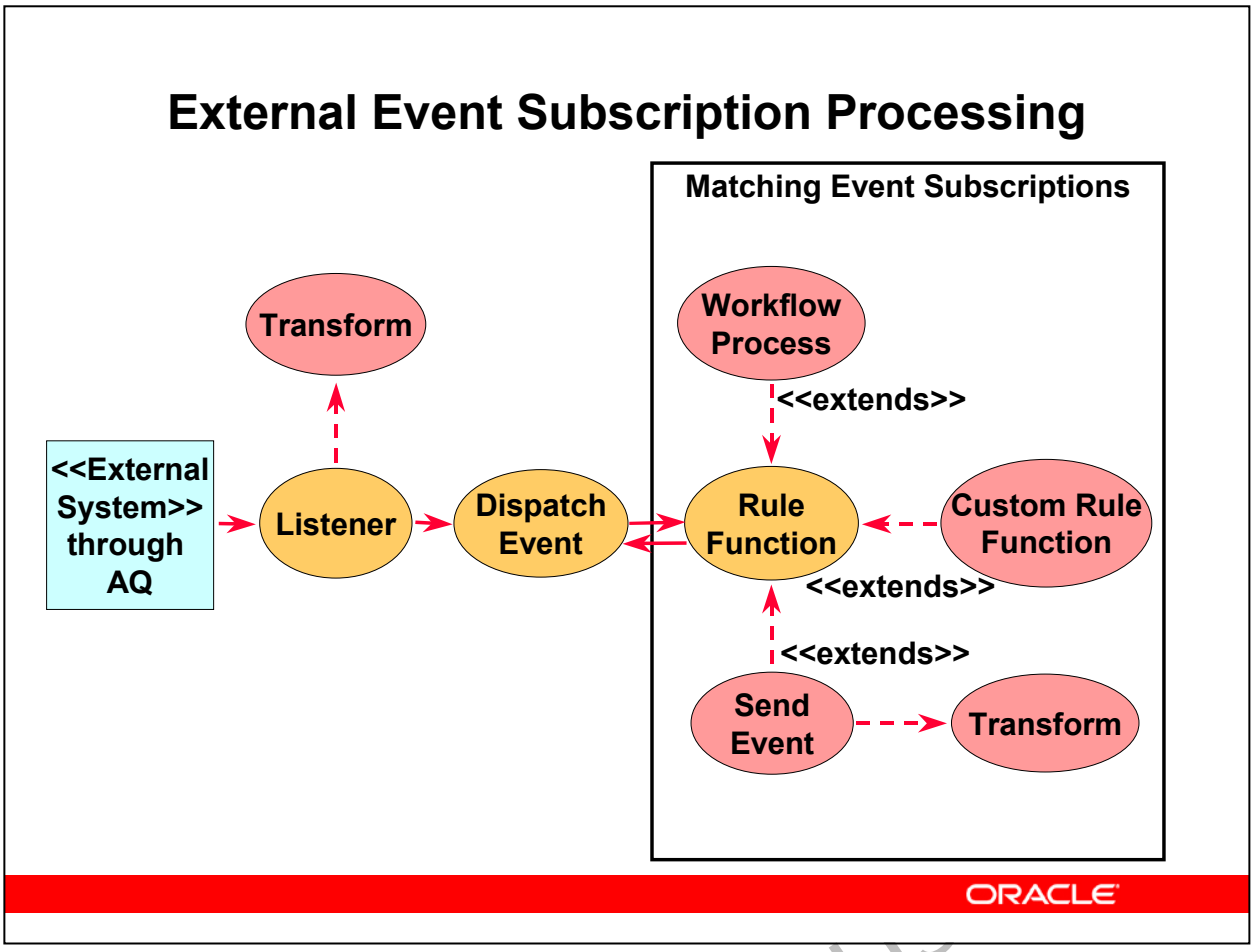
If no active subscriptions exist for the event that occurred, then you can handle the event by having Oracle Workflow execute any active subscriptions to the Unexpected event.



Local Event Subscription Processing

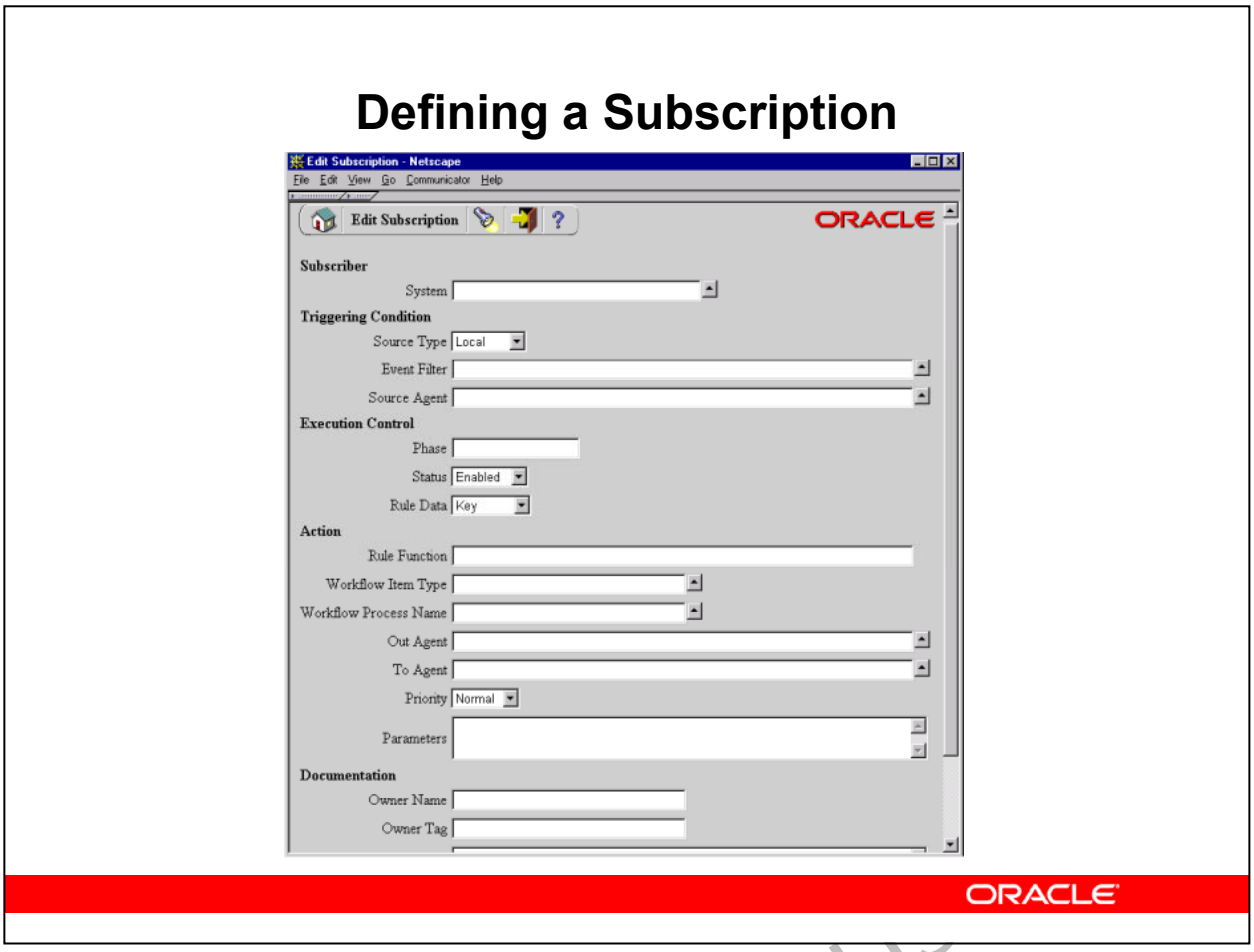
Oracle Workflow provides powerful capabilities for controlling the processing performed by your subscriptions when an event is raised.

- The Business Event System can handle event information as XML documents.
- Each event subscription has a rule function that specifies the processing it performs. This processing can include:
  - Running custom code on the event message
  - Sending the event message to a workflow process
  - Sending the event message to an agent on the local or an external system
- The Business Event System supports performing transformations on event messages at propagation time.



External Event Subscription Processing Through AQ

Event subscriptions can also be triggered when an event message is received from an external source through Oracle Advanced Queuing (AQ). The Business Event System supports performing transformations on event messages when they are received. Also, external subscriptions can perform the same types of processing as local subscriptions, including running a custom rule function, sending the event message to a workflow process, and sending the event message to an agent.



To Define a Subscription:

1. Use a web browser to connect to the Event Subscriptions web page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/wf_event_ html.listsubscriptions

   Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

   You can also access the Event Subscriptions web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

   For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Event Subscriptions option from a Workflow administrator responsibility with Event Manager functionality.

   Note: You must have workflow administrator privileges to access the Event Manager web pages.

2. The Event Subscriptions page appears, displaying a list of existing subscriptions grouped by the subscribing system and triggering event.

   Choose the Add Subscription button to open the Edit Subscription page.

3. In the Subscriber region, enter the system where the subscription executes. Click on the System field's up arrow icon to display a list of systems from which to choose.
4. In the Triggering Condition region, specify the type of source system to which the subscription applies in the Source Type field.
   - Local—The subscription applies only to events raised on the subscribing system.
   - External—The subscription applies only to events received by an inbound agent on the subscribing system.

     Note: All event messages received by an inbound agent on the subscribing system are considered to have an External source, whether the sending agent is located on a remote system or on the local system.
   - Error—The subscription applies to only to errored events dequeued from the WF_ERROR agent's queue.

     Note: Each subscription defines an action for exactly one source type. If you want to handle an event from both local and external sources, you should define two separate subscriptions, one with a source type of Local and the other with a source type of External.
5. Enter the event to which the subscription applies in the Event Filter field. Click on the Event Filter field's up arrow icon to display a list of events from which to choose.
6. Enter an optional source agent to which the subscription applies. If you specify a source agent, then the subscription is executed only when the triggering event is received from that agent. Click on the Source Agent field's up arrow icon to display a list of agents from which to choose.

   Note: In most cases, the Source Agent field is left blank.
7. In the Execution Control region, enter an optional phase number for the subscription to specify the order in which subscriptions that apply to the same event are executed.
8. Select Enabled or Disabled as the subscription status. If you disable a subscription, it still remains in the Event Subscriptions list for reference, but it can no longer be actively used to respond to events.
9. In the Rule Data field, specify the event information required by the subscription.
   - Key—The subscription requires only the event key.
   - Message—The subscription requires the complete event data.
10. In the Action region, define the subscription processing you want to perform when the triggering event occurs.
11. If you want to run a function on the event message, enter the Rule Function to run.

    If you do not specify a rule function, Oracle Workflow runs a default rule function to send the event message to the workflow process and the agent that you specify.
12. If you want to send the event message to a workflow process, enter the item type that the process belongs to in the Workflow Item Type field and the name of the process in the Workflow Process Name field. Click on each field's up arrow icon to display a list of values from which to choose.

    Note: The list of values for the Workflow Process Name field includes only the runnable processes within the item type you specify.

13. If you want to send the event to an agent, optionally enter the Out Agent that you want to send the outbound message. Click on the Out Agent field's up arrow icon to display a list of values from which to choose.

    Note: The Out Agent must be located on the subscribing system. The list of values for the Out Agent field includes only agents with a direction of Out.

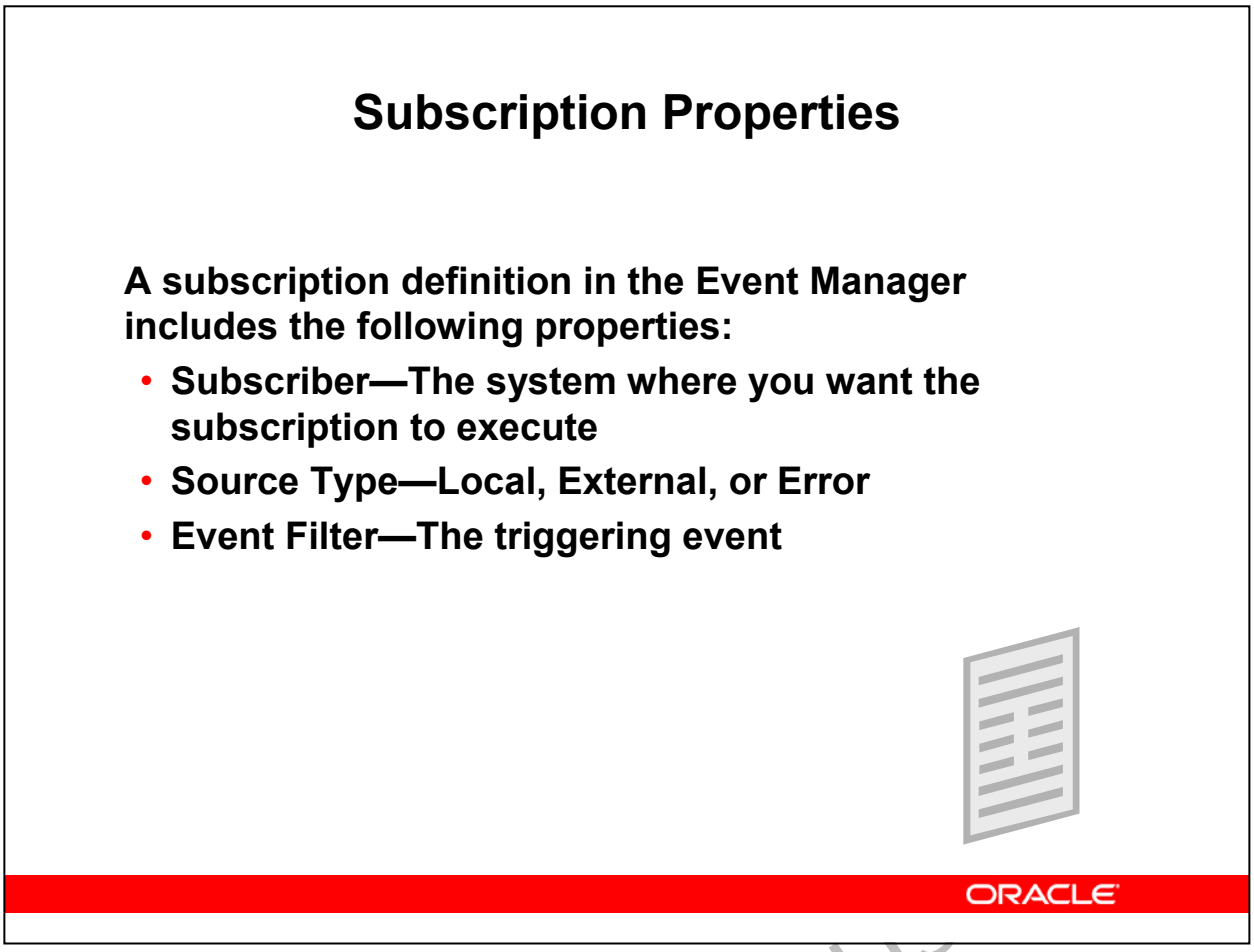
14. If you want to send the event to an agent, enter the To Agent that you want to receive the inbound message. Click on the To Agent field's up arrow icon to display a list of values from which to choose.

    Note: The list of values for the To Agent field includes only agents with a direction of In.

15. If you want to send the event message to an agent, select Normal, High, or Low as the priority with which the recipient should dequeue the message.
16. Optionally enter any additional parameters for the rule function or workflow process in the Parameters field. Use spaces to separate the parameters, and specify the name and value for each parameter in the following format:

    <name1>=<value1> <name2>=<value2> ... <nameN>=<valueN>

17. If you use a program to create subscription definitions automatically, the Event Manager displays owner information set by that program in the Owner Name and Owner Tag fields in the Documentation region. You can use the Edit Subscription page to update this information manually if necessary.
18. Enter an optional description for the subscription.
19. Choose the Submit button to save the subscription.



Subscription Properties

Subscriber

Each subscription defines an action on exactly one system, so you should define a separate subscription for each system involved in the processing you want to perform. For example, if you want to propagate data from one system to another, you should define one subscription for the sending system, and another subscription for the receiving system.

Note: You can define and store subscriptions for multiple systems in the Event Manager on your local system, and use predefined events and subscriptions to replicate those subscriptions to other systems. However, only subscriptions with the local system as the subscriber are triggered when events are raised or received on the local system.

Source Type

Subscriptions can have the following source types:

- Local—The subscription applies only to events raised on the subscribing system.
- External—The subscription applies only to events received by an inbound agent on the subscribing system.

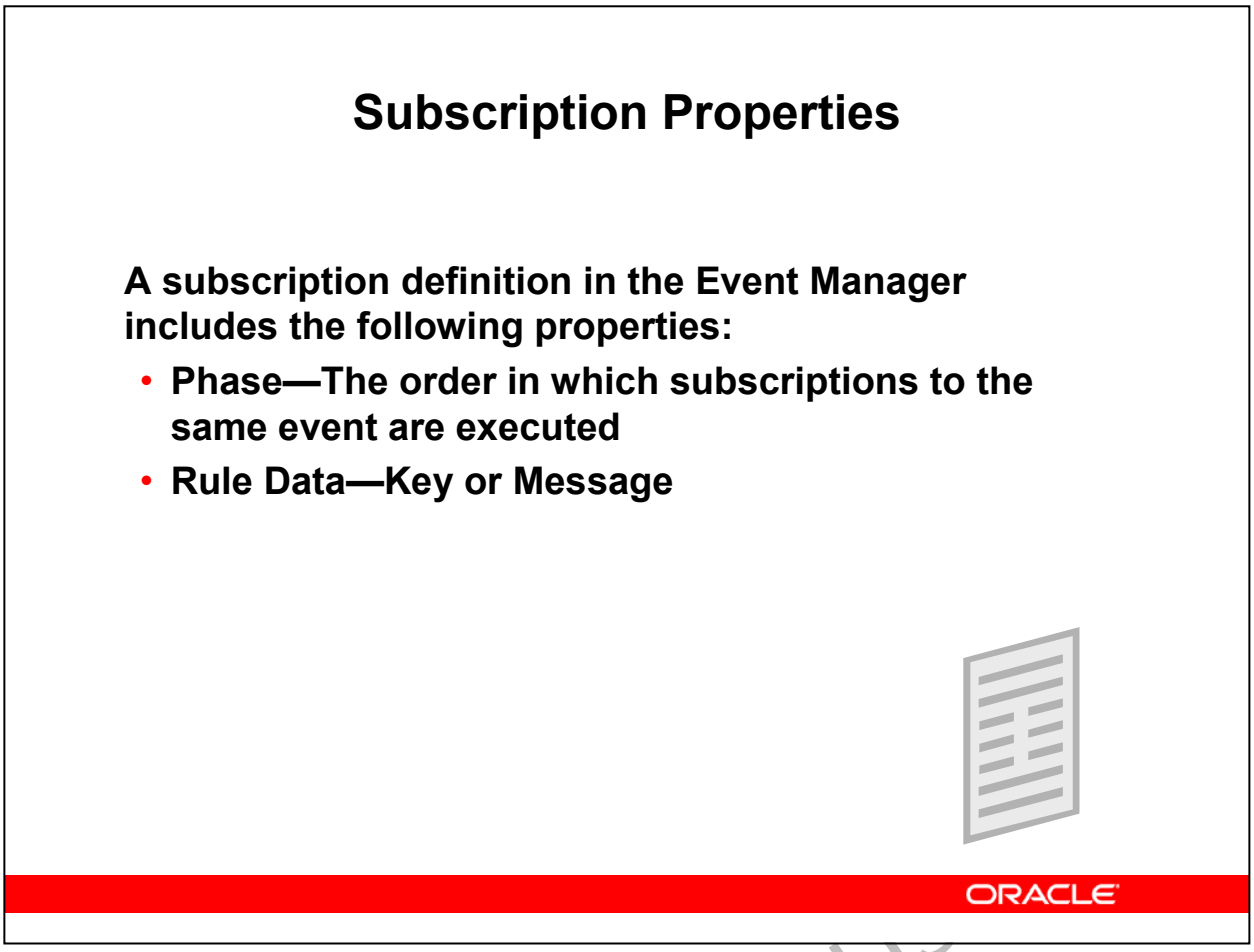
Note: All event messages received by an inbound agent on the subscribing system are considered to have an External source, whether the sending agent is located on a remote system or on the local system.

- Error—The subscription applies to only to errored events dequeued from the WF_ERROR agent's queue.

Note: Each subscription defines an action for exactly one source type. If you want to handle an event from both local and external sources, you should define two separate subscriptions, one with a source type of Local and the other with a source type of External.

Event Filter

You can choose either an individual event or an event group. If you choose an event group, the subscription will be triggered whenever any one of the group's member events occurs.



Subscription Properties

Phase

If you define multiple subscriptions to the same event, you can control the order in which the Event Manager executes those subscriptions by specifying a phase number for each subscription.

Subscriptions are executed in ascending phase order. For example, you can enter 10 for the subscription that you want to execute first when an event occurs, 20 for the subscription that you want to execute second, and so on.

You can use phases to ensure that different types of actions are performed in the appropriate order, such as executing subscriptions that perform validation before subscriptions that perform other types of processing.

You can also use the phase number for a subscription to control whether the subscription is executed immediately or is deferred. The Event Manager treats subscriptions with a phase number of 100 or higher as deferred subscriptions.

Rule Data

Depending on the processing to be performed, a subscription may have the following rule data requirements:

- Key—Requires only the event key that identifies the instance of the event
- Message—Requires the complete set of event information contained in the event data

You can improve performance by specifying Key as the rule data for subscriptions that do not require the complete event data. For locally raised events, the Event Manager checks each subscription before executing it to determine whether the subscription requires the complete event data. If the event data is required but is not already provided, the Event Manager runs the Generate function for the event to produce the event data. However, if no subscriptions to the event require the event data, then the Event Manager will not run the Generate function, minimizing the resources required to execute the subscriptions.

For events received from an external source, any necessary event data must already be included in the event message by the source system.

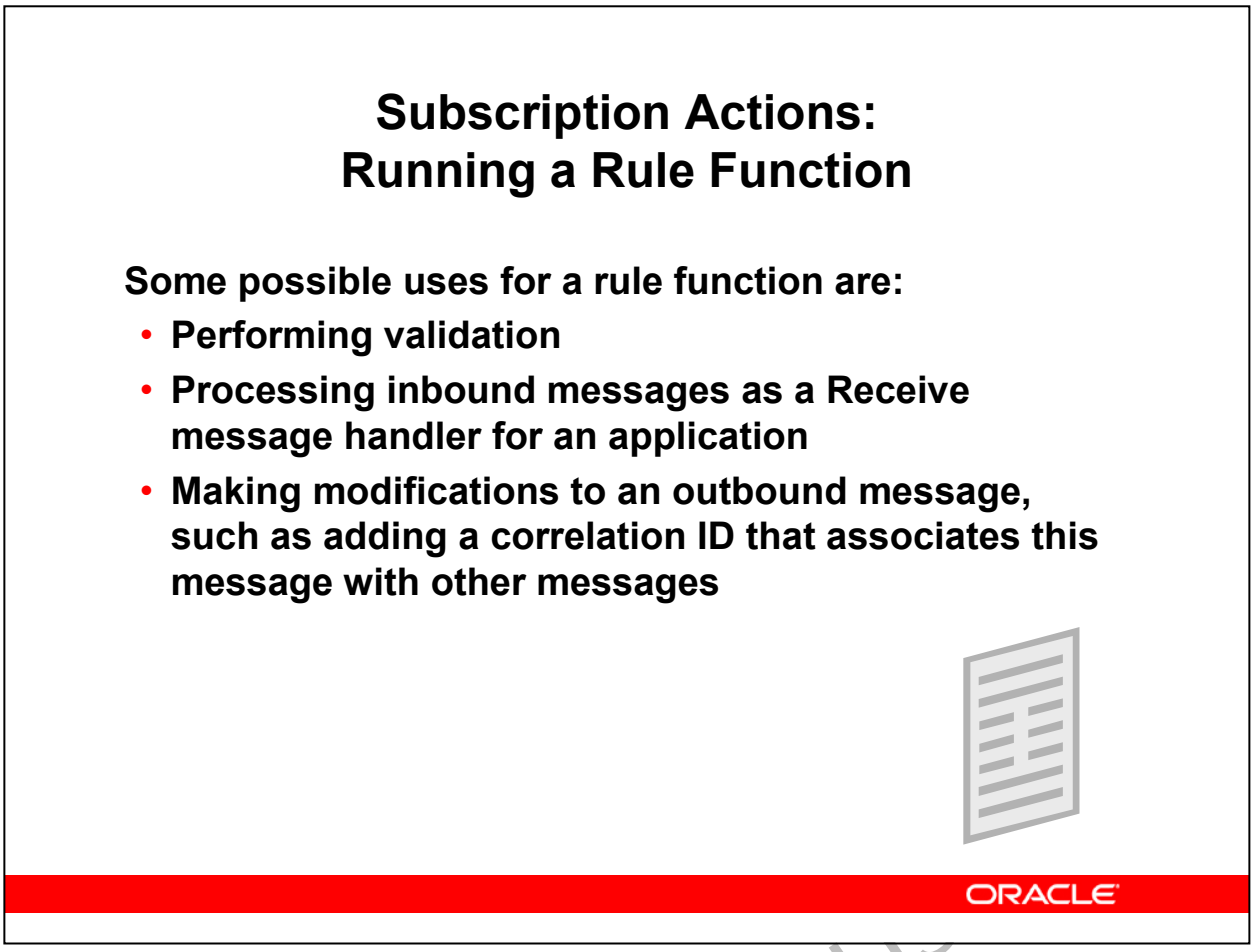
Subscription Actions: Running a Rule Function

- A subscription can include a rule function that is executed on the event message.
- Oracle Workflow provides a standard default rule function, WF_RULE.Default_Rule, that can:
  - Send the event message to a workflow process
  - Send the event message to an agent
- Oracle Workflow provides some standard rule functions for testing and debugging.
- You can extend your subscription processing by creating custom rule functions.

ORACLE

Subscription Actions: Running a Rule Function

- If you specify a custom rule function for a subscription, you can also specify in the subscription any additional parameters that you want to pass to the function.
- If you use the default rule function, you should specify in the subscription the workflow process or the agent (or both) where you want to send the event message.
- If you enter a rule function other than the default function, you can still enter workflow and agent information for your function to reference, but Oracle Workflow does not automatically send the event message to the specified workflow and agent. Instead, you must either explicitly include the send processing in your rule function, or define a separate subscription that does use the default rule function to perform the send processing.
- Custom rule functions must be defined according to a standard API.



Subscription Actions: Running a Rule Function

- The rule function controls the behavior of the processing performed for the subscription.
- If a rule function modifies the event message, any subsequent subscriptions executed on the event will access the changed message.

Subscription Actions: Sending an Event to a Workflow Process

- **To send an event to a workflow process, you must:**
  - **Specify in the subscription the item type and process name of the process.**
  - **Either use the default rule function or include send processing in your custom rule function.**
- **The item key for the process is determined either by the correlation ID specified in the event message, or by the event key if no correlation ID is specified.**

ORACLE

Subscription Actions: Sending an Event to a Workflow Process

- By sending an event to a workflow process, you can model complex processing or routing logic beyond the options of directly running a predefined function or sending the event to a predefined recipient. For example, you can branch to different functions, initiate subprocesses, send notifications, or select recipient agents, based on the contents of the event message, or modify the event message itself.
- Any parameters included in the parameter list of the event message are set as item attribute values for the workflow process. If the corresponding item attributes do not already exist in the item type, Oracle Workflow automatically creates the item attributes when the event is sent to the process.
- Additionally, the subscription's globally unique identifier (GUID) is set as a dynamic item attribute of type text named SUB_GUID so that the workflow process can reference other information in the subscription definition. You can call the WF_ENGINE.GetItemAttrText( ) API to retrieve the subscription GUID value from the SUB_GUID item attribute, and then use the GUID to select other values from the WF_EVENT_SUBSCRIPTIONS table.

- If you want to send an event to a workflow process from within a custom rule function, call:
  - WF_ENGINE.Event( ) to send the event message to a workflow process only
  - WF_RULE.Default_Rule( ) to include the default subscription processing that can send the event message both to a workflow process and to an agent

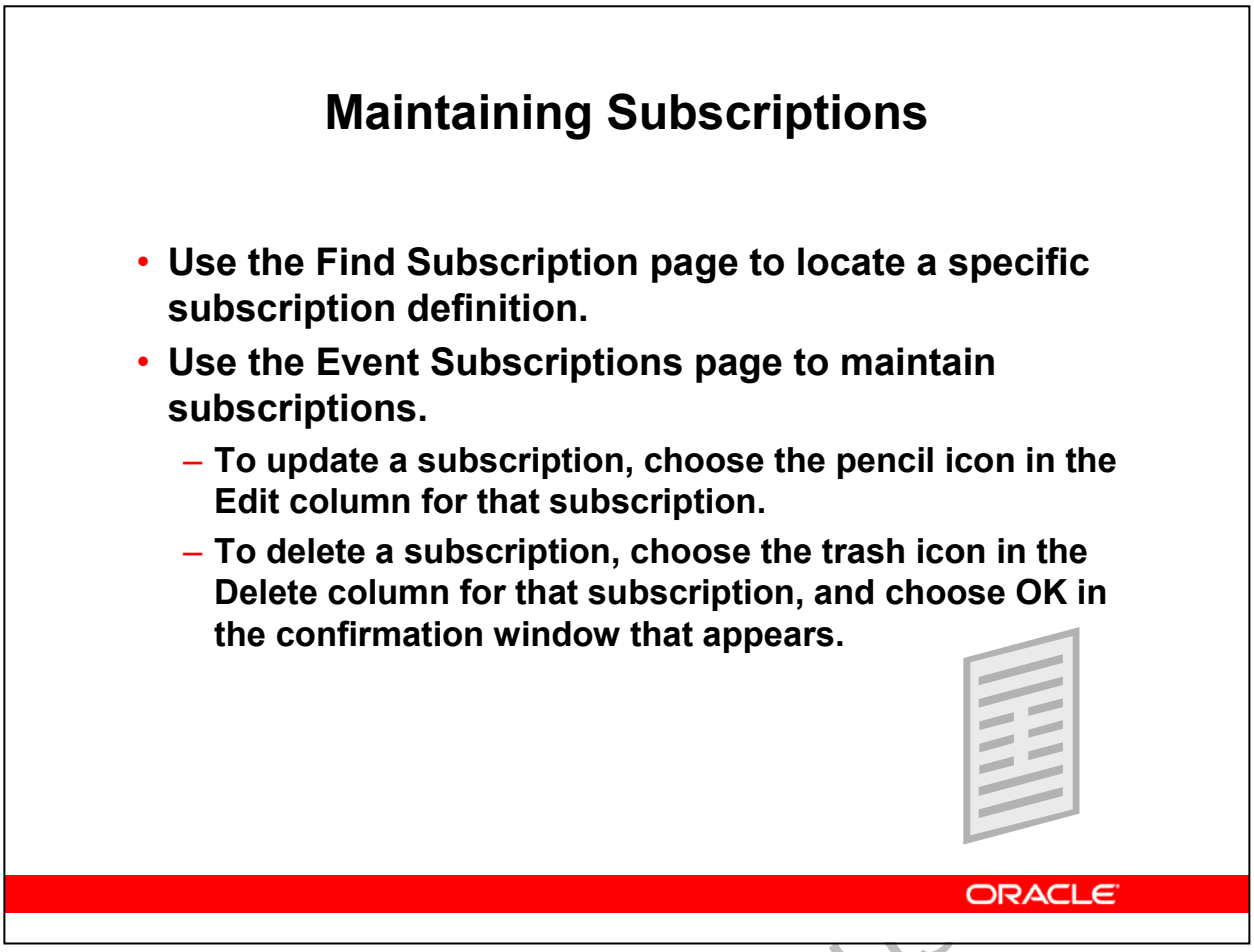
Subscription Actions: Sending an Event to an Agent

- To send an event to an agent, you must:
  - Specify in the subscription the To Agent that you want to receive the inbound message.
  - Either use the default rule function or include send processing in your custom rule function.
- You can also optionally:
  - Specify the Out Agent that you want to send the outbound message.
  - Specify the priority with which the recipient should dequeue the message.

ORACLE

Subscription Actions: Sending an Event to a Workflow Process

- If you do not specify an Out Agent, Oracle Workflow selects an outbound agent on the subscribing system whose queue type matches the queue type of the To Agent.
- If you want to send an event to an agent from within a custom rule function, call:
  - WF_EVENT.Send( ) to send the event message to an agent only
  - WF_RULE.Default_Rule( ) to include the default subscription processing that can send the event message both to a workflow process and to an agent



Maintaining Subscriptions

Finding Subscriptions

To access the Find Subscription web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_event_html.findsubscription

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find Subscription web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Event Subscriptions option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.

In the Find Subscription page, you can search for subscriptions by entering the following criteria:

- System
- Source Type
- Event
- Status

Modifying Subscriptions

To access the Event Subscriptions web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

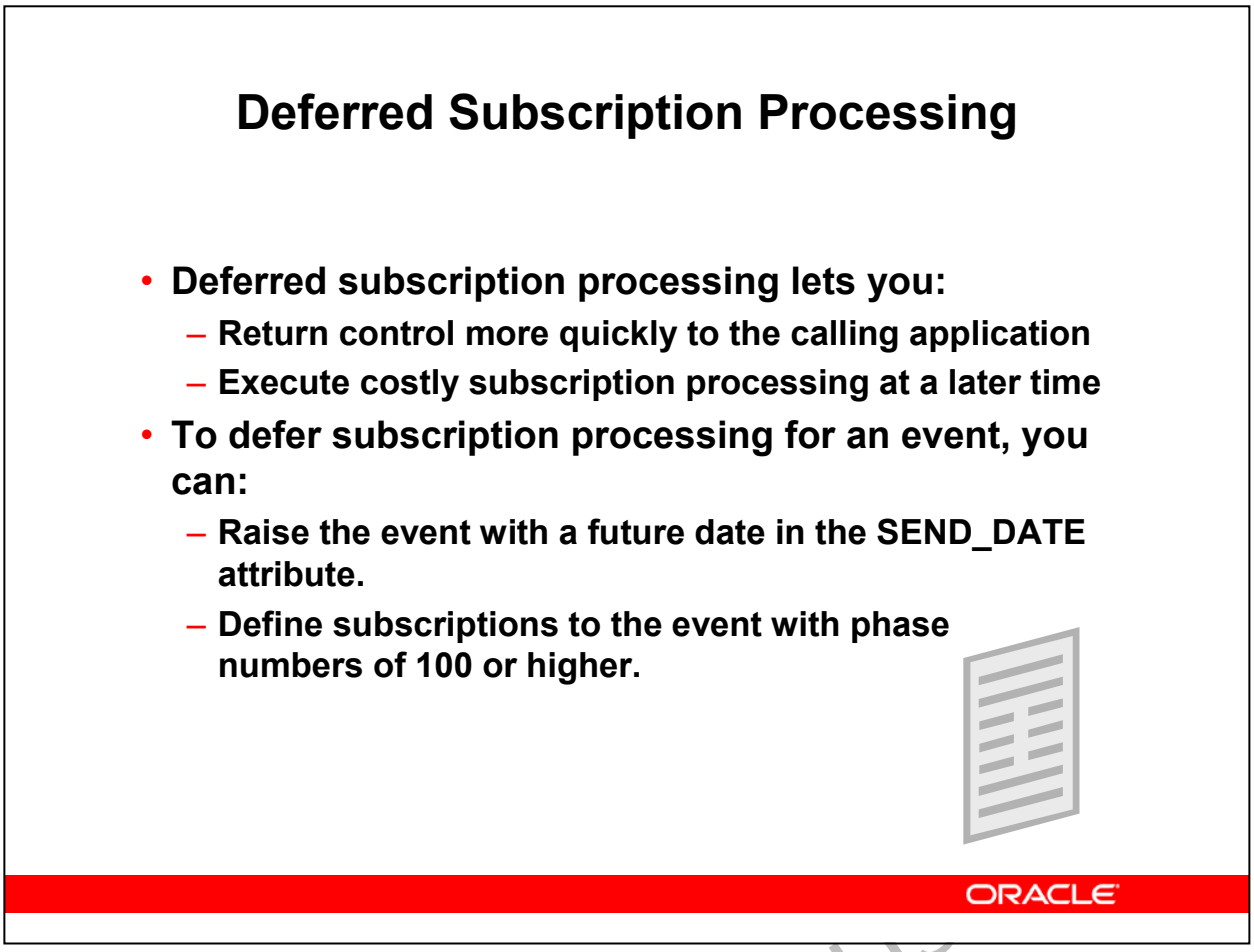
<*webagent*>/wf_event_html.listsubscriptions

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Event Subscriptions web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Add Event Subscriptions option from a Workflow administrator responsibility with Event Manager functionality.

Note: You must have workflow administrator privileges to access the Event Manager web pages.



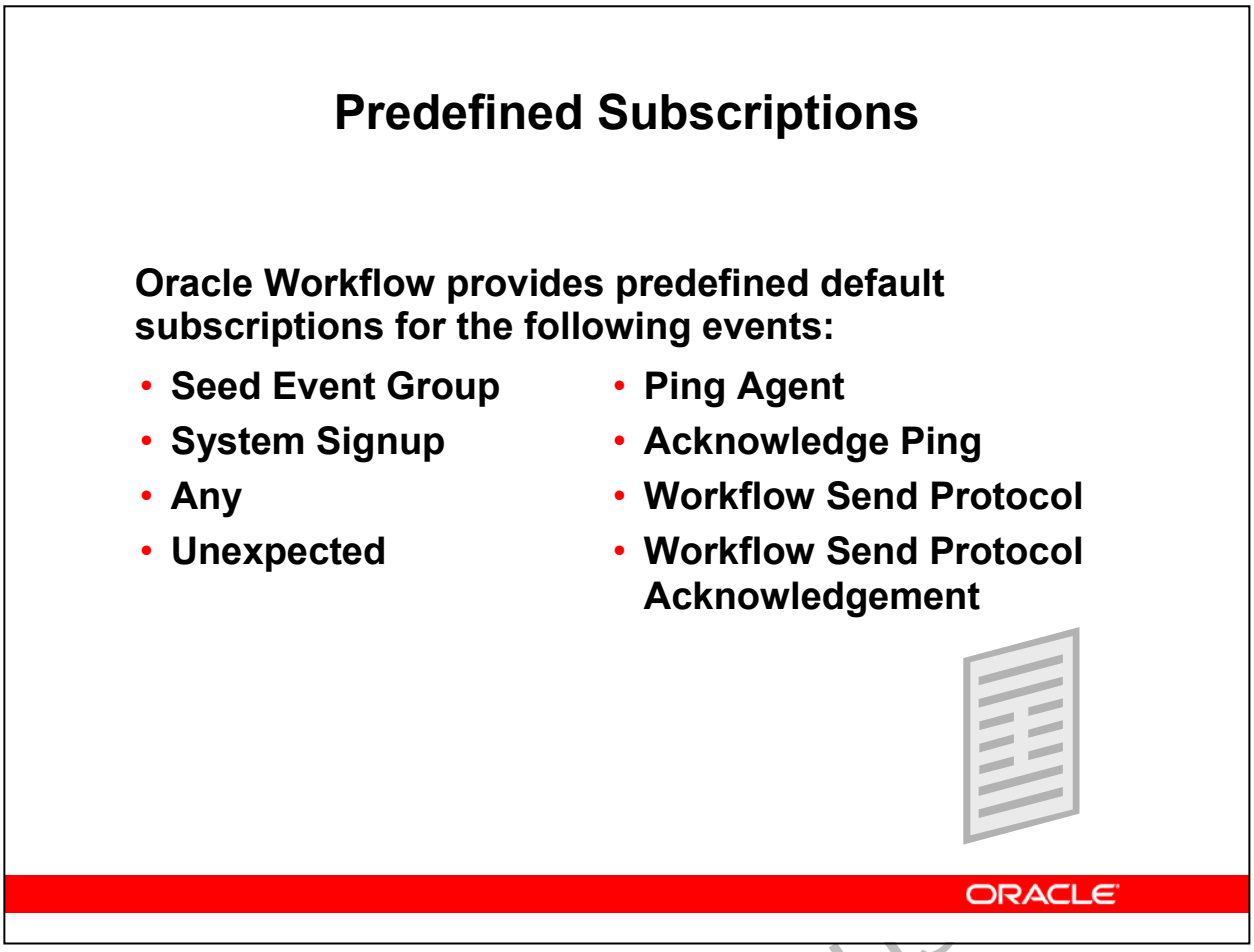
Deferred Subscription Processing

To defer subscription processing, you can :

- Raise the event with a future date in the SEND_DATE attribute. Use this method when you want to defer all subscription processing for a locally raised event until a particular effective date.
- Define subscriptions to the event with phase numbers of 100 or higher. Use this method when you want to defer processing of particular subscriptions for either local or external events.
- You can also set the dispatch mode of the Event Manager to deferred processing before raising the event. This method can be used to defer all subscription processing for a locally raised event. This method is not recommended, however, and should only be used in exceptional circumstances.

When subscription processing for an event is deferred by any of these methods, the event message is placed on the standard WF_DEFERRED queue associated with the WF_DEFERRED agent. You must schedule a listener to monitor the WF_DEFERRED agent. The listener dequeues event messages from the WF_DEFERRED agent in priority order. The event messages retain their original source type, whether Local or External. The amount of

time by which subscription processing for these events is deferred depends on the schedule defined for the listener, and, for future-dated events, on the specified effective date.



Predefined Subscriptions

The subscriber for all the predefined subscriptions is the local system. You can update, enable, or disable some of these subscriptions to perform the event processing that you want.

Seed Event Group

Oracle Workflow provides two predefined subscriptions to the Seed Event Group. These subscriptions can be triggered by any of the group's member events. The Seed Event Group includes all the event, event group, system, agent, and subscription definition events, as well as the Synchronize Event Systems event.

- The first subscription can send the Event Manager data to an agent and to a workflow process when one of the group member events is raised locally. To use this subscription, you must add the agent or workflow to which you want to send the data, and enable the subscription.
- The second subscription can load the Event Manager data into the local system when one of the group member events is received from an external source. To use this subscription, you must enable it.

System Signup Event

Oracle Workflow provides one predefined subscription to the System Signup event that loads the Event Manager data into the local system when the System Signup event is raised locally.

Any Event

Oracle Workflow provides three predefined subscriptions to the Any event.

- The first subscription can be triggered when an event is raised locally. To use this subscription, you must define the action for the subscription and enable it.
- The second subscription can be triggered when an event is received from an external source. To use this subscription, you must define the action for the subscription and enable it.
- The third subscription sends the event message to the Default Event Error process in the System: Error item type and raises an exception when an event is received from an Error source (that is, when it is dequeued from the WF_ERROR queue). To use this subscription, you must enable it.

Unexpected Event

Oracle Workflow provides three predefined subscriptions to the Unexpected event.

- The first subscription can send the event message to the Default Event Error process in the System: Error item type when an unexpected event is raised locally. To use this subscription, you must enable it.

  Attention: If you want to enable this subscription, be careful to consider all the events that can be raised on your local system and trigger the subscription. Many local events may be raised to which you do not want to subscribe. Additionally, if a large number of events are raised on the local system, enabling this subscription may flood the Business Event System.
- The second subscription sends the event message to the Default Event Error process in the System: Error item type when an unexpected event is received from an external source. This subscription allows your local system to handle any event messages received from external systems that you were not expecting.
- The third subscription sends the event message to the Default Event Error process in the System: Error item type when an unexpected event is received from an Error source (that is, when it is dequeued from the WF_ERROR queue).

  Attention: You must not change or disable the definition of the predefined Error subscription to the Unexpected event. If you disable this subscription, then the Event Manager will not be able to perform error handling for any events for which you have not defined custom Error subscriptions.

Ping Agent Events

- Ping Agent—Oracle Workflow provides one predefined subscription to the Ping Agent event that sends the Acknowledge Ping event back to the originating system when the Ping Agent event is received from an external source.
- Acknowledge Ping—Oracle Workflow provides one predefined subscription to the Acknowledge Ping event that sends the Acknowledge Ping event to the Detail Ping process in the Workflow Agent Ping/Acknowledge item type when the event is received from an external source.

Workflow Send Protocol Events

- Workflow Send Protocol—Oracle Workflow provides two predefined subscriptions to the Workflow Send Protocol event.
  - The first subscription sends the event message to the Workflow Event Protocol process in the Workflow Send Protocol item type when the Workflow Send Protocol event is raised locally. A subscription parameter specifies that the message requires an acknowledgement. You can add an outbound agent and inbound agent to the subscription to specify where you want the Workflow Event Protocol process to send the event message.
  - The second subscription sends the event message to the Workflow Event Protocol process in the Workflow Send Protocol item type when the Workflow Send Protocol event is received from an external source. You can optionally add an outbound agent and inbound agent to the subscription to specify that you want the Workflow Event Protocol process to send the event message on to another agent.
- Workflow Send Protocol Acknowledgement—Oracle Workflow provides one predefined subscription to the Workflow Send Protocol Acknowledgement Event that sends the event message to the Workflow Event Protocol process in the Workflow Send Protocol item type when the Workflow Send Protocol Acknowledgement event is received from an external source.

For more information, refer to the Predefined Workflow Events chapter in the *Oracle Workflow Guide*.

Practice - Defining a Subscription

Overview

In this practice, you will define and execute a new subscription.

- Define a subscription in the Event Manager. The subscription should run the predefined rule function WF_RULE.Success.

- Raise an event to trigger your subscription. Because the WF_RULE.Success function does not perform any processing other than returning the status code SUCCESS, there will be no result visible in the Workflow web pages after the subscription is executed. For an indication that the subscription was executed successfully, you can optionally check the Worklist to ensure that no error notification was generated.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.

- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.

- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility that includes Event Manager functionality. The username you use to log in should have this responsibility assigned to it.

Tasks

1. Define a subscription in the Event Manager.

   Solution:

   1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

2. Choose the Event Subscriptions link in standalone Oracle Workflow, or the Add Event Subscriptions link in Oracle Workflow embedded in Oracle E-Business Suite.

3. In the Event Subscriptions page, choose the Add Subscription button to open the Edit Subscription page.

4. In the System field, select the local system as the subscriber.

5. In the Source Type field, select Local.

6. In the Event Filter field, select the *XX*.oracle.workflow.bes.vacation.scheduled event that you defined in the Defining an Event practice.

7. Leave the Source Agent field blank.

8. Enter 10 in the Phase field.

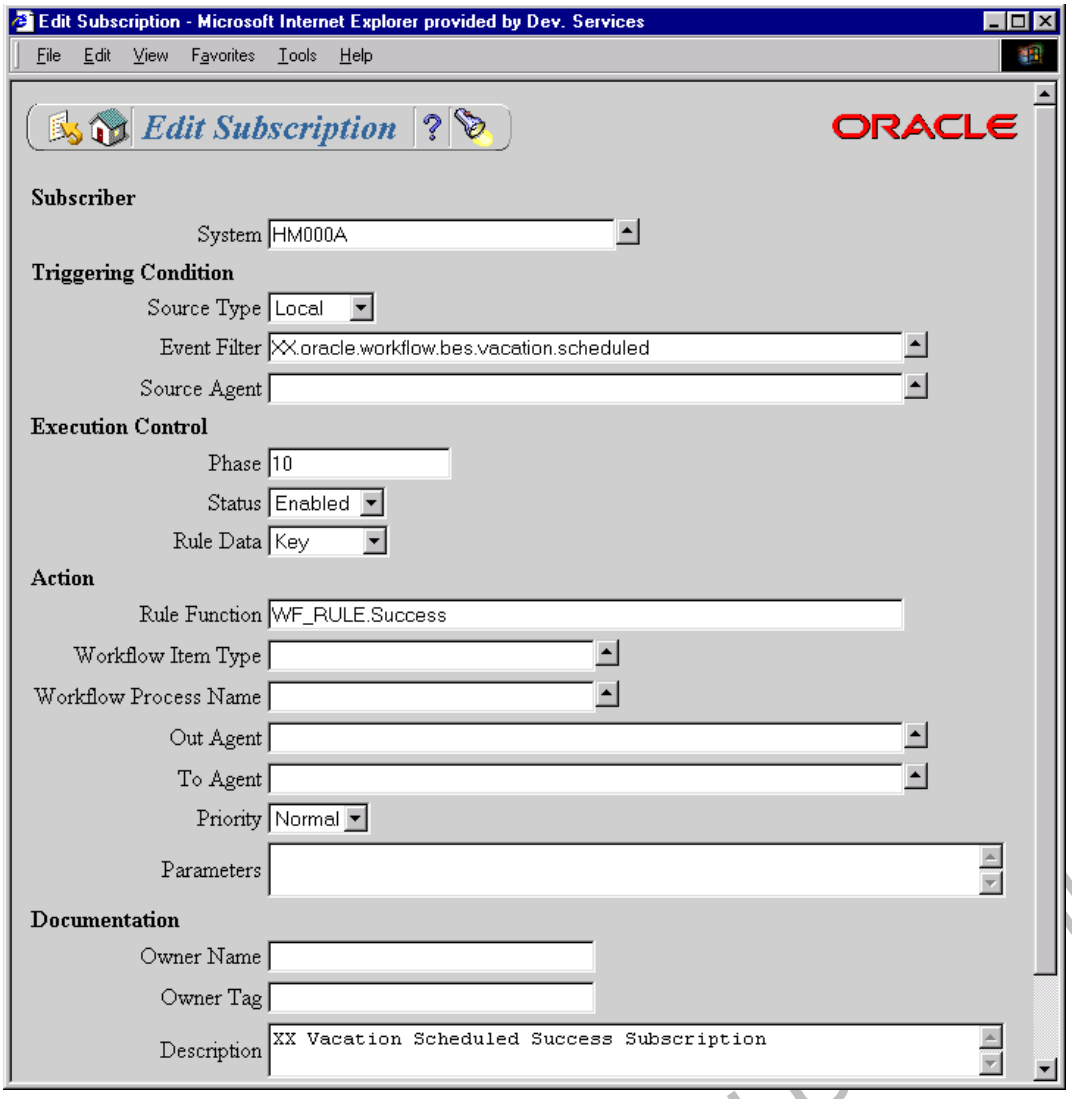
9. In the Status field, select Enabled.

10. In the Rule Data field, select Key.

11. Enter WF_RULE.Success in the Rule Function field.

12. Leave the Workflow Item Type, Workflow Process Name, Out Agent, To Agent, Parameters, Owner Name, and Owner Tag fields blank. Leave the Priority field set to the default value, which is Normal.

13. In the Description field, enter *XX* Vacation Scheduled Success Subscription.



14. Choose the Submit button to save the subscription.

2. Raise an event to trigger the subscription.

Solution:

15. Navigate back to the Oracle Workflow home page.

16. Choose the Raise Event link in standalone Oracle Workflow, or the Raise Business Event link in Oracle Workflow embedded in Oracle E-Business Suite.

17. In the Event Name field, select your *XX*.oracle.workflow.bes.vacation.scheduled event.

18. In the Event Key field, enter a unique event key such as *XX*51.

19. For the purposes of this practice, since you will not perform further processing on this event, you do not need to enter any event data. Leave the Event Data field blank.

20. Choose the Submit button to raise the event.

21. Choose OK in the confirmation window.

22. The WF_RULE.Success function does not produce any visible results in the Workflow web pages. However, for an indication that the subscription was executed successfully, you can optionally choose the Worklist link from the Oracle Workflow home page to ensure that no error notification was generated.

Practice - Sending an Event to an Agent

Overview

In this practice, you will define and execute subscriptions to send an event to an agent.

- Define a subscription that sends an event to a local inbound agent.
- Define another subscription that handles the event by returning the status code SUCCESS when the event is received by that agent.
- Raise the event to trigger your first subscription. The second subscription will be triggered in turn when the first subscription sends the event to the agent.
- Review the messages on the local inbound and outbound queues to confirm that the sending subscription was executed successfully.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility that includes Event Manager functionality. The username you use to log in should have this responsibility assigned to it.

Tasks

1. Define a subscription that sends an event to a local inbound agent.

   Solution:

   1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow

administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

2. Choose the Event Subscriptions link for standalone Oracle Workflow or the Add Event Subscriptions link for Oracle Workflow embedded in Oracle E-Business Suite.

3. In the Event Subscriptions page, choose the Add Subscription button to open the Edit Subscription page.

4. In the System field, select the local system as the subscriber.

5. In the Source Type field, select Local.

6. In the Event Filter field, select the *XX*.oracle.workflow.bes.vacation.scheduled event that you defined in the Defining an Event practice.

7. Leave the Source Agent field blank.

8. Enter 20 in the Phase field.

9. In the Status field, select Enabled.

10. In the Rule Data field, select Key.

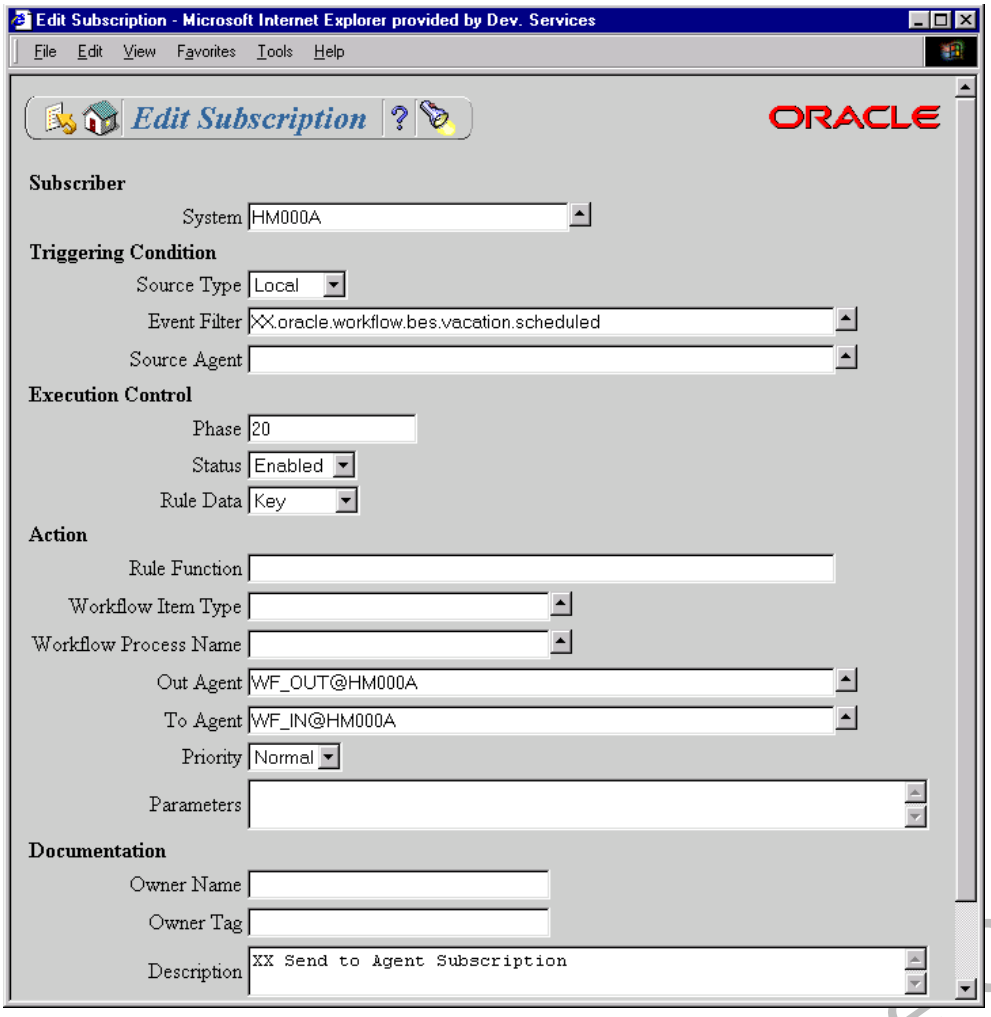
11. Leave the Rule Function field blank to use the default rule function to send the event to an agent.

12. Leave the Workflow Item Type, Workflow Process Name, Parameters, Owner Name, and Owner Tag fields blank. Leave the Priority field set to the default value, which is Normal.

13. In the Out Agent field, select the agent WF_OUT@<*local system*>.

14. In the To Agent field, select the agent WF_IN@<*local system*>.

15. In the Description field, enter XX Send to Agent Subscription.



16. Choose the Submit button to save the subscription.

2. Define another subscription that handles the event by returning the status code SUCCESS when the event is received by that agent.

Solution:

17. Navigate back to the Event Subscriptions page.

18. Choose the Add Subscription button to open the Edit Subscription page.

19. In the System field, select the local system as the subscriber.

20. In the Source Type field, select External.

21. In the Event Filter field, select the *XX*.oracle.workflow.bes.vacation.scheduled event that you defined in the Defining an Event practice.

22. Leave the Source Agent field blank.

23. Enter 20 in the Phase field.

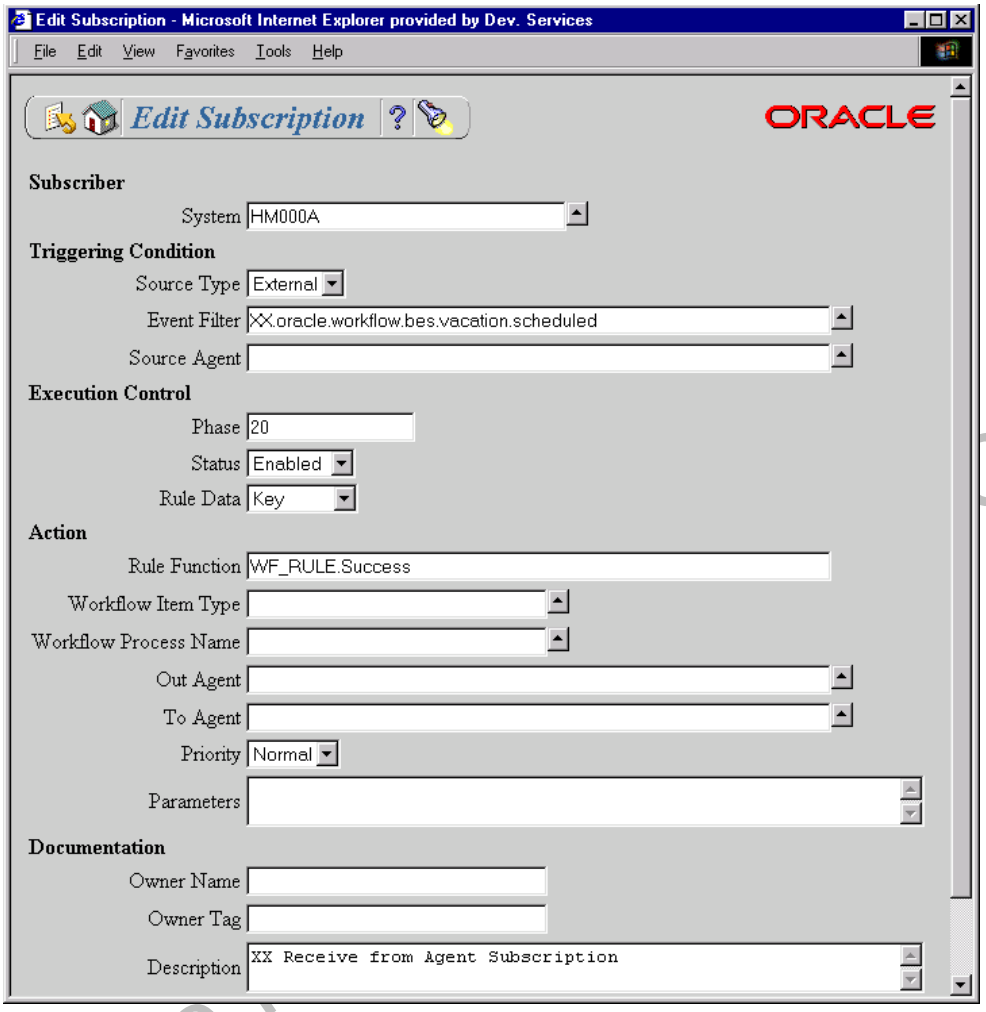
24. In the Status field, select Enabled.

25. In the Rule Data field, select Key.

26. Enter WF_RULE.Success in the Rule Function field.

27. Leave the Workflow Item Type, Workflow Process Name, Out Agent, To Agent, Parameters, Owner Name, and Owner Tag fields blank. Leave the Priority field set to the default value, which is Normal.

28. In the Description field, enter *XX* Receive from Agent Subscription.



29. Choose the Submit button to save the subscription.

3. Raise the event to trigger your first subscription.

Solution:

30. Navigate back to the Oracle Workflow home page.

31. Choose the Raise Event link for standalone Oracle Workflow or the Raise Business Event link for Oracle Workflow embedded in Oracle E-Business Suite.

32. In the Event Name field, select the *XX*.oracle.workflow.bes.vacation.scheduled event.

33. In the Event Key field, enter a unique event key such as *XX*52.

34. In the Event Data field, enter the following sample XML document:

```
<DATA>Vacation has been scheduled</DATA>
```

35. Choose the Submit button to raise the event.

36. Choose OK in the confirmation window.

4. Review the messages on the local inbound and outbound queues.

Solution:

37. Navigate back to the Oracle Workflow home page.

38. Choose the Event Queue Summary link.

39. In the Event System Local Queues page, review the number of messages on the local WF_IN and WF_OUT agents' queues.

40. Choose the flashlight icon for the WF_OUT agent.

41. In the Find Standard Event Queue Messages page, select Any in the Status field and choose Go.

42. In the Local Queue Messages page, locate the message with your event key. Choose the flashlight icon for that message to review the XML document contained in the message.

43. Navigate back to the Event System Local Queues page.

44. Choose the flashlight icon for the WF_IN agent.

45. In the Find Standard Event Queue Messages page, select Any in the Status field and choose Go.

46. In the Local Queue Messages page, locate the message with your event key. Choose the flashlight icon for that message to review the XML document contained in the message.

The amount of time it takes for the inbound message to appear on the WF_IN queue and be processed depends on the propagation scheduled for WF_OUT and the listener being run for WF_IN.

When the event message is received by the WF_IN agent, the Event Manager executes the External subscription you defined to that event. Because that subscription simply runs the WF_RULE.Success function to return the status code SUCCESS, it does not produce any visible results in the Workflow web pages. However, it is important to define the External subscription to handle the event when it is received in order to avoid having the Event Manager generate an unexpected event error notification.

Summary

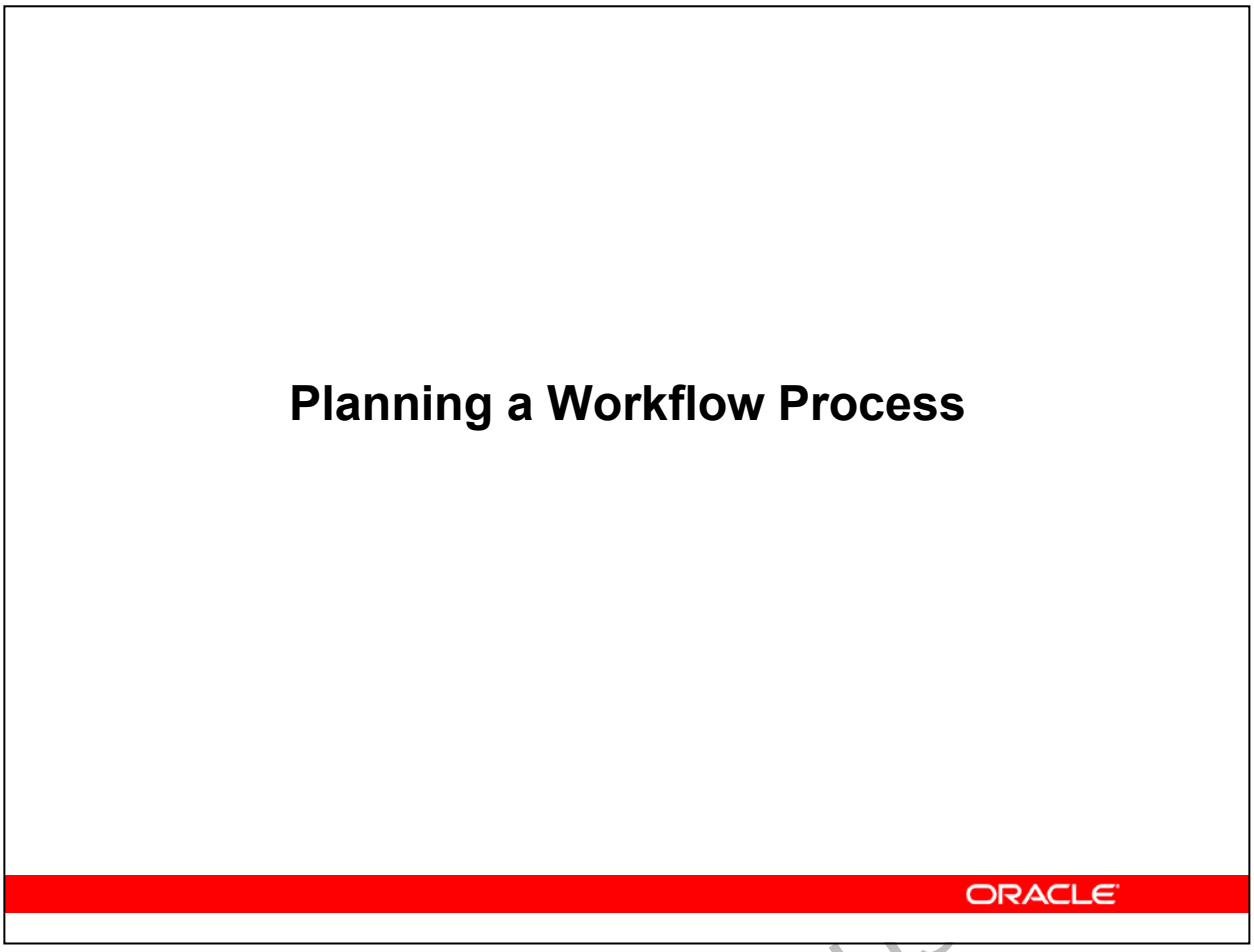
In this lesson, you should have learned how to:

- **Describe the types of processing a subscription can include.**
- **Define subscriptions.**
- **Use predefined Oracle Workflow subscriptions.**

ORACLE

Planning a Workflow Process

Chapter 6

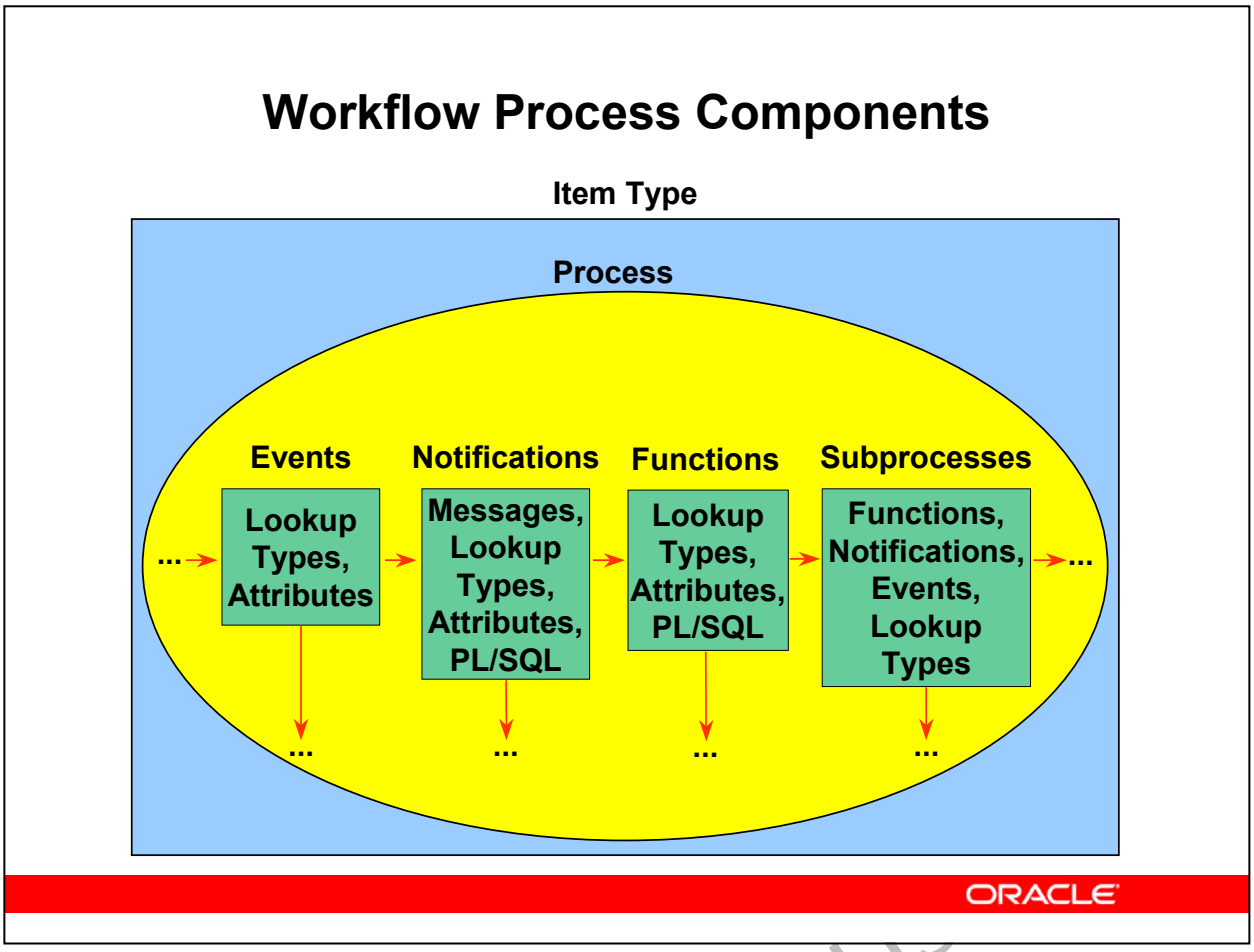


Objectives

After completing this lesson, you should be able to do the following:

- **Identify the components of a workflow process.**
- **Plan a workflow process.**
- **Incorporate standard activities within a process.**

ORACLE

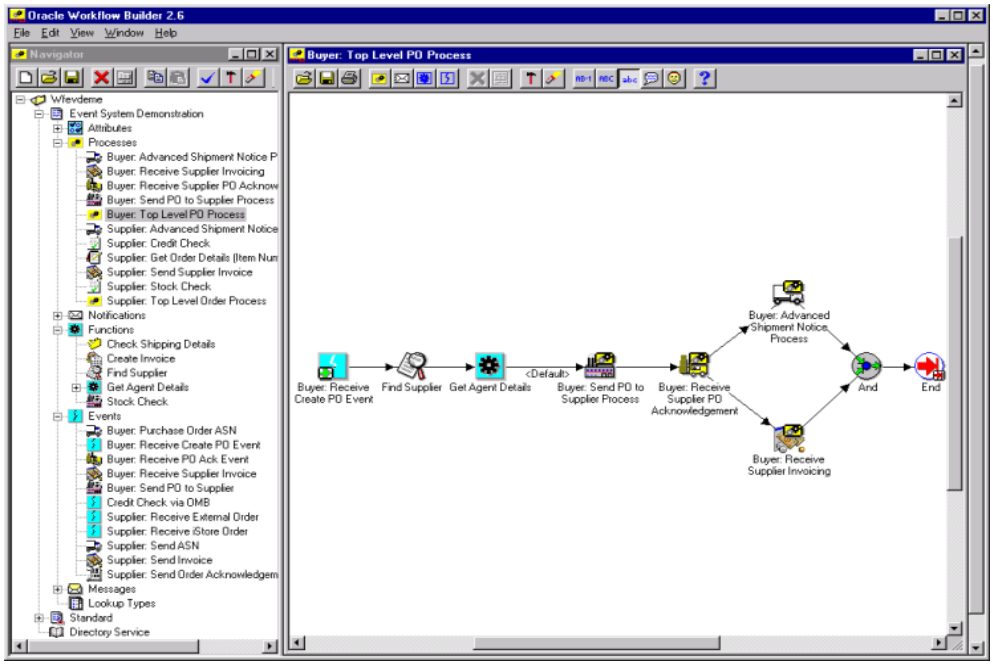


Workflow Process Components

- Data Store: A database connection or flat file that holds a workflow process definition.
- Item Type: A grouping of workflow components into a high level category. All components of a workflow process must be associated with a specific item type. An item type can contain multiple processes.
- Item Type Attribute: A feature of the item type which stores information that can be globally referenced by any activity in a process. Also referred to as an item attribute.
- Process Activity: A series of actions that need to be performed to accomplish a business goal. A process is represented by a workflow diagram. A process can include function activities, notification activities, event activities, and other process activities (subprocesses), as well as the transitions between these activities.
- Event Activity: A business event modeled as an activity so that it can be included in a workflow process.
- Notification Activity: A unit of work that requires human intervention. A notification activity sends a message to a performer.

