R12.x Implement Oracle Workflow

Activity Guide

D58320GC10
Edition 1.0
October 2009
D66010

Authors

Clara Jaeckel

Technical Contributors and Reviewers

Leta Davis, Donna Johnson, Shivasmruthi Narayanasamy

This book was published using: oracle*tutor*

Table of Contents

Preface

Profile

Before You Begin This Course

- Thorough knowledge of Oracle Database and Oracle Application Server technology
- Thorough knowledge of Oracle E-Business Suite

Prerequisites

- *Oracle Database: Introduction to SQL*
- *Oracle Database: Program with PL/SQL*

How This Course Is Organized

R12.x 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 Administrator's Guide	E12903
Oracle Workflow Developer's Guide	E12905
Oracle Workflow User's Guide	E12906
Oracle Workflow API Reference	E12904
Oracle Workflow Client Installation Guide	E12779

Additional Publications

- System release bulletins
- Installation and user's guides
- Read-me files
- International Oracle User's Group (IOUG) articles
- *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.
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 Oracle Application 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** then **Entry** then **Invoice Batches Summary**.

2. (M) From the menu, select **Query** then **Find**.

3. (B) Click the **Approve** button.

Notations:

(N) = Navigator

(M) = Menu

(T) = Tab

(B) = Button

(I) = Icon

(H) = Hyperlink

(ST) = Sub Tab

Typographical Conventions in Oracle Application 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

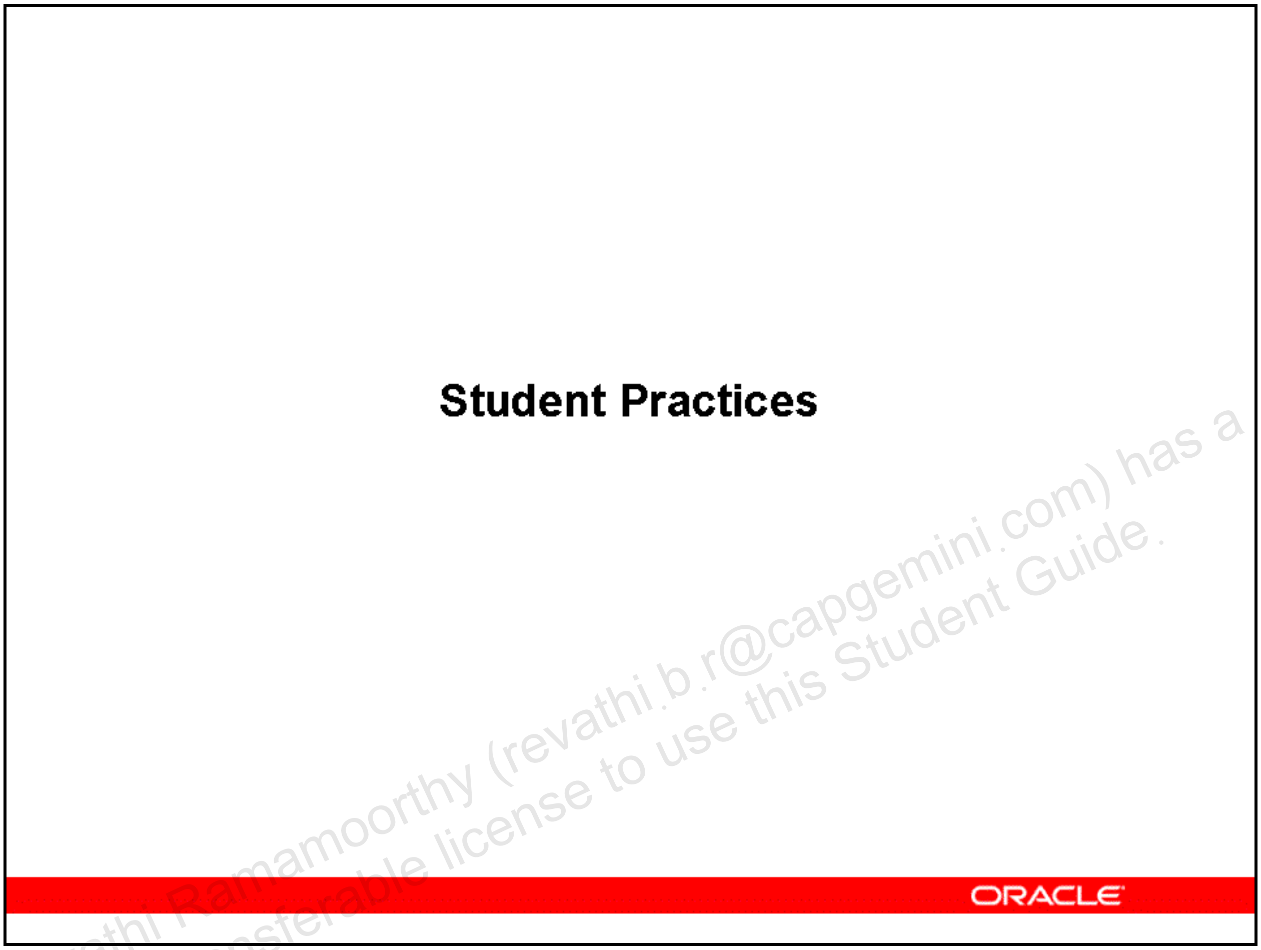
—represents the following sequence of actions:

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.

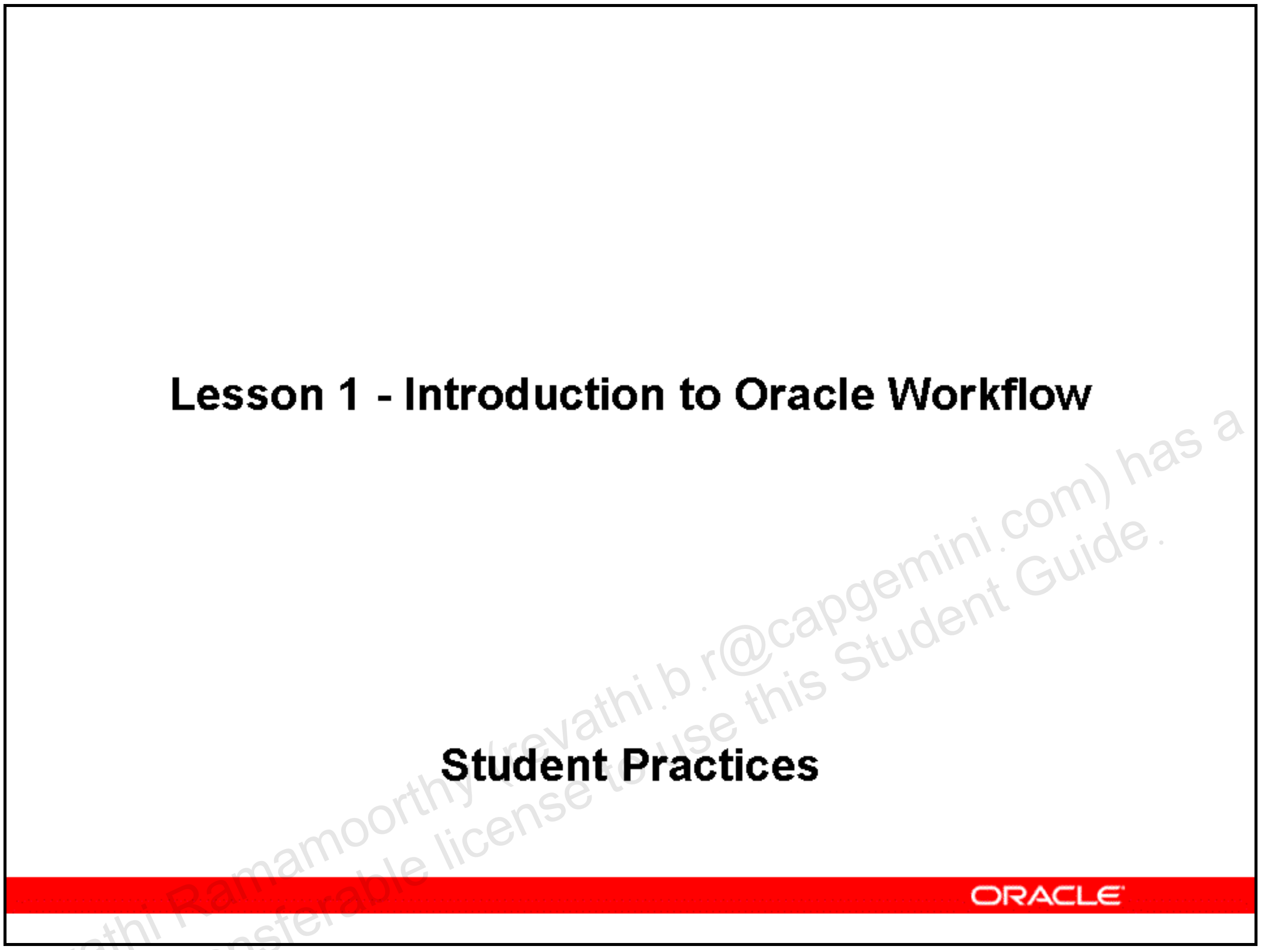
Student Practices

Chapter 30

Student Practices



Student Practices



Lesson 1 - Introduction to Oracle Workflow

Student Practices

Guided Demonstration - Loading and Running a Workflow Process

1. Start the Oracle Workflow Builder.

2. From the File menu, select Open to open the sample solution file named wfvacXX_062.wft from the file system.

3. Display the process diagram for the sample process.

4. Save the workflow definition to the class database using File > Save As. Then close the data store.

5. Use a Web browser to connect to a Workflow administrator responsibility. Log in as a user with workflow administrator privileges.

6. Click the Developer Studio link. In the Search region of the Developer Studio page, enter the name of the XX Vacation Proposal workflow item type and click Go. Then, in the Results region, click the Run icon for the XX Vacation Proposal item type.

7. Enter a process owner, item key, user key, requestor, approver, from date, and to date. The requestor and approver should have Workflow administrator and user responsibilities assigned to them.

8. Click the Submit button. A confirmation message appears. Click OK.

9. Select the Status Monitor tab to review the process status in the Status Monitor Web pages.

   - In the Search region, enter the XX Vacation Proposal item type and the item key you chose, and click Go.

   - In the Results region, select your process and click the Activity History button to review the process activities.

   - Click the Status Diagram button to review the graphical diagram of the status of the process.

10. Log off and log in again as the approver. You can use either a Workflow administrator responsibility or Workflow user responsibility.

11. Click the Notifications link.

12. In the Worklist, select the subject line for the Vacation Proposal notification sent by your process to open the notification message.

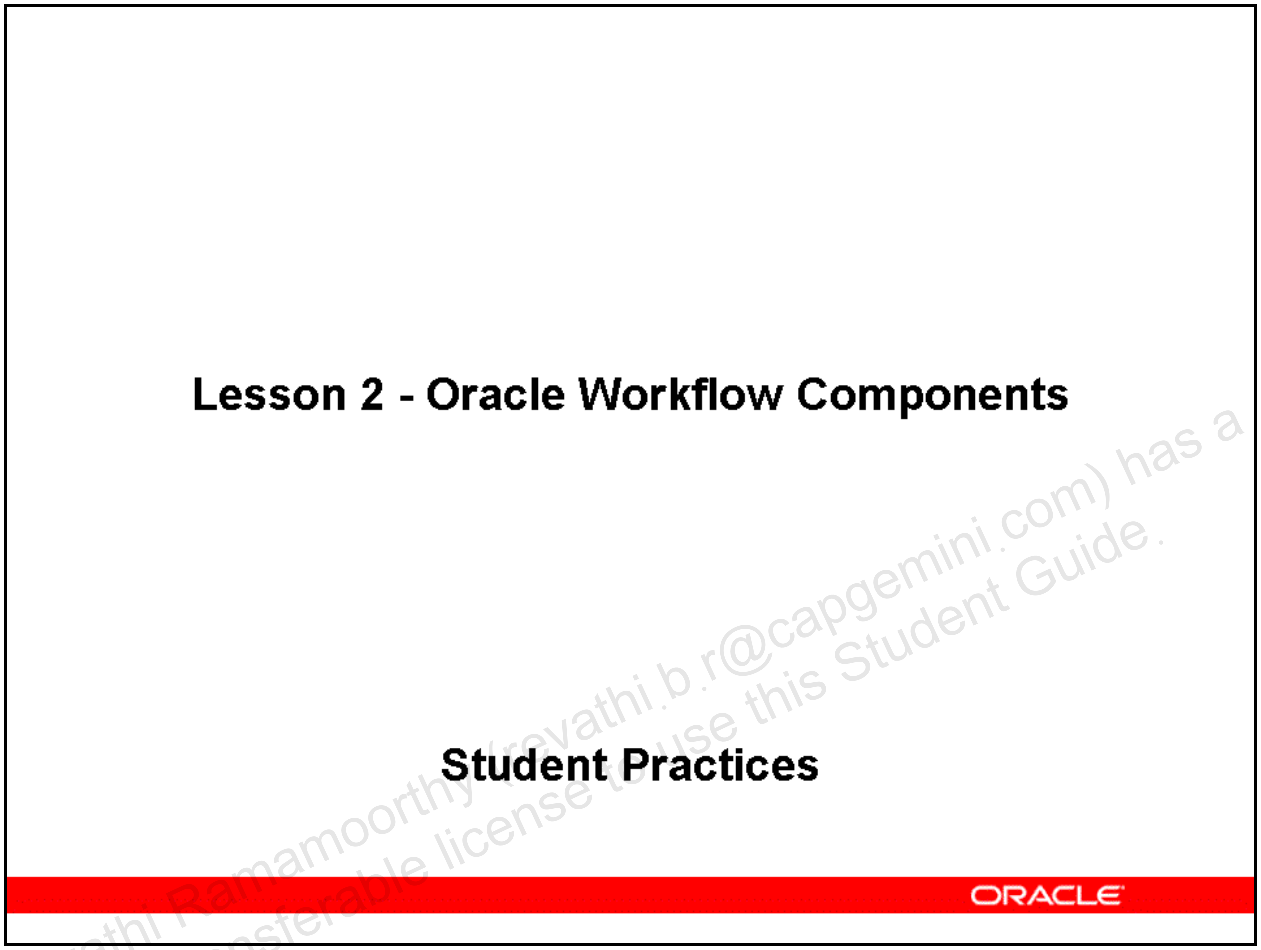
13. On the Notification Details page, approve or reject the proposal.

14. Log off and log in as the requestor. To review the updated status of the process in the Status Monitor Web pages, select the Status Monitor tab. Then search for the process with the XX Vacation Proposal item type and your item key, and view the status diagram again.

15. Open the Workflow Builder again and make a change to the process diagram, such as adding a function activity by dragging and dropping the Noop function from the Standard item type into the process. Then save the definition to the class database again.

16. Repeat steps 5 through 9 to show the new version of the workflow definition in use.

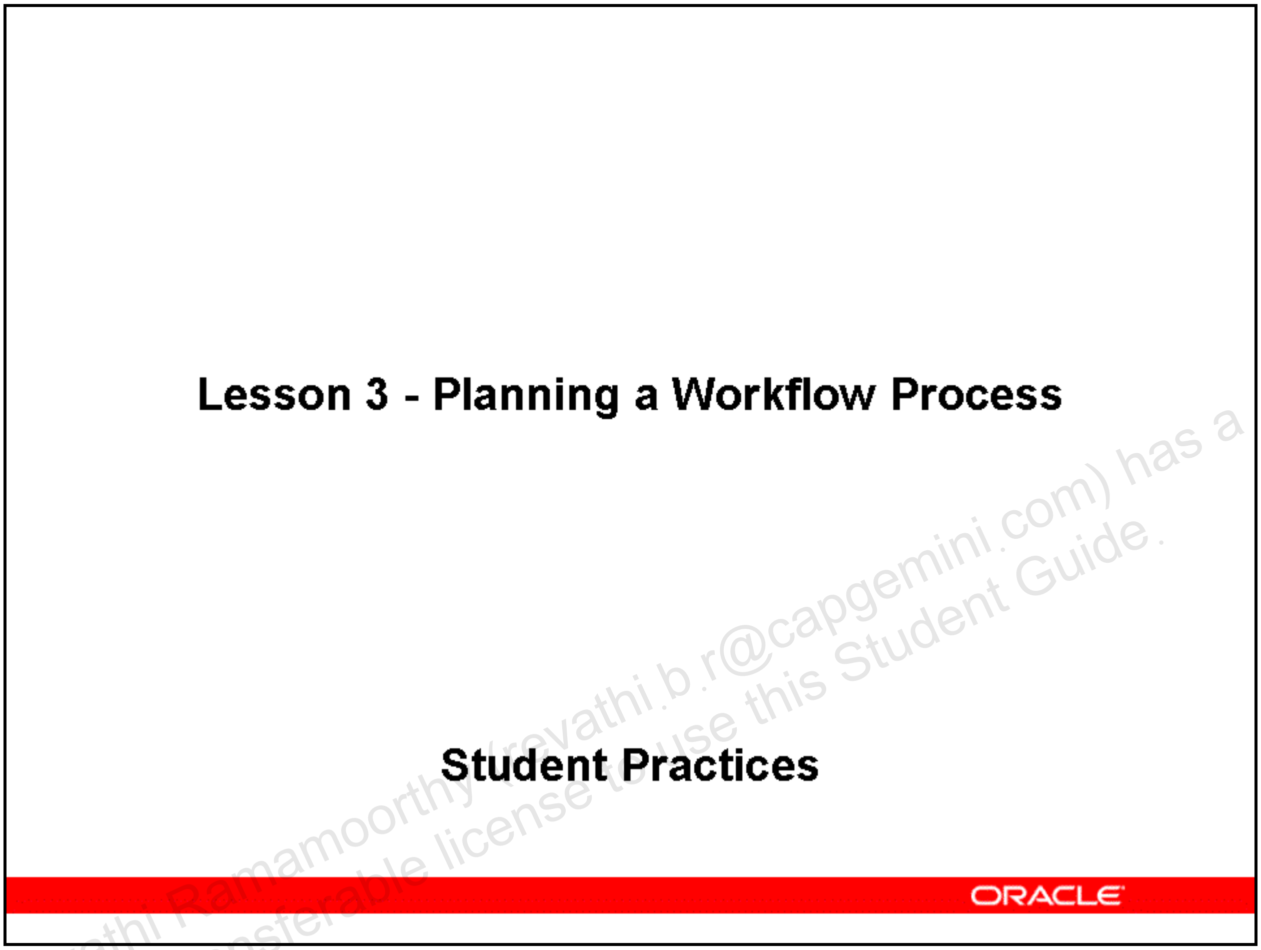
Lesson 2 - Oracle Workflow Components



Lesson 2 - Oracle Workflow Components

Student Practices

There are no practices or demonstrations in this lesson.



Lesson 3 - Planning a Workflow Process

Student Practices

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

Planning a Workflow Process

On paper, sketch a diagram of a vacation proposal process. The business process is as follows:

- A requestor sends a vacation proposal to an approver, prompting the proposal recipient to approve or reject the request.
- If the approver approves the proposal, the requestor receives a notification reporting that the vacation is approved.
- If the approver rejects the proposal, the requestor receives a notification reporting that the vacation is rejected.

Solution – Planning a Workflow Process

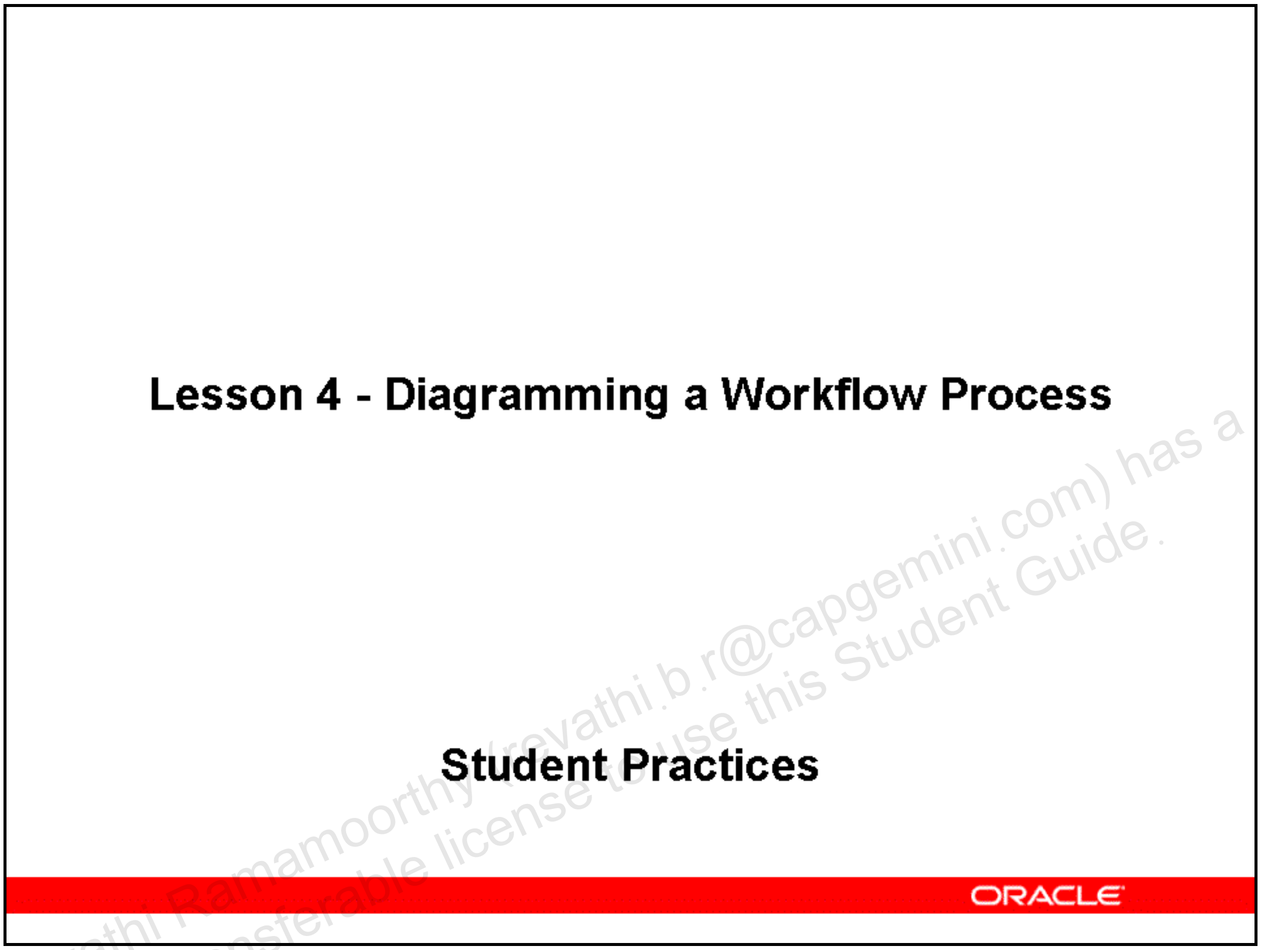
Planning a Workflow Process

1. Determine what activities will make up the vacation proposal process.

2. Make a sketch of a diagram on paper, 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.



Lesson 4 - Diagramming a Workflow Process

Student Practices

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 that you create will be a prototype of the vacation proposal process that 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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Tasks

Creating a Workflow Process

Create a workflow process in the Oracle Workflow Builder using the Quick Start Wizard. Your workflow process should model the vacation proposal process that you sketched in the Planning a Workflow Process practice.

Solution – Creating a Workflow Process

Creating a Workflow Process

1. Start the Oracle Workflow Builder.

2. From the File menu, select 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

   - Click OK. The *XX* Vacation Proposal Process window appears, populated with the Start and End nodes.

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 click an icon on the Notification tab. 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 click 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 select the Node tab. In the Start/End field, select End and then click OK.

6. In the process diagram, draw transitions between the nodes in the process to indicate the process flow. To draw a transition, right-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, select the appropriate result from the results menu that appears. Select Approve for the Vacation Approved FYI notification activity and Reject for the Vacation Rejected FYI notification activity.

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

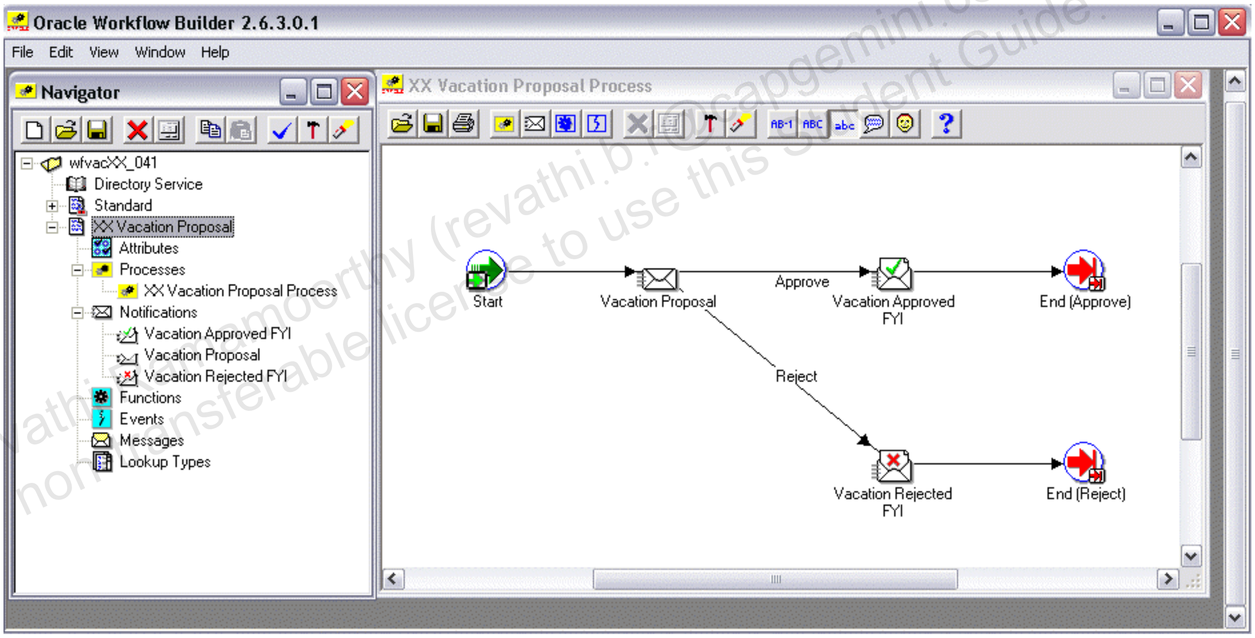
8. Select the Activity tab and select Approval as the result type for the process. Then click OK.

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

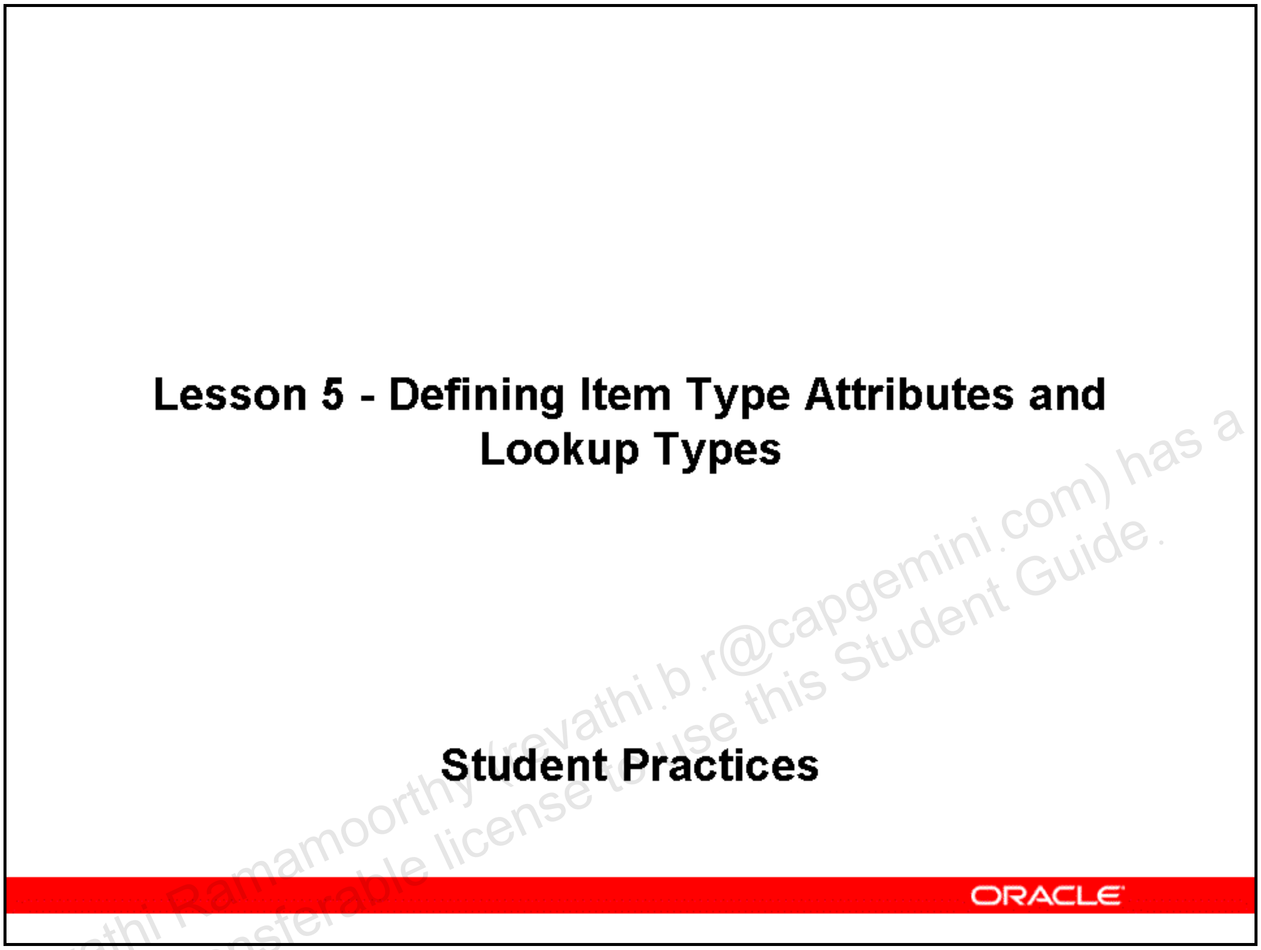
10. Double-click the End node that follows the Vacation Approved FYI node and select the Node tab. Select Approve in the Result field and then click OK.

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, select 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.



Lesson 5 - Defining Item Type Attributes and Lookup Types



Lesson 5 - Defining Item Type Attributes and Lookup Types

Student Practices

Practice - Defining Item Type Attributes

Overview

In this practice, you will define item type attributes for the Vacation Proposal item type that 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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Tasks

Defining Item Type Attributes

Define item type attributes for the Vacation Proposal item type that you created in the Creating a Workflow Process practice.

Solution – Defining Item Type Attributes

Defining Item Type Attributes

1. Start the Oracle Workflow Builder.

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

3. In the navigator tree, select your *XX* Vacation Proposal item type.

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

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

   - Click OK.

6. In the navigator tree, right click the Attributes branch under your *XX* Vacation Proposal item type, and select New Attribute.

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

   - Click OK.

8. In the navigator tree, select the Attributes branch under your *XX* Vacation Proposal item type, and click the New Object button in the Navigator window.

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

   - Click OK.

10. From the Edit menu, select 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

- Click OK.

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

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

- Click OK.

14. Create a URL item attribute to test how Oracle Workflow includes URL links in notifications. From the Edit menu, select 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

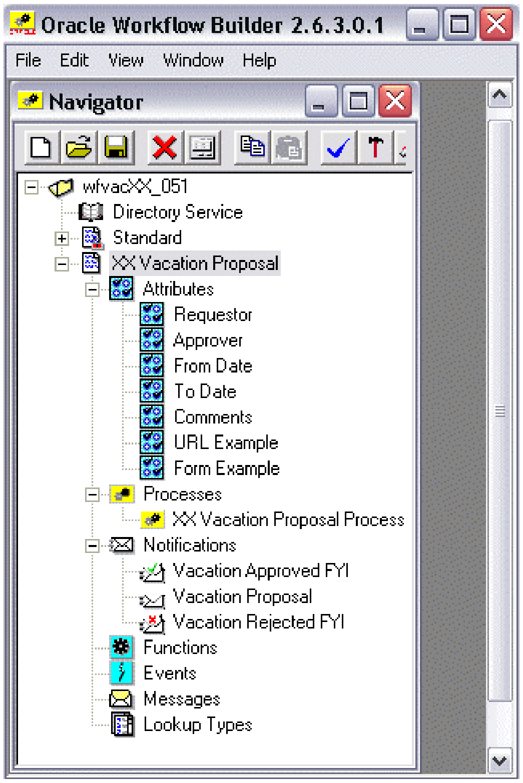
- Click OK.

16. Create a form item attribute to test how Oracle Workflow attaches Oracle E-Business Suite forms to notifications. From the Edit menu, select New > Attribute.

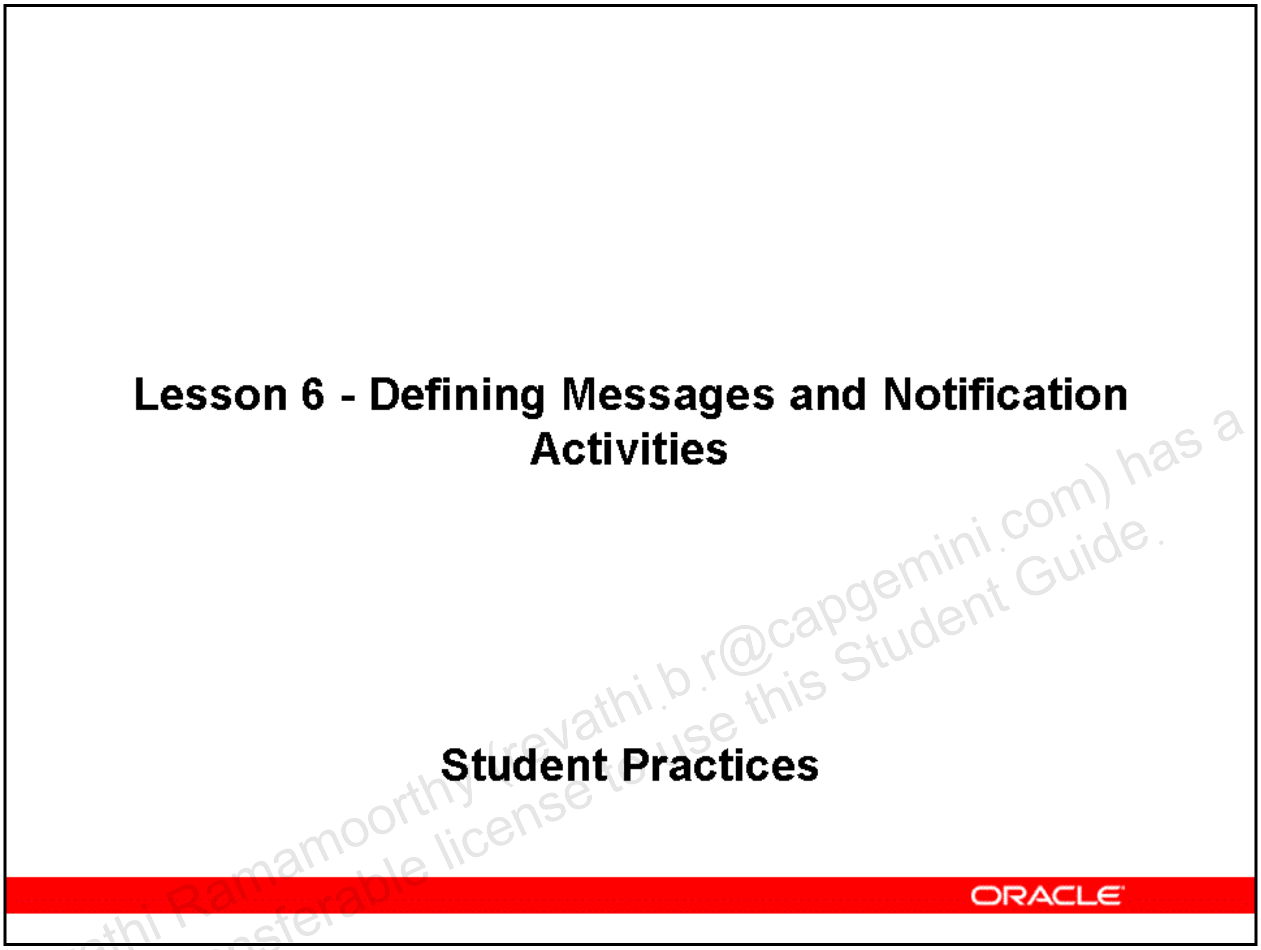
17. Define the following properties for the item attribute:
    - Internal Name: FORM_EXAMPLE
    - Display Name: Form Example
    - Type: Form
    - Default Value: FND_FNDSCAUS

- Click OK.