- Function Activity: An automated unit of work, usually defined as a PL/SQL stored procedure. A function activity can also run an external function. In the standalone version of Oracle Workflow, a function activity can also run a Java program on the middle tier.
- Message: The information sent by a notification activity. The message may request the performer (the role receiving the message) to do some work or may simply provide information.
- Lookup Type: A list of values that can be referenced by any activity in a workflow process. The values in the list are called lookup codes.
- Transition: The relationship that defines the completion of one activity and the activation of another activity within a process. In a process diagram, a transition is represented as an arrow between two activities.
- Item: A specific business document or transaction.
- Process Instance: A unique item being managed by a process.

Oracle Workflow Builder

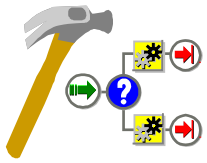


ORACLE

Oracle Workflow Builder

The Oracle Workflow Builder:

- **Is used to create the components of a process**
- **Consists of a navigator tree, which lists all process components, and a process diagram window**
- **Groups all components into an item type**

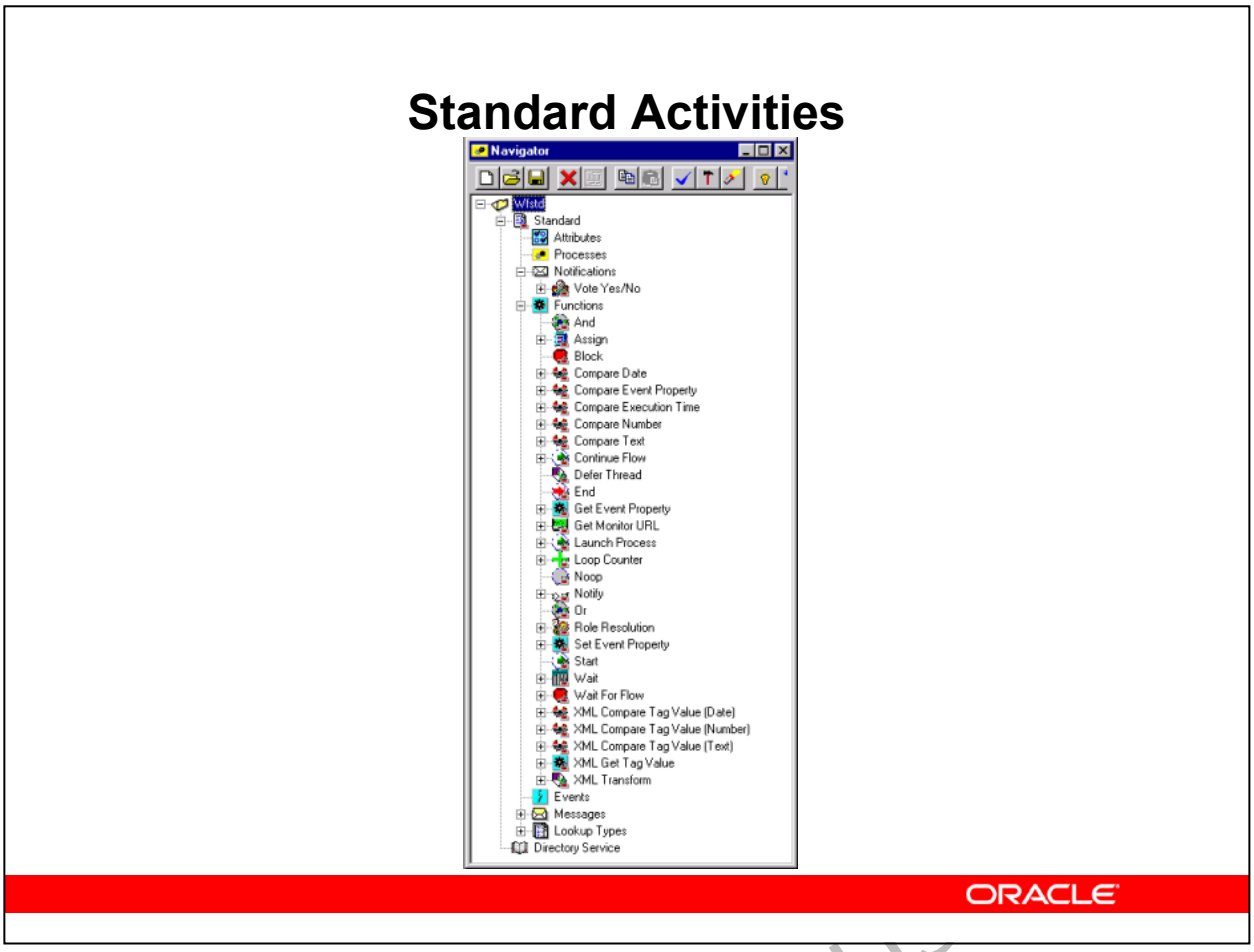


ORACLE

Oracle Workflow Builder

In the navigator tree, each item type includes the following branches:

- Attributes
- Processes (Subprocesses)
- Notifications
- Functions
- Events
- Messages
- Lookup Types



Standard Activities

The Standard item type includes function and notification activities provided by Oracle Workflow that can be included in process diagrams associated with other item types. These activities include:

- And
- Or
- Compare Date
- Compare Number
- Compare Text
- Compare Execution Time
- Wait
- Block
- Defer Thread
- Launch Process
- Noop

- Loop Counter
- Start
- End
- Role Resolution
- Notify
- Vote Yes/No
- Continue Flow
- Wait for Flow
- Assign
- Get Monitor URL
- Get Event Property
- Set Event Property
- Compare Event Property
- XML Get Tag Value (Date)
- XML Get Tag Value (Number)
- XML Get Tag Value (Text)
- XML Compare Tag Value (Date)
- XML Compare Tag Value (Number)
- XML Compare Tag Value (Text)
- XML Transform

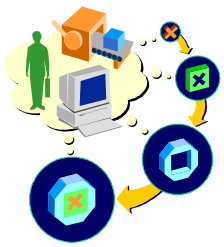
For a full explanation of these standard activities, refer to the Predefined Workflow Activities chapter in the *Oracle Workflow Guide*.

Note: The XML Get Tag Value, XML Compare Tag Value, and XML Transform activities are external Java function activities and require the Java Function Activity Agent as well as a background engine to complete their processing. External Java function activities are currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.

Note: If you are using Oracle XML Gateway with the Oracle E-Business Suite, you can also use standard activities supporting XML messaging that are provided in the XML Gateway Standard item type. For more information, refer to the Integrating Oracle XML Gateway with Oracle Workflow Business Event System chapter in the *Oracle XML Gateway User's Guide*.

Planning a Workflow Process

1. Identify your business needs.
2. Identify the activities that will accomplish those needs. For example:
   a. What is each activity's type: Function, Notification, Event, or Process?
   b. Will an activity's result affect the transition that the Workflow Engine takes? If so, define the activity's possible results as lookup codes in a lookup type.
   c. Which standard activities can you leverage to minimize the development work required for your process?



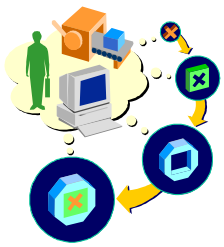
ORACLE

Planning a Workflow Process

Consider modeling your process so that it can be customized and extended in the future.

Planning a Workflow Process

3. Identify how the process will be initiated. For example:
   a. What condition initiates the process? Is it a completed transaction, a database exception, a button on a form, or some other condition?
   b. Will there be an interface to initiate the process?
   c. Will the process begin with a Receive event activity, or with a Start function activity?
4. Identify the possible results of the process. List the possible results as lookup codes in a lookup type.



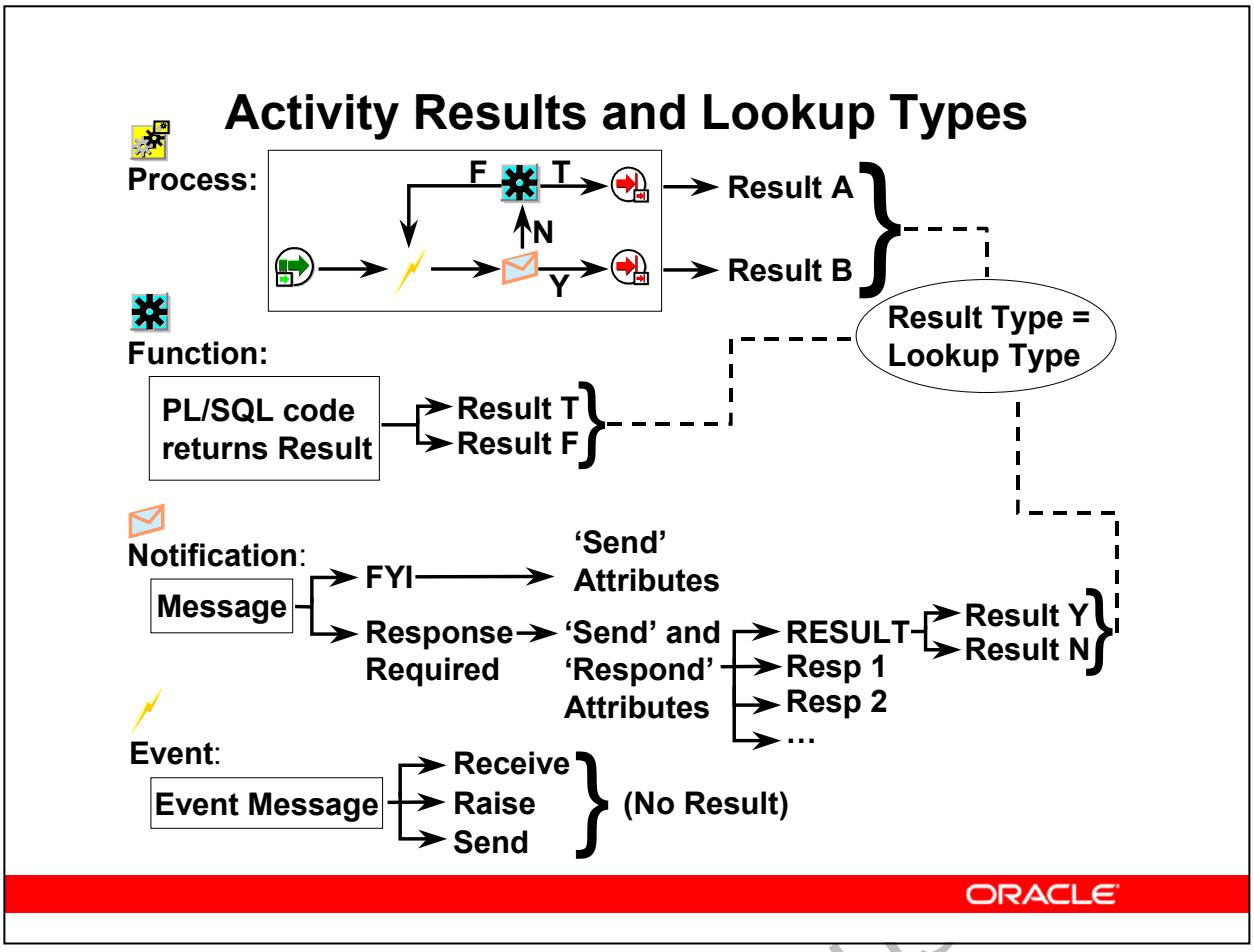
ORACLE

Planning a Workflow Process

If the process will begin with a Receive event activity, the calling application must raise an event, either with the WF_EVENT.Raise API or with a Raise event activity in another workflow process. The event must trigger a subscription that sends the event message to the workflow process that has the Receive event activity, which must be marked as a Start activity node.

If the process will begin with a standard Start function activity, the application must execute a procedure that calls the appropriate Workflow Engine APIs, either WF_ENGINE.CreateProcess and WF_ENGINE.StartProcess, or WF_ENGINE.LaunchProcess, which is a wrapper combining the CreateProcess and StartProcess APIs. The activity must be marked as a Start activity node.

Note: Using the standard Start function activity lets you visually represent the start of the process in your diagram. Alternatively, you can choose to start the process with another function, notification, or process activity. In all cases, however, the activity must be marked as a Start activity node.



Activity Results and Lookup Types.

Process Activities

The diagram for a process activity can include functions, notifications, events, and other processes.

A process activity can be completed with a specific result that is defined in a lookup type. For example, the New Hire Process can be completed with a final result of Accepted or Declined, both of which are codes defined in the Applicant Response lookup type.

Function Activities

The PL/SQL procedure for a function activity can return some result that determines the next transition the Workflow Engine takes. That result should be defined as a code in a lookup type.

Notification Activities

The message sent by a notification activity can simply convey information or it can require a response.

- If the message only provides information (such as an "FYI"), then create and include "Send" message attributes in the message as necessary.

- If the message requires a response, then create “Send” message attributes that provide enough information for the recipient to respond, and also create “Respond” message attributes that prompt a response.
- If a particular response defines the actual result of the notification activity, then complete the Result tab for the message. The information you enter creates a special “Respond” message attribute that has an internal name of “RESULT”. The “RESULT” attribute has a data type of lookup and must reference a lookup type that defines the possible result codes and matches the result type for the notification activity.

Event Activities

An event activity can either receive, raise, or send a business event from the Business Event System. An event activity does not have a result, but you can branch on the contents of an event message using comparison activities provided in the Standard item type.

Practice - Planning a Workflow Process

Overview

In this practice, you will plan a workflow process in preparation for defining it in the Oracle Workflow Builder.

Tasks

1. Plan a workflow process.

   Solution:

   1. On paper, sketch a diagram of a vacation proposal process. The business process is as follows:
      - A vacation proposal is sent from a requestor to an approver, prompting the proposal recipient to approve or reject. The approver can respond with comments.
      - If the proposal is approved, a notification is sent to the requestor reporting that the vacation is approved.
      - If the proposal is rejected, a notification is sent to the requestor reporting that the vacation is rejected.

   2. Determine what activities will make up the process, and make a sketch placing the activities in the approximate order in which they will occur.

      Hint: The process should include only start, end, and notification activities.

   3. Draw arrows between the activities to indicate the process flow.

Summary

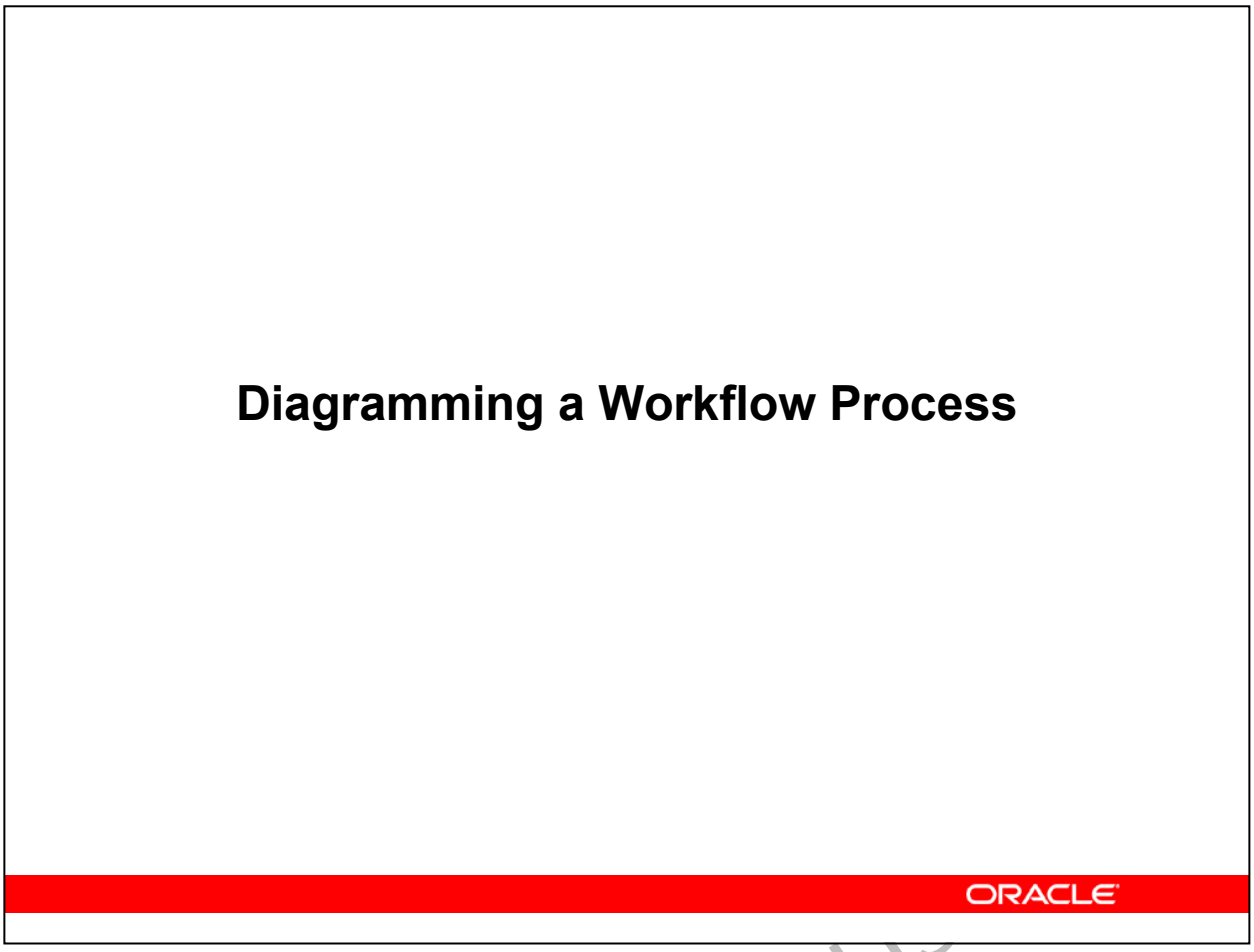
In this lesson, you should have learned how to:

- **Identify the components of a workflow process.**
- **Plan a workflow process.**
- **Incorporate standard activities within a process.**

ORACLE

Diagramming a Workflow Process

Chapter 7



Objectives

After completing this lesson, you should be able to do the following:

- **Diagram a workflow process.**
- **Define an item type.**
- **Define a process activity.**
- **Review the process definition for an item type.**

ORACLE

Creating a New Workflow Process

You can create new workflow processes by two different design methods:

- **Top-Down Design: If you prefer to approach your design from a high level, you can first sketch out the process diagram with activities, and then go back later to create the supporting objects for each activity.**
- **Bottom-Up Design: If you prefer to take a more programmatic approach to your design, you can first define each of the supporting objects of your process before beginning to create a higher level process diagram.**

ORACLE

Creating a New Process from Top Down

1. Use the Quick Start Wizard to create the process framework, including the item type and initial proces activity.
2. Define function, notification, event, or process activities without underlying dependencies.
3. Diagram the relationship of the activities.
4. Define item type attributes for the process.
5. Define lookup types and lookup codes.
6. Define messages.
7. Define message attributes.
8. Update processes, functions, events, and notifications with underlying components.

ORACLE

Creating a New Process from Bottom Up

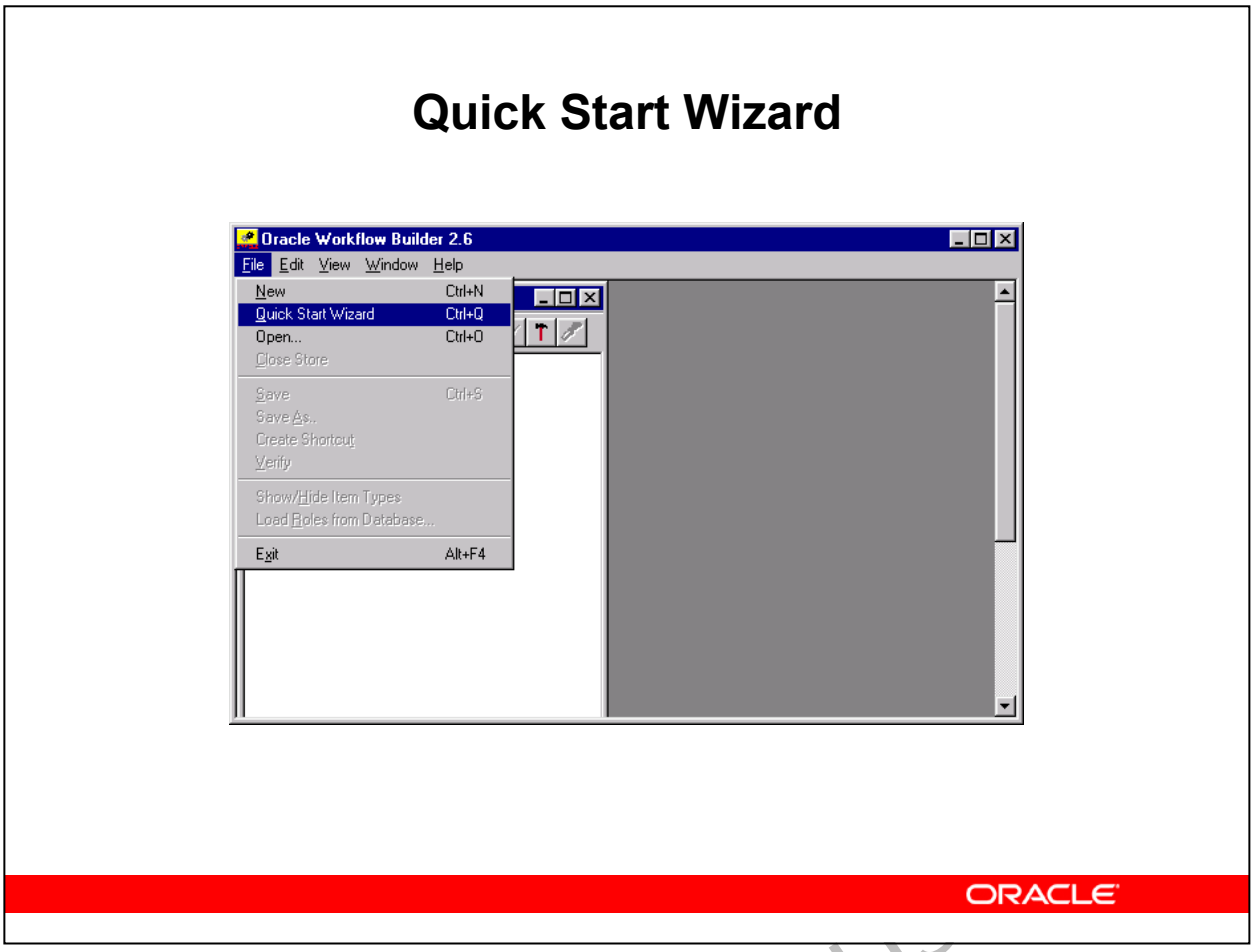
1. Define the item type of the process.
2. Define the item type attributes for the process.
3. Define lookup types and lookup codes.
4. Define messages.
5. Define message attributes.
6. Define function, notification, event, or process activities.
7. Diagram the relationship of the activities.

ORACLE

Diagramming a Process

1. To begin drawing a process diagram, you must display the process window.
   - Use the Quick Start Wizard to create a process activity and initial diagram with Start and End nodes.
   - Display the process window for a process activity that is defined in the navigator tree.
     - Double-click the process activity on the navigator tree.
     - Select the process activity and press Ctrl + E.
     - Select the process activity and choose Process Details from the Edit menu.

ORACLE

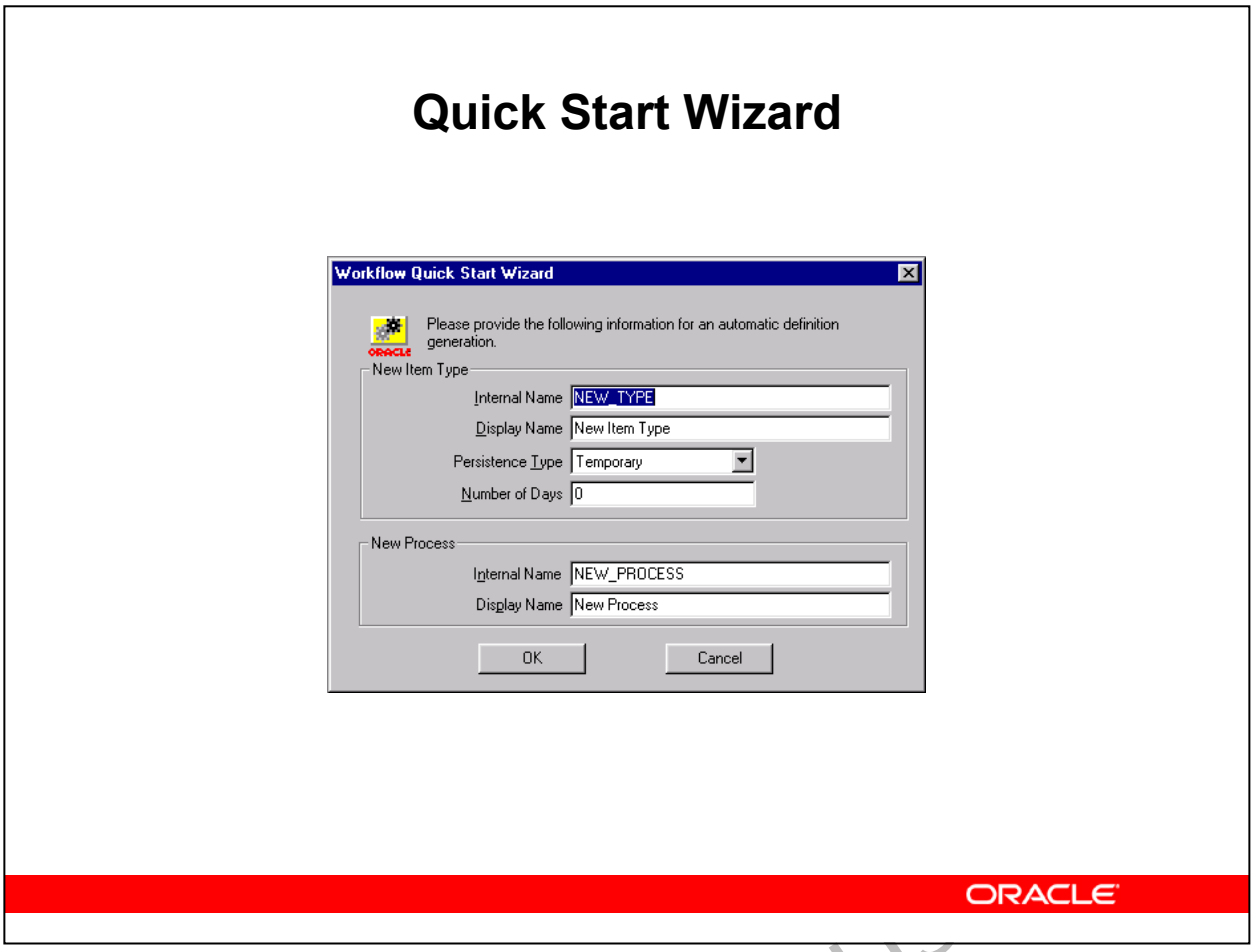


Quick Start Wizard

If you want to use the top-down design method, you can use the Quick Start Wizard to jump-start the creation of a new item type and process.

To initiate the Quick Start Wizard in Oracle Workflow Builder:

- Choose Quick Start Wizard from the File menu
- Click the Quick Start Wizard button on the navigator tree toolbar
- Select the navigator tree window and press Ctrl + Q



Quick Start Wizard

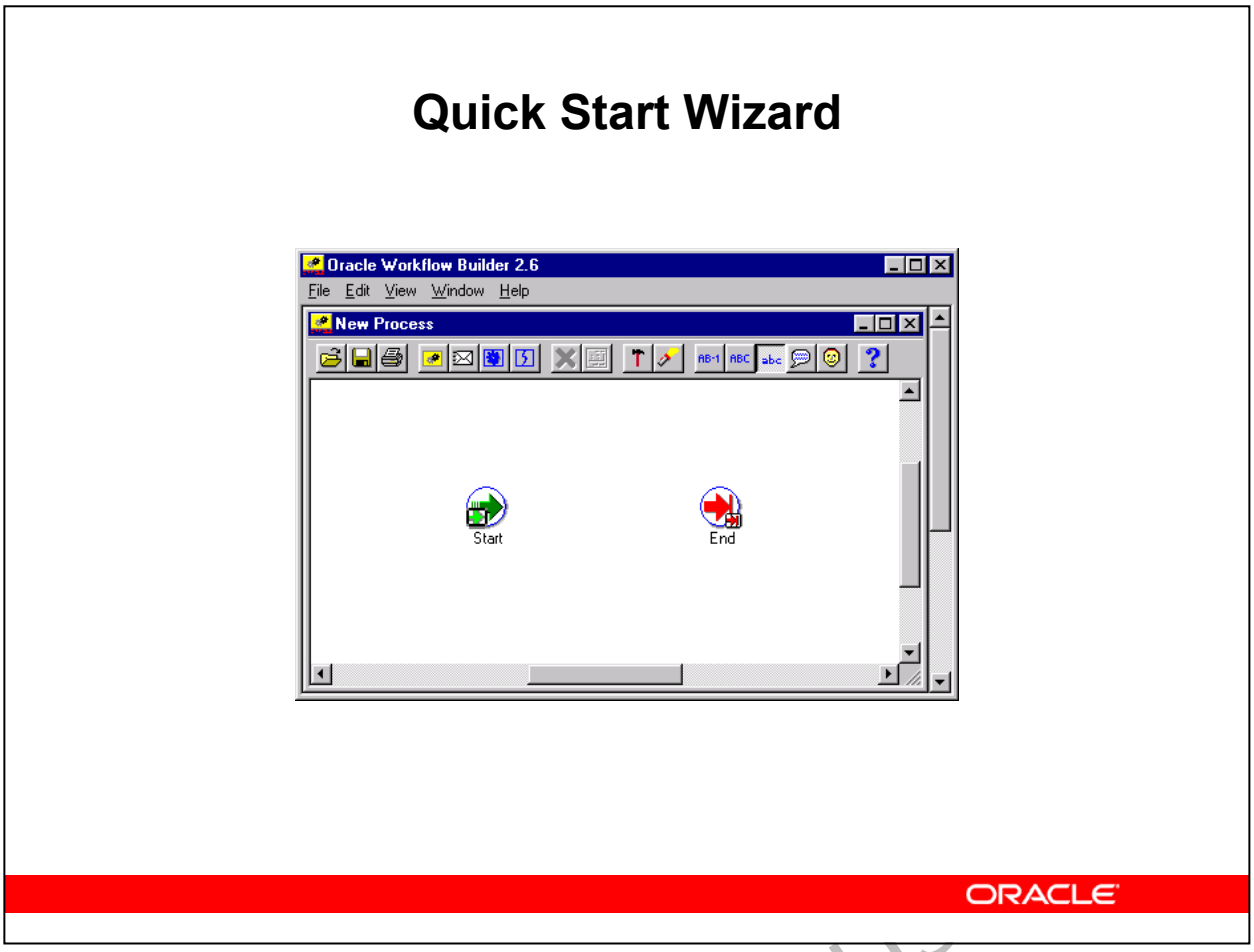
The Quick Start Wizard prompts for the minimum information necessary to create a new item type and process.

New Item Type

- Internal Name: Specify an all uppercase internal name with a maximum of eight characters. Do not include colons or leading or trailing spaces.
- Display Name: Enter a translatable display name for the item type.
- Persistence Type: Specify Temporary or Permanent persistence for the status audit trail of the item type. The persistence type controls when the run-time process status information is eligible for purging after the process is complete.
- Number of Days: If the persistence type is Temporary, specify the number of days from the time an item type instance completes before its status audit trail can be purged. •

New Process

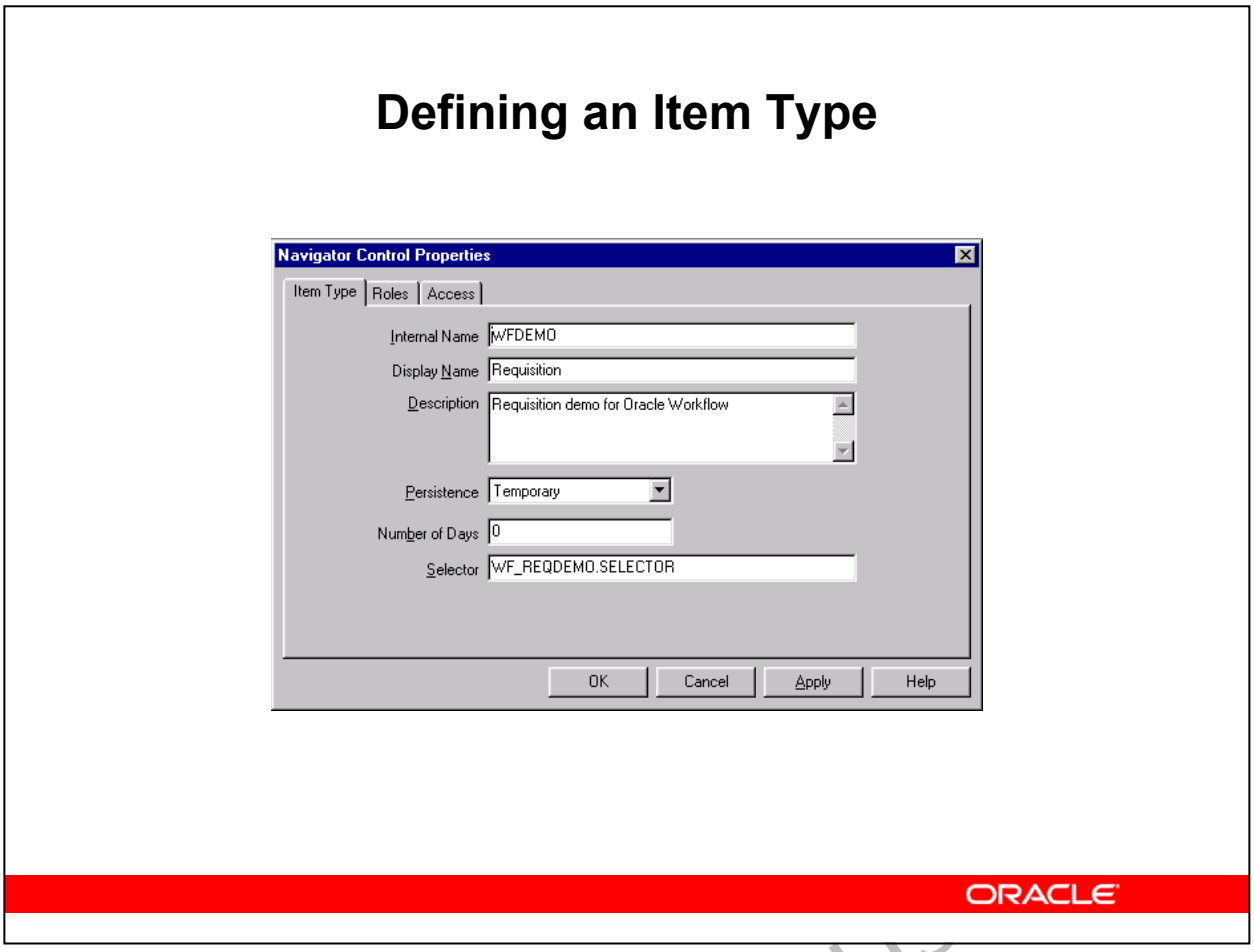
- Internal Name: Specify an all uppercase internal name. Do not include colons or leading or trailing spaces.
- Display Name: Enter a translatable display name for the process activity.



Quick Start Wizard

The Quick Start Wizard does the following:

- Creates a new data store in the Navigator window.
- Uses the information you entered in the Workflow Quick Start Wizard window to create a new item type and process activity in the data store.
- Loads the Standard item type into the new data store so that you can include standard activities in your new process.
- Opens the process window for the new process activity you defined. The process window automatically displays a Start node and an End node.



Defining an Item Type

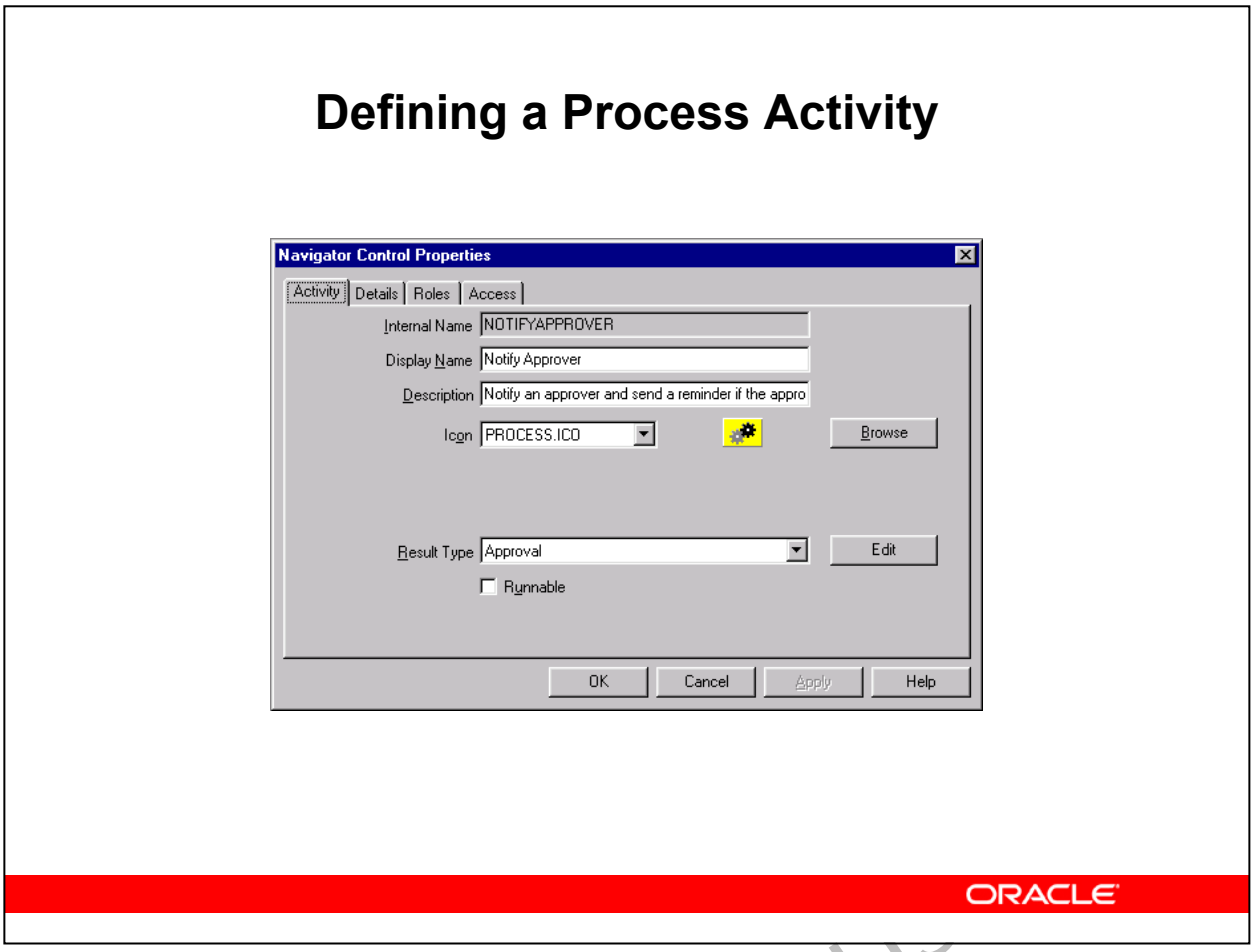
You can also define a new item type manually without using the Quick Start Wizard.

To define an item type:

1. Item types are defined in the navigator tree within a data store. Select a data store, or create a new data store by choosing New from the File menu.
2. Create a new item type by selecting New > Item Type from the Edit menu.
3. In the Item Type property page, enter an all uppercase internal name with a maximum of eight characters. Do not include colons or leading or trailing spaces.
4. Enter a translatable display name for the item type.
5. Enter a description of the item type.
6. Specify Temporary or Permanent persistence for the status audit trail of the item type. The persistence type controls when the run-time process status information is eligible for purging after the process is complete.
7. If the persistence type is Temporary, specify the number of days from the time an item type instance completes before its status audit trail can be purged.
8. Optionally enter the name of a selector/callback function for the item type.

- The item type selector/callback function is a PL/SQL procedure that the Workflow Engine can call to reset or test context information for an item type before executing an activity.
- An item type can have multiple workflow processes associated with it. If it does, the selector/callback function can also identify the appropriate process to run.
- Associating a selector/callback function with an item type makes the item type better suited to future customizations.

9. Choose Apply to save your changes.
10. Optionally select the Access tab to set the access levels allowed to modify this item type.



Defining a Process Activity

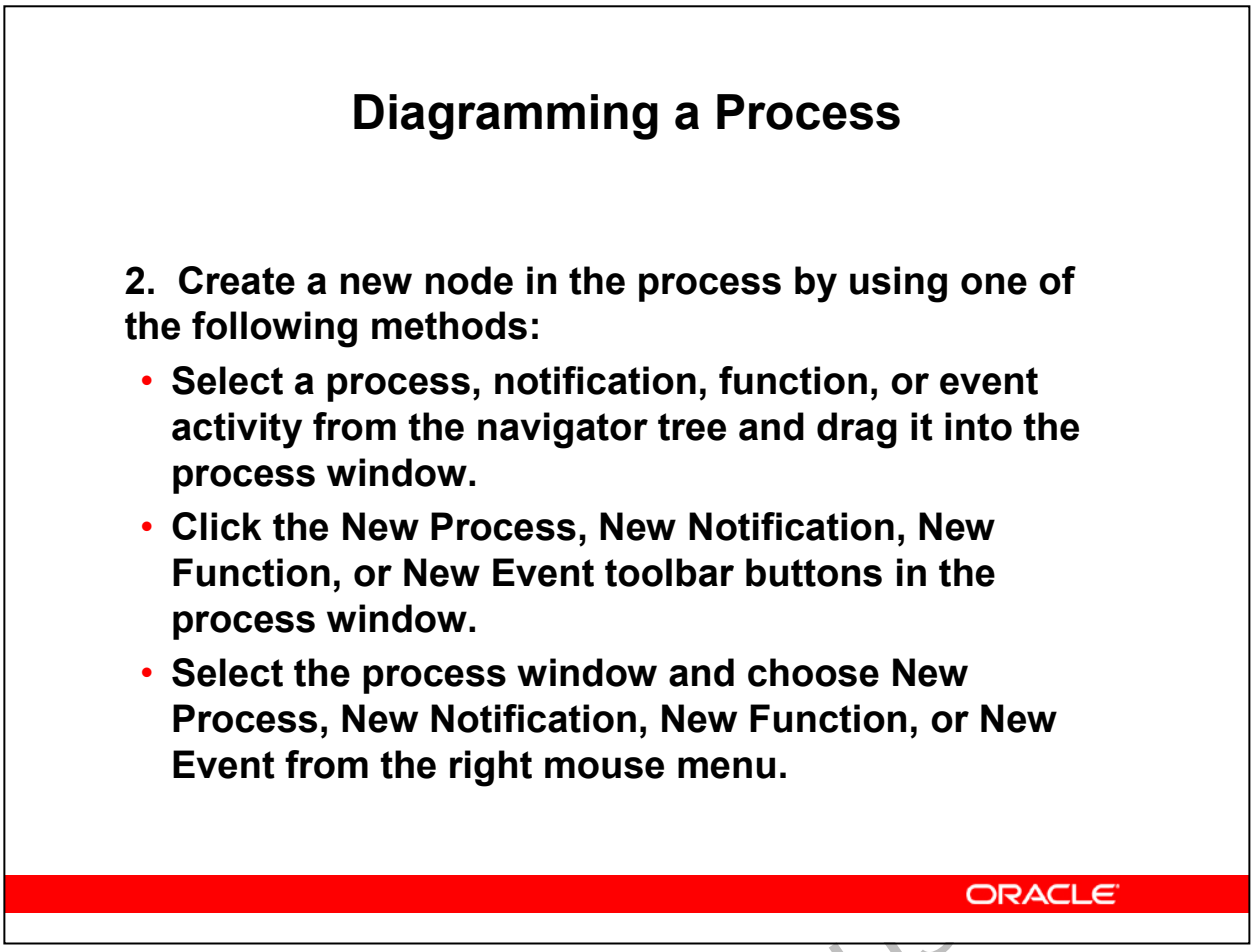
You can also define a new process activity manually without using the Quick Start Wizard. A process activity must be defined before a business process can be diagrammed. Process activities must be associated with an item type and are created in the navigator tree beneath the Processes branch of the item type.

To define a process activity:

1. In Oracle Workflow Builder, select the item type that you want in the navigator tree. Then choose New Process from the Edit menu.
2. In the Activity property page, enter an internal name for the activity. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the activity.
4. Enter a description of the activity.
5. Choose an icon that identifies the activity.
6. Specify the result type that contains the possible results that the process can return.
7. Select the Runnable check box if this is a top-level workflow process that can be initiated. If the Runnable check box is selected, the process will appear in the Process

Name poplist in the Initiate Workflow web page, and you can successfully pass this process internal name in a call to the Workflow Engine API WF_Engine.CreateProcess. Clear the Runnable check box if this process activity can only be executed as a subprocess called by another higher-level process.

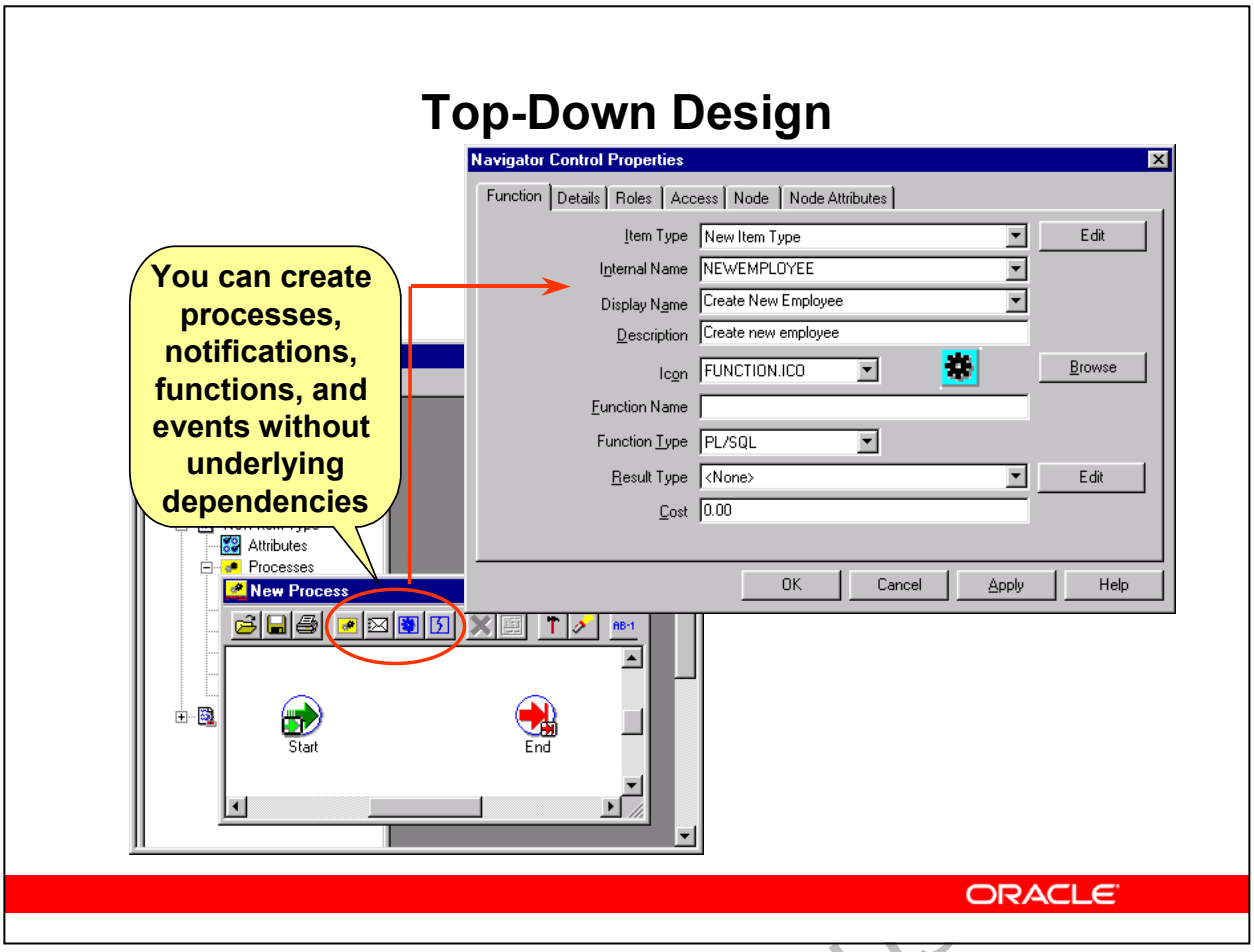
8. Choose Apply to save your changes.
9. Optionally select the Details tab to display and modify additional activity details.
10. Optionally select the Access tab to set the access levels allowed to modify this activity.



Diagramming a Process

Each process diagram must have one or more Start activities and one or more End activities.

Note: In the process window, you can display information about an activity by positioning your mouse over the activity. The Instance Label, Internal Name, Display Name, Comment, and Performer appear in a tool tip display.

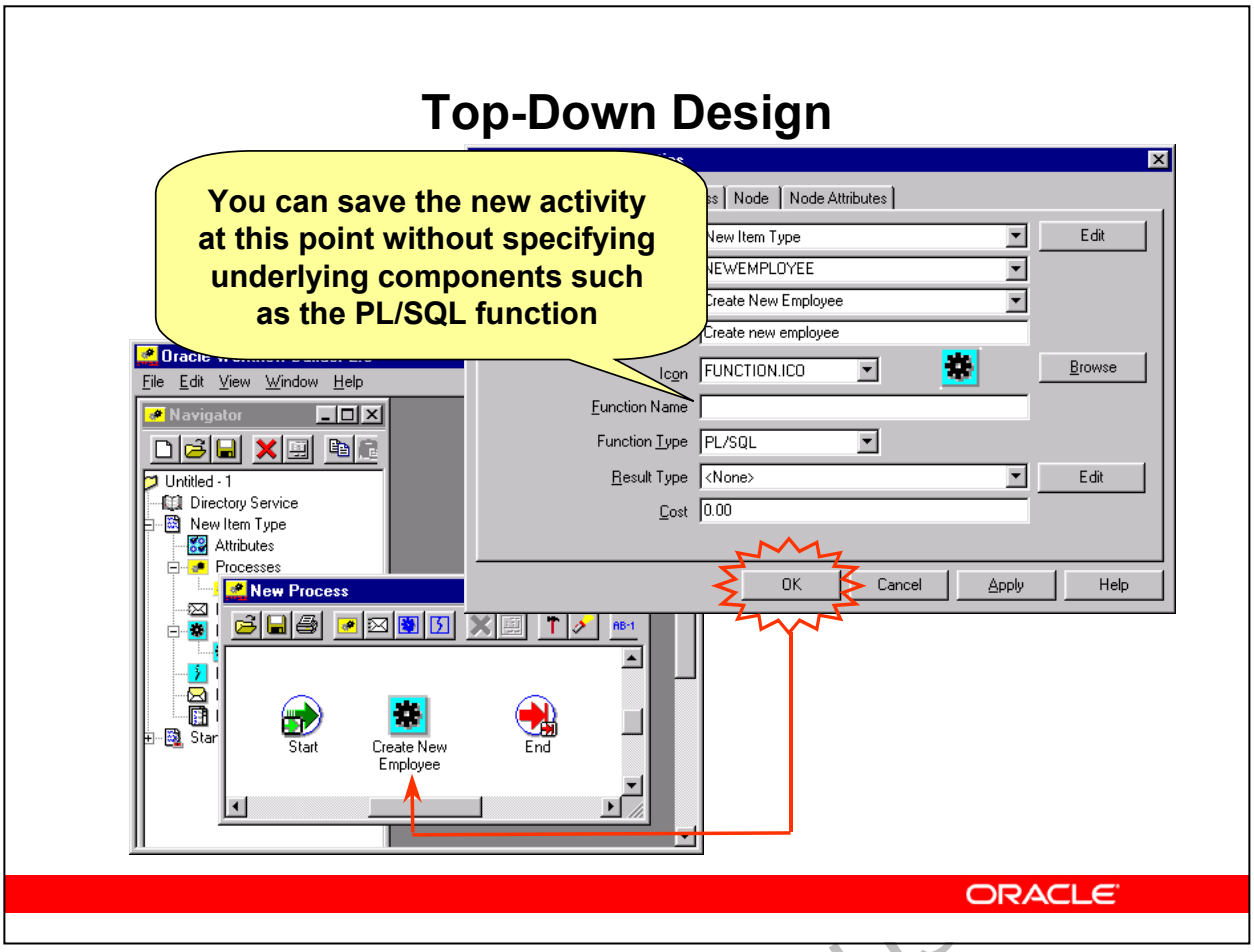


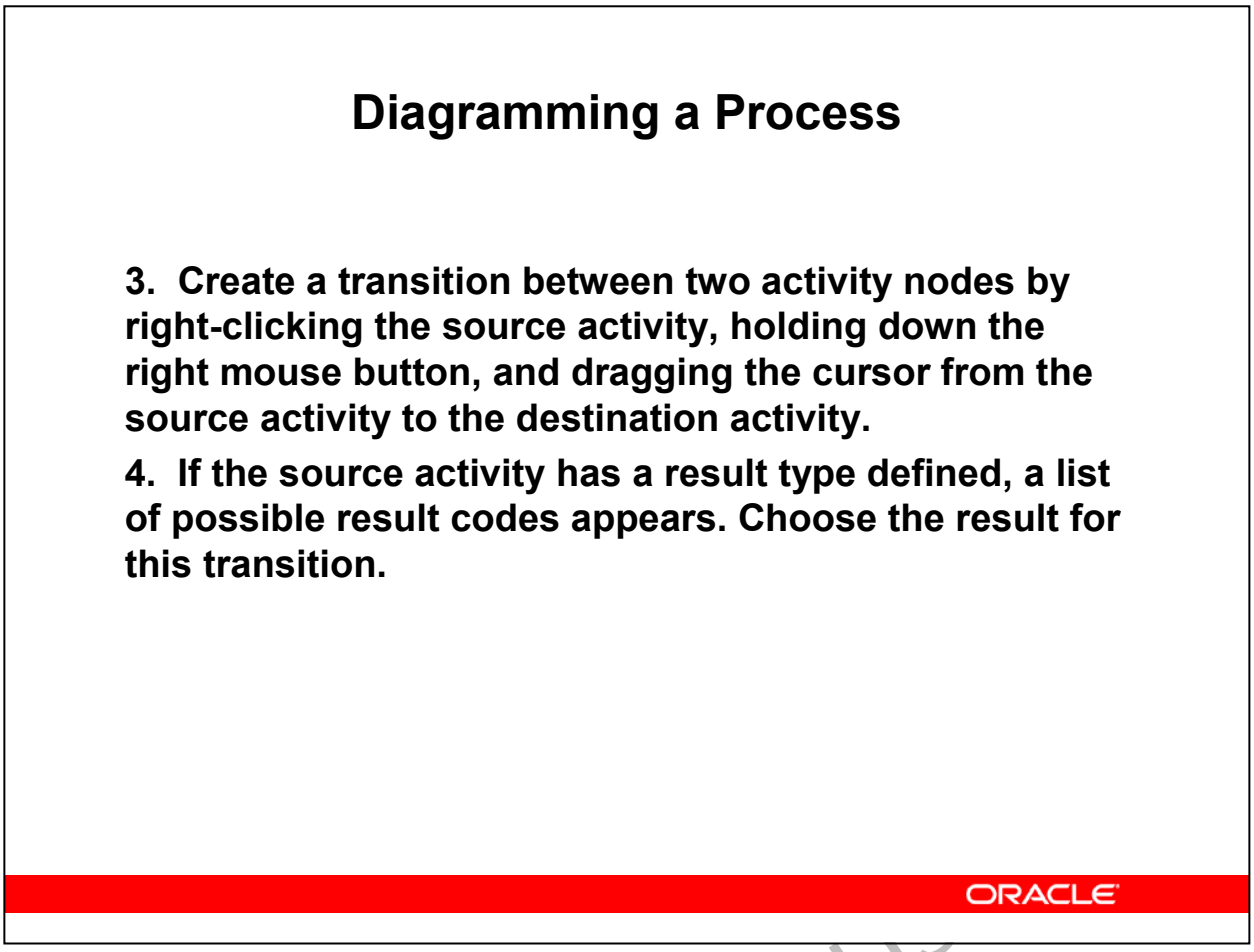
Top-Down Design

In top-down design, you draw the process diagram first and then define the underlying components for the activities.

When you create a new activity in the process window, the property pages displayed for activities from the process diagram window include all tabs for the base activity definition as well as for the usage of the individual activity node. You must enter at least an internal name and a display name for the activity in order to use the activity in the diagram. You can also enter the remaining details in the activity and activity node property pages immediately, or you can save the activity and complete these details at a later time.

When you save a partially completed top-down process, Oracle Workflow Builder displays an error window with a list of invalid or missing information. You can save the errored process definition and correct the errors later.





Diagramming a Process

If the source activity has no result type associated with it, by default no label appears on the transition. You can choose to show a label for such a transition by selecting the transition, holding down the right mouse button, and clearing the Hidden Label option. The label will display as <Default>.

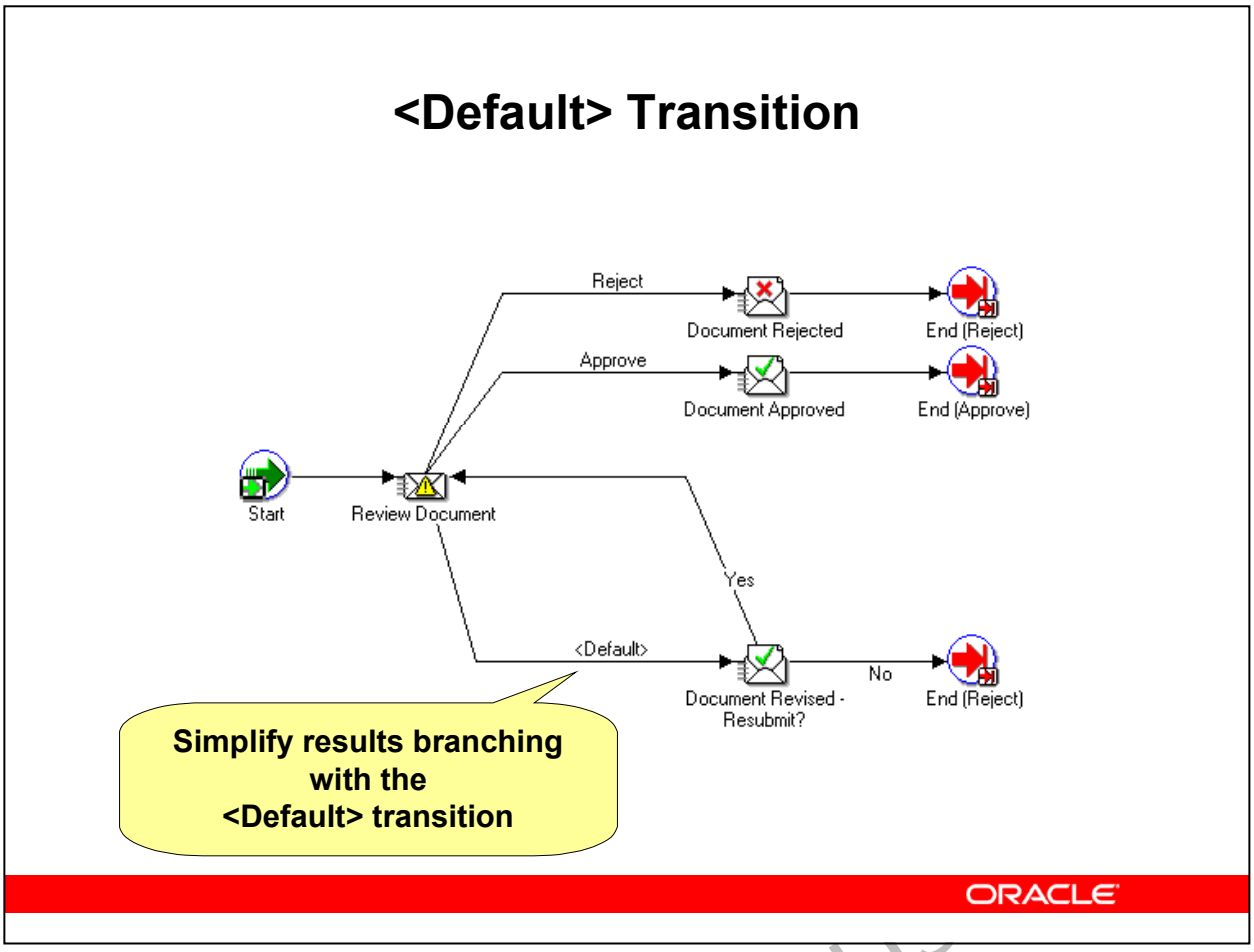
If the source activity has an associated result type, you can create transitions to follow:

- When a specific result is returned by the activity, labeled with the result display name
- When a result returned is not covered by other transitions, labeled <Default>
- Regardless of result returned, labeled <Any>
- When the activity times out, labeled <Timeout>

Note: The Timeout property for the node must be set to enable the <Timeout> option in the list of results for the transition.

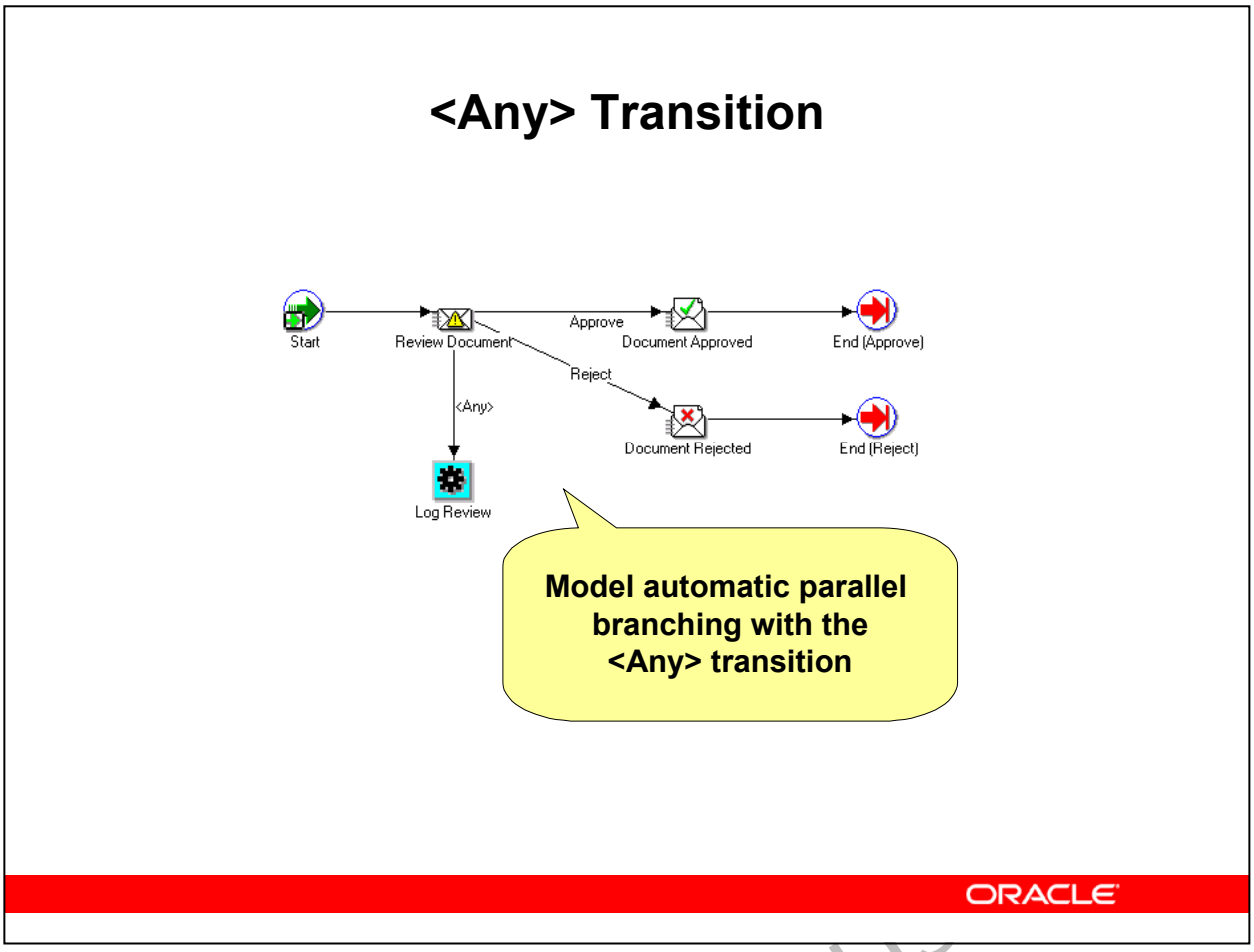
If the source activity is a voting activity, you can also create transitions to follow:

- When the vote results in a tie, labeled <Tie>
- When the vote doesn't return a result, labeled <No Match>



<Default> Transition

The Workflow Engine follows a <Default> transition if no other transition matching the completion result exists. This feature allows you to simplify your process diagram modeling and maintenance by only drawing transitions for unique results, allowing all other results to follow the default transition.



<Any> Transition

The Workflow Engine follows an <Any> transition regardless of what completion result the activity returns. This feature allows you to include a generic activity in the process that the Workflow Engine executes in parallel with any result-specific activity.

Editing a Transition

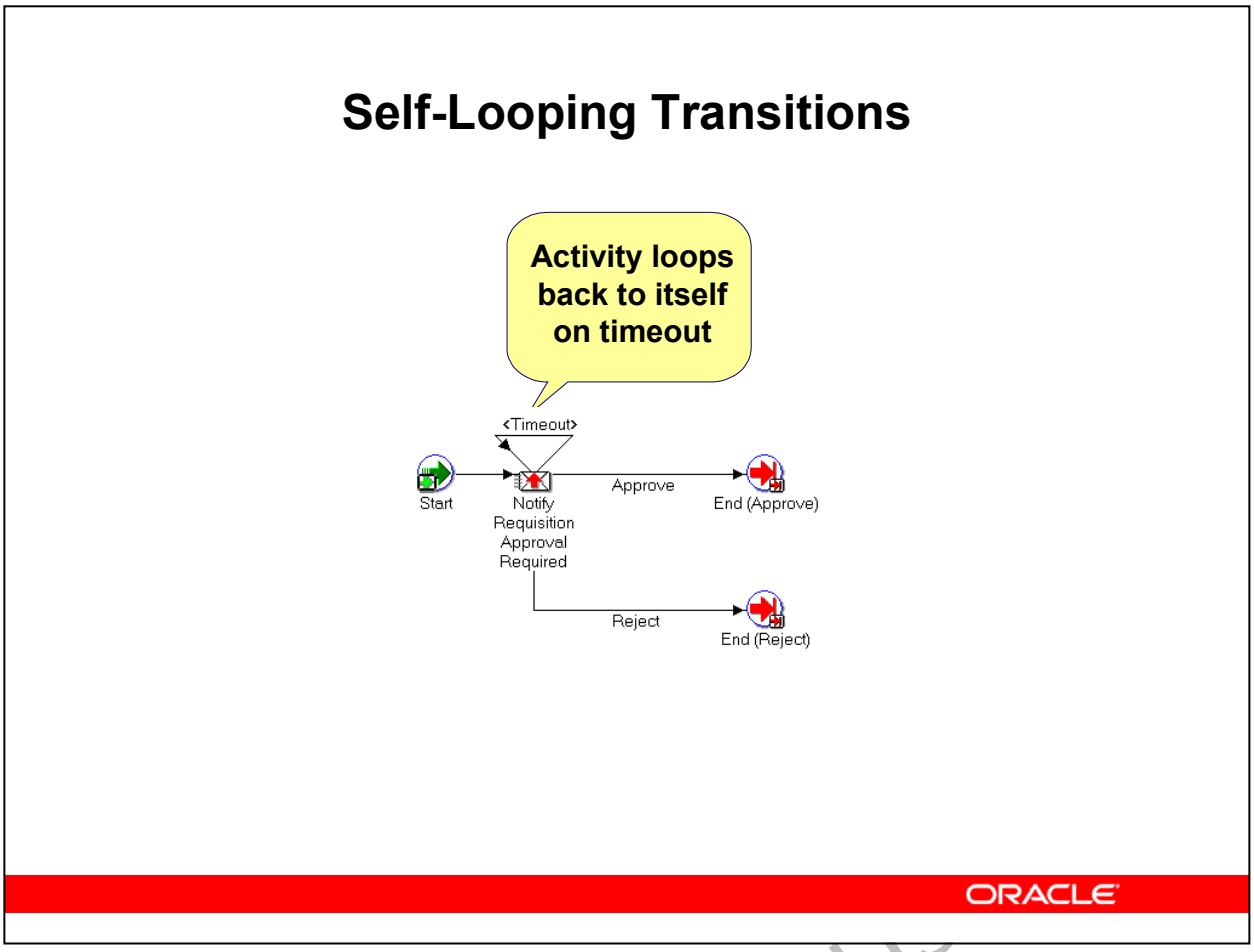
1. Select the transition you want to edit.
2. Right-click the transition and select the edit option you want:
   - Delete Selection: Deletes the selected transition
   - Locked: Toggles between locking and unlocking the transition from further edits
   - Hidden Label: Toggles between displaying and hiding the transition label
   - Straighten: Straightens the transition by removing extra vertex points causing the bend
   - Results: Lets you change the result label if one is already assigned

ORACLE

Editing a Transition

3. To reposition a transition label, select the label with the mouse and drag to the new position
4. To bend a transition, select the transition, hold down the left mouse button, and drag the cursor to a new position to create a vertex point.
5. To remove a vertex point, select the vertex point and drag over another to combine the two points.

ORACLE



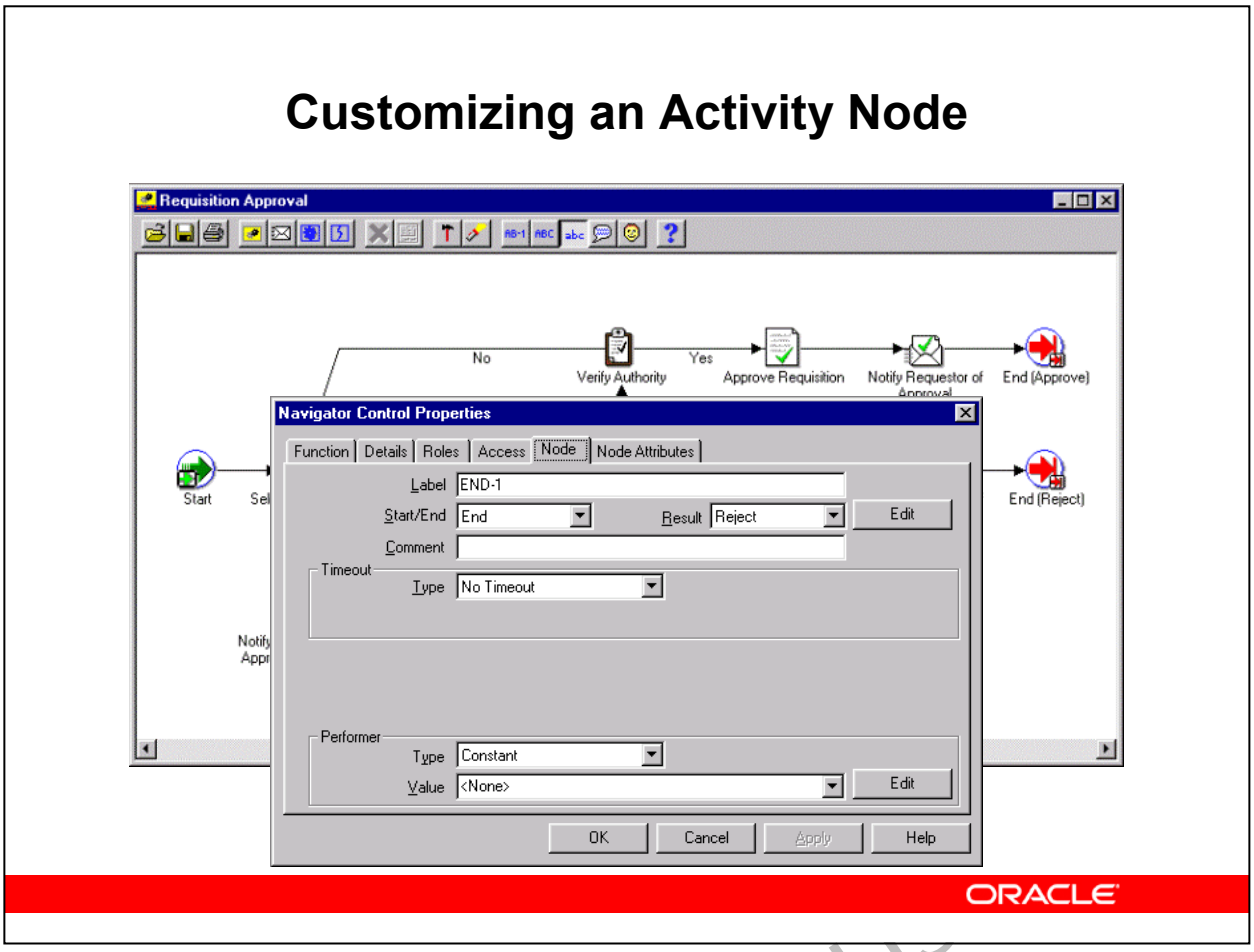
Self-Looping Transitions

To draw a self-looping transition:

1. Select the activity node.
2. Hold down the right mouse button.
3. Drag the cursor away from and back to the same activity node.
4. Release the right mouse button and select the transition label from the list of results.

Alternatively, you can also use the following steps:

1. Draw a transition to some other activity.
2. Add a vertex point to the transition.
3. Select the arrowhead of the transition with the left mouse button and drag it back to the originating activity.



Customizing an Activity Node

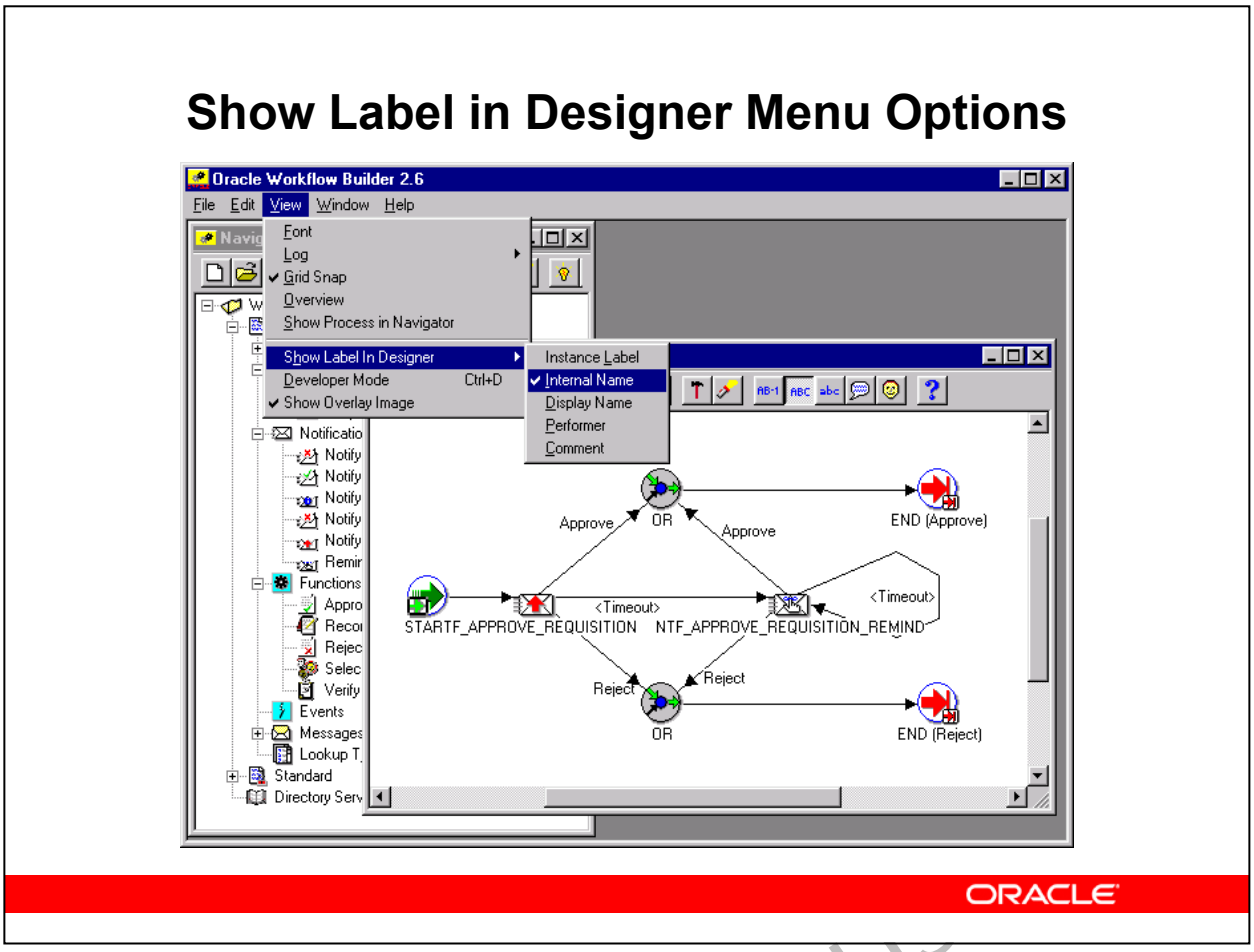
An activity can be reused throughout a process. An activity node represents a single instance of that activity, and its properties are unique to that instance.

To define node properties:

1. In Oracle Workflow Builder, double-click the activity node in the process window to display the property pages for the node. Select the Node tab.
2. Specify a label for the node. By default, the label name is the activity name. If you use an activity more than once in a process, *-N* is appended to the label name to identify each node uniquely, where *N* represents the *N*th instance of the activity in use.
3. Specify whether the activity is a Start, End, or Normal activity in the process.

   For an End activity, if the process itself has a result type associated with it, you should specify what the result of the process is when this End activity completes. Choose from the possible results listed in the result type of the process activity.
4. Optionally enter a comment about this node.
5. For a notification activity, a process activity, or a Receive event activity, use the Timeout region to specify whether the activity must be completed by some specified time.

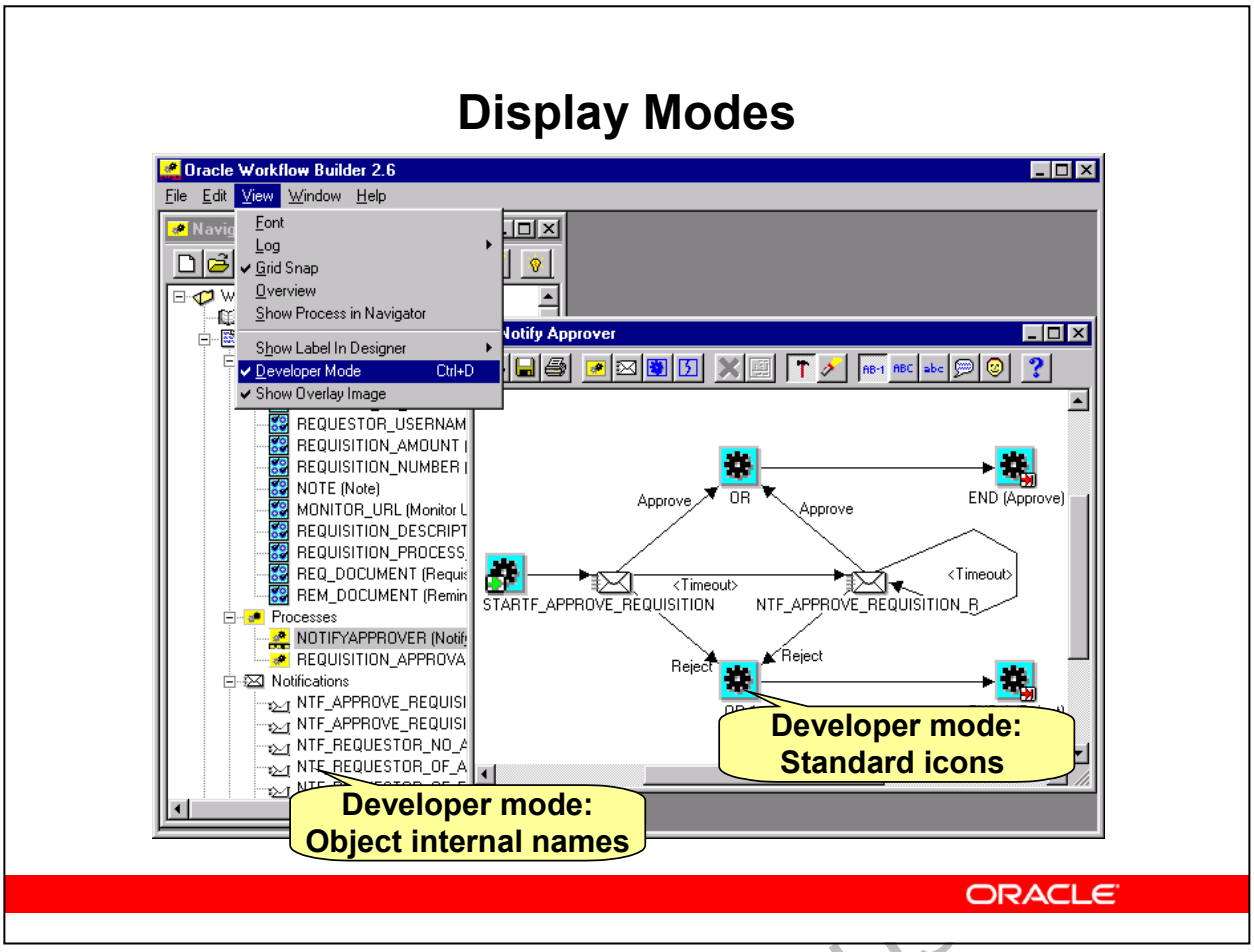
6. For a notification activity or a Send event activity, optionally specify the message priority.
7. For a notification activity , specify a performer. A performer is a role that can consist of one or more users.
   - You can specify a static role. Specify Constant for the Performer Type, and select a role name. Note that you must first load roles from your database.
   - You can alternatively specify a dynamic role. Select Item Attribute for the Performer Type, and select an item type attribute that returns a role name at runtime.
8. Choose Apply to save your changes.



Show Label in Designer Menu Options

You can choose Show Label in Designer from the View menu to select the type of label you want to display for the activity nodes in the process window.

- Instance Label: Uses the node label for each activity node
- Internal Name: Uses the internal name of the activity as the label for each activity node
- Display name: Uses the display name of the activity as the label for each activity node
- Performer: Uses the activity's performer as the label for each activity node
- Comment: Uses the activity's comment as the label for each activity node



Display Modes

You can toggle between the following two display modes in Oracle Workflow Builder:

- Presentation mode: Oracle Workflow Builder shows the display names and activity icons that you specify in the property page for each activity.
- Developer mode: Oracle Workflow Builder assists you in developing large, complex processes by:
  - Providing visual distinctions between top-level processes and subprocesses in the navigator tree
  - Setting the icons for each activity and object to the default icon of its object type or subtype in both the navigator tree and the process diagram window
  - Identifying each object in the navigator tree by its internal name (developer name)
  - Identifying each activity in the process diagram window by its activity label

To change modes, choose Developer Mode from the View menu.

Verifying a Process Definition

- When you save your work, Oracle Workflow Builder:
  - Automatically validates the process definition for any invalid or missing information and
  - Displays the results of the validation in a Workflow Error verification window
- You can also verify the definitions in the current data store at any time during a Builder session.
  - Select an object such as a data store, item type, or process in the navigator tree
  - Choose Verify from the File menu

ORACLE

Verifying a Process Definition

You should always validate any new process definition you create, because validation helps you identify any potential problems with the definition that might prevent it from executing successfully.

Note: The Verify option is enabled in the File menu only when the navigator tree is the current window.

Validation Performed by the Verify Command

- The process contains at least one Start activity and one End activity.
- A process does not contain itself as a process activity.
- The same subprocess is not being used twice in a process.
- All possible activity results are modeled as outgoing transitions.
- All activity nodes marked as END nodes do not have any outgoing transitions.

ORACLE

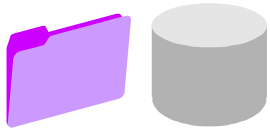
Validation Performed by the Verify Command

- **Each notification activity's result type matches the lookup type defined for the message's RESULT message attribute.**
- **All message attributes referenced in a message body for token substitution exist in the message definition.**
- **If a process references objects from another item type, all required item attributes associated with the other item type exist.**

ORACLE

Saving Process Definitions

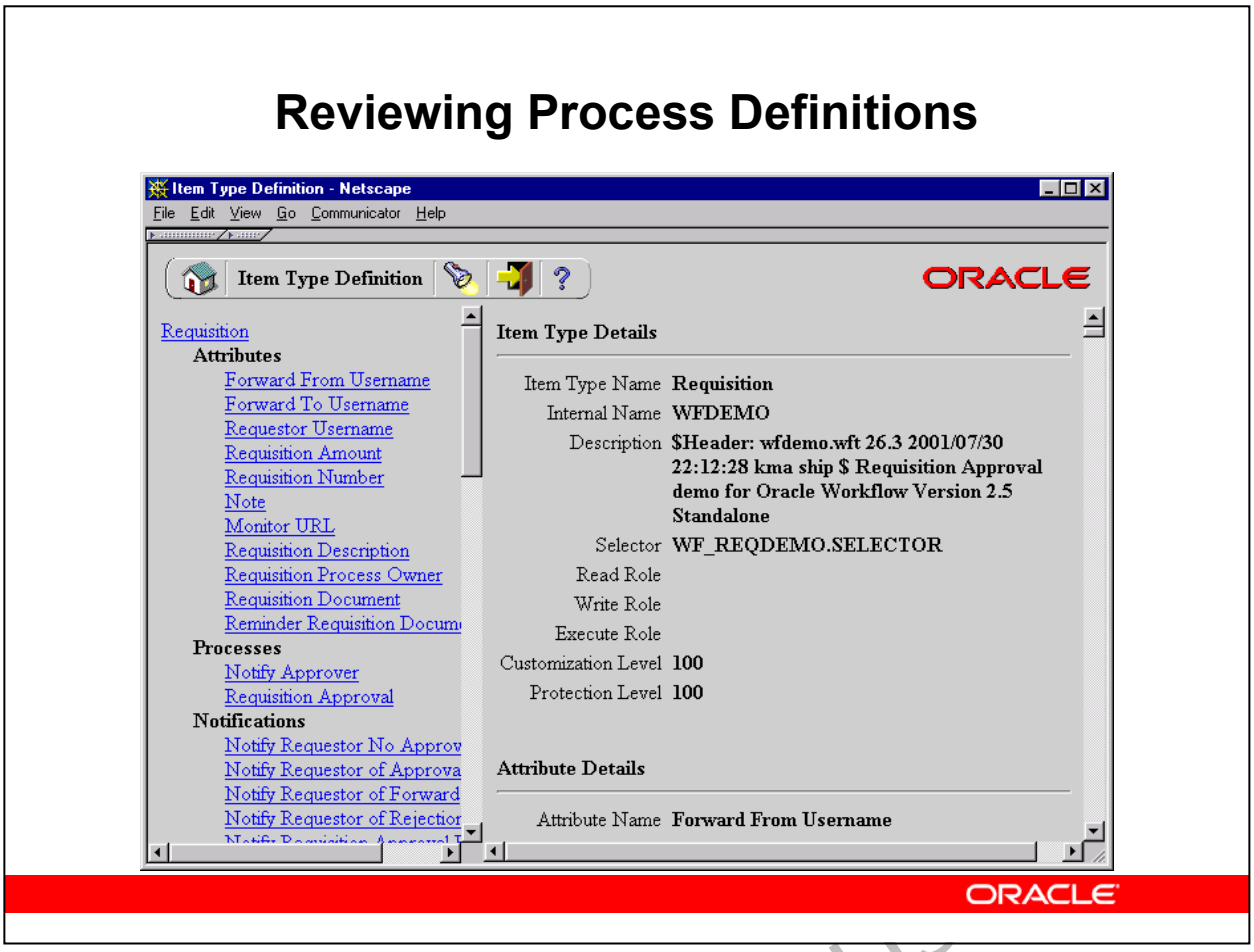
- Always save the process definition to a flat file for source control.
- Before saving the process definition to the production database, save to a development database for testing.
- Oracle Workflow Builder uses a utility program called the Workflow Definitions Loader to move workflow definitions between databases and flat files.
- The Workflow Definitions Loader can also be used to move definitions between development and production databases, or to apply upgrades to existing definitions.



ORACLE

Saving Process Definitions

If a nonunique display name is encountered when a process definition is being uploaded or upgraded in a database, the Workflow Definitions Loader automatically resolves the display name conflict by adding a "@" character to the beginning of conflicting display names.



Reviewing Process Definitions

The Item Type Definition web page provides distributed access to workflow definitions stored in your Oracle Workflow database. This web page shows a detailed view of attributes, processes, notifications, functions, events, messages, and lookup types associated with a given item type. The information is displayed in two frames.

- The left frame lists all the objects in the item type in an expandable hierarchy tree, similar to the navigator tree in Oracle Workflow Builder.
- The right frame displays the details of the object you select in the hierarchy tree.

You can select either frame at any time and use the Print command in your web browser to print all the information in the selected frame.

To view an item type definition:

1. Use a web browser to connect to the Item Type web page.

   For standalone Oracle Workflow, connect to the following URL:

   <*webagent*>/ wf_item_definition.find_item_type

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.You can also access the Item Type web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Item Type Definition option from a Workflow responsibility.

2. The Item Type page appears.

   Select the item type you want to display in the Item Type field.

3. Specify the effective date and time of the item type definition you want to display, using the date format specified in your User Preferences web page.

4. Choose the Find button to open the Item Type Definition page and review the specified item type definition.

Practice - Creating a Workflow Process

Overview

In this practice, you will create a workflow process in the Oracle Workflow Builder using the Quick Start Wizard. The process you create will be a prototype of the vacation proposal process you sketched in the Planning a Workflow Process practice.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Note: In order to use the sample solution scripts provided for these practices, you must enter the internal names for all objects you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names you define.

Tasks

1. Create a workflow process in the Oracle Workflow Builder using the Quick Start Wizard.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Quick Start Wizard.

   3. Define the following properties for the new item type:
      - Internal Name: WFVAC*XX*
      - Display Name: *XX* Vacation Proposal
      - Persistence Type: Temporary
      - Number of Days: 5

      Define the following properties for the process:
      - Internal Name: WFVAC*XX*_PROCESS
      - Display Name: *XX* Vacation Proposal Process

      Choose OK.

   4. Create notification activity nodes for Vacation Proposal, Vacation Rejected FYI, and Vacation Approved FYI notification activities.
      - For each notification, click the New Notification button in the process diagram window toolbar, and then click an open spot in the diagram between the Start and End nodes
      - In the node properties window for each notification, enter an internal name and display name and choose an icon. For the Vacation Proposal notification, select the

Approval lookup type that is provided in the Standard item type as the Result Type for the activity. Then choose OK.

5. Create a new End node by dragging the End function from the Standard item type in the navigator tree into the process diagram window. Double-click the new End node and choose the Node tab. In the Start/End field, select End.

6. In the process diagram, draw transitions between the nodes in the process to indicate the process flow. To draw a transition, click the source activity, hold down the right mouse button, drag the cursor to the destination activity, and release the right mouse button. For transitions from the Vacation Proposal node, choose the appropriate result from the results menu that appears.

7. Select the *XX* Vacation Proposal Process in the navigator tree, right-click the process activity, and choose Properties to open the property pages for the process.

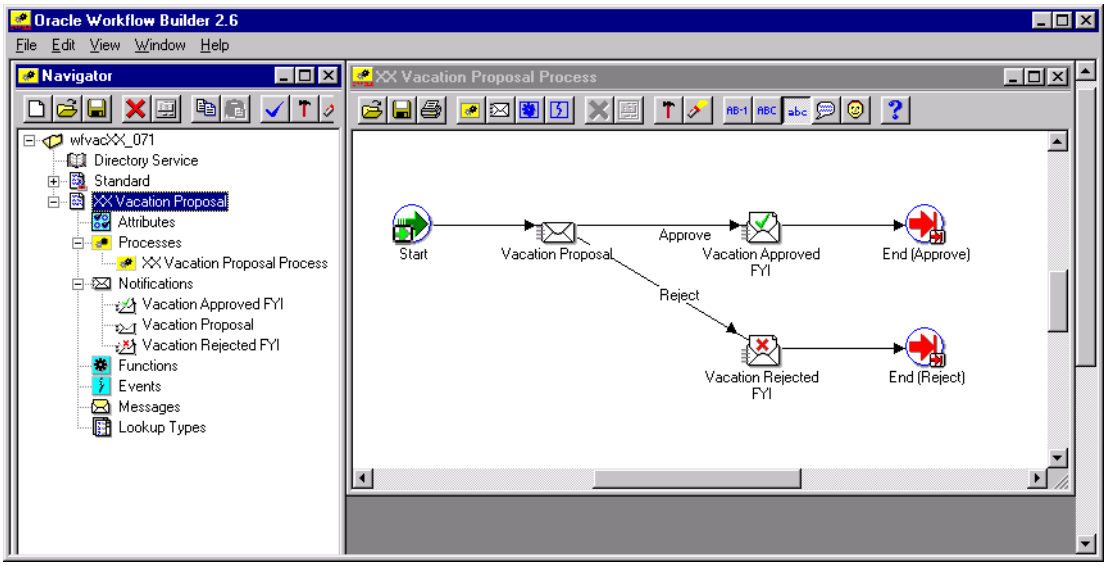
8. Choose the Activity tab and select Approval as the result type for the process. Then choose OK.

9. Double-click the End node that follows the Vacation Rejected FYI node and choose the Node tab. Select Reject in the Result field.

10. Double-click the End node that follows the Vacation Approved FYI node and choose the Node tab. Select Approve in the Result field.

11. In the Navigator window, click the Verify button to verify your workflow. Because you have not yet defined the underlying components for your process, the Workflow Error window displays warnings for validation checks that the process does not yet satisfy. Review the error messages to learn about the errors. In later practices, you will add additional functionality to your workflow to eliminate these errors. Click Cancel to exit the Workflow Error window.

12. From the File menu, choose Save As and save your new data store and item type to a workflow definition file named wfvac*XX*.wft. The Oracle Workflow Builder will verify your workflow again. Click Save in the Workflow Error window to finish saving the workflow definition.



Summary

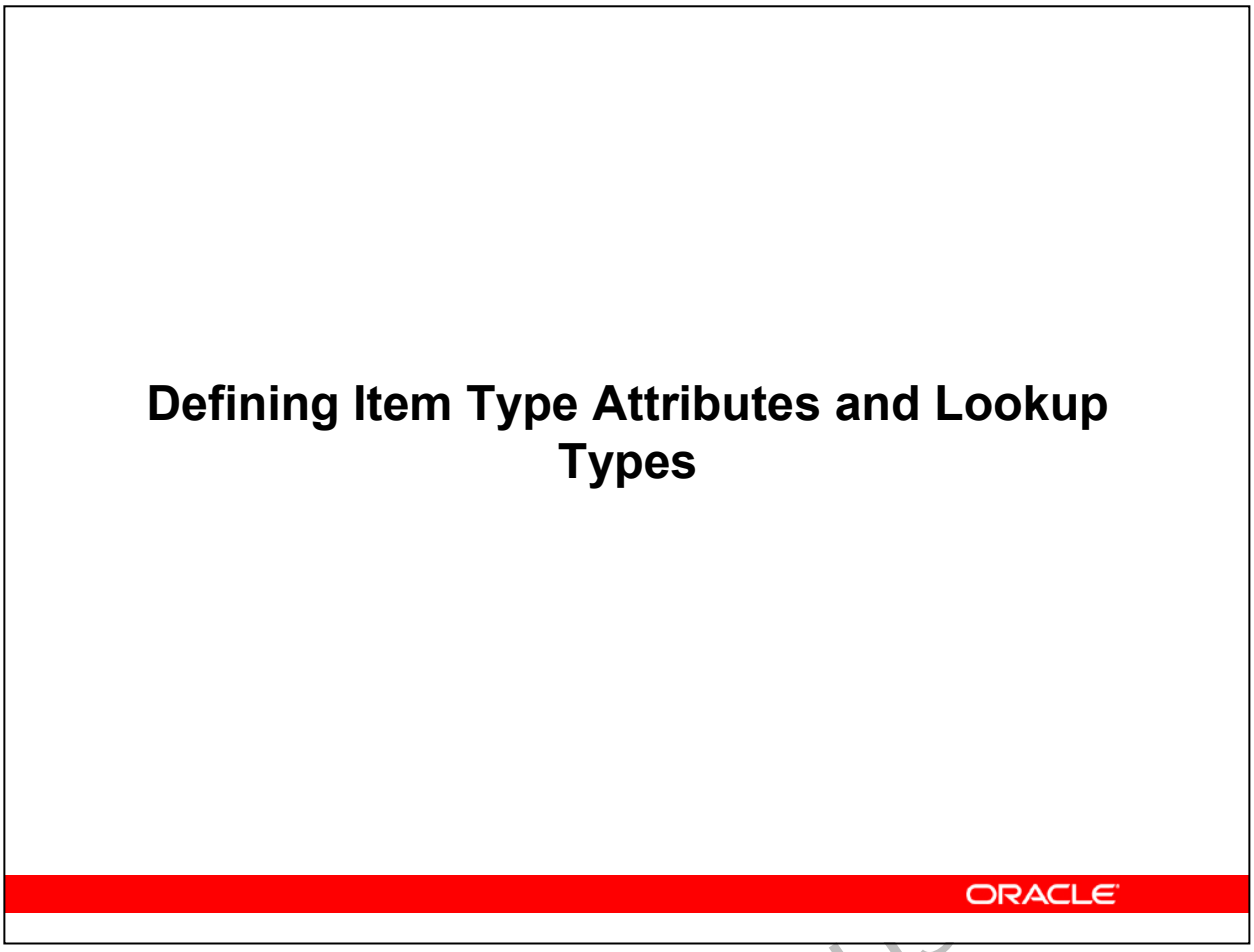
In this lesson, you should have learned how to:

- **Diagram a workflow process.**
- **Define an item type.**
- **Define a process activity.**
- **Review the process definition for an item type.**

ORACLE

Defining Item Type Attributes and Lookup Types

Chapter 8



Objectives

After completing this lesson, you should be able to do the following:

- **Define item type attributes.**
- **Define lookup types and lookup codes.**

ORACLE

Defining Item Type Attributes

- An item type attribute is a property associated with a given item type.
- It acts as a global variable that can be referenced or updated by any activity within a process.
- You must define an item type before you can define the item attributes associated with it.
  - Use the Quick Start Wizard to quickly create a new process framework, including an item type
  - Define item types manually in the navigator tree

ORACLE

Defining Item Type Attributes

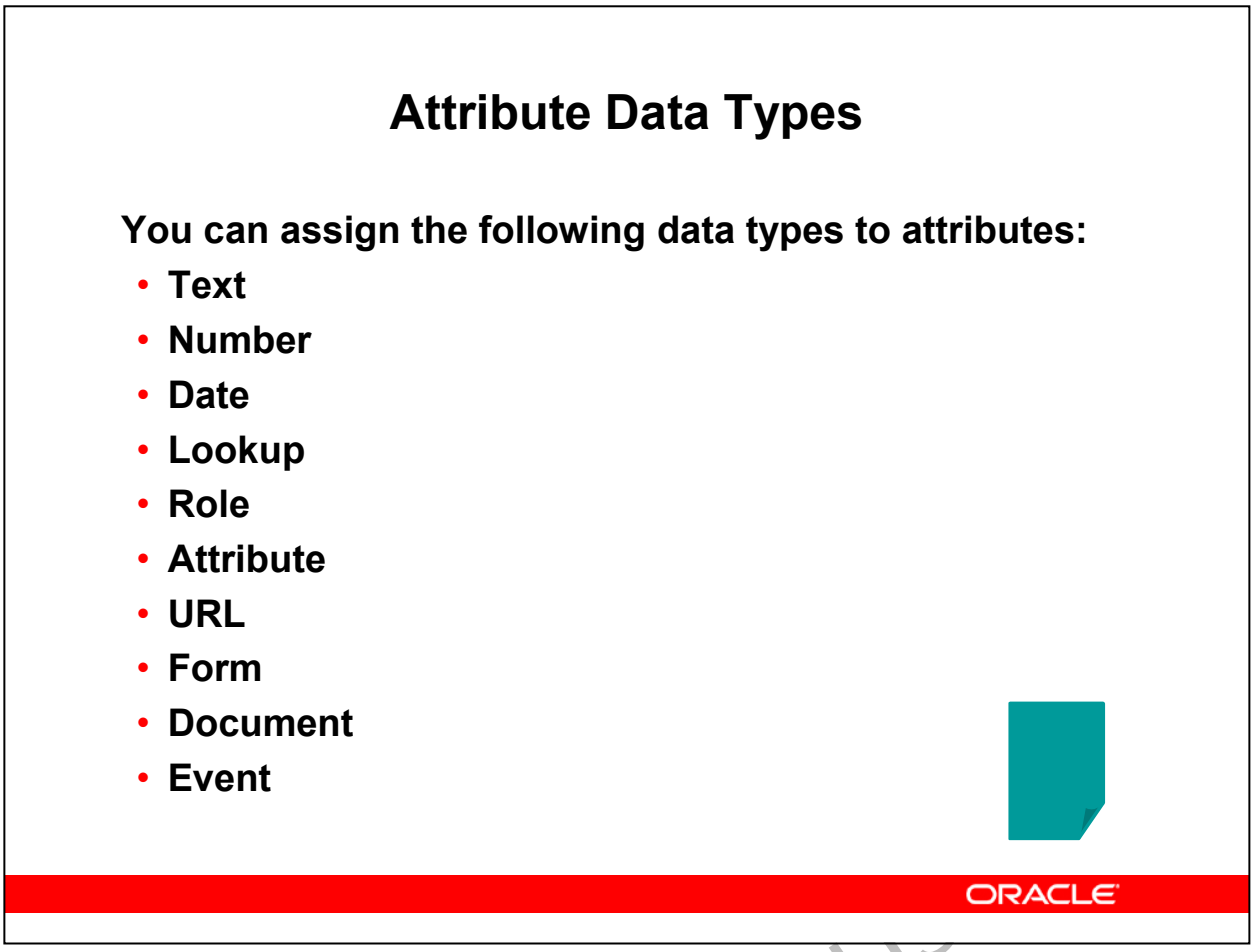
What properties should be defined as item type attributes?

- **Information required by notification messages, such as the display format of a currency value**
- **Information required by function activities, such as values that link back to applications data like a person_ID**
- **Information maintained by a workflow activity, such as data for activities that identify who to forward information to and from in a loop construct**
- **Business event messages required by Send or Receive event activities, or event details required by Raise event activities**

ORACLE

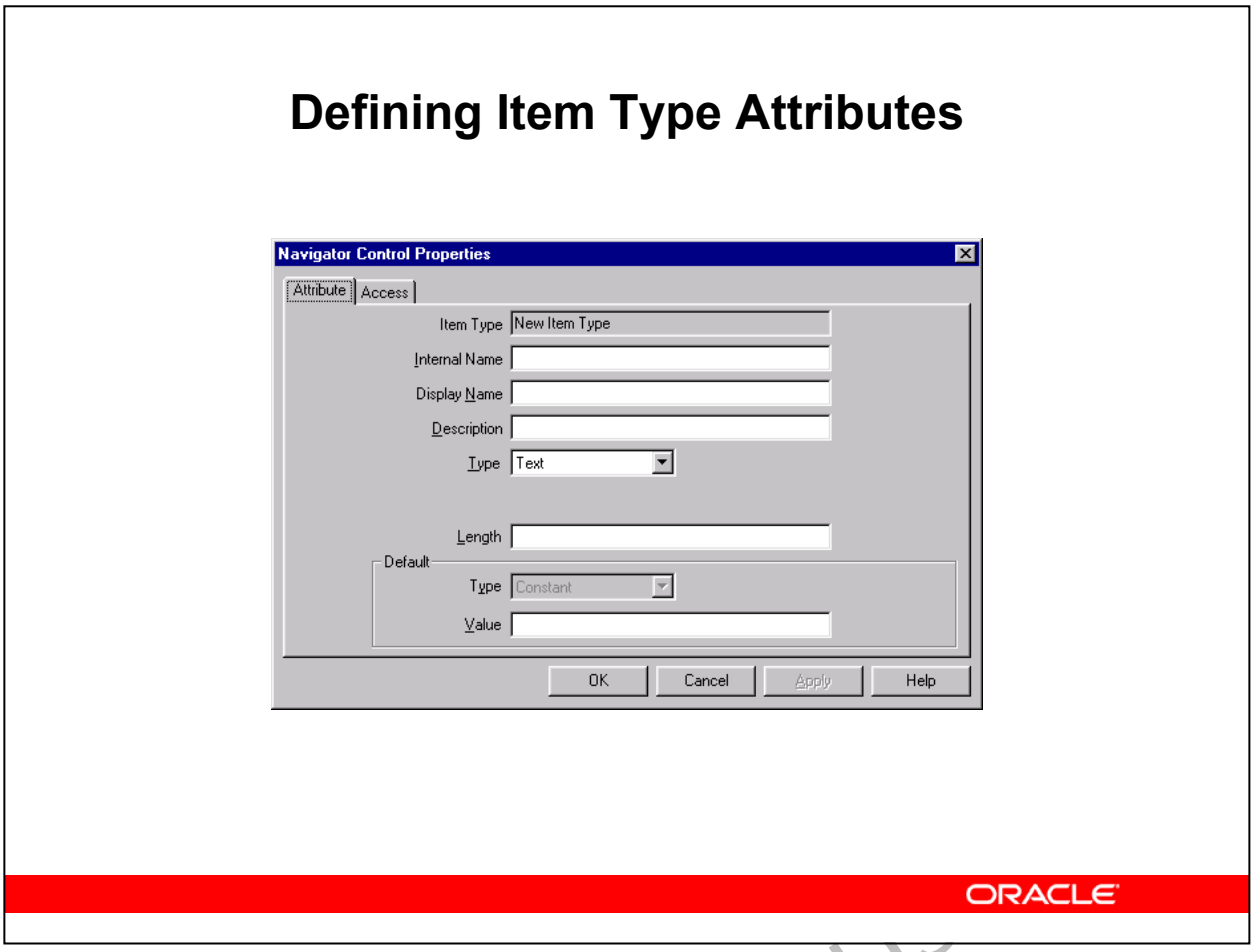
Defining Item Type Attributes

Note: Do not use item type attributes as a denormalized data store. Always refer back to the base application to retrieve up-to-date values. There is a trade-off between the convenience of storing global process data in item type attributes and the overhead incurred by loading item type attributes when a process instance is created. Minimize the number of item type attributes to improve the performance of your workflow process.



Attribute Data Types

- Text: A string of text of specified character length
- Number: A number with an optional format mask
- Date: A date with an optional format mask
- Lookup: A lookup code value from a specified lookup type
- Role: A role name from the list of roles defined in the directory service
- Attribute: The internal name of another existing item type attribute to which you want to maintain references in a process
- URL: A Universal Resource Locator (URL) to a network location
- Form: The internal function name and any optional form parameters of an Oracle E-Business Suite form function (available only for Oracle Workflow embedded in Oracle E-Business Suite
- Document: An attached document, as specified by the document type (PL/SQL and PL/SQL CLOB documents are files representing data from the database as a character string or a character large object (CLOB), respectively)
- Event: A Business Event System event message in the standard WF_EVENT_T structure

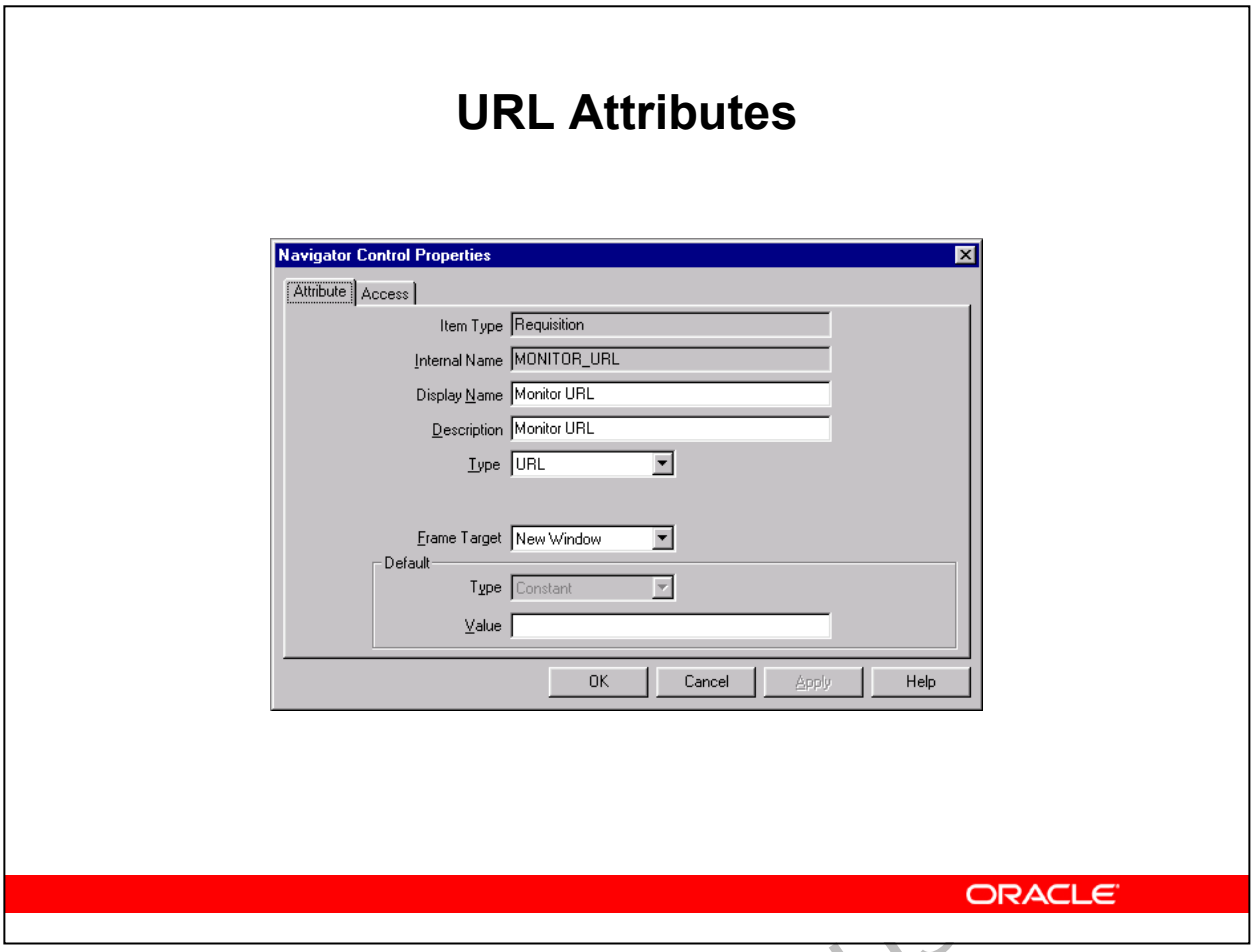


Defining Item Type Attributes

Item type attributes are defined in the navigator tree beneath a given item type.

To define an item type attribute:

1. In the Oracle Workflow Builder, select the item type you want in the navigator tree. Then choose New Attribute from the Edit menu.
2. In the Attribute property page, enter an internal name for the attribute. The internal name must be all uppercase without any leading or trailing spaces.
3. Enter a display name for the attribute.
4. Enter a description of the attribute.
5. In the Type field, select the data type for the attribute.
6. Depending on the data type, enter format information and a default value, if applicable.
7. Choose Apply to save your changes.



URL Attributes

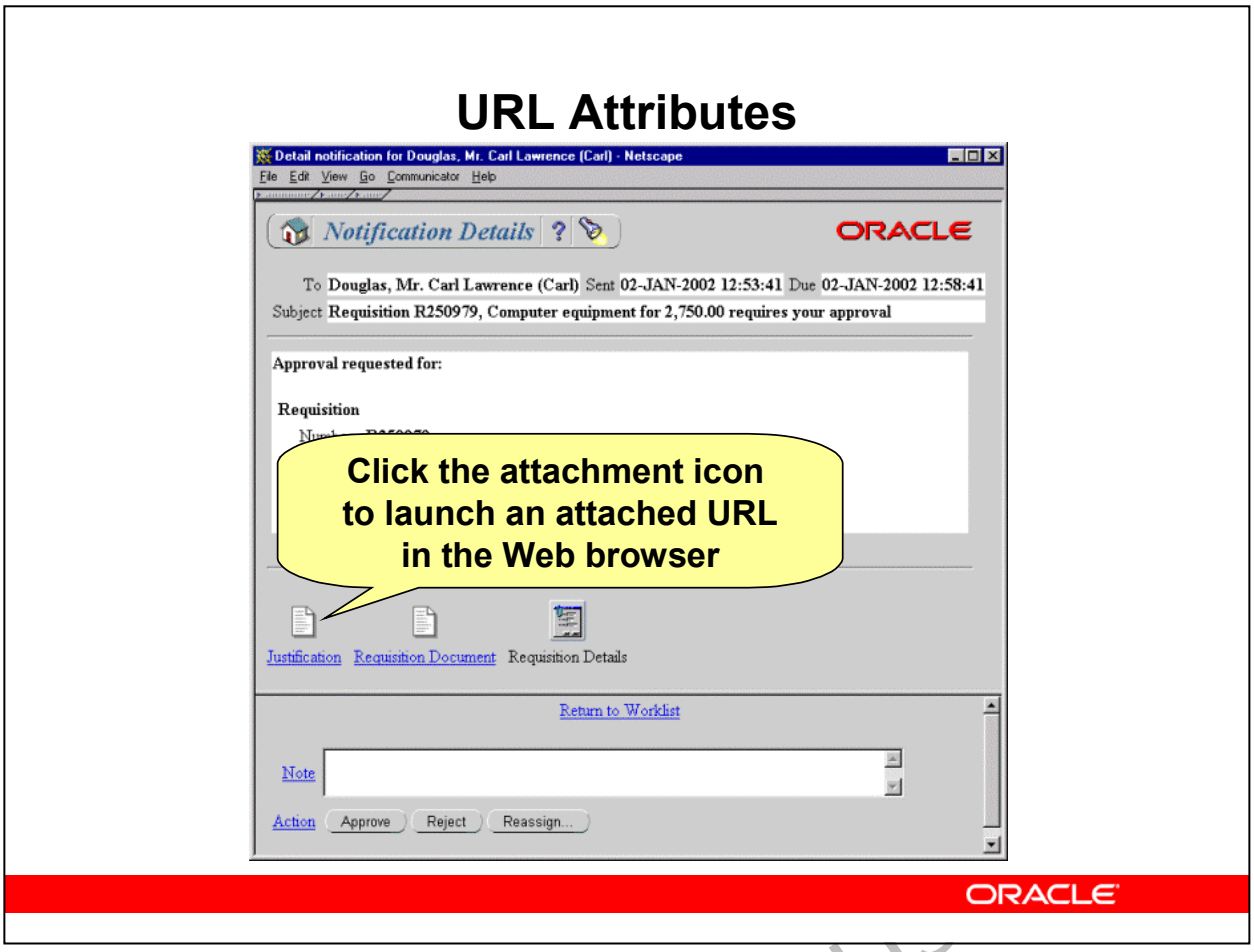
When you create an item attribute of type URL, specify a frame target for the attribute. If you reference this item attribute as the default value of a message attribute in a message, the URL frame opens according to what you specified as the frame target. The options are:

- New Window: The URL loads in a new, unnamed browser window.
- Same Frame: The URL loads in the same frame as the element that references the URL attribute.
- Parent Frameset: The URL loads into the immediate FRAMESET parent of the current frame. This value is equivalent to Same Frame, if the current frame has no parent.
- Full Window: The URL loads into the full, original window, thus canceling all other frames. This value is equivalent to Same Frame, if the current frame has no parent.

URL Attribute Values

The value of a URL attribute can be a text string or can be token substituted. A URL value is specified in the format:

```
http://<location>
```

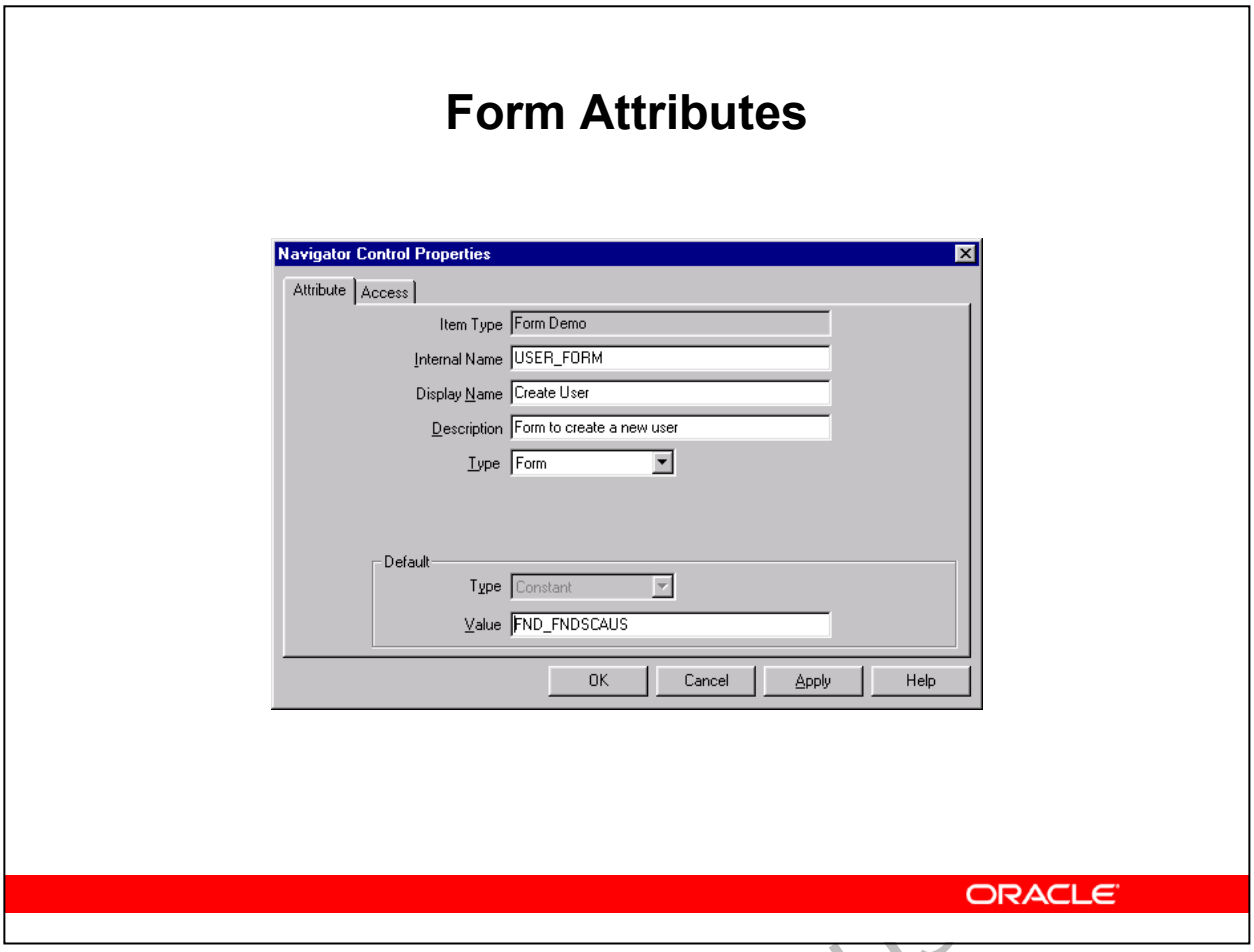


URL Attributes

The Notification Details web page supports message attributes of type URL. These attributes appear in a notification message body as a hypertext link or below the message as an attachment icon. When you open your notification, you can click the link or attachment icon to display the URL according to the frame target specified for the attribute.

In a plain text e-mail notification, the URL location value for an attached URL attribute is included as plain text.

In an HTML-formatted e-mail notification, a URL attribute appears as a hypertext link in the message body. For attached URL attributes, an attachment called Notification References is appended to the message. This attachment includes a link to each attached URL attribute for the message. You can navigate to a URL by choosing its link.



Form Attributes

Form Attribute Values

The value for an attribute of type form must be the internal function name of an Oracle E-Business Suite form function, together with any optional form parameters that should be prequeried with a value when the form is opened.

The default value for the form attribute must be entered using the following format:

function_name:parameter1=value1 parameter2=value2 ...parameterN=valueN

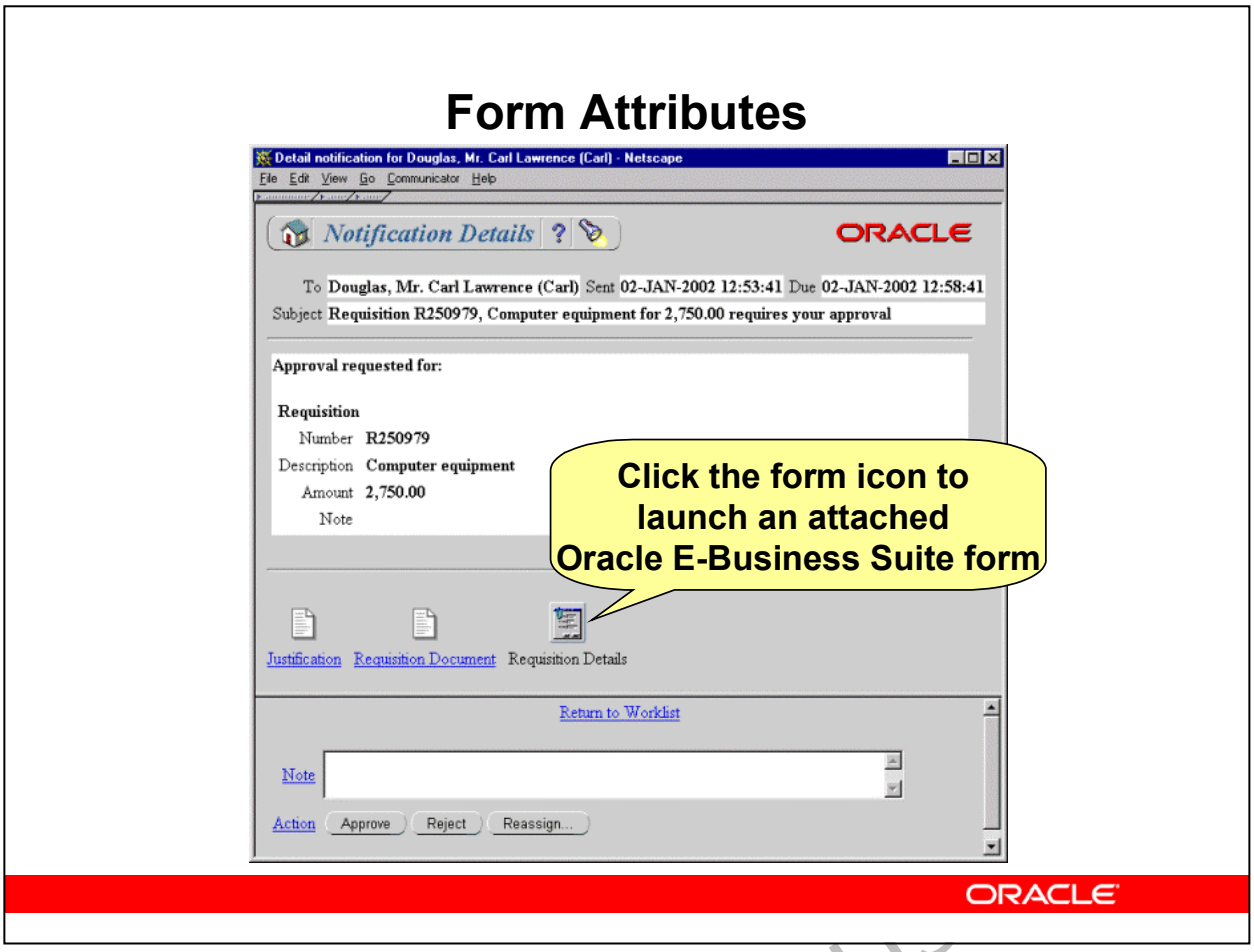
The value of a form parameter can be a text string enclosed in quotes (" ") or can be token substituted with another predefined item type attribute in either of the following ways:

- *parameterN = "&item_type_attribute"*
- *parameterN = "Value &item_type_attribute"*, where *&item_type_attribute* represents the rest of the value

To find the internal function name for a form function, navigate to the form, choose About Oracle Applications from the Help menu, and locate the internal form name in the Current Form section of the information displayed. Then log into the Application Developer responsibility and use the Forms window to query for the user form name based on the internal

form name and the Form Functions windows to query for the form function name based on the user form name.

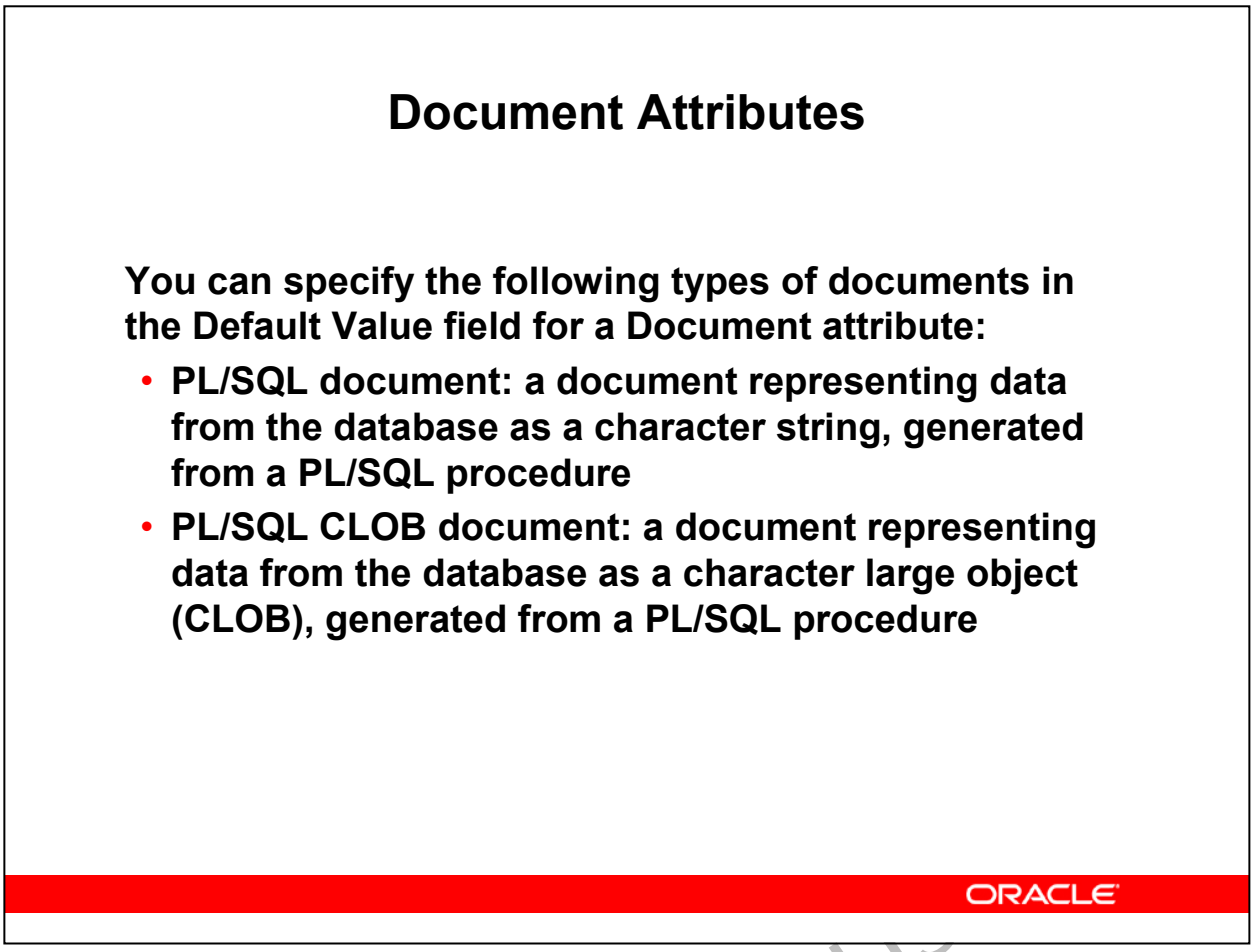
Note: This attribute type is not relevant for the standalone version of Oracle Workflow.



Form Attributes

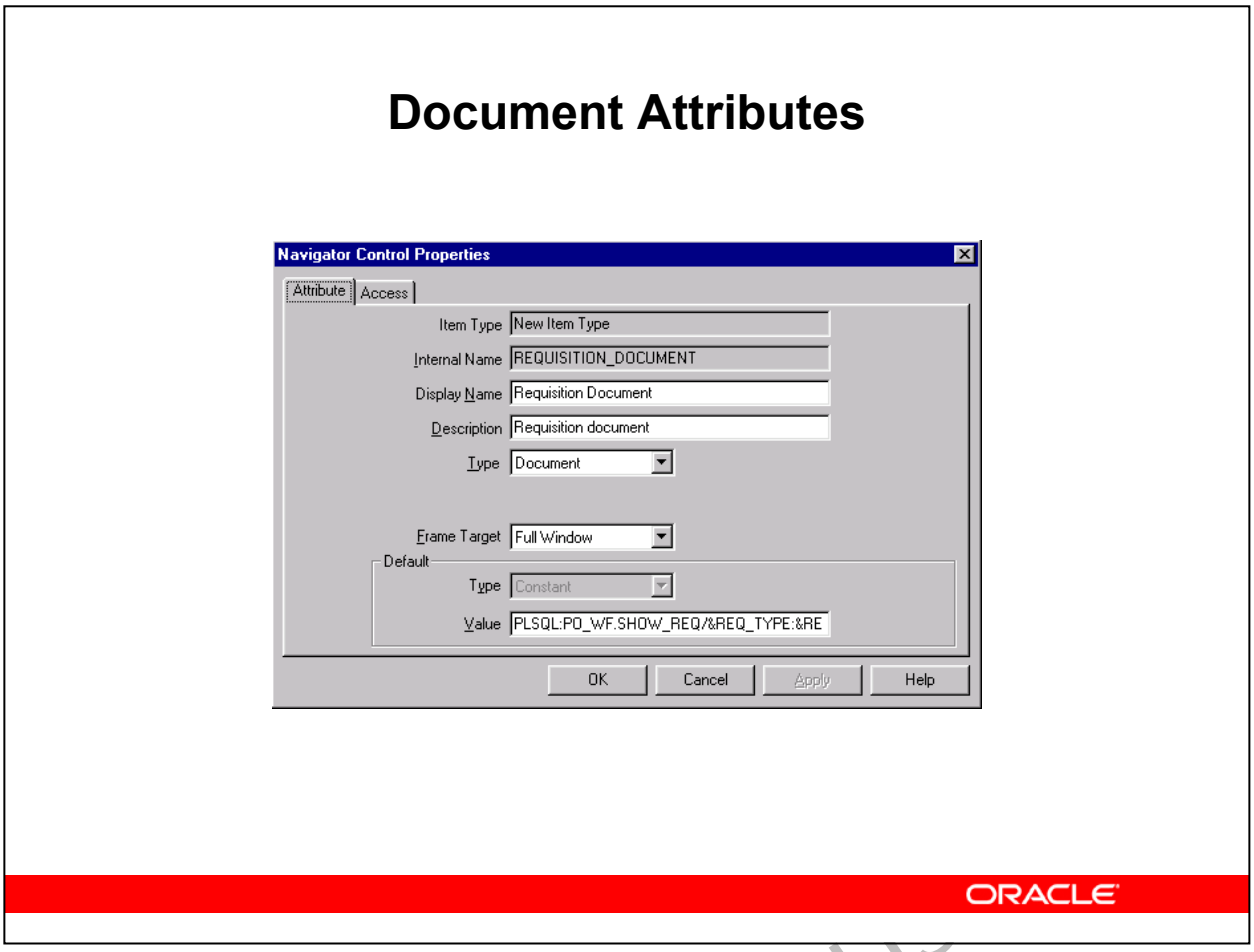
The Notification Details web page supports message attributes of type form. These attributes appear in a notification message as a form icon. When you open your notification, you can click the attached form icon to drill down to the referenced form.

Note: This attribute type is not relevant for the standalone version of Oracle Workflow. Also, form attributes cannot be attached to e-mail notifications. To view form attachments, you must view the notification in the Notification Details web page.



Document Type Attributes

PL/SQL and PL/SQL CLOB documents are ideal for use when document content varies. For example, these documents can be used to represent purchase orders with variable numbers of lines, or sales orders with variable lines, discounts, and shipping addresses.



Document Type Attributes

Document Attribute Values

The value of a document attribute can be a text string or can be token substituted.

- A PL/SQL document value is specified in the format:

  `PLSQL:<procedure>/<document_identifier>`

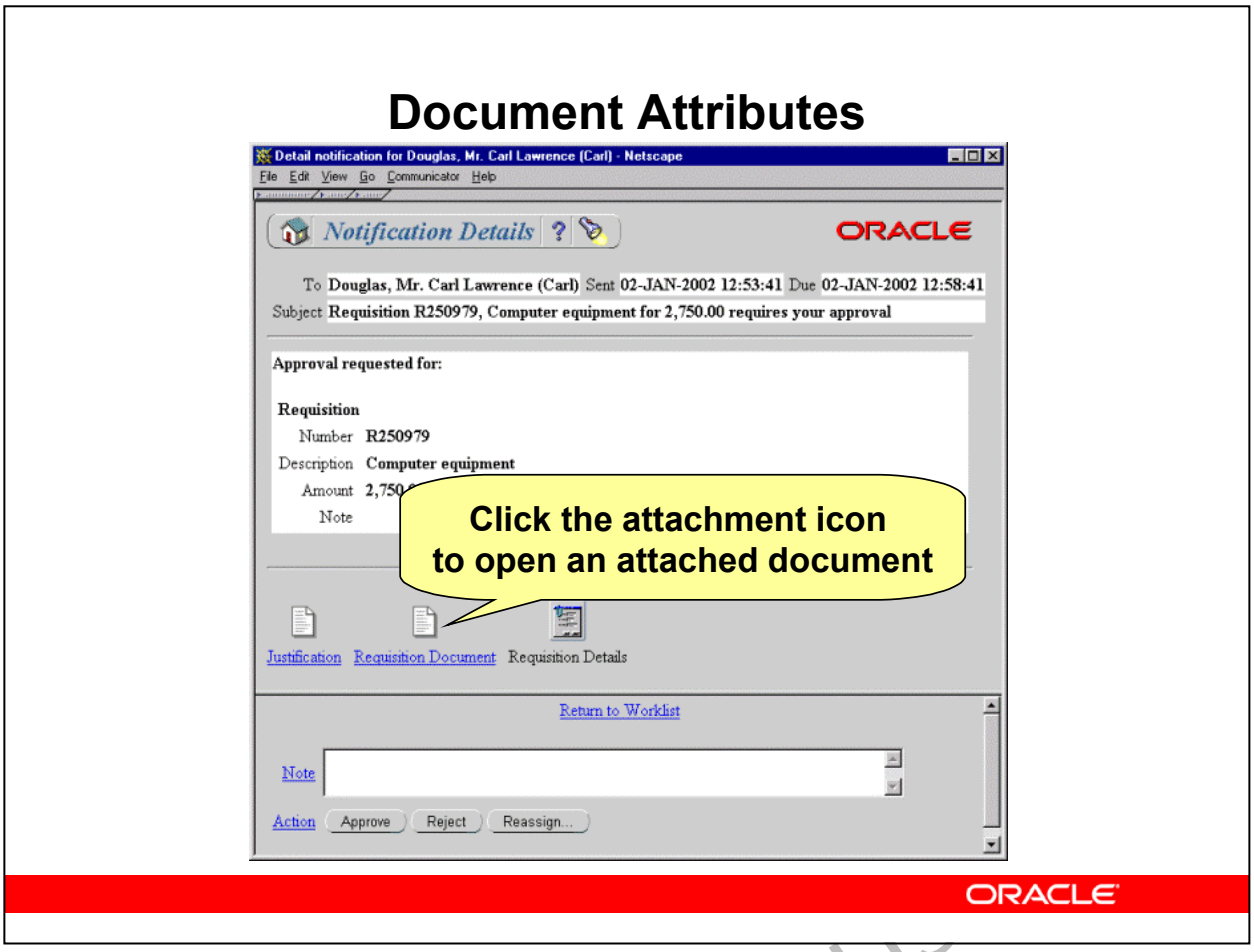
- A PL/SQL CLOB document value is specified in the format:

  `PLSQLCLOB:<procedure>/<document_identifier>`

Replace *<procedure>* with the PL/SQL package and procedure name, separated by a period. Replace *<document_identifier>*with the PL/SQL argument string that you want to pass directly to the procedure. The argument string should identify the document.

If you wish to generate the document identifier for a PL/SQL or PL/SQL CLOB document dynamically, you can token substitute the document identifier with other item type attributes. The item attribute names must be in uppercase and must be separated by a colon. For example:

```
PLSQL:PO_WF.SHOW_REQ/&REQ_TYPE:&REQ_NUMBER
```

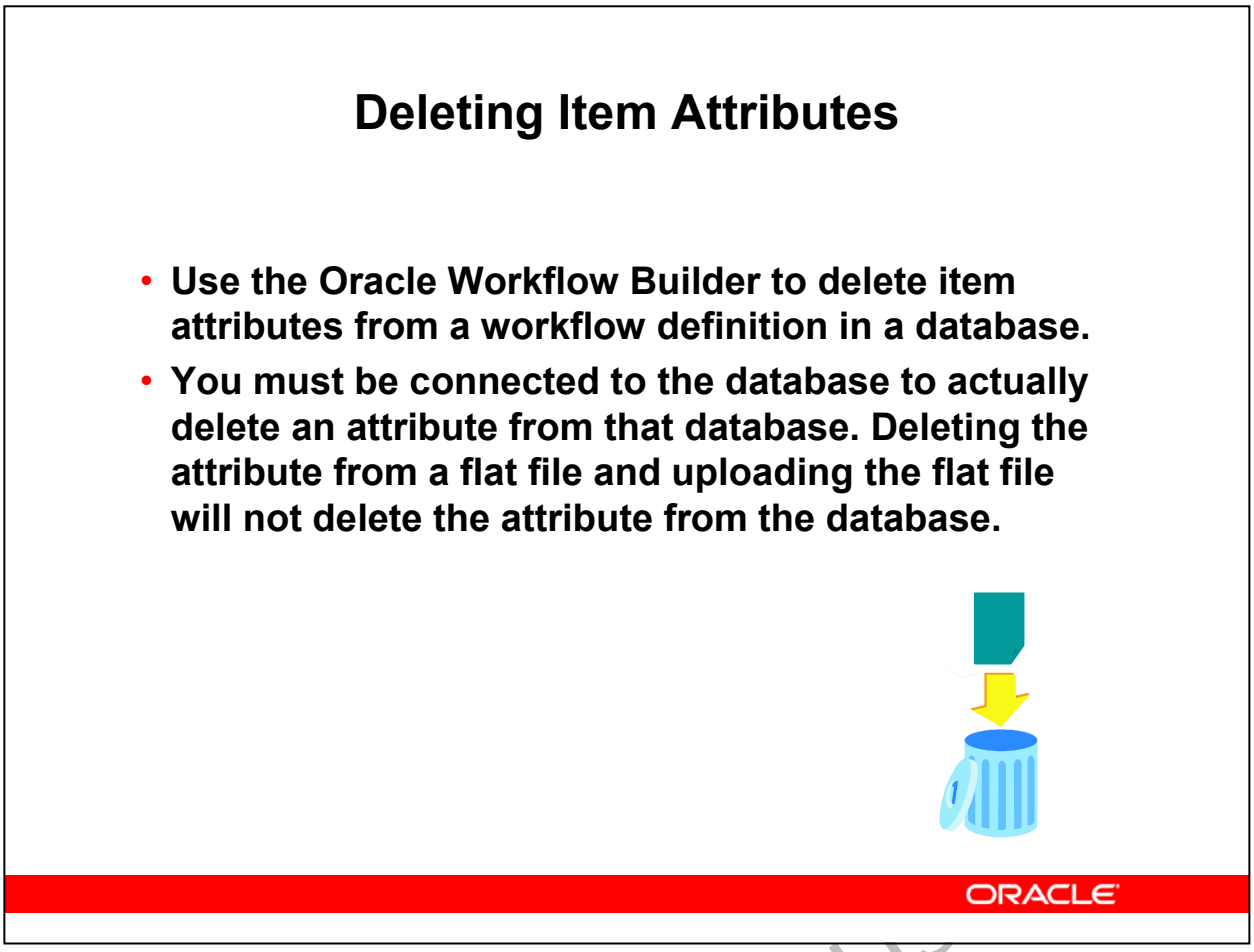


Document Type Attributes

The Notification Details web page supports message attributes of type document. These attributes appear in a notification message as an inline link or as an attachment icon. When you open your notification, you can click a link or an attachment icon to open the referenced document.