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



- wfvacXX_051
  - Directory Service
  - Standard
  - XX Vacation Proposal
    - Attributes
      - Requestor
      - Approver
      - From Date
      - To Date
      - Comments
      - URL Example
      - Form Example
    - Processes
      - XX Vacation Proposal Process
    - Notifications
      - Vacation Approved FYI
      - Vacation Proposal
      - Vacation Rejected FYI
    - Functions
    - Events
    - Messages
    - Lookup Types



Lesson 6 - Defining Messages and Notification Activities

Student Practices

Practice - Defining Messages

Overview

In this practice, you will define messages for the Vacation Proposal item type that 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Tasks

Defining Messages

Define messages for your item type.

Solution – Defining Messages

Defining Messages

1. Start the Oracle Workflow Builder.

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

3. In the navigator tree, select your *XX* Vacation Proposal item type.

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

5. Define a Vacation Proposal message that informs the approver of the vacation proposal.

   - On the Message tab, enter VACATION_PROPOSAL as the internal name, enter "Vacation Proposal" as the display name, and set the message priority to Normal.

   - On the Body tab, enter the following message text body including the message attribute tokens for the requestor, approver, and the start and end dates of the proposed vacation:

     ```
     The following vacation proposal requires your approval.

     Requestor: &REQUESTOR

     Approver: &APPROVER

     From Date: &FROM_DATE

     To Date: &TO_DATE
     ```

   - On the Result tab, enter "Approve" as the display name, enter "Please approve or reject this proposal" as the description, and 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, select New > Message.

8. Define a Vacation Rejected that informs the requestor that the vacation proposal was rejected.

   - On the Message tab, enter VACATION_REJECTED as the internal name, enter "Vacation Rejected" as the display name, and set the message priority to Low.

- On the Body tab, enter the following message text body including the message attribute tokens for the requestor, approver, comments, and the start and end dates of the proposed vacation:

  ```
  The following vacation proposal was rejected.

  Requestor: &REQUESTOR

  Approver: &APPROVER

  From Date: &FROM_DATE

  To Date: &TO_DATE

  Comments: &COMMENTS
  ```

- 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, select New > Message.

10. Define a Vacation Approved message that informs the requestor that the vacation proposal was approved.

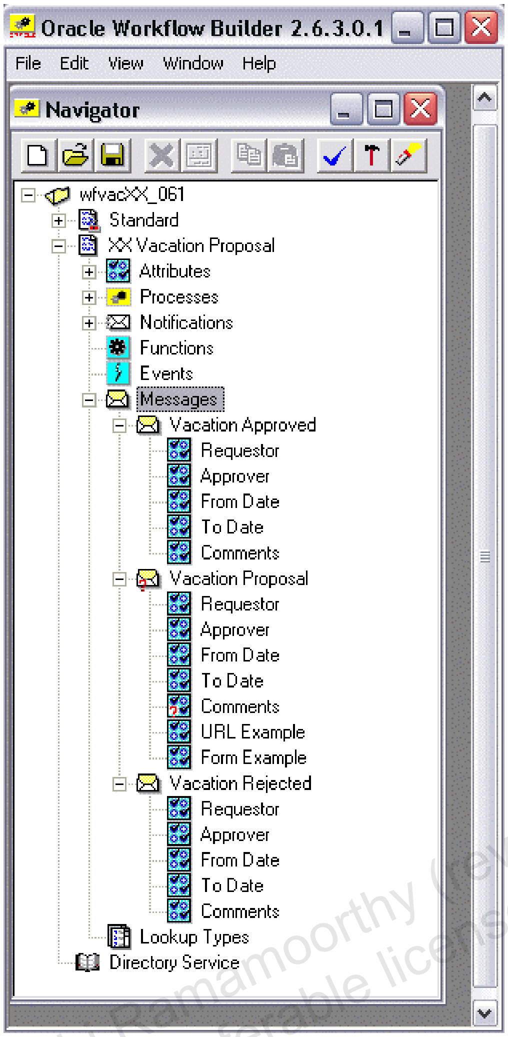
    - On the Message tab, enter VACATION_APPROVED as the internal name, enter "Vacation Approved" as the display name, and set the message priority to High.

    - On the Body tab, enter the following message text body including the message attribute tokens for the requestor, approver, comments, and the start and end dates of the proposed vacation:

      ```
      The following vacation proposal was approved.

      Requestor: &REQUESTOR

      Approver: &APPROVER

      From Date: &FROM_DATE

      To Date: &TO_DATE

      Comments: &COMMENTS
      ```

    - Drag and drop the appropriate item attributes onto the message to create the corresponding message attributes with those item attributes as their default values.

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


Oracle Workflow Builder 2.6.3.0.1
File Edit View Window Help
Navigator
wfvacXX_061
Standard
XX Vacation Proposal
Attributes
Processes
Notifications
Functions
Events
Messages
Vacation Approved
Requestor
Approver
From Date
To Date
Comments
Vacation Proposal
Requestor
Approver
From Date
To Date
Comments
URL Example
Form Example
Vacation Rejected
Requestor
Approver
From Date
To Date
Comments
Lookup Types
Directory Service

Practice - Defining Notification Activities

Overview

In this practice, you will define a special message attribute and complete the notification activity definitions for the Vacation Proposal item type that 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.

Tasks

Defining a Special Message Attribute

Define a special message attribute to hide the Reassign button in the Notification Details page for one of the messages that you created in the Defining Messages practice.

Defining Notification Activities

Use the messages that you created in the Defining Messages practice to complete the notification activity definitions for the item type.

Solution – Defining Notification Activities

Defining a Special Message Attribute

1. Start the Oracle Workflow Builder.

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

3. In the Navigator window, select the Vacation Proposal message. Then, from the Edit menu, select New > Attribute.

4. Define the following properties for the message attribute:
   - Internal Name: #HIDE_REASSIGN
   - Display Name: Hide Reassign Button
   - Type: Text
   - Source: Send
   - Default Type: Constant
   - Default Value: B

- Click OK. Defining the special #HIDE_REASSIGN message attribute and setting its value to B will hide the Reassign button in the Notification Details page for the notification that sends this message. This attribute will also prevent users from reassigning the notification in a mass reassignment from the Advanced Worklist, Personal Worklist, and self-service home page, but still allow reassignment through vacation rules.

Defining Notification Activities

5. In the Navigator window, double-click the Vacation Proposal notification activity to open its property window. Select Vacation Proposal in the Message field, and click OK.

6. In the Navigator Window, double-click the Vacation Approved FYI notification activity to open its property window. Select Vacation Approved in the Message field, and click OK.

7. In the Navigator Window, double-click the Vacation Rejected FYI notification activity to open its property window. Select Vacation Rejected in the Message field, and click OK.

8. Open the process diagram window for the process.

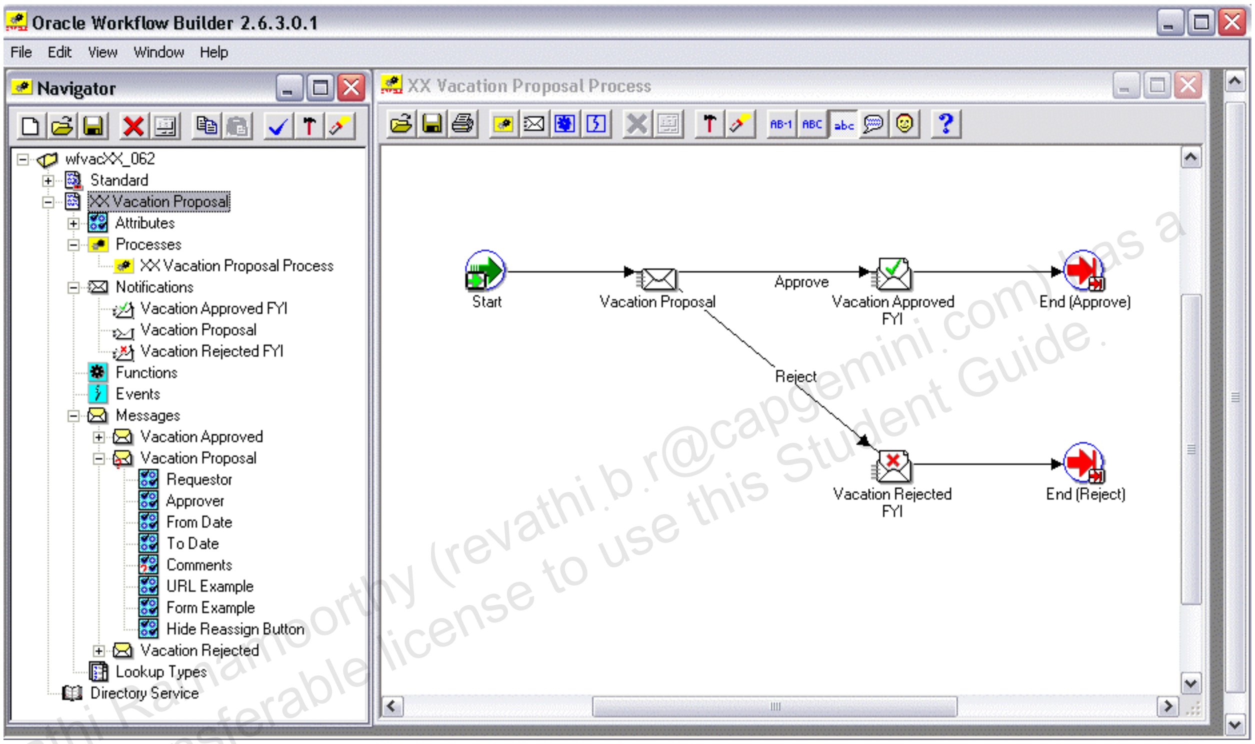
9. Double-click the Vacation Proposal node and select the Node tab. Set the performer for the node to the Approver item attribute.

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

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

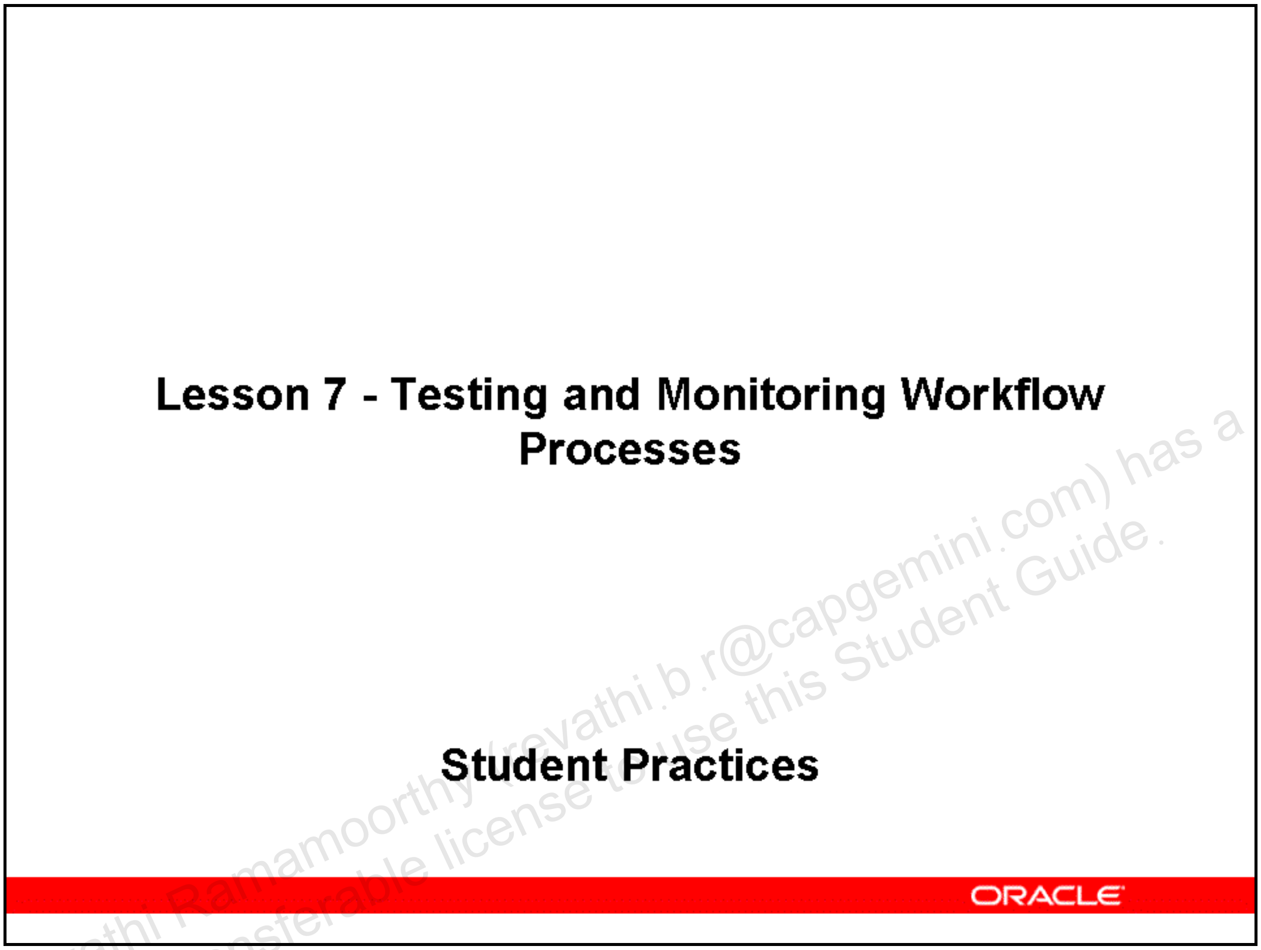
12. In the Navigator window, click the Verify button to verify your workflow. Because you have now defined the underlying components for your process, the Workflow Builder should not display any warnings. Click OK.

13. From the File menu, select Save to save your work to your workflow definition file.



14. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

15. Close the data store.



Lesson 7 - Testing and Monitoring Workflow Processes

Student Practices

Practice - Running a Workflow Process

Overview

In this practice, you will run the Vacation Proposal workflow process that you defined in previous practices. You can use the Status Monitor Web pages to review the status of 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the username and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them. Additionally, if you attached the Form Example attribute to the Vacation Proposal message, the user name for the approver should have the System Administrator responsibility assigned to it to be able to view the attached form.

Tasks

Running a Workflow Process

Run the workflow process using the Developer Studio.

Solution – Running a Workflow Process

Running a Workflow Process

1. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

2. Click the Developer Studio link. In the Search region of the Developer Studio page, select the name of your *XX* Vacation Proposal item type and click Go. Then, in the Results region, click the run icon for the *XX* Vacation Proposal item type.

3. In the Workflow Owner field, select the role that you want to use as the requestor for the vacation proposal.

4. Enter a unique item key such as *XX*071 in the Item Key field.

5. Ensure that your *XX* Vacation Proposal Process is displayed in the Process field.

6. Next, enter starting values for the necessary item attributes. In the Requestor field, select the role that you want to use as the requestor for the vacation proposal.

7. In the Approver field, select the role that you want to use as the approver for the vacation proposal.

8. In the From Date field, enter the vacation from date.

9. In the To Date field, enter the vacation to date.

10. Leave the Comments field blank.

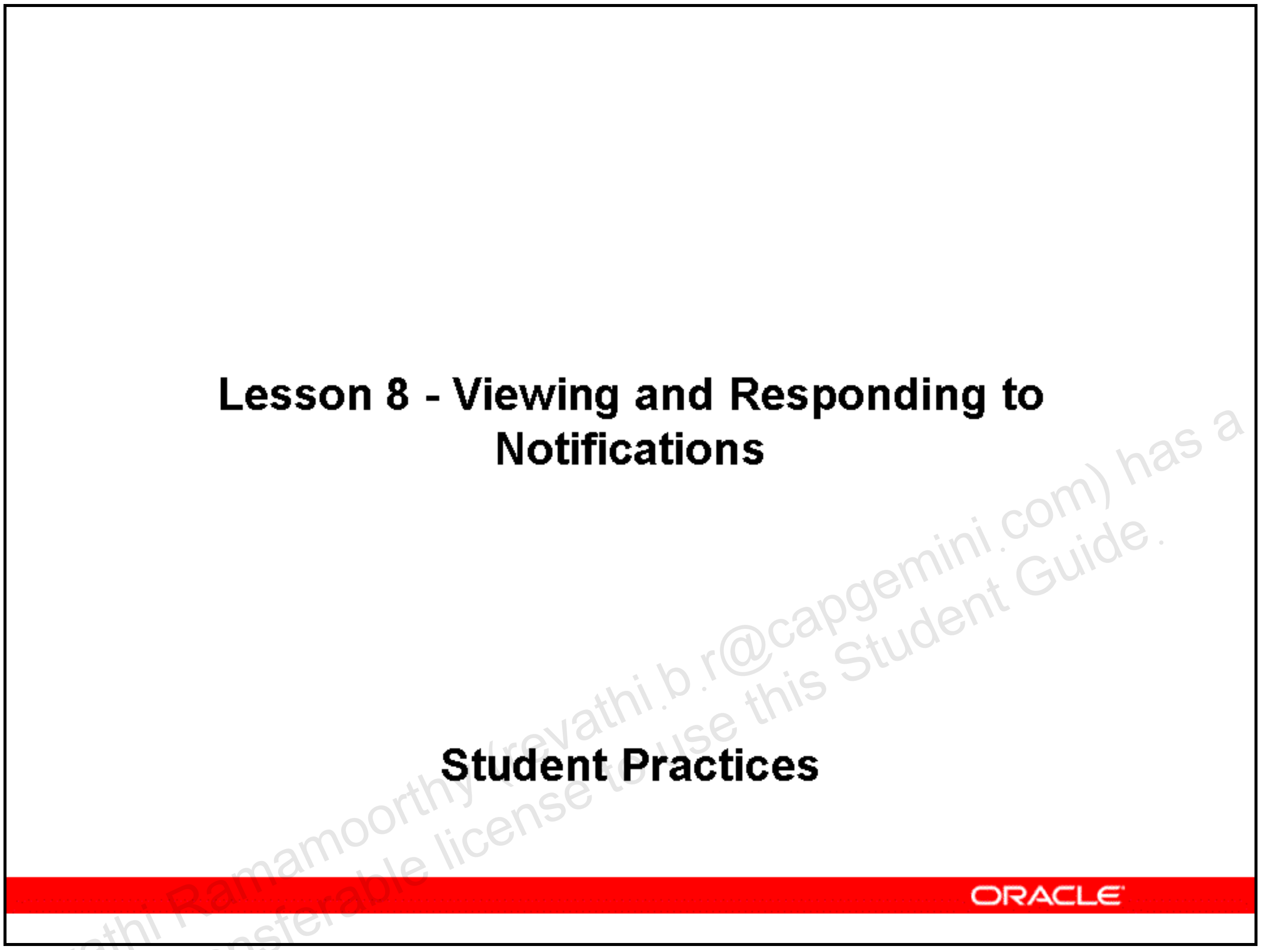
11. Click the Submit button to run the workflow. Oracle Workflow launches the workflow process with the values that you entered. A confirmation message appears. Click OK.

12. To review the process status in the Status Monitor Web pages, select the Status Monitor tab.

    - In the Search region, enter your *XX* Vacation Proposal item type and the item key you chose, and click Go.

    - In the Results region, select your process and click the Activity History button to review the process activities.

    - Then click the Status Diagram button to review the graphical diagram of the status of the process.



Lesson 8 - Viewing and Responding to Notifications

Student Practices

Practice - Responding to Notifications

Overview

In this practice, you will use the Worklist Web pages to respond as the approver to the notification sent by the Vacation Proposal workflow process that you ran in the Running 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.

- The instructor will provide you with the URL for the login page.

- The instructor will provide you with the names of users that you can use as the requestor and approver in the Vacation Proposal process. These user names should have Workflow administrator and user responsibilities assigned to them. Additionally, the user name for the approver should have the System Administrator responsibility assigned to it to be able to view the attached form from the Form Example attribute for the Vacation Proposal message.

Tasks

Responding to Notifications

Use the Worklist Web pages to view and respond to notifications.

Solution – Responding to Notifications

Responding to Notifications

1. Use a Web browser to connect to a Workflow administrator or user responsibility with the URL provided by the instructor. Log in as the approver that you assigned for the vacation proposal in the previous practice.

2. Click the Notifications link.

3. In the Worklist, select the subject line for the Vacation Proposal notification sent by your process to open and review the notification message.

   - Because you defined the special #HIDE_REASSIGN attribute for the Vacation Proposal message, the Reassign button should not appear in the Notification Details page.

     Note: If the user you are logged in as has workflow administrator privileges, as defined on the Workflow Configuration page, then the Reassign button always appears in the Notification Details page regardless of the #HIDE_REASSIGN attribute. This ability to override restrictions on reassignment lets administrators intervene when necessary in exception cases.

   - Because you attached the URL Example attribute to the Vacation Proposal message, Oracle Workflow displays the URL link at the end of the notification.

   - Because you attached the Form Example attribute to the Vacation Proposal message, and you are logged in as a user who has a responsibility that has access to that form, Oracle Workflow displays an attached form icon at the end of the notification. You can click the form icon to drill down to the form. If you have multiple responsibilities with access to the form, a list of your responsibilities appears to let you select the one with which you want to navigate to the form. If you do not have responsibility access to the form, the attached form icon is not displayed.

4. Navigate back to the Notification Details page.

5. Enter comments if you want and select Approve or Reject to respond to the notification.

6. Log off and log in as the requestor. You can use either a Workflow administrator or user responsibility.

7. Click the Notifications link.

8. Open the Vacation Approved or Vacation Rejected notification sent by your process and review the notification message. Click OK to return to the Worklist.

9. To review the updated status of the process in the Status Monitor Web pages, select the Status Monitor tab. Then search for the process with the *XX* Vacation Proposal item type and your item key, and view the status diagram again by clicking the Status Diagram button.

Practice - Modifying A Workflow Process

Overview

In this practice, you will modify the Vacation Proposal workflow process that 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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Creating an HTML Body for a Message

Modify the Vacation Proposal workflow process that you created in the Creating a Workflow Process practice. First, create an HTML body for the Vacation Proposal message.

Modifying a Notification to Accept a Response

Next, modify the process to allow the approver to respond to the Vacation Proposal message with alternate vacation dates.

Adding a Loop to a Process

Finally, modify the process to define a loop that enables the requester to resubmit another proposal to the same approver if the first proposal is rejected.

Solution – Modifying a Workflow Process

Creating an HTML Body for a Message

1. Start the Oracle Workflow Builder.

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

3. In the navigator tree, open your *XX* Vacation Proposal item type and select the Vacation Proposal message.

4. Open the property pages for the Vacation Proposal message and select the Body tab. Within the Body tab, select 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. Click OK.

6. In the Navigator window, click the Verify button to verify your workflow.

Modifying a Notification to Accept a Response

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

8. From the Edit menu, select 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

- Click OK.

10. From the Edit menu, select 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

- Click 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. Select the Body tab:

    - Add the following message attribute tokens for the alternate from and to dates into the message body before the comments token:

      ```
      Alternate From Date: &ALT_FROM_DATE

      Alternate To Date: &ALT_TO_DATE
      ```

    - Change the subject to "Vacation Rejected – Resubmit?".

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 set to Send.

Adding a Loop to a Process

15. On the property pages for the Vacation Rejected message, select the Body tab and add the text “Do you want to resubmit your proposal?” at the end of the message body. Then select 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.

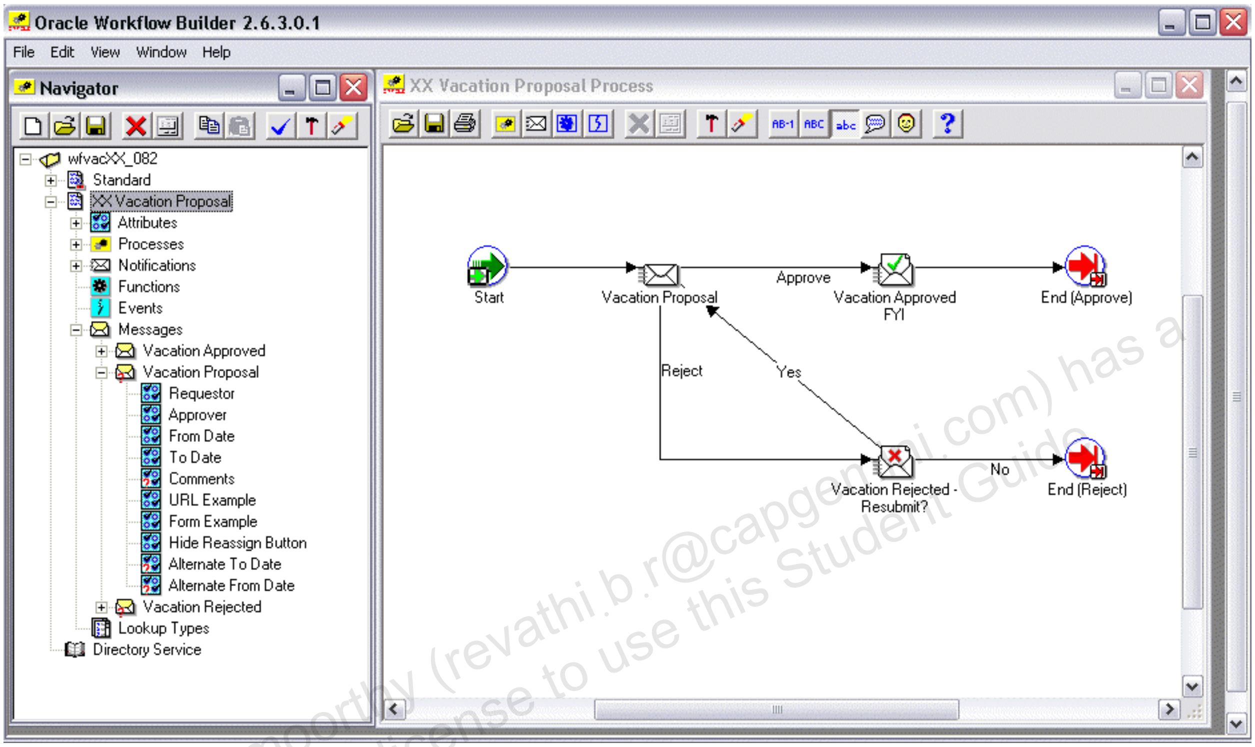
19. Select the transition that connects the Vacation Proposal notification and the Vacation Rejected - Resubmit? notification and drag the transition to create a vertex point. Select the transition again and right-click it. Select the Locked option from the menu that appears. In this way, you can avoid drawing a new transition on top of this one.

20. Delete the existing transition between the Vacation Rejected - Resubmit? node and the End node. Draw a new transition from the Vacation Rejected - Resubmit? node to the End node and select No from the transition results menu.

21. Draw a new transition from the Vacation Rejected - Resubmit? 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, select Save to save your work to your workflow definition file.



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

25. Close the data store.

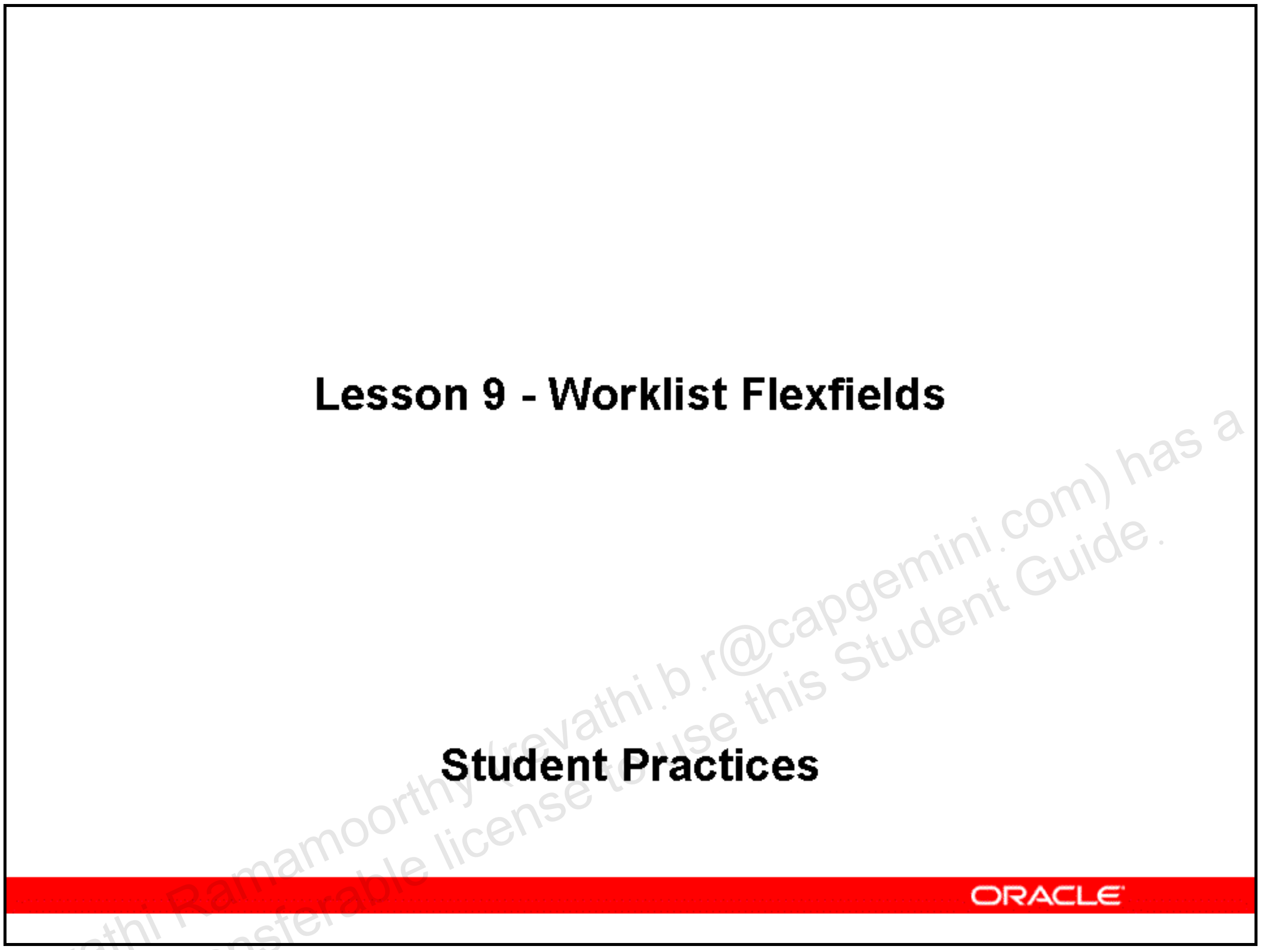
26. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

27. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process and use the Status Monitor Web pages 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.

- As the approver, view and respond to the new proposal.



Lesson 9 - Worklist Flexfields

Student Practices

Practice - Defining a Specialized Worklist View Using Worklist Flexfields

Overview

In this practice, you will define a worklist flexfields rule based on the vacation item type that you created in the Creating a Workflow Process practice. Then you will create a specialized worklist view that includes the columns mapped by the worklist flexfields rule.

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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.

- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.

- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility and a Workflow user responsibility. The user name that you use to log in should have these responsibilities assigned to it.

- The instructor will perform the necessary setup steps to provide the Personal Worklist as a menu option on a Workflow user responsibility and to enable you to define personalized worklist views. These setup steps include adding the Personal Worklist function to the application menu for the responsibility, assigning the responsibility to a user, and setting the Personalize Self-Service Defn profile option to Yes for that user.

Tasks

Defining a Worklist Flexfields Rule

Define a worklist flexfields rule based on the vacation item type that you created in the Creating a Workflow Process practice.

Simulating the Effects of a Worklist Flexfields Rule

Perform a worklist flexfields rule simulation to review and verify which attributes are available in which worklist flexfields columns for your item type.

Creating a Personalized View

Create a personalized worklist view that includes the columns mapped by your worklist flexfields rule and displays notifications from your vacation item. You can also specify your desired column settings and sort order for the personalized view.

Viewing Notifications Using a Personalized Worklist View

Use the personalized view that you created to view the vacation message attributes mapped by your rule in the Personal Worklist with your desired settings and sort order.

Solution – Defining a Specialized Worklist View Using Worklist Flexfields

Defining a Worklist Flexfields Rule

1. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

2. Click the Worklist Flexfields Rules link.

3. On the Worklist Flexfields Rules page, click the Create Rule button.

4. Enter the following values in the Create Worklist Flexfields Rule: Enter General Properties page:
   - Rule Name: *XX*WFVACRULE
   - Display Name: *XX* Vacation Worklist Flexfields Rule
   - Status: Enabled
   - Phase: 110
   - Owner Name: *XX* Practices
   - Owner Tag: FND

   - Click Next.

5. In the Create Worklist Flexfields Rule: Set Filter Criteria page, search for the *XX* Vacation Proposal item type that you created in the Creating a Workflow Process practice.

6. Select the *XX* Vacation Proposal item type from the Available Filter Criteria list and move it to the Selected Filter Criteria list. Click Next.

7. In the Create Worklist Flexfields Rule: Select Message Attributes page, select the following attributes from the Available list and move them to the Selected list:
   - Alternate From Date
   - Alternate To Date
   - From Date
   - To Date
   - Comments

   - Click Next.

8. In the Create Worklist Flexfields Rule: Map Attributes to Columns page, ensure the message attributes are mapped to the following columns:
   - Alternate From Date: DATE_ATTRIBUTE1
   - Alternate To Date: DATE_ATTRIBUTE2
   - From Date: DATE_ATTRIBUTE3
   - To Date: DATE_ATTRIBUTE4
   - Comments: TEXT_ATTRIBUTE1

- Click Finish.

9. Click the Home global link to return to the Oracle E-Business Suite home page.

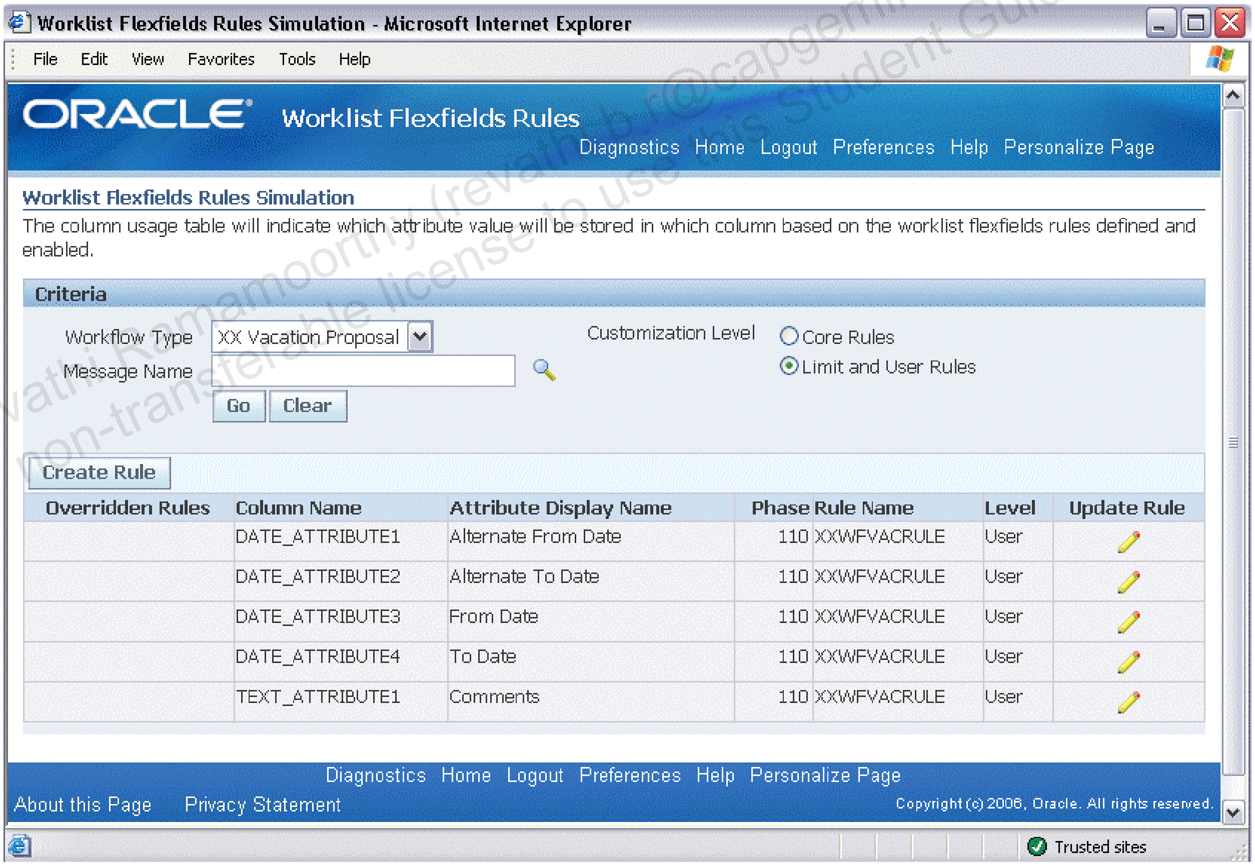
Simulating the Effects of a Worklist Flexfields Rule

10. Click the Worklist Flexfields Rules Simulation link in a Workflow administrator responsibility.

11. Enter the following values in the Criteria region of the Worklist Flexfields Rules Simulation page:
    – Workflow Type: *XX* Vacation Proposal
    – Customization Level: Limit and User Rules checked

- Click Go.

12. Review and verify which attributes are available in which worklist flexfields columns for your item type based on your worklist flexfields rule.



Worklist Flexfields Rules Simulation - Microsoft Internet Explorer

Worklist Flexfields Rules Simulation

The column usage table will indicate which attribute value will be stored in which column based on the worklist flexfields rules defined and enabled.

Criteria

Workflow Type: XX Vacation Proposal
Message Name:
Customization Level: ○ Core Rules ● Limit and User Rules

Overridden Rules	Column Name	Attribute Display Name	Phase	Rule Name	Level	Update Rule
	DATE_ATTRIBUTE1	Alternate From Date	110	XXWFVACRULE	User	✎
	DATE_ATTRIBUTE2	Alternate To Date	110	XXWFVACRULE	User	✎
	DATE_ATTRIBUTE3	From Date	110	XXWFVACRULE	User	✎
	DATE_ATTRIBUTE4	To Date	110	XXWFVACRULE	User	✎
	TEXT_ATTRIBUTE1	Comments	110	XXWFVACRULE	User	✎

Creating a Personalized View

13. Return to the Oracle E-Business Suite home page and select the Workflow user responsibility provided by the instructor.

14. Click the Personal Worklist link.

15. Click the Personalize Page global link.

16. In the Personalization Structure region, click the Seeded User Views icon for the "Table: Customizable and …" item.

17. On the Personalize Views page, click the Create View button.

18. On the Create View page, enter the following values to define your view:
    - View Name: *XX*WFVACVIEW
    - Number of Rows Displayed: 25 Rows
    - Securing Function: (Leave this field blank)
    - Description: *XX* Worklist Personal View

19. Click the Rename Columns/Totaling button and enter the following column names:
    - DATE_ATTRIBUTE1: Alternate From Date
    - DATE_ATTRIBUTE2: Alternate To Date
    - DATE_ATTRIBUTE3: From_Date
    - DATE_ATTRIBUTE4: To_Date
    - TEXT_ATTRIBUTE1: Comments

- Click Apply.

20. Define the following properties in the Column Properties region for your view:

- Move the following columns from the Available Columns list to the Columns Displayed list:
    - Alternate From Date
    - Alternate To Date
    - From Date
    - To Date
    - Comments

- Move the following columns from the Columns Displayed list to the Available Columns list:
    - Due
    - From

- Define the following sort settings for your view:
    - First Sort: Sent (descending order)
    - Second Sort: From Date (ascending order)
    - Third Sort: Subject (ascending order)

21. Define the following values in the Search Query to Filter Data in your Table region:

- Select the "Show table data when any condition is met" option.

- Select Type in the Add Another pull-down menu and click the Add button. For the Type filter, select the “contains” filter operator and then enter “*XX* Vacation Proposal” as the workflow type filter value.

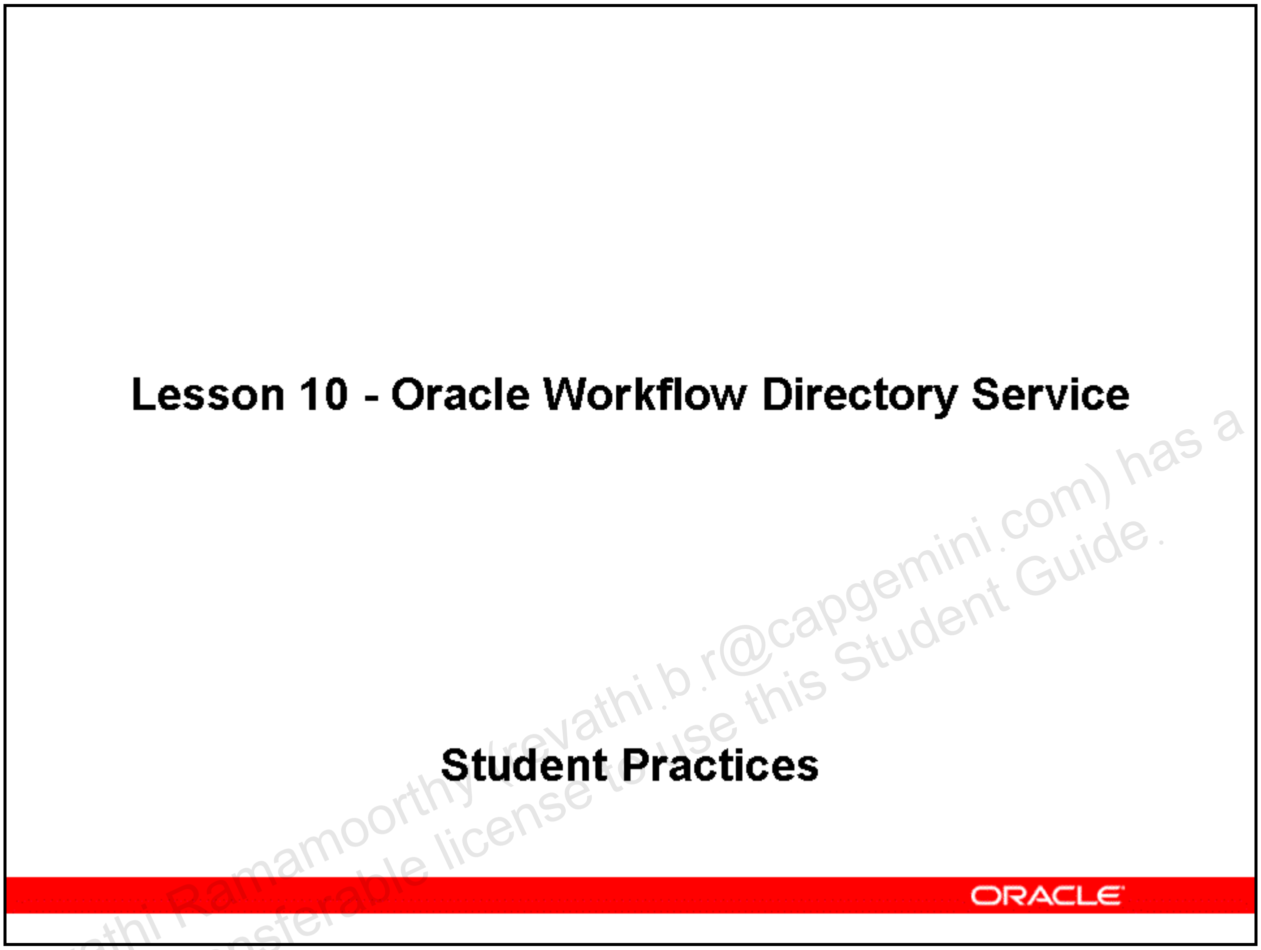
22. Click Apply.

Viewing Notifications Using a Personal Worklist View

23. Return to the Oracle E-Business Suite home page, and click the Personal Worklist link.

24. In the Personal Worklist, select the *XX*WFVACVIEW view from the View pull-down menu and click Go. Verify that the notifications sent from your *XX* Vacation Proposal item type are displayed in the worklist with the attribute columns that you specified and your desired settings and sort order.

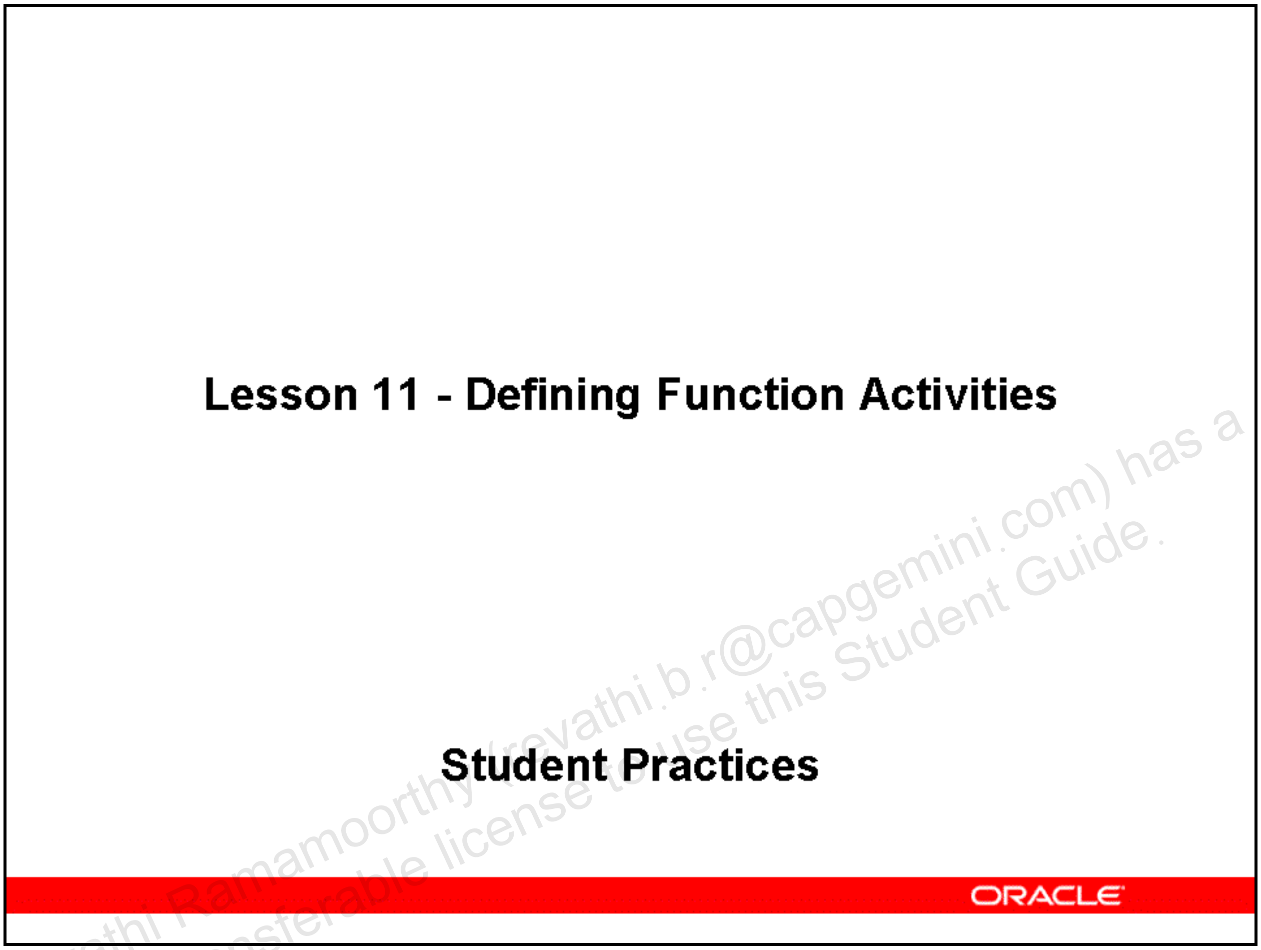
Lesson 10 - Oracle Workflow Directory Service



Lesson 10 - Oracle Workflow Directory Service

Student Practices

There are no practices or demonstrations in this lesson.



Lesson 11 - Defining Function Activities

Student Practices

Practice - Defining a Function Activity

Overview

In this practice, you will define a function activity to update a schedule of planned vacations. For this practice, you will use a predefined PL/SQL procedure for your function activity.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining a Function Activity

In the Vacation Proposal item type that you created in the Creating a Workflow Process practice, add a function activity that updates a schedule of planned vacations.

Loading a PL/SQL Package

Load the PL/SQL package containing the predefined PL/SQL procedure that the function activity calls.

Solution – Defining a Function Activity

Defining a Function Activity

1. Start the Oracle Workflow Builder.

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

3. In the navigator tree, open your *XX* Vacation Proposal item type and open the process diagram window for the Vacation Proposal process.

4. Delete the transition between the Vacation Approved FYI node and the End (Approve) node.

5. 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.

6. 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 WFVAC*XX*.SCHEDULE_UPDATE should have the internal name SCHEDULE_UPDATE.

7. Select the icon DB_UPD.ICO for the activity.

8. Enter the package and procedure name WFVAC*XX*.SCHEDULE_UPDATE in the Function Name field for the activity. This function will have no result. Click OK.

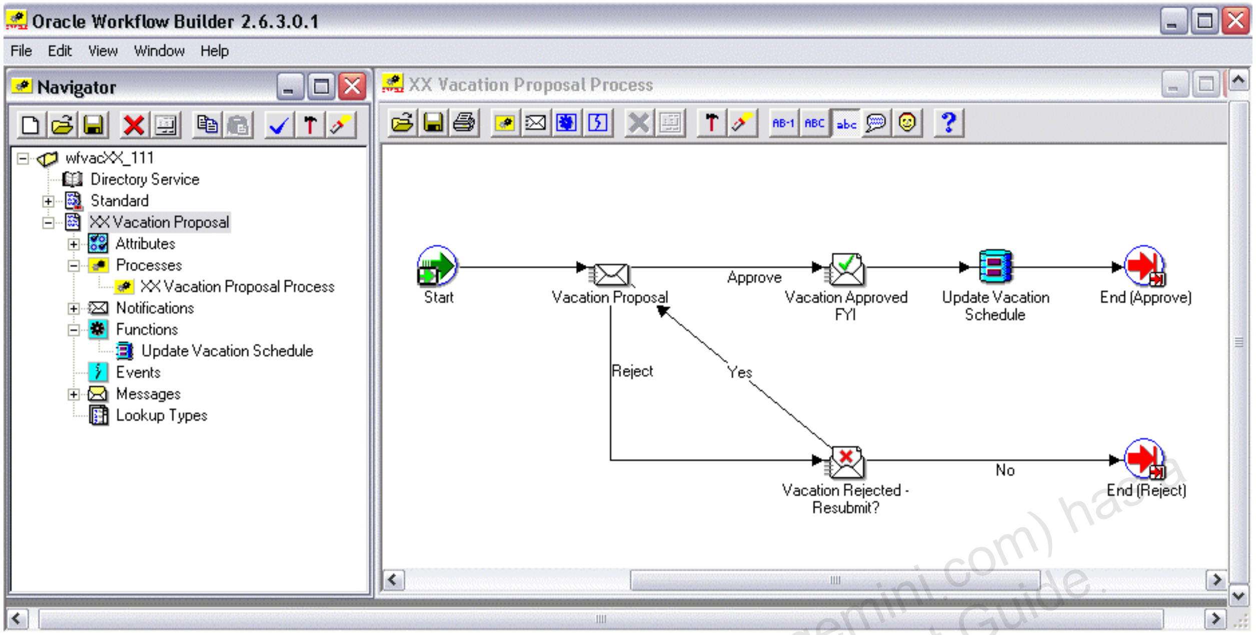
   Note: If you are defining a function activity during the process design phase and you do not yet know the package and procedure name for the PL/SQL procedure you want the activity to call, you can enter WF_STANDARD.NOOP as a placeholder in the Function Name field. WF_STANDARD.NOOP is a standard Workflow PL/SQL procedure that sets the resultout parameter to wf_engine.eng_completed and returns.

9. Draw transitions from the Vacation Approved FYI node to the Update Vacation Schedule node, and from the Update Vacation Schedule node to the End (Approve) node.

10. To verify error handling information for the Vacation Proposal process, open the property pages for the Vacation Proposal process and select the Details tab. Ensure that the error item type is WFERROR and the error process is DEFAULT_ERROR. Click OK.

11. In the Navigator window, click the Verify button to verify your workflow.

12. From the File menu, select Save to save your work to your workflow definition file.



13. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

14. Close the data store.

Loading a PL/SQL Package

15. 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

16. 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.

17. Log in to SQL*Plus using the database user name, password, and connect string provided by the instructor. Run the table creation, package specification, and package body scripts in that order by entering the following commands from the directory where the scripts are located:

    ```
    @wfvacxxc

    @wfvacxxs

    @wfvacxxb
    ```

    Alternatively, you can run the scripts from the default prompt if you include the directory path for the scripts in the commands. For example, if your scripts are located in the E:\Labs folder, then enter the following commands:

    ```
    @E:\Labs\wfvacxxc

    @E:\Labs\wfvacxxs

    @E:\Labs\wfvacxxb
    ```

    Note: You can ignore the “table or view does not exist” error in the output from the first script, `wfvacxxc`. The script follows programming standards that require attempting to drop the table before attempting to create it, to remove any preexisting table of that name. After this error message, you should see messages indicating that the table and index have been successfully created.

18. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

19. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process and approve the vacation proposal, and use the Status Monitor Web pages 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 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.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining a Function Activity with a Result Type

Add a function activity to check whether the approver is the same as the requestor in the Vacation Proposal item type that you created in the Creating a Workflow Process practice. The function activity should have a result type of Yes/No.

Branching on a Function Activity Result

Model different branches in the process based on the function activity result.

Solution – Branching on a Function Activity Result

Defining a Function Activity with a Result Type

1. Start the Oracle Workflow Builder.

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

3. Open the process diagram window for the Vacation Proposal process.

4. Delete the transition between the Start node and the Vacation Proposal node.

5. 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.

6. 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 WFVAC*XX*.CHECK_APPROVER should have the internal name CHECK_APPROVER.

7. Select the icon COMPARE.ICO for the activity.

8. Enter the package and procedure name WFVAC*XX*.CHECK_APPROVER as the function name for the activity. The WFVAC*XX*.CHECK_APPROVER procedure checks whether the approver is the same as the requestor.

   Note: The WFVAC*XX*.CHECK_APPROVER procedure is provided in the sample WFVAC*XX* package that you loaded in the Defining a Function Activity practice. If you have not already copied, edited, and run the sample package specification and body scripts to load this package, follow the instructions in the Defining a Function Activity practice to do so now.

9. Select the Yes/No result type for the activity.

10. Click OK.

Branching on a Function Activity Result

11. Draw a transition from the Start node to the Approver Same as Requestor? node.

12. Draw a transition from the Approver Same as Requestor? node to the Vacation Proposal node and select No from the transition results menu.

13. Create a new End node by dragging the End function from the Standard item type in the navigator tree into the process diagram. Open the property pages for the node, and select the Node tab. In the Start/End field, select End. In the Result field, select Reject.

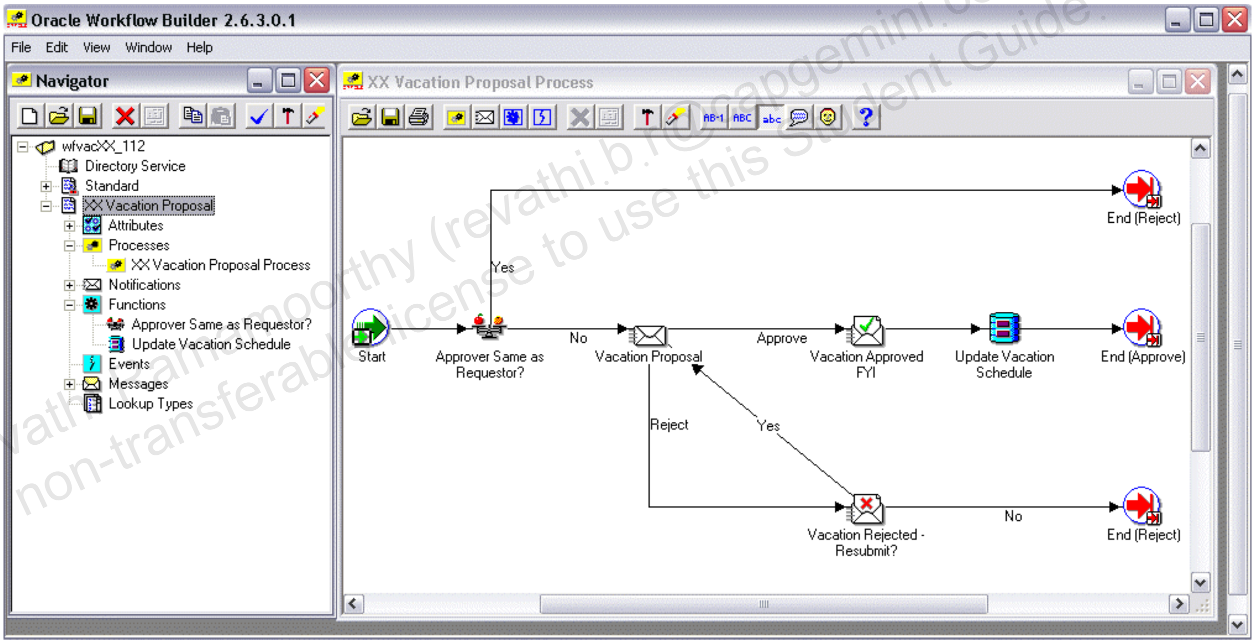
    Using multiple End nodes can help simplify your diagram and uniquely identify which End activity is executed when different paths through a process can terminate with the same result. For example, the Vacation Proposal can now end with a result of Reject for two unique reasons:

    – A vacation proposal is submitted with the approval equal to the requestor.
    – A vacation proposal is rejected by the approver.

14. 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.

15. In the Navigator window, click the Verify button to verify your workflow.

16. From the File menu, select Save to save your work to your workflow definition file.



17. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

18. Close the data store.

19. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

20. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process, and use the Status Monitor Web pages 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, and no notification should be sent.

- 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.

Practice - Using the Standard Assign Activity

Overview

In this practice, you will create a subprocess that uses the standard Assign activity to automatically reset item attribute values when the requestor resubmits the vacation proposal.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Creating a Subprocess to Reset Item Attribute Values

Create a subprocess within the Vacation Proposal workflow process that you created in the Creating a Workflow Process practice. The subprocess should use the standard Start, Assign, And, and End activities from the Standard item type.

Defining Assign Activity Nodes

Complete the definitions of the Assign activity nodes to prepare the necessary item type attributes to submit a fresh vacation proposal. Use the Assign activity nodes to automatically reset the From Date and To Date attributes to the alternate values provided by the approver when the initial vacation proposal was rejected, and to clear the Alternate From Date, Alternate To Date, and Comments attributes.

Solution – Using the Standard Assign Activity

Creating a Subprocess to Reset Item Attribute Values

1. Start the Oracle Workflow Builder.

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

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

4. Open the property pages for the From Date message attribute associated with the Vacation Rejected message and set the Source field to Send. The requestor will now no longer respond to this message with a new from date; instead, the From Date attribute will be automatically set to the alternate from date suggested by the approver.

5. Open the property pages for the To Date message attribute associated with the Vacation Rejected message and set the Source field to Send. The requestor will now no longer respond to this message with a new to date; instead, the To Date attribute will be automatically set to the alternate to date suggested by the approver.

6. From the Edit menu, select New > Process.

7. Define the following properties for the process activity:
   - Internal Name: SET_DATES
   - Display Name: Reset Dates and Comments
   - Icon: PROCESS.ICO
   - Result Type: <None>
   - Runnable: Deselected

   - Click OK.

8. Open the process diagram window for the XX Vacation Proposal Process.

9. Delete the transition between the Vacation Rejected - Resubmit? node and the Vacation Proposal node.

10. Drag the Reset Dates and Comments process activity into the process diagram between the Vacation Rejected - Resubmit? node and the Vacation Proposal node.

11. Draw a transition from the Vacation Rejected - Resubmit? node to the Reset Dates and Comments node and select Yes from the transition results menu.

12. Draw a transition from the Reset Dates and Comments node to the Vacation Proposal node.

13. Double-click the Reset Dates and Comments node to display the process diagram window.

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

15. Create five Assign nodes by dragging the Assign function from the Standard item type in the navigator tree into the process diagram window five times.

16. Create an And node by dragging the And function from the Standard item type in the navigator tree into the process diagram window.

17. Create an 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 select the Node tab. In the Start/End field, select End.

18. Draw a transition from the Start node to the first Assign node.

19. Draw a transition from the first Assign node to the second Assign node.

20. Draw transitions from the second Assign node to the third, fourth, and fifth Assign nodes. This design specifies that the last three Assign nodes can be processed in parallel.

21. Draw transitions from the third, fourth, and fifth Assign nodes to the And node. This design specifies that the process should not continue until all three of these Assign nodes have been completed.

22. Draw a transition from the And node to the End node.

Defining Assign Activity Nodes

23. Double-click the first Assign node and select the Node tab. Enter “Assign value from Alternate From Date attribute to From Date” as the comment for the node and click Apply.

24. Select the Node Attributes tab. In the Attribute region, define the following properties for the Item Attribute activity attribute:
    - Name: Item Attribute
    - Type: Constant
    - Value: From Date

    - Click Apply.

25. Next, in the Attribute region, define the following properties for the Date Value activity attribute:
    - Name: Date Value
    - Type: Item Attribute
    - Value: Alternate From Date

    - Click OK.

26. Double-click the second Assign node and select the Node tab. Enter "Assign value from Alternate To Date attribute to To Date" as the comment for the node and click Apply.

27. Select the Node Attributes tab. In the Attribute region, define the following properties for the Item Attribute activity attribute:
    - Name: Item Attribute
    - Type: Constant
    - Value: To Date

    - Click Apply.

28. Next, in the Attribute region, define the following properties for the Date Value activity attribute:
    - Name: Date Value
    - Type: Item Attribute
    - Value: Alternate To Date

    - Click OK.

29. Double-click the third Assign node and select the Node tab. Enter "Clear value provided by approver from Alternate From Date" as the comment for the node and click Apply.

30. Select the Node Attributes tab. In the Attribute region, define the following properties for the Item Attribute activity attribute:
    - Name: Item Attribute
    - Type: Constant
    - Value: Alternate From Date

    - Click Apply.

31. Next, in the Attribute region, define the following properties for the Date Value activity attribute:
    - Name: Date Value
    - Type: Constant
    - Value: (Leave this field blank)

    - Click OK.

32. Double-click the fourth Assign node and select the Node tab. Enter "Clear value provided by approver from Alternate To Date" as the comment for the node and click Apply.

33. Select the Node Attributes tab. In the Attribute region, define the following properties for the Item Attribute activity attribute:
    - Name: Item Attribute
    - Type: Constant
    - Value: Alternate To Date

    - Click Apply.

34. Next, in the Attribute region, define the following properties for the Date Value activity attribute:
    - Name: Date Value
    - Type: Constant
    - Value: (Leave this field blank)

- Click OK.

35. Double-click the fifth Assign node and select the Node tab. Enter “Clear value provided by approver from Comments” as the comment for the node and click Apply.

36. Select the Node Attributes tab. In the Attribute region, define the following properties for the Item Attribute activity attribute:
    - Name: Item Attribute
    - Type: Constant
    - Value: Comments

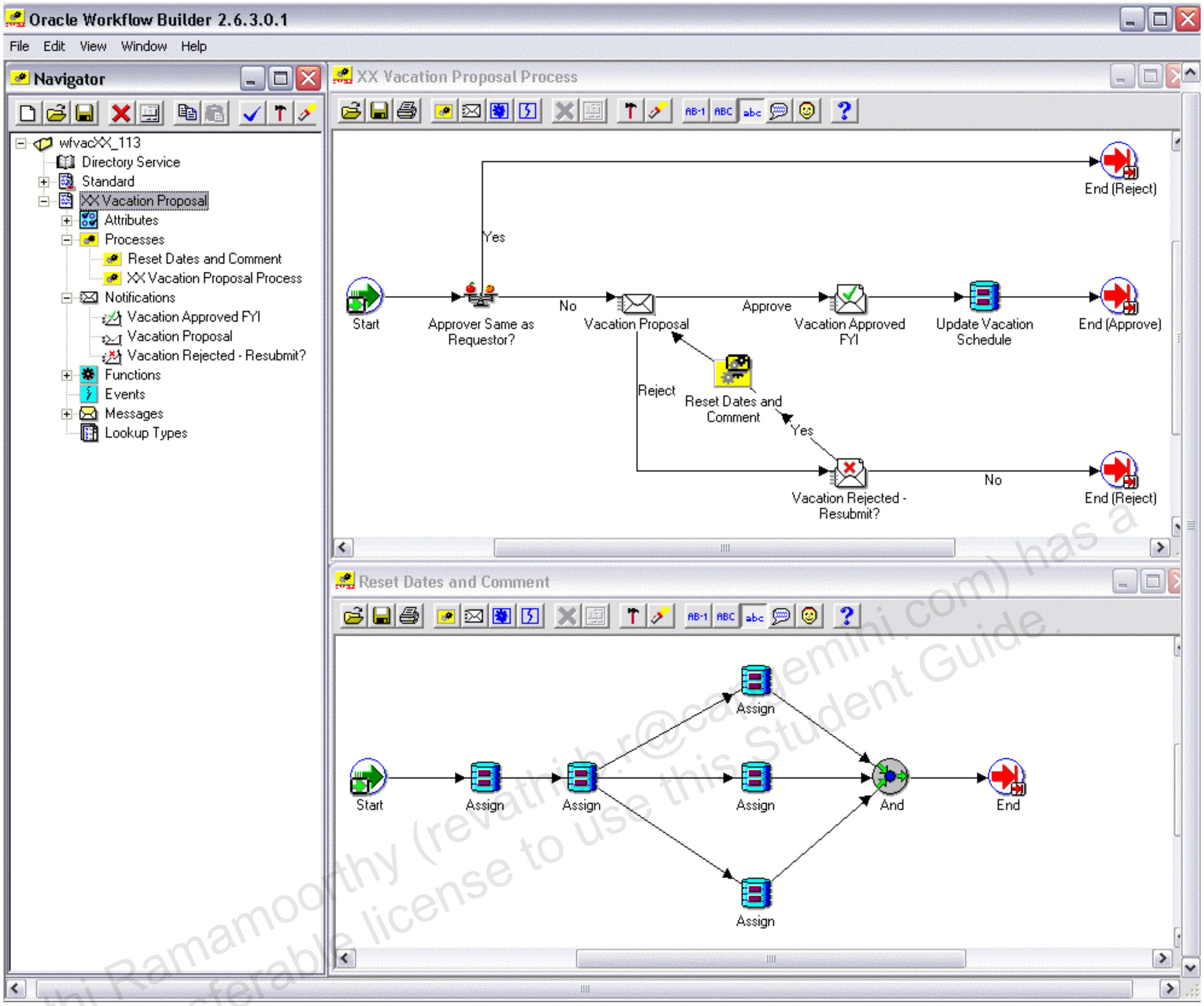
- Click Apply.

37. Next, in the Attribute region, define the following properties for the Date Value activity attribute:
    - Name: Text Value
    - Type: Constant
    - Value: (Leave this field blank)

- Click OK.

38. In the Navigator window, click the Verify button to verify your workflow.

39. From the File menu, select Save to save your work to your workflow definition file.



40. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

41. Close the data store.

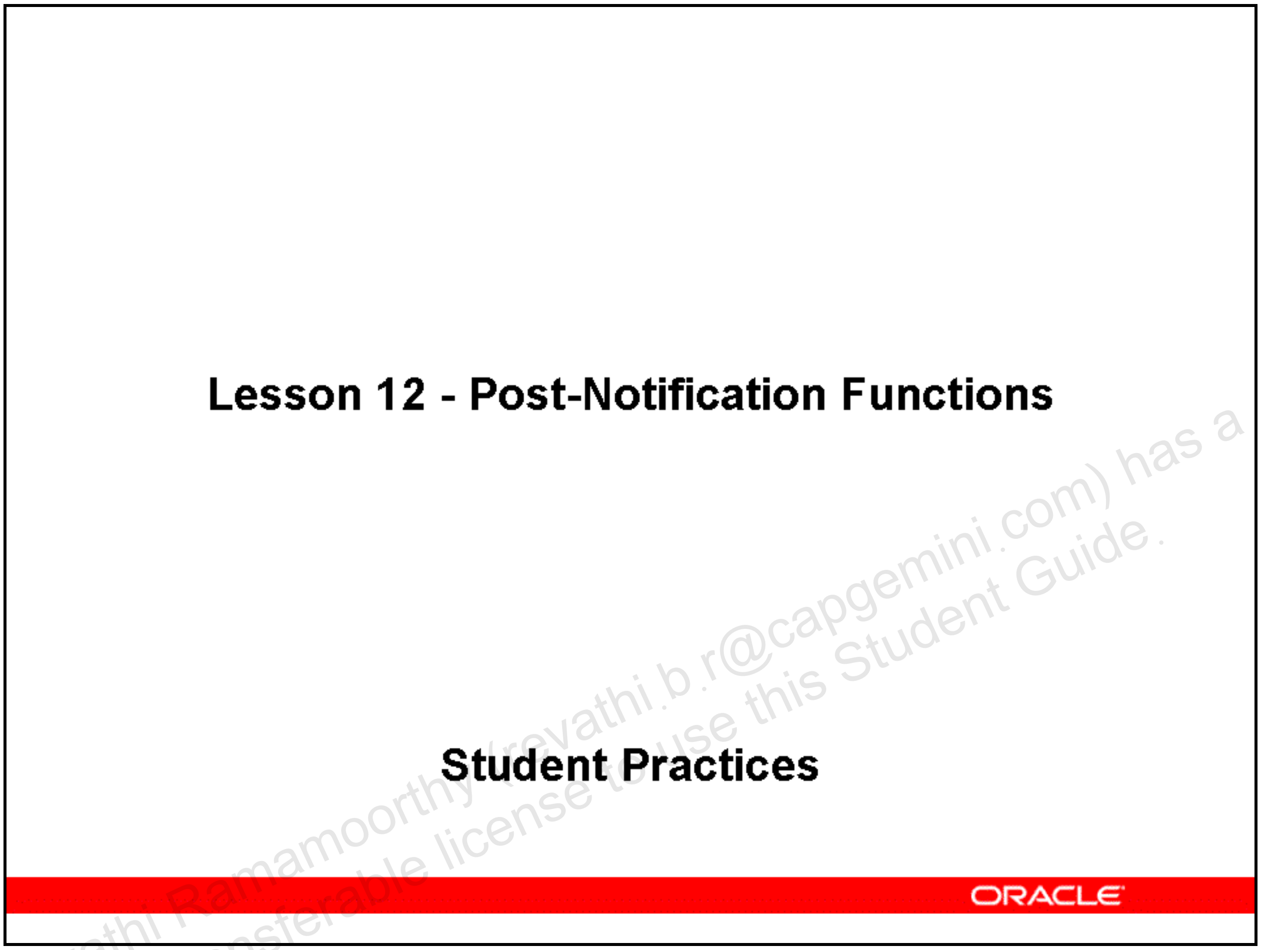
42. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

43. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process and use the Status Monitor Web pages to review the status of the process.

- As the approver, reject the initial vacation proposal and provide alternate from and to dates and comments.

- Then, as the requestor, respond to the Vacation Rejected – Resubmit? notification, specifying that you want to resubmit the vacation proposal. You should not need to enter any other response values.

- As the approver, review the new vacation proposal. The From Date and To Date should now be set to the alternate dates that you provided earlier. The Alternate From Date, Alternate To Date, and Comments should now be null again.



Lesson 12 - Post-Notification Functions

Student Practices

Practice - Defining a Post-Notification Function

Overview

In this practice, you will define a post-notification function to validate action taken on the Vacation Proposal notification in the Vacation Proposal item type that you created in the Creating a Workflow Process practice. For this practice, you will use a predefined PL/SQL procedure as the post-notification function.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.

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

- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.

- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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, approver, and new recipient in the Vacation Proposal process. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining a Post-Notification Function for a Notification

Define a post-notification function to validate action taken on the Vacation Proposal notification activity in the Oracle Workflow Builder.

Solution – Defining a Post-Notification Function

Defining a Post-Notification Function for a Notification

1. Start the Oracle Workflow Builder.

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

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

4. Open the property pages for the Vacation Proposal notification.

5. In the Function Name field, enter the package and procedure name WFVAC*XX*.NTF_VACATION_PROPOSAL for the activity. Ensure that the function type is set to PL/SQL. Click OK.

   The WFVAC*XX*.NTF_VACATION_PROPOSAL procedure validates the action taken on the notification to enforce the following requirements:
   - If the approver rejects the vacation proposal, alternate vacation dates must be provided, and the alternate from date must be prior to the alternate to date.
   - The approver is allowed to delegate (forward) the notification to another role.
   - The approver is not allowed to transfer the notification to another role.