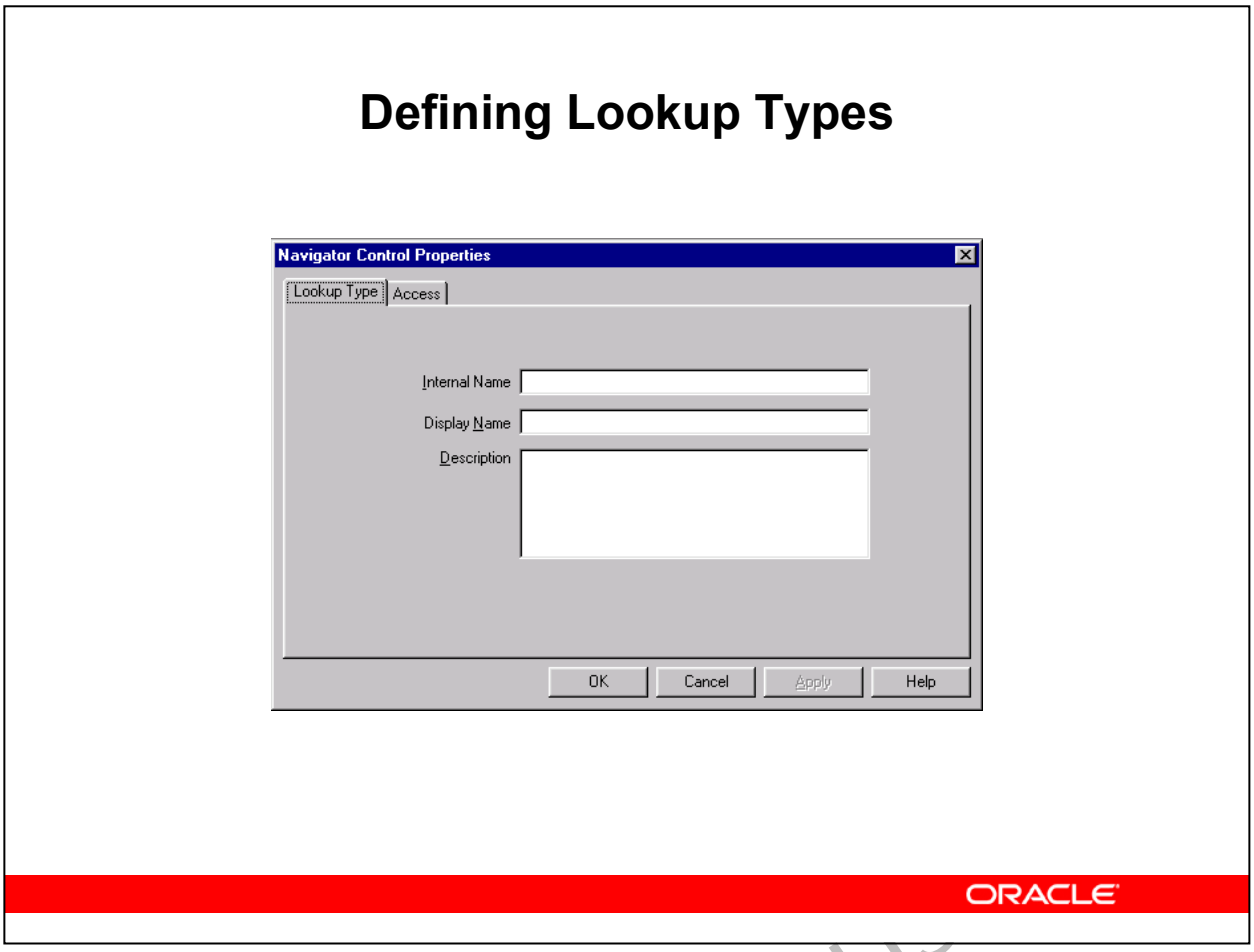
In a plain-text e-mail notification, PL/SQL or PL/SQL CLOB document attributes are attached as plain text to their parent notification. Note that this may render some attachments unreadable if the attachment includes special formatting or your plain text e-mail reader does not recognize attachments. To view these attachments, you should display your notifications in the Notification Details web page.

In an HTML-formatted e-mail notification with an attached PL/SQL or PL/SQL CLOB document attribute, the fully generated PL/SQL document is fetched and attached to the message.



Deleting Item Attributes

Deleting an item attribute from a workflow definition stored in a flat file and then uploading that flat file definition to a database will not delete the item attribute from the definition stored in the database. To delete an item attribute completely, you must delete it from your flat file definition and also delete it specifically from any databases in which that workflow item type is loaded while connected to those databases.



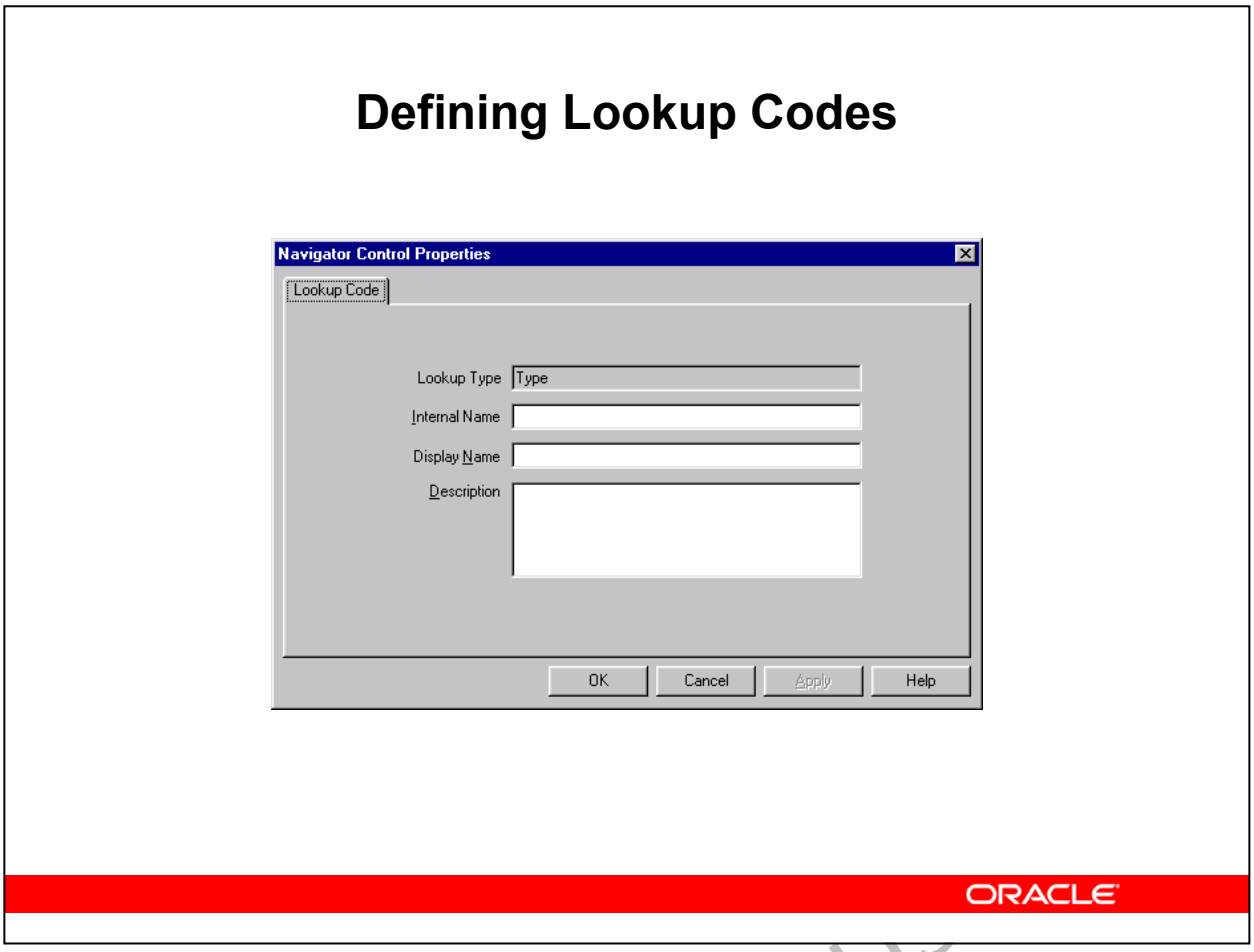
Defining Lookup Types

A lookup type is a static list of values. These lists can be referenced by activities and by item type, message, or activity attributes. For example, an activity can reference a lookup type for its possible result values, while a message attribute can reference a lookup type as a means of providing a list of possible responses to the performer of a notification.

Although lookup types must be associated with a given item type and are defined in the navigator tree beneath the Lookup Types branche of that item type, lookup types can be referenced by components in other item types.

To define a lookup type:

1. In the Oracle Workflow Builder, select the item type you want in the navigator tree. Then choose New Lookup Type from the Edit menu.
2. In the Lookup Type property page, enter an internal name for the lookup type. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the lookup type.
4. Enter a description of the lookup type.
5. Choose Apply to save your changes.



Defining Lookup Codes

Lookup codes represent the actual values in a lookup type. The lookup codes for a lookup type are defined in the navigator tree beneath the lookup type.

To define a lookup code:

1. In the Oracle Workflow Builder, select the lookup type you want in the navigator tree. Then choose New Lookup Code from the Edit menu.
2. In the Lookup Code property page, enter an internal name for the lookup code. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the lookup code.
4. Enter a description of the lookup code.
5. Choose Apply to save your changes.

Practice - Defining Item Type Attributes

Overview

In this practice, you will define item type attributes for the Vacation Proposal item type you created in the Creating a Workflow Process practice.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Note: In order to use the sample solution scripts provided for these practices, you must enter the internal names for all objects you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names you define.

Tasks

1. Define item type attributes.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Open to open the wfvac*XX*.wft data store you defined in the Creating a Workflow Process practice.

   3. In the navigator tree, select your item type.

   4. From the Edit menu, choose New > Attribute.

   5. Define the following properties for the item attribute:
      - Internal Name: REQUESTOR
      - Display Name: Requestor
      - Type: Role

      Choose OK.

   6. From the Edit menu, choose New > Attribute.

   7. Define the following properties for the item attribute:
      - Internal Name: APPROVER
      - Display Name: Approver
      - Type: Role

      Choose OK.

8. From the Edit menu, choose New > Attribute.

9. Define the following properties for the item attribute:
   - Internal Name: FROM_DATE
   - Display Name: From Date
   - Type: Date
   - Format: DD-MON-RRRR

   Choose OK.

10. From the Edit menu, choose New > Attribute.

11. Define the following properties for the item attribute:
   - Internal Name: TO_DATE
   - Display Name: To Date
   - Type: Date
   - Format: DD-MON-RRRR

   Choose OK.

12. From the Edit menu, choose New > Attribute.

13. Define the following properties for the item attribute:
   - Internal Name: COMMENTS
   - Display Name: Comments
   - Type: Text

   Choose OK.

14. You can optionally create a URL item attribute to test how Oracle Workflow includes URL links in notifications. From the Edit menu, choose New > Attribute.

15. Define the following properties for the item attribute:
   - Internal Name: URL_EXAMPLE
   - Display Name: URL Example
   - Type: URL
   - Frame Target: Full Window
   - Default Value: http://www.oracle.com

   Choose OK.

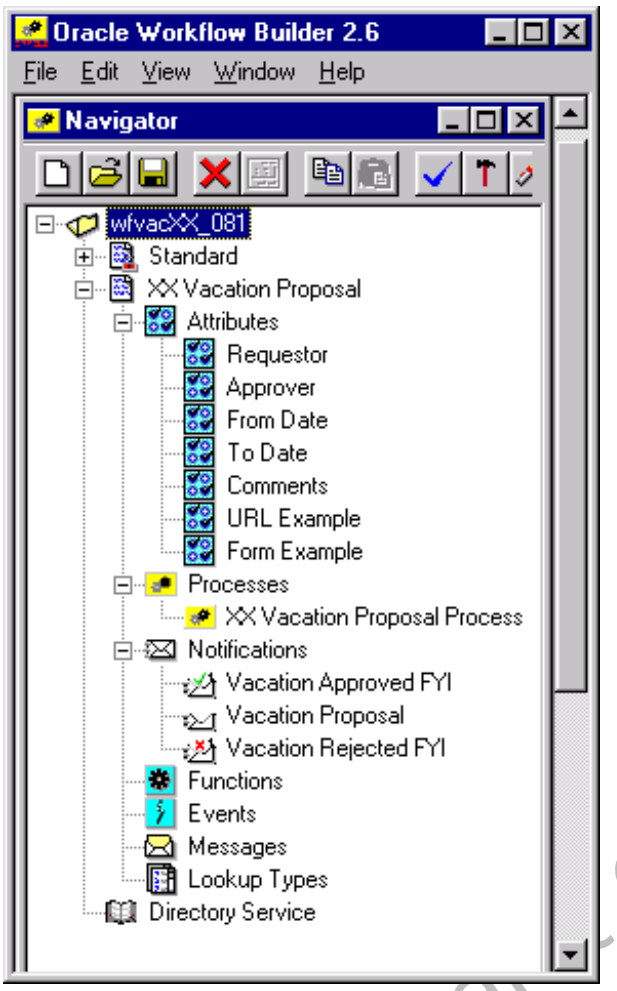
16. If you are using Oracle Workflow embedded in Oracle E-Business Suite, you can optionally create a form item attribute to test how Oracle Workflow attaches Oracle E-Business Suite forms to notifications. From the Edit menu, choose New > Attribute.

17. Define the following properties for the item attribute:
   - Internal Name: FORM_EXAMPLE
   - Display Name: Form Example

- Type: Form
- Default Value: FND_FNDSCAUS

Choose OK.

18. From the File menu, choose Save. The Oracle Workflow Builder will verify your workflow. Click Save in the Workflow Error window to finish saving the workflow definition.



Summary

In this lesson, you should have learned how to:

- **Define item type attributes.**
- **Define lookup types and lookup codes.**

ORACLE

Defining Messages and Notification Activities

Chapter 9

Defining Messages and Notification Activities

ORACLE

Objectives

After completing this lesson, you should be able to do the following:

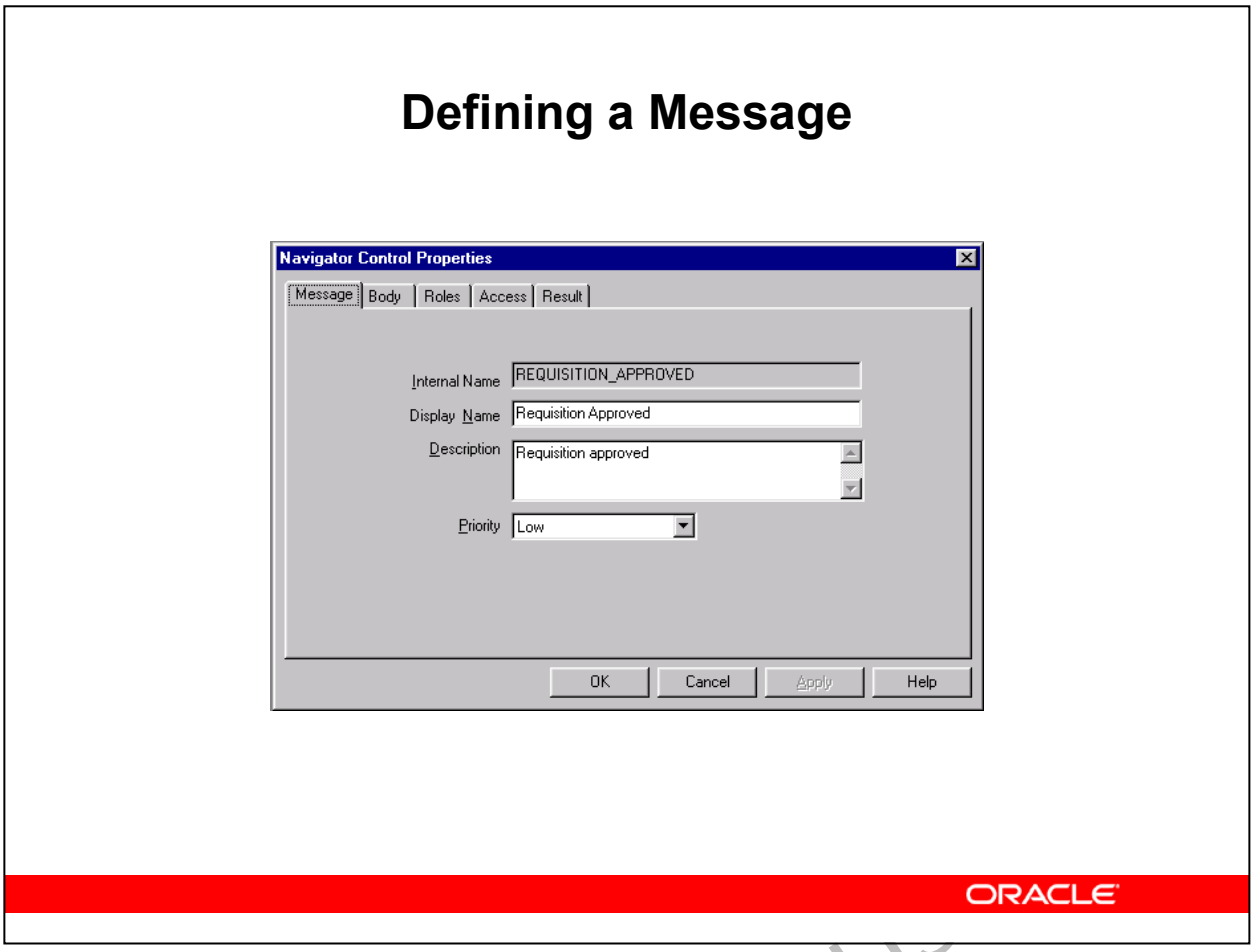
- **Define messages.**
- **Define message attributes.**
- **Define notification activities.**

ORACLE

Notification Activities

- A notification activity is a step in a workflow process to communicate with a human being.
- A notification activity sends a message to a user containing the information needed to complete the work.
- The notification may or may not require a response from the recipient.

ORACLE

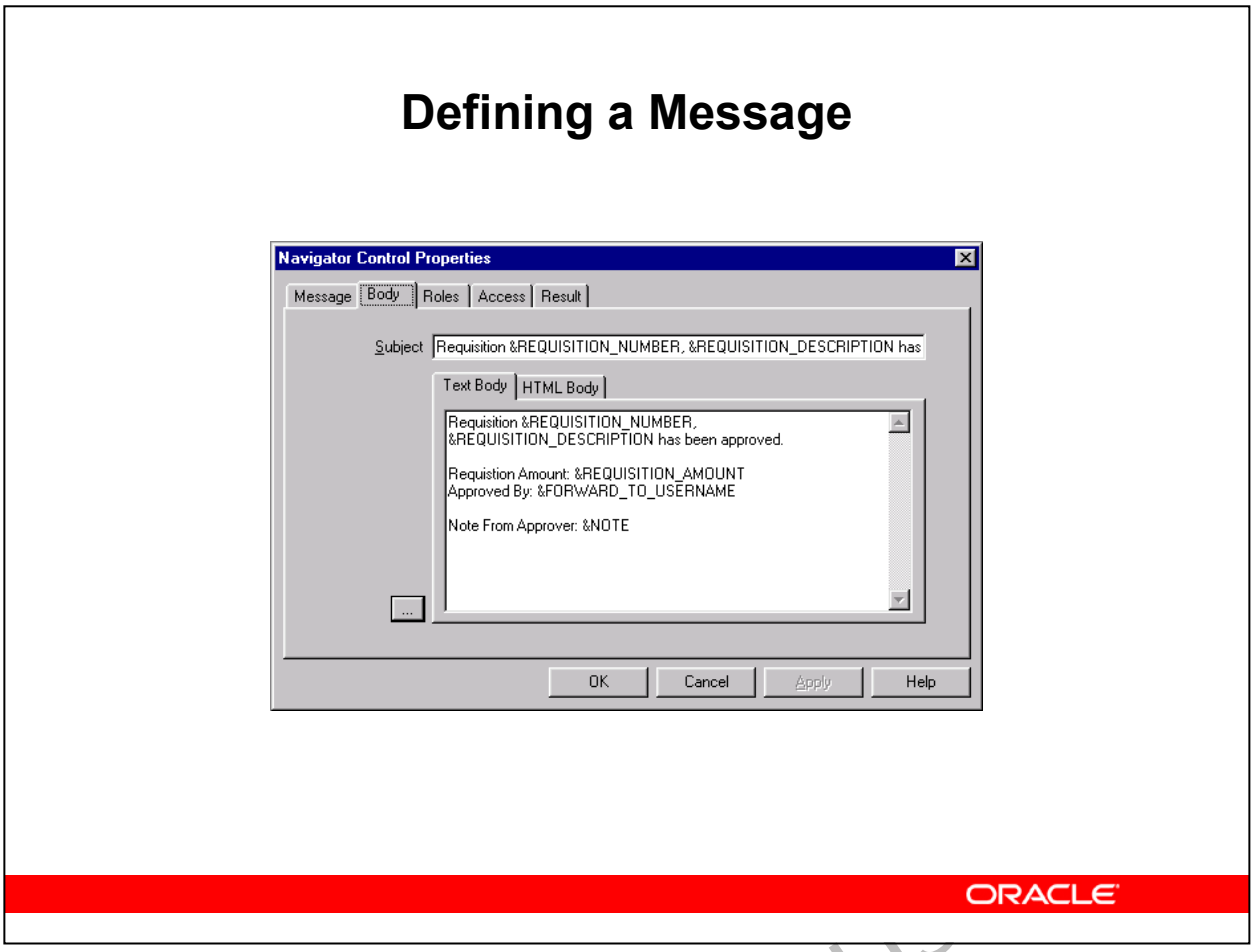


Defining a Message

Messages must be associated with an item type and are created in the navigator tree under the Messages branch of the item type.

To create a new message:

1. In the Oracle Workflow Builder, select the item type you want in the navigator tree. Then choose New Message from the Edit menu.
2. In the Message property page, enter an internal name for the message. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the message.
4. Enter a description of the message.
5. Select the priority for the message (High, Normal, or Low).
   - The priority entered here is specific to the message.
   - Message recipients can sort their notifications based on the priority.
6. Choose Apply to save your changes.



Defining a Message

After creating a new message, you must define the subject and body of the message.

You can optionally include message attributes in the subject or body.

- In the subject or body of the message, specify the internal name of the message attribute and precede that internal name with an ampersand ("&").
- Message attributes are token substituted with a runtime value when the message is sent.

To define the body for a message:

1. In the message property pages, choose the Body tab.
2. Enter the subject line for the message.
3. Enter the plain text message body in the Text Body tab.
4. Enter the HTML-formatted message body in the HTML Body tab.
5. Choose Apply to save your changes.

Images in an HTML message:

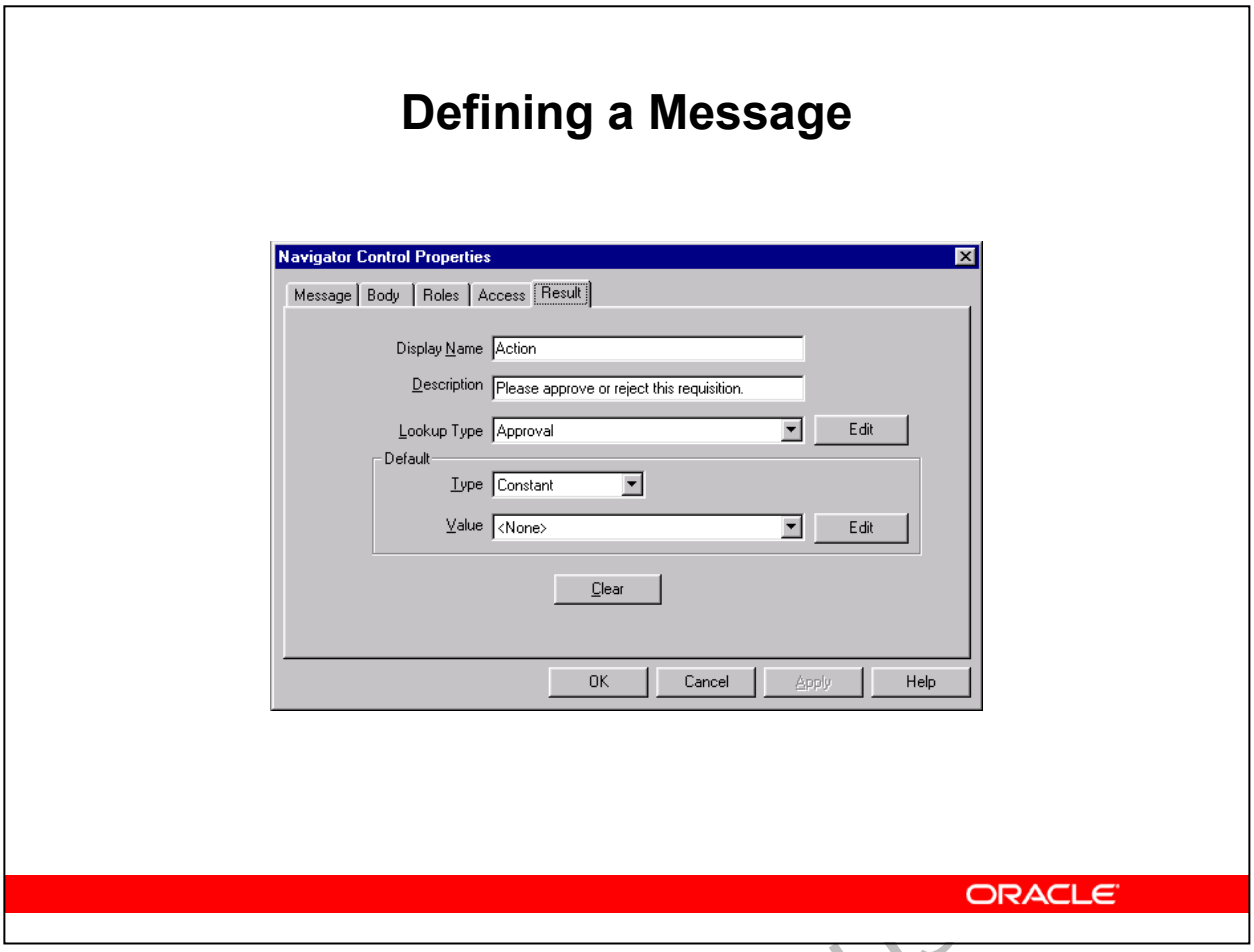
Reference the image source in your HTML message body as

```
<img SRC="/OA_MEDIA/<your image file>.gif">
```

All image files referenced in an HTML message body must be stored in the /OA_MEDIA/ virtual directory as configured in your web listener. To display the files available, go to:

`<wfwebagent>/OA_MEDIA/`

You can also create a custom virtual directory mapping in your web listener and use this as the directory source of your image files.



Defining a Message

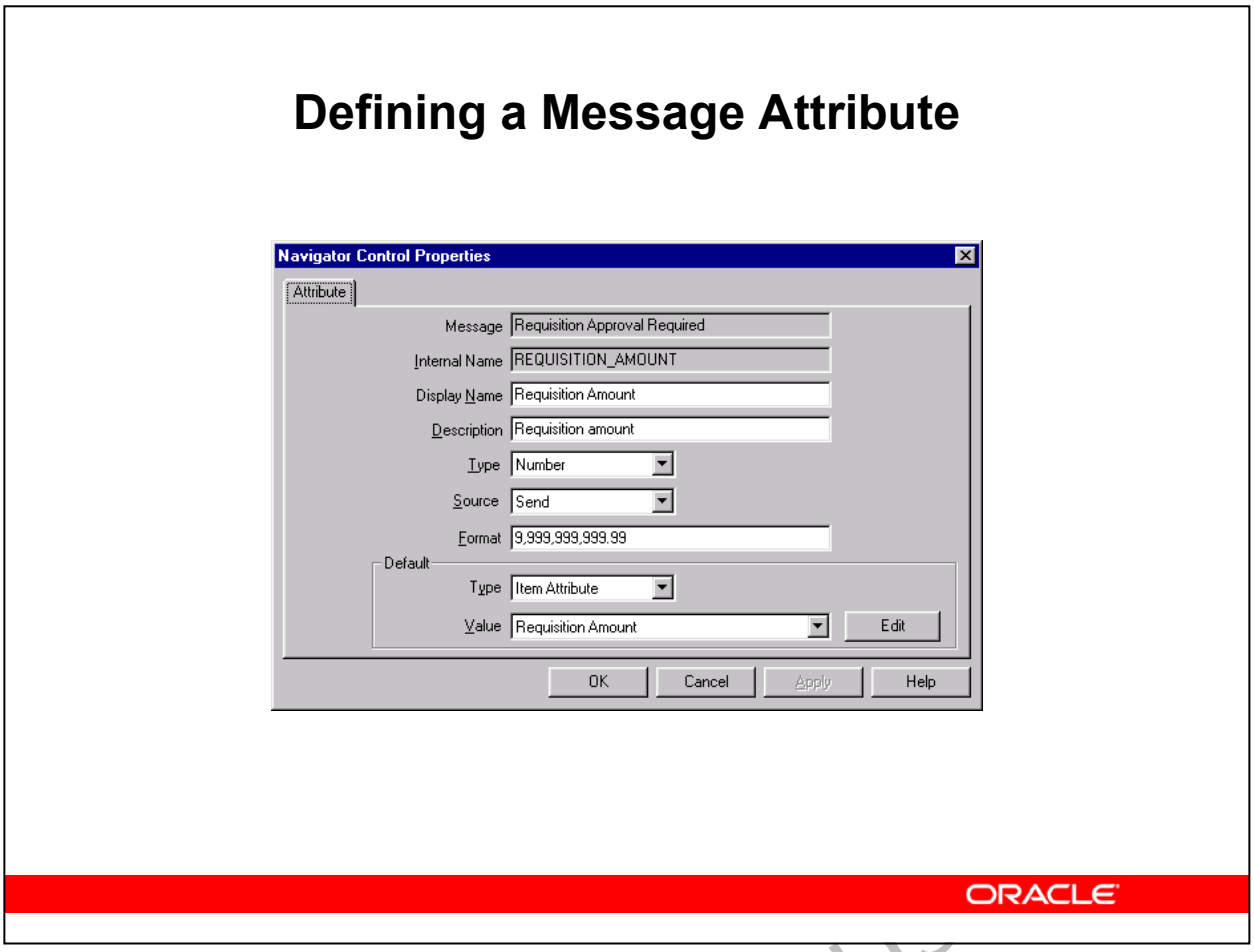
Define a result for a message when you want to send a message that prompts for a response and then uses that response as the result of its notification activity. The result is a special 'Respond' message attribute with the internal name RESULT.

The icon for a message in the navigator tree includes a red question mark in its bottom left corner when the message has a result defined for it.

To define the result for a message:

1. In the message property pages, choose the Result tab.
2. Enter a display name for the RESULT attribute. Oracle Workflow uses the display name as the response prompt in the notification message.
3. Enter a description for the RESULT attribute.
4. Select a lookup type to provide the possible result values. The lookup type associated with the message result should be the same as the result type of the notification activity that sends the notification message.
5. Optionally set the default value of the RESULT attribute to reference either a constant or the runtime value of an item type attribute.

6. Choose Apply to save your changes.



Defining a Message Attribute

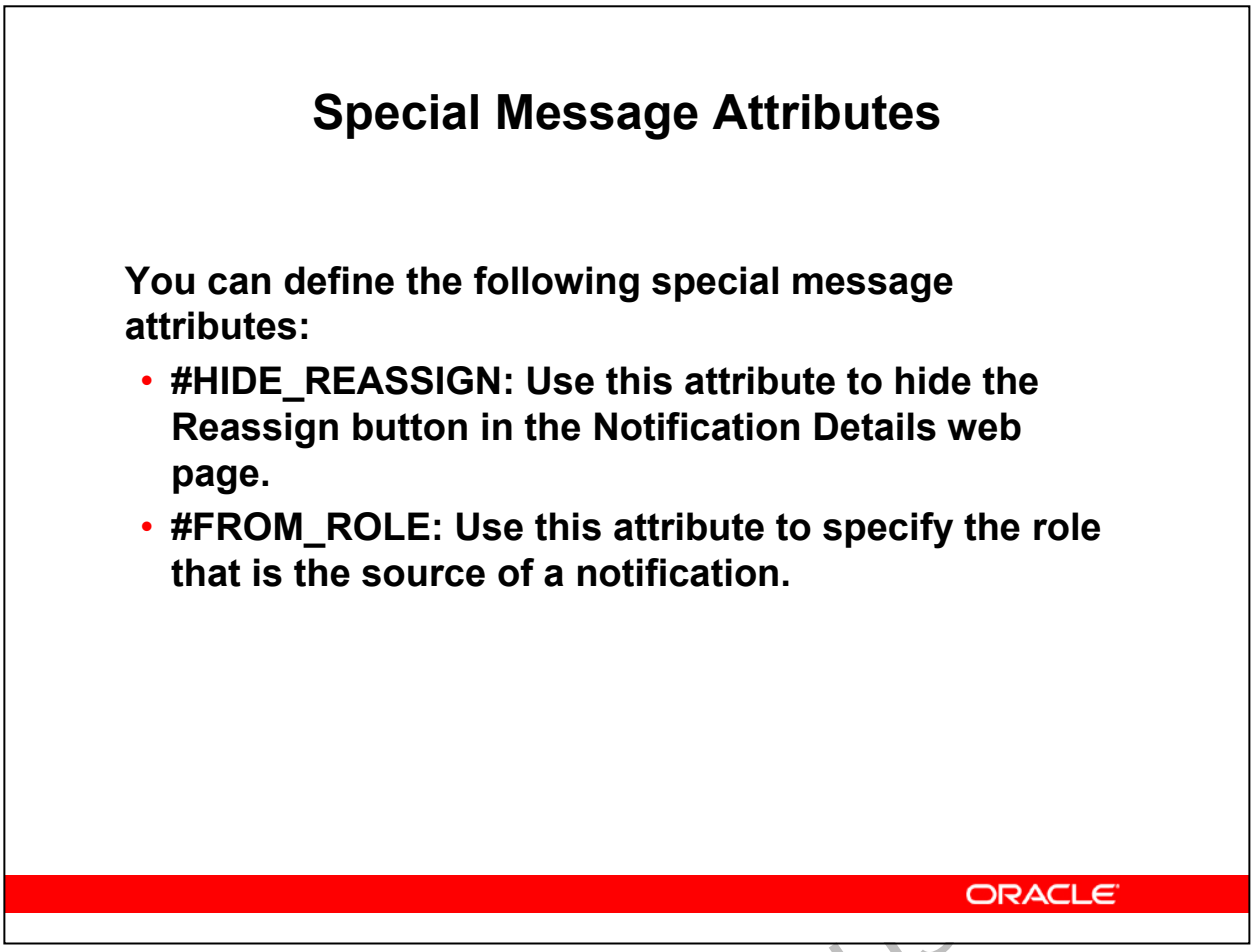
Message attributes must be associated with a message and are defined in the navigator tree beneath that message.

To define a message attribute:

1. In the Oracle Workflow Builder, select the message you want in the navigator tree. Then choose New Attribute from the Edit menu.
2. In the Attribute property page, enter an internal name for the attribute. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the attribute.
4. Enter a description of the attribute.
5. In the Type field, select the data type for the attribute.
6. Depending on the data type, enter format information, if applicable.
7. Specify either Send or Respond in the Source field to indicate whether the attribute should send information to the recipient or prompt the recipient for a response, respectively. You should set the source to Send for any message attributes that you include in the message subject and message body for token substitution.

8. Set the default value of a Send attribute to reference either a constant or the runtime value of an item type attribute. You can optionally set a default value for a Respond attribute as well.
9. Choose Apply to save your changes.

Note: To define a message attribute that references an existing item attribute, you can also select the item attribute in the navigator tree and drag it onto the message you want in the navigator tree. Then edit the property page that appears, making sure the message attribute has the appropriate source. The default type is automatically set to Item Attribute and the default value references the originating item attribute.



Special Message Attributes

#HIDE_REASSIGN

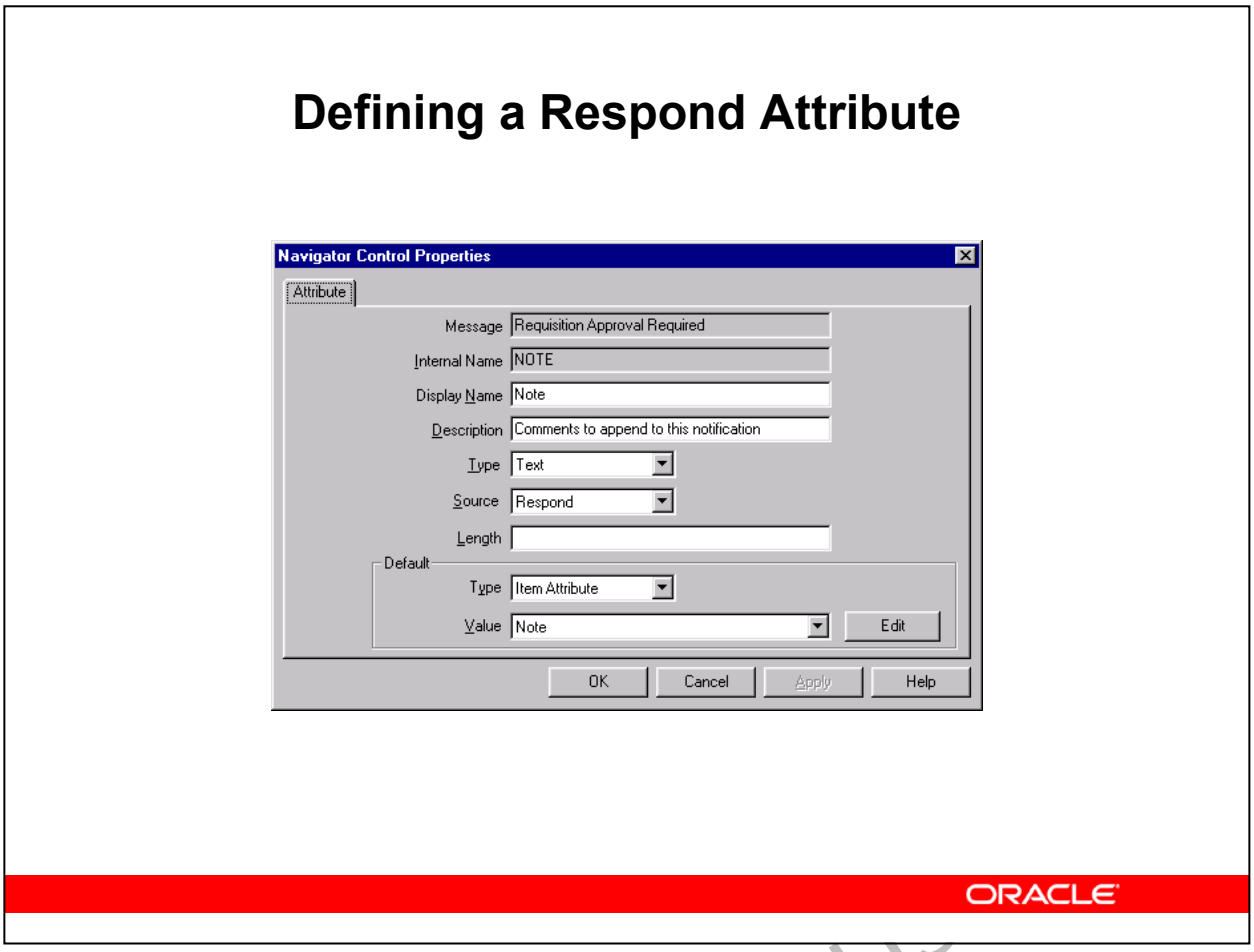
When users view a notification from their Worklist web page, the response frame in the Notification Details page includes the Reassign button by default. If you want to prevent users from reassigning a notification, you can add the #HIDE_REASSIGN attribute to control whether the Reassign button is displayed or hidden.

The #HIDE_REASSIGN attribute must be of type text. To hide the Reassign button, set the value of this attribute to Y. To display the Reassign button, set the value to N.

#FROM_ROLE

The From Role for each notification is displayed in the Worklist web page to give users additional information for reviewing and responding to the notifications. Additionally, the Find Notifications page lets you search for notifications based on the From Role. For example, if you have a notification that informs an approver that a requisition was submitted, you can set the requisition preparer as the From Role for the message.

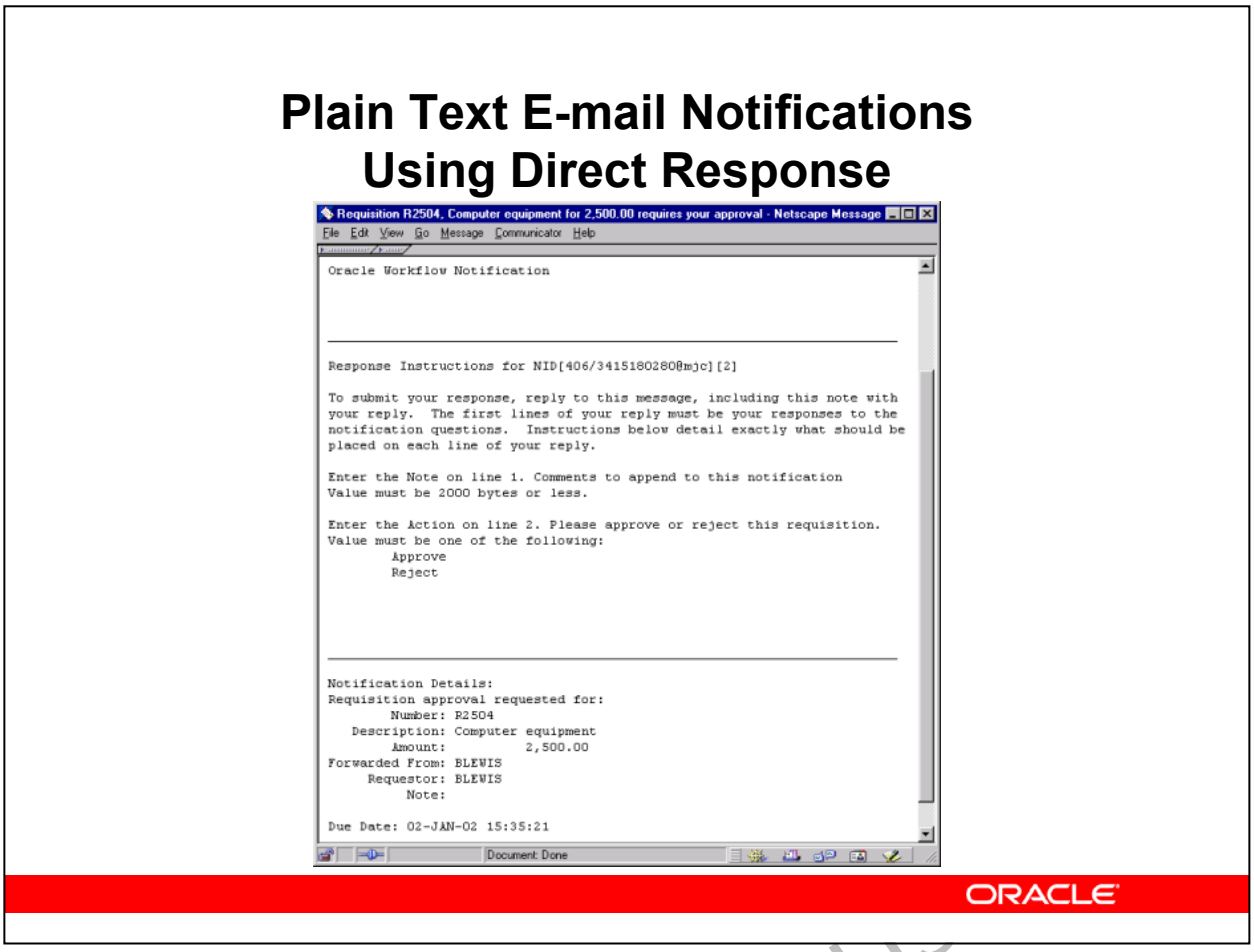
The #FROM_ROLE attribute must be of type role and must have its default value set to an item attribute that holds the value you want to set as the From Role for the message.



Defining a Respond Attribute

If you want to create a message that prompts for a response from the recipient, then define a message attribute with the source set to Respond.

The Notification System uses a message's Respond attributes to generate the response section of the notification. A message can have multiple Respond attributes. The icon for a Respond attribute in the navigator tree includes a red question mark in its bottom left corner to distinguish it from Send attributes.



Plain Text E-mail Notifications Using Direct Response

The Notification System uses the Respond attributes defined for the message to generate the response section of the notification sent to the recipient. The response section is composed differently depending on whether the recipient receives the notification as a plain text e-mail notification, as an HTML-formatted e-mail notification, or through the Notification Details web page. For plain text e-mail notifications, the response section is also composed differently depending on whether your organization uses the direct response method or the templated response method. (HTML e-mail notifications always use the templated response method.)

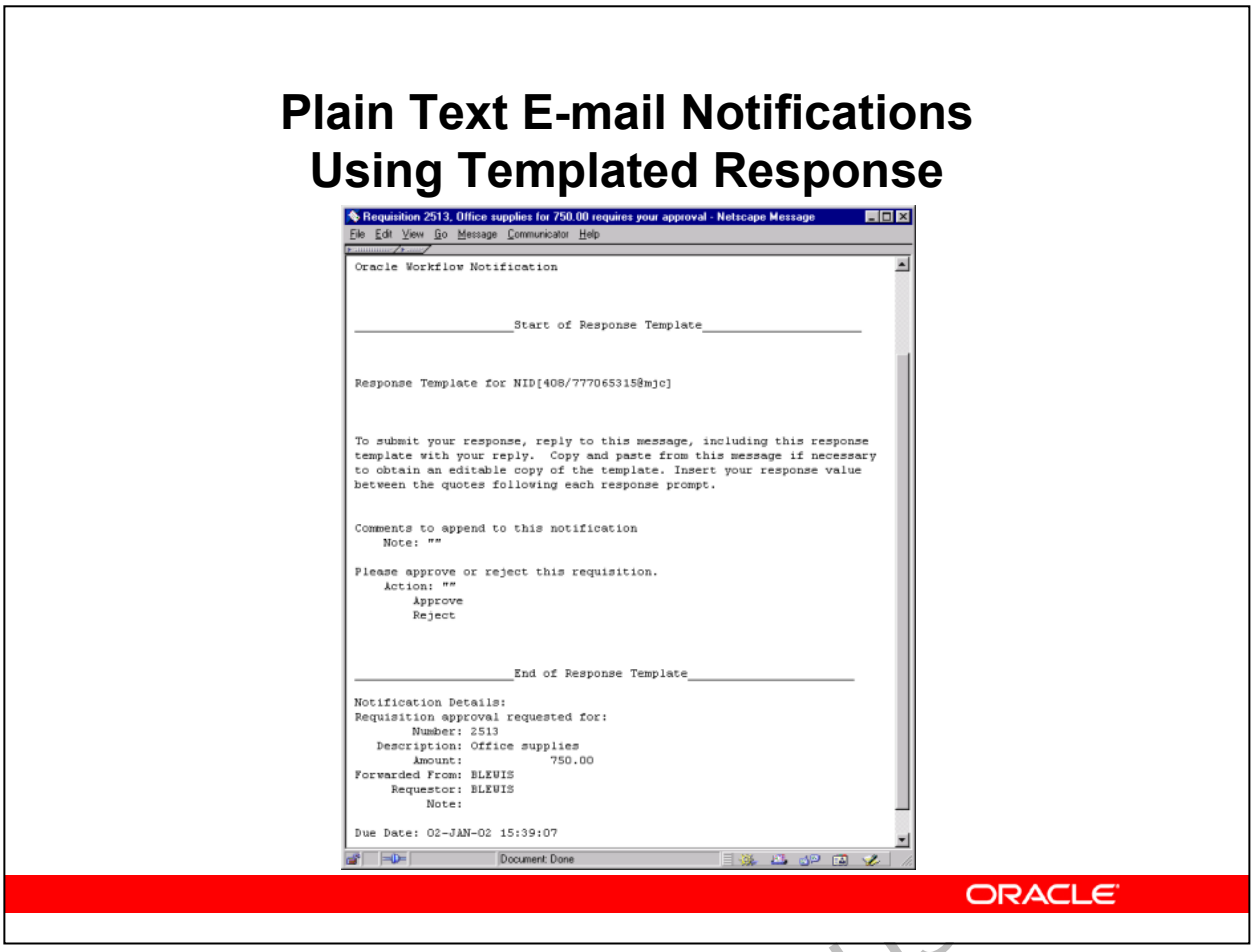
The response section of a plain text e-mail notification using direct response is generated by the following boilerplate text and by information drawn from the properties of the Respond attributes.

```
Enter the <Display Name> on line <Sequence>.
<Description>.     <Type Hint>
```

- Display Name: The display name of the Respond attribute; this name specifies the response value that is expected.
- Sequence: The sequence position of this Respond attribute among all the Respond attributes of the message, as listed in the navigator tree.

- Description: The description of the Respond attribute; this description provides optional information on how to complete a response.
- Type Hint: A standard hint that is provided based on the attribute's type:

Type	Type Hint
Lookup	Value must be one of the following: `<list of lookup codes>`
Date	Value must be a date in the form "`<format>`".
Number	Value must be a number in the form "`<format>`".
Text	Value must be `<format>` bytes or less.

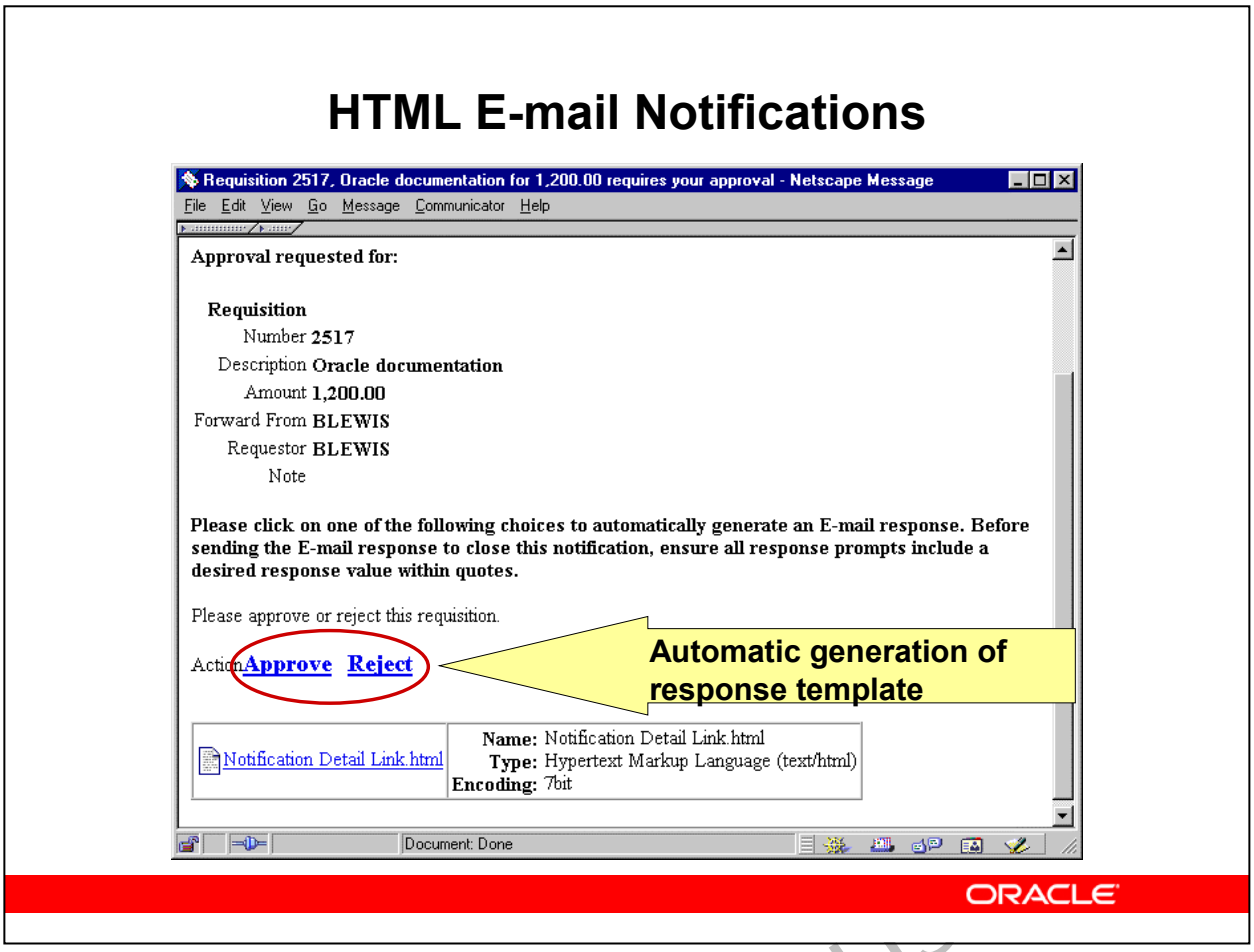


Plain Text E-mail Notifications Using Templated Response

The response section of a plain text e-mail notification using templated response is generated by the following boilerplate text and by information drawn from the properties of the Respond attributes.

```
<Description>
<Display Name>: " "
    <list of lookup codes>
```

- Description: The description of the Respond attribute; this description provides optional information on how to complete a response.
- Display Name: The display name of the Respond attribute; this name specifies the response value that is expected.
- List of lookup codes: If the attribute is of type lookup, the lookup codes that are the possible response values are listed.



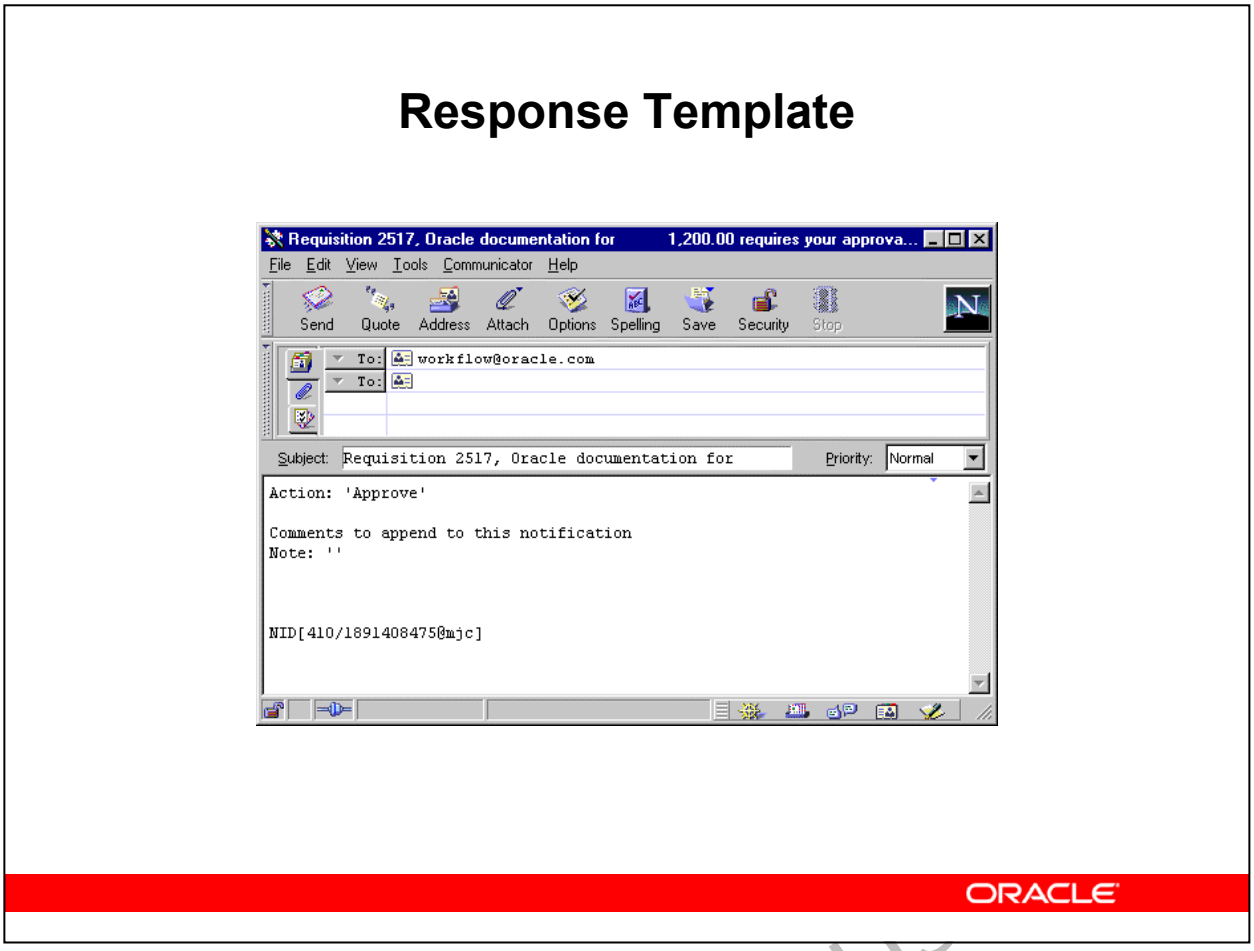
HTML E-mail Notifications

HTML e-mail notifications always use the templated response method.

The response section of an HTML e-mail notification includes links for the possible result response values at the end of the notification. After reviewing the message body of the notification, you click one of the response links.

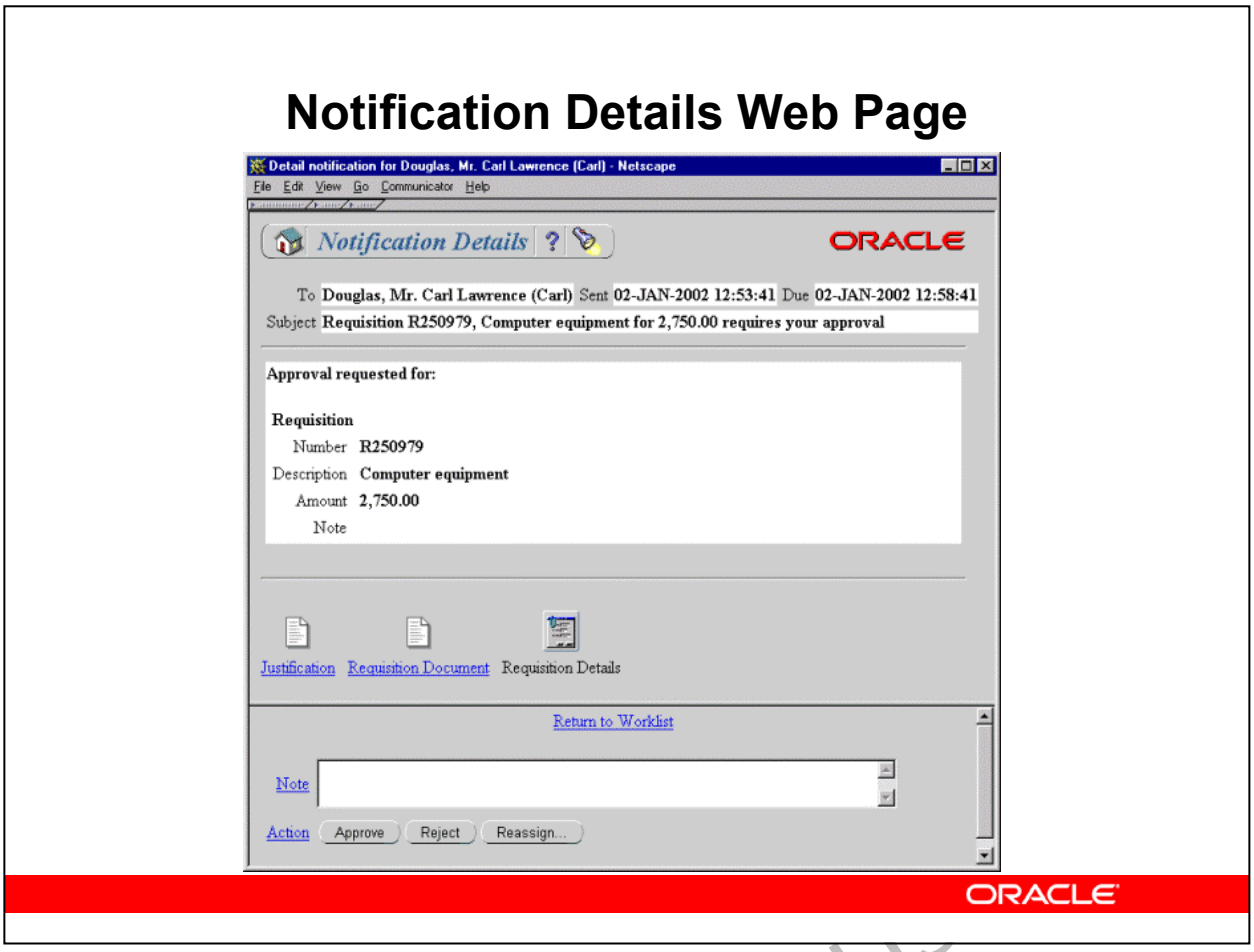
Each response link automatically generates a plain text e-mail reply that contains the correct Reply To: e-mail address as well as a response template in the message body. The response template consists of the required notification ID and access key that identify the notification you are responding to and a response prompt edited with your selected result response.

Depending on the notification, the auto-generated e-mail response template may also prompt you for other information in addition to your selected result response. Any additional prompts and values are generated by information drawn from the properties of each Respond attribute, similarly to the response section for plain text templated response.



Response Template

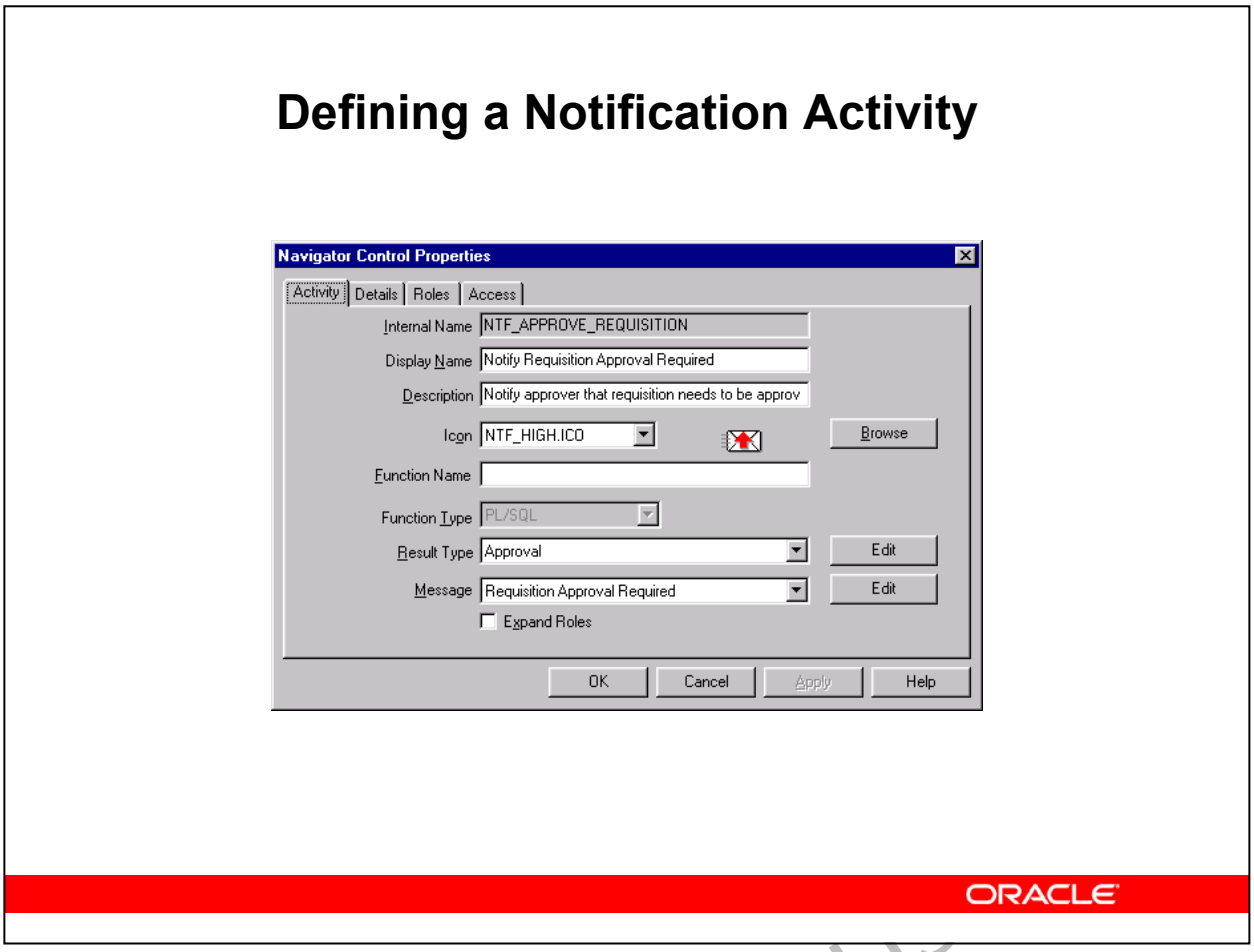
When responding to a notification with an automatically generated template, supply your responses by editing the response value text between the quotes following each prompt.



Notification Details Web Page

The response section of a notification in the Notification Details web page is also generated by information drawn from the properties of the Respond attributes.

- The response prompts are drawn from the display names of the Respond attributes and specify the response values expected.
- Each response prompt also links to information about the properties of that Respond attribute.
- A single Respond attribute of type URL lets you replace the standard notification response section and processing with that of another web page. However, this behavior appears only if the notification is viewed from the Notification Details web page. (Multiple URL Respond attributes do not replace the standard response section; they simply let you provide the recipient with multiple links to other Web pages.)



Defining a Notification Activity

Notification activities must be associated with an item type and are created in the navigator tree beneath the Notifications branch of the item type.

To define a notification activity:

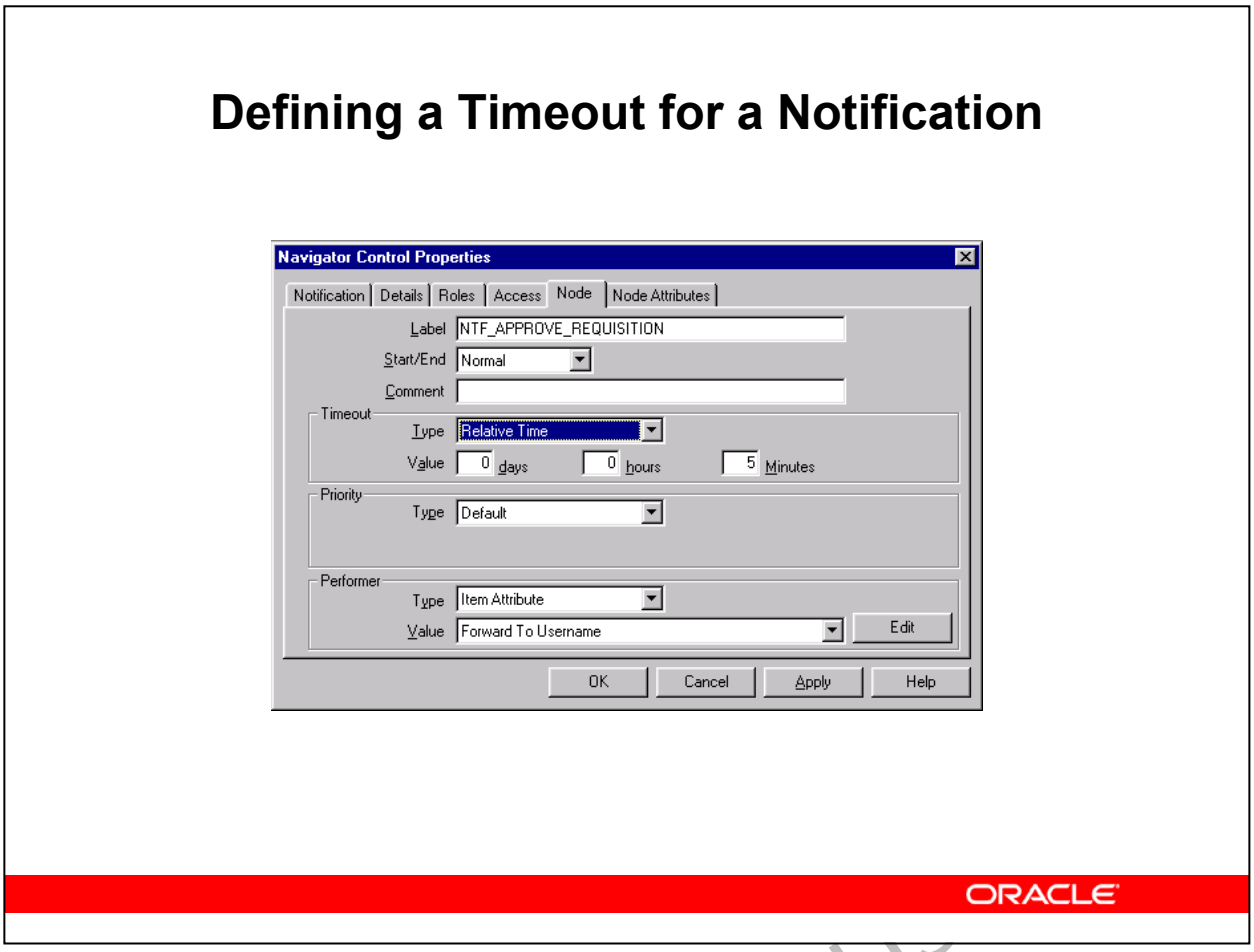
1. In Oracle Workflow Builder, select the item type that you want in the navigator tree. Then choose New Notification from the Edit menu.
2. In the Activity property page, enter an internal name for the activity. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the activity.
4. Enter a description of the activity.
5. Choose an icon that identifies the activity.
6. Optionally specify a PL/SQL stored procedure in the Function field. The procedure is known as a post-notification function and can be used to couple processing logic to the notification activity.

7. Specify the result type that contains the possible results that the notification can return. If the message for the notification includes a result, then the result type of the notification activity must be the same as the lookup type that the message result references.
8. Select the message that this notification sends.
9. Select the Expand Roles check box if you want to send this notification to a multi-user role and ensure that each user receives an individual copy of the notification. You should always select the Expand Roles check box for FYI-type notifications.
9. Choose Apply to save your changes.
10. Optionally select the Details tab to display and modify additional activity details.
11. Optionally select the Access tab to set the access levels allowed to modify this activity.