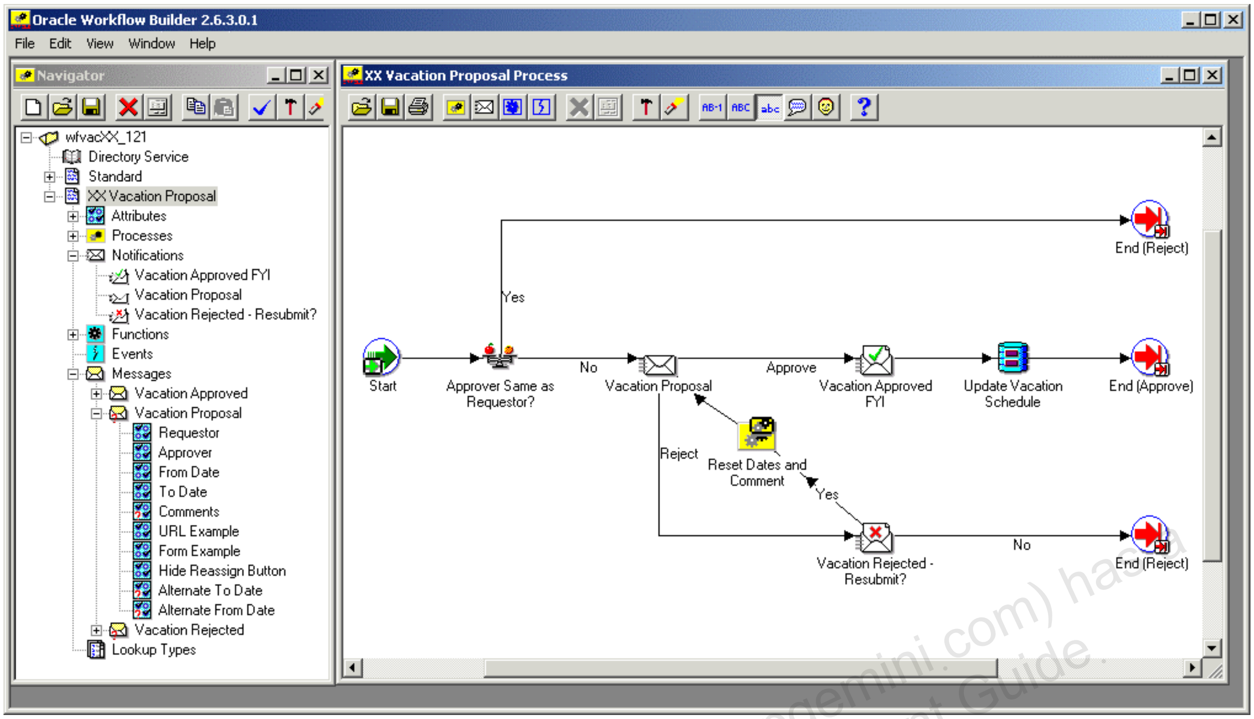
   Note: The WFVAC*XX*.NTF_VACATION_PROPOSAL procedure is provided in the sample WFVAC*XX* package that you loaded in the Defining a Function Activity practice. If you have not already copied, edited, and run the sample package specification and body scripts to load this package, follow the instructions in the Defining a Function Activity practice to do so now.

6. Open the property pages for the Hide Reassign Button message attribute for the Vacation Proposal message.

7. You will need to reassign Vacation Proposal notifications to test your post-notification function, so you will need the Reassign button to be displayed, not hidden. Change the default value for the Hide Reassign Button message attribute to N and click OK.

8. In the Navigator window, click the Verify button to verify your workflow.

9. From the File menu, select Save to save your work to your workflow definition file.



10. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

11. Close the data store.

12. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

13. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process and use the Status Monitor Web pages to review the status of the process.

- Run the process and approve the vacation proposal. The response should be successful.

- Run the process and reject the vacation proposal notification, but do not enter any alternate dates. You should receive an error message stating that you must provide alternate dates.

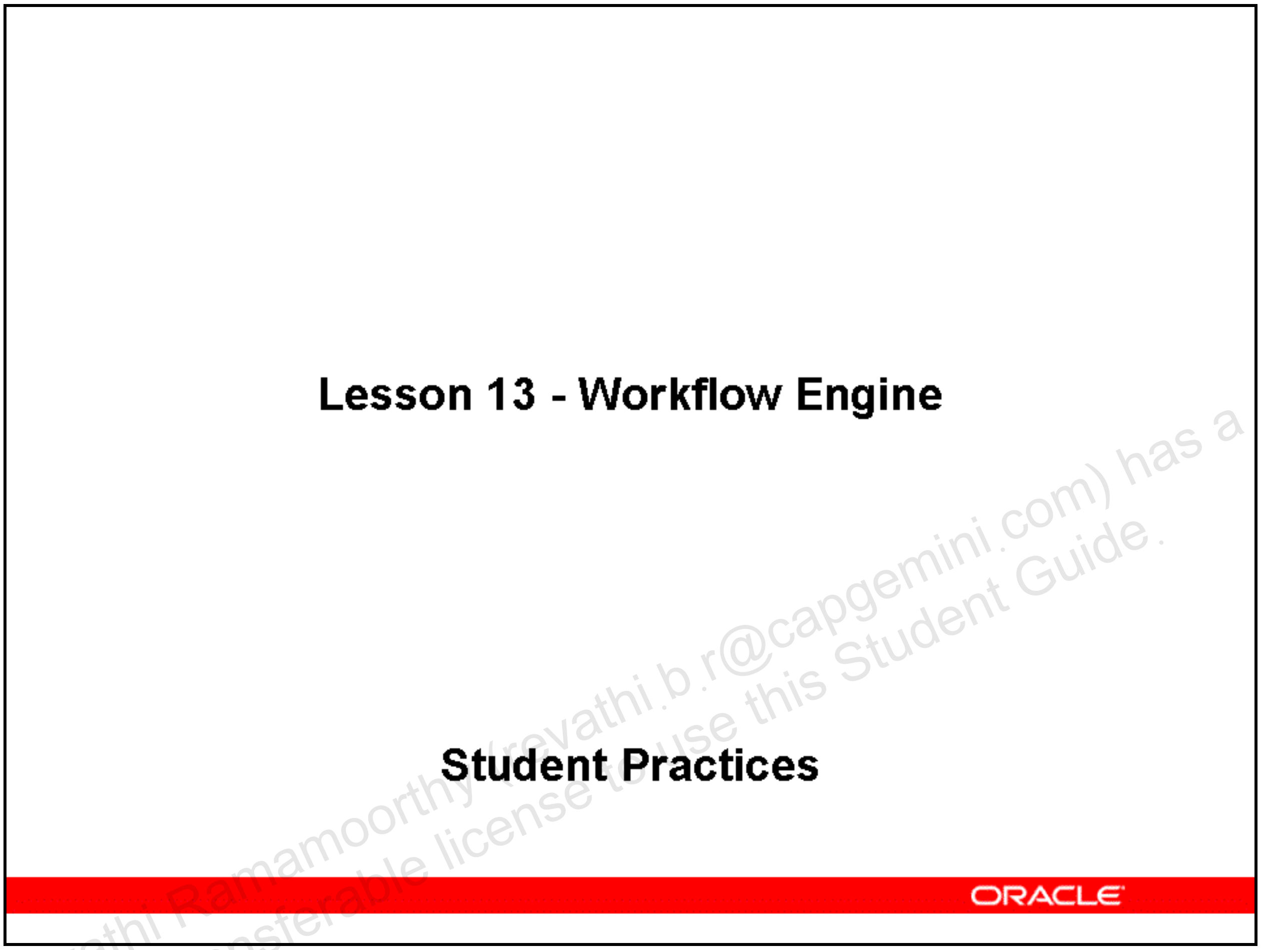
- Reject the vacation proposal notification with an alternate from date later than the alternate to date. You should receive an error message stating that the alternate from date must be prior to the alternate to date.

- Reject the vacation proposal with valid alternate dates. The response should be successful.

- Run the process and click the Reassign button in the Notification Details page to reassign the notification. Select a user name and select the Transfer option to attempt to transfer the notification to that user. When you submit the reassignment, you should receive an error message stating that transfers are not allowed.

- Choose the Delegate option to delegate the notification to the selected user. The reassignment should be successful.

Lesson 13 - Workflow Engine



Lesson 13 - Workflow Engine

Student Practices

Practice - Implementing Timeout Processing

Overview

In this practice, you will implement one type of timeout processing in the Vacation Proposal workflow process that 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining Timeout Processing with a Loop Counter

Modify the Vacation Proposal workflow process that you created in the Creating a Workflow Process practice by defining timeout processing to transition to a loop counter, and on the third loop, exit to approve the proposal automatically.

Running a Background Engine

Run a background engine to process timed out activities.

Solution – Implementing Timeout Processing

Defining Timeout Processing with a Loop Counter

1. Start the Oracle Workflow Builder.

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

3. Open the process diagram window for the Vacation Proposal process and open the property pages for the Vacation Proposal notification activity node.

4. Select the Node tab. In the Timeout region, select Relative Time as the type. For the value, enter a short duration such as 1 minute for testing purposes. Click OK.

5. Select the Loop Counter function activity in the Standard item type. To locate the Loop Counter activity, you can choose Find from the Edit menu or press Ctrl+F. Enter "Loop Counter" in the Search Text field, select the Display Name and Function check boxes, and click the Search button. The Loop Counter activity will be automatically selected. Close the Find window.

6. Drag and drop the Loop Counter activity into the Vacation Proposal process diagram, positioning it above the Vacation Proposal notification node.

7. Open the property pages for the Loop Counter node and select the Node Attributes tab to set the number of times the Workflow Engine should execute the loop. Select the Loop Limit attribute. Select the type Constant and enter the value 2 for the attribute. Click OK.

8. Draw a transition from the Vacation Proposal node to the Loop Counter node and select <Timeout> from the transition results menu.

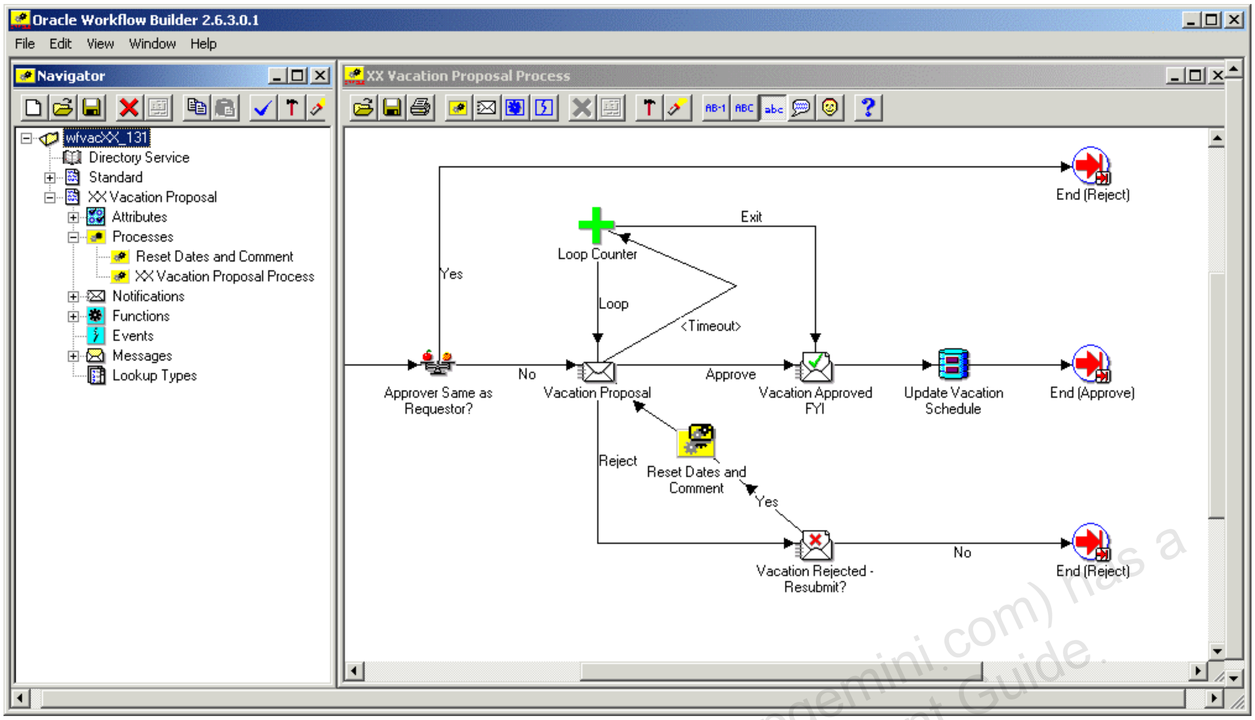
9. Create a vertex point in the transition between the Vacation Proposal node and the Loop Counter node.

10. Draw a transition from the Loop Counter node back to the Vacation Proposal node and select Loop from the transition results menu.

11. Draw a transition from the Loop Counter node to the Vacation Approved FYI node and select Exit from the transition results menu.

12. In the Navigator window, click the Verify button to verify your workflow.

13. From the File menu, select Save to save your work to your workflow definition file.



14. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

15. Close the data store.

16. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

17. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process, and use the Status Monitor Web pages to review the status of the process.

   - Verify your work by reviewing the Vacation Proposal notification on the Notification Details Web page. The message header should now display both a sent date and a due date. The interval between these times should be equal to the relative time value you entered for the notification timeout. Write down the times for the sent date and due date.

   - To test your work, do not respond to the Vacation Proposal notification, but allow the notification activity to time out instead by waiting for the timeout interval you specified to elapse.

Running a Background Engine

18. Run a background engine for your item type to process the timed out activity and allow the process to continue. Run the background engine repeatedly to process timed out activities for each execution of the loop. You can run a background engine through the Oracle Workflow Manager component of Oracle Applications Manager. You can also run a background engine using the WF_ENGINE.Background API.

19. To run a background engine through Oracle Applications Manager to process timed out activities:

    - Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

    - Click the Workflow Manager link.

    - On the Workflow System page, select Background Engines from the Submit Request For menu and click Go to submit the concurrent request through Oracle Self-Service Web Applications.

    - Click Next and enter the following parameters:
        - Item Type: *XX* Vacation Proposal
        - Process Deferred: No
        - Process Timeout: Yes
        - Process Stuck: No

    - Click Next in each remaining page of the wizard until the Summary page, and then click Submit.

20. To run the WF_ENGINE.Background API to process timed out activities, open SQL*Plus and enter the following command:

```
Exec WF_ENGINE.Background('WFVACXX', NULL, NULL, FALSE, TRUE,
FALSE);
```

21. Log in to Oracle Workflow as the requestor and use the Status Monitor Web pages to review the status of the process. Check the activity history to verify that the process followed the timeout transition to execute the Loop Counter activity.

22. Allow the Vacation Proposal notification activity to time out again. Then run a background engine again to process the timed out activity.

23. Use the Status Monitor Web pages to review the status of the process and verify that the process followed the timeout transition to execute the Loop Counter activity a second time.

24. Allow the Vacation Proposal notification activity to time out again. Then run a background engine again to process the timed out activity.

25. Use the Status Monitor Web pages to review the status of the process and verify that the process followed the timeout transition to execute the Loop Counter activity a third time and then followed the exit transition to execute the Vacation Approved FYI activity and the rest of the process. Use the Worklist Web pages to review the notification sent by the Vacation Approved FYI activity.

Practice - Implementing Deferred Processing

Overview

In this practice, you will implement deferred processing for a function activity in the Vacation Proposal workflow process that 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining Deferred Processing

Modify the Vacation Proposal workflow process that you created in the Creating a Workflow Process practice by implementing deferred processing for a function activity in the process. To do so, set the function activity cost above the engine threshold to defer the activity when the process is executed.

Running a Background Engine

Run a background engine to process deferred activities.

Solution – Implementing Deferred Processing

Defining Deferred Processing

1. Start the Oracle Workflow Builder.

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

3. Open the property pages for the Update Vacation Schedule function activity. Change the cost for the activity to 1. Because this cost is above the default engine threshold, the activity will be deferred when the process is executed. Click OK.

4. In the Navigator window, click the Verify button to verify your workflow.

5. From the File menu, select Save to save your work to your workflow definition file.

6. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

7. Close the data store.

8. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

9. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process, and use the Status Monitor Web pages to review the status of the process.

   - To test your work, log in as the approver and approve the Vacation Proposal notification.

   - Log in as the requestor and use the Status Monitor Web pages to review the status of the process. The process should be stopped at the Update Vacation Schedule function activity.

Running a Background Engine

10. Run a background engine for your item type to process the deferred activity and allow the process to continue. You can run a background engine through the Oracle Workflow Manager component of Oracle Applications Manager. You can also run a background engine using the WF_ENGINE.Background API.

11. To run a background engine through Oracle Applications Manager to process deferred activities:

    - Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

- Click the Workflow Manager link.

- On the Workflow System page, select Background Engines from the Submit Request For menu and click Go to submit the concurrent request through Oracle Self-Service Web Applications.

- Click Next and enter the following parameters:
  - Item Type: *XX* Vacation Proposal
  - Process Deferred: Yes
  - Process Timeout: No
  - Process Stuck: No

- Click Next in each remaining page of the wizard until the Summary page, and then click Submit.

12. To run the WF_ENGINE.Background API to process deferred activities, open SQL*Plus and enter the following command:

    ```
    Exec WF_ENGINE.Background('WFVACXX', NULL, NULL, TRUE, FALSE,
    FALSE);
    ```

13. Log in to Oracle Workflow as the requestor and use the Status Monitor Web pages to view the diagram for the process. The process should now have completed.

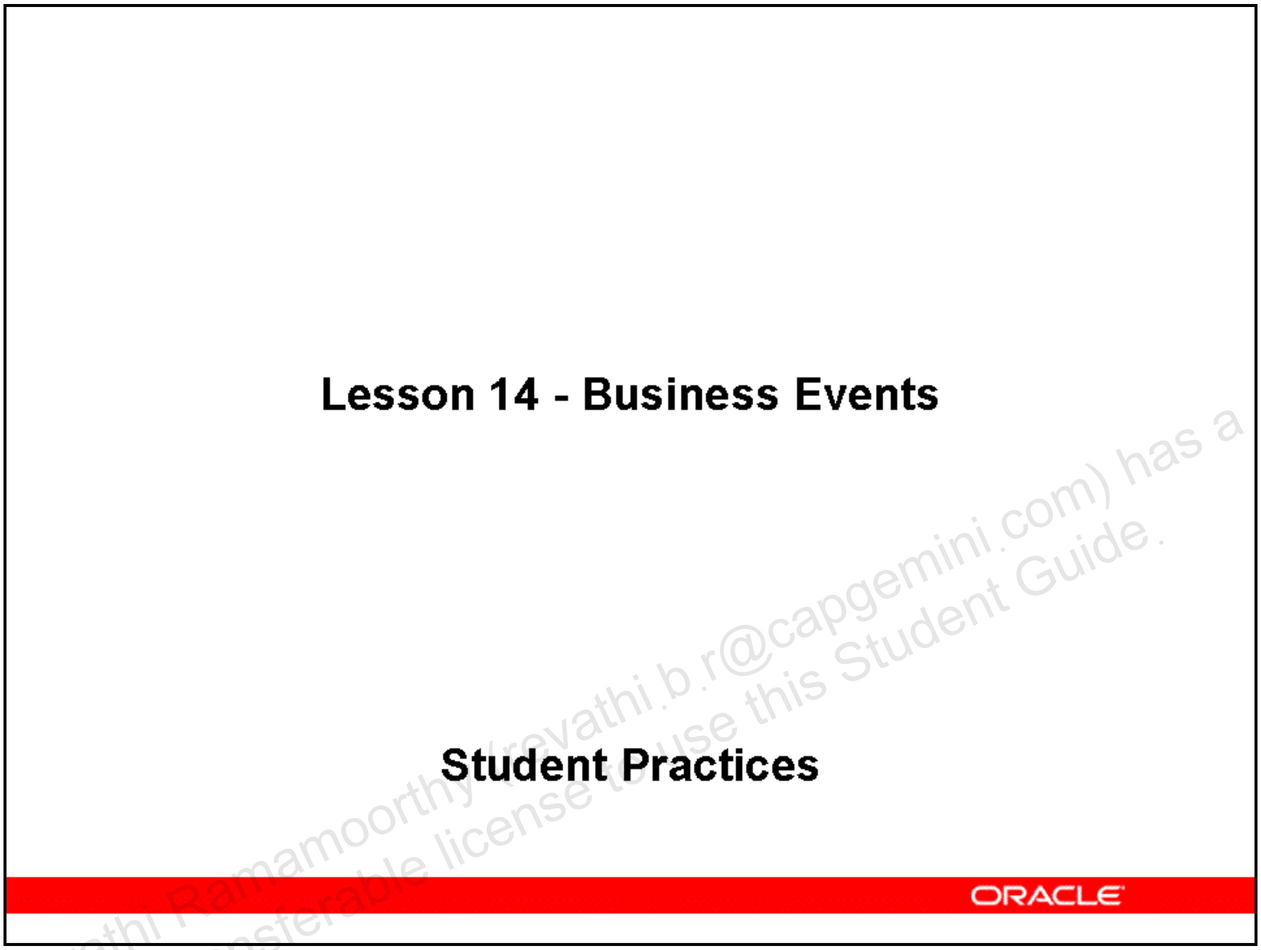
14. To remove the deferred processing so that you will not need to run the background engine in later practices, start the Oracle Workflow Builder, open the wfvac*XX*.wft data store, open the property pages for the Update Vacation Schedule function activity, and change the cost for the activity back to 0.

15. In the Navigator window, click the Verify button to verify your workflow.

16. From the File menu, select Save to save your work to your workflow definition file.

17. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

Lesson 14 - Business Events



Lesson 14 - Business Events

Student Practices

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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility that includes Event Manager functionality. The user name that you use to log in should have this responsibility assigned to it.

Tasks

Defining an Event

Define an event named *XX*.oracle.workflow.bes.vacation.scheduled in the Event Manager.

Solution – Defining an Event

Defining an Event

1. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

2. Click the Business Events link, and select Events in the horizontal navigation if the Events page is not already displayed. On the Events page, click the Create Event button to open the Create Event page.

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

4. In the Display Name field, enter *XX* Vacation Scheduled.

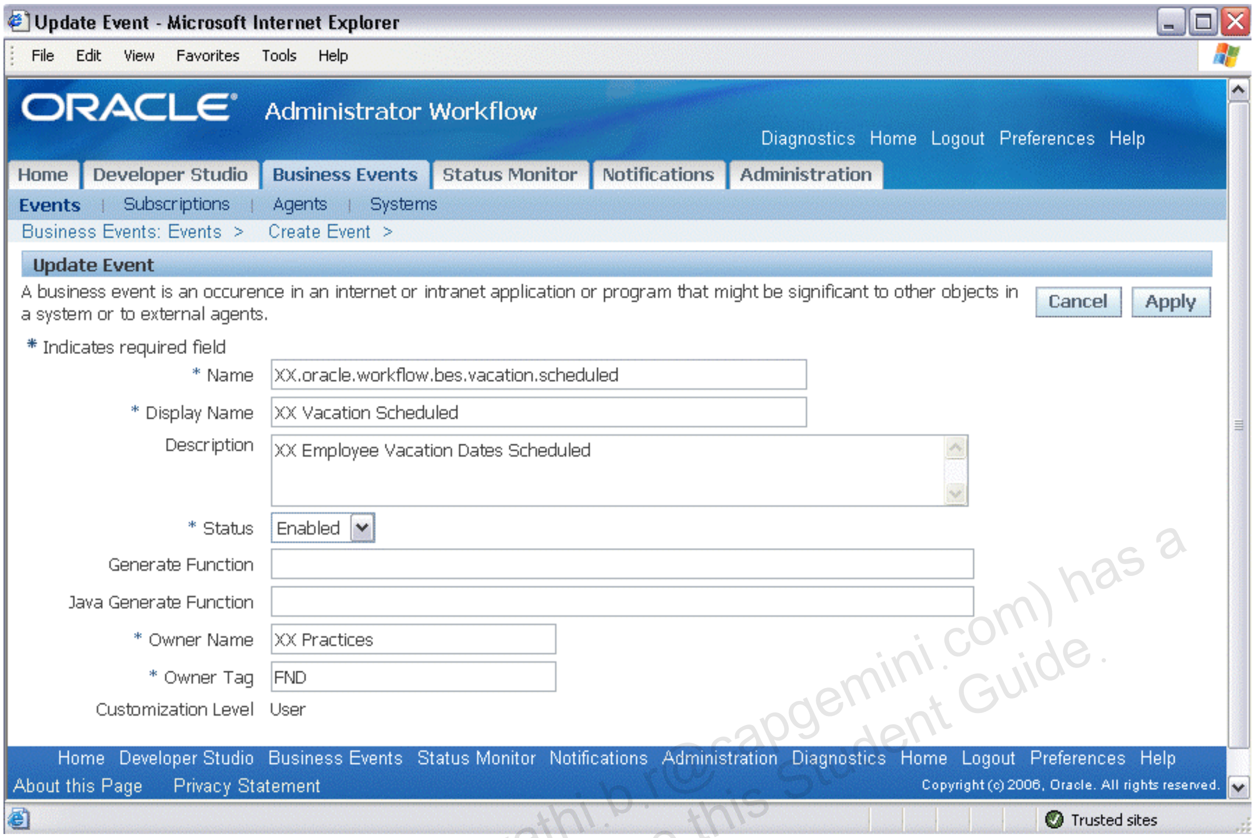
5. In the Description field, enter *XX* Employee Vacation Dates Scheduled.

6. In the Status field, select Enabled.

7. Leave the Generate Function and Java Generate Function fields blank.

8. In the Owner Name field, enter *XX* Practices.

9. In the Owner Tag field, enter FND.


Update Event - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ORACLE Administrator Workflow
Diagnostics Home Logout Preferences Help
Home Developer Studio Business Events Status Monitor Notifications Administration
Events | Subscriptions | Agents | Systems
Business Events: Events > Create Event >
Cancel
Apply
Home Developer Studio Business Events Status Monitor Notifications Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.
Trusted sites

Update Event

A business event is an occurence in an internet or intranet application or program that might be significant to other objects in a system or to external agents.

* Indicates required field

* Name	XX.oracle.workflow.bes.vacation.scheduled
* Display Name	XX Vacation Scheduled
Description	XX Employee Vacation Dates Scheduled
* Status	Enabled
Generate Function	
Java Generate Function	
* Owner Name	XX Practices
* Owner Tag	FND
Customization Level	User

10. Click the Apply button to save the event.

Practice - Raising an Event

Overview

In this practice, you will raise an event manually from the Event Manager Web pages. The event that 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 Event Manager Web pages; however, you can optionally review the workflow process on the Status Monitor Web pages 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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility that includes Event Manager functionality. The user name that you use to log in should have this responsibility assigned to it.

Tasks

Raising an Event

Raise the oracle.apps.wf.event.wf.send event manually from the Event Manager Web pages. This is the predefined Workflow Send Protocol event used in the Workflow Send Protocol process.

Solution – Raising an Event

Raising an Event

1. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

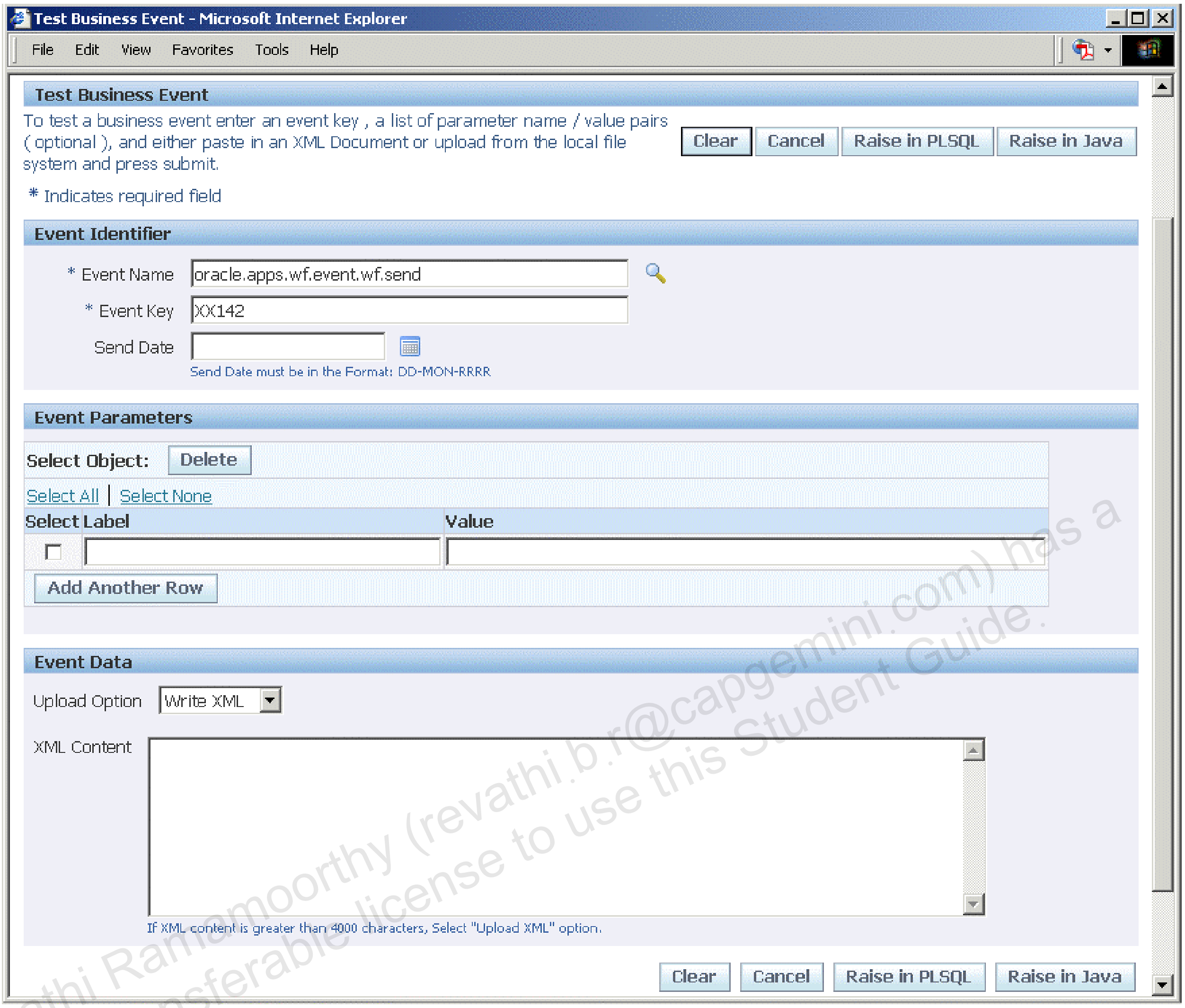
2. Click the Business Events link, and select Events in the horizontal navigation if the Events page is not already displayed. In the Search region of the Events page, enter the event name oracle.apps.wf.event.wf.send and click Go. Then, in the Results region, click the test icon for the oracle.apps.wf.event.wf.send event.

3. On the Test Business Event page, the Event Name oracle.apps.wf.event.wf.send is automatically displayed.

4. In the Event Key field, enter a unique event key such as *XX*142.

5. Leave the Send Date field and the Event Parameters region blank.

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



Test Business Event - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Test Business Event

To test a business event enter an event key , a list of parameter name / value pairs (optional), and either paste in an XML Document or upload from the local file system and press submit.

Clear
Cancel
Raise in PLSQL
Raise in Java

* Indicates required field

Event Identifier

* Event Name	oracle.apps.wf.event.wf.send
* Event Key	XX142
Send Date	

Send Date must be in the Format: DD-MON-RRRR

Event Parameters

Select	Label	Value
□		

Event Data

Upload Option: Write XML

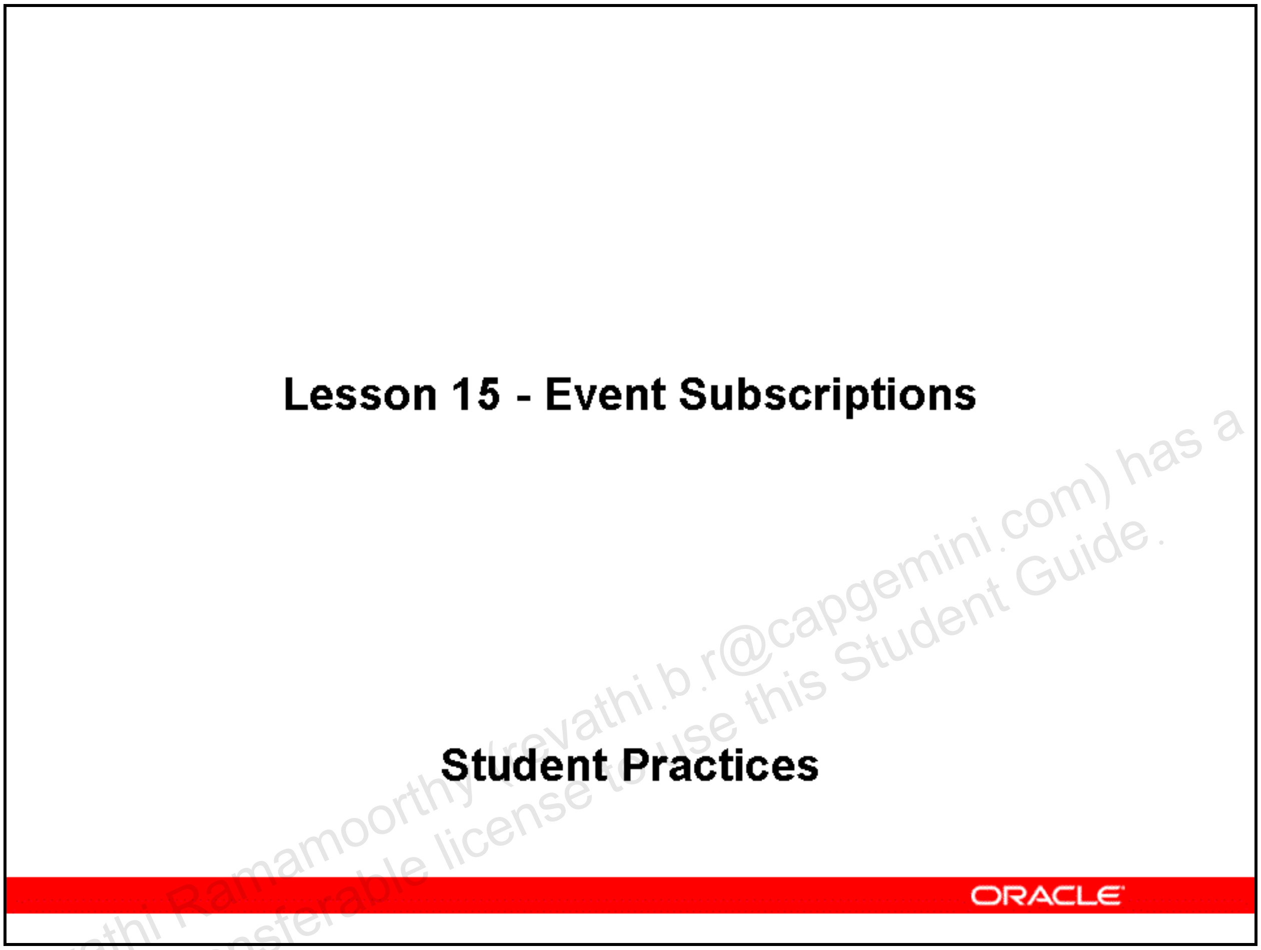
XML Content

If XML content is greater than 4000 characters, Select "Upload XML" option.

7. Click the Raise in PLSQL button to raise the event.

   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, because 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 on the Status Monitor Web pages. Select the Status Monitor tab and search for the process with the Workflow Send Protocol item type and with your event key as the item key.



Lesson 15 - Event Subscriptions

Student Practices

Practice - Defining a Subscription

Overview

In this practice, you will define and execute a subscription that sends an FYI message.

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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility that includes Event Manager functionality. The user name that you use to log in should have this responsibility assigned to it.

Tasks

Defining an Item Type with an FYI Message

Define an item type that contains an FYI message. The message should inform the recipient that a new responsibility has been created.

Defining an Event

Define an event named *XX*.oracle.apps.fnd.resp.insert in the Event Manager.

Note: This event is based on an Oracle Application Object Library event named oracle.apps.fnd.resp.insert. For the purposes of this practice, you will define your own copy of the event so that you can distinguish between the events raised by you and by your classmates, and verify that your subscription is executed when you manually raise your own event. In a production system, the application code that creates a new responsibility should raise the event programmatically at the same time.

Defining a Subscription

Define a subscription in the Event Manager. The triggering event for the subscription should be the *XX*.oracle.apps.fnd.resp.insert event. The subscription should send your FYI message to the SYSADMIN role.

Executing a Subscription

Raise an event to trigger your subscription. Then check the Worklist for the SYSADMIN role to verify that the message was sent.

Solution – Defining a Subscription

Defining an Item Type with an FYI Message

1. Start the Oracle Workflow Builder.

2. Create a new data store by selecting New from the File menu.

3. Create a new item type by selecting New > Item Type from the Edit menu.

4. Define the following properties for the new item type:
   - Internal Name: WFNOT*XX*
   - Display Name: *XX* Responsibility Notification
   - Persistence Type: Temporary
   - Number of Days: 5

- Click OK.

5. From the Edit menu, select New > Message.

6. Define the following properties in the Message tab for the new message:
   - Internal Name: *XX*RESPFYI
   - Display Name: *XX* Responsibility Created
   - Priority: Normal

- Click Apply.

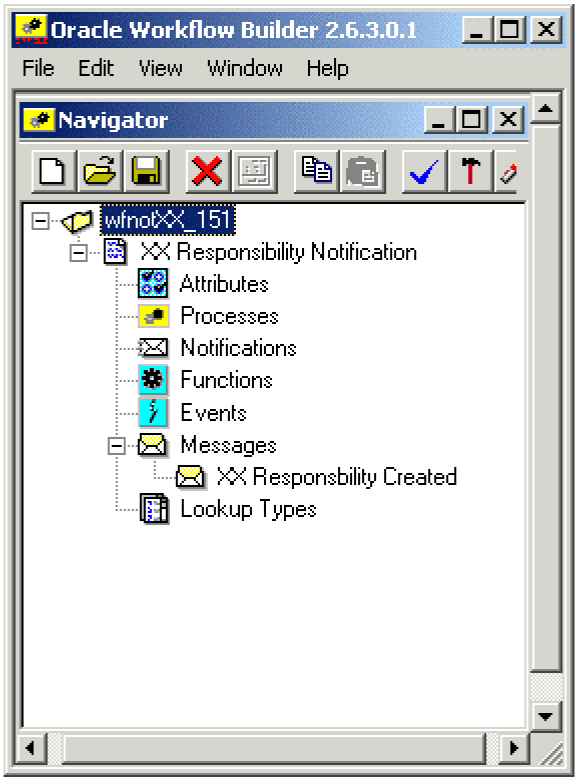
7. Enter the following text in the Text Body field in the Body tab for the message:

```
A new responsibility has been created.
```

- Click OK.

8. In the Navigator window, click the Verify button to verify your item type.

9. From the File menu, select Save As and save your new data store and item type to a workflow definition file named wfnotXX.wft.



10. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

11. Close the data store.

Defining an Event

12. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

13. Click the Business Events link, and select Events in the horizontal navigation if the Events page is not already displayed. On the Events page, click the Create Event button to open the Create Event page.

14. In the Name field, enter *XX*.oracle.apps.fnd.resp.insert as the internal name of the event.

15. In the Display Name field, enter *XX* Responsibility Created.

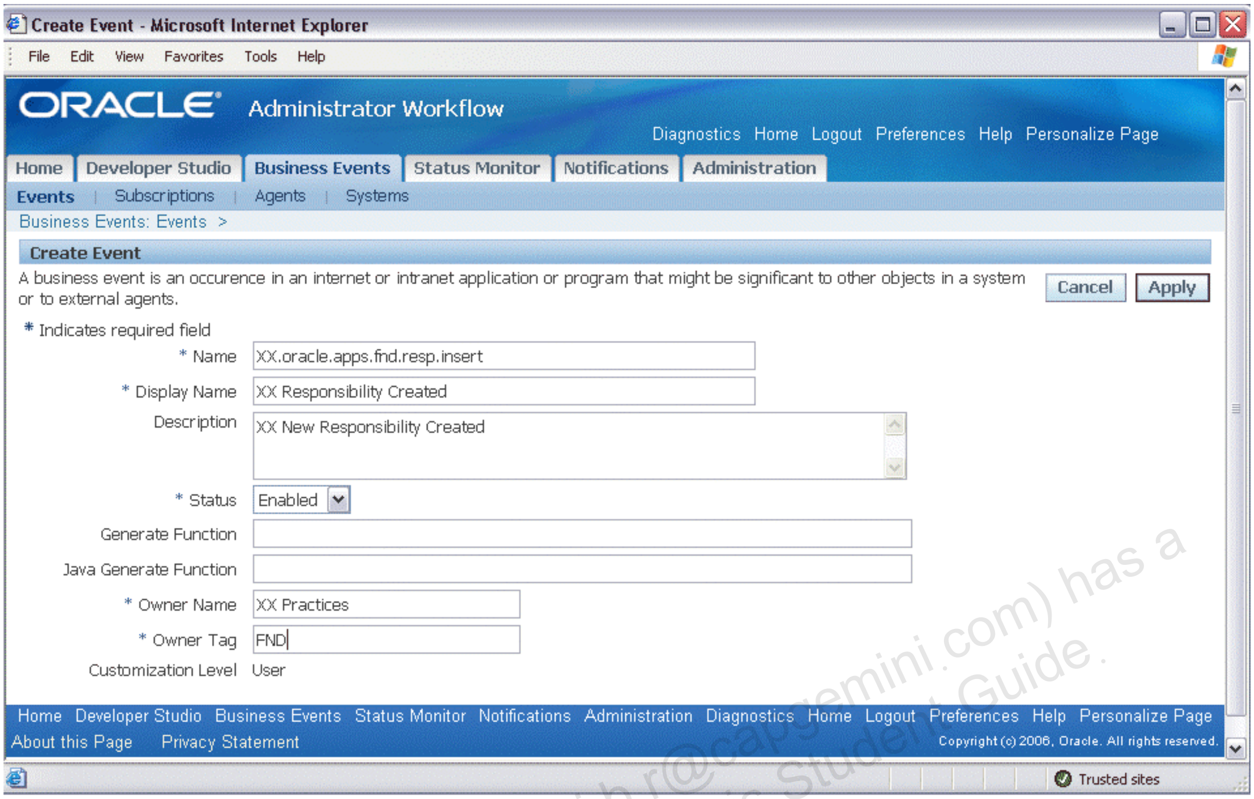
16. In the Description field, enter *XX* New Responsibility Created.

17. In the Status field, select Enabled.

18. Leave the Generate Function and Java Generate Function fields blank.

19. In the Owner Name field, enter *XX* Practices.

20. In the Owner Tag field, enter FND.


Create Event - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ORACLE Administrator Workflow
Diagnostics Home Logout Preferences Help Personalize Page
Home Developer Studio Business Events Status Monitor Notifications Administration
Events | Subscriptions | Agents | Systems
Business Events: Events >
Cancel
Apply
Home Developer Studio Business Events Status Monitor Notifications Administration Diagnostics Home Logout Preferences Help Personalize Page
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.
Trusted sites

Create Event

A business event is an occurence in an internet or intranet application or program that might be significant to other objects in a system or to external agents.

* Indicates required field

* Name	XX.oracle.apps.fnd.resp.insert
* Display Name	XX Responsibility Created
Description	XX New Responsibility Created
* Status	Enabled
Generate Function	
Java Generate Function	
* Owner Name	XX Practices
* Owner Tag	FND
Customization Level	User

21. Click the Apply button to save the event.

Defining a Subscription

22. Select Subscriptions in the horizontal navigation. On the Event Subscriptions page, click the Create Subscription button to open the Create Event Subscription page.

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

24. In the Source Type field, select Local.

25. In the Event Filter field, select the *XX*.oracle.apps.fnd.resp.insert event that you defined previously.

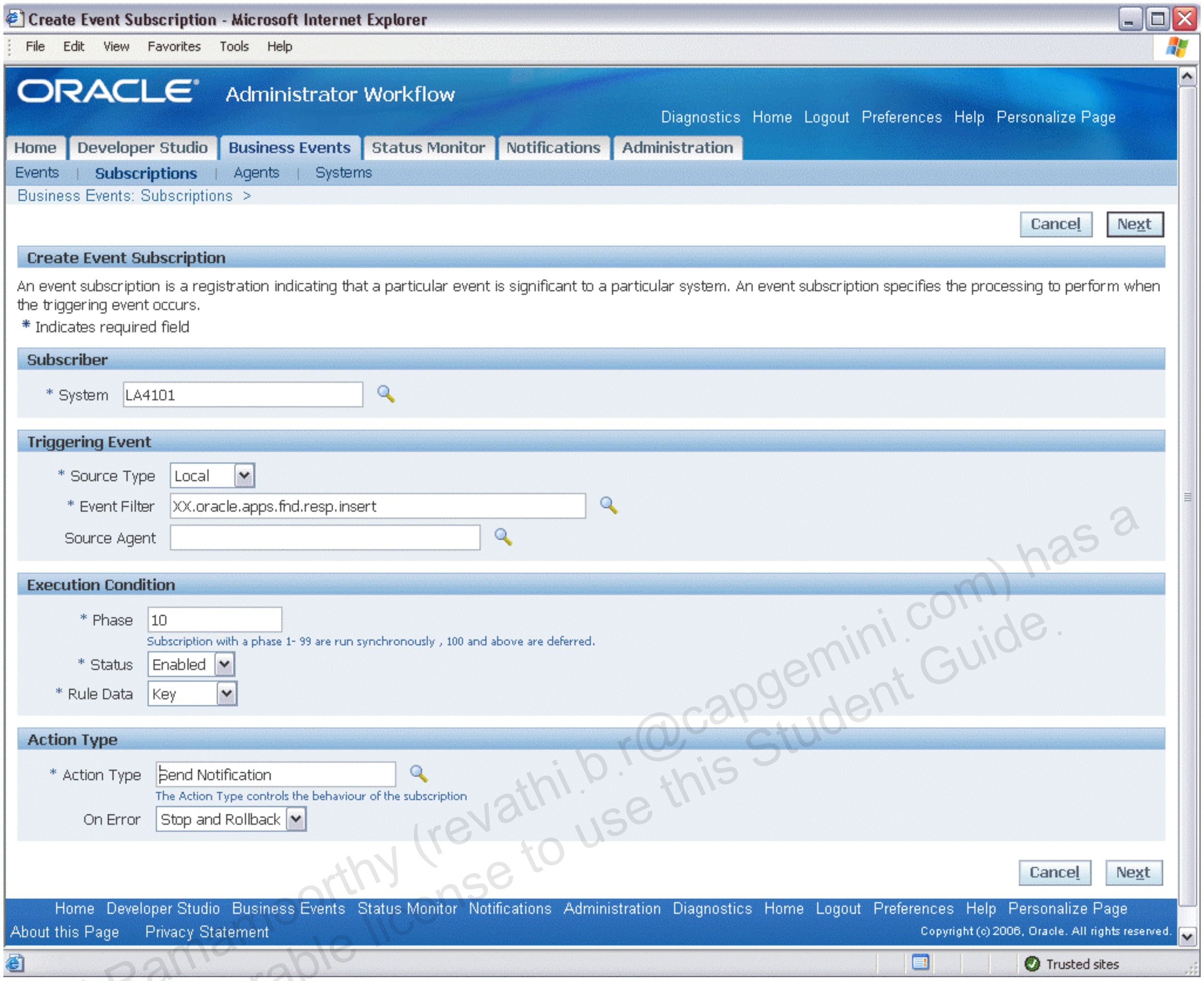
26. Leave the Source Agent field blank.

27. Enter 10 in the Phase field.

28. In the Status field, select Enabled.

29. In the Rule Data field, select Key.

30. Select Send Notification in the Action Type field, and select Stop and Rollback in the On Error field. Then click Next.


Create Event Subscription - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ORACLE Administrator Workflow
Diagnostics Home Logout Preferences Help Personalize Page
Home Developer Studio Business Events Status Monitor Notifications Administration
Events | Subscriptions | Agents | Systems
Business Events: Subscriptions >
Cancel Next

Create Event Subscription

An event subscription is a registration indicating that a particular event is significant to a particular system. An event subscription specifies the processing to perform when the triggering event occurs.

* Indicates required field

Subscriber

* System LA4101

Triggering Event

* Source Type: Local

* Event Filter: XX.oracle.apps.fnd.resp.insert

Source Agent:

Execution Condition

* Phase: 10

Subscription with a phase 1- 99 are run synchronously , 100 and above are deferred.

* Status: Enabled

* Rule Data: Key

Action Type

* Action Type: Send Notification

The Action Type controls the behaviour of the subscription

On Error: Stop and Rollback

Cancel Next
Home Developer Studio Business Events Status Monitor Notifications Administration Diagnostics Home Logout Preferences Help Personalize Page
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.
Trusted sites

31. Select WFNOT*XX* in the Message Type field and *XX*RESPFYI in the Message Name field.

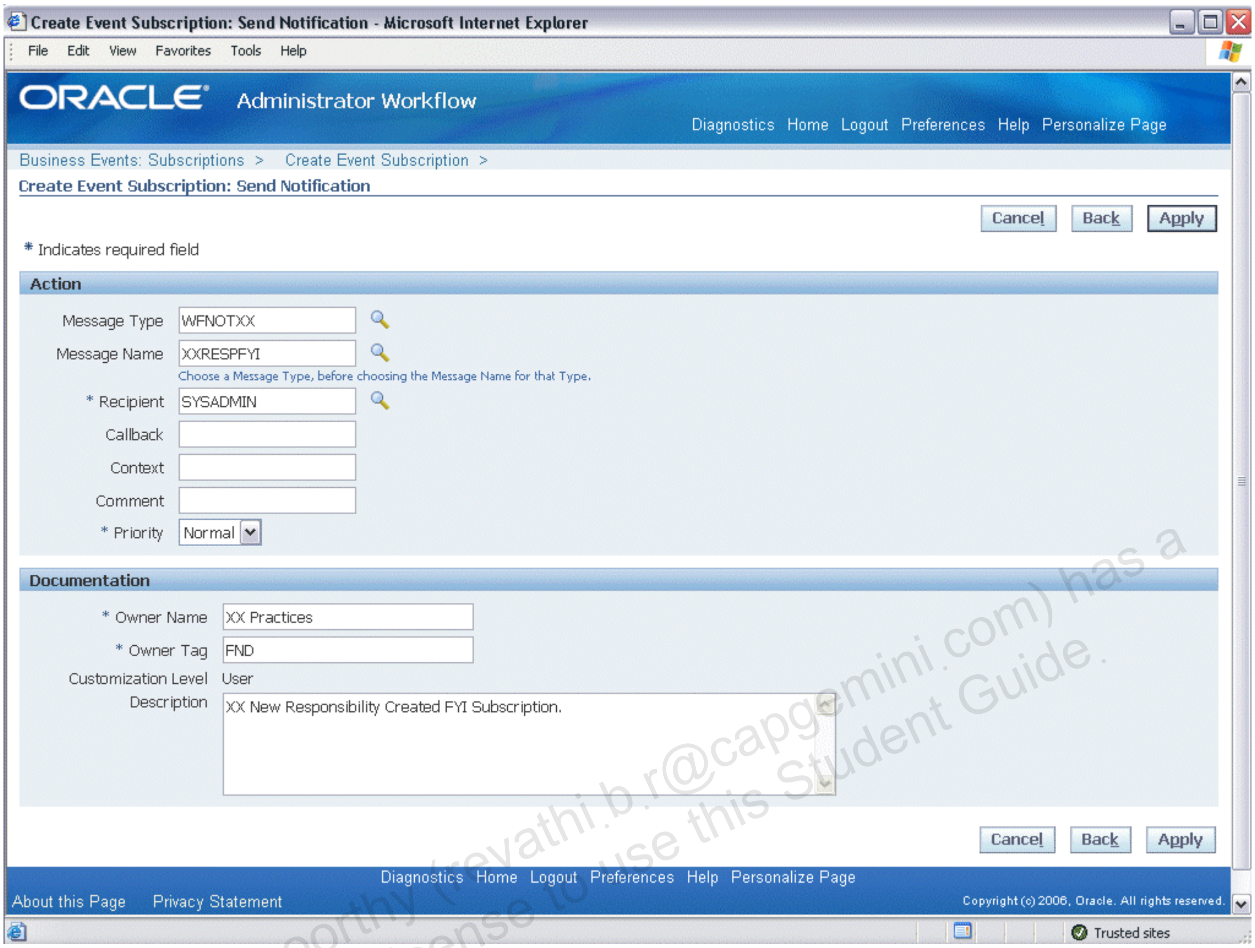
32. Select the SYSADMIN role in the Recipient field.

33. Leave the Callback, Context, and Comment fields blank. Leave the Priority field set to the default value, which is Normal.

34. In the Owner Name field, enter *XX* Practices.

35. In the Owner Tag field, enter FND.

36. In the Description field, enter *XX* New Responsibility Created FYI Subscription.



Create Event Subscription: Send Notification - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ORACLE Administrator Workflow
Diagnostics Home Logout Preferences Help Personalize Page
Business Events: Subscriptions > Create Event Subscription >

Create Event Subscription: Send Notification

* Indicates required field

Action

Message Type	WFNOTXX
Message Name	XXRESPFYI Choose a Message Type, before choosing the Message Name for that Type.
* Recipient	SYSADMIN
Callback	
Context	
Comment	
* Priority	Normal

Documentation

* Owner Name	XX Practices
* Owner Tag	FND
Customization Level	User
Description	XX New Responsibility Created FYI Subscription.

Cancel Back Apply
Diagnostics Home Logout Preferences Help Personalize Page
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.
Trusted sites

37. Click the Apply button to save the subscription.

Executing a Subscription

38. Select Events in the horizontal navigation. In the Search region of the Events page, enter the name of your *XX*.oracle.apps.fnd.resp.insert event and click Go. Then, in the Results region, select the test icon for the *XX*.oracle.apps.fnd.resp.insert event.

39. On the Test Business Event page, the Event Name field displays the *XX*.oracle.apps.fnd.resp.insert event.

40. In the Event Key field, enter a unique event key such as *XX*151.

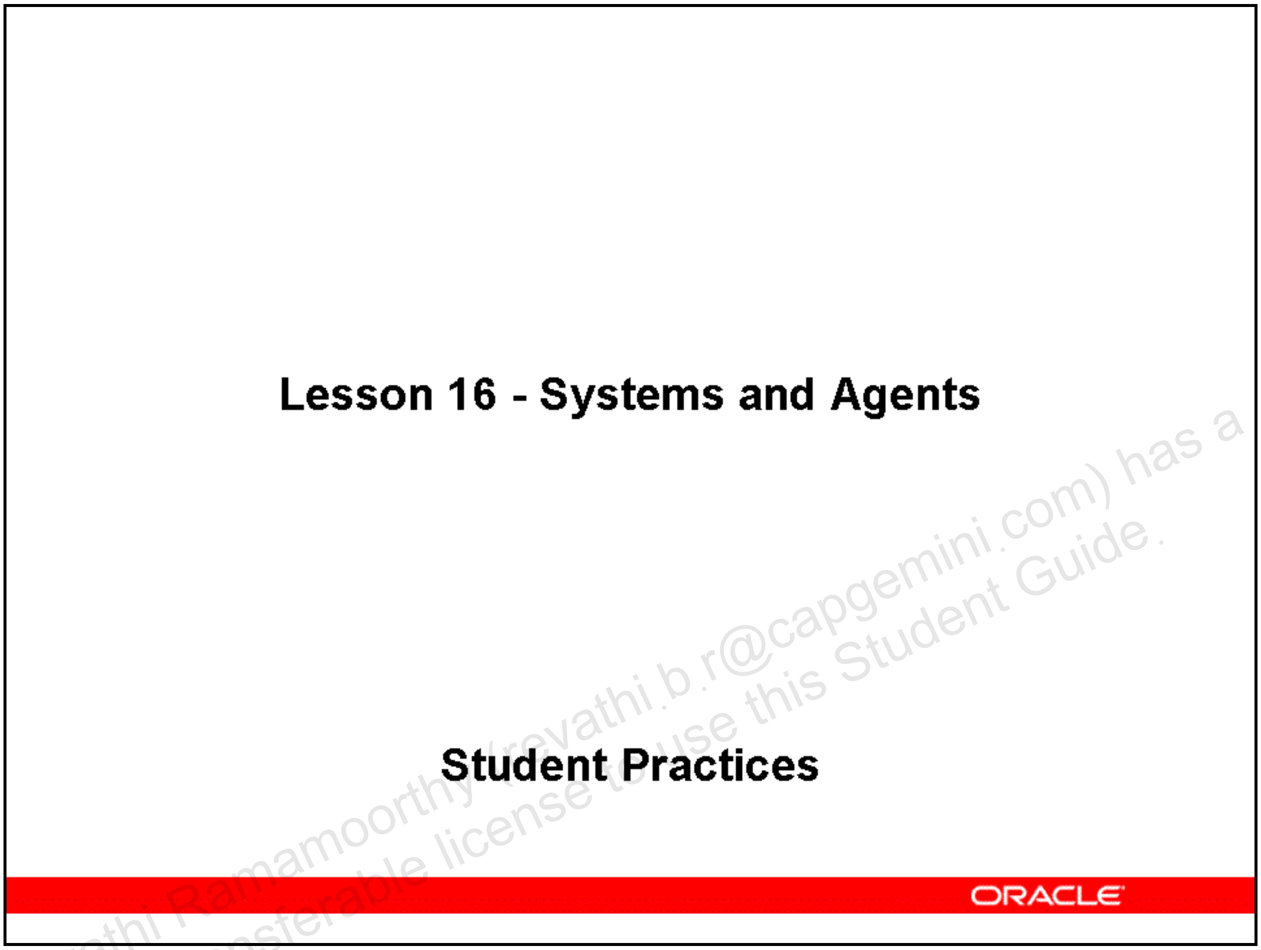
41. Leave the Send Date field, the Event Parameters region, and the Event Data region blank.

42. Click the Raise in PLSQL button to raise the event.

43. Log off and log in again as the SYSADMIN user. You can use either a Workflow administrator or Workflow user responsibility.

44. Check the Worklist to verify that the *XX* Responsibility Created message was sent successfully.

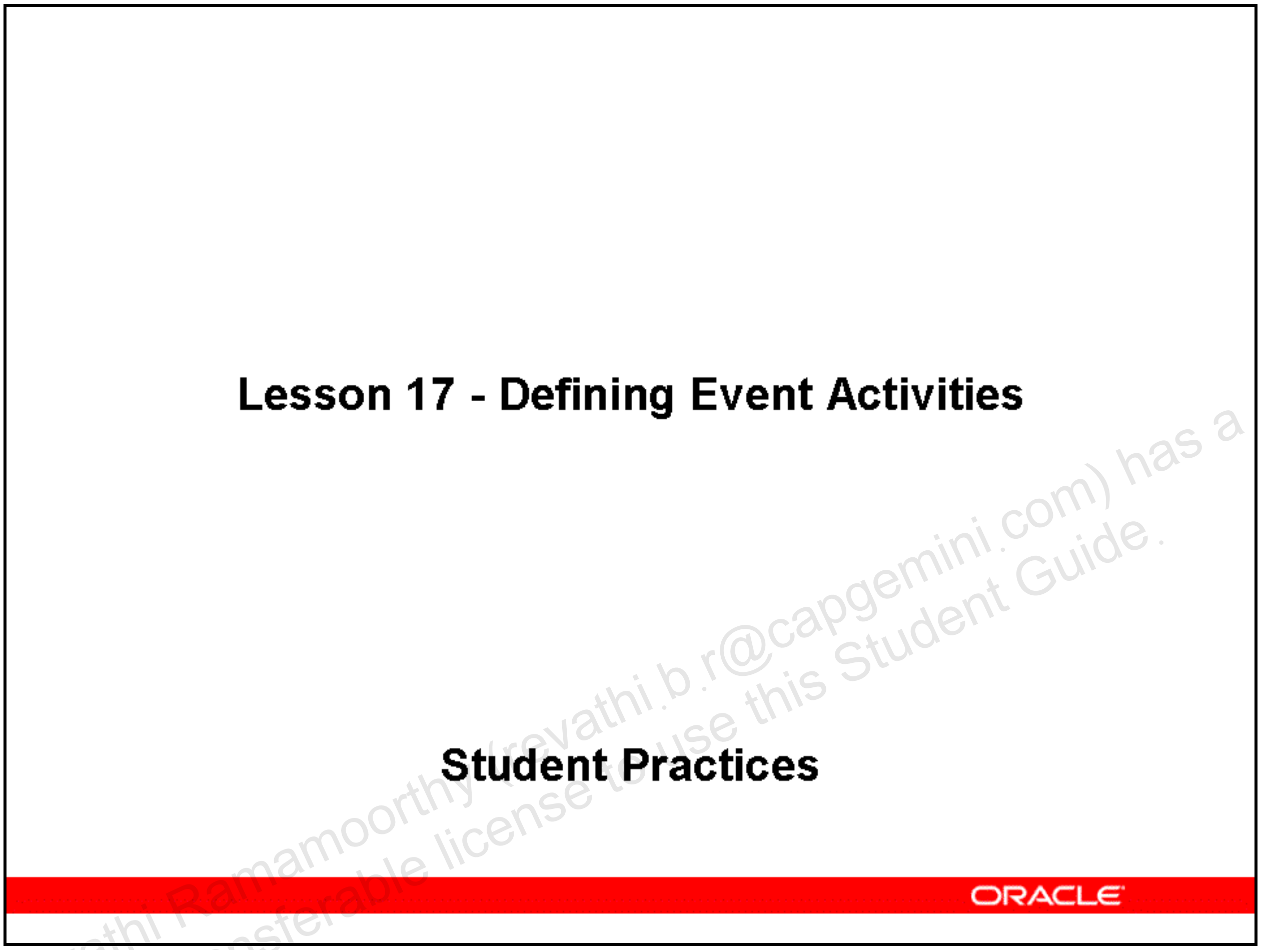
Lesson 16 - Systems and Agents



Lesson 16 - Systems and Agents

Student Practices

There are no practices or demonstrations in this lesson.



Lesson 17 - Defining Event Activities

Student Practices

Practice - Defining Event Activities

Overview

In this practice, you will add a Raise event activity to the Vacation Proposal process that you created in the Creating a Workflow Process practice. Then you will define another a workflow process started by a Receive event activity to send a notification when the event is received.

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 that you are defining. In this way, you can ensure that the definitions you create are unique.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility that includes Event Manager functionality. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining a Raise Event Activity

Add a Raise event activity to the Vacation Proposal process.

Defining an Item Type with a Receive Event Activity

Define a new item type that includes a Receive event activity as well as some item attributes related to the event and a notification activity.

Defining a Process Started by an Event

Define the process diagram for the process activity in the new item type. The diagram should include the Receive event activity, the notification activity, and a standard End activity.

Defining a Subscription to Send an Event to a Workflow Process

Define a subscription to send the event to the new workflow process when the event is raised.

Testing Event-based Processes

Run the Vacation Proposal process to raise the event, trigger the subscription, and run the new workflow process. Check that both the main workflow process that raised the event and the new workflow process that received the event have completed successfully.

Solution – Defining Event Activities

Defining a Raise Event Activity

1. Start the Oracle Workflow Builder.

2. From the File menu, select Open to open the wfvacXX.wft data store that you defined in the Creating a Workflow practice.

3. In the navigator tree, select your *XX* Vacation Proposal item type.

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

5. Define the following properties for the item attribute:
   - Internal Name: VACEVENTNAME
   - Display Name: Vacation Event Name
   - Type: Text
   - Default Value: *XX*.oracle.workflow.bes.vacation.scheduled

   Note: The event name is case-sensitive. Make sure that you enter the exact event name that you created in the Defining an Event practice.

   - Click OK.

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

7. Define the following properties for the item attribute:
   - Internal Name: VACEVENTKEY
   - Display Name: Vacation Event Key
   - Type: Text

   - Click OK.

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

9. Define the following properties for the attribute:
   - Internal Name: VACEVENTDATA
   - Display Name: Vacation Event Data
   - Type: Text

   - Click OK.

10. From the Edit menu, select New > Event.

11. Define the following properties for the event activity:
    - Internal Name: RAISE_VAC_EVENT
    - Display Name: Raise Vacation Scheduled Event

12. Leave the icon and cost set to the default values.

13. Select Raise as the event action.

14. Click OK.

15. In the Navigator window, drag and drop the Requestor, Approver, From Date, and To Date item attributes onto the Raise Vacation Scheduled Event activity to create the corresponding event activity attributes with those item attributes as their default values.

    Any activity attributes defined for a Raise event activity node will be included as parameters in the parameter list within the event message structure. When the event message is received by the second process, the Workflow Engine will set the event parameters as item type attributes for that process.

16. Open the process diagram window for the Vacation Proposal process.

17. Delete the transition between the Update Vacation Schedule node and the End (Approve) node.

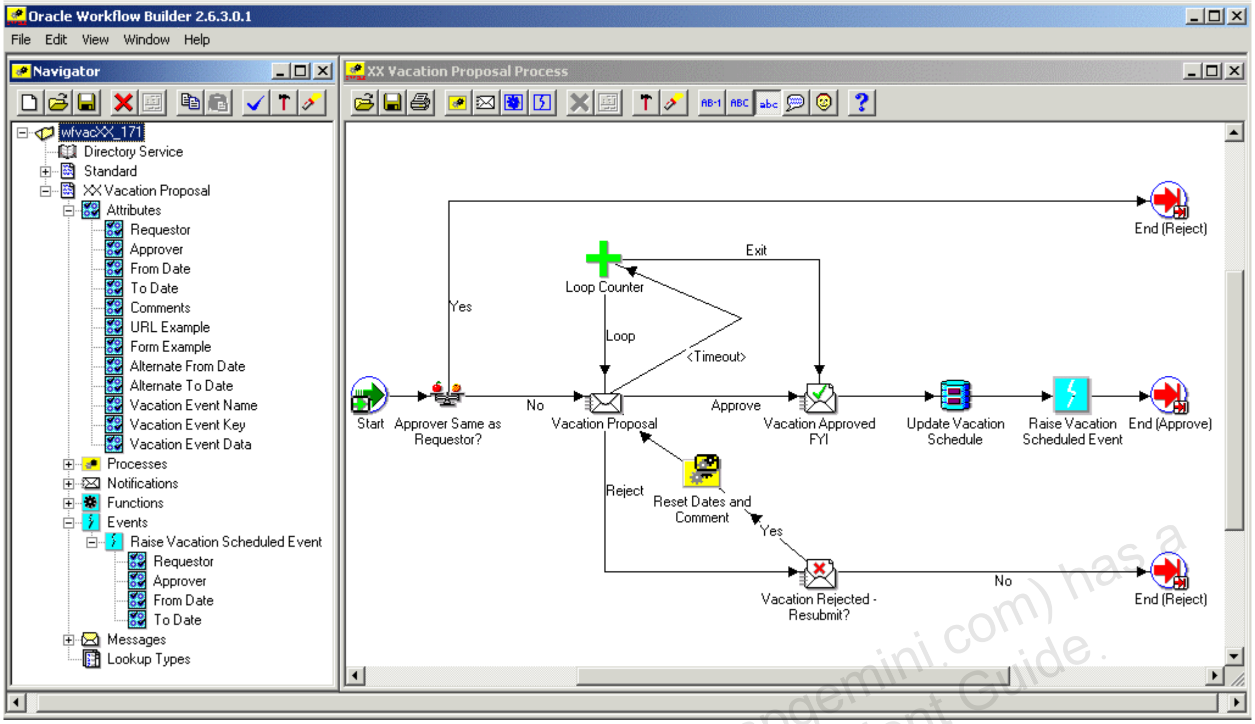
18. Drag the Raise Vacation Scheduled Event activity into the process diagram and position it between the Update Vacation Schedule node and the End (Approve) node.

19. Open the property pages for the Raise Vacation Scheduled Event node, and select the Event Details tab. Select Item Attribute as the Event Name type, the Vacation Event Name attribute as the Event Name value, the Vacation Event Key attribute as the Event Key value, and the Vacation Event Data attribute as the Event Data value. Click OK.

20. Draw transitions from the Update Vacation Schedule node to the Raise Vacation Scheduled Event node, and from the Raise Vacation Scheduled Event node to the End (Approve) node.

21. In the Navigator window, click the Verify button to verify your workflow.

22. From the File menu, select Save to save your work to your workflow definition file.



23. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

24. Close the data store.

Defining an Item Type with a Receive Event Activity

25. From the File menu, select Quick Start Wizard.

26. 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

- Click OK.

27. Close the Vacation Salary process diagram window. You will define the diagram for this process in the next task.

28. In the Navigator window, select the Vacation Salary item type.

29. From the Edit menu, select New > Attribute.

30. Define the following properties for the item attribute:
    - Internal Name: VACEVENTNAME
    - Display Name: Vacation Event Name
    - Type: Text

- Click OK.

31. From the Edit menu, select New > Attribute.

32. Define the following properties for the item attribute:
    - Internal Name: VACEVENTKEY
    - Display Name: Vacation Event Key
    - Type: Text

- Click OK.

33. From the Edit menu, select New > Attribute.

34. Define the following properties for the attribute:
    - Internal Name: VACEVENTMESSAGE
    - Display Name: Vacation Event Message
    - Type: Event

- Click OK.

35. From the Edit menu, select New > Attribute.

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

- Click OK.

37. From the Edit menu, select New > Attribute.

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

- Click OK.

39. From the Edit menu, select New > Attribute.

40. Define the following properties for the item attribute:
    - Internal Name: FROM_DATE
    - Display Name: From Date

  - Type: Date
  - Format: DD-MON-RRRR

- Click OK.

41. From the Edit menu, select New > Attribute.

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

- Click OK.

43. From the Edit menu, select New > Event.

44. Define the following properties for the event activity:
    - Internal Name: REC_VAC_EVENT
    - Display Name: Receive Vacation Scheduled Event

45. Leave the icon and cost set to the default values.

46. Select Receive as the event action.

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

48. Click OK.

49. From the Edit menu, select New > Message.

50. Define a Vacation Salary message that informs the approver that the requestor will be paid from the vacation salary account during the vacation dates that have been approved.

- In the Body tab, enter the following message text body including the message attribute tokens for the requestor, approver, and the start and end dates of the proposed vacation:

```
&REQUESTOR will be paid from the vacation salary account
during the vacation dates that you approved.

Requestor: &REQUESTOR

Approver: &APPROVER

From Date: &FROM_DATE

To Date: &TO_DATE
```

- Set the message priority to Normal.

- Drag and drop the appropriate item attributes onto the message to create the corresponding message attributes with those item attributes as their default values.

51. From the Edit menu, select New > Notification.

52. Define the following properties for the notification:
    - Internal Name: NTF_VAC_SALARY
    - Display Name: Vacation Salary Notification
    - Icon: NTF_INFO.ICO
    - Message: Vacation Salary

- Click OK.

53. From the File menu, select Save As and save your new data store and item type to a workflow definition file named wfbes*XX*.wft.

Defining a Process Started by an Event

54. In the navigator tree, select the Vacation Salary process.

55. Double-click the Vacation Salary process to display the process diagram window.

56. Delete the default Start activity from the process diagram.

57. Drag the Receive Vacation Scheduled Event activity into the process diagram.

58. Double-click the Receive Vacation Scheduled Event activity node and select the Node tab.

59. In the Start/End field, select Start. Click Apply.

60. Select the Event Details tab.

61. In the Event Name field, select the Vacation Event Name item attribute.

62. In the Event Key field, select the Vacation Event Key item attribute.

63. In the Event Message field, select the Vacation Event Message item attribute.

64. Click OK.

65. Drag the Vacation Salary Notification activity into the process diagram between the Receive Vacation Scheduled Event node and the End node.

66. Double-click the Vacation Salary Notification node and select the Node tab. Set the performer for the node to the Approver item attribute. Click OK.