Creating a Notification Activity Node

- When you add a notification activity node to a workflow process, define the properties for that individual instance of the activity in the Node tab of the activity property pages.
- Important properties for a notification activity node include:
  - Timeout
  - Priority
  - Performer

ORACLE

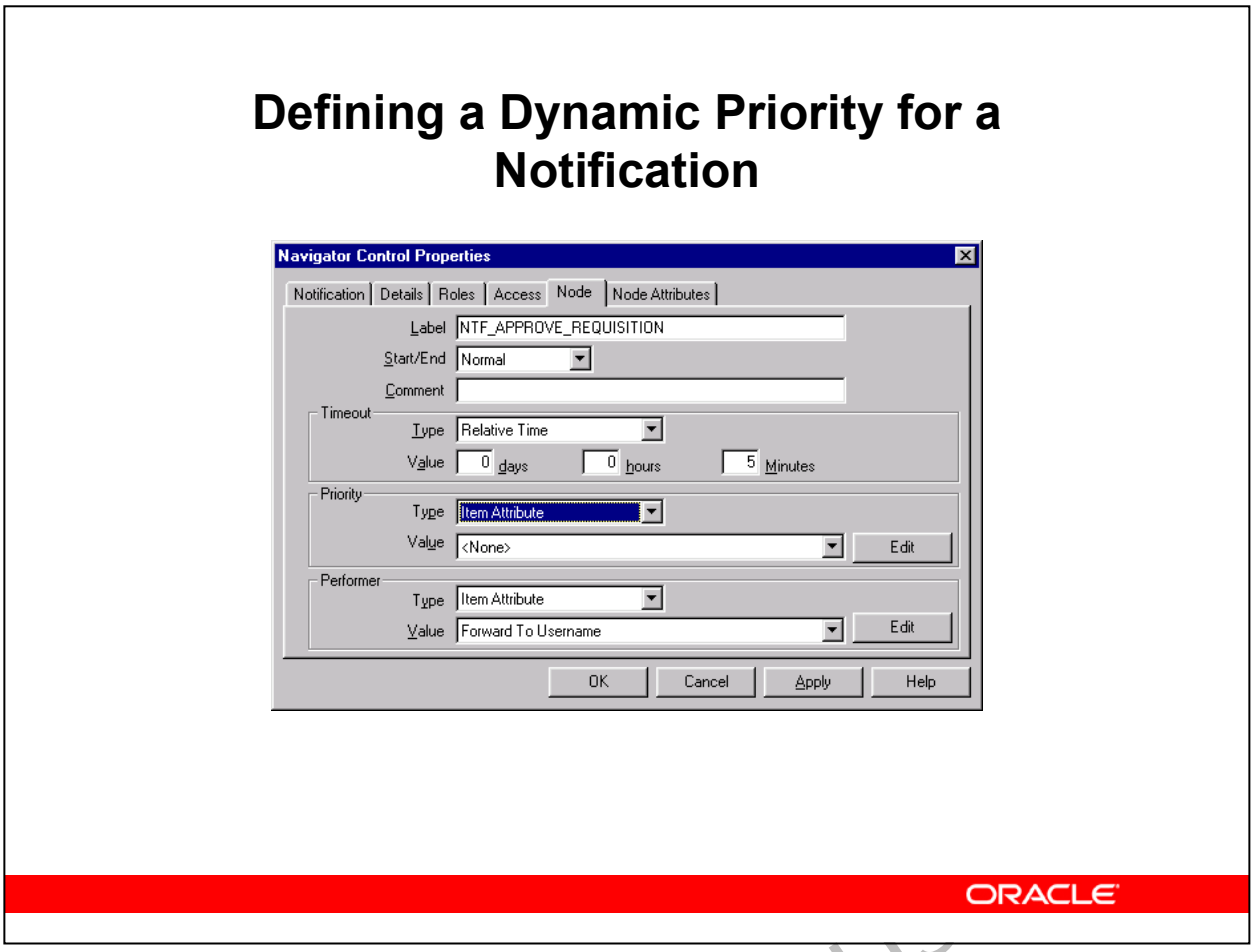


Defining a Timeout for a Notification

The timeout value for a notification activity node specifies whether the notification response must be received by a certain time.

- Choose 'No Timeout' if the notification does not have to be completed by a certain time.
- Choose 'Relative Time' if you want the notification to be completed by some constant relative time. You can enter any combination of days, hours, and minutes to specify when the activity times out. The value you enter is interpreted as a relative offset from the begin date of the activity, in the unit of minutes.

  Note: A relative timeout value of zero means there is no timeout.
- Choose 'Item Attribute' if you want the notification to be completed by some relative time or date that is determined dynamically at runtime. You must first create an item attribute of type number or date to store the dynamic timeout value and then reference that predefined item attribute here.



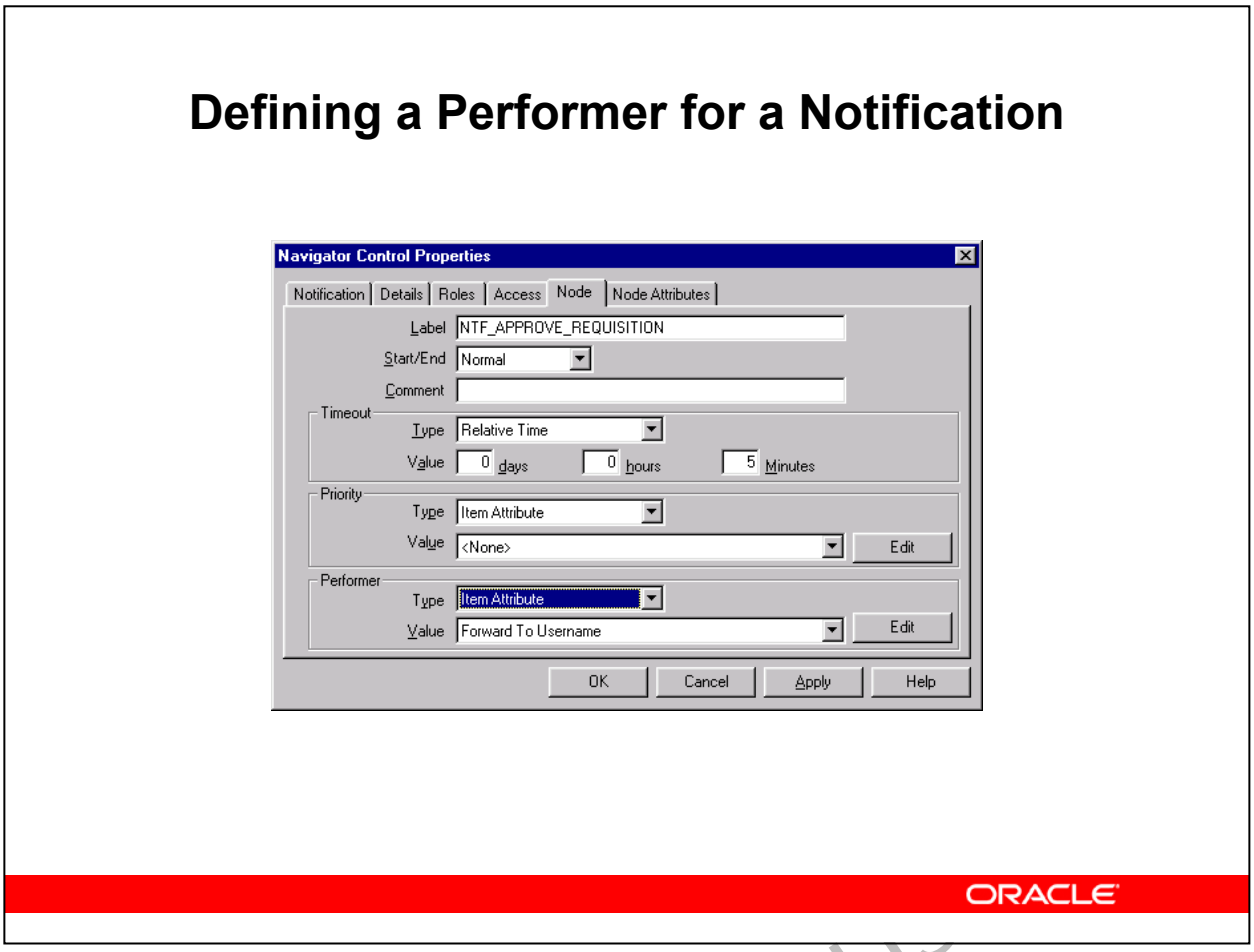
Defining a Dynamic Priority for a Notification

By default, the Notification System assigns a priority to a notification based on the priority value set for the message associated with the notification.

You can set a new priority for an individual notification activity node that overrides the default priority of the message it sends. At runtime, the message is assigned the override priority.

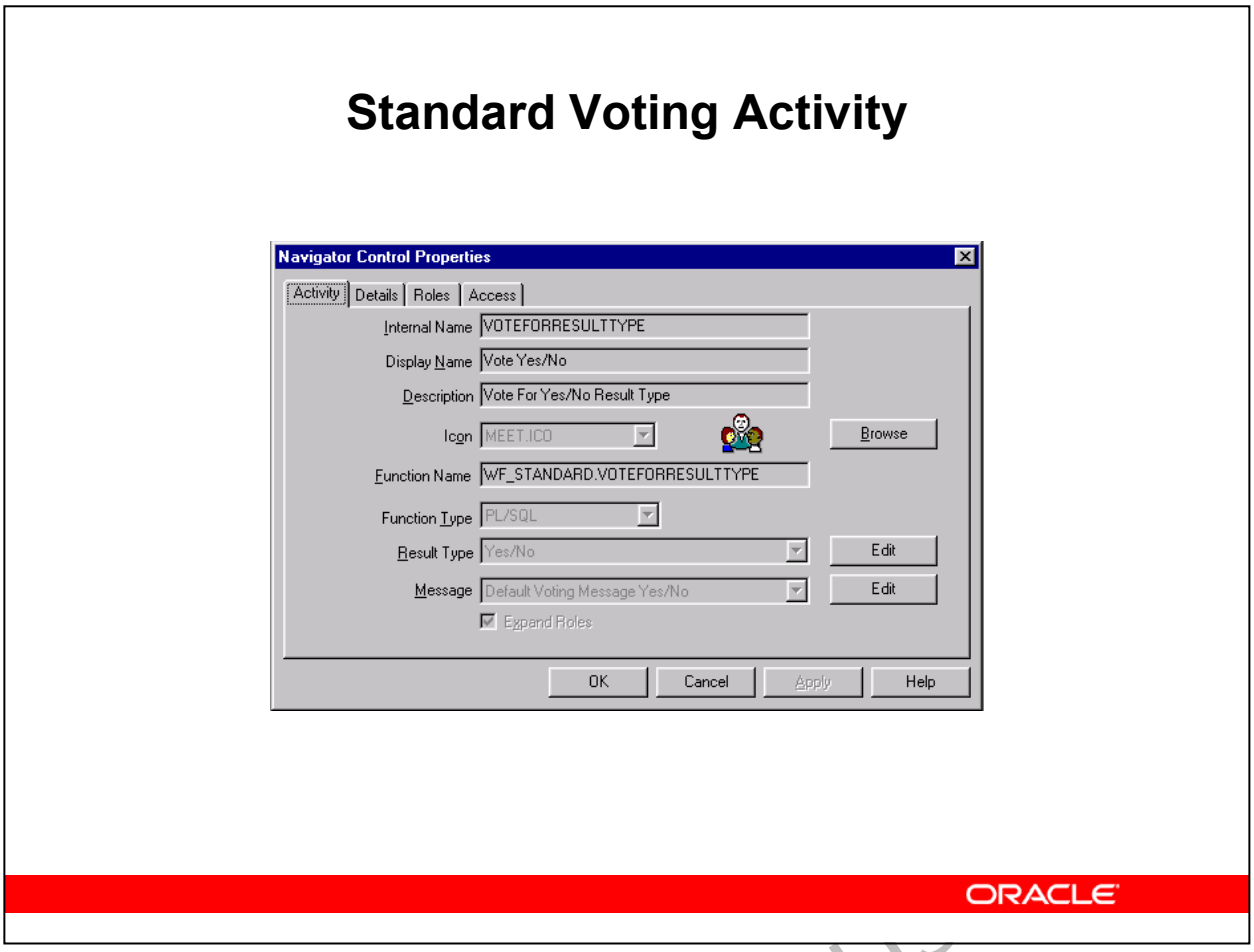
You can set the override priority as a constant or reference an item attribute. The item attribute must be of type number, and the following ranges apply:

- High : 1-33
- Normal : 34-66
- Low : 67-99



Defining a Performer for a Notification

The performer is the role to whom the notification is sent. You can select either a constant role name or an item type attribute that dynamically determines the role at runtime. Note that you must first create an item attribute of type role or text to store the dynamic role name before you can reference that item attribute in the Performer field.



Standard Voting Activity

Vote Yes/No is a notification activity associated with the Standard item type. This activity can be customized. Use this activity to send a notification to all users in a role and tally the Yes/No responses of the users. The tallied responses determine the final result of the notification activity.

The activity sends a message called “Default Voting Message Yes/No” that includes a message result. The message prompts the recipient to reply with “Yes” or “No.” The possible results of this notification activity are stored in the lookup type called Yes/No.

Choose “Expand Roles” for the notification activity to ensure that all users in the role receive a copy of the notification.

When the activity receives responses, it executes a PL/SQL procedure called WF_STANDARD.VOTEFORRESULTTYPE to tally the Yes/No responses.

Vote Yes/No has three activity attributes associated with it:

- Voting Option: Specifies when and how to tally the votes
- Percent Yes: Specifies what percentage of Yes responses are required for a Yes result
- Percent No: Specifies what percentage of No responses are required for a No result

Practice - Defining Messages and Notification Activities

Overview

In this practice, you will define messages and complete the notification activity definitions for the Vacation Proposal item type you created in the Creating a Workflow Process practice.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the connect string for the class database and the username and password of the Oracle Workflow database account.

Tasks

1. Define messages for your item type.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Open to open the wfvac*XX*.wft data store you defined in the Creating a Workflow Process practice.

   3. In the navigator tree, select your item type.

   4. From the Edit menu, choose New > Message.

   5. Define a Vacation Proposal message that informs the approver of the vacation proposal. The message should include the requestor, approver, and the start and end dates of the proposed vacation. Set the message priority to Normal. In the Result tab, select the Approval lookup type.

   6. In the Navigator window, drag and drop the Requestor, Approver, From Date, To Date, and Comments item attributes onto the message to create the corresponding message attributes with those item attributes as their default values. If you created URL Example and Form Example attributes, drag and drop those attributes onto the message as well. For the Comments attribute, set the Source field to Respond. For the URL Example attribute, select the Attach Content check box.

   7. From the Edit menu, choose New > Message.

8. Define a Vacation Rejected message that informs the requestor that the vacation proposal was rejected. Set the message priority to Low. Drag and drop the appropriate item attributes onto the message to create the corresponding message attributes with those item attributes as their default values.

9. From the Edit menu, choose New > Message.

10. Define a Vacation Approved message that informs the requestor that the vacation proposal was approved. Set the message priority to High. Drag and drop the appropriate item attributes onto the message to create the corresponding message attributes with those item attributes as their default values.

2. Complete the notification activity definitions for the item type.

Solution:

11. In the Navigator window, drag and drop each message onto the corresponding notification activity to assign that message to the activity.

12. Open the process diagram window for the process.

13. Double-click the Vacation Proposal node and choose the Node tab. Set the performer for the node to the Approver item attribute.

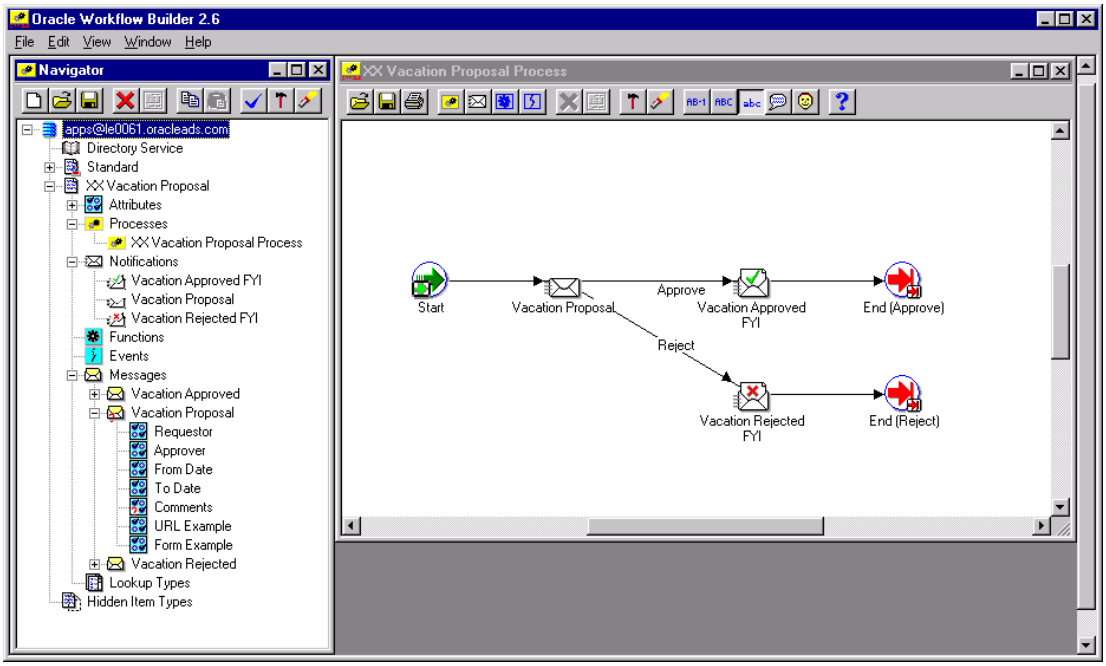
14. Double-click the Vacation Rejected FYI node and choose the Node tab. Set the performer for the node to the Requestor item attribute.

15. Double-click the Vacation Approved FYI node and choose the Node tab. Set the performer for the node to the Requestor item attribute.

16. In the Navigator window, click the Verify button to verify your workflow. Because you have now defined the underlying components for your process, the Workflow Builder should not display any warnings. Click OK.

17. From the File menu, choose Save to save your work to your workflow definition file.

18. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



Summary

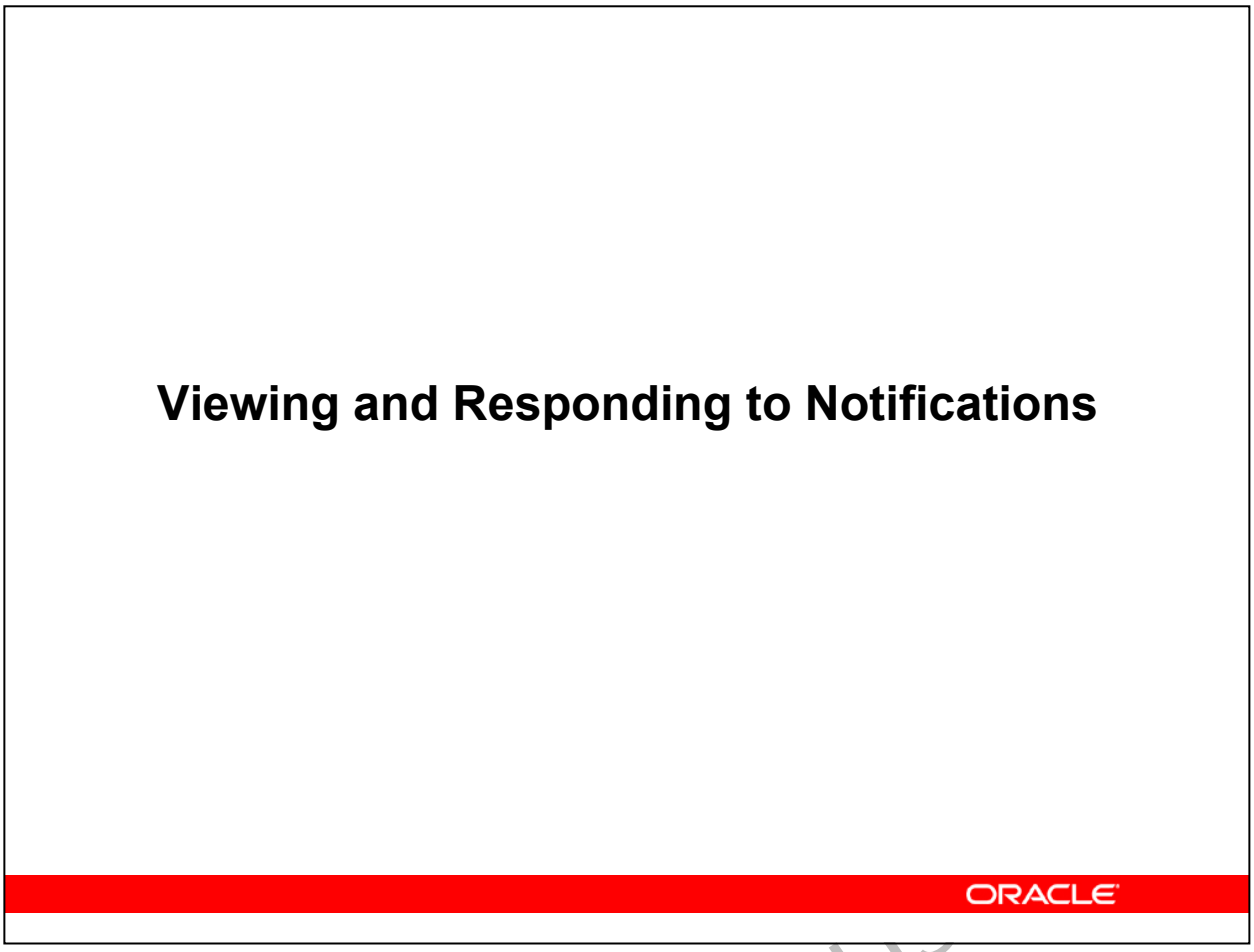
In this lesson, you should have learned how to:

- Define messages.
- Define message attributes.
- Define notification activities.

ORACLE

Viewing and Responding to Notifications

Chapter 10



Objectives

After completing this lesson, you should be able to do the following:

- **Access the Oracle Workflow web pages.**
- **Respond to notifications from the Notifications Worklist web page.**
- **Respond to notifications using e-mail.**
- **Define notification routing rules.**

ORACLE

Accessing the Oracle Workflow Web Pages

Standalone Oracle Workflow

If you are using the standalone version of Oracle Workflow, use the Oracle Workflow home page to access Oracle Workflow's web-based features.

To access the standalone Oracle Workflow home page, use a web browser to connect to the following URL:

<*webagent*>/wfa_html.home

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

Embedded Oracle Workflow

If you are using the version of Oracle Workflow embedded in Oracle E-Business Suite, use a Workflow responsibility to access Oracle Workflow's web-based features.

New Self-Service Web Page Format

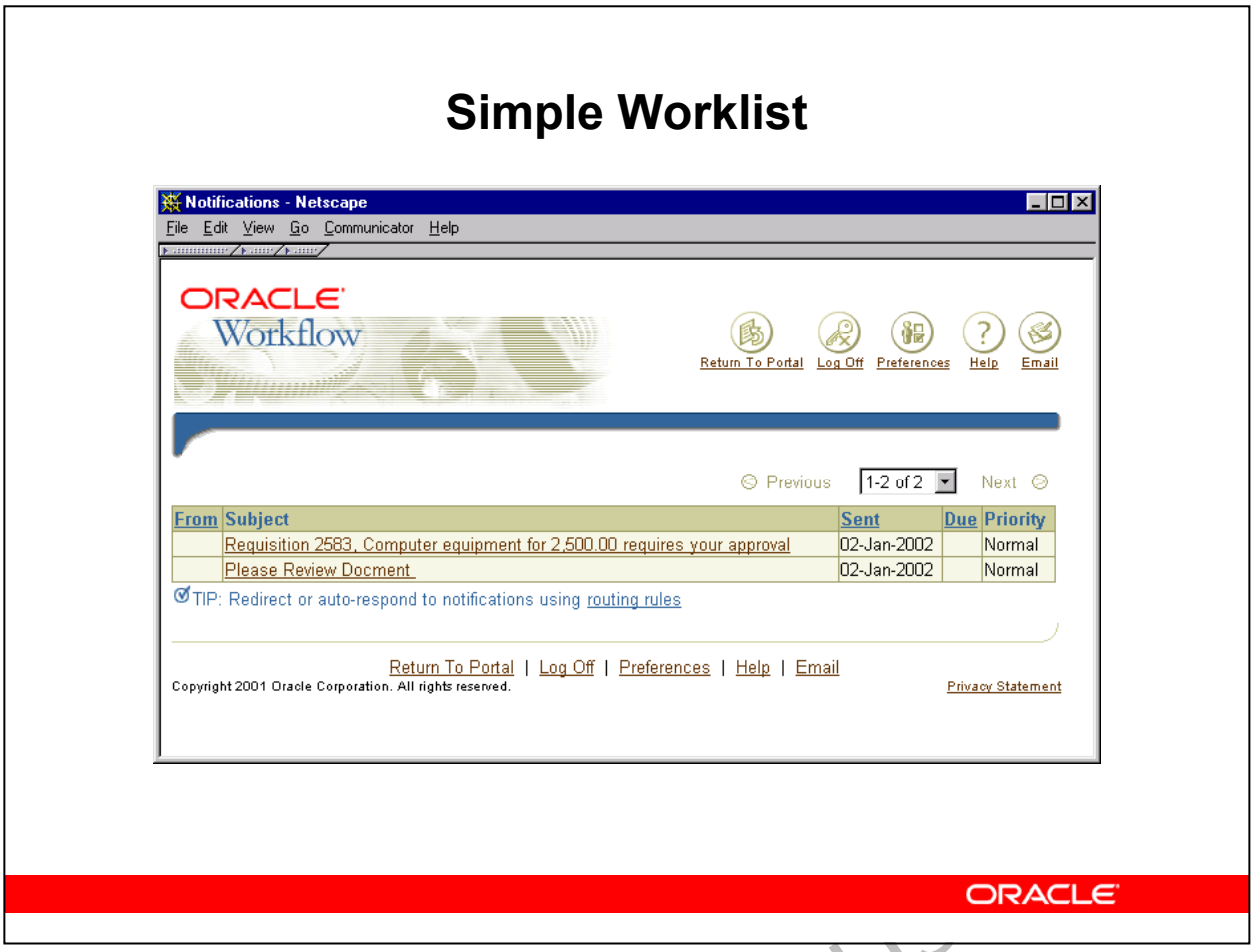
The Oracle Workflow web pages are being converted to the new format for Oracle self-service web applications. Depending on your version of Oracle Workflow and which patches you have applied, you may see Oracle Workflow web pages in the previous format or in the new format.

Currently, the Notifications Worklist and the Notification Rules web pages are available in the new format for the version of Oracle Workflow embedded in Oracle E-Business Suite.

Viewing Notifications from a Web Browser

- **The Oracle Workflow Notifications web pages let you view and respond to your notifications.**
  - **Find Notifications page**
  - **Notifications Worklist page**
  - **Notification Details page**
- **All users can view their notifications in these web pages, regardless of their notification preference.**
- **If your notification preference is set to 'Do not send me mail' in the User Preferences web page, then you can only access your notifications through the Notifications web pages.**

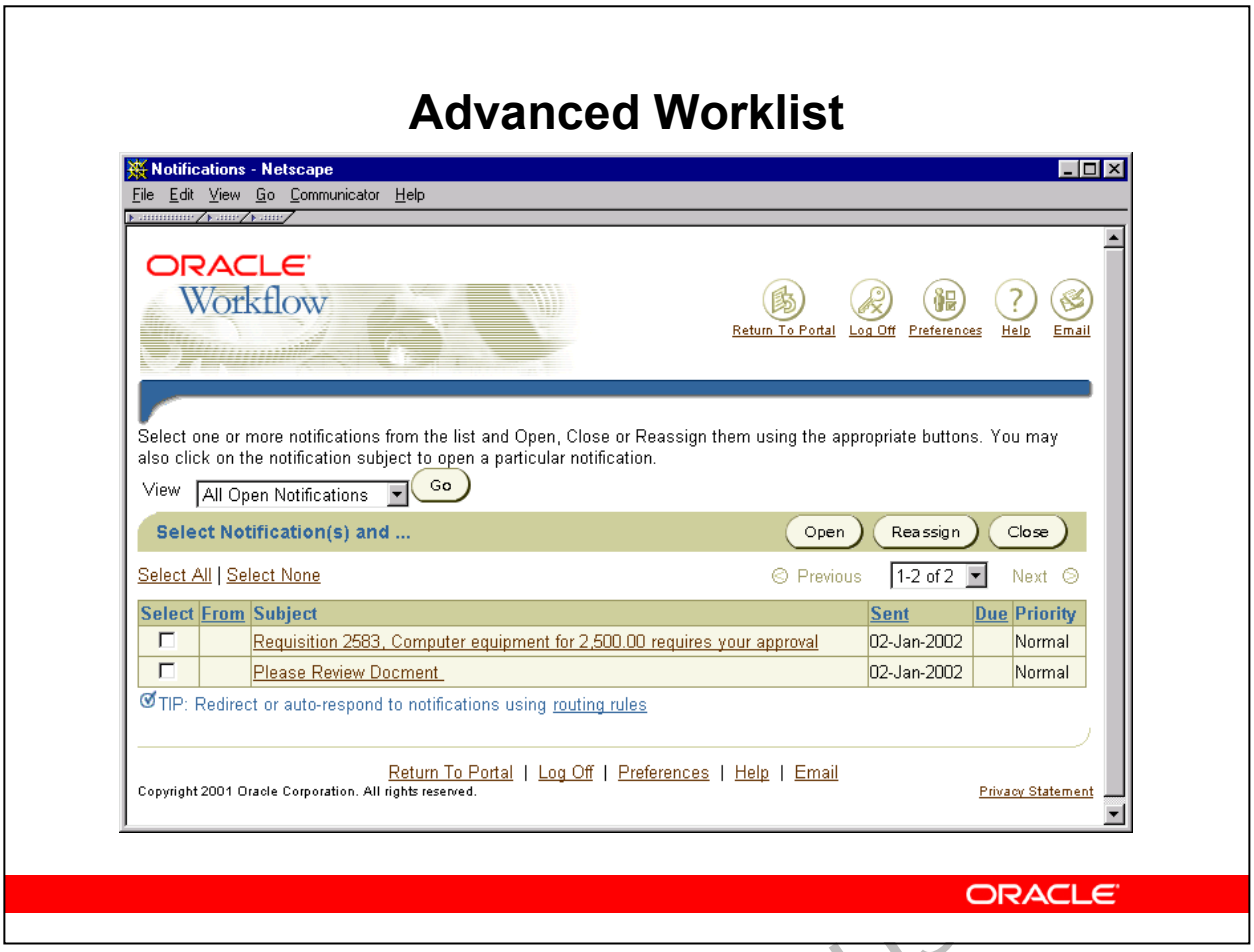
ORACLE



Simple Worklist

- The Simple Worklist in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch. Choose Simple Worklist from a Workflow responsibility to access this page.
- The Simple Worklist lists all your open notifications, displaying the from role, subject, sent date, due date, and priority for each notification.

  Note: The due date is determined by the timeout value set for the notification.
- Click the From, Sent, Due, or Priority column heading to sort your notifications by that column in ascending or descending order.
- To navigate to the full details of any notification, click the notification's Subject link.

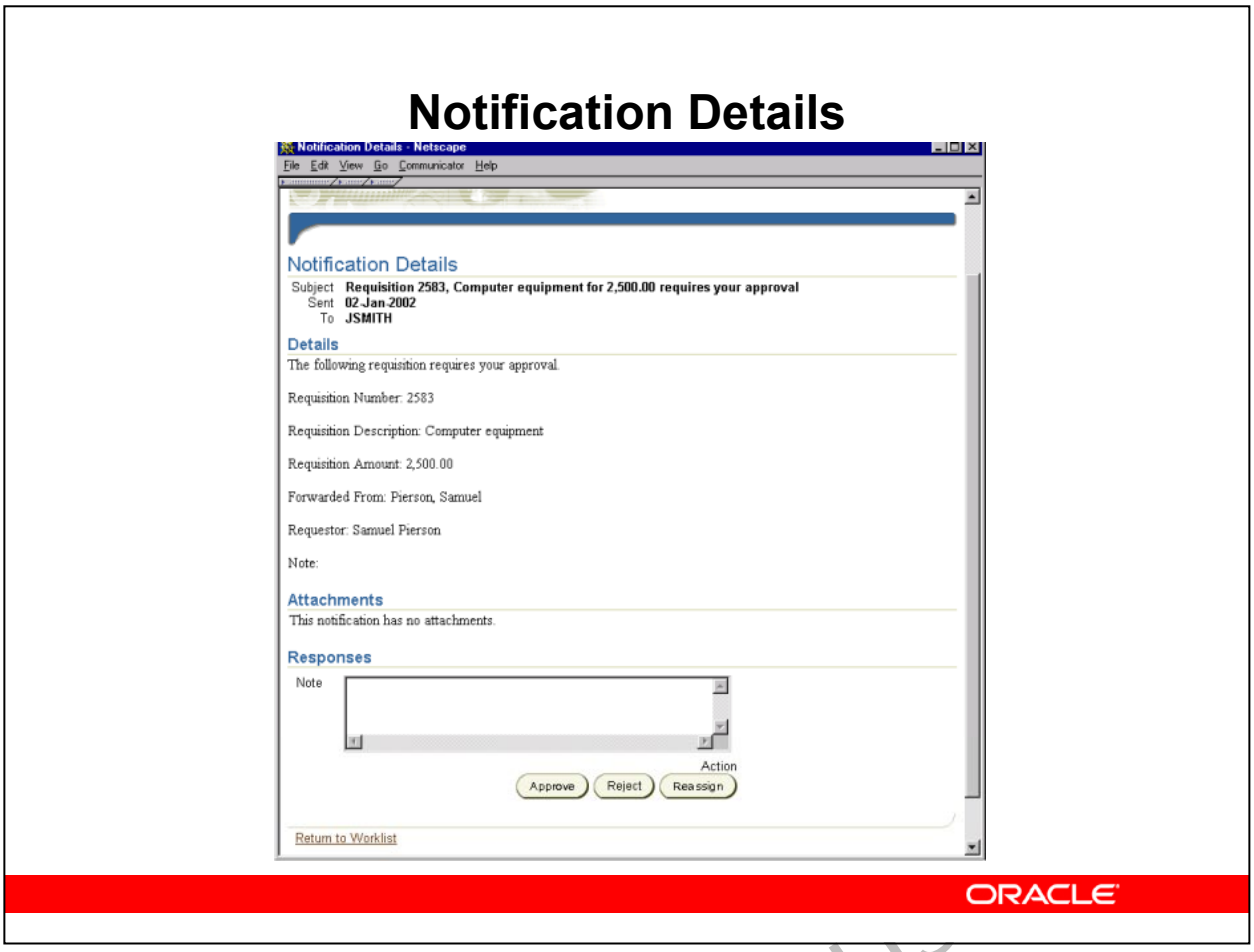


Advanced Worklist

- The Advanced Worklist in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch. Choose Advanced Worklist from a Workflow responsibility to access this page.
- The Advanced Worklist lists all your open notifications, displaying the from role, subject, sent date, due date, and priority for each notification.
- Click the From, Sent, Due, or Priority column heading to sort your notifications by that column in ascending or descending order.
- A View poplist lets you specify the types of notifications to display in the Worklist. You can choose to view:
  - All Open Notifications
  - FYI Notifications
  - All To Do Notifications
  - All Notifications

  After selecting the set of notifications to display, choose Go to display those notifications in the Worklist.

- The Advanced Worklist lets you simultaneously close multiple FYI-type notifications that do not require a response. Check Select for each FYI-type notification you want to close, and then choose Close.
- You can also collectively reassign a group of notifications. Check Select for the notifications you want to reassign, and then choose Reassign. A Reassign page appears that lets you specify to whom and how you wish to reassign the notifications.
- To navigate to the full details of any notification, click the notification's Subject link or select the notification and then choose Open.

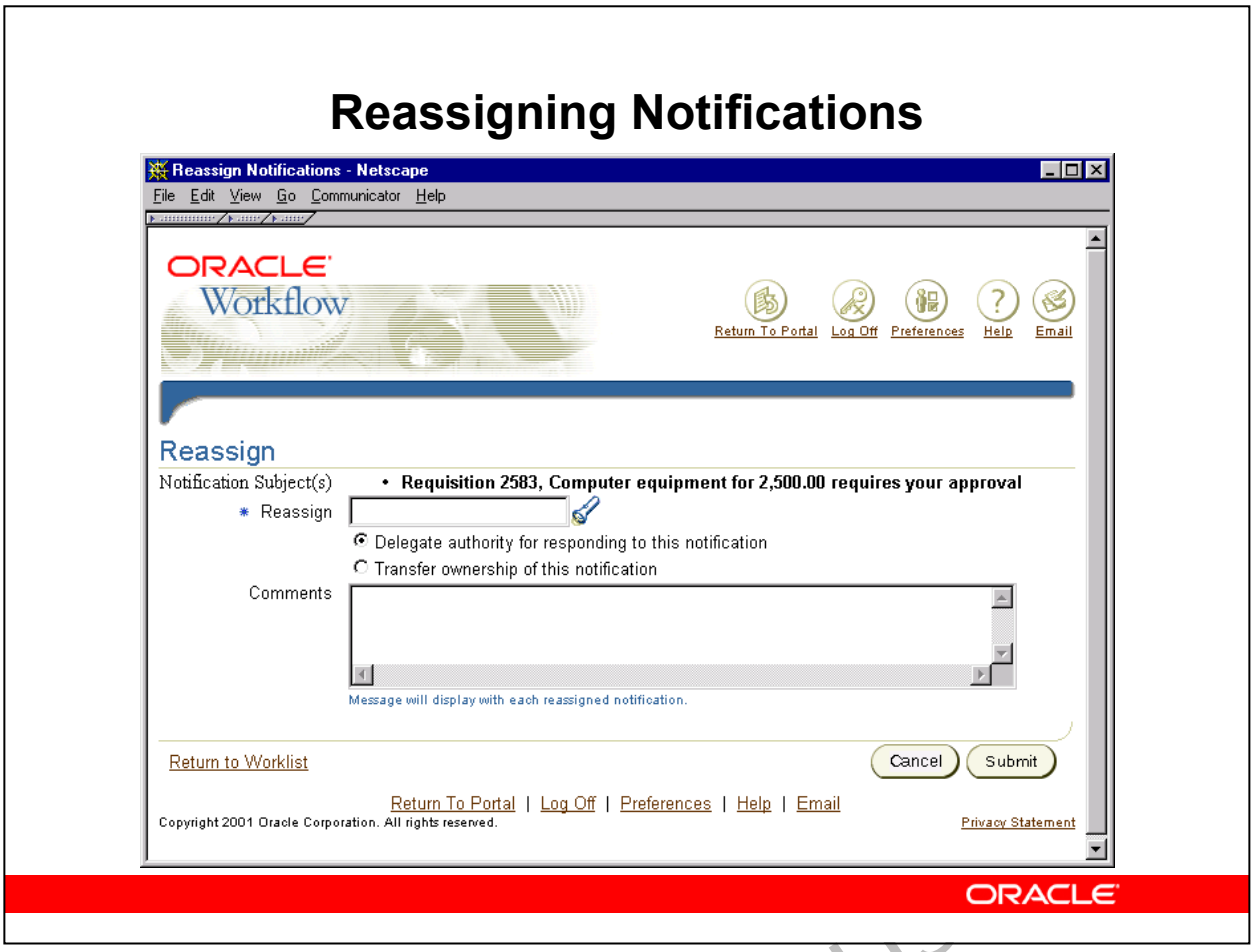


Notification Details

- The Notification Details page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can access the Notification Details page from either the Simple Worklist or the Advanced Worklist.
- The Notification Details page shows the full details of the notification and may also include attachments and a response section.
- The Attachments section may include icons that let you drill down to URL, document, or form attachments.
- The Response section appears as follows:
  - If a notification requires a response, but none of the responses affect the result of the notification activity, the response prompts all appear as fields and/or poplists. When you are done entering your response values, submit your response by choosing the Submit button.
  - If a notification requires a response, and one of the responses becomes the result of the notification activity, then that determining response will appear last as a set of buttons to choose from. The buttons represent the possible choices to the response prompt. All

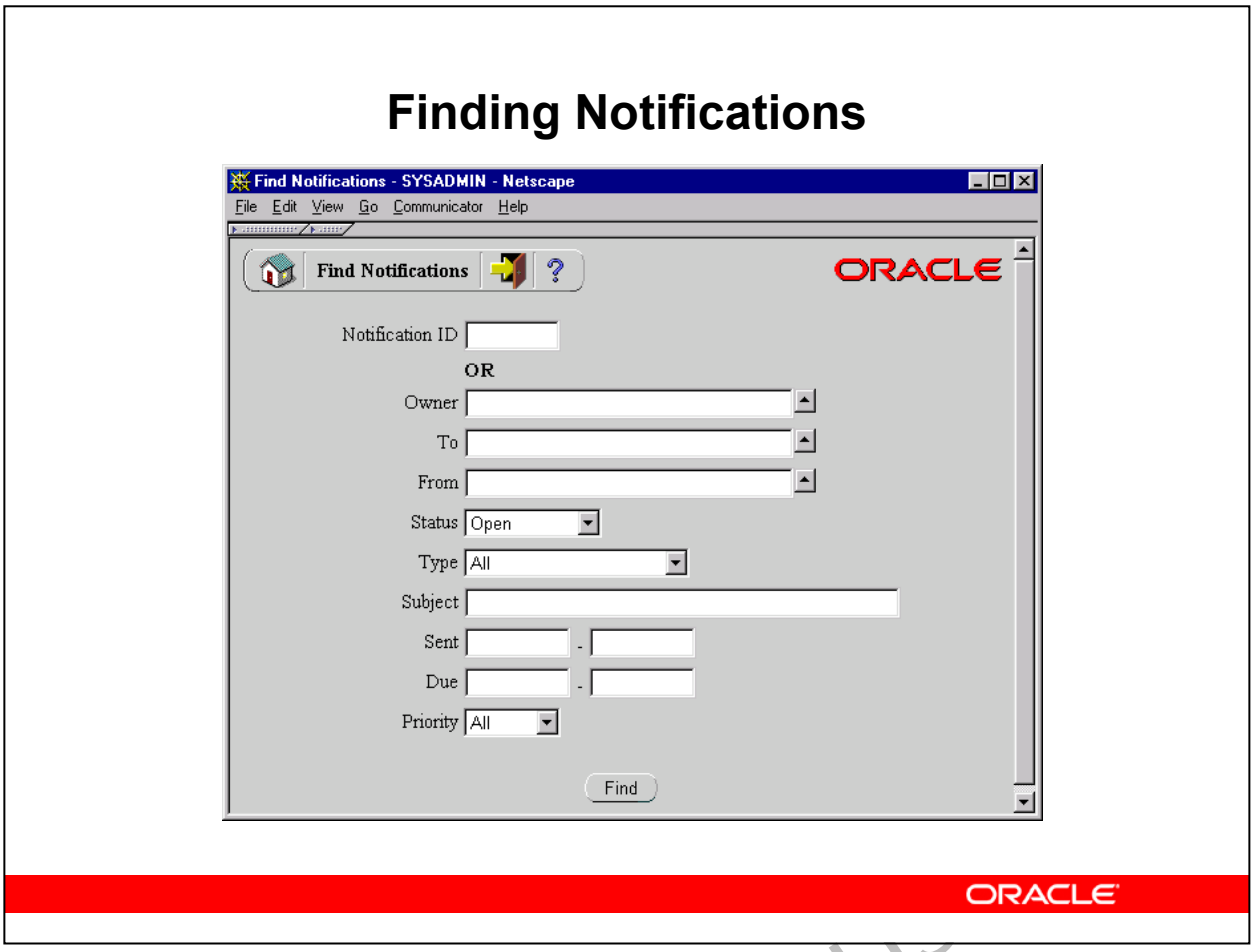
other response prompts, if any, appear as fields or poplists above that prompt. When you choose a button for that last response prompt, you also submit your response for the notification.

- If a notification does not require a response, the response section indicates that. Choose Close in the Response section to close the notification so that it does not appear in your worklist the next time you query for open notifications.
- The Response section may also include links that let you drill down to an Oracle E-Business Suite form to complete your response.



Reassigning Notifications

- The Reassign page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can reassign notifications either by selecting notifications in the Advanced Worklist and choosing Reassign or by viewing a notification in the Notification Details page and choosing Reassign in the response section of the notification. Select the user to whom you want to reassign the notification, the reassign option you want to use, and any optional comments you want to include. Then choose Submit.
- Use the following options to reassign notifications:
  - Delegate authority for responding to this notification: Grant permission to someone else to act on a notification on your behalf while you remain listed as the owner and performer of the notification activity. The delegate function is WF_NOTIFICATION.Forward.
  - Transfer ownership of this notification: Transfer responsibility to act on a notification to someone else and transfer the ownership and performance of the activity to that person. The transfer function is WF_NOTIFICATION.Transfer.



Finding Notifications

To access the Find Notifications web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wfa_html.find

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

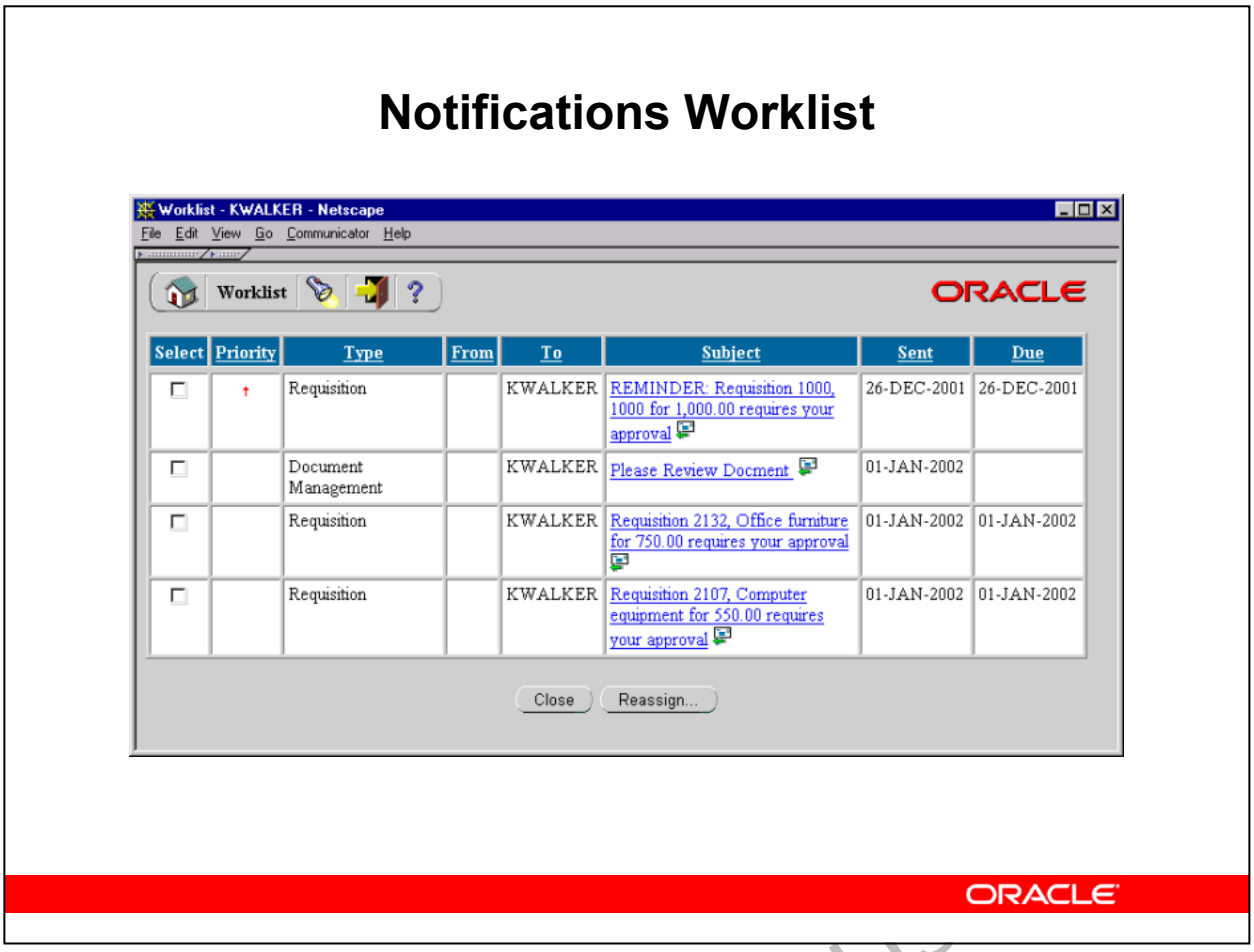
You can also access the Find Notifications web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Notifications option from a Workflow responsibility.

Searching for Notifications

- Use the Find Notifications web page to search for a specific set of notifications.
- Enter search criteria to locate the notifications you want to view.
  - Users who have workflow administrator privileges can search for notifications that belong to any other user or role by specifying an internal role name in the Owner field.

  - Users who do not have workflow administrator privileges can only search for their own notifications.
- Choose Find to locate notifications that meet your specified criteria and display those notifications in the Notifications Worklist web page.



Notifications Worklist

You can access the Notifications Worklist web page either directly or from the Find Notifications web page.

To access the Notifications Worklist web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wfa_html.worklist

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

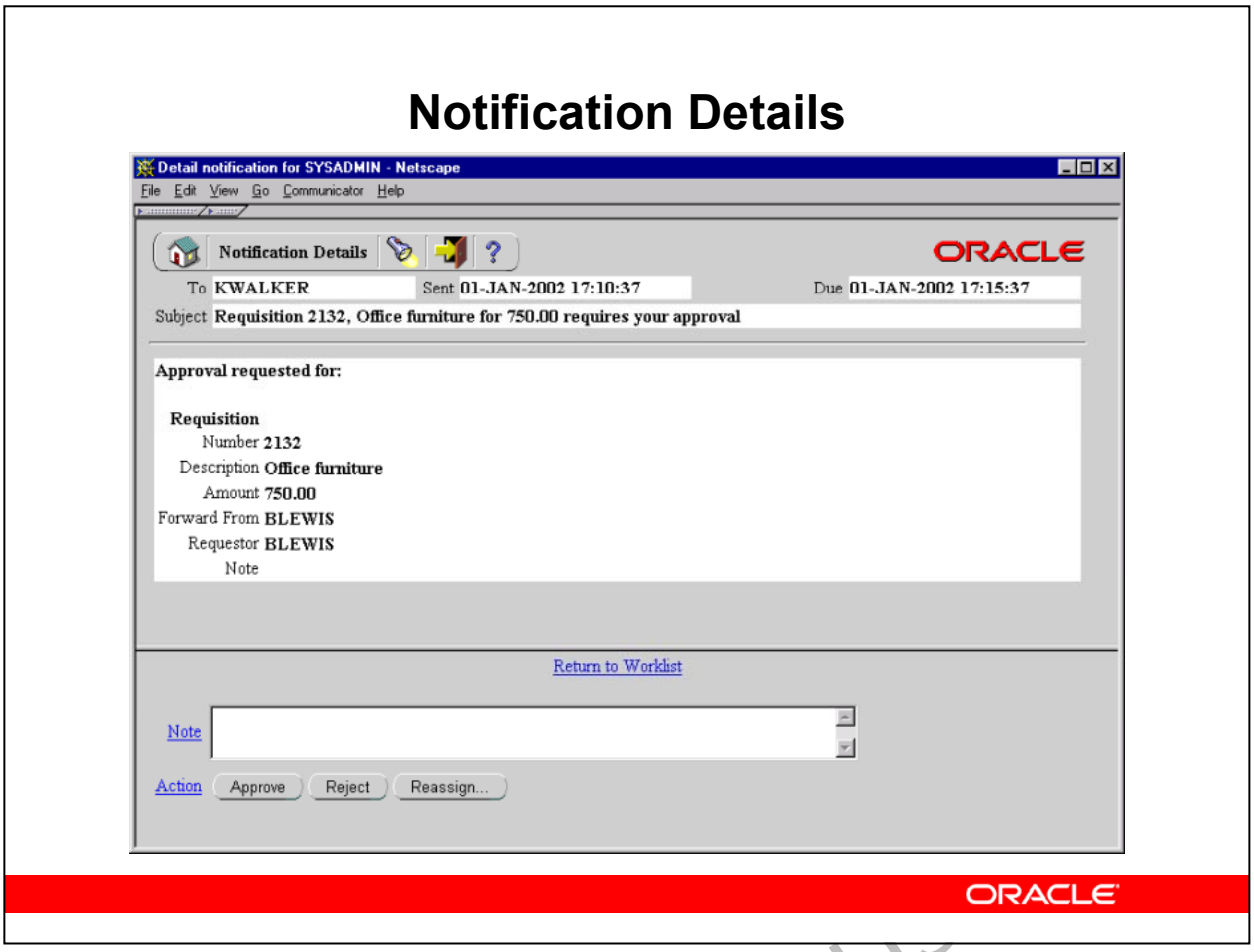
You can also access the Notifications Worklist web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Notifications option from a Workflow responsibility.

Reviewing Notifications in the Notifications Worklist

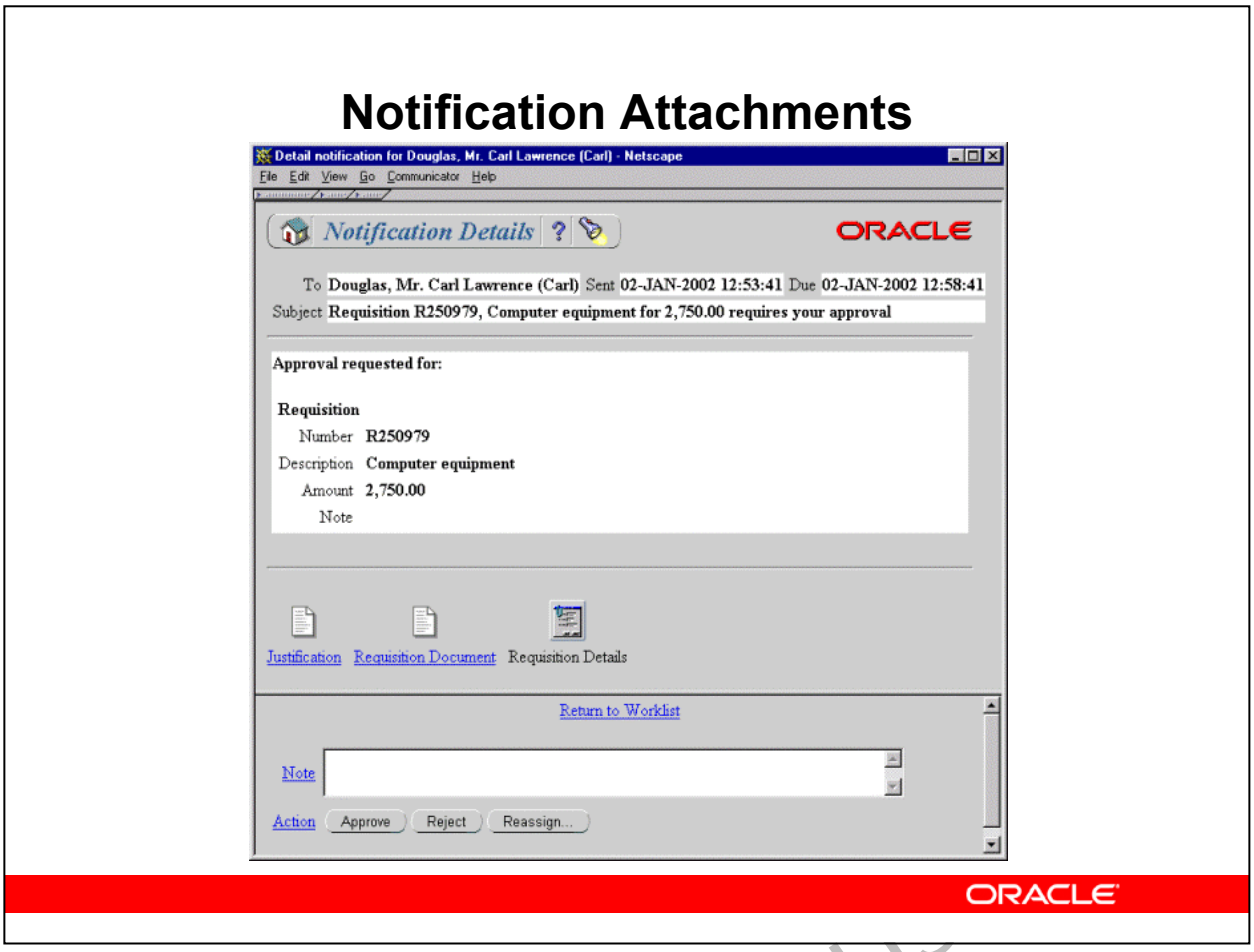
- The Notifications Worklist presents a summary list of notifications.
  - When you access the Notifications Worklist directly, it displays the list of all your open notifications.

  - If you use the Find Notifications page to search for specific notifications, the Notifications Worklist displays only the notifications that match your search criteria.
- Click any column heading to sort the list by that column in ascending order.
- Click a notification subject link to display the details of the notification in the Notification Details web page.
  - In the Notification Details page, you can respond to or reassign a notification. After you respond to or reassign a notification, it no longer appears in your Worklist when you return to the Worklist page.
  - If a notification does not require a response, you can choose Close to remove it from your Worklist.
- You can simultaneously close multiple notifications that do not require a response by selecting those notifications in the Worklist and choosing Close.
- You can collectively reassign a group of notifications by selecting those notifications in the Worklist and choosing Reassign.



Notification Details

The Notification Details web page lets you view details for a single notification and respond to, reassign, or close the notification.



Notification Attachments

Attached URLs

The Notification Details web page supports message attributes of type URL. These attributes appear in a notification message body as a hypertext link or below the message as an attachment icon. When you open your notification, you can click the link or attachment icon to launch the URL in the web browser according to the frame target specified for the attribute.

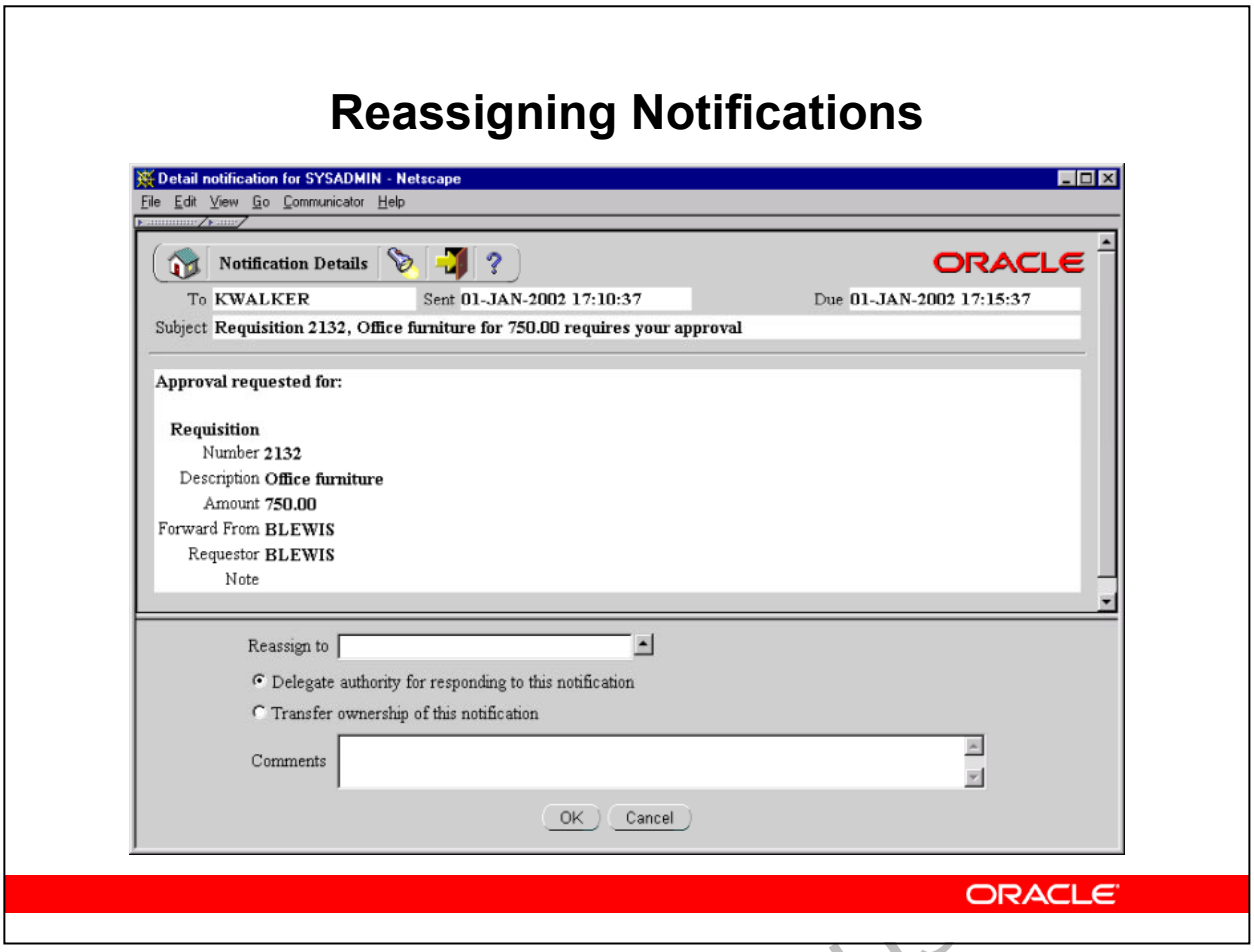
Attached Forms

The Notification Details web page supports message attributes of type form. These attributes appear in a notification message as a form icon. When you open your notification, you can click the attached form icon to launch the referenced Oracle E-Business Suite form.

Note: This attribute type is not relevant for the standalone version of Oracle Workflow.

Attached Documents

The Notification Detail web page supports message attributes of type document. These attributes appear in a notification message as an inline link or as an attachment icon. When you open your notification, you can click a link or an attachment icon to open the referenced document.



Reassigning Notifications

You can reassign notifications either by selecting notifications in the Notifications Worklist and choosing Reassign or by viewing a notification in the Notification Details page and choosing Reassign in the response section of the notification. Select the user to whom you want to reassign the notification, the reassign option you want to use, and any optional comments you want to include. Then choose OK.

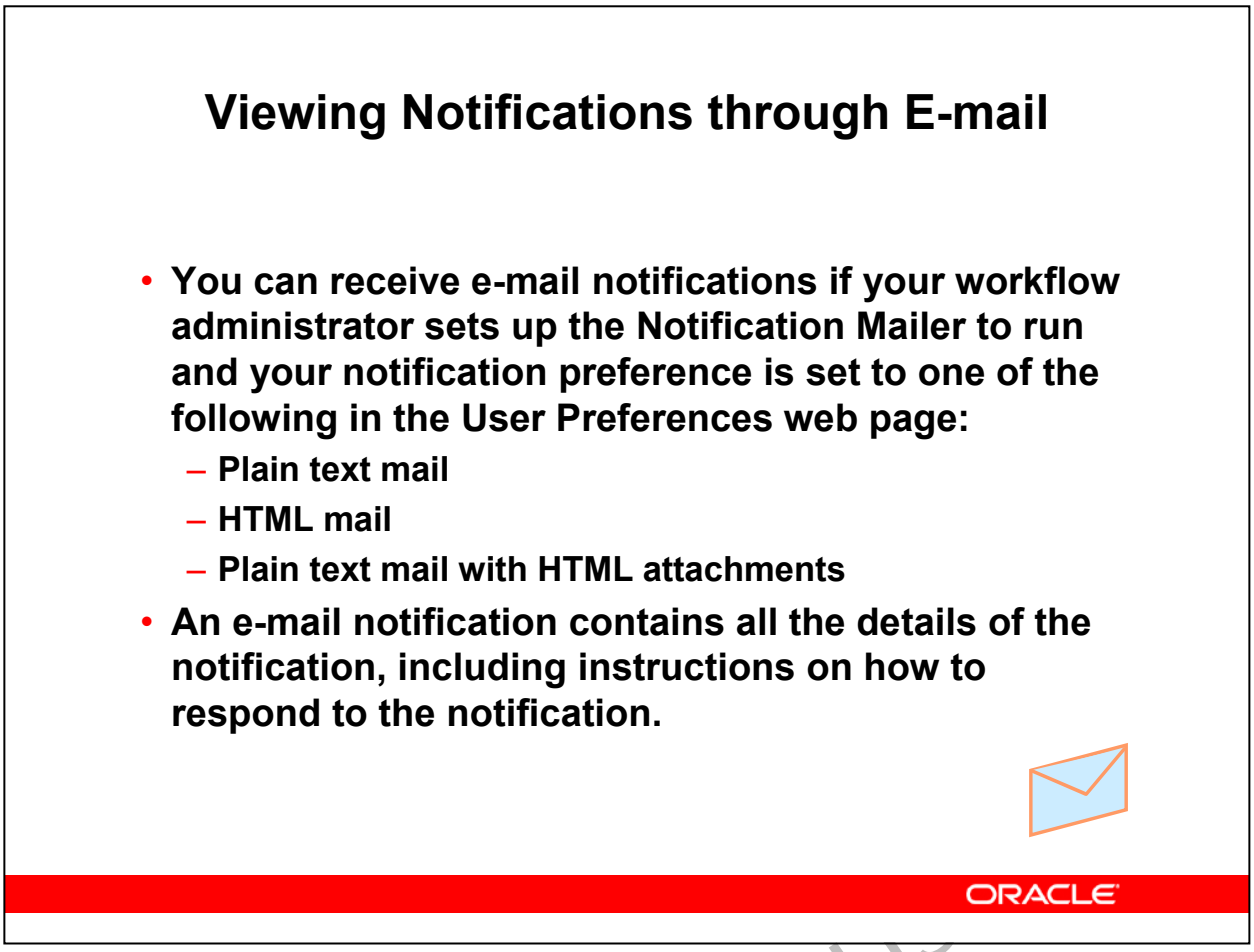
Use the following options to reassign notifications:

- Delegate: Grant permission to someone else to act on a notification on your behalf while you remain listed as the owner and performer of the notification activity. The delegate function is WF_NOTIFICATION.Forward.
- Transfer: Transfer responsibility to act on a notification to someone else and transfer the ownership and performance of the activity to that person. The transfer function is WF_NOTIFICATION.Transfer.

The following Oracle Workflow areas use these notification reassignment values:

- The Workflow Monitor Diagram web page
- The Notifications web pages

If you view your notifications through e-mail, you can only reassign a notification by using the Forward function, which performs similarly to the Delegate option.



Viewing Notifications through E-mail

- Plain text mail: The notification message appears as plain text, with no attachments. If the message requires a response, you can respond by replying to the e-mail message.
- HTML mail: The notification message appears as HTML–formatted text, with at least one other attachment that is a link to the notification in the Notification Details web page. If the notification message has 'Content–Attached' message attributes, these attributes appear as additional attachments to the message. If the message requires a response, you can respond by clicking one of the response links or you can navigate to the Notification Details page and respond there.
- Plain text mail with HTML attachments: The notification message appears as plain text, with at least two other attachments. One attachment is an HTML–formatted version of the message, and the other is a link to the notification in the Notifications web page. If the notification message has 'Content–Attached' message attributes, these attributes appear as additional attachments to the message. If the message requires a response, you can respond by replying to the e-mail message or by opening the HTML-formatted version of the message and clicking one of the response links, or you can navigate to the Notification Details page and respond there.

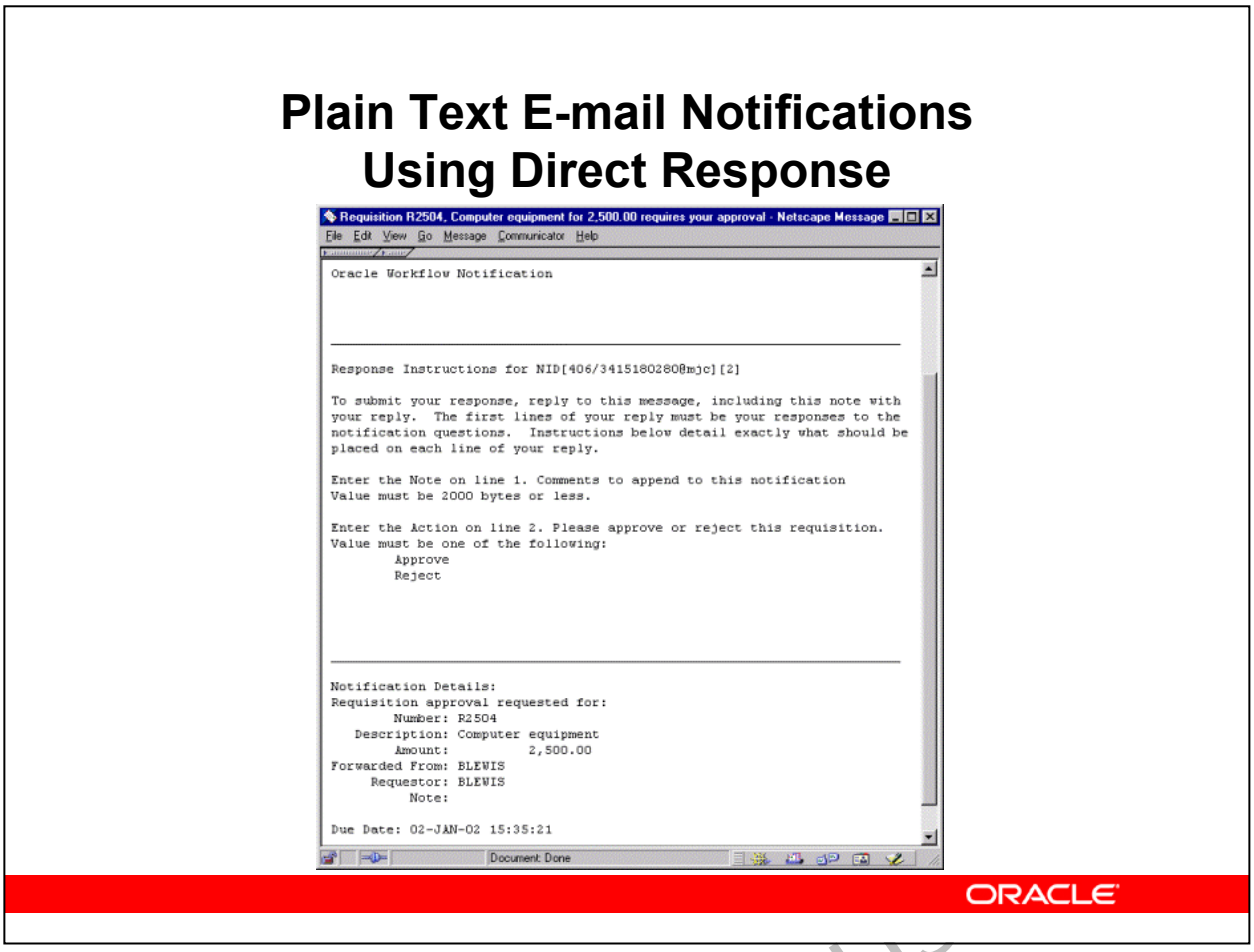
E-mail Response Methods

- **Your workflow administrator determines the response method for plain text e-mail notifications when setting up the Notification Mailer.**
  - **Direct response: Enter your response values directly as the first lines of your reply.**
  - **Templated response: Reply using the template of response prompts provided in the notification and enter your response values between the quotes following each prompt.**
- **HTML-formatted notifications always use the templated response method.**

ORACLE

E-mail Response Methods

The Notification System interprets your response values literally, so uppercase values are distinct from lowercase values.