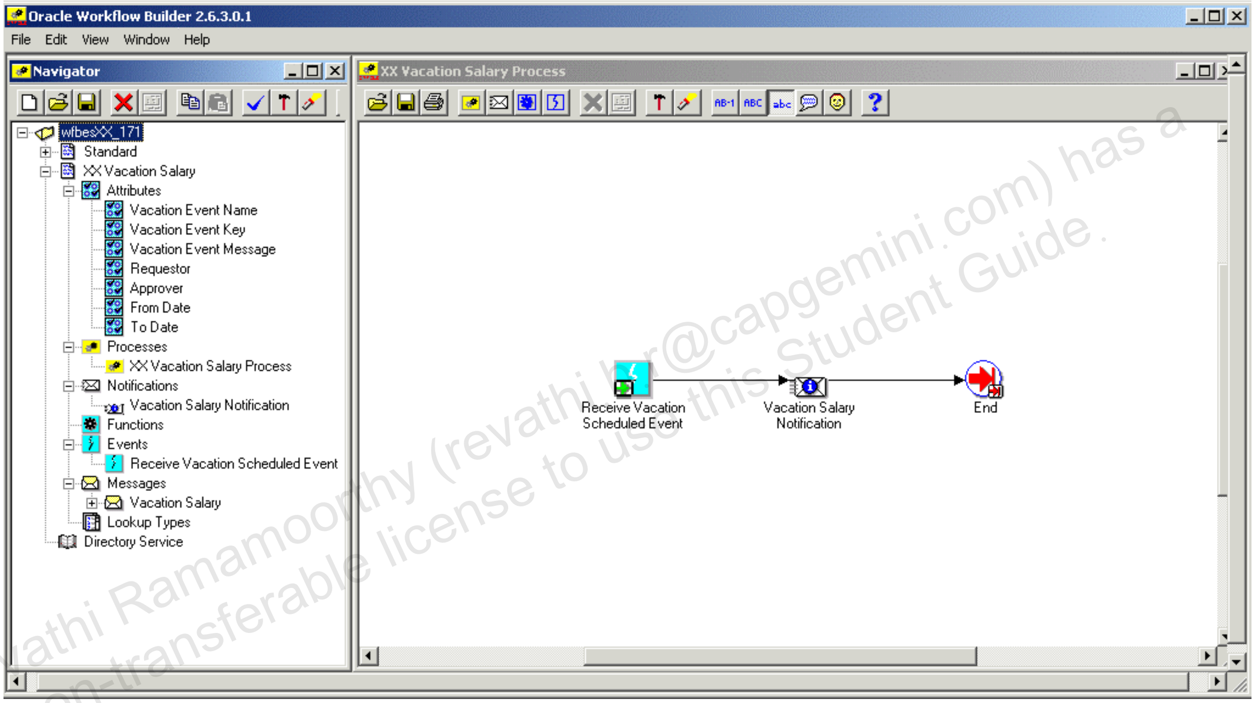
67. Draw a transition from the Receive Vacation Scheduled Event activity to the Vacation Salary Notification activity.

68. Draw a transition from the Vacation Salary Notification activity to the End activity.

69. Double-click the End activity node and select the Node tab. Ensure that the Start/End field is set to End. Click OK.

70. In the Navigator window, click the Verify button to verify your workflow.

71. From the File menu, select Save to save your work to your workflow definition file.



72. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

73. Close the data store.

Defining a Subscription to Send an Event to a Workflow Process

74. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

75. Click the Business Events link, and select Subscriptions in the horizontal navigation. On the Event Subscriptions page, click the Create Subscription button to open the Create Event Subscription page.

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

77. In the Source Type field, select Local.

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

79. Leave the Source Agent field blank.

80. Enter 30 in the Phase field.

81. In the Status field, select Enabled.

82. In the Rule Data field, select Key.

83. Select Launch Workflow in the Action Type field, and select Stop and Rollback in the On Error field. Then click Next.

84. In the Workflow Type field, select your *XX*VACSAL item type.

85. In the Workflow Process field, select your *XX*_VACATION_SALARY process.

86. Leave the Priority field set to the default value, which is Normal. Leave the Additional Options field and Subscription Parameters region blank.

87. In the Owner Name field, enter *XX* Practices.

88. In the Owner Tag field, enter FND.

89. In the Description field, enter *XX* Send Vacation Scheduled Event to Vacation Salary Workflow.

90. Click the Apply button to save the subscription.

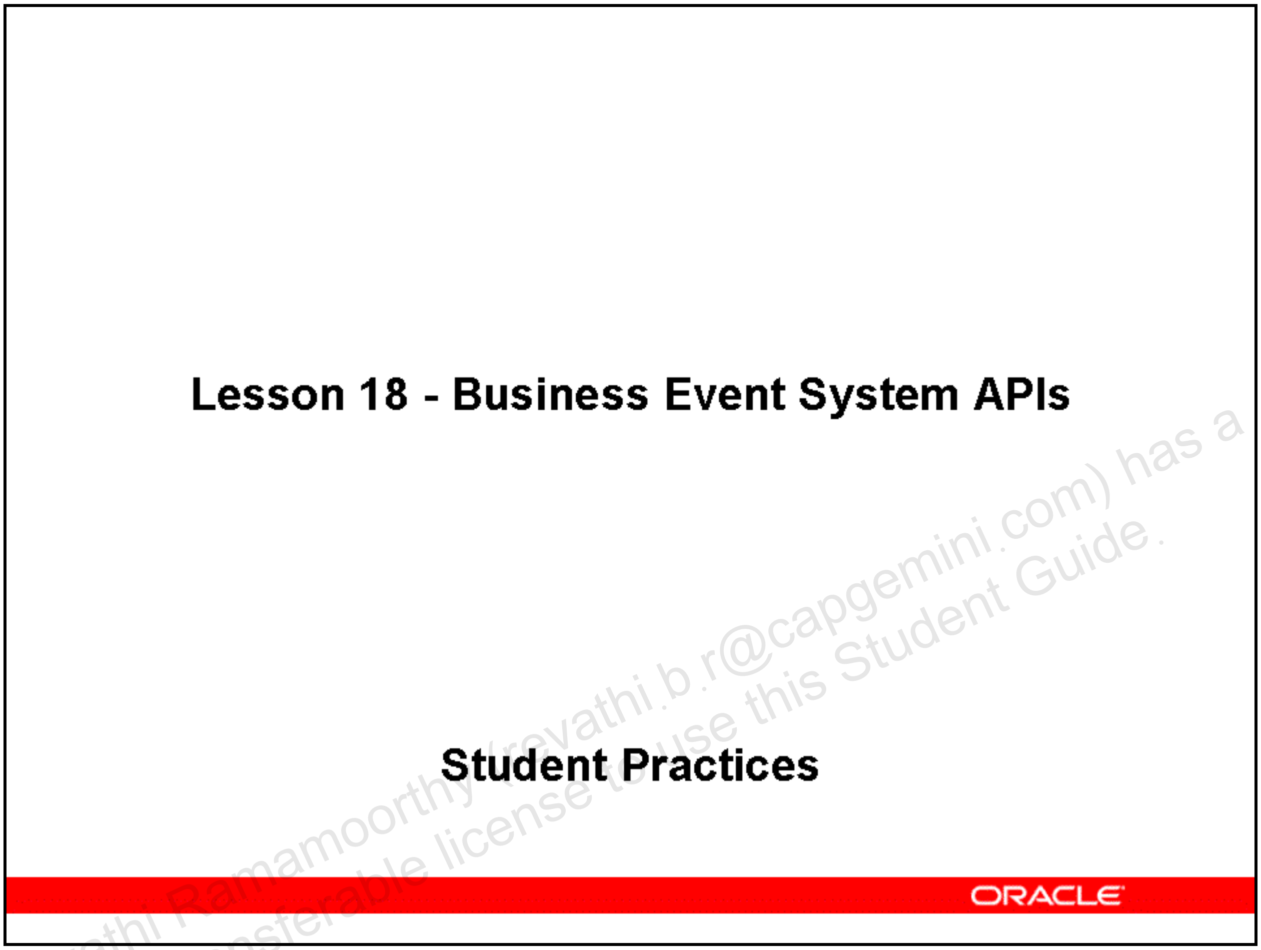
Testing Event-based Processes

91. Use the Developer Studio to launch your Vacation Proposal workflow process and test your work. In addition to entering a requestor, approver, from date, and to date, also enter a unique event key such as *XX*171E in the Vacation Event Key field and `<DATA>Vacation has been scheduled</DATA>` as the vacation event data. You can use the Worklist Web pages to view the notifications sent by the processes and use the Status Monitor Web pages to review the status of the Vacation Proposal process and the Vacation Salary process.

    - Approve the vacation proposal.

    - When the Vacation Proposal process raises the *XX*.oracle.workflow.bes.vacation.scheduled event, the Business Event System should run all local subscriptions to the event, including the subscription that sends the event to start the Vacation Salary process. Review the notification sent by the Vacation Salary process to confirm that the process was run and completed successfully. Also review both the Vacation Proposal process and the Vacation Salary process in the Status Monitor to ensure that both processes completed successfully.

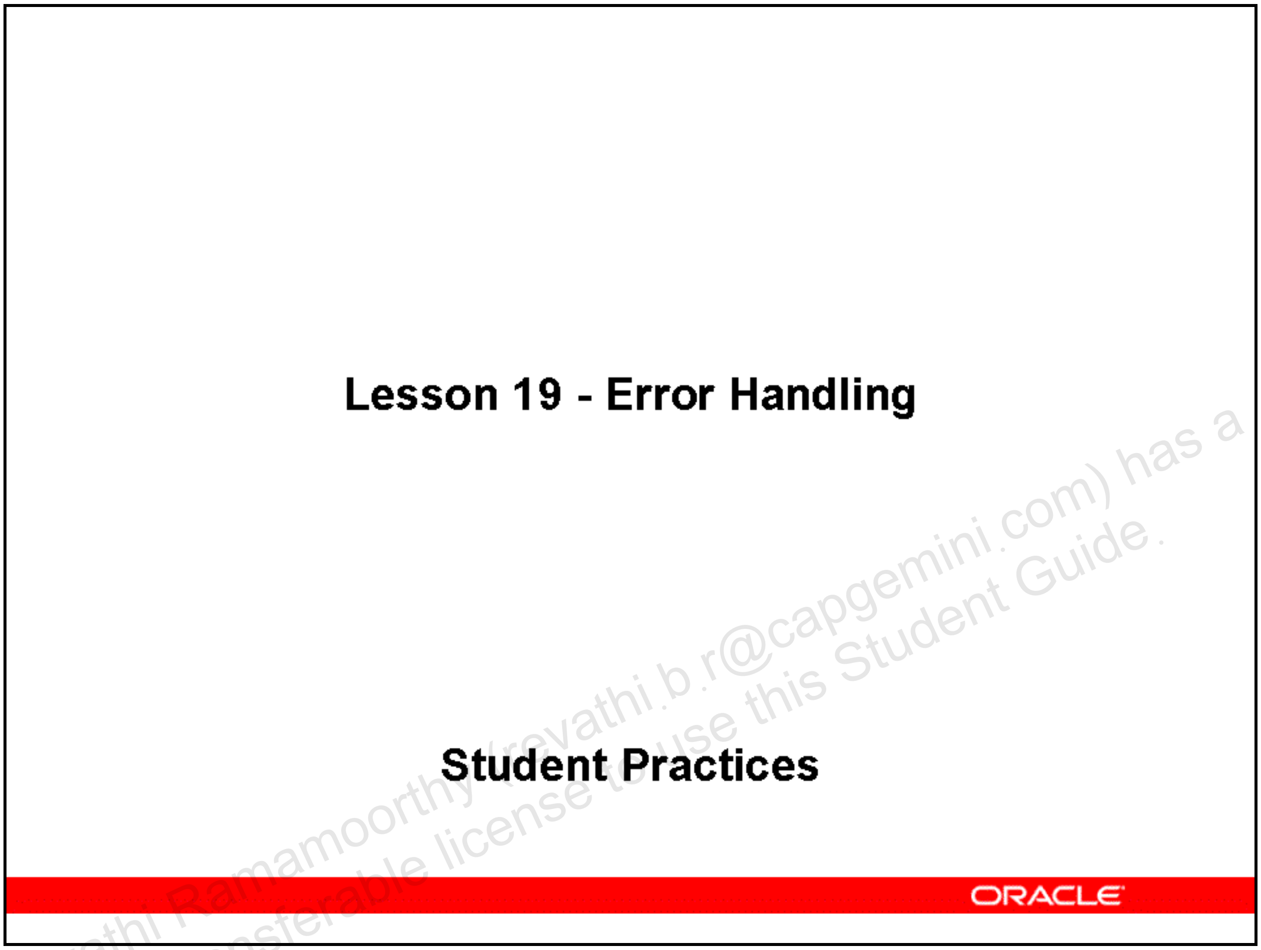
Lesson 18 - Business Event System APIs



Lesson 18 - Business Event System APIs

Student Practices

There are no practices or demonstrations in this lesson.



Lesson 19 - Error Handling

Student Practices

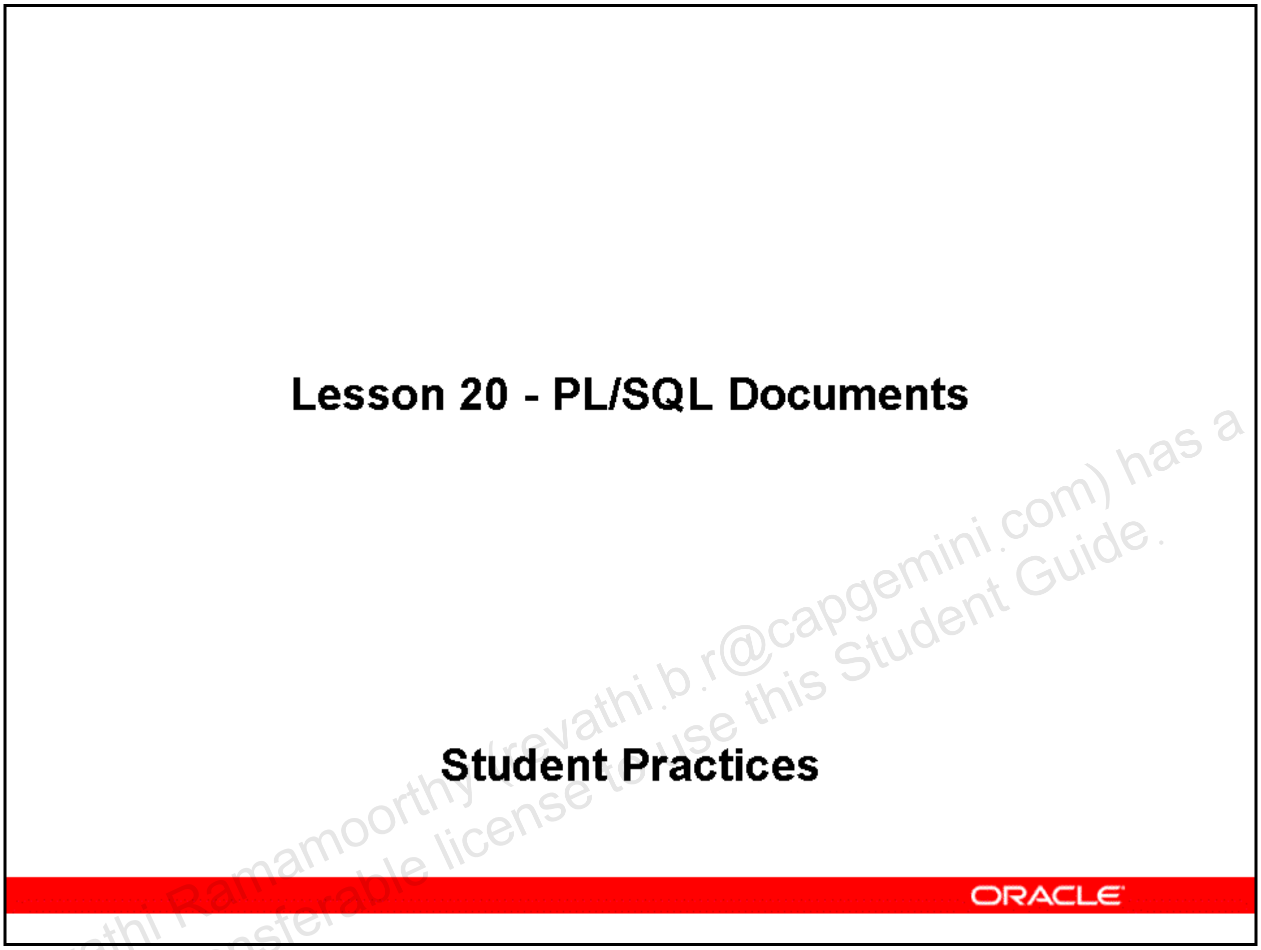
Guided Demonstration - Error Handling

1. Start the Oracle Workflow Builder.

2. From the File menu, select Open. Select the Database option and connect to the class database, using the database user name, password, and connect string for the class. In the Show Item Types window, move the System: Error item type to the Visible list and click OK.

3. Expand the System: Error item type in the navigator tree and display the diagrams for the following error processes:
   - Default Error Process
   - Retry-only
   - Default Event Error Process
   - Default Event Error Process (One Retry Option)

4. Discuss how these processes handle errors in Workflow Engine processing and Business Event System subscription processing.



Lesson 20 - PL/SQL Documents

Student Practices

Practice - Using a PL/SQL Document Attribute

Overview

In this practice, you will implement a PL/SQL document attribute in the Vacation Proposal item type that you created in the Creating a Workflow Process practice. The PL/SQL document will show the scheduled vacation for a vacation proposal requestor as stored in your WFVAC*XX*_VACATION_SCHEDULE table. For this practice, you will use a predefined PL/SQL procedure for PL/SQL document.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining a PL/SQL Document Attribute

Define an item attribute of type PL/SQL document in the Vacation Proposal item type that you created in the Creating a Workflow Process practice.

Defining a Notification with a PL/SQL Document

Define a notification that includes the PL/SQL document in its message.

Solution – Using a PL/SQL Document Attribute

Defining a PL/SQL Document Attribute

1. Start the Oracle Workflow Builder.

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

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

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

5. Define the following properties for the item attribute:
   - Internal Name: VACATION_SCHEDULE_DOC
   - Display Name: Vacation Schedule Document
   - Type: Document
   - Frame Target: Full Window
   - Default Value: PLSQL:WFVAC*XX*.VACATION_SCHEDULED/&REQUESTOR

   - Click OK.

   The PL/SQL document generated by the WFVAC*XX*.VACATION_SCHEDULED procedure displays the scheduled vacation for a vacation proposal requestor as stored in your WFVAC*XX*_VACATION_SCHEDULE table.

   Note: The WFVAC*XX*_VACATION_SCHEDULE table is provided in a sample table creation script that you ran in the Defining a Function Activity Practice, and the WFVAC*XX*.VACATION_SCHEDULED procedure is provided in the sample WFVAC*XX* package that you loaded in the Defining a Function Activity practice. If you have not already copied, edited, and run the sample table creation, package specification, and package body scripts to create this table and load this package, follow the instructions in the Defining a Function Activity practice to do so now.

Defining a Notification with a PL/SQL Document

6. From the Edit menu, select New > Message.

7. Define a Vacation Schedule message that informs the requestor of his or her scheduled vacation dates using the PL/SQL document. Define the following properties for the message:
   - Internal Name: VACATION_SCHEDULE
   - Display Name: Vacation Schedule

   - To embed the contents of the PL/SQL document within the message, include the message attribute token &VACATION_SCHEDULE_DOC in the message body. In the case of this message, the message attribute token forms the entire message body.

- Text Body: &VACATION_SCHEDULE_DOC

- Click OK.

8. Drag and drop the Requestor item attribute, the Vacation Schedule Document item attribute, and any other necessary item attributes onto the Vacation Schedule message to create the corresponding message attributes with those item attributes as their default values.

   Note: The &REQUESTOR portion of the default value for the PL/SQL document message attribute will be token replaced with the run-time value of the REQUESTOR message attribute. Even if the REQUESTOR attribute is not used in the message body, it must still be defined as a message attribute to enable the token substitution in the PL/SQL document message attribute.

9. From the Edit menu, select New > Notification.

10. Define the following properties for the notification activity:
    - Internal Name: VACATION_SCHEDULE
    - Display Name: Vacation Schedule
    - Icon: NTF_INFO.ICO
    - Message: Vacation Schedule

- Click OK.

11. Open the process diagram window for the Vacation Proposal process.

12. Delete the transition between the Update Vacation Schedule node and the Raise Vacation Scheduled Event node.

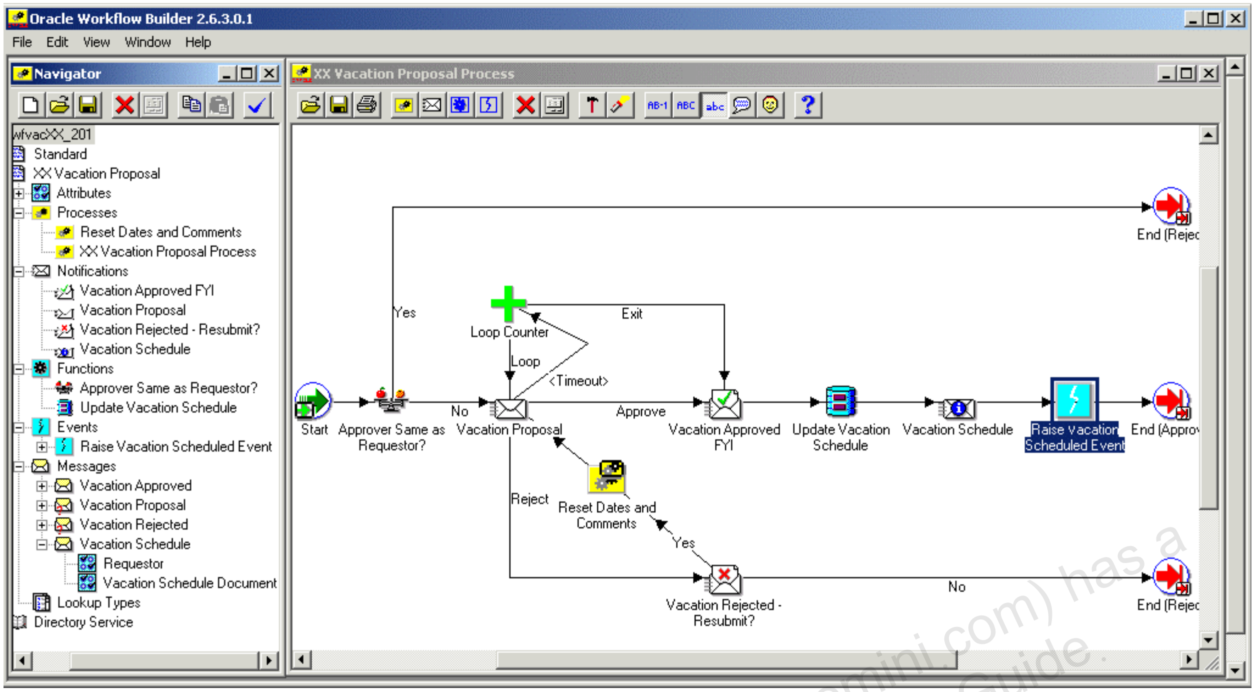
13. Drag and drop the Vacation Schedule notification into the process diagram, positioning it between the Update Vacation Schedule node and the Raise Vacation Scheduled Event node.

14. Draw transitions from the Update Vacation Schedule node to the Vacation Schedule node and from the Vacation Schedule node to the Raise Vacation Scheduled Event node.

15. Double-click the Vacation Schedule node and select 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.

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



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

19. Close the data store.

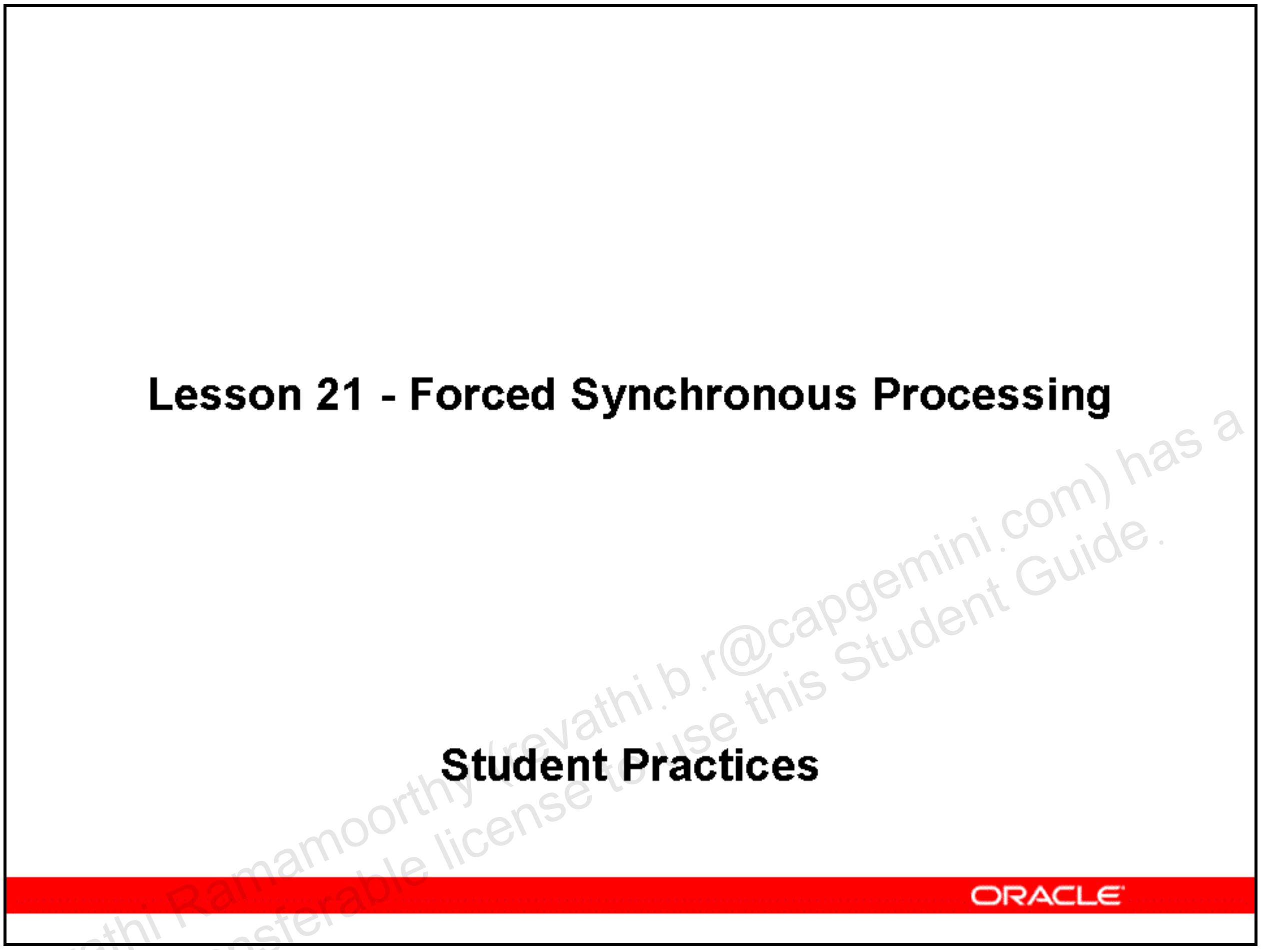
20. Use a Web browser to connect to a Workflow administrator responsibility with the URL provided by the instructor. Log in as a user with workflow administrator privileges.

21. Use the Developer Studio to launch your workflow process and test your work. You can use the Worklist Web pages to view the notifications sent by the process and use the Status Monitor Web pages to review the status of the process.

- Run the process and approve the vacation proposal.

- Use the Worklist Web pages to view the Vacation Schedule notification sent to the requestor and review the vacation schedule document. The PL/SQL document is generated when the notification is viewed through the Notification Details Web page.

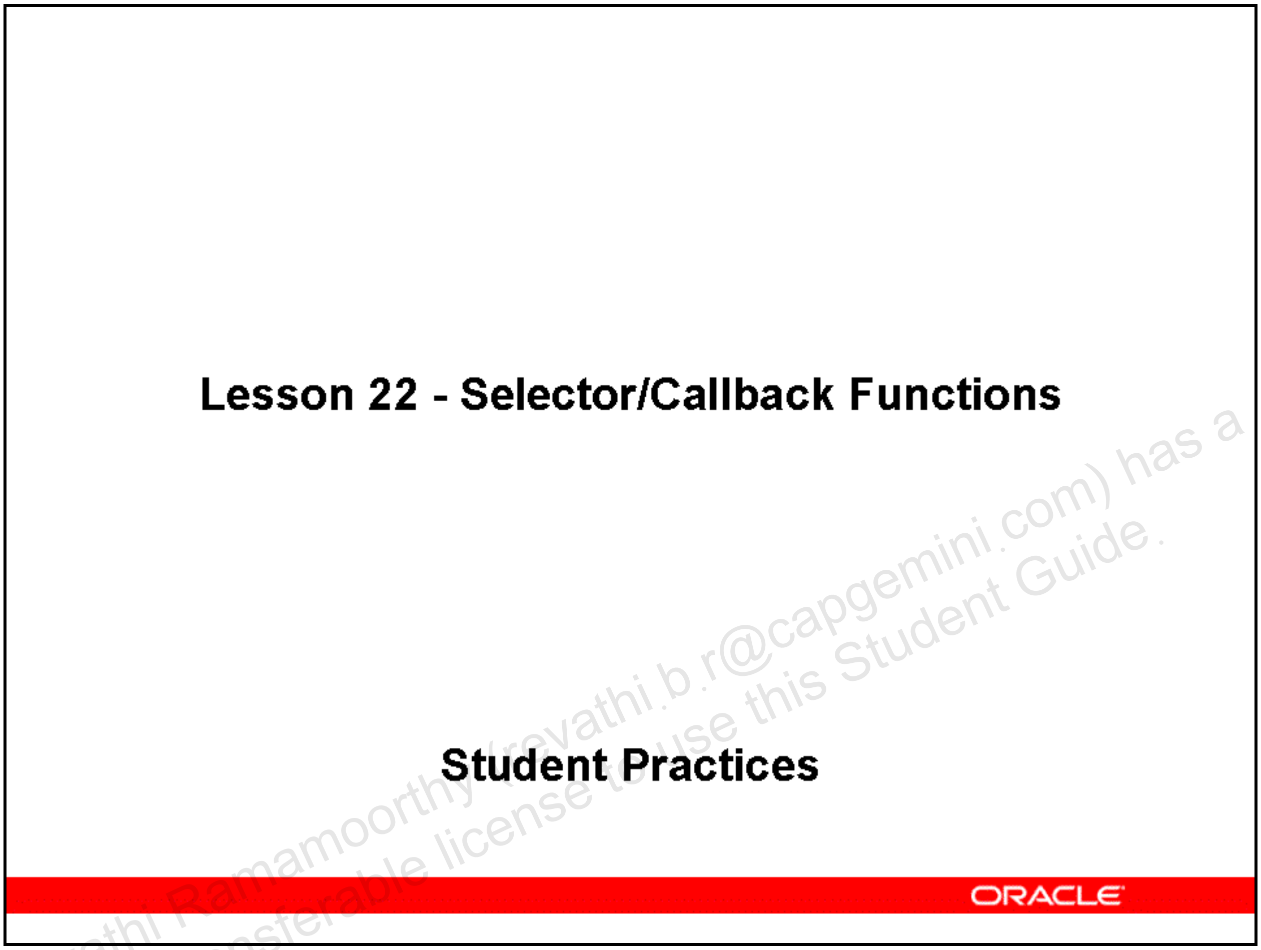
Lesson 21 - Forced Synchronous Processing



Lesson 21 - Forced Synchronous Processing

Student Practices

There are no practices or demonstrations in this lesson.



Lesson 22 - Selector/Callback Functions

Student Practices

Practice - Defining a Selector Function

Overview

In this practice, you will define a selector function to determine which process to run in the Vacation Proposal item type that you created in the Creating a Workflow Process practice. For this practice, you will use a predefined PL/SQL procedure as your selector function.

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 that 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 that you define exactly as shown in the instructions. Otherwise, you must modify the sample code to reference the object names that you define.

Assumptions

- You must have access to an Oracle E-Business Suite Vision database, or a comparable training or test instance at your site on which to complete this practice.
- The instructor will provide you with the connect string for the class database and the user name and password of the Oracle Workflow database account.
- The instructor will provide you with the user name and password of a user with workflow administrator privileges. The workflow administrator is defined on the Workflow Configuration page.
- The instructor will provide you with the URL for the login page and the name of a Workflow administrator responsibility. The user name that 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. These user names should have Workflow administrator and user responsibilities assigned to them.

Tasks

Defining an Additional Process

Define a new process within the Vacation Proposal item type that you created in the Creating a Workflow Process practice.

Defining a Selector Function for an Item Type

Define a selector function that determines which process to run in the Vacation Proposal item type.

Solution – Defining a Selector Function

Defining an Additional Process

1. Start the Oracle Workflow Builder.

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

3. In the navigator tree, select and expand your WFVAC*XX* item type.

4. In the navigator tree, drag and drop the Vacation Proposal process onto the WFVAC*XX* item type to create a copy of the process. The property pages for the copied process open automatically so that you can enter new internal and display names for the new process.
   - Internal Name: WFVAC*XX*_ALTERNATE_PROCESS
   - Display Name: *XX* Alternate Vacation Proposal

5. Open the process diagram window for the Alternate Vacation Proposal process.

6. Delete the Loop Counter node. The transitions to and from the node are automatically deleted as well.

7. Create a timeout transition from the Vacation Proposal notification back to itself. To do so, right-click the Vacation Proposal node and hold down the right mouse button. Drag the cursor away from the notification and then back to the notification, and then release the right mouse button. Select <Timeout> from the transition results menu.

Defining a Selector Function for an Item Type

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

9. Open the property pages for the item type. In the Selector field, enter WFVAC*XX*.SELECTOR as the selector function for the item type.

   The WFVAC*XX*.SELECTOR process determines which process to run according to the following logic:

   - If the first four characters of the item key are CNTR, run the Vacation Proposal process which contains the Loop Counter timeout implementation.

   - If the first four characters of the item key are SELF, run the *XX* Alternate Vacation Proposal process which contains the self-looping timeout implementation.

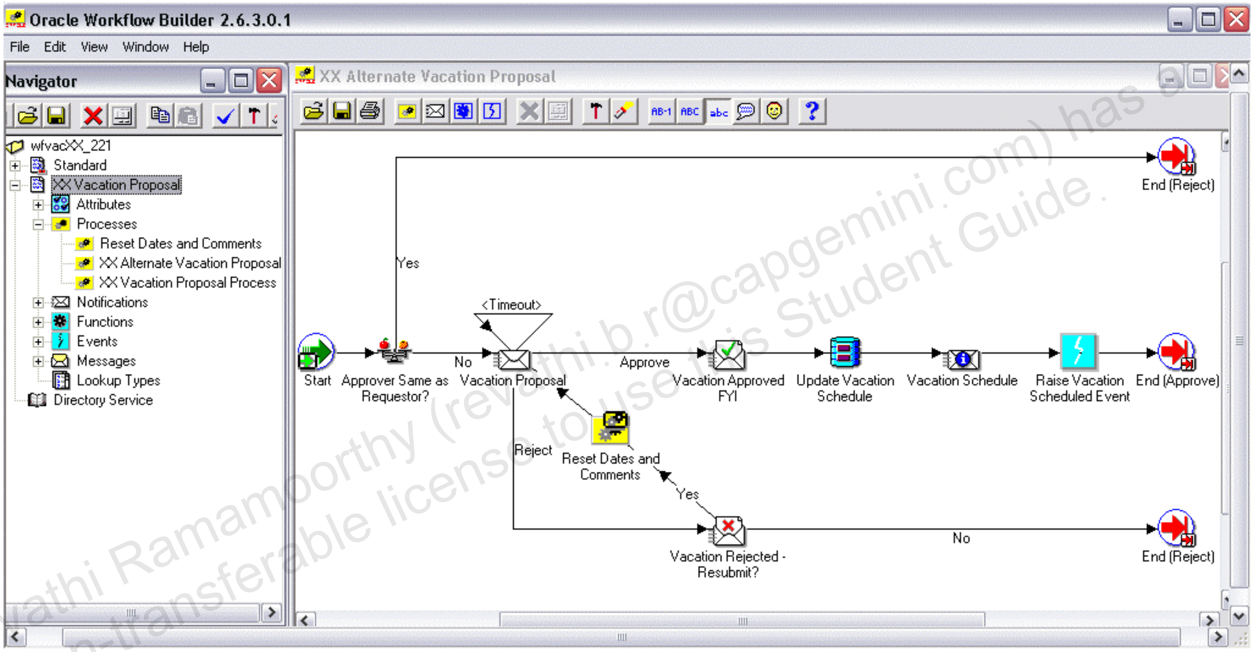
   - If the item key begins neither with CNTR nor with SELF, raise an error.

   Note: This example is contrived for class purposes only and does not reflect a practical implementation of selector function logic. The selector function is generally expected to use the item key as the primary key to retrieve supporting application data. That application data would then be used to determine which process is appropriate to run.

Note: The WFVAC*XX*.SELECTOR procedure is provided in the sample WFVAC*XX* package that you loaded in the Defining a Function Activity practice. If you have not already copied, edited, and run the sample package specification and body scripts to load this package, follow the instructions in the Defining a Function Activity practice to do so now.

10. In the Navigator window, click the Verify button to verify your workflow.

11. From the File menu, select Save to save your work to your workflow definition file.



12. From the File menu, select Save As and save your item type to the class database, using the database user name, password, and connect string provided by the instructor.

13. Close the data store.

14. Use SQL*Plus to launch your workflow process and test your work. First, launch a work item with an item key that begins with CNTR. Do not specify the process to run within the item type, so that the Workflow Engine will run the selector function for the item type to determine which process to run. The Workflow Engine will run the selector function only if the process parameter is not passed in the call to create the new work item.