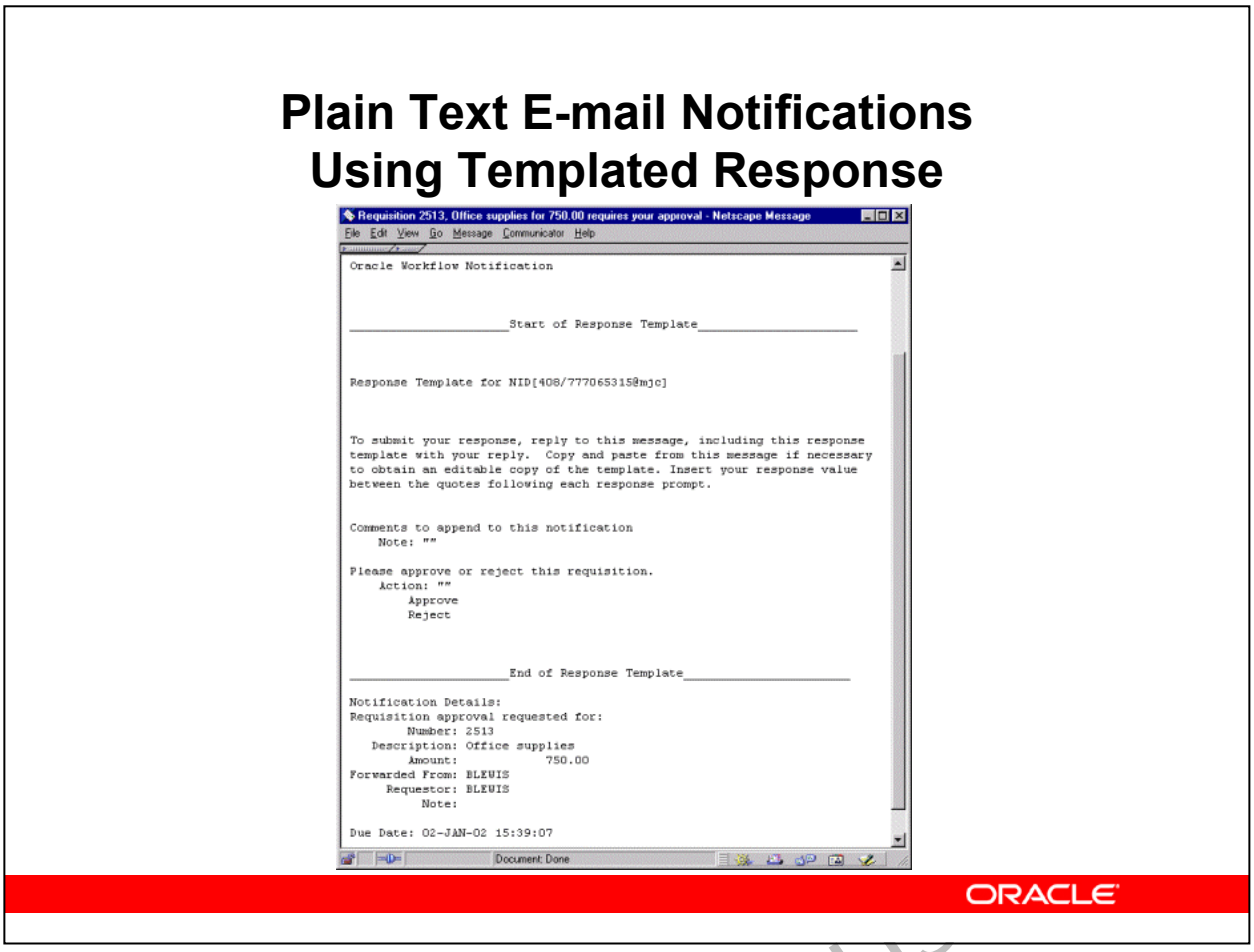
Plain Text E-mail Notifications Using Direct Response

Include the text of the original notification in your reply to ensure that you include the special notification ID and access key that the Notification Mailer requires to identify the notification you are responding to. The first lines of your reply are interpreted as your notification response, where each line represents a separate response value listed in the same order as its corresponding response prompt. If a response value requires more than one line, enclose the entire response value in double quotes. If a response prompt provides a default value, then you can accept the default value by leaving the appropriate response line blank.

Attachments

In a plain text e-mail notification, the URL location value for an attached URL attribute is included as plain text. PL/SQL or PL/SQL CLOB document attributes are attached as plain text to their parent notification. Note that this may render some attachments unreadable if the attachment includes special formatting or your plain text e-mail reader does not recognize attachments. To view these attachments, you should display your notifications in the Notification Details web page.

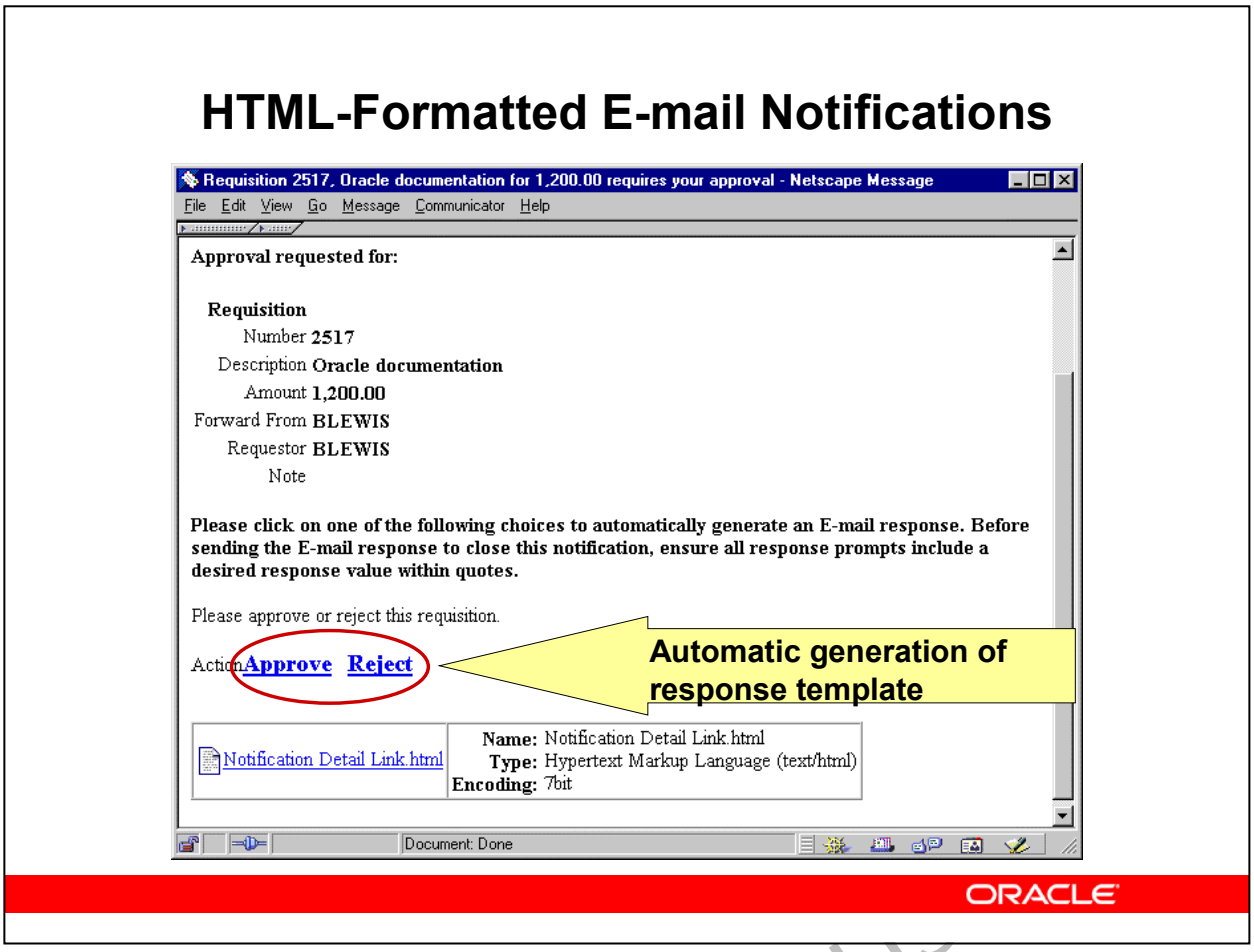
Note: Form attributes cannot be attached to e-mail notifications. To view form attachments, you must view the notification in the Notification Details web page.



Plain Text E-mail Notifications Using Templated Response

Include the response template from the original notification in your reply. In addition to the response prompts, the response template includes the special notification ID and access key that the Notification Mailer requires to identify the notification you are responding to. If your mail application includes an editable copy of the original message when it generates the reply message, you can use that copy to enter your response values. Otherwise, copy and paste from the original message to obtain a copy of the response template that you can edit.

Follow the response template instructions and insert your response values between the quotes (“ ”) following each response prompt.



HTML-Formatted E-mail Notifications

HTML e-mail notifications always use the templated response method.

The response section of an HTML e-mail notification includes links for the possible result response values at the end of the notification. After reviewing the message body of the notification, click one of the response links.

Each response link automatically generates a plain text e-mail reply that contains the correct Reply To: e-mail address as well as a response template in the message body. The response template consists of the required notification ID and access key that identify the notification you are responding to and a response prompt edited with your selected result response. Depending on the notification, the auto-generated e-mail response template may also prompt you for other information in addition to your selected result response.

An HTML-formatted notification always includes at least one attachment. The attachment is called Notification Detail Link. When you select this attachment, your e-mail reader opens a browser window that displays your notification in the Notification Details web page. You can respond directly to your notification from this web page, bypassing the need to process your response through the Notification Mailer.

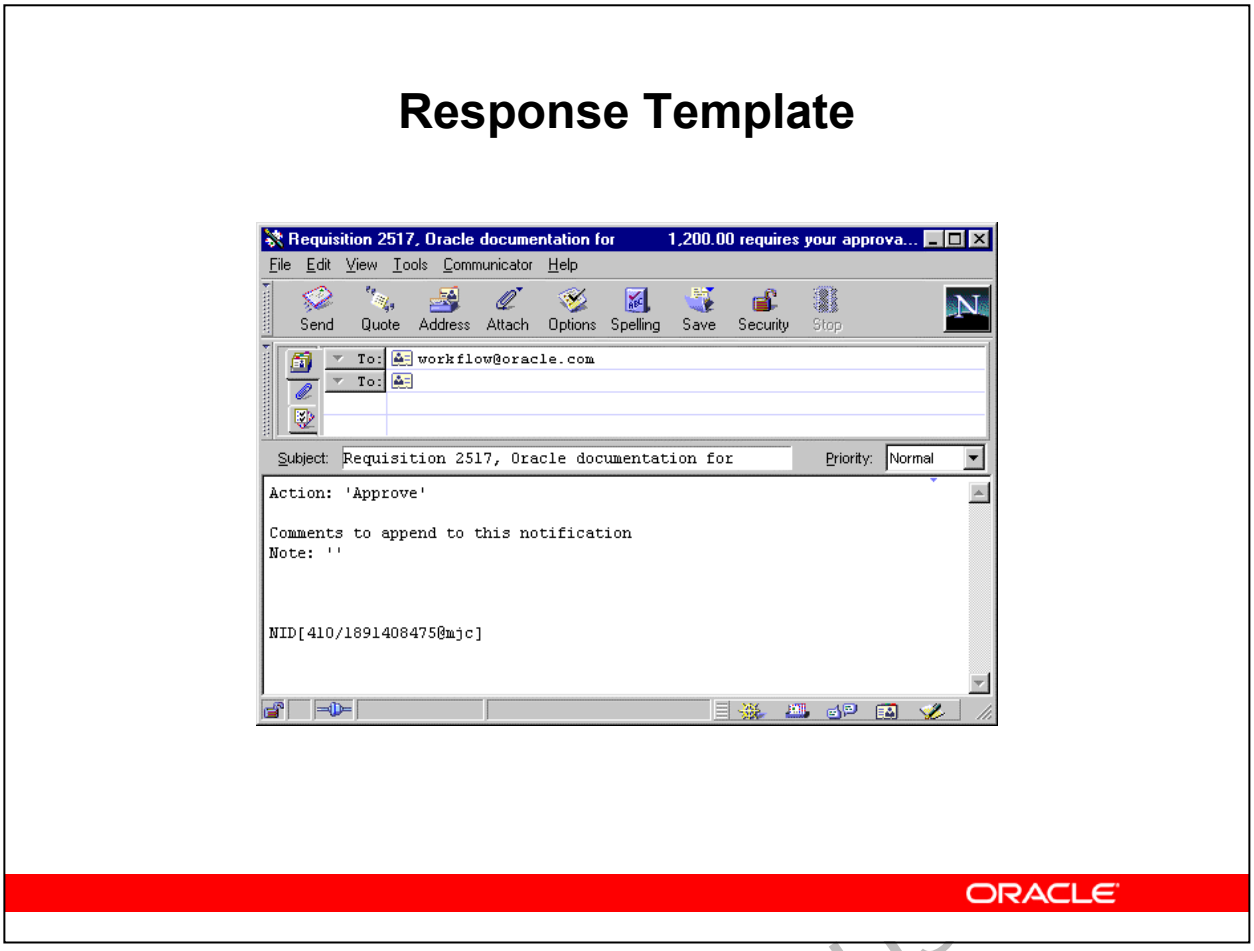
Note: If your system administrator has set the SEND_ACCESS_KEY parameter in the Notification Mailer configuration file to N, and you are not already logged in, you will be prompted to log in when you select the Notification Detail Link before you can access the Notification Details web page.

Attachments

In an HTML-formatted e-mail notification, a URL attribute appears as a hypertext link in the message body. For attached URL attributes, an attachment called Notification References is appended to the message. This attachment includes a link to each attached URL attribute for the message. You can navigate to a URL by choosing its link.

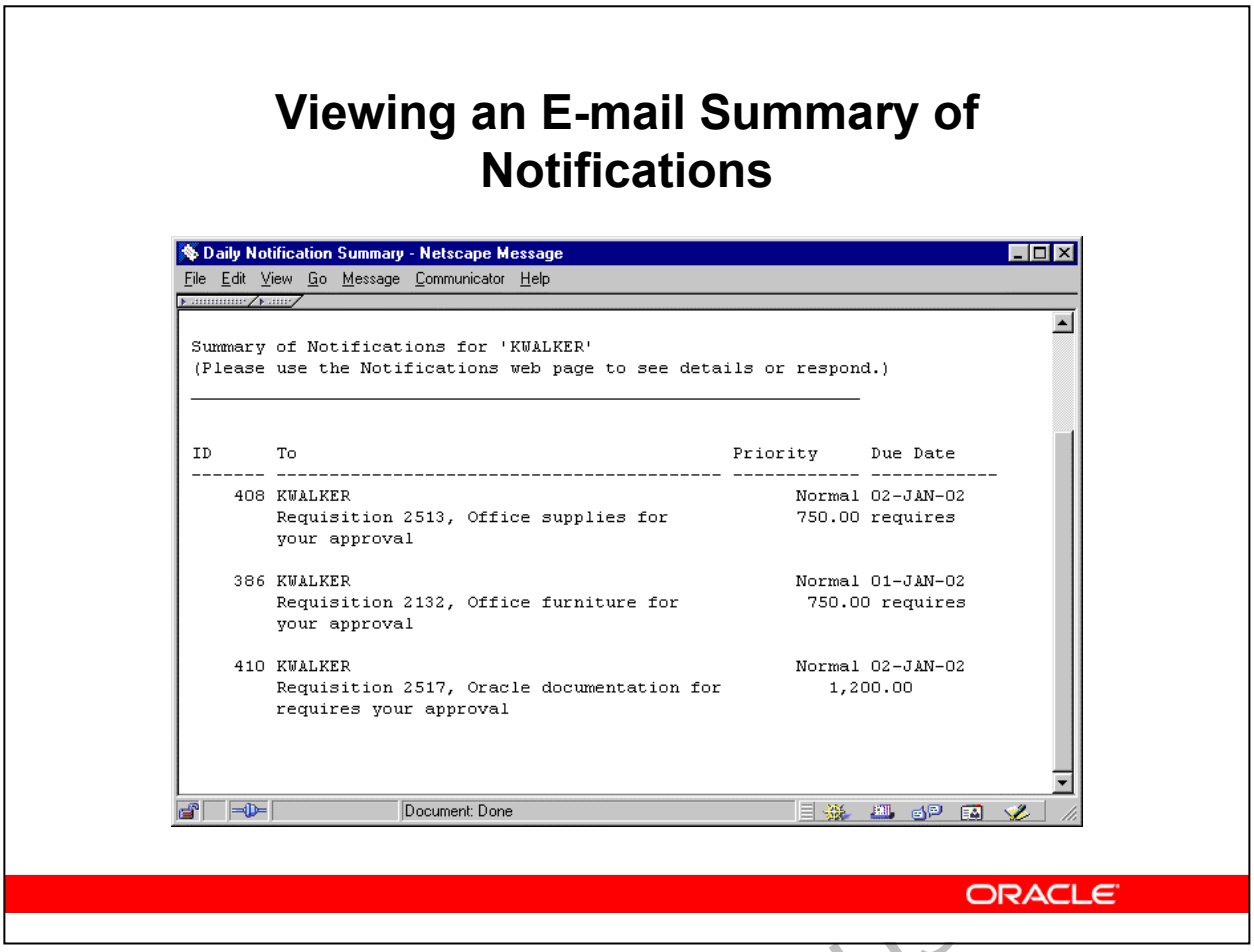
If an HTML-formatted e-mail notification has an attached PL/SQL or PL/SQL CLOB document attribute, the fully generated PL/SQL document is fetched and attached to the message.

Note: Form attributes cannot be attached to e-mail notifications. To view form attachments, you must view the notification in the Notification Details web page.



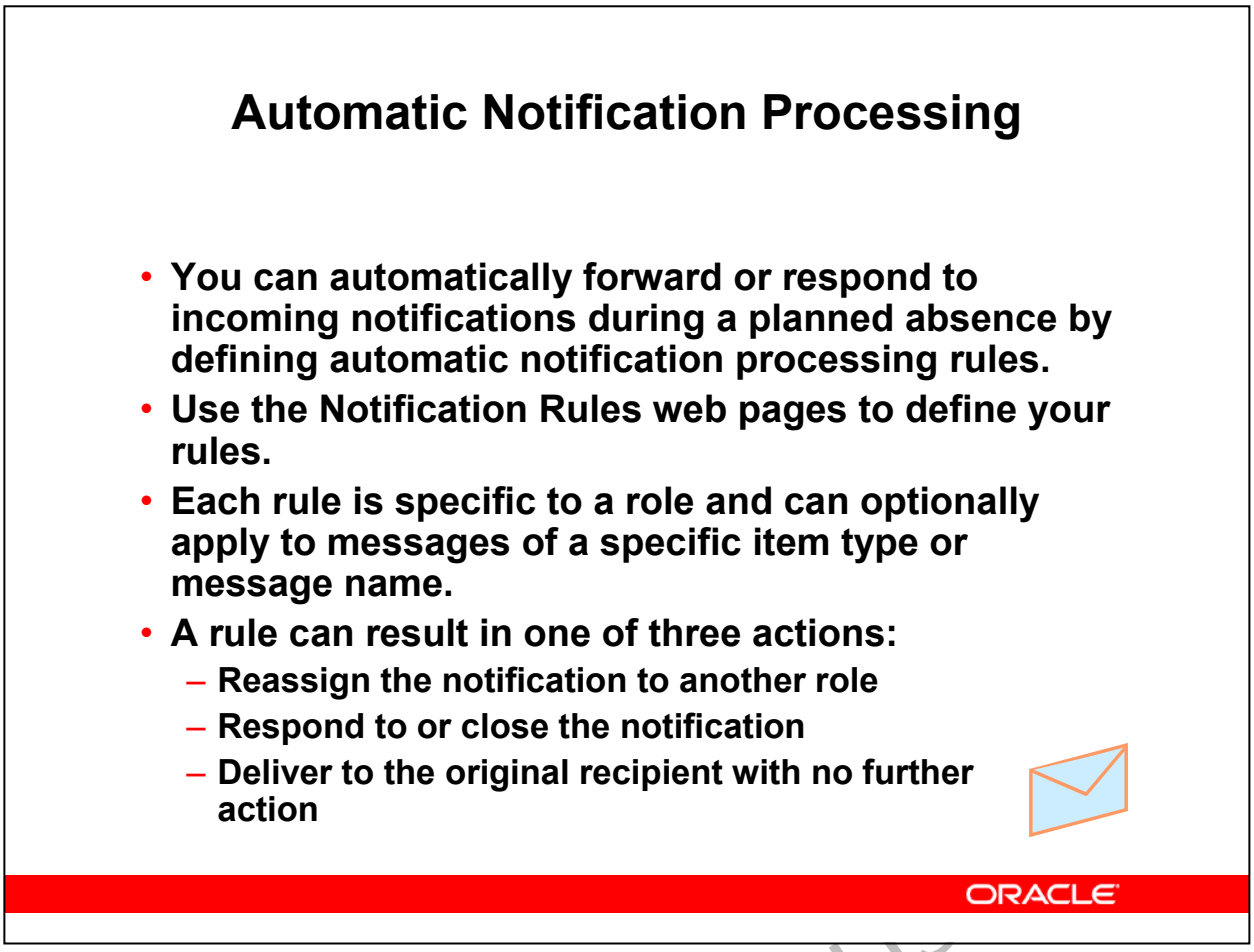
Response Template

When responding to a notification with an automatically generated template, supply your responses by editing the response value text between the quotes following each prompt.



Viewing an E-mail Summary of Notifications

- You can receive an e-mail summary of notifications if your workflow administrator sets up the Notification Mailer to run and your notification preference is set to 'Plain text summary mail' in the User Preferences web page.
- The Notification Mailer delivers a single e-mail message summarizing your current list of open notifications.
- To respond to individual notifications listed in the summary, you must use the Notification Details web page.
- How often you receive e-mail summaries depends on how frequently the Notification Mailer that handles e-mail summaries is scheduled to run.

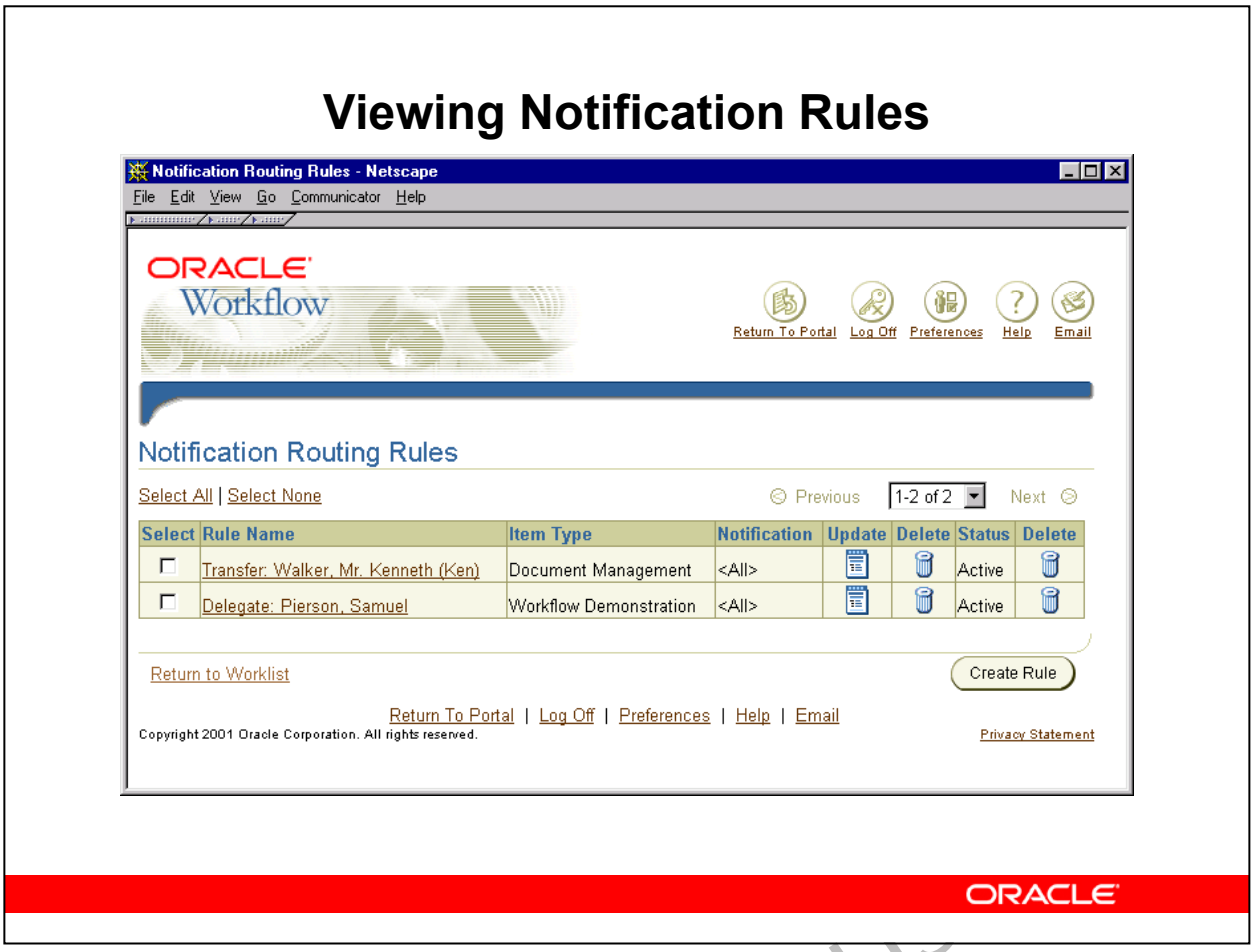


Automatic Notification Processing

- When a notification is sent or reassigned to a role, Oracle Workflow tests the notification against that role's list of automatic processing rules for the most specific match based on the following order of criteria:

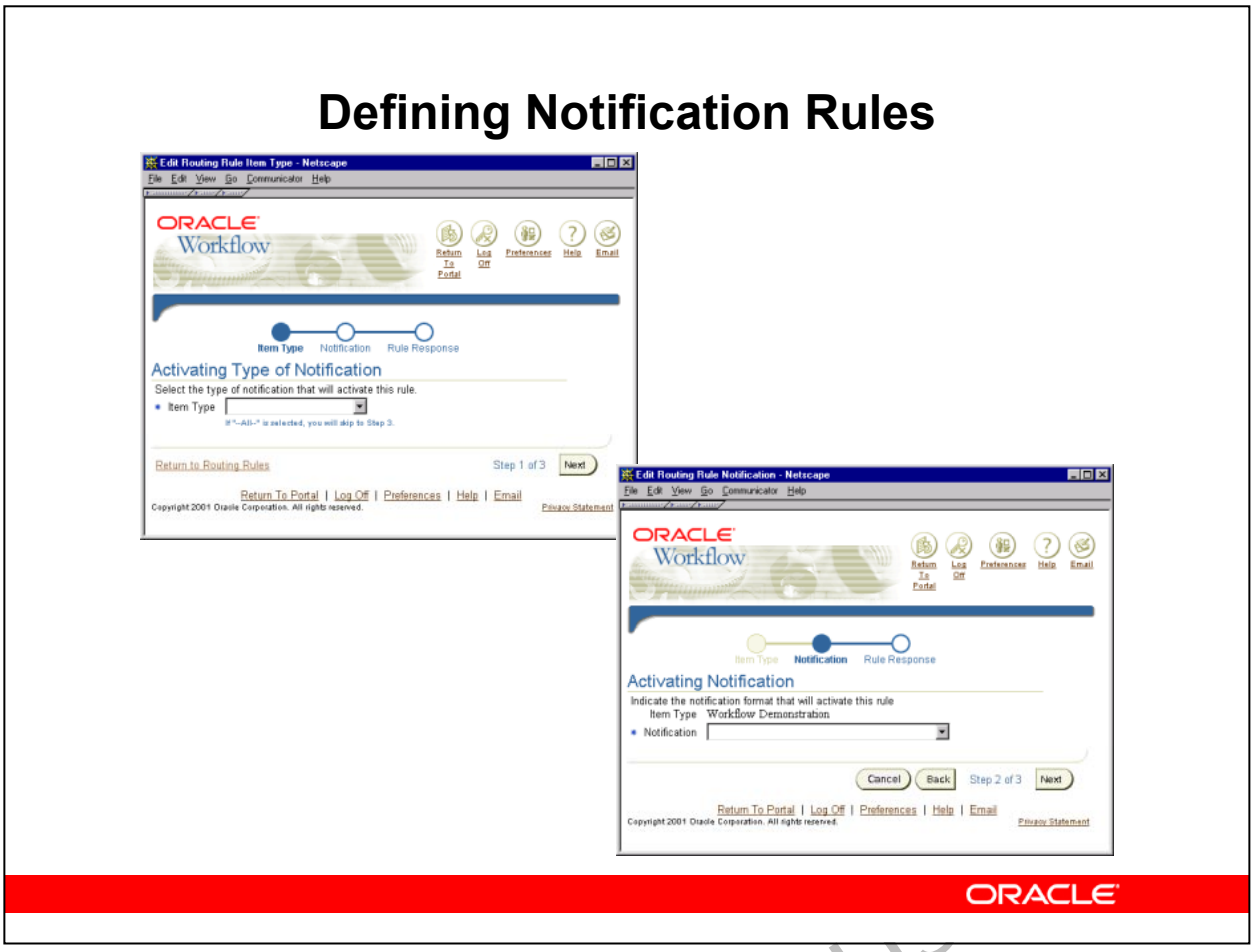
  ROLE = <role> and:
  1. ITEM_TYPE = <*type*> and MESSAGE_NAME = <*name*>
  2. ITEM_TYPE = <*type*> and MESSAGE_NAME is null
  3. ITEM_TYPE is null and MESSAGE_NAME is null

- If a rule reassigns a notification, Oracle Workflow again performs rule matching against the new recipient role's list of rules.
- A count of the number of times that a notification is forwarded helps detect perpetual forwarding cycles. If a notification is automatically forwarded more than ten times, Oracle Workflow assumes a forwarding cycle has occurred, ceases all further automatic forwarding, and marks the notification as being in error.



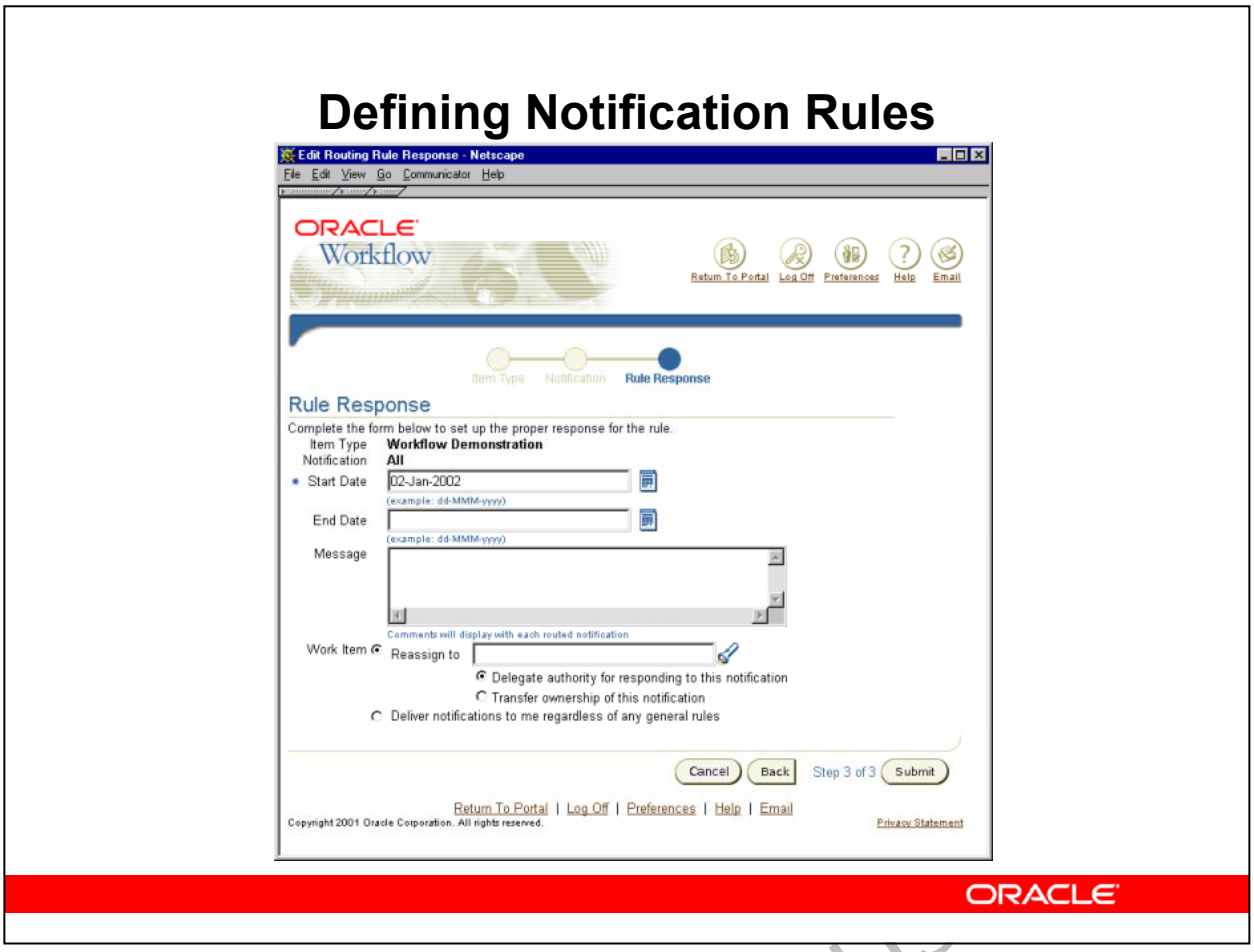
Viewing Notification Rules

- The Notification Routing Rules page in the new web page format is available for Oracle Workflow embedded in Oracle E-Business Suite if your installation includes the corresponding patch.
- You can access the Notification Routing Rules page by clicking the routing rules link in the tip at the bottom of either the Simple Worklist or the Advanced Worklist.
- The Notification Routing Rules page displays a list of the existing rules for the current role. The following information is listed for each rule:
  - Rule name describing the action the rule performs
  - Item type
  - Notification name, or <ALL> for all notifications within that item type
  - Active status
- Click the rule link in the Rule Name column or the edit icon in the Update column to edit a rule.
- Click the delete icon in the Delete column to delete a rule.
- Choose the Create Rule button to define a new rule.



Defining New Notification Rules

1. Choose Create Rule in the Notification Routing Rules page.
2. In the Activating Type of Notification page, specify the item type to which you want the rule to apply, and choose Next.
   - Rules apply to notification messages, and all messages are associated with a specific item type.
   - Choose <ALL> if you want the rule to apply to all item types.
3. In the Activating Notification page, specify the name of the notification message to which you want the rule to apply, and choose Next.
   - Messages are listed shown by subject name.
   - Choose <ALL> if you want the rule to apply to all messages.
   - Skip this step if you selected <ALL> as the item type.



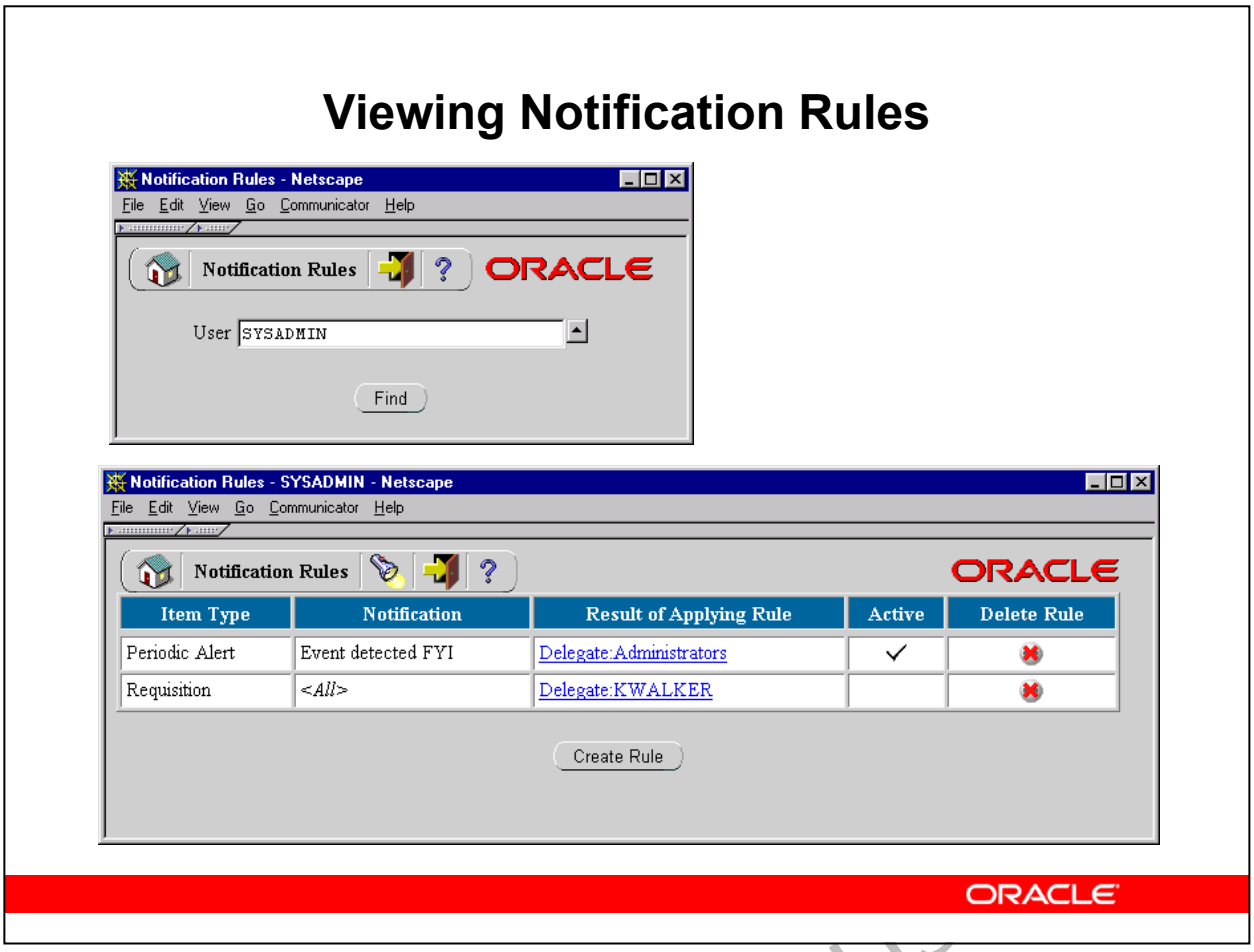
Defining New Notification Rules

4. In the Rule Response page, specify the period during which you want the rule to be active by entering a start and end date and time.
   - Use the default date format of your database.
   - Use the time format HH24:MI:SS.
   - A blank Start Date means the rule is effective immediately.
   - A blank End Date means the rule is effective indefinitely.

   Note: Because you can define multiple rules for the same notifications, make sure your rules for the same notifications do not overlap in their effective dates.
5. Use the Message field to specify text that you want to append to the notification in the "Prior comments" field when the rule is applied.
6. Select the action for the rule.
   - Reassign to: Forward the notification to the specified role. In this case you must also specify a role and select either 'Delegate Authority for Responding to Notifications' or 'Transfer Ownership of Notifications'.
   - Respond: Respond to the message with a set of predefined response values.

- Deliver Notifications to me, regardless of any general rules: Leave the notification in your inbox and do nothing. Use this action to exclude a subset of notifications from a more encompassing rule.

7. Click the Submit button to save the rule. You can also choose Cancel if you do not want to save the rule.



Viewing Notification Rules

To access the Notification Rules web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_route.find

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Notification Rules web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Notification Rules option from a Workflow responsibility.

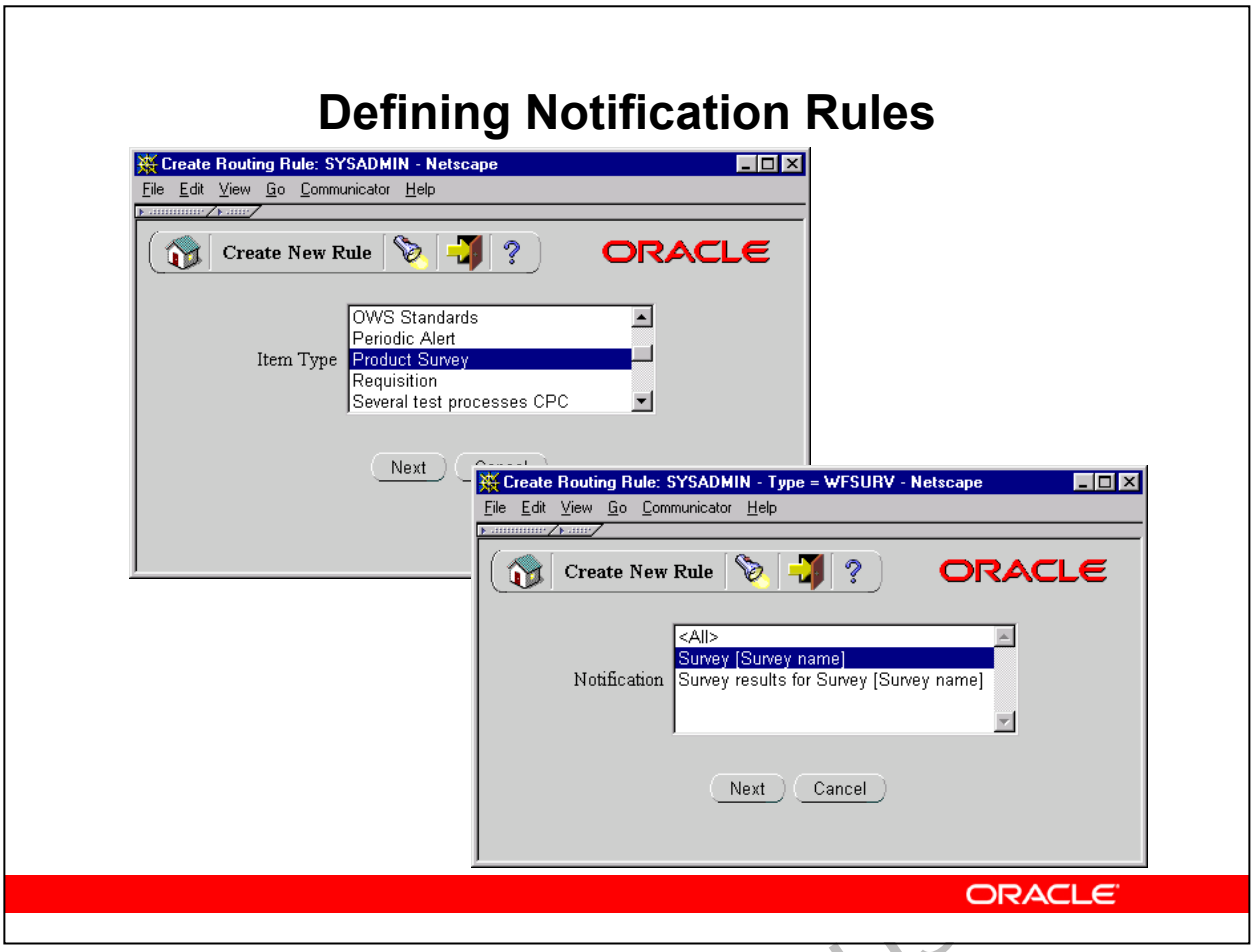
In the Notification Rules search page, enter a role name in the User field and choose Find to display the list of rules for that role. You can view rules for roles other than your own only if you have workflow administrator privileges.

To access the Notification Rules list page directly for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/wf_route.list

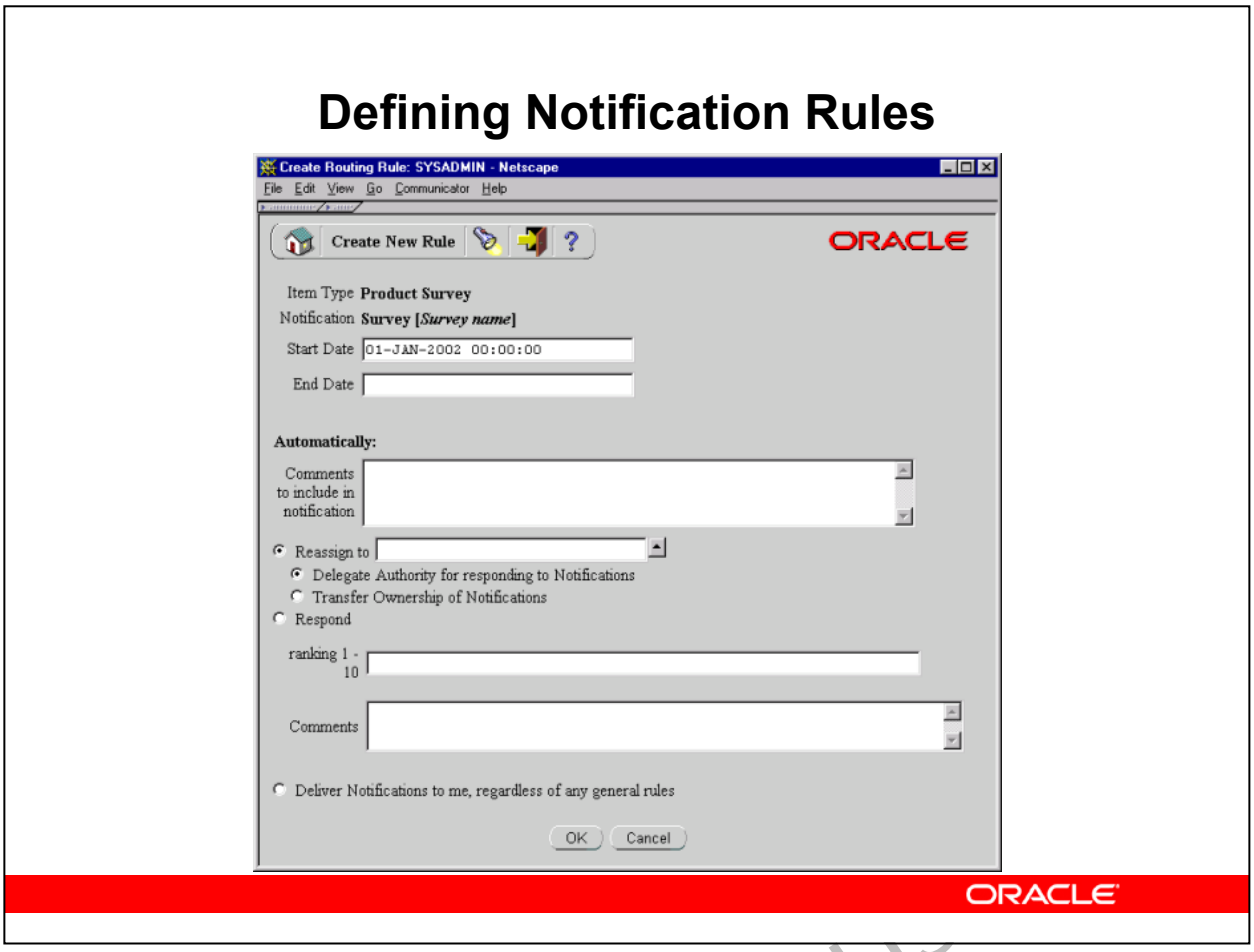
Notification Rules List

- When a notification is sent or reassigned to a role, Oracle Workflow tests the notification against that role's list of automatic processing rules for the most specific match based on the following order of criteria:
- The Notification Rules list page displays the existing rules for the role. The following information is listed for each rule:
  - Item type
  - Notification name, or <ALL> for all notifications within that item type
  - Result of applying the rule
  - Active status
- Click the action link in the "Result of Applying Rule" column to edit a rule.
- Choose the delete icon in the Delete Rule column to delete a rule.
- Choose the Create Rule button to define a new rule.



Defining New Notification Rules

1. Choose Create Rule in the Notification Rules list page.
2. Specify the item type to which you want the rule to apply, and choose Next.
   - Rules apply to notification messages, and all messages are associated with a specific item type.
   - Choose <ALL> if you want the rule to apply to all item types.
3. Specify the name of the notification message to which you want the rule to apply, and choose Next.
   - Messages are listed shown by subject name.
   - Choose <ALL> if you want the rule to apply to all messages.
   - Skip this step if you selected <ALL> as the item type.



Defining New Notification Rules

4. Specify the period during which you want the rule to be active by entering a start and end date and time.
   - Use the default date format of your database.
   - Use the time format HH24:MI:SS.
   - A blank Start Date means the rule is effective immediately.
   - A blank End Date means the rule is effective indefinitely.

   Note: Because you can define multiple rules for the same notifications, make sure your rules for the same notifications do not overlap in their effective dates.
5. Use the "Comments to include in notification" field to specify text that you want to append to the notification in the "Prior comments" field when the rule is applied.
6. Select the action for the rule.
   - Reassign to: Forward the notification to the specified role. In this case you must also specify a role and select either 'Delegate Authority for Responding to Notifications' or 'Transfer Ownership of Notifications'.
   - Respond: Respond to the message with a set of predefined response values.

- Deliver Notifications to me, regardless of any general rules: Leave the notification in your inbox and do nothing. Use this action to exclude a subset of notifications from a more encompassing rule.

7. Click the OK button to save the rule. You can also choose Cancel if you do not want to save the rule.

Summary

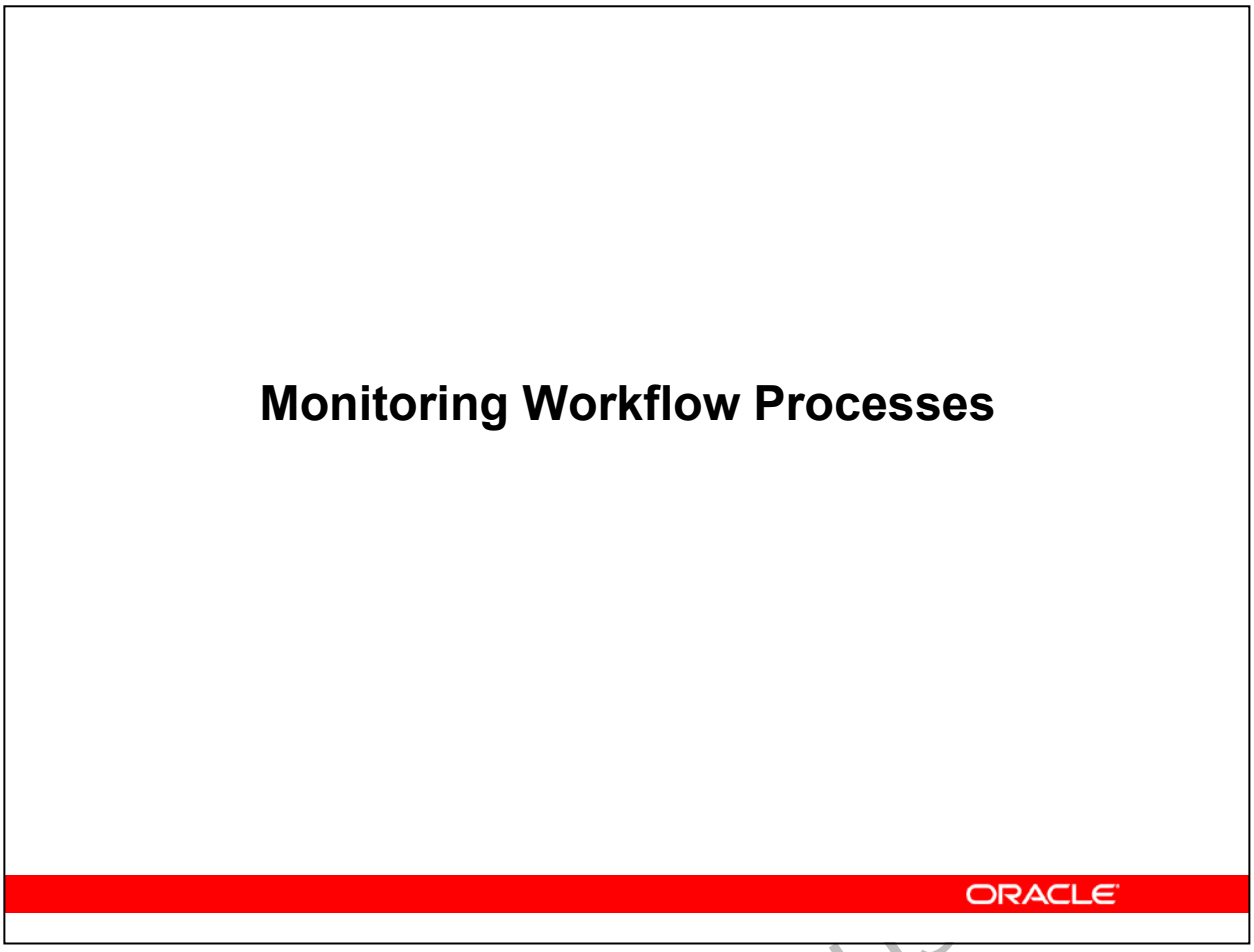
In this lesson, you should have learned how to:

- **Access the Oracle Workflow web pages.**
- **Respond to notifications from the Notifications Worklist web page.**
- **Respond to notifications using e-mail.**
- **Define notification routing rules.**

ORACLE

Monitoring Workflow Processes

Chapter 11

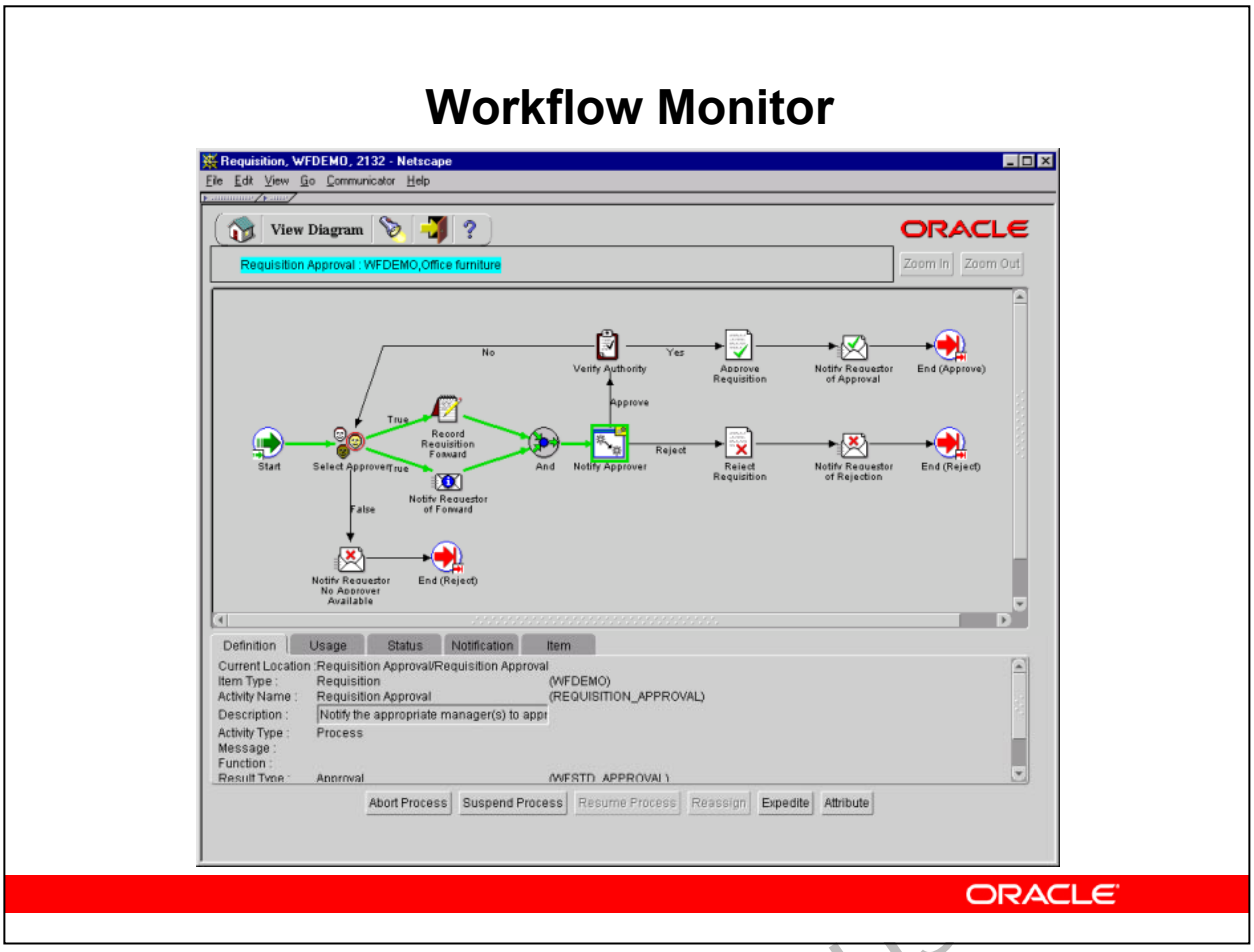


Objectives

After completing this lesson, you should be able to do the following:

- **Check the progress of a workflow using the Workflow Monitor.**
- **Launch a test process.**

ORACLE



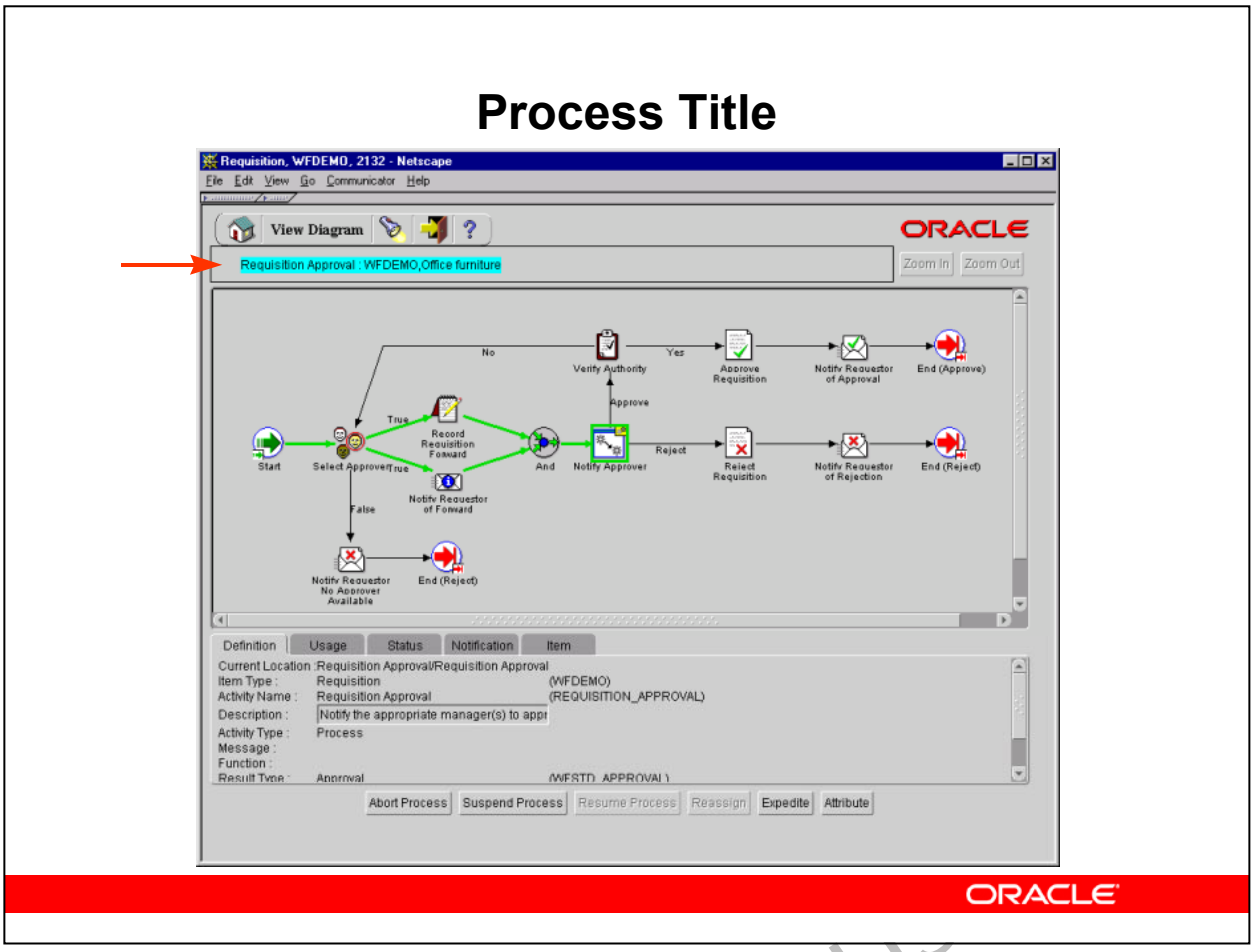
Workflow Monitor

The Workflow Monitor is a Java-based tool that lets you view and administer the progress of a workflow process instance. Workflow Monitor features include:

- A point-and-click interface
- A display for detailed status information for individual activities as well as for the whole process
- The ability to run in USER mode or in ADMIN mode, which provides additional details and functionality pertinent only to a workflow administrator

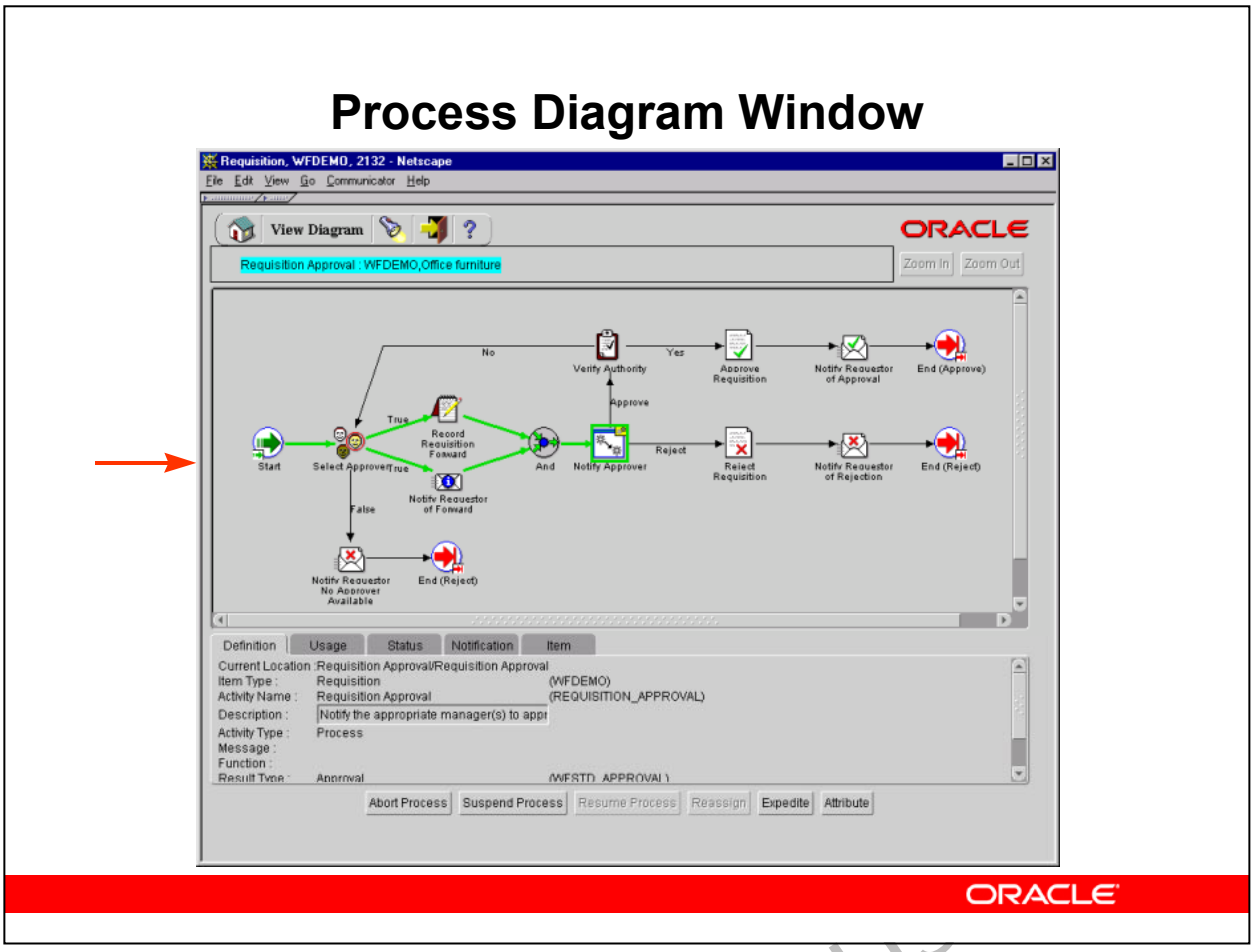
User interface components include:

- Process title
- Process diagram window
- Detail tab window
- Administration buttons



Process Title

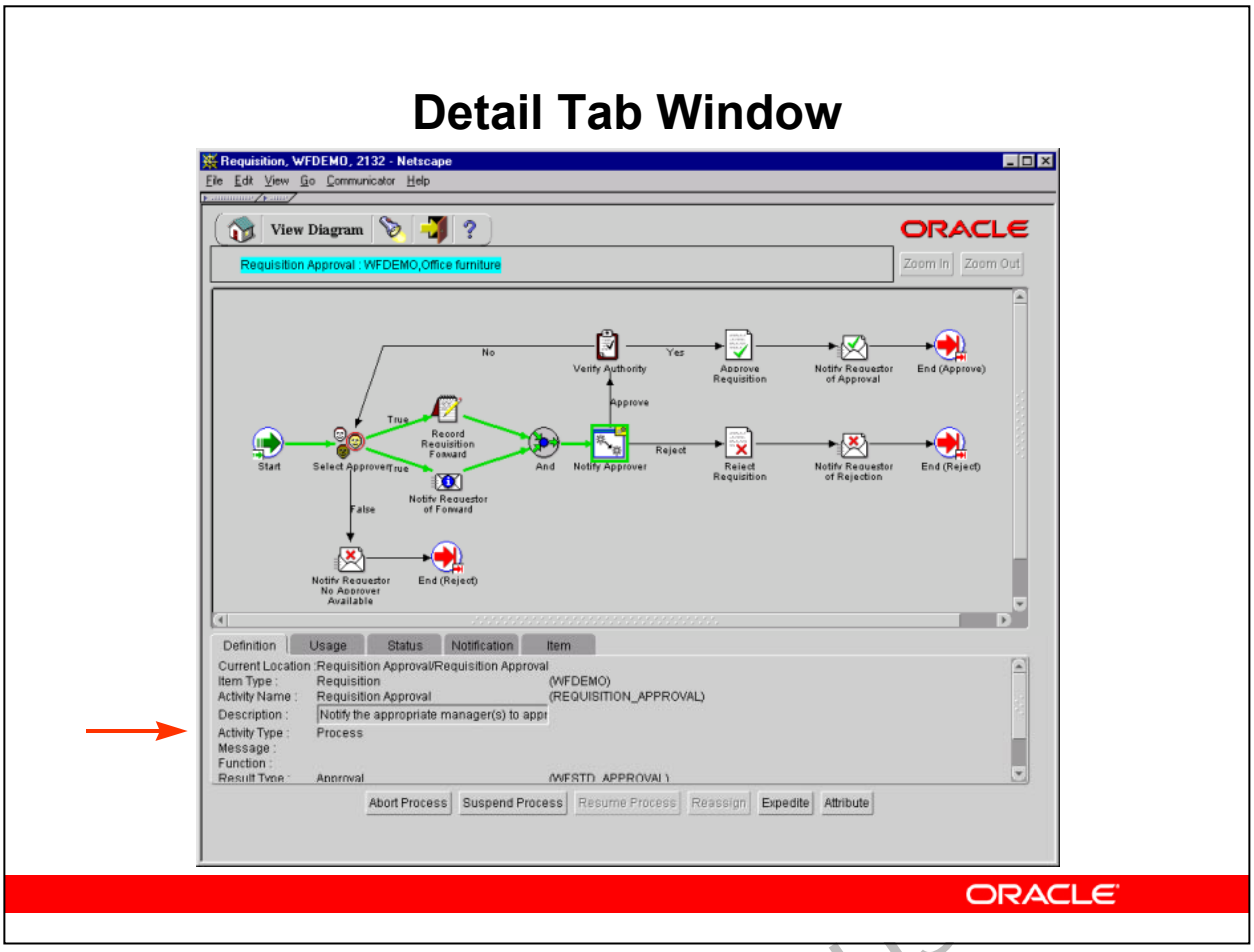
The process title displays the process name, followed by the item type and user key (or item key) that uniquely identifies the process instance. If you drill down to a subprocess in the process diagram window, the process title shows the name of the subprocess.



Process Diagram Window

The process diagram window is a noneditable window that displays the process diagram you created in Oracle Workflow Builder. It provides visual cues about the status of the process and its activities.

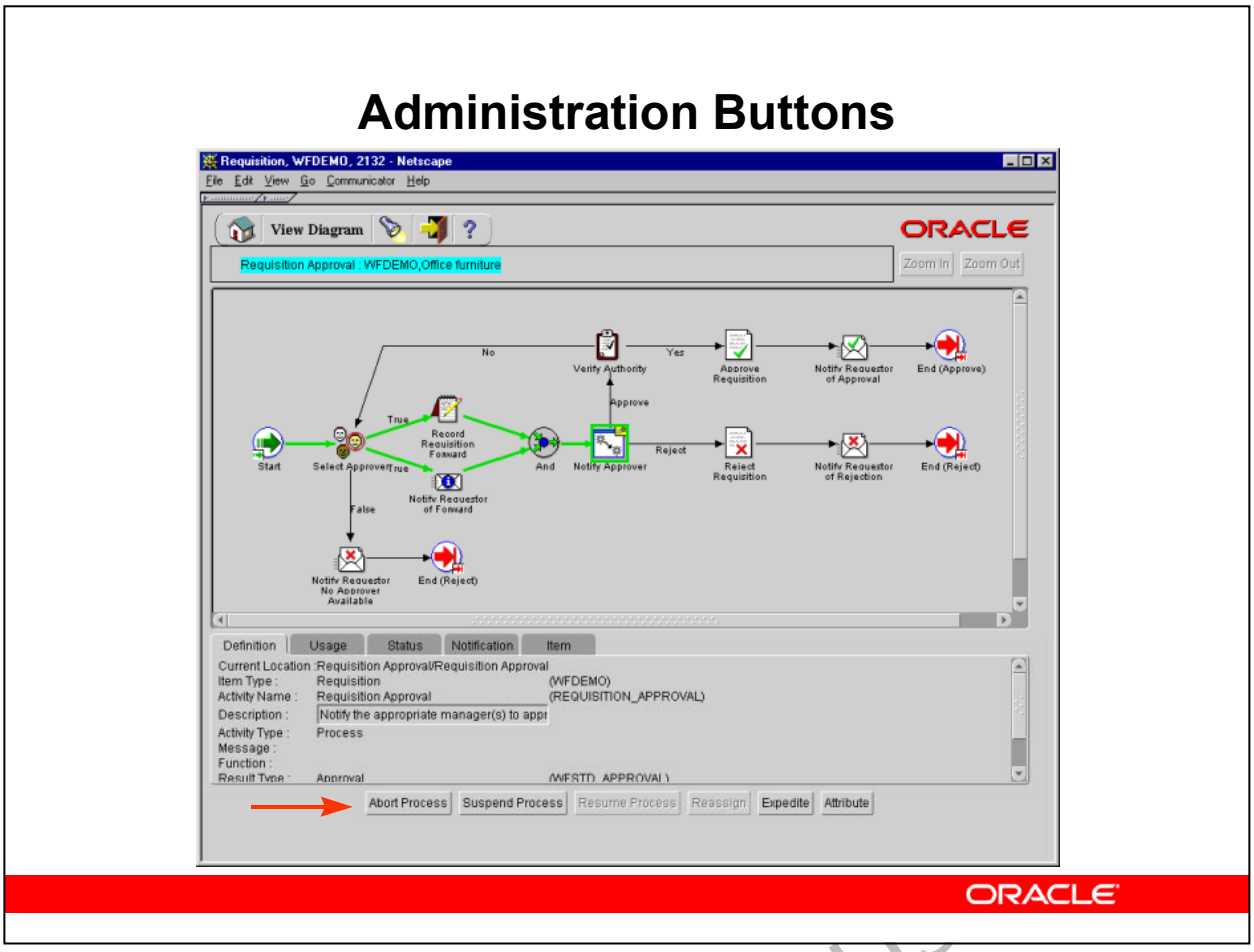
- An activity can be highlighted with a colored box to indicate a distinctive state:
  - Red: The activity is in an error state.
  - Green: The activity is active or in progress.
  - Yellow: The activity is suspended.
- A transition arrow can have a thick green line to indicate that it has been traversed, or it can have a thin black line to indicate that it has not been traversed.
- Click an activity to display information about it in the detail tab window.
- Click any empty space in the diagram to clear a selected activity and display information about the process as a whole in the detail tab window.
- Double-click a subprocess activity to drill down and display the diagram of the subprocess and its information in the detail tab window. You can also select the subprocess activity and then choose Zoom In.