    Next, launch a work item with an item key that begins with SELF. Again, do not specify the process to run within the item type.

    Finally, launch a work item with an item key that begins neither with CNTR nor with SELF. For example, use an item key that begins with TEST. Again, do not specify the process to run within the item type.

Note: The selector function provided in the sample solution package raises an error if the item key begins neither with CNTR nor with SELF, so the expected behavior when you attempt to launch the process in this case is an `ORA-20002: Invalid itemkey` error message.

You can use the sample work item launch script to launch these three work items all at once.

- Copy and edit the sample work item launch script named wfslctxx.sql. Open a copy of the sample file and replace all instances of XX with your own terminal number. Also, replace all instances of <requestor_username> with the user name of the requestor, all instances of <approver_username> with the user name of the approver, all instances of <from_date> with the from date, and all instances of <to_date> with the to date. Then save the file and rename it by replacing *xx* with your terminal number.

- Log in to SQL*Plus using the database user name, password, and connect string provided by the instructor. Run the work item launch script by entering the following command from the directory where the script is located: `@wfslct`*`xx`*

  Alternatively, you can run the script from the default prompt if you include the directory path for the script in the command. For example, if your script is located in the E:\Labs folder, then enter the following command:

  ```
  @E:\Labs\wfslctxx
  ```

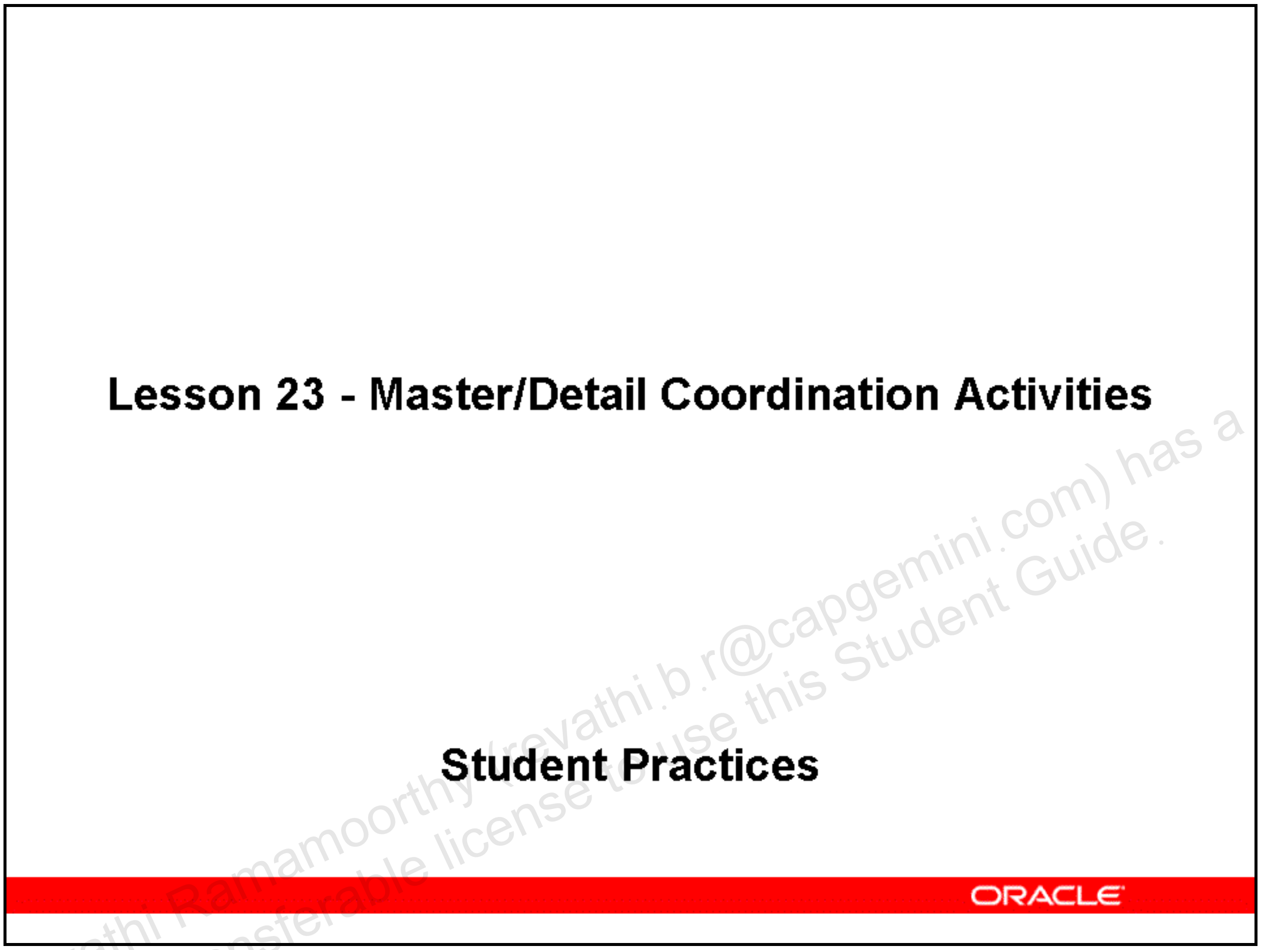
15. You can use the Worklist Web pages to view the notifications sent by the processes and use the Status Monitor Web pages to review the status of the processes. For each process you launched, verify that the expected process was run. The process that was run appears in the following administrator Status Monitor pages:

- Activity History, in the Activity and Parent Activity columns

- Status Diagram, in the process title and the diagram itself

16. For extra practice, you can optionally create another process within your item type that implements timeout processing in which you transition to a reminder notification when the Vacation Proposal notification times out. Add a timeout loop for the reminder notification as well. Then modify your selector function to run this new process when the upper case value of the first four characters of the item key is RMND, and test your changes.

Lesson 23 - Master/Detail Coordination Activities

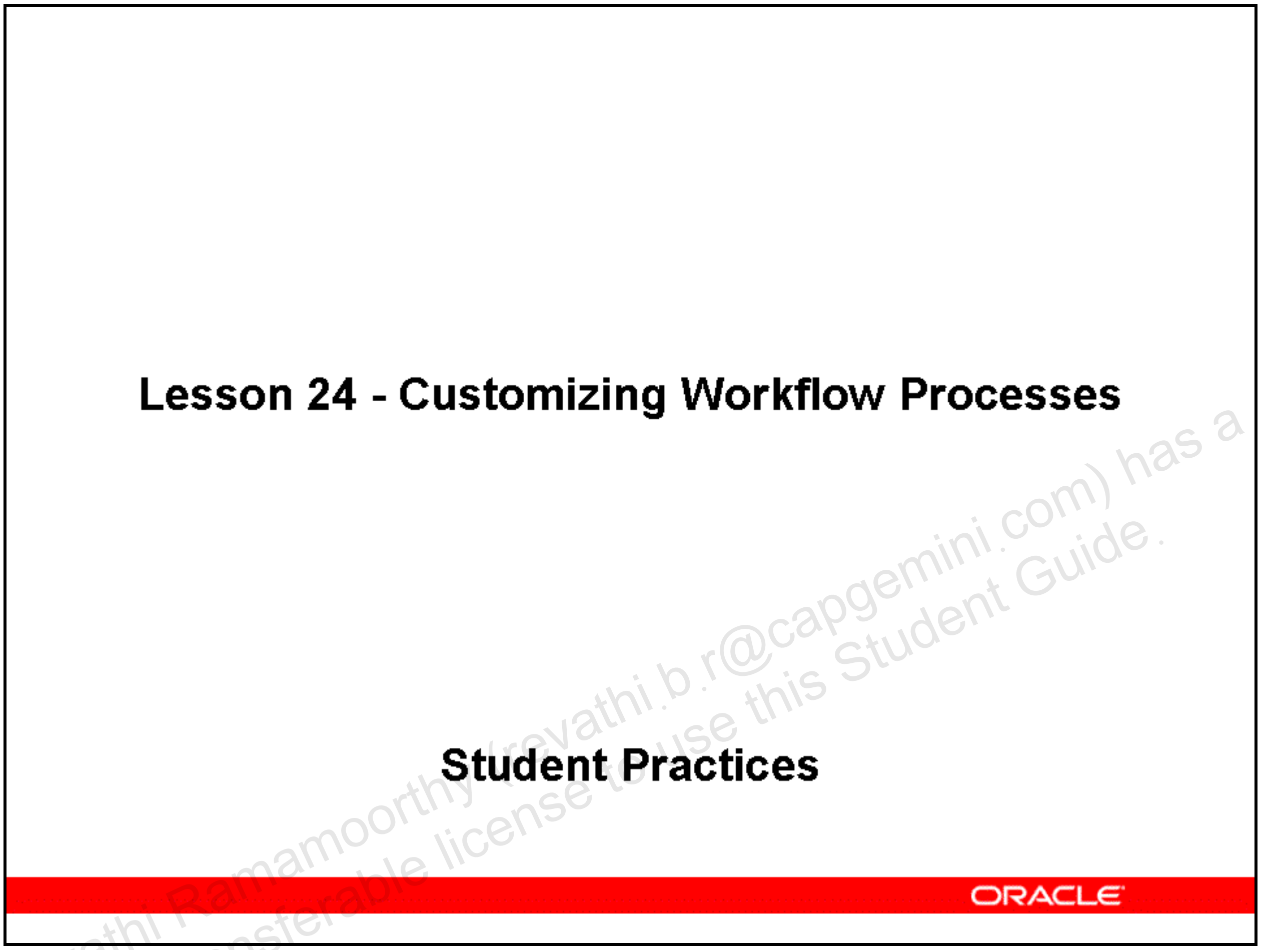


Lesson 23 - Master/Detail Coordination Activities

Student Practices

There are no practices or demonstrations in this lesson.

Lesson 24 - Customizing Workflow Processes

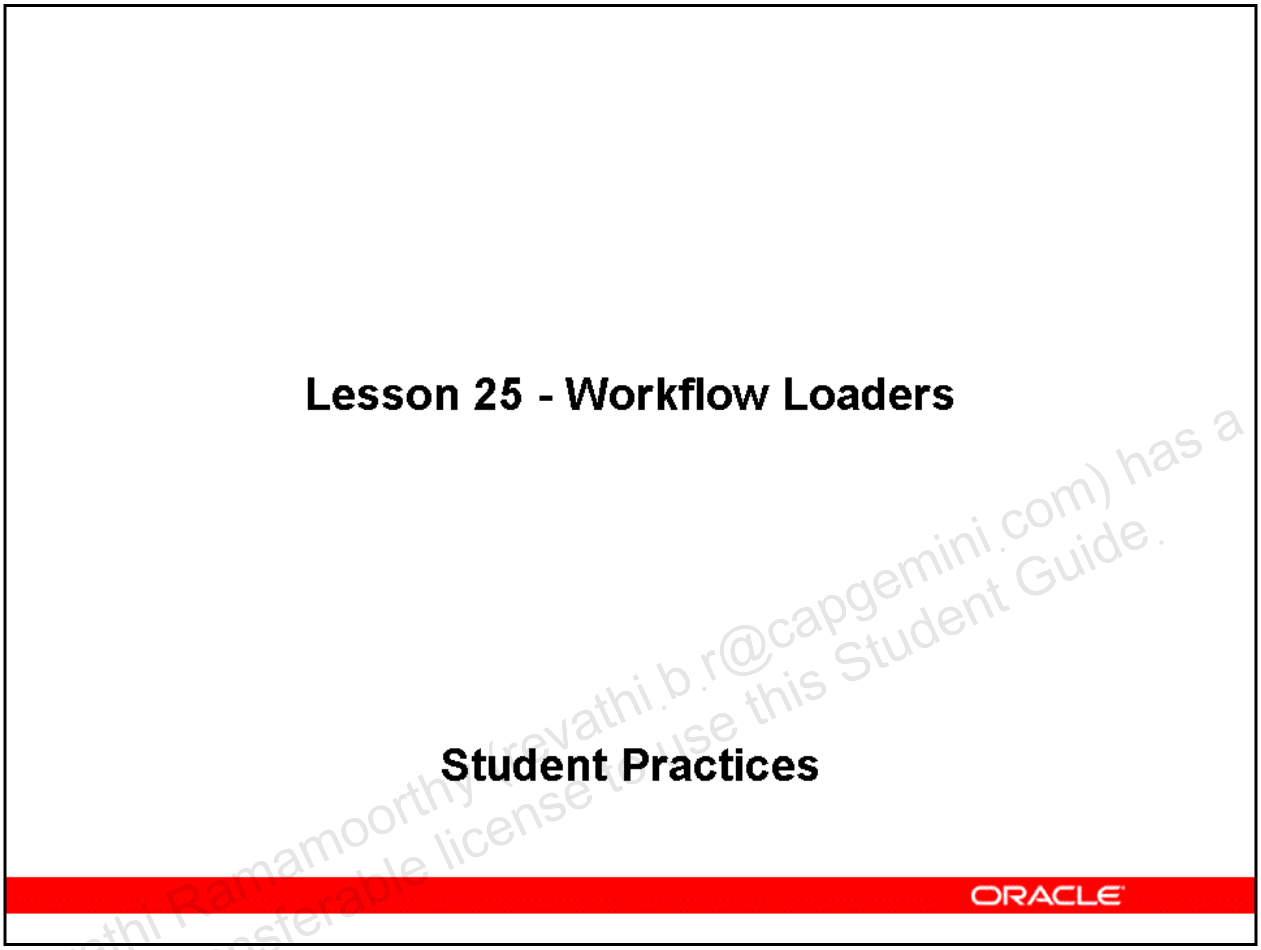


Lesson 24 - Customizing Workflow Processes

Student Practices

There are no practices or demonstrations in this lesson.

Lesson 25 - Workflow Loaders

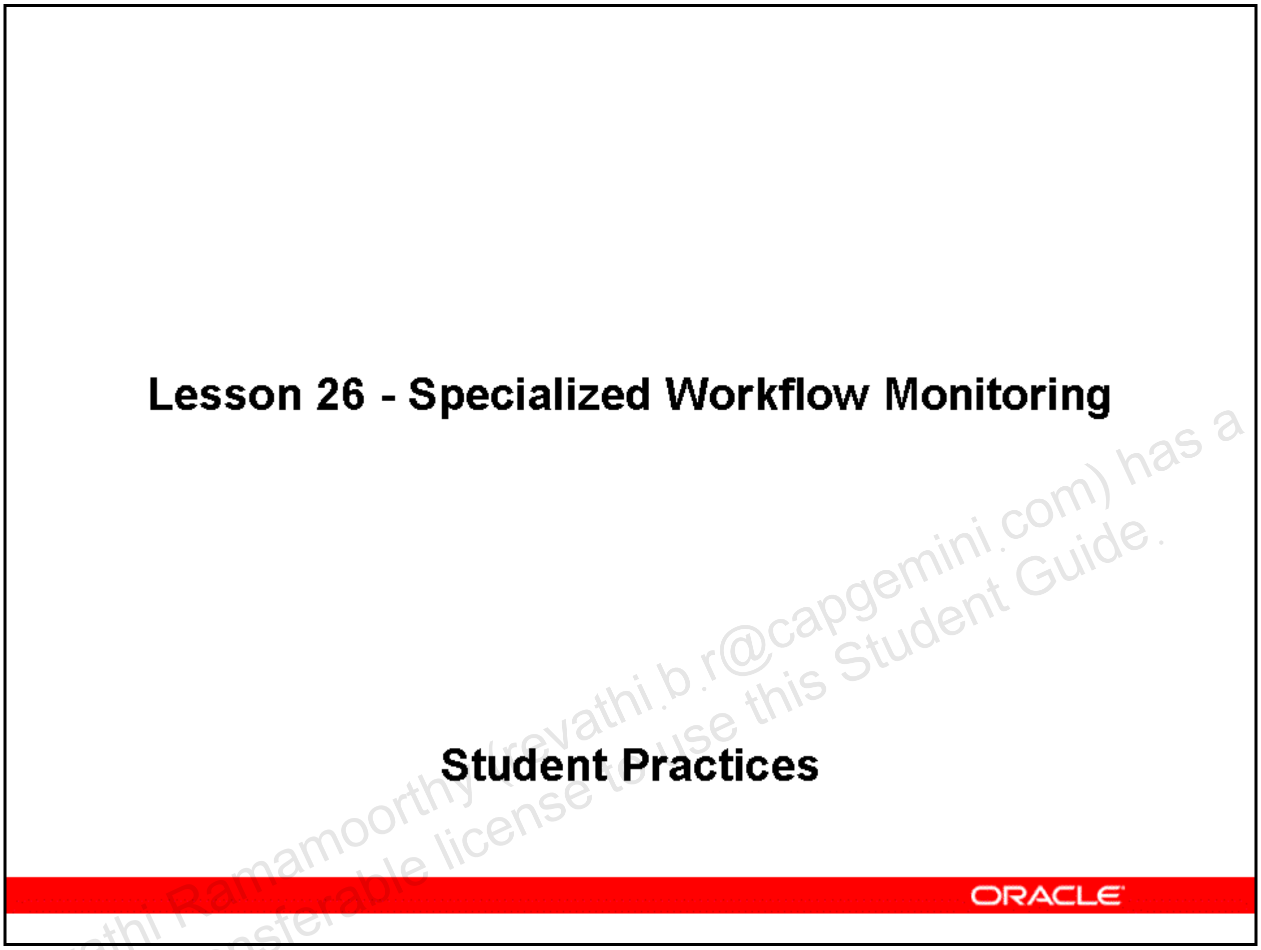


Lesson 25 - Workflow Loaders

Student Practices

There are no practices or demonstrations in this lesson.

Lesson 26 - Specialized Workflow Monitoring



Lesson 26 - Specialized Workflow Monitoring

Student Practices

Guided Demonstration - Setting Up Specialized Workflow Monitoring

Granting Restricted Access to Workflow Monitoring Data

1. Use a Web browser to connect to the Functional Developer responsibility.

2. Enter Workflow Items in the Name field and click Go.

3. Click the Workflow Items name link to access the Workflow Items Object page.

4. Click Create Instance Set in the Object Instance Sets tab.

5. Enter the following values for the instance set:
   - Name: Workflow Vacation Instance Set
   - Code: WFVACSET
   - Description: Object Instance Set for Workflow Vacation
   - Predicate: &TABLE_ALIAS.ITEM_TYPE = &GRANT_ALIAS.PARAMETER1

   - Click Apply.

6. Select the Grants tab and click Update.

7. Select Create Grant in the Grants tab to access the Define Grant page.

8. Enter the following values to create a grant:
   - Name: Workflow Vacation Grant
   - Description: Grant Workflow Vacation Instance Set
   - Grantee Type: Specific User
   - Grantee: Douglas, Carl (email: cdouglas@vision.com)
   - Responsibility: Workflow Administrator Web Applications
   - Data Security Object: WORKFLOW_ITEMS

   - Click Next
     - Data Context Type: Instance Set
     - Instance Set: Workflow Vacation Instance Set

   - Click Next
     - Parameter 1: WFVACXX
     - Permission Set: Business workflow item permission set (Code: WF_ADMIN_ITEM_PSET)

   - Click Next to review your grant and then click Finish.

Assigning Privileges for Administrative Actions in the Status Monitor

9. Use a Web browser to connect to the User Management responsibility.

10. Click the Users link to access the User Maintenance page.

11. Search for the user Carl Douglas with the email address cdouglas@vision.com and click Go.

12. Click the Update icon for Carl Douglas to access the Update User page.

13. Select Assign Roles.

14. Select Code in the Search By field and enter WF_ADMIN_ROLE% as the code to search by.

15. Select Workflow Admin Role from the search results and click Select to return to the Update User page.

16. Enter "Assign workflow admin role with privileges for all administrative actions within the Status Monitor to specialized workflow administrator" as the justification.

17. Click Save.

Associating the Workflow Administrator Web Applications Responsibility with a User

18. Use a Web browser to connect to the System Administrator responsibility.

19. Select Security: User > Define to open the Users window.

20. Query CDOUGLAS as the user name by selecting View > Query by Example > Enter. Enter the user name CDOUGLAS and then execute the query by selecting View > Query by Example > Run to display the user details.

21. Click the New record icon from the application toolbar to add the Workflow Administrator Web Applications responsibility.

22. Save your record and close the Users window.

Launching the Vacation Proposal Workflow

23. Use a Web browser to connect to a Workflow administrator responsibility. Log in as a user with workflow administrator privileges.

24. Use the Developer Studio to launch the Vacation Proposal workflow process that you loaded in the Loading and Running a Workflow Process demonstration.

    Note: Do not log in as an approver to approve the vacation proposal, but leave the notification open, so that the workflow process will remain active and you can verify the CDOUGLAS user's privileges to perform administrative actions in the Status Monitor. After a workflow process is completed, you can no longer perform administrative actions for that process in the Status Monitor, even if you have privileges to perform such actions.

Verifying Specialized Workflow Monitoring Privileges

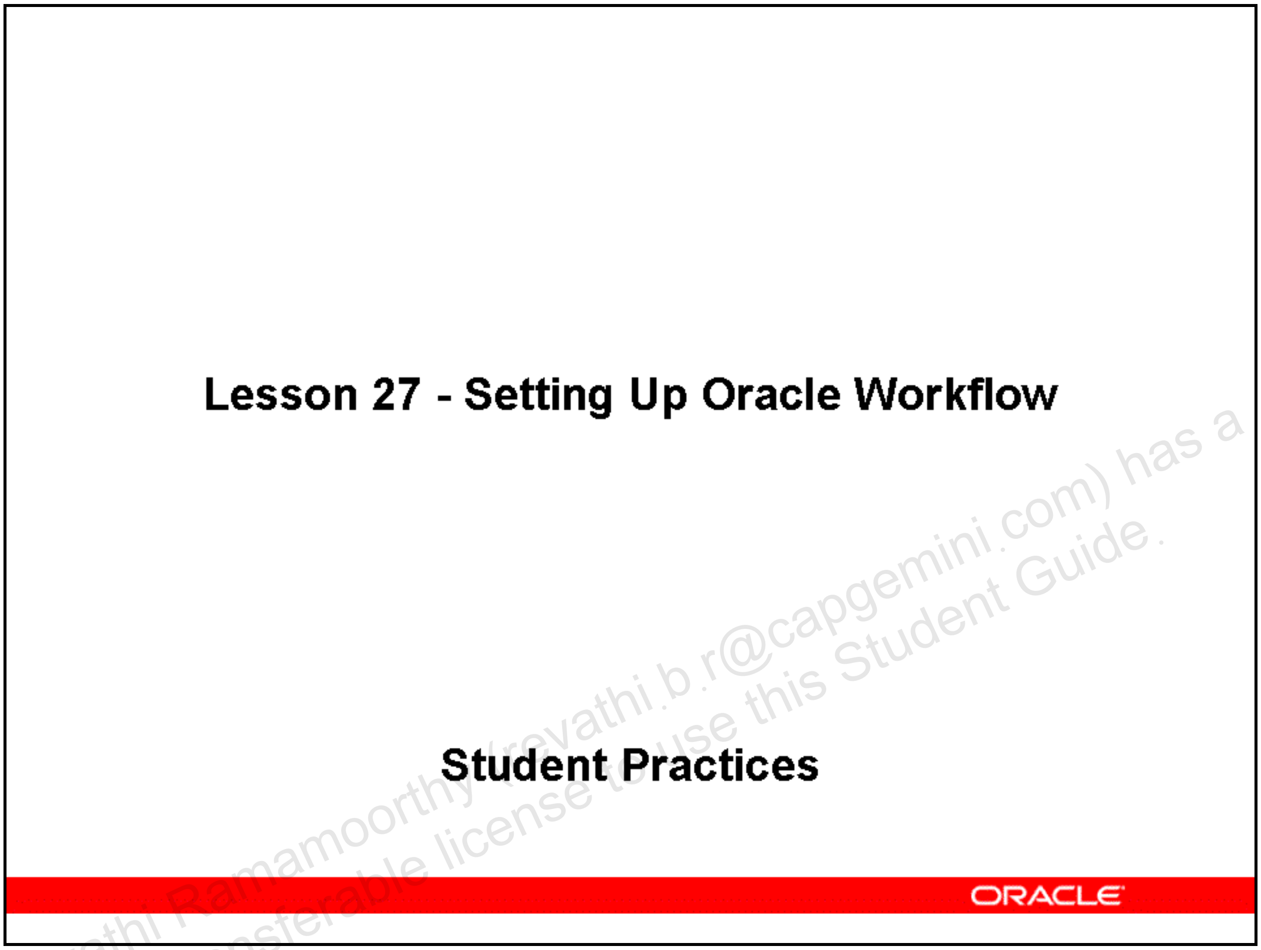
25. Log off and log in as the user CDOUGLAS.

26. Use a Web browser to connect to the Workflow Administrator Web Applications responsibility.

27. Use the Status Monitor Web pages to review the status of the Vacation Proposal process that you launched. The overall process status should still be Active.

28. Click the Action History button and verify that the following administrative action buttons are available in the Action History page:

    - Retry
    - Skip
    - Update Attributes
    - Rewind
    - Suspend Workflow
    - Cancel Workflow



Lesson 27 - Setting Up Oracle Workflow

Student Practices

Guided Demonstration - Scheduling Agent Listeners and Propagation

1. Log on to a Workflow administrator responsibility with Event Manager functionality. Log in as a user with workflow administrator privileges.

2. Click the Administration link, and ensure that the Business Event Local System: Status is set to Enabled on the Workflow Configuration page.

3. Ensure that the job_queue_processes database parameter is set to 10 or higher.

   - You can use the Oracle Workflow Manager component of Oracle Applications Manager to check the database parameters required for propagation.

     - Connect to a Workflow administrator responsibility as a user with workflow administrator privileges, and click the Workflow Manager link.

     - Review the information in the Related Database Parameters region of the Workflow System page.

   - You can also use the following command to check the parameter value in SQL*Plus:

     ```
     select name, value from v$parameter
     where name='job_queue_processes';
     ```

   - If necessary, you can increase the job_queue_processes value dynamically, without needing to restart the database. Connect to SQL*Plus as the SYS user and execute the following commands:

     ```
     alter system set job_queue_processes = 10;

     exit
     ```

   - Note that this setting is necessary because propagation schedules are handled by job queue processes. If no job queue processes are available in the database because they are all being used for other types of jobs, then the propagation will not take place. So the job_queue_processes parameter needs to be set sufficiently high to ensure that a process will be available to handle propagation.

4. Schedule propagation to run once a minute for the local WF_JMS_OUT agent and the Local destination. Connect to SQL*Plus and execute the following command:

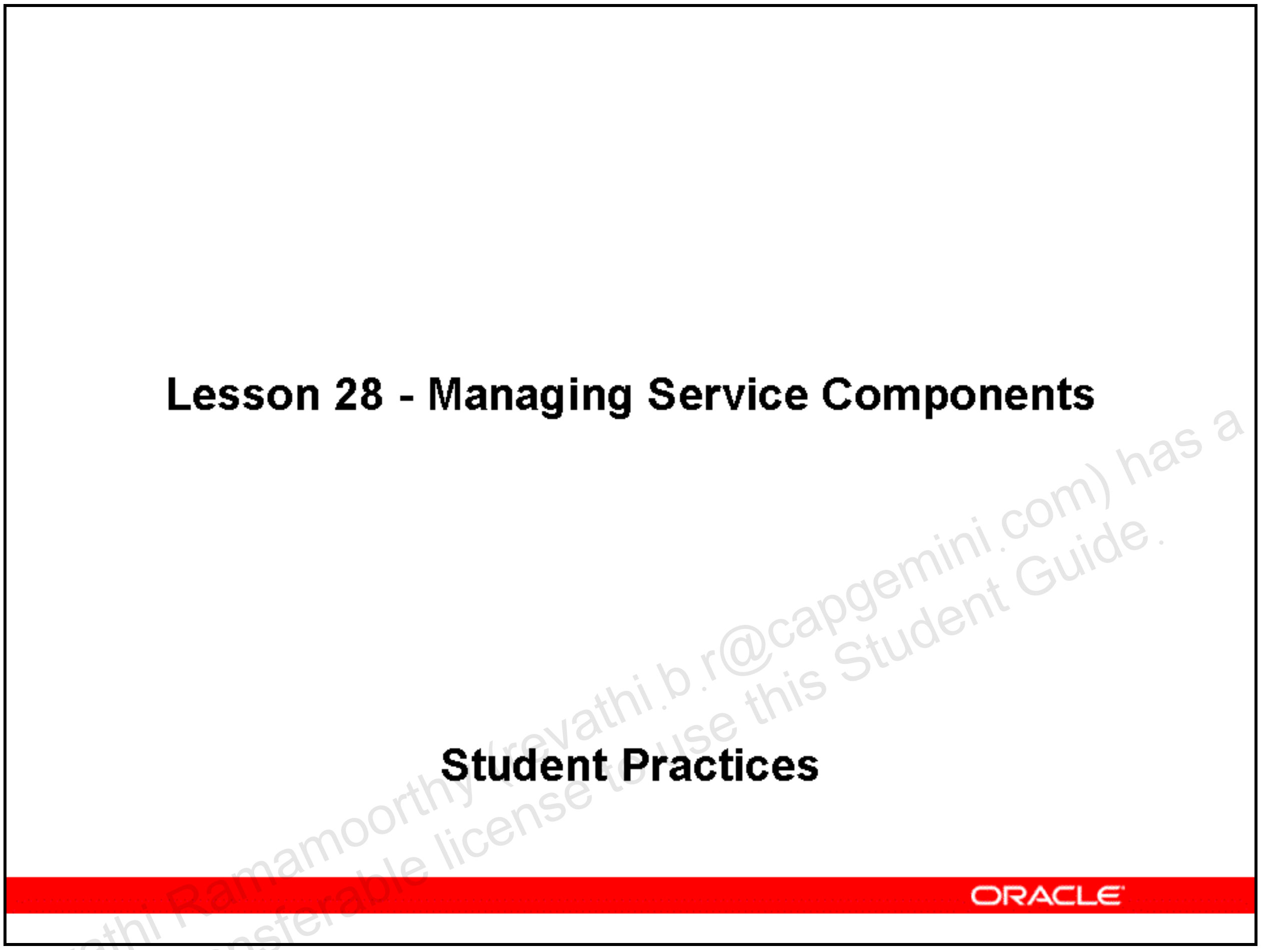
   ```
   Exec DBMS_AQADM.Schedule_Propagation('APPLSYS.WF_JMS_OUT',
     NULL, SYSDATE, 30, 'SYSDATE + 1/1440', NULL);
   ```

5. Run an agent listener for the local WF_JMS_IN agent.

- You can use the Oracle Workflow Manager component of Oracle Applications Manager to configure and run an agent listener service component for WF_JMS_IN.
  - Connect to a Workflow administrator responsibility as a user with workflow administrator privileges, and click the Workflow Manager link.
  - Click the Agent Listeners status icon on the Workflow System page.
  - Click the Create button.
  - Select Workflow Agent Listener and click Continue.
  - Enter "WF Inbound JMS Listener" in the Name field, select Automatic in the Startup Mode field, and select WF_JMS_IN in the Inbound Agent field. Then click Next.
  - Select the Workflow Agent Listener Service in the Container field and click Next.
  - Click Next.
  - Click Finish.
  - The container should automatically start the agent listener within a few minutes. Check back until the status of the agent listener appears as Running on the Service Components page. The agent listener should now run repeatedly.
- You can also use the WF_EVENT.Listen API to run an agent listener manually from SQL*Plus. Connect to SQL*Plus as the APPS user for Oracle E-Business Suite, and use the following command:

```
exec WF_EVENT.Listen('WF_JMS_IN');
```

In this case, note that running the API once means running the listener only once. The listener will dequeue all messages currently enqueued on the agent's queue and then exit. To automatically resubmit the listener, configure an agent listener service component.



Lesson 28 - Managing Service Components

Student Practices

Guided Demonstration - Service Components

1. Access Oracle Applications Manager by connecting to a Workflow administrator responsibility as a user with workflow administrator privileges and clicking the Workflow Manager link.

2. On the Workflow System page, click the Service Components status icon and review the types and statuses of service components that are configured in this instance.

3. Select a service component and click View Log to view the log of the container to which this service component belongs. Close the log window when you are finished reviewing the log.

4. Select a service component and click View Details to review the configuration of that service component. Click OK.

5. Select a service component and click View Event History to review the events that have been scheduled to control the running of that service component.

6. Click Workflow in the locator links at the top of the page to navigate back to the Workflow System page.

7. Click the Agent Listeners status icon to navigate to the Service Components page with the list filtered to show only agent listener service components. Review statuses of the standard agent listeners provided by Oracle Workflow, such as Workflow Deferred Agent Listener, Workflow Error Agent Listener, and Workflow Inbound Notifications Agent Listener. You can also click View Details for these agent listeners to review their configurations.

8. Navigate back to the Workflow System page.

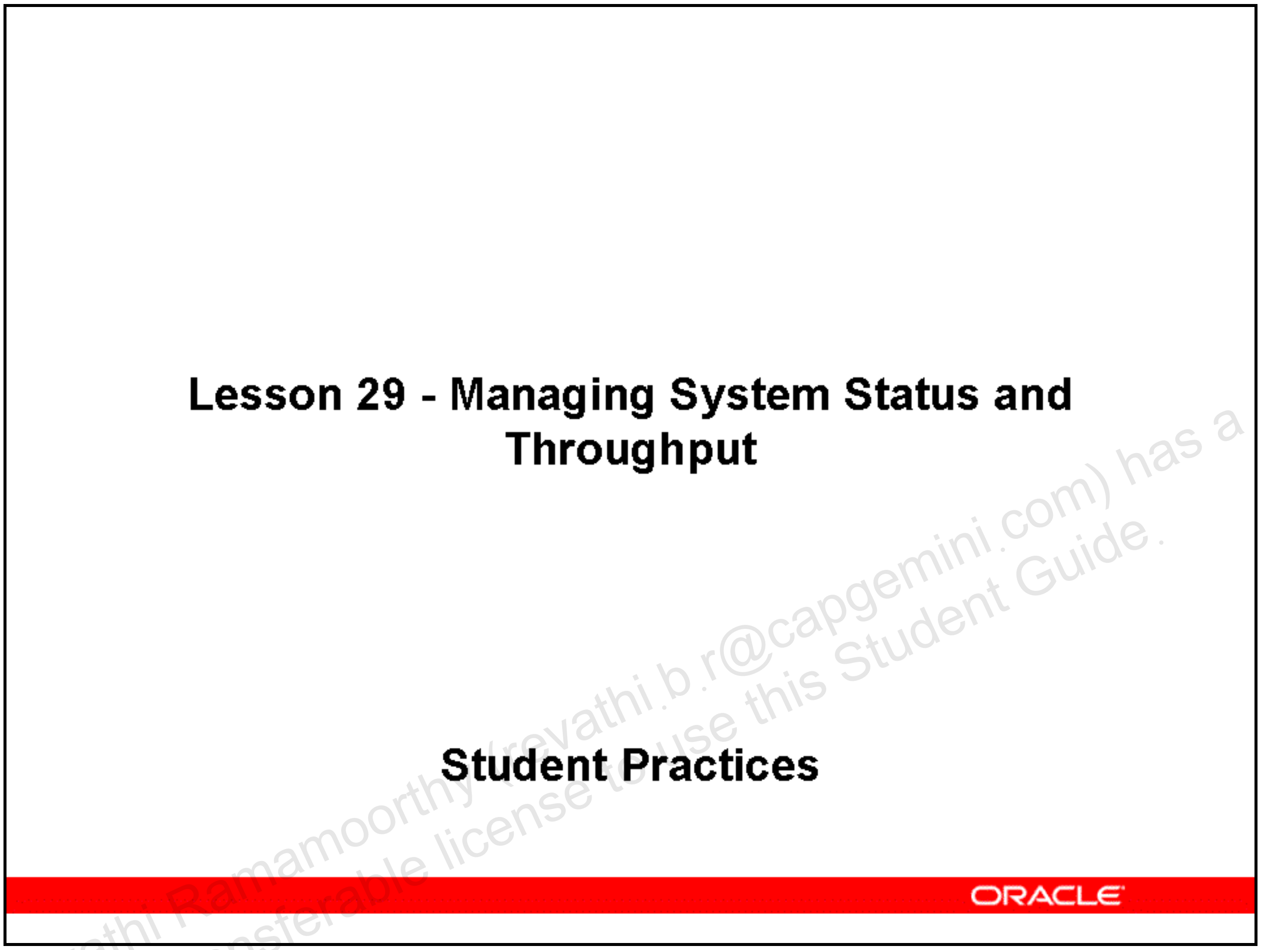
9. Click the Notification Mailers status icon to navigate to the Service Components page with the list filtered to show only notification mailer service components.