Detail Tab Window

The detail tab window displays detailed information about the selected activity or process. Additional information may appear in these tabs if you are accessing the Workflow Monitor in ADMIN mode. Select from the following tabs:

- Definition: Displays the properties of the activity or process.
- Usage: Displays the properties for the activity as a node in the process.
- Status: Displays status and result information about the activity. Also shows error information if the activity status is ERROR.
- Notification: Displays notification details for the selected notification activity.
- Item: Displays item type and item type attribute information.



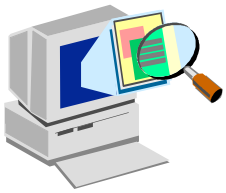
Administration Buttons

The administration buttons appear only when you run the Workflow Monitor in ADMIN mode. The buttons and their behavior are as follows:

- Abort Process: Calls WF_ENGINE.AbortProcess to abort the selected process and cancel any outstanding notifications. Prompts for a result to assign to the aborted process.
- Suspend Process: Calls WF_ENGINE.SuspendProcess to suspend the selected process.
- Resume Process: Calls WF_ENGINE.ResumeProcess to resume the selected suspended process to normal execution status.
- Reassign: Calls WF_ENGINE.AssignActivity to reassign the selected notification activity to a different performer. Prompts for a role name.
- Expedite: Calls WF_ENGINE.HandleError to alter the state of an errored activity, or to undo the selected activity and all other activities following it to rollback part of the process. Prompts you to select one of two values:
  - Skip, to skip the activity and assign it a specified result.
  - Retry, to reexecute the activity.
- Attribute: Lets you change the value of an item attribute.

Application-Controlled Access to the Workflow Monitor

- A calling application can launch a Web browser and pass a Workflow Monitor URL.
- You can monitor a specific item type and key through the following URLs:
  - Process Diagram URL
  - Notifications List URL
  - Activities List URL



ORACLE

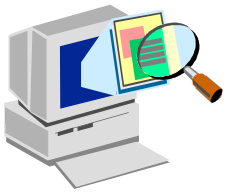
Workflow Monitor

The Workflow Monitor requires a Web browser that supports Java Development Kit (JDK) Version 1.1.8 or higher and Abstract Windowing Toolkit (AWT), such as Netscape Communicator version 4.76 or a higher version of 4.7x, or Microsoft Internet Explorer version 5.0x or 5.5.

From an Oracle E-Business Suite form, use FND_UTILITIES.OPEN_URL( ) to launch a browser and pass a Workflow Monitor URL.

Application-Controlled Access to the Workflow Monitor

- To generate Workflow Monitor URL strings, call the following functions:
  - Process Diagram: WF_MONITOR.GetDiagramURL( )
  - Notifications List: WF_MONITOR.GetEnvelopeURL( )
  - Activities List: WF_MONITOR.GetAdvancedEnvelopeURL( )
- The calling application must supply the Web agent string, item type, and item key, and specify whether to run the monitor in ADMIN or USER mode.



ORACLE

Application-Controlled Access to the Workflow Monitor

Workflow Web Agent String

Use the function wf_core.Translate('WF_WEB_AGENT') to retrieve the <webagent> string used in creating the Find Processes and Monitor URLs.

Workflow Monitor URLs

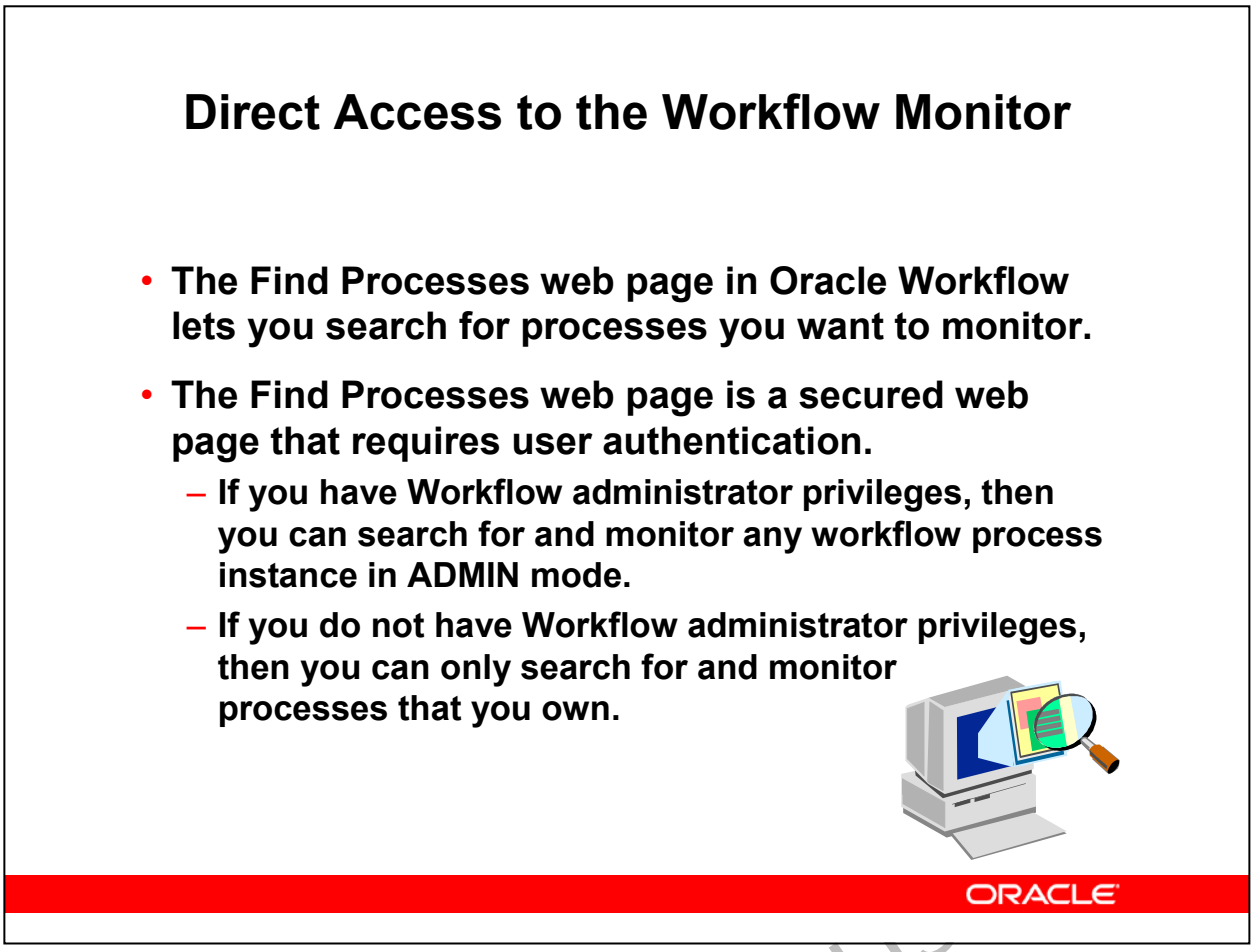
The Workflow Monitor Process Diagram URL is composed as follows:

```
<webagent>wf_monitor.html?x_item_type=<item_type>
&x_item_key=<item_key>&x_admin_mode=<YES_or_NO>
&x_access_key=<access_key>
```

The Notifications List and Activities List URLs are similar to this URL.

User Authentication

The Workflow Monitor functions for access to the Process Diagram, Notifications List, and Activities List all return a hidden password in the URL string that provides the user access to the pages in either ADMIN or USER mode.



Direct Access to the Workflow Monitor

To access the Find Processes web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/ `wf_monitor.find_instance`

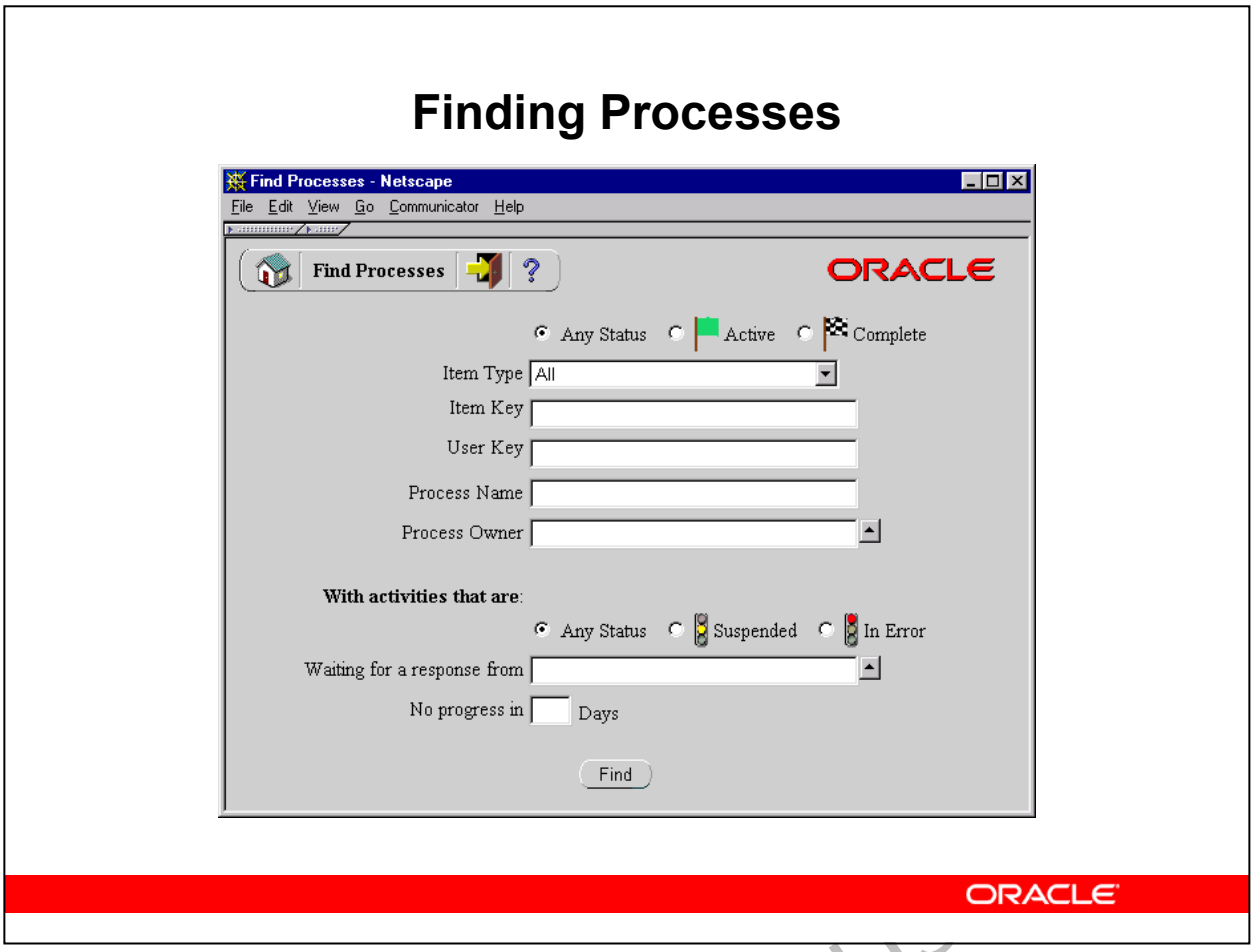
Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Find Processes web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Find Processes option from a Workflow responsibility.

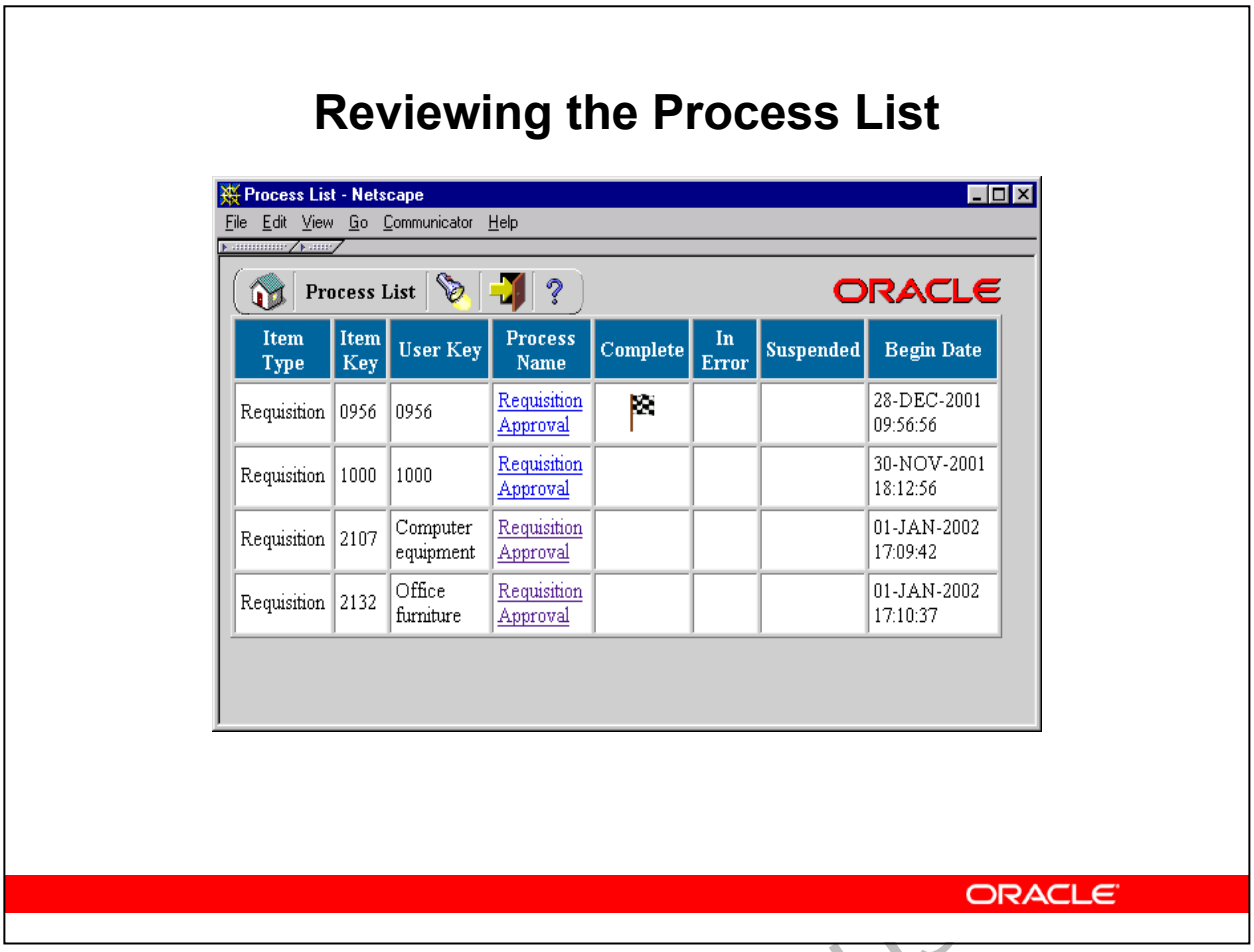
Workflow Administrator Privileges

Users have workflow administrator privileges if they belong to the Oracle Workflow administration role, which is defined in the Global Workflow Preferences web page.



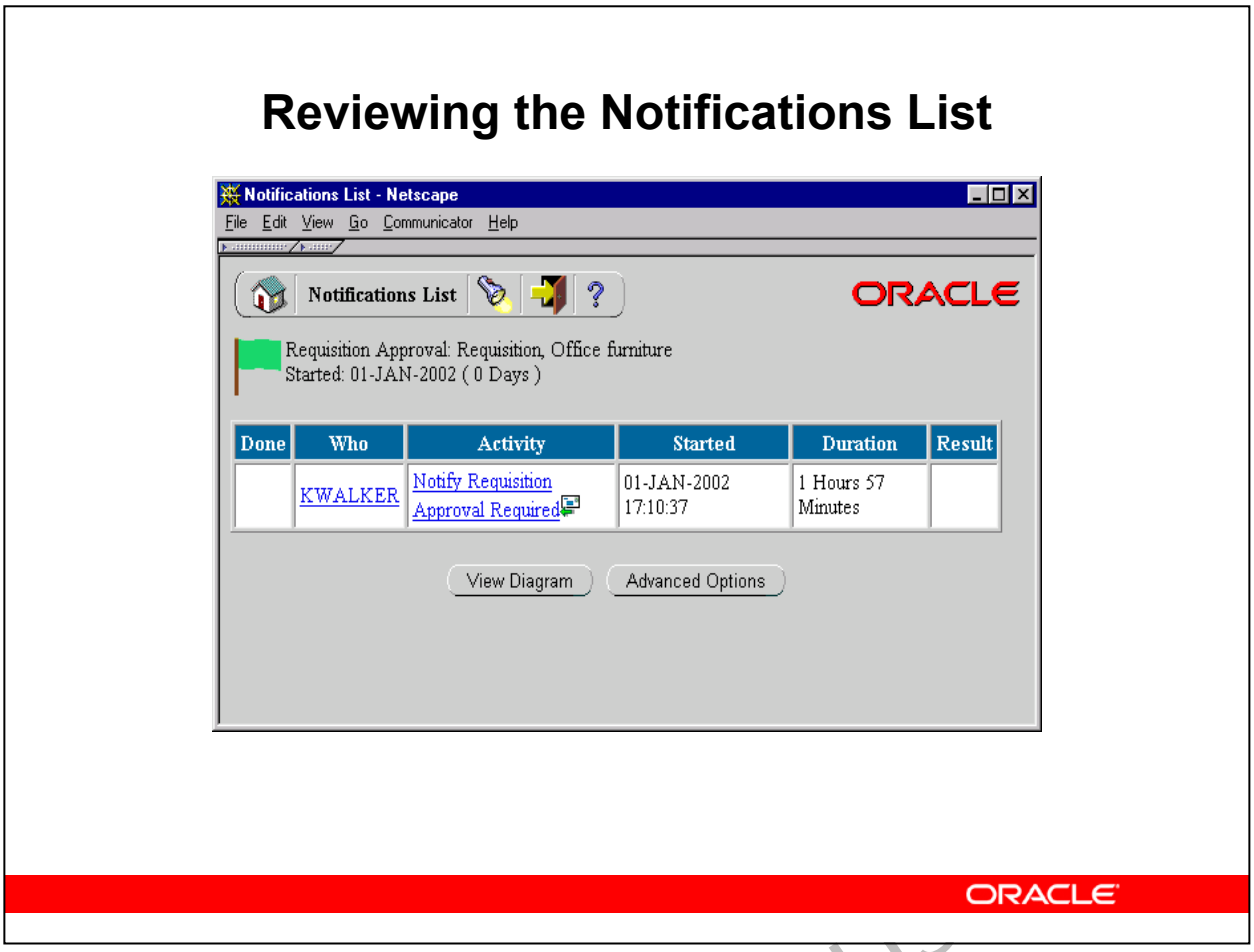
Finding Processes

- The Find Processes Web page lets you specify the following search criteria to locate workflow process instances:
  - Process Status
  - Item Type
  - Item Key
  - User Key
  - Process Name
  - Process Owner (if you have workflow administrator privileges; otherwise you can only search for process instances that you own)
- You can also locate workflow process instances with activities that:
  - Have any status, are suspended, or in error
  - Are waiting for a response from a specified role
  - Have had no progress in a specified number of days
- Choose Find to locate the process instances that meet your specified criteria and display those processes in the Process List.



Reviewing the Process List

- The Process List provides a summary of all process instances that match your Find Processes search criteria.
- Process instances are listed in ascending order, first by item type, then by item key.
- The Process List summarizes the status of each process instance.
- Choose a Process Name link to display the Notifications List, which shows the notification activities that have been initiated for that process instance.

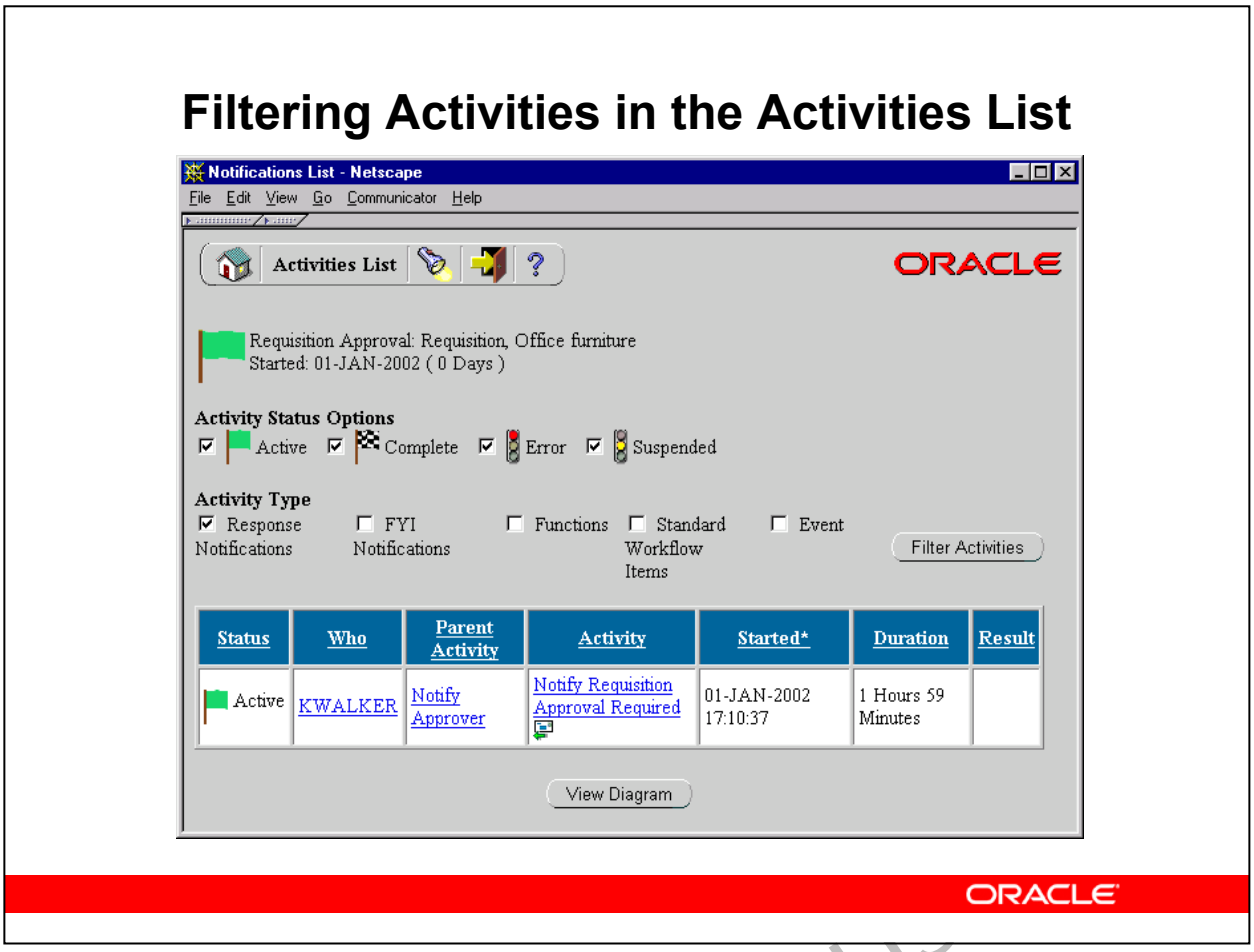


Reviewing the Notifications List

- The Notifications List displays all the current notifications that have been sent that require a Result response. It summarizes what each notification activity is, who it is assigned to, when it was sent, whether it has been completed, how many days have passed before completion and what the result is.

  Note: If the process instance is in an error state and the cause of the error was from a notification, you can click the link in the Result column, if any, to display the cause of the error.
- Choose a user link in the Who column to send an e-mail to the user to whom a notification is assigned.
- Choose a notification activity link in the Activity column to view the full definition of a notification activity.
- If a notification is still open and requires a response, an icon will appear after the notification activity name. You can click this icon to navigate to the Notification Details page, where you can respond to the notification if you are logged in as the notification recipient or if you are logged in with administrator privileges.

- Choose Advanced Options to navigate to the Activities List web page, where you can specify advanced criteria to search for and display specific activities of interest for the process.
- Choose View Diagram to display the selected process instance in the Workflow Monitor for a graphical representation of the process status. When connected to the current session with workflow administrator privileges, the Workflow Monitor displays the process in ADMIN mode; otherwise the process is displayed in USER mode.



Filtering Activities in the Activities List

The Activities List lets you use different criteria to filter for specific activities of interest in the current process instance.

- Activity Status Options:
  - Active (includes Notified, Deferred, and Waiting states)
  - Complete
  - Error
  - Suspended
- Activity Types:
  - Response Notifications
  - FYI Notifications
  - Functions
  - Standard Workflow Items
  - Events

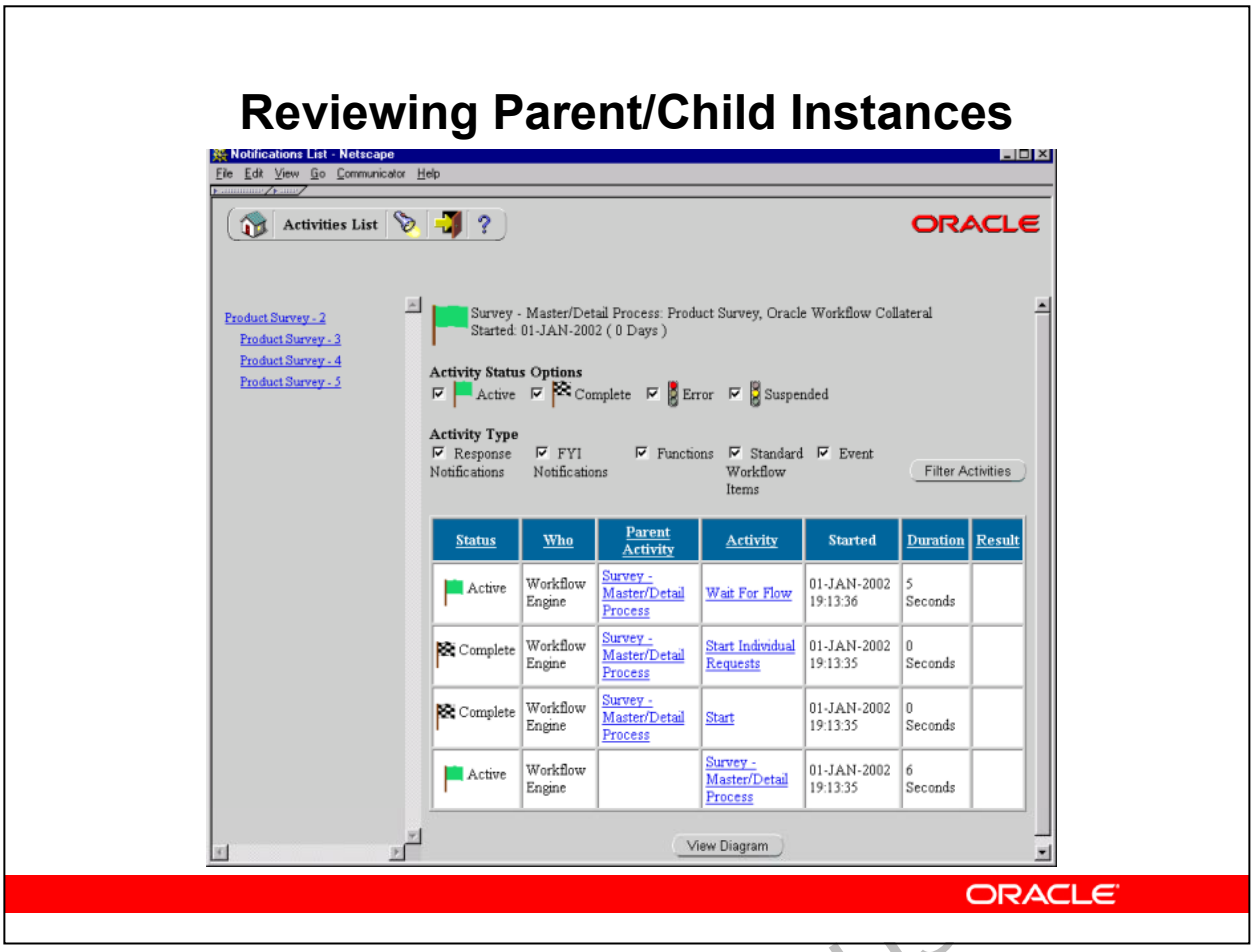
Choose Filter Activities to display the activities that match your criteria.

Filtering Activities in the Activities List

You can sort the Activities List based on any column by clicking the column heading.

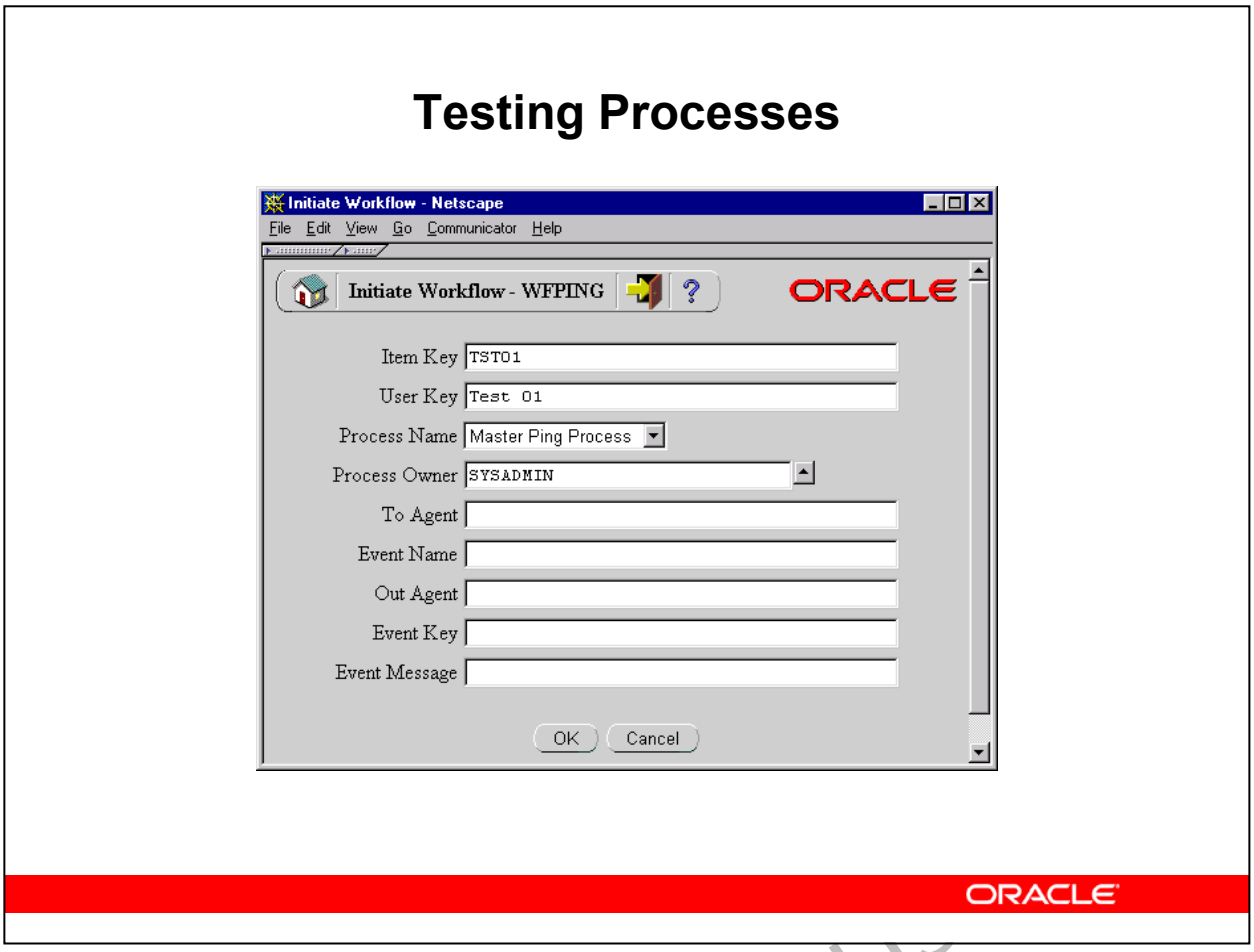
- An asterisk (*) to the left of the sort column title indicates an ascending sort order.
- An asterisk (*) to the right of the sort column title indicates a descending sort order.

- Choose a user link in the Who column to send an e-mail to the user to whom a notification is assigned.
- Choose an activity link in the Activity column to view the full definition of that activity.
- If a notification activity is still open and requires a response, an icon appears after the notification activity name. You can click this icon to navigate to the Notification Details page, where you can respond to the notification if you are logged in as the notification recipient or if you are logged in with administrator privileges.
- Choose View Diagram to display the selected process instance in the Workflow Monitor for a graphical representation of the process status. When connected to the current session with workflow administrator privileges, the Workflow Monitor displays the process in ADMIN mode; otherwise the process is displayed in USER mode.



Reviewing Parent/Child Instances

If the selected process is a member of a parent/child process, a parent/child hierarchy list appears in the left pane of the Activities List page. The hierarchy list shows links to related parent and child instances of the current process. Each link lets you navigate to the Notifications List for the selected parent or child instance.



Testing Processes

You can use the Launch Processes web page to launch a workflow process for testing purposes.

To access the Launch Processes web page for standalone Oracle Workflow, use a web browser to connect to the following URL:

<*webagent*>/ `wf_initiate.ItemType`

Replace <*webagent*> with the base URL of the web agent configured for Oracle Workflow in your Web server.

You can also access the Launch Processes web page from the Oracle Workflow home page, <*webagent*>/wfa_html.home.

For Oracle Workflow embedded in Oracle E-Business Suite, choose the Launch Processes option from a Workflow responsibility.

In the Launch Processes page, choose the item type you want to test. The Initiate Workflow page appears. In this page, specify:

- A unique item key for the process instance
- A user-defined key that you want to use to identify the process

- The name of the process to test
- An optional process owner
- Values for any required item type attributes

Choose OK. The Workflow Monitor Activities List page appears, displaying the activities for the process you launched. You can review the progress of the process in the Workflow Monitor and respond to any notifications using the Notifications Worklist.

Practice - Running a Workflow Process

Overview

In this practice, you will run the Vacation Proposal workflow process you defined in previous practices. You can use the Workflow Monitor to review the status of the process and use the Notifications Worklist to respond to the notifications sent by the process.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility. The username you use to log in should have this responsibility assigned to it.
- The instructor will provide you with the names of users that you can assign as the requestor and approver in the Vacation Proposal process.

Tasks

1. Run the workflow process using the Launch Processes page.

   Solution:

   1. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.
   2. Choose the Launch Processes link.
   3. In the Launch Processes page, choose your *XX* Vacation Proposal item type.

4. In the Initiate Workflow page, enter a unique item key such as *XX*111 in the Item Key field.

5. In the Process Owner field, select the role that you want to use as the requestor for the vacation proposal.

6. In the Process Name field, select your *XX* Vacation Proposal Process.

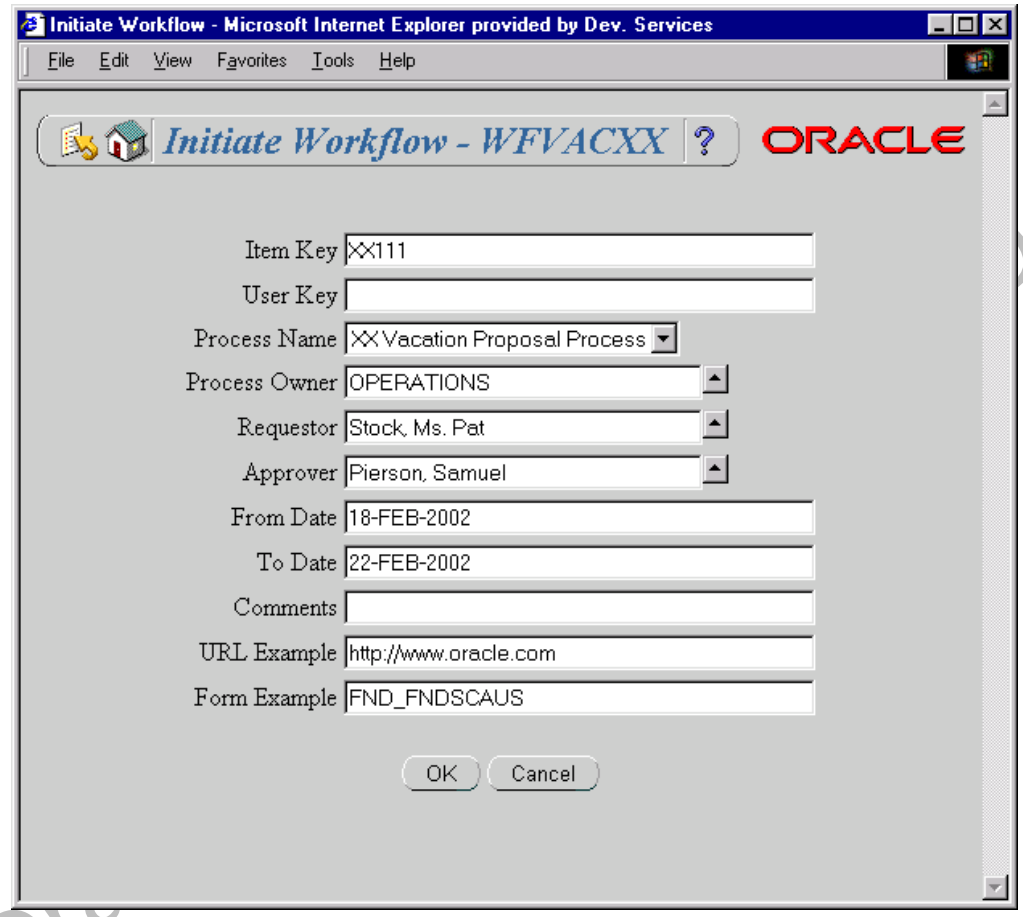
7. In the Requestor field, select the role that you want to use as the requestor for the vacation proposal.

8. In the Approver field, select the role that you want to use as the approver for the vacation proposal.

9. In the From Date field, enter the vacation from date.

10. In the To Date field, enter the vacation to date.

11. Leave the Comments field blank.



12. Choose OK. Oracle Workflow launches the workflow process with the values you entered.

13. Review the process activities in the Activities List page that appears. If an error occurred in your process, you can use the Result Exception link in the Activities List to determine the cause of the error.

14. Choose View Diagram to review the status of the process in the Workflow Monitor.

15. Log off and log in again as the approver.

16. Choose the Worklist link.

17. Open the Vacation Proposal notification sent by your process and review the notification message.

    Note: If you attached the URL Example attribute to the Vacation Proposal message, Oracle Workflow displays the URL link at the end of the notification. If you attached the Form Example attribute to the Vacation Proposal message, Oracle Workflow displays an attached form icon at the end of the notification. If you are viewing the notification from a responsibility that has access to that form, you can click the form icon to drill down to the form. Otherwise, a message is displayed stating that your responsibility does not have access to the form.

18. Enter comments if you want and choose Approve or Reject to respond to the notification.

19. Log off and log in as the requestor.

20. Choose the Worklist link.

21. Open the Vacation Approved FYI or Vacation Rejected FYI notification sent by your process and review the notification message.

22. You can review the status of the process in the Workflow Monitor by choosing the Find Processes link from the Workflow home page and searching for the process with the *XX* Vacation Proposal item type and your item key.

Practice - Modifying a Workflow Process

Overview

In this practice, you will modify the Vacation Proposal workflow process you created in the Creating a Workflow Process practice.

- Create an HTML body for the Vacation Proposal message.
- Modify the process to allow the approver to respond to the Vacation Proposal message with alternate vacation dates.
- Modify the process to allow the requester to resubmit another proposal to the same approver if the first proposal is rejected.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Note: In order to use the sample solution scripts provided for these practices, you must enter the internal names for all objects you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names you define.

Assumptions

- The instructor will provide you with the connect string for the class database and the username and password of the Oracle Workflow database account.
- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility. The username you use to log in should have this responsibility assigned to it.
- The instructor will provide you with the names of users that you can assign as the requestor and approver in the Vacation Proposal process.

Tasks

1. Create an HTML body for the Vacation Proposal message.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Open to open the wfvac*XX*.wft data store you defined in the Creating a Workflow Process practice.

   3. In the navigator tree, select your item type and select the Vacation Proposal message.

   4. Open the property pages for the Vacation Proposal message and choose the Body tab. Within the Body tab, choose the HTML Body tab.

   5. Enter an HTML-formatted version of your message. You can use the Import button to import a sample HTML body from the wfvacxx.html sample solution file on your file system. Choose OK.

   6. In the Navigator window, click the Verify button to verify your workflow.

2. Modify the process to allow the approver to respond to the Vacation Proposal message with alternate vacation dates.

   Solution:

   7. In the navigator tree, select your item type.

   8. From the Edit menu, choose New > Attribute.

   9. Define the following properties for the item attribute:
      - Internal Name: ALT_FROM_DATE
      - Display Name: Alternate From Date
      - Type: Date
      - Format: DD-MON-RRRR

      Choose OK.

   10. From the Edit menu, choose New > Attribute.

   11. Define the following properties for the item attribute:
      - Internal Name: ALT_TO_DATE
      - Display Name: Alternate To Date
      - Type: Date
      - Format: DD-MON-RRRR

      Choose OK.

12. Drag and drop the Alternate From Date and Alternate To Date item attributes onto the Vacation Proposal message to create the corresponding message attributes. Set the Source field for both alternate date attributes to Respond.

13. Open the property pages for the Vacation Rejected message. Choose the Body tab and add the alternate from and to dates into the message body.

14. Drag and drop the Alternate From Date and Alternate To Date item attributes onto the Vacation Rejected message to create the corresponding message attributes. The Source field for both alternate date attributes should be Send.

3. Modify the process to allow the requester to resubmit another proposal to the same approver if the first proposal is rejected.

Solution:

15. In the property pages for the Vacation Rejected message, choose the Result tab. In the Display Name field for the Result, enter "Resubmit Vacation Proposal?". In the Lookup Type field, select Yes/No.

16. Ensure that the Vacation Rejected message has From Date and To Date message attributes. Set the Source field to Respond for these attributes.

17. Open the process diagram window.

18. Double-click the Vacation Rejected FYI notification activity node. Change the display name for the activity to "Vacation Rejected - Resubmit?". Change the result type for the activity to Yes/No to match the message result.

19. Select the transition that connects the Vacation Proposal notification and the Vacation Rejected - Resubmit? notification and drag the transition to create a vertex point. Select the transition again and right-click it. Choose the Locked option from the menu that appears. In this way, you can avoid drawing a new transition on top of this one.

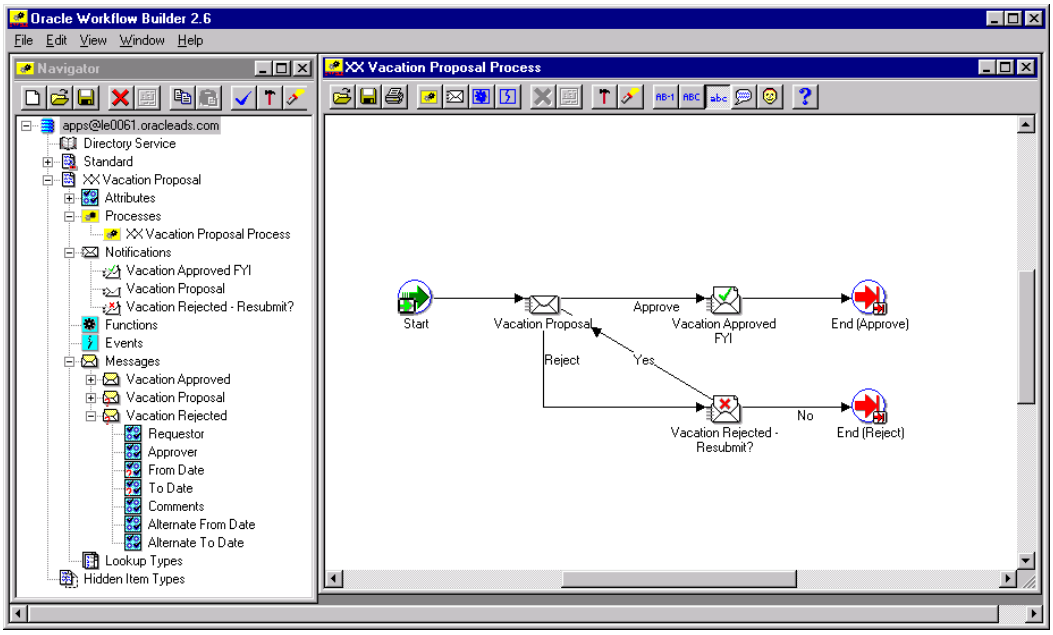
20. Delete the existing transition between the Vacation Rejected - Resbumit? node and the End node. Draw a new transition from the Vacation Rejected - Resbumit? node to the End node and select No from the transition results menu.

21. Draw a new transition from the Vacation Rejected - Resbumit? node to the Vacation Proposal node and select Yes from the transition results menu.

22. In the Navigator window, click the Verify button to verify your workflow. Because you have defined all the underlying components for your process, the Workflow Builder should not display any warnings. Click OK.

23. From the File menu, choose Save to save your work to your workflow definition file.

24. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



25. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

26. Use the Launch Processes page to launch your workflow process and test your work. You can use the Notifications Worklist to view the notifications sent by the process and use the Workflow Monitor to review the status of the process. As the approver, reject the initial vacation proposal. Then, as the requestor, respond to the Vacation Rejected – Resubmit? notification by submitting new vacation dates.

Summary

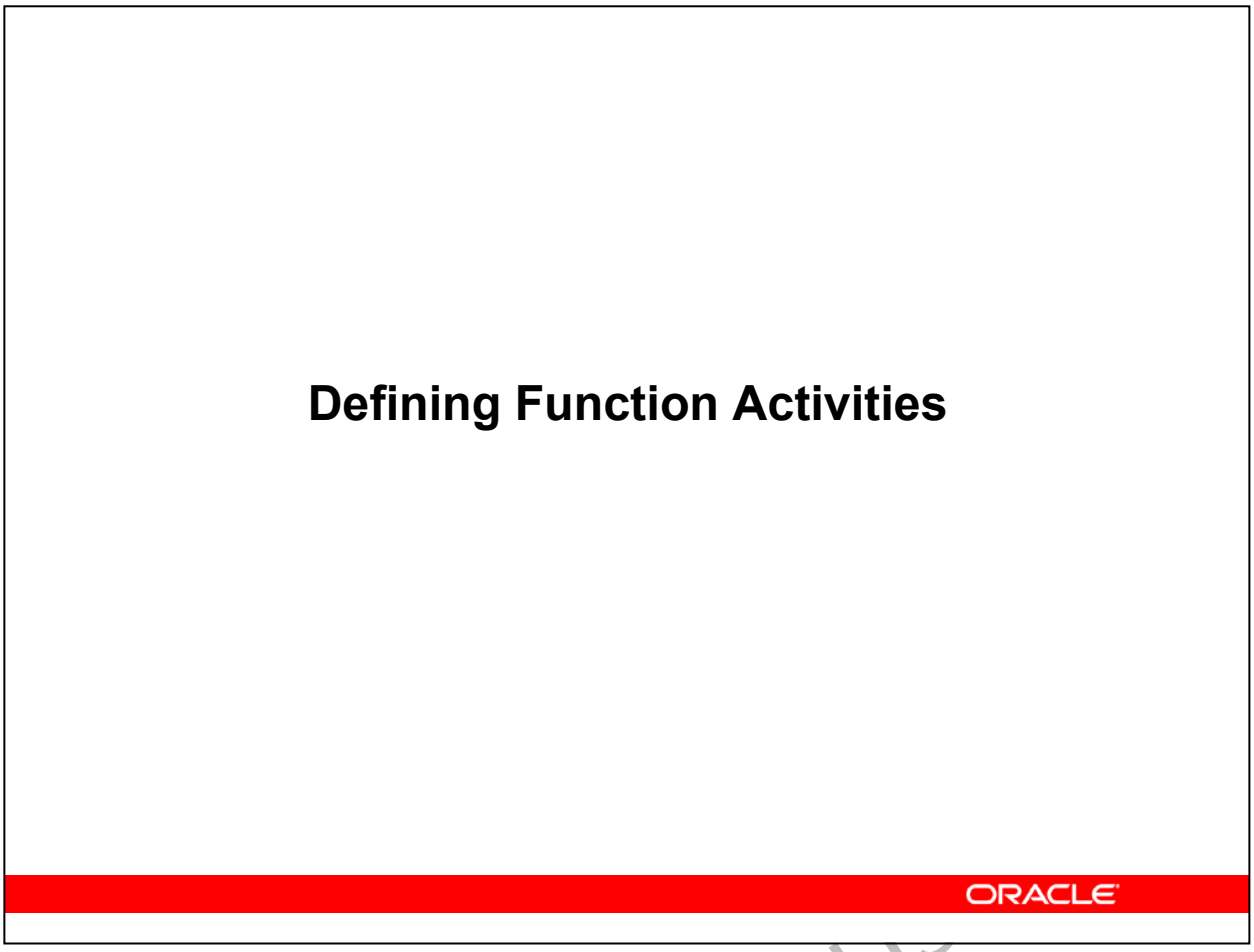
In this lesson, you should have learned how to:

- **Check the progress of a workflow using the Workflow Monitor.**
- **Launch a test process.**

ORACLE

Defining Function Activities

Chapter 12



Objectives

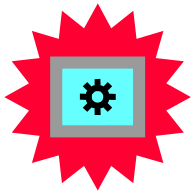
After completing this lesson, you should be able to do the following:

- **Define function activities.**
- **Define activity details.**
- **Define activity attributes.**

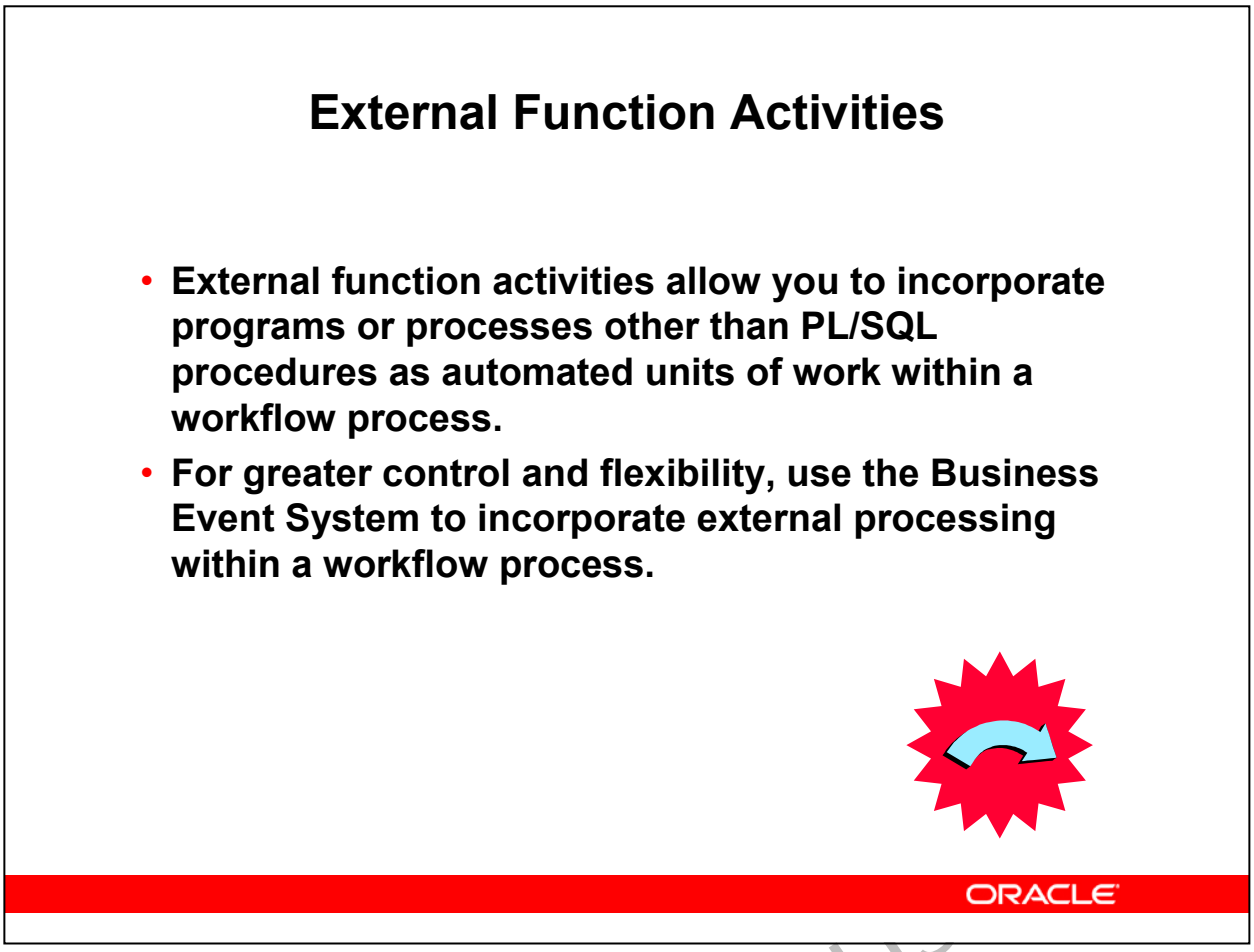
ORACLE

Function Activities

- A function activity is a step in a workflow process to perform an automated unit of work.
- A function activity is usually defined by a PL/SQL stored procedure.
- The PL/SQL procedure for a function activity must follow a standard API.



ORACLE



External Function Activities

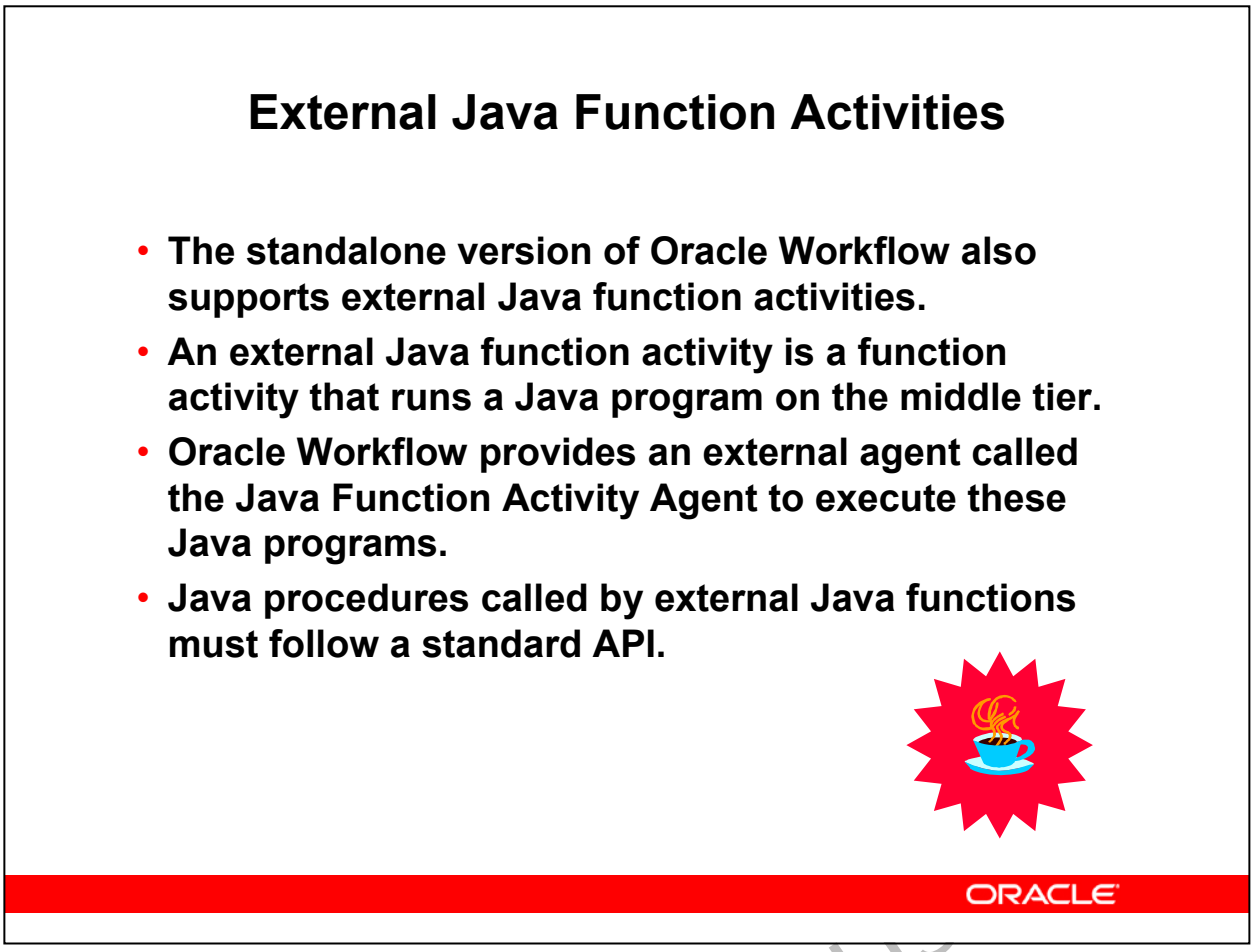
The most flexible and powerful way to integrate external processing into a workflow process is by using business events and subscriptions. However, Oracle Workflow also supports processing external to the database through external function activities. This capability facilitates integration with external products and legacy systems.

The Workflow Engine leverages Oracle Advanced Queuing to support the execution of external function activities.

- An 'Outbound' queue and an 'Inbound' queue are established in Oracle Workflow. When the Workflow Engine encounters an external function activity, a message is written to the Outbound queue. The payload of the message includes any activity attribute name and value pairs.
- A message on the workflow outbound queue is read by an external agent. External execution agents may be any application that is external to the database. The external system consumes and processes the message.
- The external agent enqueues a message in the workflow inbound queue for Oracle Workflow to consume and process. The workflow inbound queue message can be thought of as the "reply" to the original workflow outbound queue message. The inbound queue

message payload can include activity attribute name and value pairs with updated values. The Workflow background engine consumes and processes the inbound message and completes the original function activity.

Note: The ‘Outbound’ and ‘Inbound’ queues used for external function activities are separate from the queues used for the Business Event System. Also, a message in this context is different from the messages associated with notification activities.



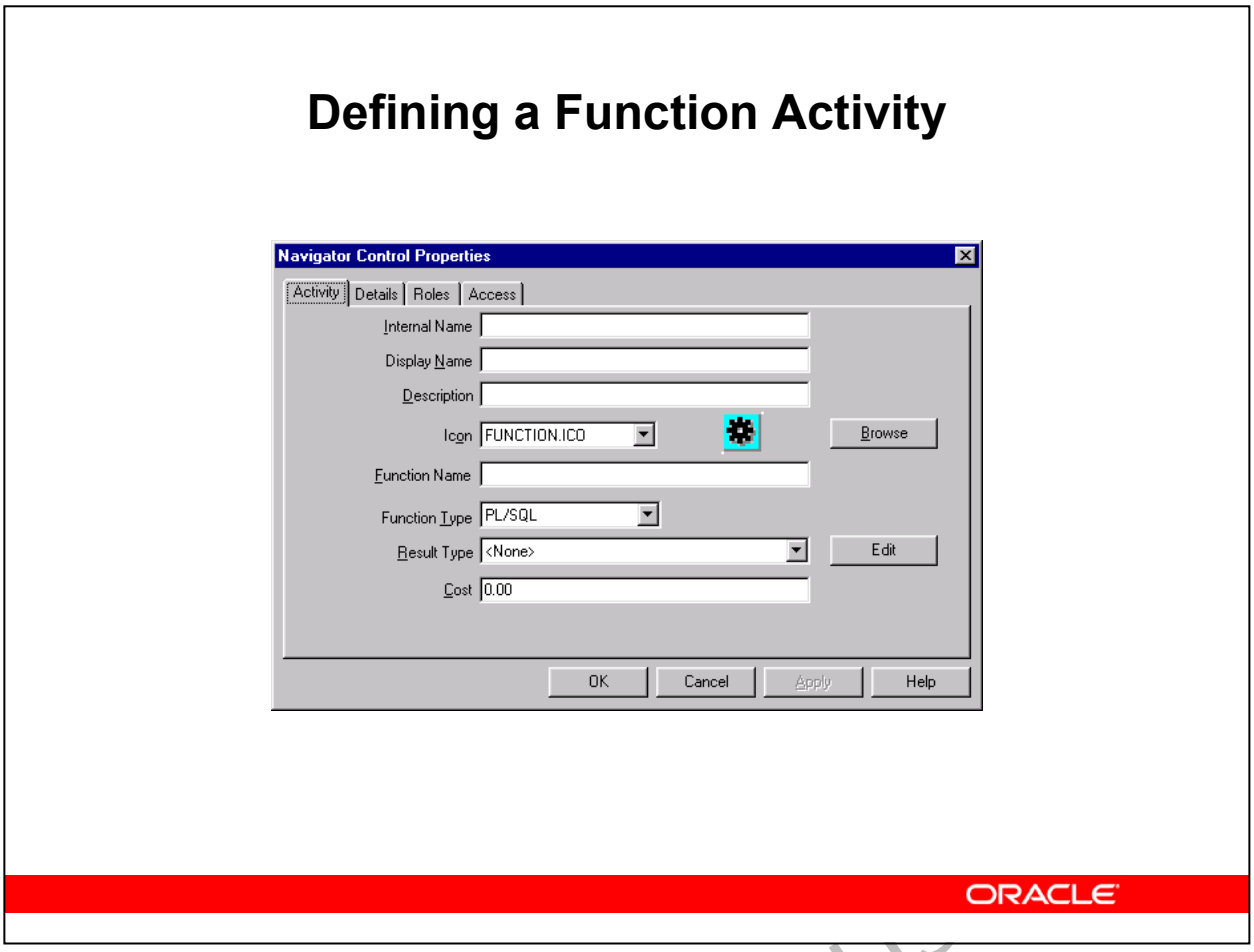
External Java Function Activities

Note: External Java function activities are currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.

An external Java function activity enqueues an entry on the 'Outbound' queue. The Oracle Workflow Java Function Activity Agent dequeues messages of this type, executes the Java program, and enqueues the results on the 'Inbound' queue. A background engine dequeues messages from the ‘Inbound’ queue and completes the function activity to continue the workflow process.

Before you can execute external Java function activities, you must perform the setup steps to run the Java Function Activity Agent and a background engine, to process messages on the ‘Outbound’ and ‘Inbound’ queues, respectively.

For more information, refer to the Setting Up Oracle Workflow chapter in the *Oracle Workflow Guide*.



Defining a Function Activity

Function activities must be associated with an item type and are created in the navigator tree beneath the Functions branch of the item type.

To define a function activity:

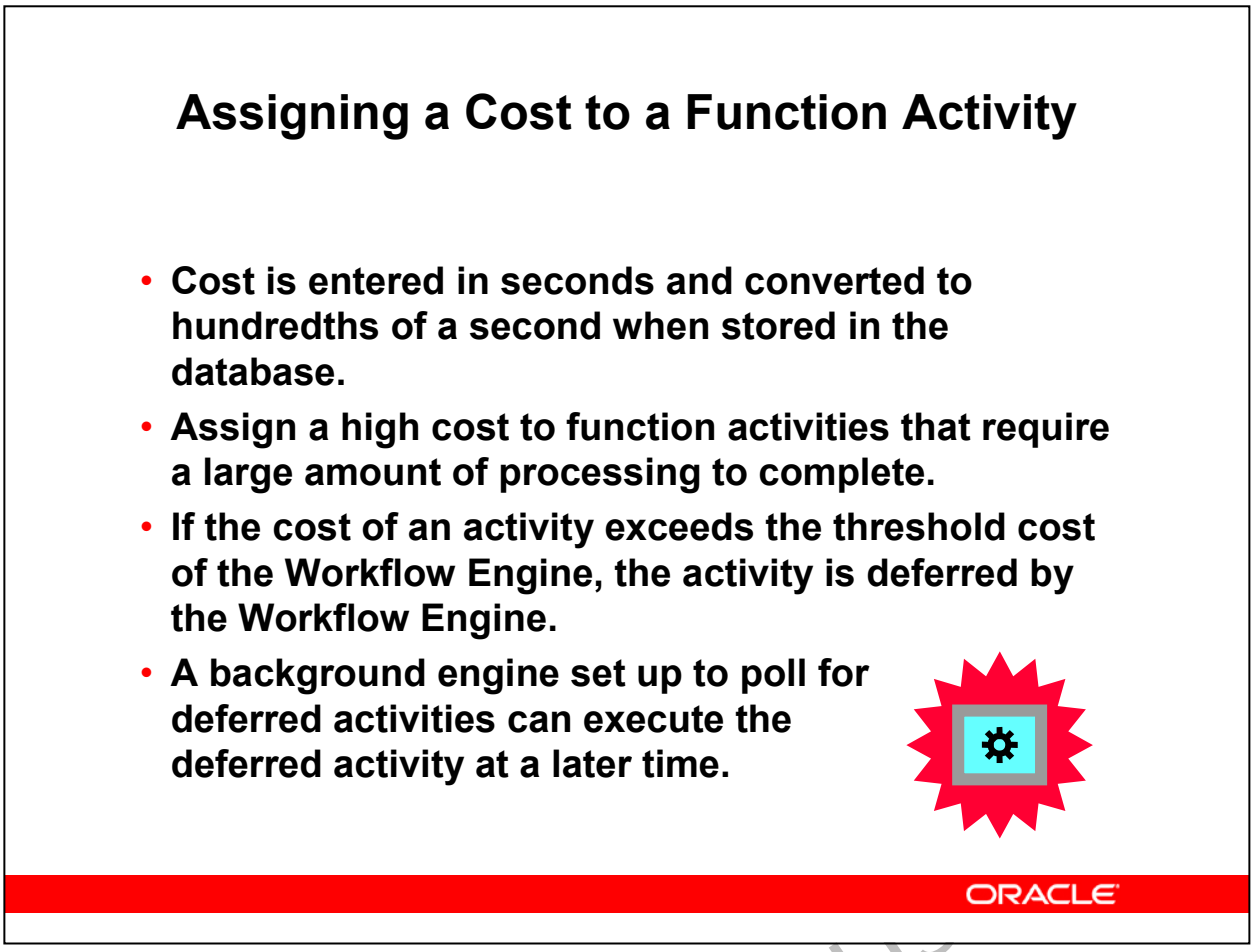
1. In Oracle Workflow Builder, select the item type that you want in the navigator tree. Then choose New Function from the Edit menu.
2. In the Activity property page, enter an internal name for the activity. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the activity.
4. Enter a description of the activity.
5. Choose an icon that identifies the activity.
6. Enter a function name and function type for the activity.

- PL/SQL: Enter the name of the PL/SQL stored procedure that you want the Workflow Engine to execute for this function activity. Use the following format:
  `<package_name>.<procedure_name>`

- External: Enter the name of the external program you want the Workflow Engine to enqueue on the Outbound queue. You must provide an external agent to dequeue and consume the entry on the Outbound queue. The external agent can similarly enqueue an entry onto the Inbound queue for a Workflow background engine to consume and process to complete the function activity.
- External Java: Enter the class name of a custom Java class you want to run on the middle tier. Use the following format:
  `<customPackage>.<customClass>`

  Note: External Java function activities are currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.

7. Specify the result type that contains the possible results that the function can return.
8. Optionally specify a relative cost, in seconds, that represents how long the procedure takes to run. If the cost exceeds the Workflow Engine threshold, the activity will be deferred for later processing by a background engine.
9. Choose Apply to save your changes.
10. Optionally select the Details tab to display and modify additional activity details.
11. Optionally select the Access tab to set the access levels allowed to modify this activity.

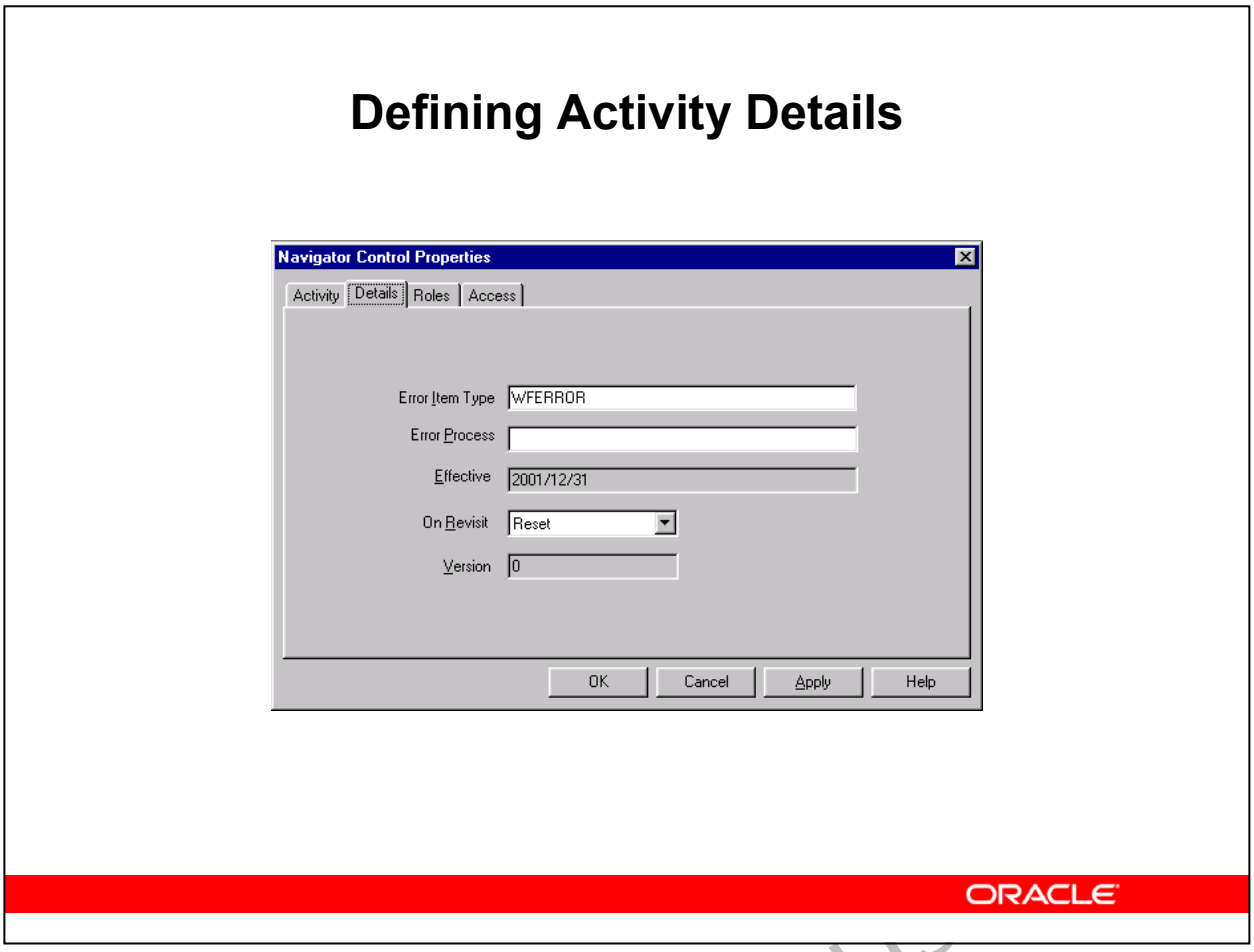


Assigning a Cost to a Function Activity

Activity costs are entered in the Workflow Builder in seconds and stored in the database in hundredths of a second. For example, a function with an execution cost of 10 milliseconds would be entered in the function properties Cost field as 0.01 and stored in the cost database column as 1.

The default Workflow Engine threshold is 500 milliseconds. This value is equivalent to 0.5 when compared with function activity costs entered in the function properties in the Workflow Builder and 50 when compared to function activity costs stored in the database.

The Workflow Engine executes one activity at a time. If possible, you should avoid placing costly activities along the critical path of a process. Instead, place costly activities on parallel branches in a process, where they can be deferred for a background engine to process later.



Defining Activity Details

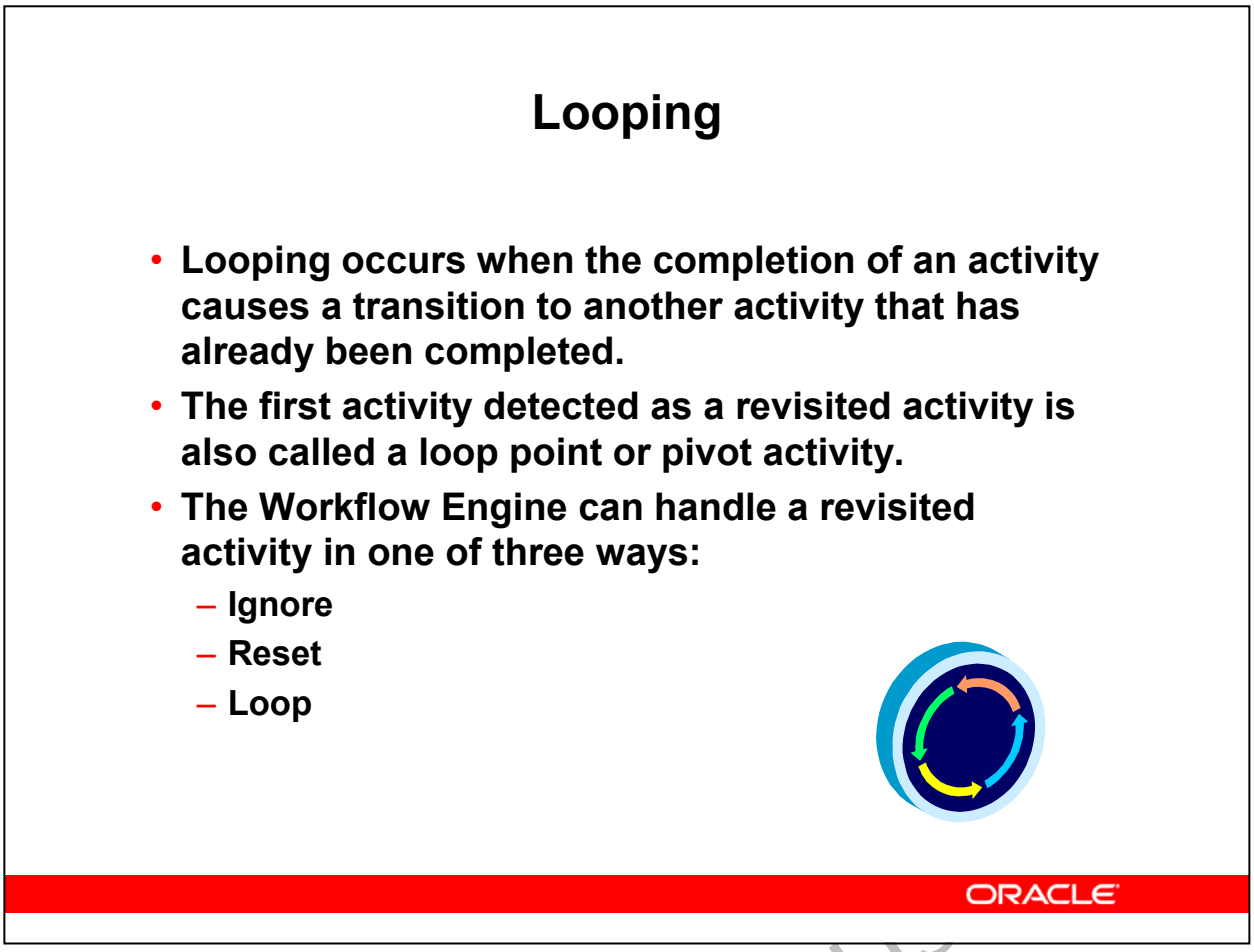
Every activity has a Details property page. The activity details for process, notification, event, and function activities are similar.

To define activity details:

1. Open the property pages for the activity you want in the Oracle Workflow Builder and choose the Details tab.
2. Specify the internal name of the item type that owns the error process that you want to execute if an error occurs in the current activity.
3. Specify the internal name of the error process.
4. Specify the On Revisit value to determine how the Workflow Engine handles this activity when it transitions to the activity more than once.
   - Ignore: Ignores the activity after the activity has been executed once; used for OR-type operations where the rest of the branch is ignored.
   - Reset: Resets completed activities in a loop by executing activities in CANCEL mode before re-executing them in RUN mode.
   - Loop: Re-executes activities in the loop without resetting them.

5. Choose Apply to save your changes.

Note: The Details tab also displays the effective date and version of the activity. You set the effective date when you save your changes using the Save As option in the File menu. All your activity modifications share the same effective date when you save. The version number of the activity is maintained by Oracle Workflow.



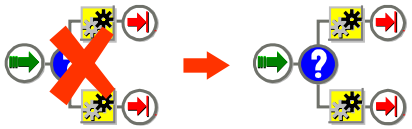
Looping

To handle a revisited activity, the Workflow Engine can:

- Ignore the activity and stop further processing of the thread, so that in effect, the activity can only run once.
- Reset the loop to the loop point before re-executing by first running cancel logic to undo the activities within the loop.
- Loop through the activities to the loop point, re-executing each one without running cancel logic to undo any previous logic. The Loop setting is recommended if you do not need to reset the status of activities within the loop, because it results in faster performance than Reset if it is possible to execute the activities without first running cancel logic.

Error Handling

- **Avoid relying on an error process to handle exceptions in the PL/SQL procedure of a function activity. Instead, try to model your process to handle any known errors that could occur.**
- **Specify an error process in the activity details to handle any unexpected errors encountered during execution of a workflow process.**

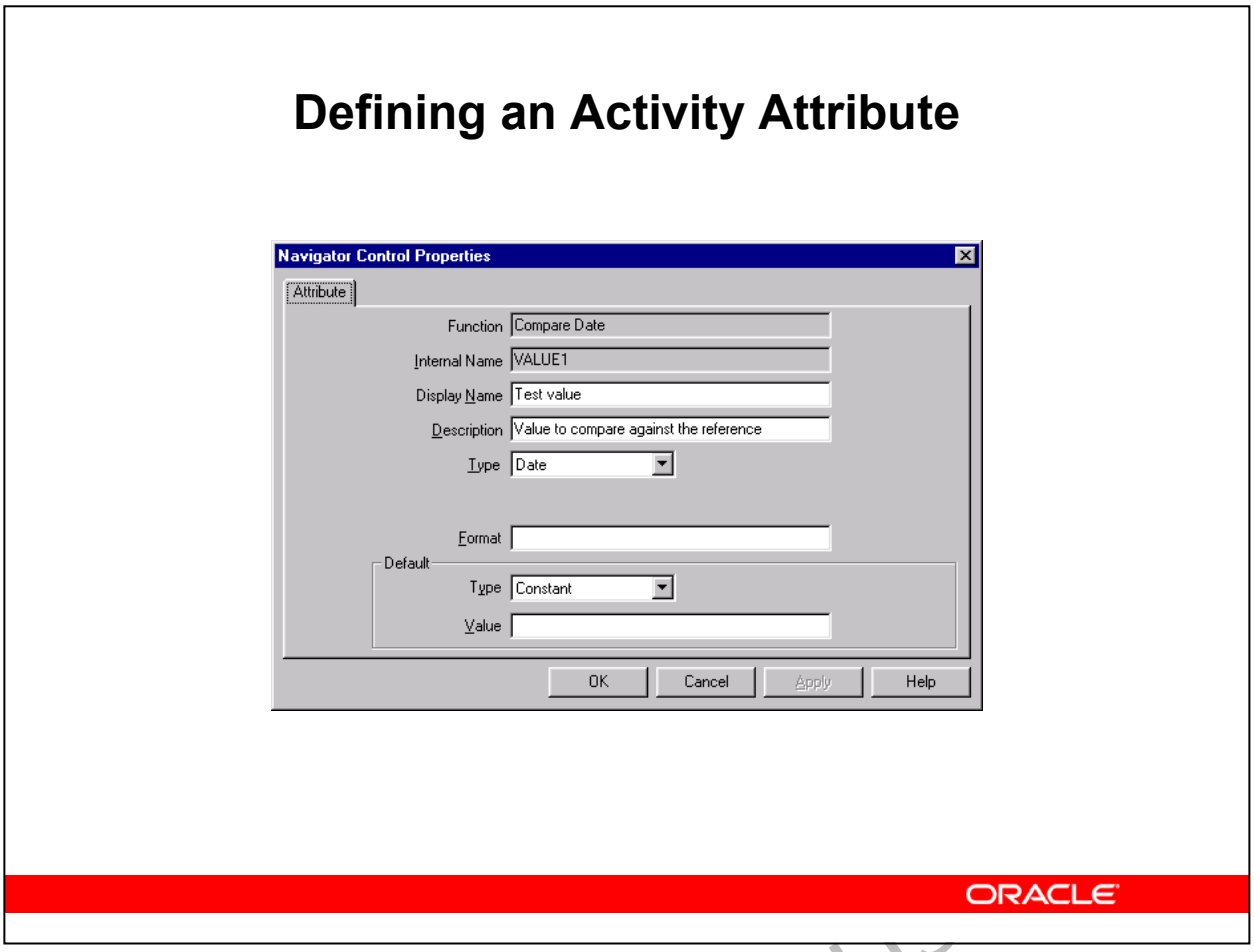


ORACLE

Error Handling

Oracle Workflow provides an item type called System: Error which contains processes that you can use for generic error handling.

- Default Error Process
- Retry-only Process



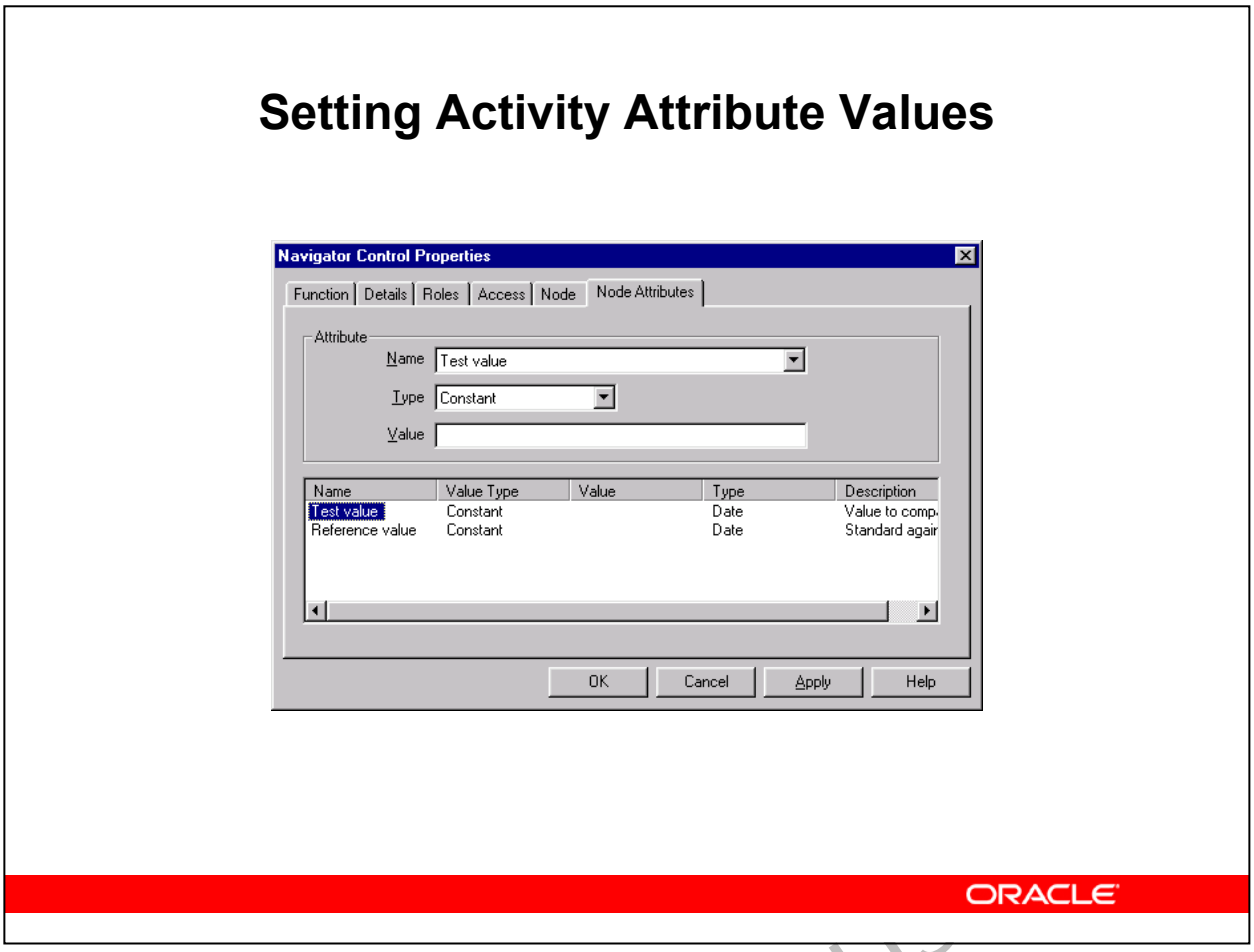
Defining an Activity Attribute

- If the PL/SQL procedure or external program for a function activity contains external parameters, you can redefine those parameters as activity attributes.
- Activity attributes apply only to the current activity and, unlike item type attributes, are not global to a process.
- You can change the definition of an activity attribute at any time in Oracle Workflow Builder.
- Activity attributes can be only be associated with function activities or Raise event activities.
- Activity attributes are created in the navigator tree beneath the activity with which they are associated.

To define an activity attribute:

1. In the Oracle Workflow Builder, select the activity you want in the navigator tree. Then choose New Attribute from the Edit menu.
2. In the Attribute property page, enter an internal name for the attribute. The internal name must be all uppercase without any colons or leading or trailing spaces.

3. Enter a display name for the attribute.
4. Enter a description of the attribute.
5. In the Type field, select the data type for the attribute.
6. Depending on the data type, enter format information and a default value, if applicable.
7. Choose Apply to save your changes.



Setting Activity Attribute Values

Because an activity can be reused throughout a process, the value of an activity attribute can vary from node to node. Specify the values for the attributes in the Node Attributes property page.

To set activity attribute values:

1. In the Oracle Workflow Builder, display the property pages of an activity node, and select the Node Attributes tab.
2. Select an attribute.
3. Enter a value for the attribute. The value can be either constant or dynamic.
   - For a constant value, select Constant and enter a value.
   - For a dynamic value, select Item Attribute and then select the name of an item type attribute that returns a value of the appropriate data type during runtime.
4. Choose Apply to save your changes.

You can use the WF_ENGINE.GetActivityAttribute APIs to retrieve the value of an activity attribute within a PL/SQL function. The following example shows a variable being set to the value of an activity attribute of type text:

```
aname := wf_engine.GetActivityAttrText(itemtype, itemkey,
                                       actid, 'VALUE1');
```

Practice - Defining a Function Activity

Overview

In this practice, you will define a function activity to update a schedule of planned vacations. Add the function activity in the Vacation Proposal item type you created in the Creating a Workflow Process practice. For this practice, you will use a predefined PL/SQL procedure for your function activity. In a later practice, you will have the opportunity to develop your own PL/SQL procedure for this function.

- Define a function activity.
- Load the PL/SQL package containing the PL/SQL procedure that the function activity calls.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Note: In order to use the sample solution scripts provided for these practices, you must enter the internal names for all objects you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names you define.

Assumptions

- The instructor will provide you with the connect string for the class database and the username and password of the Oracle Workflow database account.
- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility. The username you use to log in should have this responsibility assigned to it.
- The instructor will provide you with the names of users that you can assign as the requestor and approver in the Vacation Proposal process.

Tasks

1. Define a function activity.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Open to open the wfvac*XX*.wft data store you defined in the Creating a Workflow Process practice.

   3. In the navigator tree, select your item type.

   4. Open the process diagram window for the Vacation Proposal process.

   5. Delete the transition between the Vacation Approved FYI node and the End node.

   6. Click the New Function button, position the cross-hair cursor between the Vacation Approved FYI node and the End node, and click again to create a new function activity node there and open the property pages for the node.

   7. Define the following properties for the function activity:
      - Internal Name: SCHEDULE_UPDATE
      - Display Name: Update Vacation Schedule

      Note: As a workflow development standard, where possible, you should use the procedure name of the function called by the activity as the internal name of the activity. For example, the activity that calls the function WFVACXX.SCHEDULE_UPDATE should have the internal name SCHEDULE_UPDATE.

   8. Select the icon DB_UPD.ICO for the activity.

   9. Enter the package and procedure name WFVACXX.SCHEDULE_UPDATE for the activity. This function will have no result.

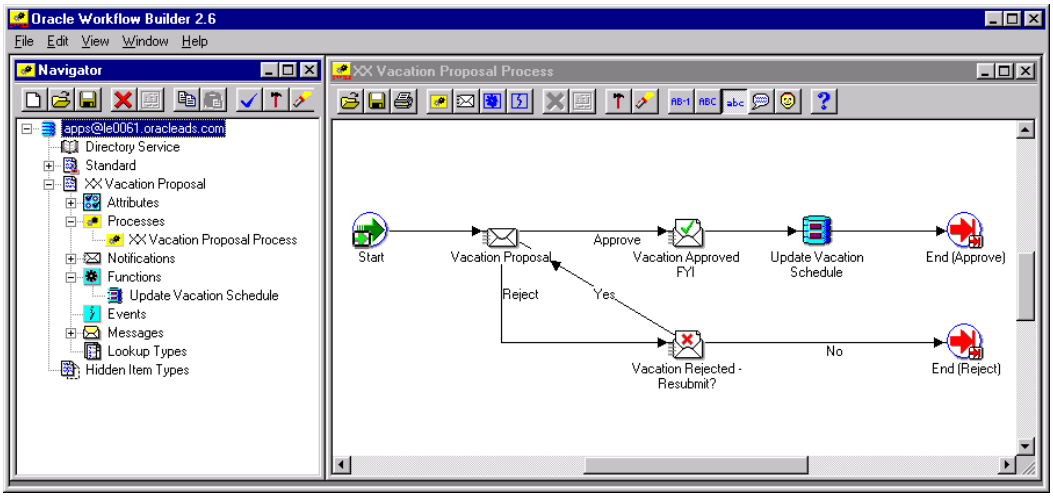
      Note: If you are defining a function activity during the process design phase and you do not yet know the package and procedure name for the PL/SQL procedure you want the activity to call, you can enter WF_STANDARD.NOOP as a placeholder in the Function Name field. WF_STANDARD.NOOP is a standard Workflow PL/SQL procedure that sets the resultout to wf_engine.eng_completed and returns.

   10. Draw transitions from Vacation Approved FYI to Update Vacation Schedule, and from Update Vacation Schedule to End (Approve).

   11. To specify error handling for the Vacation Proposal process, open the property pages for the Vacation Proposal process activity and choose the Details tab. Enter WFERROR as the error item type and DEFAULT_ERROR as the error process.

   12. From the File menu, choose Save to save your work to your workflow definition file.

13. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



2. Load the PL/SQL package containing the PL/SQL procedure that the function activity calls.

Solution:

14. Copy and edit the sample table creation script named wfvacxxc.sql. Open a copy of the sample file and replace all instances of XX with your own terminal number. Save the file and rename it by replacing *xx* with your terminal number.

    This script creates a vacation schedule table named WFVAC*XX*_VACATION_SCHEDULE. The table includes the following columns:
    - REQUESTOR - varchar2(30)
    - APPROVER - varchar2(30)
    - FROM_DATE - date
    - TO_DATE - date

15. Copy and edit the sample package specification and body scripts. The sample scripts are named wfvacxxs.sql and wfvacxxb.sql, respectively. They create a package named WFVAC*XX* that contains all the sample PL/SQL procedures for all practices in this course. The procedure specific to this practice is WFVAC*XX*.SCHEDULE_UPDATE. Open a copy of each sample file and replace all instances of XX with your own terminal number. Save the files and rename them by replacing *xx* with your terminal number.

    The WFVAC*XX*.SCHEDULE_UPDATE procedure records an approved vacation proposal in the WFVAC*XX*_VACATION_SCHEDULE table.

16. Log in to SQL*Plus using the database username, password, and connect string provided by the instructor. Run the table creation, package specification, and package body scripts in that order.

17. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow

administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

18. Use the Launch Processes page to launch your workflow process and test your work. You can use the Notifications Worklist to view the notifications sent by the process and approve the vacation proposal, and use the Workflow Monitor to review the status of the process.
    - Run the process and approve the vacation proposal.
    - Use SQL*Plus to verify that the appropriate row was inserted into your WFVAC*XX*_VACATION_SCHEDULE table. Enter the following command:

      ```
      select * from WFVACXX_VACATION_SCHEDULE;
      ```

Practice - Branching on a Function Activity Result

Overview

In this practice, you will create a function activity to check whether the approver is the same as the requestor in the Vacation Proposal item type you created in the Creating a Workflow Process practice, and model different branches in the process based on the function activity result. For this practice, you will use a predefined PL/SQL procedure for your function activity. In a later practice, you will have the opportunity to develop your own PL/SQL procedure for this function.

- Load the PL/SQL package containing the PL/SQL procedure for the function activity to call.
- Define the function activity with a result type.
- Modify the process to branch on the function activity result.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Note: In order to use the sample solution scripts provided for these practices, you must enter the internal names for all objects you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names you define.

Assumptions

- The instructor will provide you with the connect string for the class database and the username and password of the Oracle Workflow database account.
- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility. The username you use to log in should have this responsibility assigned to it.
- The instructor will provide you with the names of users that you can assign as the requestor and approver in the Vacation Proposal process.

Tasks

1. Load the PL/SQL package containing the PL/SQL procedure for the function activity to call.

   Solution:

   1. If you have not already done so, copy and edit the sample package specification and body scripts. The sample scripts are named wfvacxxs.sql and wfvacxxb.sql, respectively. They create a package named WFVAC*XX* that contains all the sample PL/SQL procedures for all practices in this course. The procedure specific to this practice is WFVAC*XX*.CHECK_APPROVER. Open a copy of each sample file and replace all instances of XX with your own terminal number. Save the files and rename them by replacing *xx* with your terminal number.

      The WFVAC*XX*.CHECK_APPROVER procedure checks whether the approver is the same as the requestor.

      Note: If you have already copied, edited, and run the complete sample package specification and body scripts containing all the sample procedures for all practices, you do not need to repeat these steps.

   2. If you have not already done so, log in to SQL*Plus using the database username, password, and connect string provided by the instructor. Run the package specification and package body scripts in that order.

2. Define the function activity with a result type.

   Solution:

   3. Start the Oracle Workflow Builder.

   4. From the File menu, choose Open to open the wfvac*XX*.wft data store you defined in the Creating a Workflow Process practice.

   5. Open the process diagram window for the Vacation Proposal process.

   6. Delete the transition between the Start node and the Vacation Proposal node.

   7. Click the New Function button, position the cross-hair cursor between the Start node and the Vacation Proposal node, and click again to create a new function activity node there and open the property pages for the node.

   8. Define the following properties for the function activity:
      - Internal Name: CHECK_APPROVER
      - Display Name: Approver Same as Requestor?

      Note: As a workflow development standard, where possible, you should use the procedure name of the function called by the activity as the internal name of the activity.

For example, the activity that calls the function WFVACXX. CHECK_APPROVER should have the internal name CHECK_APPROVER.

9. Select the icon COMPARE.ICO for the activity.

10. Enter the package and procedure name WFVACXX. CHECK_APPROVER as the function name for the activity.

11. Select the Yes/No result type for the activity.

3. Modify the process to branch on the function activity result.

Solution:

12. Draw a transition from the Start node to the Approver Same as Requestor? node.

13. Draw a transition from the Approver Same as Requestor? node to the Vacation Proposal node and select No from the transition results menu.

14. Drag and drop an additional End node from the Standard item type into the process diagram. Open the property pages for the node, and choose the Node tab. In the Start/End field, select End. In the Result field, select Reject.

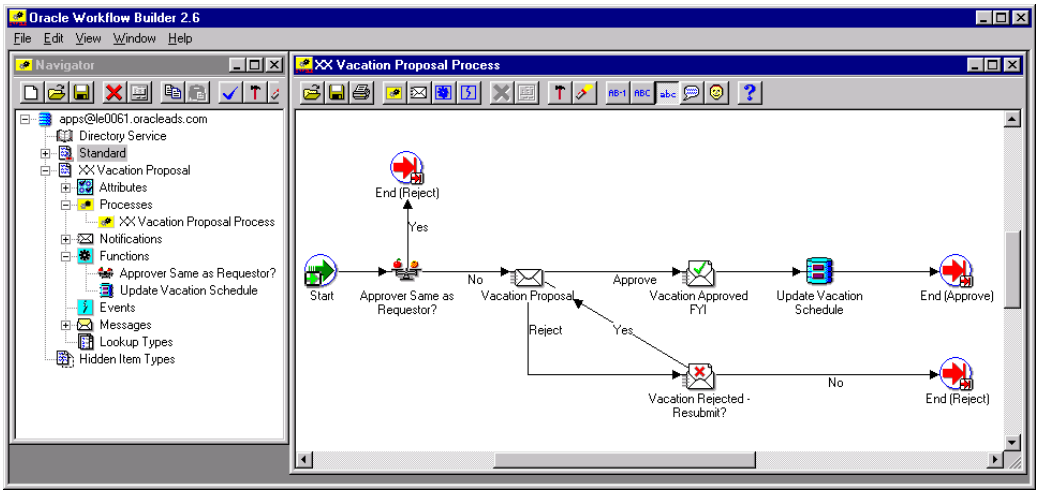
    Using multiple End nodes can help simplify your diagram and uniquely identify which End activity is executed when different paths through a process can terminate with the same result. For example, the Vacation Proposal can now end with a result of Reject for two unique reasons:
    - A vacation proposal is submitted with the approval equal to the requestor.
    - A vacation proposal is rejected by the approver.

15. Draw a transition from the Approver Same as Requestor? node to the new End node and select either Yes or <Default> from the transition results menu.

16. In the Navigator window, click the Verify button to verify your workflow.

17. From the File menu, choose Save to save your work to your workflow definition file.

18. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



19. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

20. Use the Launch Processes page to launch your workflow process and test your work. You can use the Notifications Worklist to view the notifications sent by the process, and use the Workflow Monitor to review the status of the process.
    - First, run the process and enter the same role for the requestor and the approver. The process should end with a result of Reject.
    - Then, run the process again and enter different roles for the requestor and the approver. The process should proceed to send the Vacation Proposal notification to the approver.

Summary

In this lesson, you should have learned how to:

- **Define function activities.**
- **Define activity details.**
- **Define activity attributes.**

ORACLE

Defining Event Activities

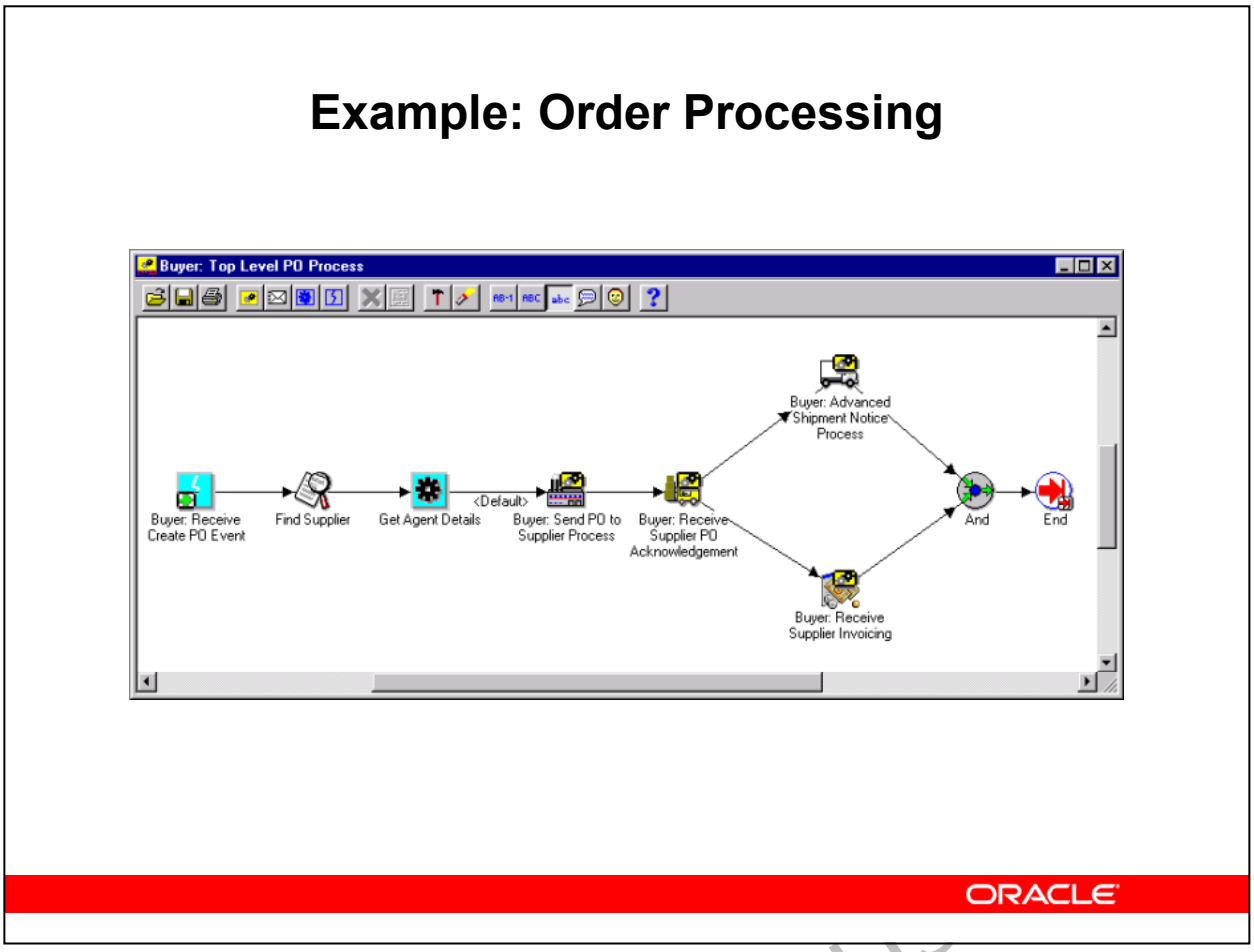
Chapter 13

Objectives

After completing this lesson, you should be able to do the following:

- **Define event activities.**
- **Define event details for event activity nodes.**
- **Send an event to a workflow process to start or continue the process.**
- **Use standard activities to manage information from event messages.**

ORACLE



Example: Order Processing

This example shows a workflow process that includes business events. The example comes from the Event System Demonstration workflow which is available with the standalone version of Oracle Workflow. This process includes activities that receive a purchase order event to launch the workflow, send the purchase order event to a supplier, and wait to receive other events from the supplier in response to the order, such as an order acknowledgement, advanced shipment notice, and invoice.

Event Activities

- An event activity represents a business event within a workflow process.
- You can include event activities in workflow processes to model complex processing or routing logic based on the content of an event.

ORACLE

Event Activity Actions

An event activity can:

- **Receive an event from the Event Manager**
- **Raise an event to the Event Manager**
- **Send an event message to an agent**

ORACLE

Receive Event Activities

- If you send an event to a workflow process from a subscription, the process must include a Receive event activity to accept the event.
- An event sent to a process can:
  - Launch a new process—The Receive event activity must be marked as a Start activity.
  - Continue an existing process—The correlation ID in the event message matches the event with the running process.

ORACLE

Receive Event Activities

If a Receive event activity is marked as a Start activity, it is always enabled to receive events. Otherwise, the Receive event activity can only receive events after the process transitions to that activity.

Receive Event Activities

- **The event filter for a Receive event activity determines which event the activity can receive.**
- **If the event filter is set to an individual event's internal name, the activity can receive only the specified event.**
- **If the event filter is blank, the activity can receive any event.**

ORACLE

Receive Event Activities

You can only specify an individual event as the event filter for a Receive event activity. The event filter cannot be an event group.

Receive Event Activities

When a Receive event activity receives an event, the Workflow Engine:

- **Stores the event name, event key, and event message in the item type attributes specified for the activity node**
- **Sets any parameters in the event message parameter list as item type attributes for the process, creating new item type attributes if a corresponding attribute does not already exist for any parameter**
- **Continues the thread of execution from the event activity**

ORACLE

Raise Event Activities

- A Raise event activity:
  - Retrieves the event name, event key, and event data specified for the activity node and sets them into an event message structure
  - Retrieves the names and values of any activity attributes specified for the activity node and sets them as parameters in the parameter list within the event message
  - Also sets the item type and item key for the current workflow process as parameters in the parameter list within the event message
  - Raises the event to the Event Manager
- When the event is raised, it triggers any eligible subscriptions to that event by the local system.

ORACLE

Raise Event Activities

If the event message is later received by another process, the Workflow Engine uses the item type and item key that were included as parameters in the parameter list to automatically set the process that raised the event as the parent for the process that receives the event.

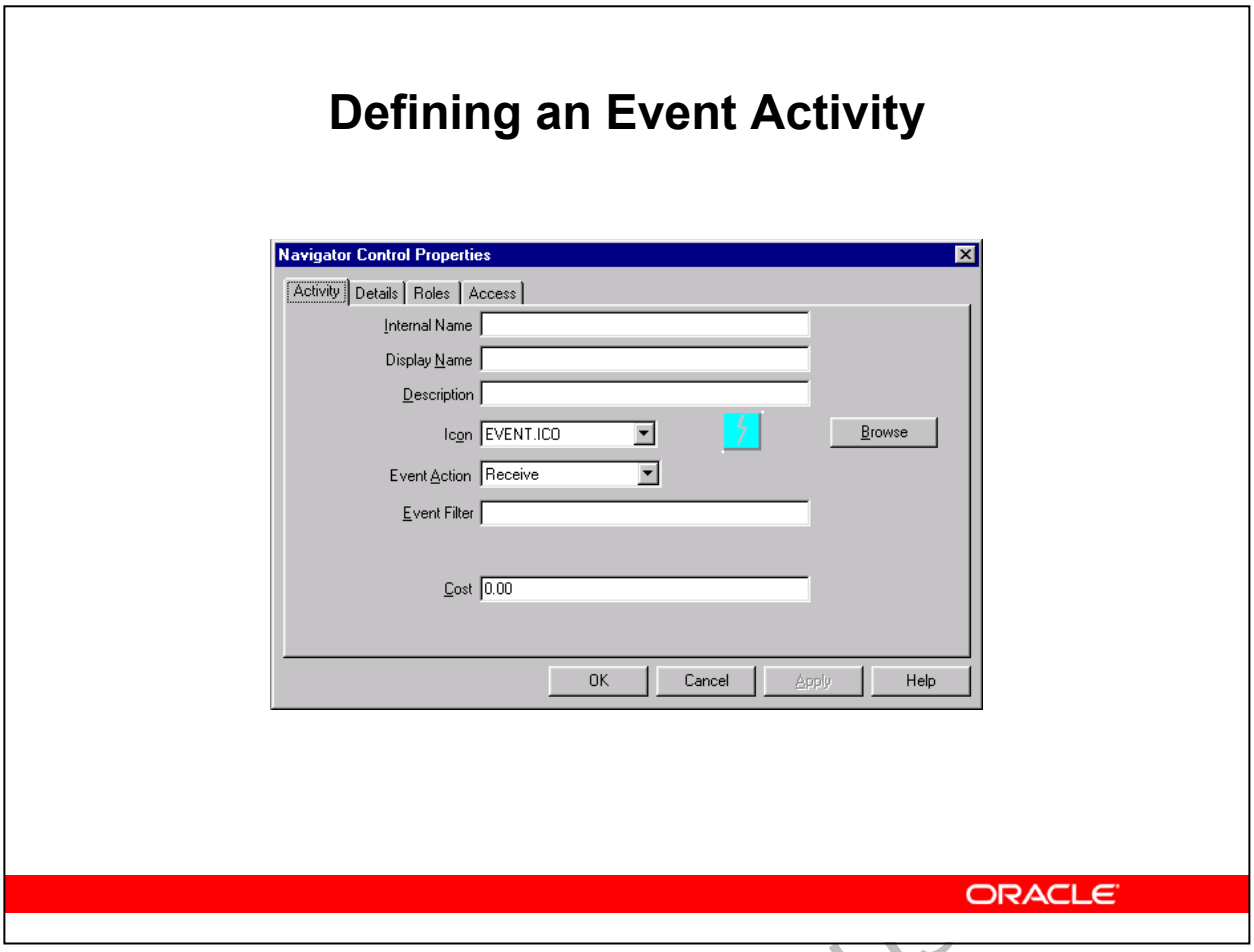
Send Event Activities

- **A Send event activity:**
  - **Retrieves the event name, event key, event message, Out Agent, and To Agent specified for the activity node**
  - **Sends the event message by placing it on the Out Agent's queue, addressed to the To Agent**
- **When the event is received by the To Agent, it triggers any eligible subscriptions to that event by that agent's system.**

ORACLE

Send Event Activities

A Send event activity sends the event directly from the Out Agent to the To Agent without raising the event to the Event Manager. No subscription processing is performed for the event on the sending system.



To Define an Event Activity:

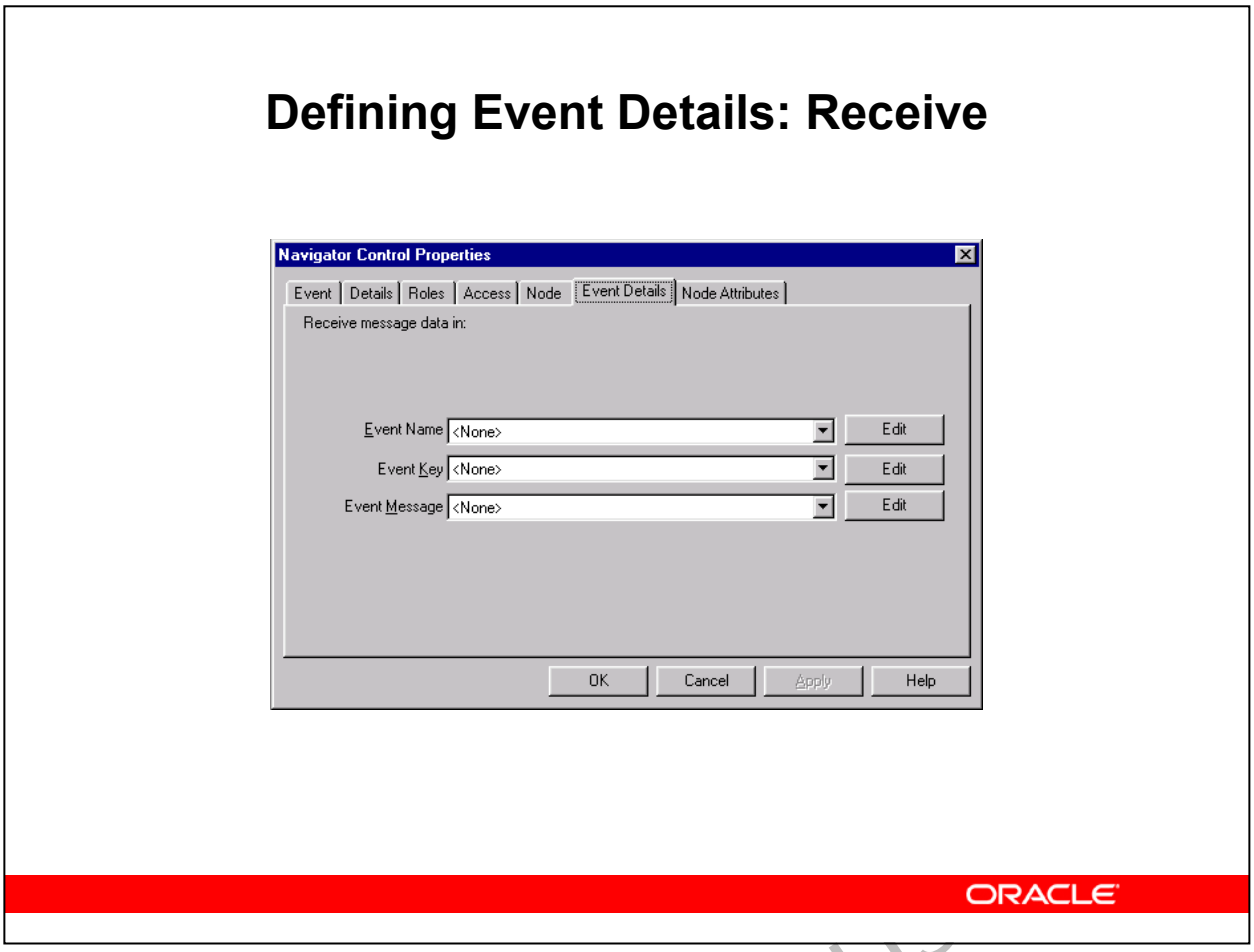
1. In the Oracle Workflow Builder, select the item type that you want in the navigator tree. Then choose New Event from the Edit menu.
2. In the Activity property page, enter an internal name for the activity. The internal name must be all uppercase without any colons or leading or trailing spaces.
3. Enter a display name for the activity.
4. Enter a description of the activity.
5. Choose an icon that identifies the activity.
6. Select the Event Action for the activity.
   - Receive
   - Raise
   - Send
7. If you are defining a Receive event activity, you can optionally enter an event filter to specify the event that the activity can receive.
   - To allow only a specified event for the activity, enter the full internal event name.
   - To allow any event for the activity, leave the Event Filter field blank.

8. Enter an optional cost for the activity. For Raise or Send event activities, you can use the cost to defer long running activities to a background engine.
9. Choose Apply to save your changes.
10. Optionally select the Details tab to display and modify additional activity details.
11. Optionally select the Access tab to set the access levels allowed to modify this event.

Event Details

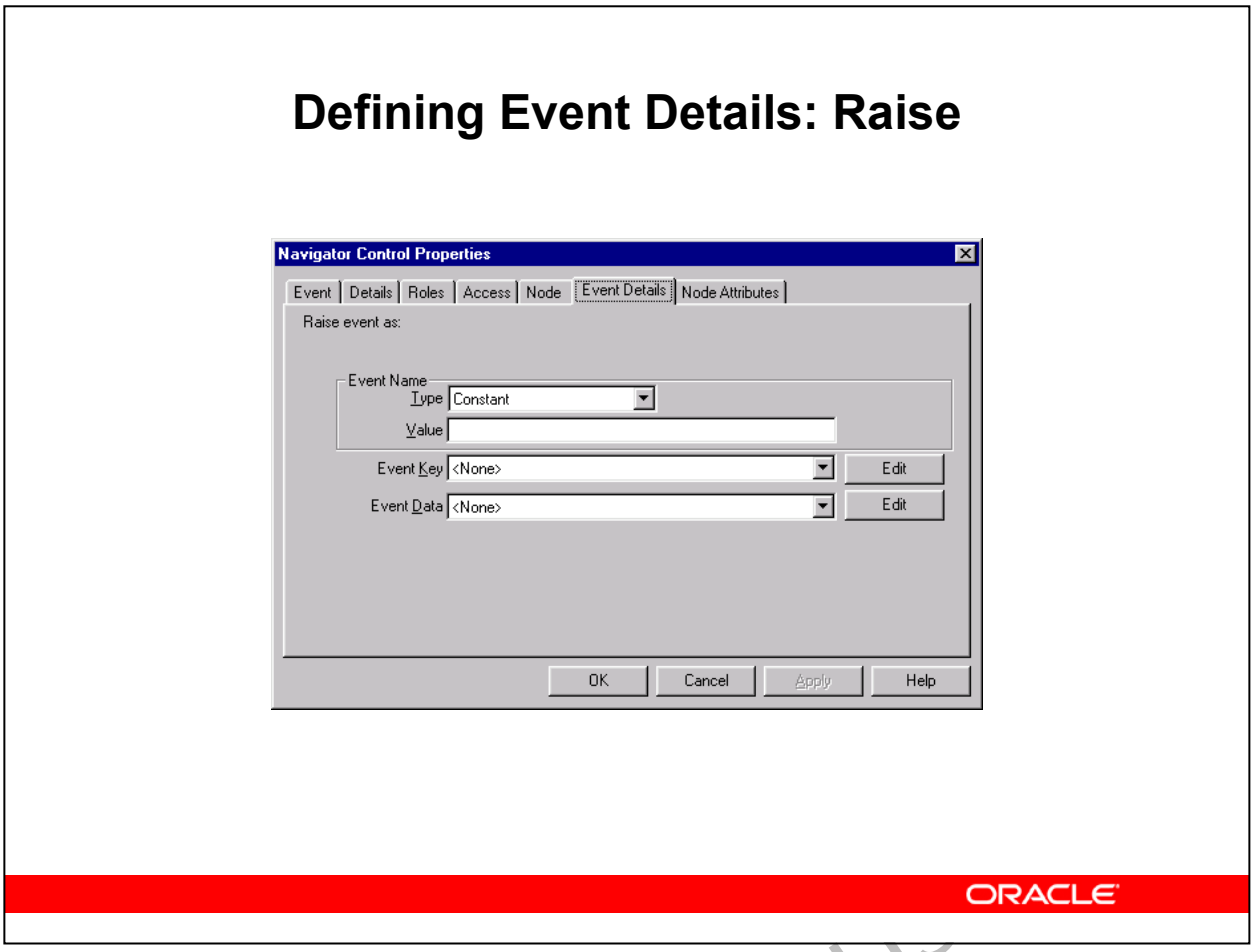
- When you add an event activity node to a workflow process, define event details for the node.
- The event details that are required depend on the event activity's action.

ORACLE



To Define Event Details for a Receive Event Activity Node:

1. In the Oracle Workflow Builder, display the property pages of a Receive event activity node. Select the Event Details tab.
2. Enter the following event details:
   - Event Name—Optionally select an item type attribute of type text where you want to store the event name that the node receives.
   - Event Key—Optionally select an item type attribute of type text where you want to store the event key that the node receives.
   - Event Message—Optionally select an item type attribute of type event where you want to store the event message that the node receives.
3. Choose Apply to save your changes.



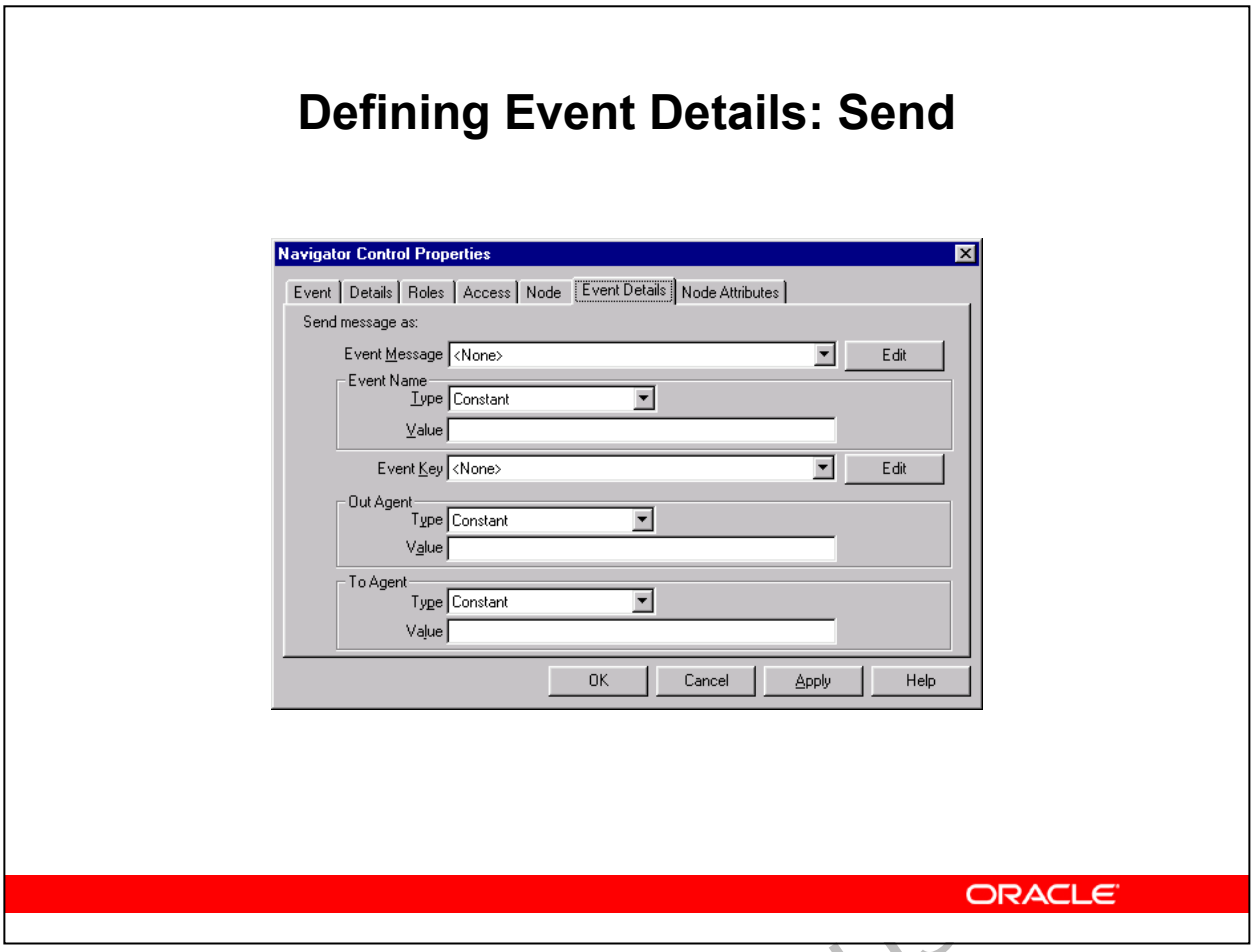
To Define Event Details for a Raise Event Activity Node:

1. In the Oracle Workflow Builder, display the property pages of a Raise event activity node. Select the Event Details tab.
2. Enter the following event details:
   - Event Name—Enter the name of the event that the node raises. You can either specify a constant event name or select an item type attribute of type text that dynamically determines the event name at runtime.
   - Event Key—Select the item type attribute of type text that contains the event key for the event that the node raises.
   - Event Data—Optionally select an item type attribute of type text that contains the event data for the event that the node raises.

   Note: The event name and event key are required for a Raise event activity.

   Note: The maximum length of the data you can enter in a text attribute is 4000 bytes. If the event data exceeds 4000 bytes, you should assign a Generate function in the event definition to generate the event data, rather than providing the event data through a text attribute.

3. Choose Apply to save your changes.



To Define Event Details for a Send Event Activity Node:

1. In the Oracle Workflow Builder, display the property pages of a Send event activity node. Select the Event Details tab.
2. Enter the following event details:
   - Event Message—Select the item type attribute of type event that contains the event message that the node sends.
   - Event Name—Optionally enter the name of the event that the node sends. You can either specify a constant event name or select an item type attribute of type text that dynamically determines the event name at runtime. The event name that you enter here overrides the previous event name value in the event message.
   - Event Key—Optionally select an item type attribute of type text that contains the event key of the event that the node sends. The event key that you enter here overrides the previous event key value in the event message.
   - Out Agent—Optionally enter the outbound agent from which the node sends the event. Specify both the agent name and the system name for the agent using the following format:

     <agent_name>@<system_name>

You can either specify a constant Out Agent name or select an item type attribute of type text that dynamically determines the Out Agent name at runtime. The Out Agent that you enter here overrides the previous outbound agent value in the event message.

- To Agent—Optionally enter the inbound agent to which the node sends the event. Specify both the agent name and the system name for the agent using the following format:

  <agent_name>@<system_name>

  You can either specify a constant To Agent name or select an item type attribute of type text that dynamically determines the To Agent name at runtime. The To Agent that you enter here overrides the previous inbound agent value in the event message.

Note: The Event Message is required for a Send event activity. Additionally, you must either include the To Agent within the event message or specify the To Agent in the event details for this node. If you do not specify a To Agent, the event cannot be sent.

3. Choose Apply to save your changes.

Standard Activities

Use the following standard activities to manage the contents of event messages:

- Get Event Property—Retrieves a property of an event message and stores the property value in an item attribute.
- Set Event Property—Sets the value of a property in an event message.
- Compare Event Property—Compares a property of an event message with a test value.
- XML Get Tag Value—Retrieves the data contained within a particular XML tag set in an event message and stores the data in an item attribute.
- XML Compare Tag Value (Date)—Compares the data contained within a particular XML tag set in an event message with a test date value.
- XML Compare Tag Value (Number)—Compares the data contained within a particular XML tag set in an event message with a test number value.
- XML Compare Tag Value (Text)—Compares the data contained within a particular XML tag set in an event message with a test text value.
- XML Transform—Applies an XML style sheet to the payload of an event message and stores the resulting document in an item attribute.

Note: The XML Get Tag Value, XML Compare Tag Value, and XML Transform activities are external Java function activities and require the Java Function Activity Agent as well as a background engine to complete their processing. External Java function activities are currently only available for the standalone version of Oracle Workflow. This functionality is not currently available for the version of Oracle Workflow embedded in Oracle E-Business Suite.

Practice - Defining Event Activities

Overview

In this practice, you will add a Raise event activity to the Vacation Proposal process you created in the Creating a Workflow Process practice and define another a workflow process that includes a Receive event activity to perform additional processing when the event is received.

- Add a Raise event activity to the Vacation Proposal process.
- Define a new item type that includes a Receive event activity as well as some item attributes related to the event.
- Define the process diagram for the process activity in the new item type. The diagram should include the Receive event activity, a function activity as a placeholder for additional processing, and a standard End activity.
- Define a subscription to send the event to the new workflow process when the event is raised.
- Run the Vacation Proposal process to raise the event, trigger the subscription, and run the new workflow process. You can review the processes in the Workflow Monitor to confirm that the subscription was executed and the processes completed successfully.

Note: Because many students access the system and create objects during this course, you need a way to distinguish between the objects created by you and by your classmates. Therefore, you will be assigned a terminal number by your instructor. Use this number as a prefix wherever you see *XX* included in the name of something you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- The instructor will provide you with the connect string for the class database and the username and password of the Oracle Workflow database account.
- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined in the Global Workflow Preferences page.
- For standalone Oracle Workflow, the instructor will provide you with the URL for the Oracle Workflow home page. The URL is <*webagent*>/wfa_html.home, where <*webagent*> is the base URL of the web agent configured for Oracle Workflow in your Web server.
- For Oracle Workflow embedded in Oracle E-Business Suite, the instructor will provide you with the name of a Workflow administrator responsibility that includes Event

Manager functionality. The username you use to log in should have this responsibility assigned to it.

- The instructor will provide you with the names of users that you can assign as the requestor and approver in the Vacation Proposal process.

Tasks

1. Add a Raise event activity to the Vacation Proposal process.

   Solution:

   1. Start the Oracle Workflow Builder.

   2. From the File menu, choose Open to open the wfvacXX.wft data store you defined in the Creating a Workflow practice.

   3. In the navigator tree, select your item type.

   4. From the Edit menu, choose New > Attribute.

   5. Define the following properties for the item attribute:
      - Internal Name: EVENTNAME
      - Display Name: Event Name
      - Description: Event name
      - Type: Text
      - Default Value: *XX*.oracle.workflow.bes.vacation.scheduled

      Choose OK.

   6. From the Edit menu, choose New > Attribute.

   7. Define the following properties for the item attribute:
      - Internal Name: EVENTKEY
      - Display Name: Event Key
      - Description: Event key
      - Type: Text

      Choose OK.

   8. From the Edit menu, choose New > Attribute.

   9. Define the following properties for the attribute:
      - Internal Name: EVENTDATA
      - Display Name: Event Data
      - Description: Event data
      - Type: Text

Choose OK.

10. From the Edit menu, choose New > Event.

11. Define the following properties for the event:
    - Internal Name: RAISE_VAC_EVENT
    - Display Name: Raise Vacation Scheduled Event
    - Description: Raise Vacation Scheduled event

12. Leave the icon and cost set to the default values.

13. Select Raise as the event action.

14. Choose OK.

15. Open the process diagram window for the Vacation Proposal process.

16. Delete the transition between the Update Vacation Schedule node and the End (Approve) node.

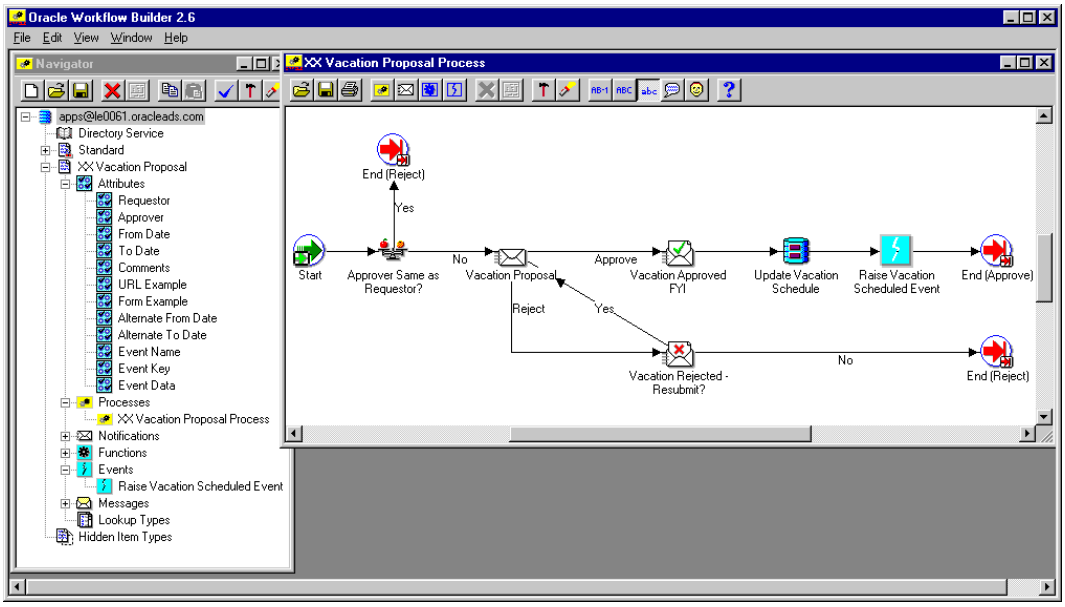
17. Drag the Raise Vacation Schedule Event activity into the process diagram and position it between the Update Vacation Schedule node and the End (Approve) node.

18. Open the property pages for the Raise Vacation Schedule Event node, and choose the Event Details tab. Select the Event Name attribute as the Event Name value, the Event Key attribute as the Event Key value, and the Event Data attribute as the Event Data value.

19. Draw transitions from the Update Vacation Schedule node to the Raise Vacation Schedule Event node, and from the Raise Vacation Schedule Event node to the End (Approve) node.

20. From the File menu, choose Save to save your work to your workflow definition file.

21. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



22. Close your data store.

2. Define a new item type that includes a Receive event activity as well as some item attributes related to the event.

Solution:

23. From the File menu, choose Quick Start Wizard.

24. Define the following properties for the new item type:
    - Internal Name: *XX*VACSAL
    - Display Name: *XX* Vacation Salary
    - Persistence Type: Temporary
    - Number of Days: 5

    Define the following properties for the process:
    - Internal Name: *XX*_VACATION_SALARY
    - Display Name: *XX* Vacation Salary Process

    Choose OK.

25. Close the Vacation Salary process diagram window. You will define the diagram for this process in the next task.

26. In the Navigator window, select the Vacation Salary item type.

27. From the Edit menu, choose New > Attribute.

28. Define the following properties for the item attribute:
    - Internal Name: EVENTNAME
    - Display Name: Event Name

- Description: Event name
- Type: Text

Choose OK.

29. From the Edit menu, choose New > Attribute.

30. Define the following properties for the item attribute:
    - Internal Name: EVENTKEY
    - Display Name: Event Key
    - Description: Event key
    - Type: Text

    Choose OK.

31. From the Edit menu, choose New > Attribute.

32. Define the following properties for the attribute:
    - Internal Name: EVENTMESSAGE
    - Display Name: Event Message
    - Description: Event message
    - Type: Event

    Choose OK.

33. From the Edit menu, choose New > Event.

34. Define the following properties for the event:
    - Internal Name: REC_VAC_EVENT
    - Display Name: Receive Vacation Scheduled Event
    - Description: Receive Vacation Scheduled event

35. Leave the icon and cost set to the default values.

36. Select Receive as the event action.

37. In the Event Filter field, enter *XX*.oracle.workflow.bes.vacation.scheduled, the event that you defined in the Defining an Event practice.

38. Choose OK.

39. From the File menu, choose Save As and save your new data store and item type to a workflow definition file named wfbes*XX*.wft.

3. Define the process diagram for the process activity in the new item type.

Solution:

40. In the navigator tree, select the Vacation Salary process.

41. Double-click the Vacation Salary process to display the process diagram window.

42. Delete the default Start activity from the process diagram.

43. Drag the Receive Vacation Scheduled Event activity into the process diagram.

44. Double-click the Receive Vacation Scheduled Event activity node and choose the Node tab.

45. In the Start/End field, select Start.

46. Choose the Event Details tab.

47. In the Event Name field, select the Event Name item attribute.

48. In the Event Key field, select the Event Key item attribute.

49. In the Event Message field, select the Event Message item attribute.

50. Choose OK.

51. Click the New Function button, position the cross-hair cursor between the Receive Vacation Scheduled Event node and the End node, and click again to create a new function activity node there and open the property pages for the node. This activity represents a point in the process where you could add additional processing for the event. For this practice, you will define a placeholder function that does not actually perform any processing.

52. Define the following properties for the function activity:
    - Internal Name: PAY_VACATION_SALARY
    - Display Name: Pay Vacation Salary
    - Description: Pay salary for vacation dates from vacation account

53. Select the icon currepor.ico for the activity.

54. Enter the package and procedure name WF_STANDARD.NOOP as a placeholder in the Function Name field. WF_STANDARD.NOOP is a standard Workflow PL/SQL procedure that sets the resultout to wf_engine.eng_completed and returns. This function will have no result.

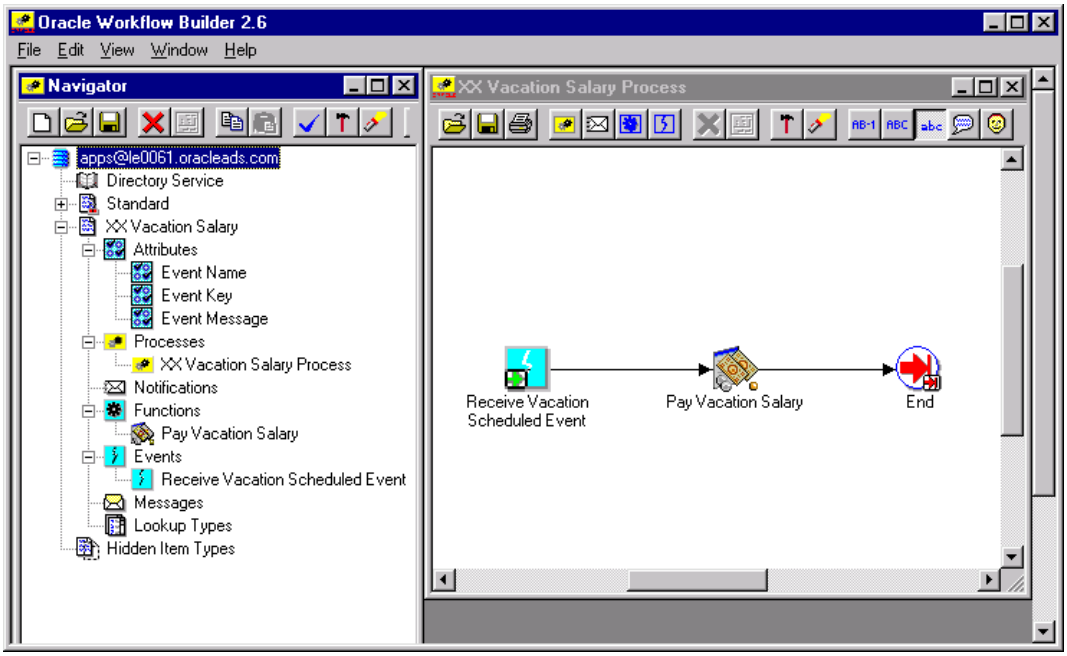
55. Draw a transition from the Receive Vacation Scheduled Event activity to the Pay Vacation Salary activity.

56. Draw a transition from the Pay Vacation Salary activity to the End activity.

57. Double-click the End activity node and choose the Node tab. Ensure that the Start/End field is set to End. Choose OK.

58. From the File menu, choose Save to save your work to your workflow definition file.

59. From the File menu, choose Save As and save your item type to the class database, using the database username, password, and connect string provided by the instructor.



4. Define a subscription to send the event to the new workflow process when the event is raised.

Solution:

60. Use a web browser to connect to the Oracle Workflow home page with the URL provided by the instructor for standalone Oracle Workflow, or to a Workflow administrator responsibility provided by the instructor for Oracle Workflow embedded in Oracle E-Business Suite. Log in as a user with workflow administrator privileges.

61. Choose the Event Subscriptions link for standalone Oracle Workflow or the Add Event Subscriptions link for Oracle Workflow embedded in Oracle E-Business Suite.

62. In the Event Subscriptions page, choose the Add Subscription button to open the Edit Subscription page.

63. In the System field, select the local system as the subscriber.

64. In the Source Type field, select Local.

65. In the Event Filter field, select the *XX*.oracle.workflow.bes.vacation.scheduled event that you defined in the Defining an Event practice.

66. Leave the Source Agent field blank.

67. Enter 30 in the Phase field.

68. In the Status field, select Enabled.

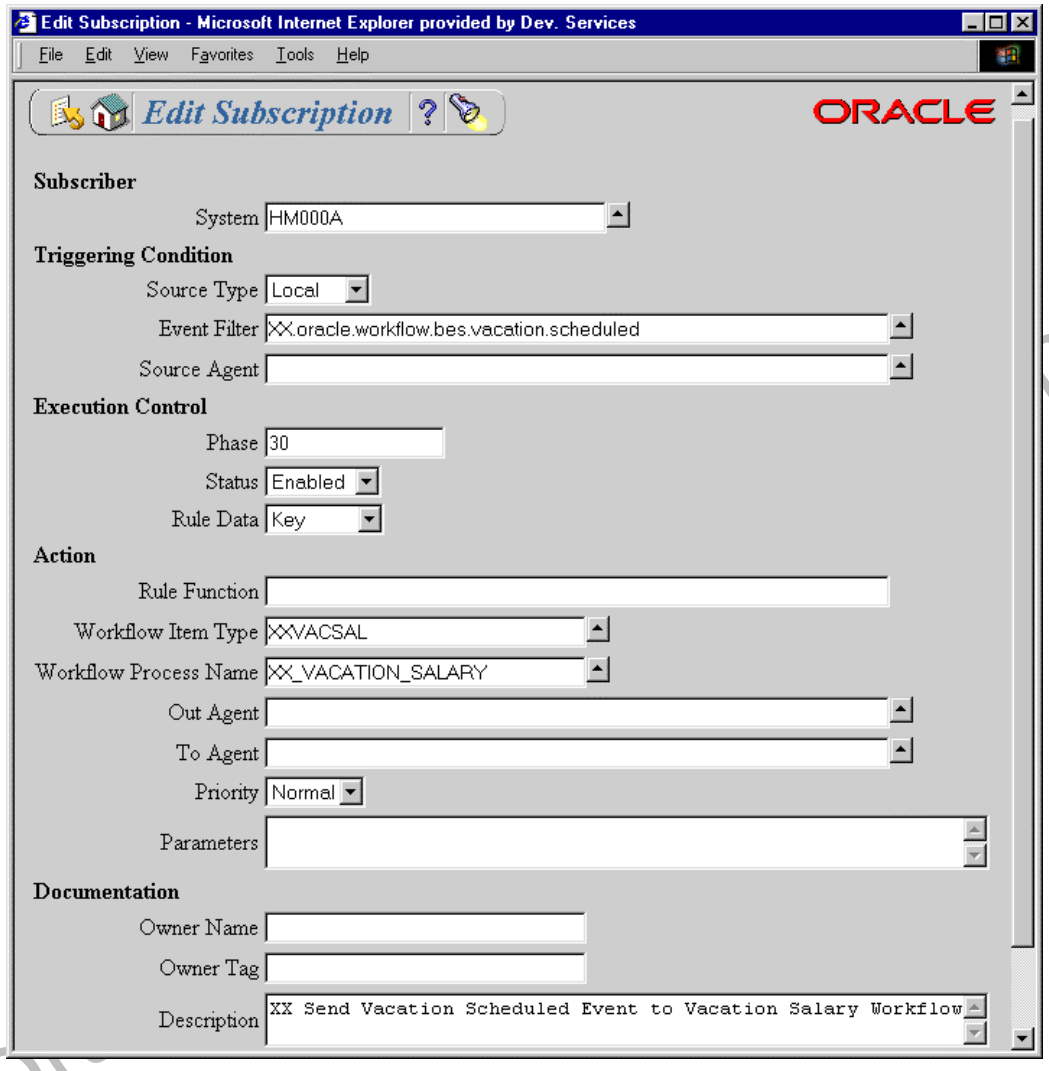
69. In the Rule Data field, select Key.

70. Leave the Rule Function field blank to use the default rule function to send the event to a workflow process.

71. In the Workflow Item Type and Workflow Process Name fields, select your *XX*VACSAL item type and *XX*_VACATION_SALARY process.

72. Leave the Out Agent, To Agent, Parameters, Owner Name, and Owner Tag fields blank. Leave the Priority field set to the default value, which is Normal.

73. In the Description field, enter *XX* Send Vacation Scheduled Event to Vacation Salary Workflow.



74. Choose the Submit button to save the subscription.

5. Run the Vacation Proposal process to raise the event, trigger the subscription, and run the new workflow process.

   Solution:

   75. Use the Launch Processes page to launch your Vacation Salary workflow process and test your work. You can use the Notifications Worklist to view the notifications sent by the process and approve the vacation proposal, and use the Workflow Monitor to review the status of the Vacation Proposal process and the Vacation Salary process.
       - When you launch the Vacation Proposal process, enter a unique event key such as XX131E in the Event Key field and `<DATA>Vacation has been scheduled</DATA>` in the Event Data field.
       - Run the process and approve the vacation proposal.
       - When the Vacation Proposal process raises the event, the Business Event System should run all local subscriptions to the event, including the subscription which sends the event to start the Vacation Salary process. You can review both processes in the Workflow Monitor to confirm that the processes completed successfully.

Summary

In this lesson, you should have learned how to:

- **Define event activities.**
- **Define event details for event activity nodes.**
- **Send an event to a workflow process to start or continue the process.**
- **Use standard activities to manage information from event messages.**

ORACLE