10. Review the status of the standard Workflow Notification Mailer service component provided by Oracle Workflow. Select this service component and click Edit to review the basic notification mailer configuration. Review the following parameters:
    - Outbound E-mail Account (SMTP): Server Name - The name of the outbound SMTP mail server.
    - Inbound E-mail Account (IMAP): Inbound Processing - Enable or disable inbound e-mail processing with this notification mailer.
    - Inbound E-mail Account (IMAP): Server Name - The name of the inbound IMAP mail server.
    - Inbound E-mail Account (IMAP): Username - The user name of the mail account that the notification mailer uses to receive e-mail messages.
    - Inbound E-mail Account (IMAP): Password - The password for the mail account that the notification mailer uses.

- Inbound E-mail Account (IMAP): Reply-To Address - The address of the e-mail account that receives incoming messages.

11. Navigate back to the Workflow System page.

12. In the Related Links > Throughput region, click the Notification Mailers link to review the mailer throughput graph.



Lesson 29 - Managing System Status and Throughput

Student Practices

Guided Demonstration - System Status and Throughput

1. Access Oracle Applications Manager by connecting to a Workflow administrator responsibility as a user with workflow administrator privileges and clicking the Workflow Manager link.

2. On the Workflow System page, click the Show link for the Work Items graph in the Workflow Metrics region.

3. Drill down to Active work items by selecting the Active bar in the graph.

4. Select a work item type and click View Details. If necessary, enter a value in the Filter Start Date Within Last _ Days field and click Go to display details that are not displayed by default.

5. Select a work item activity stage and click View Details.

6. Select a work item activity and click Suspend. Click OK on the confirmation pages.

7. Select Suspended Work Items from the View menu and click Go.

8. Select the same work item type and click View Details.

9. Select the same work item activity stage and click View Details.

10. Select the same work item activity and click Resume. Click OK on the confirmation pages.

11. Select Deferred Work Items from the View menu and click Go. Drill down to view the further details. Note that large numbers of deferred work items may indicate that you need to run more background engines.

12. Click Workflow in the locator links at the top of the page to navigate back to the Workflow System page.

13. Select Background Engines from the Submit Request For menu and click Go to submit the concurrent request.

    - Click Next and enter the following parameters:
      - Item Type: <select an item type display name>
      - Process Deferred: Yes
      - Process Timeout: No
      - Process Stuck: No

    - Click Next on each remaining page of the wizard until the Summary page, and then click Submit.

14. Navigate back to the Workflow System status page.

15. Select the Error bar in the Work Items graph. Drill down to view the further details.

16. Navigate back to the Workflow System status page.

17. In the Related Links > Throughput region, select Work Items.

18. In the Completed Work Items region, select a work item type and click View Details. If necessary, enter a value to in the Filter End Date Within Last _ Days field and click Go to display details that are not displayed by default.

19. Select a work item activity stage and click View Details.

20. Select Completed Work Items from the View menu and click Go to return to the Workflow Purge page.

21. Click the Purge button to submit a request for the Purge Obsolete Workflow Runtime Data concurrent program.

    - Click Next and select the item type to purge in the Item Type parameter. Leave the other parameters blank.

    - Click Next on each remaining page of the wizard until the Summary page, and then click Submit.

22. Navigate back to the Workflow Purge page and check for the item type that you purged in the Completed Work Items region to confirm that the eligible items of that type have been purged.

23. You can also use SQL*Plus to manually review and purge work items. In SQL*Plus, review the contents of a Workflow run-time table such as WF_ITEM_ACTIVITY STATUSES. For example:

```
SELECT ITEM_TYPE,ITEM_KEY, PROCESS_ACTIVITY, ACTIVITY_STATUS
FROM WF_ITEM_ACTIVITY_STATUSES
WHERE ITEM_TYPE LIKE ‘WF%’;
```

24. In SQL*Plus, run the WF_PURGE.Total API with appropriate parameters to purge another item type. For example:

```
Exec WF_PURGE.Total('<itemtype>');
```

25. In SQL*Plus, review the contents of the WF_ITEM_ACTIVITY STATUSES table again to show that the data has been purged.

26. Navigate back to the Workflow System page in Oracle Workflow Manager.

27. Select Control Queue Cleanup from the Submit Request For menu and click Go to submit the concurrent request.

- Click Next on each page of the wizard until the Summary page, and then click Submit. This program does not require any parameters.

28. Navigate back to the Workflow System page.

29. Click the Show link for the Agent Activity graph in the Workflow Metrics region.

30. Drill down by selecting a status bar in the graph.

31. Click an agent link in the Agent column to view details about the queue associated with that agent. Then click OK.

32. Select an agent with messages on it and select Search Agent Entry Details.

33. Select a time period in the Enqueue Date field in the Search Criteria region, and click Go.

34. Click the icon in the View XML column for a message. Note that the View XML icon is disabled if the event data within the event message is empty.

35. Navigate back to the Workflow System page.

36. In the Related Links > Configuration region, click the Queue Propagation link.

37. Select a queue schedule and click View Details.

38. Navigate back to the Workflow System page and review the overall status as shown by the status icons at the beginning of the page.

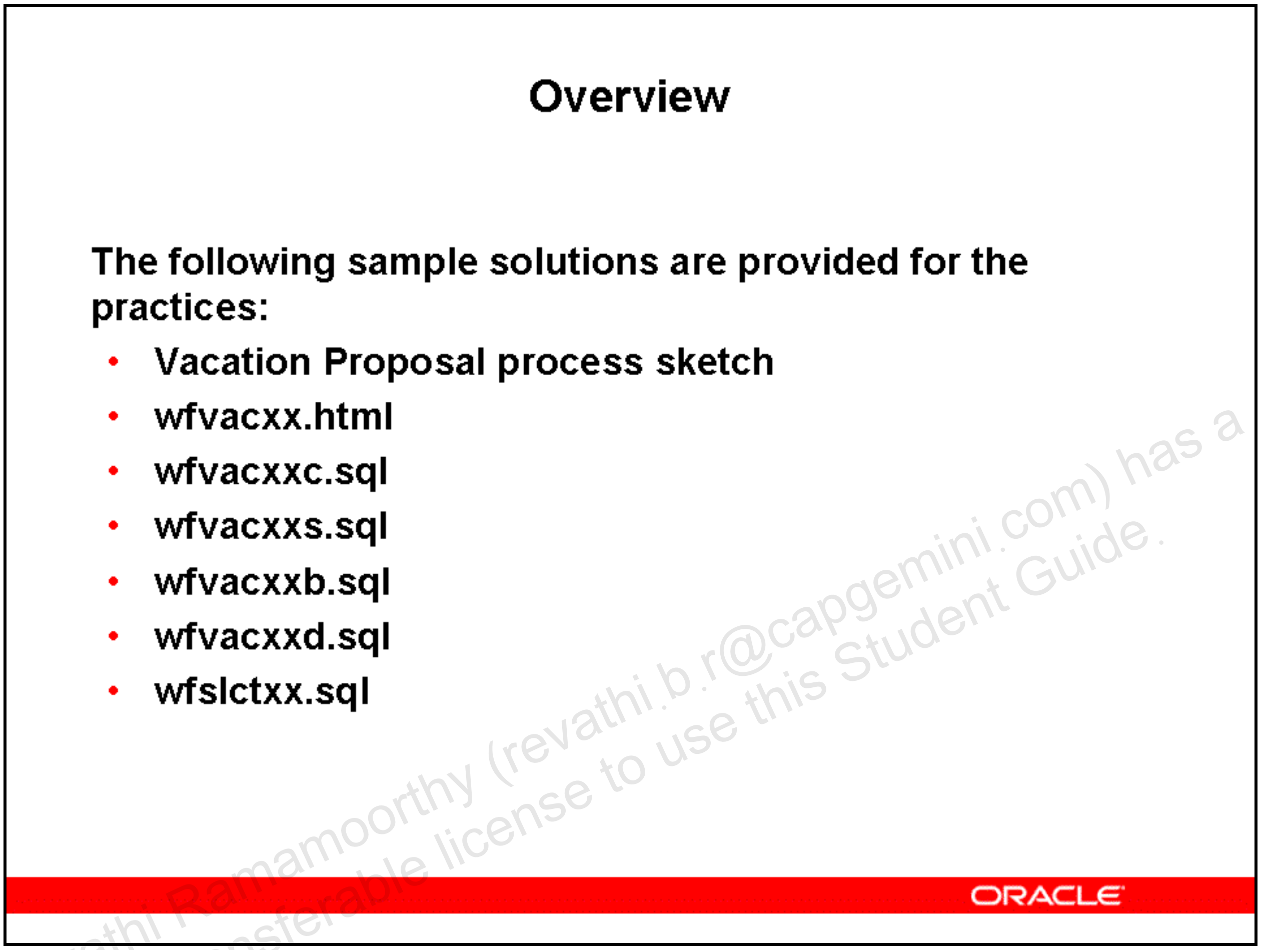
Sample Solutions

Chapter 31

Sample Solutions

Sample Solutions

Overview



Overview

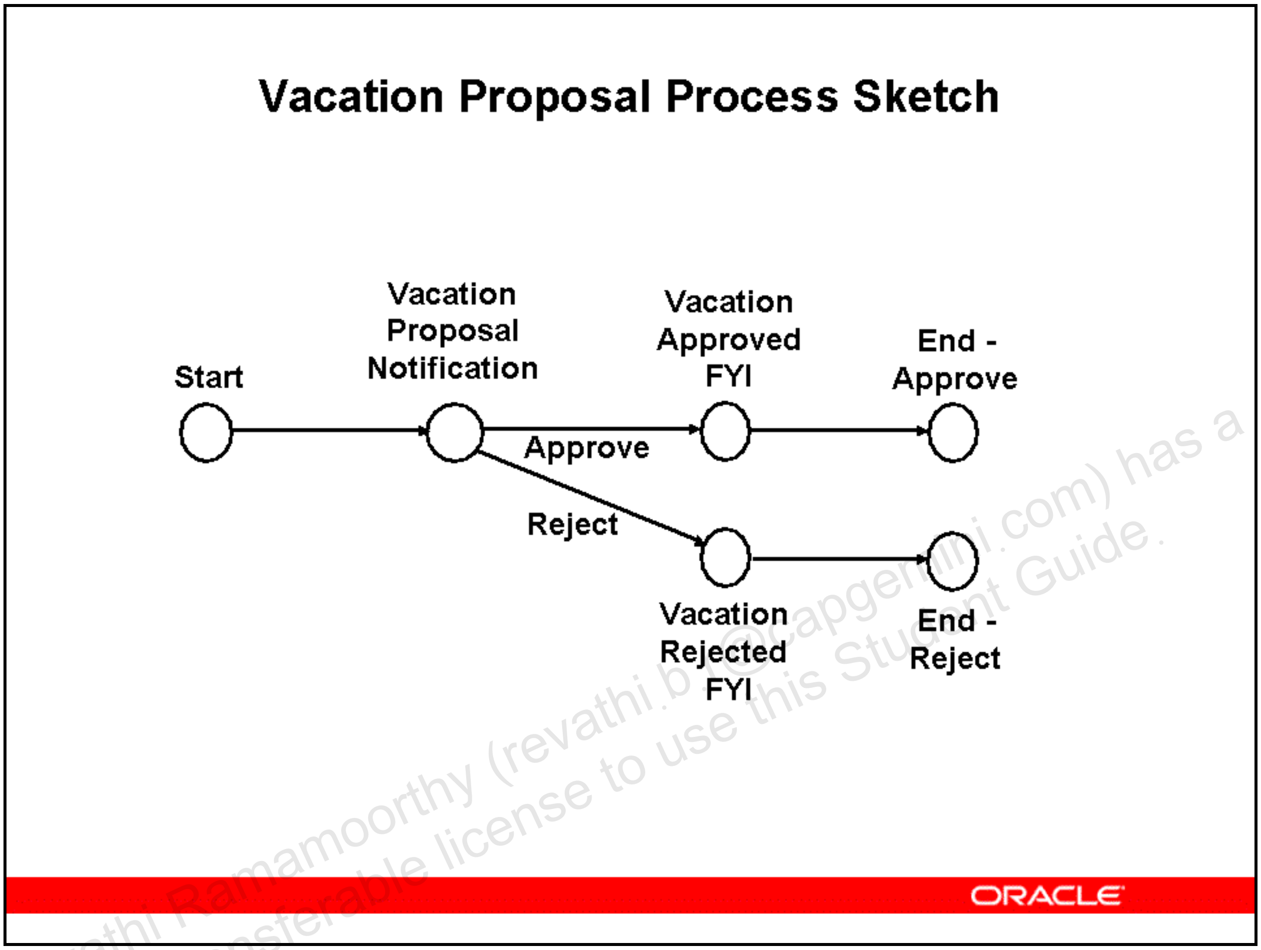
The following sample solutions are provided for the practices:

- Vacation Proposal process sketch
- wfvacxx.html
- wfvacxxc.sql
- wfvacxxs.sql
- wfvacxxb.sql
- wfvacxxd.sql
- wfslctxx.sql

Overview

- wfvacxx.html: HTML message body
- wfvacxxc.sql: Script to create the WFVACXX_VACATION_SCHEDULE table and its index
- wfvacxxs.sql: Script to create the WFVACXX package specification
- wfvacxxb.sql: Script to create the WFVACXX package body
- wfvacxxd.sql: Script to drop the WFVACXX_VACATION_SCHEDULE table, its index, and the WFVACXX package
- wfslctxx.sql: Script to launch work items without specifying the process to start

Vacation Proposal Process Sketch



Vacation Proposal Process Sketch

Start

Vacation Proposal Notification

Approve

Reject

Vacation Approved FYI

Vacation Rejected FYI

End - Approve

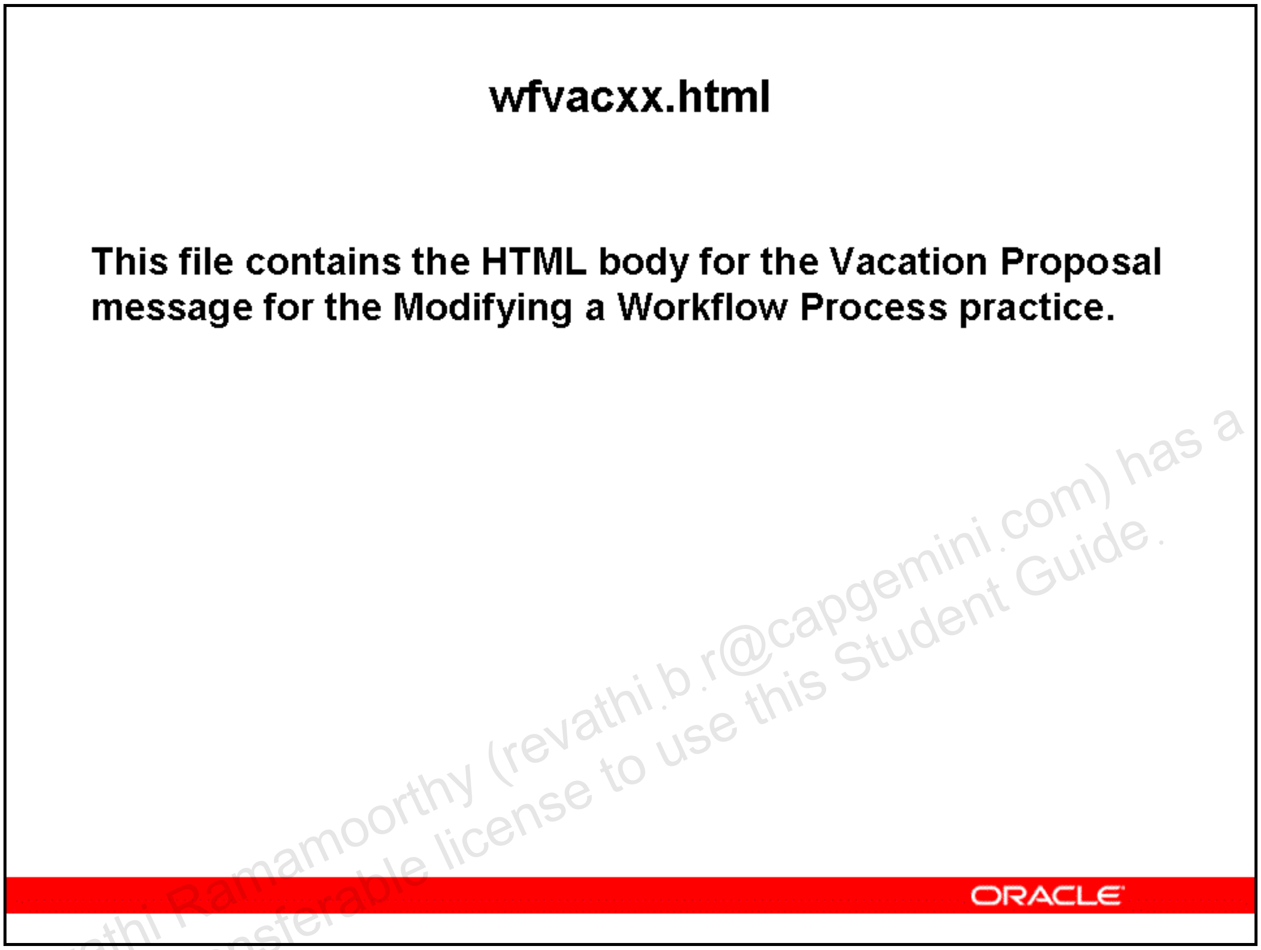
End - Reject

ORACLE

Vacation Proposal Process Sketch

This sketch shows a sample plan for the Vacation Proposal process as described in the Planning a Workflow Process practice.

wfvacxx.html



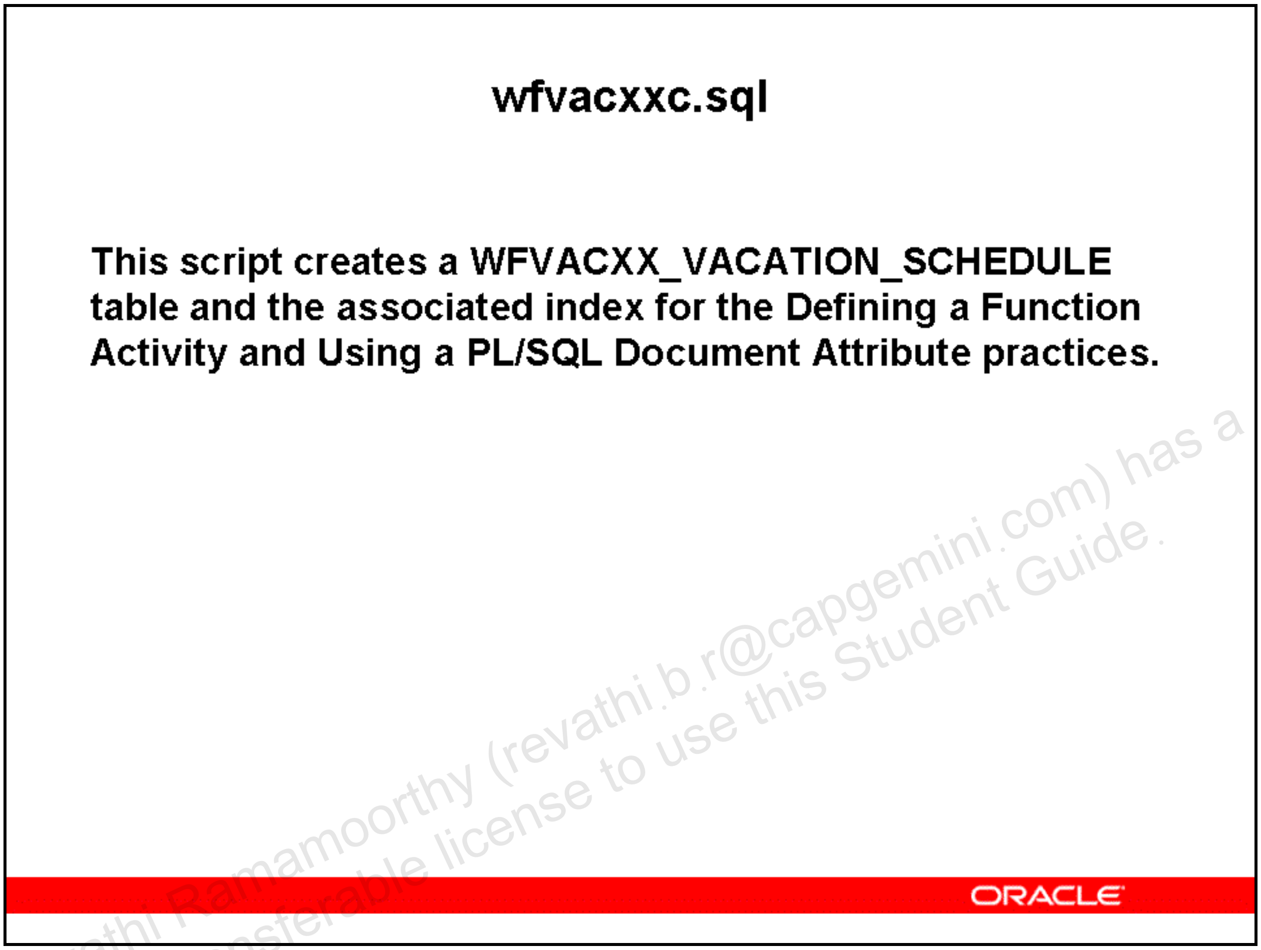
wfvacxx.html

This file contains the HTML body for the Vacation Proposal message for the Modifying a Workflow Process practice.

wfvacxx.html

```
<TABLE>
<TR><TD><img SRC="/OA_MEDIA/calendar.gif"></TD></TR>
<TR><TD>Vacation Proposal from <B>&REQUESTOR</B> for
     <B>&FROM_DATE</B> to <B>&TO_DATE</B></TD></TR>
</TABLE>
Please approve vacation as proposed or suggest alternate dates.
```

wfvacxxc.sql



wfvacxxc.sql

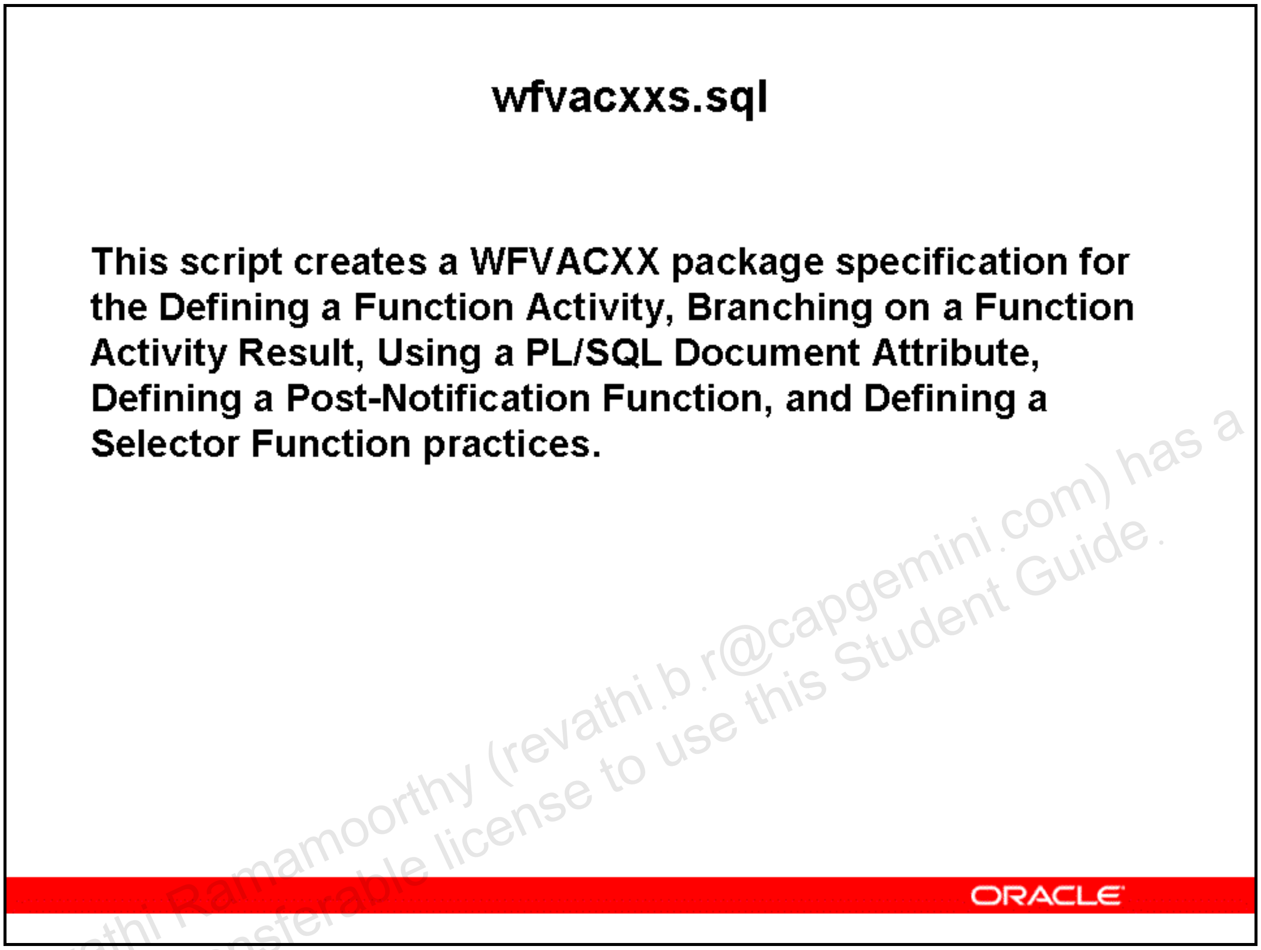
This script creates a WFVACXX_VACATION_SCHEDULE table and the associated index for the Defining a Function Activity and Using a PL/SQL Document Attribute practices.

wfvacxxc.sql

```
/*=======================================================================+
 |  Copyright (c) 1995 Oracle Corporation Redwood Shores, California, USA|
 |                          All rights reserved.                         |
 +=======================================================================+
 | FILENAME
 |   wfvacxxc.sql
 |
 | DESCRIPTION
 |  Create Workflow Vacation Schedule table and index
 | NOTES
 |   This file is a SAMPLE that should be modified with your own
 |   names before installing.  Names that include
 |   XX should be replaced with values for your implementation.
 |
 *=======================================================================*/
```

```
/* $Header$ */
whenever sqlerror continue;

drop   table WFVACXX_VACATION_SCHEDULE;
create table WFVACXX_VACATION_SCHEDULE
(   requestor_username      varchar2(30)      not null,
    approver_username       varchar2(30)      not null,
    from_date         date        not null,
    to_date           date        not null);
--
create index WFVACXX_VACATION_SCHEDULE_N1 on WFVACXX_VACATION_SCHEDULE
(requestor_username,approver_username);
--
commit;

--exit
```

wfvacxxs.sql



wfvacxxs.sql

This script creates a WFVACXX package specification for the Defining a Function Activity, Branching on a Function Activity Result, Using a PL/SQL Document Attribute, Defining a Post-Notification Function, and Defining a Selector Function practices.

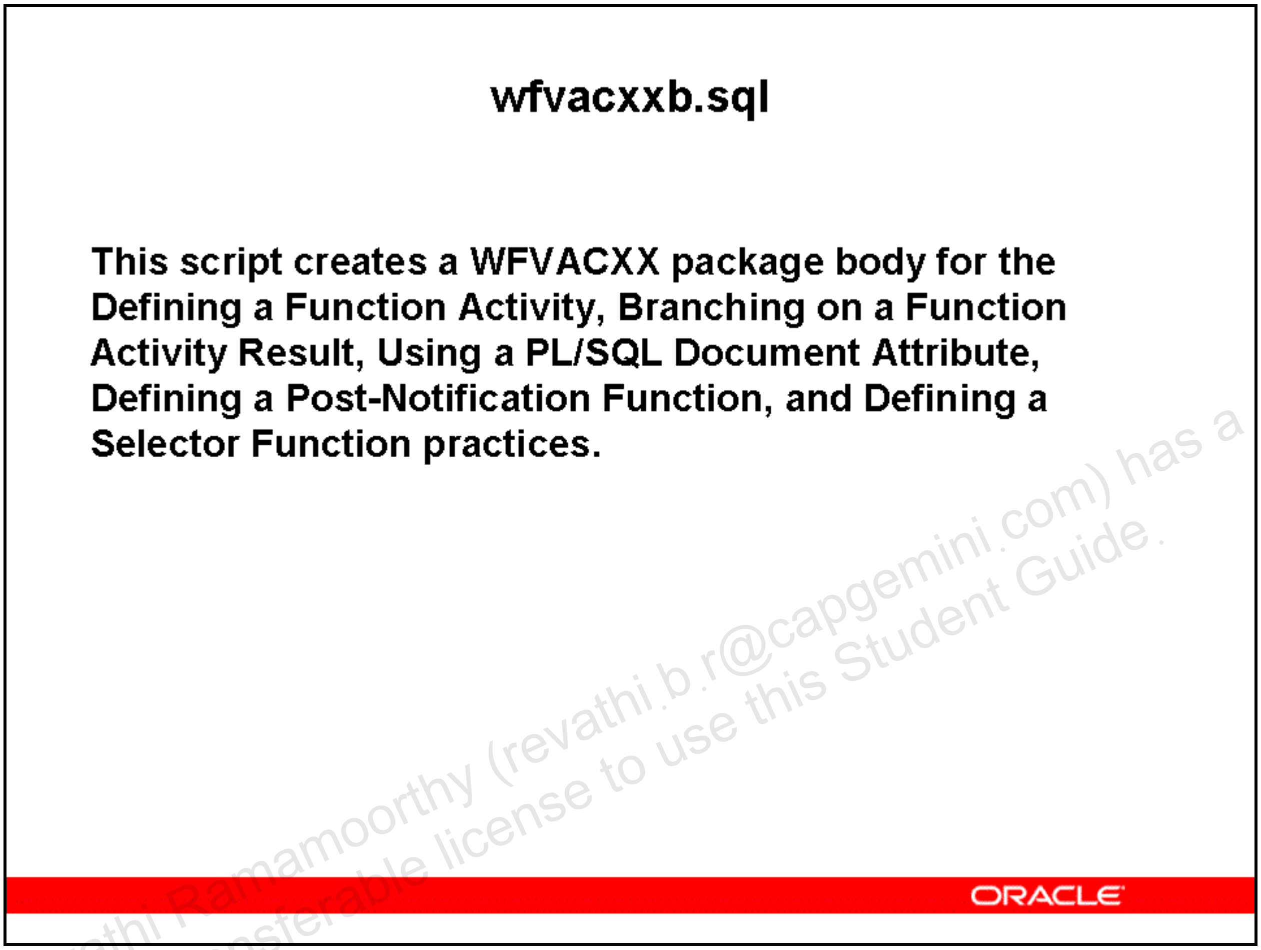
wfvacxxs.sql

```
/*=========================================================================+
 |  Copyright (c) 1995 Oracle Corporation Redwood Shores, California, USA|
 |                          All rights reserved.                          |
 +=========================================================================+
 | FILENAME
 |   wfvacxxs.sql
 |
 | DESCRIPTION
 |   CLASS SAMPLE PL/SQL spec for package WFVACXX
 |
 | NOTES
 |   This file is a SAMPLE that should be modified with your own
 |   package and procedure names before installing.
 |   The script is written so that you can simply replace XX with the
 |   unique number assigned to your station.
 |
 |   It may be convenient to use the following naming standard
 |       - package name is equivalent to the item type internal name
 |       - procedure names are equivalent to the workflow activity
 |         internal name that the procedure implements.
 *=======================================================================*/

whenever sqlerror exit failure rollback;

create or replace package WFVACXX as
/* $Header$ */

-- PROCEDURE SCHEDULE_UPDATE
--
-- <describe the activity here>
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - function execution mode ('RUN', 'CANCEL', 'TIMEOUT', ...)
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivity
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure SCHEDULE_UPDATE(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout in out varchar2);

-- PROCEDURE NTF_VACATION_PROPOSAL
--
-- <describe the activity here>
```

```
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - post-notification function execution mode
--   ('RESPOND','FORWARD','TRANSFER','RUN', 'CANCEL', 'TIMEOUT')
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivty
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure NTF_VACATION_PROPOSAL(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout    in out varchar2);

-- PROCEDURE VACATION_SCHEDULED
--
-- Report vacation scheduled for the current Vacation Proposal requestor
--
-- IN
--   document_id  - string that uniquely identifies the document
--   display_type - text/html or text/plain
-- OUT
--   document     - outbound text buffer
--   document_type- outbound document type of text/html, text/plain, or ''

procedure VACATION_SCHEDULED(
     document_id   in varchar2,
     display_type  in varchar2,
     document      in out varchar2,
     document_type in out varchar2);
```

```
-- PROCEDURE CHECK_APPROVER
--
-- Compares the value of the Vacation Proposal Requestor to the Approver.
-- If the Approver = Requestor, return result Y (Yes)
-- If the Approver <>Requestor, return result N (No)
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - function execution mode ('RUN', 'CANCEL', 'TIMEOUT', ...)
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivity
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure CHECK_APPROVER(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout in out varchar2);

-- PROCEDURE SELECTOR
--
-- Examines the value of item attribute TIMEOUT_CHOICE to select
-- which process to run
-- If TIMEOUT_CHOICE = LOOP COUNTER, then run WFVACXX_PROCESS
-- If TIMEOUT_CHOICE = SELF LOOP, then run WFVACXX_ALTERNATE_PROCESS
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   command   - execution mode ('RUN', 'SET_CTX','TEST_CTX',...)
-- OUT
--   resultout
--       - name of process to run
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure SELECTOR(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     command   in varchar2,
     resultout in out varchar2);

end WFVACXX;
/

show errors package WFVACXX
commit;
--exit;
```

wfvacxxb.sql



wfvacxxb.sql

This script creates a WFVACXX package body for the Defining a Function Activity, Branching on a Function Activity Result, Using a PL/SQL Document Attribute, Defining a Post-Notification Function, and Defining a Selector Function practices.

wfvacxxb.sql

```
/*========================================================================+
 |  Copyright (c) 1995 Oracle Corporation Redwood Shores, California, USA|
 |                             All rights reserved.                       |
 +========================================================================+
 | FILENAME
 |   wfvacxxb.sql
 |
 | DESCRIPTION
 |   CLASS SAMPLE PL/SQL spec for package WFVACXX
 |
 | NOTES
 |   This file is a SAMPLE that should be modified with your own
 |   names and procedures before installing.
 |   The script is written so that you can simply replace XX with the
 |   unique number assigned to your station.
 |
 |   It may be convenient to use the following naming standard
 |       - package name is equivalent to the item type internal name
 |       - procedure names are equivalent to the workflow activity
 |         internal name that the procedure implements.
 *===========================================================================*/

whenever sqlerror exit failure rollback;

create or replace package body WFVACXX as
/* $Header$ */

-- PROCEDURE SCHEDULE_UPDATE
--
-- Insert a row into an employee based vacation schedule table
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - function execution mode. this is set by the engine
--               as either 'RUN', 'CANCEL', 'TIMEOUT'
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivity
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure SCHEDULE_UPDATE(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout in out varchar2)
is
     lrequestor_username    varchar2(30);
     lapprover_username     varchar2(30);
     lfrom_date             date;
     lto_date               date;
```

```
begin

  --
  -- RUN mode - normal process execution
  --
  if (funcmode = 'RUN') then

    -- retrieve requestor, approver, from, and to dates
    lrequestor_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                 itemkey => itemkey,
                                 aname => 'REQUESTOR');
    lapprover_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                 itemkey => itemkey,
                                 aname => 'APPROVER');
    lfrom_date := wf_engine.GetItemAttrDate(itemtype => itemtype,
                                 itemkey => itemkey,
                                 aname => 'FROM_DATE');
    lto_date := wf_engine.GetItemAttrDate(itemtype => itemtype,
                                 itemkey => itemkey,
                                 aname => 'TO_DATE');


    -- insert row into vacation schedule table
    insert into WFVACXX_VACATION_SCHEDULE
           (REQUESTOR_USERNAME,
            APPROVER_USERNAME,
            FROM_DATE,
            TO_DATE)
    values (lrequestor_username,
            lapprover_username,
            lfrom_date,
            lto_date);

    -- no result needed
    resultout  := wf_engine.eng_completed||':'||wf_engine.eng_null;
    return;
  end if;


  --
  -- CANCEL mode - activity 'compensation'
  --
  -- This is in the event that the activity must be undone,
  -- for example when a process is reset to an earlier point
  -- due to a loop back.
  --
  if (funcmode = 'CANCEL') then

    -- retrieve requestor, approver, from, and to dates
    lrequestor_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                    itemkey => itemkey,
                                    aname => 'REQUESTOR');
    lapprover_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                    itemkey => itemkey,
                                    aname => 'APPROVER');
    lfrom_date := wf_engine.GetItemAttrDate(itemtype => itemtype,
                                    itemkey => itemkey,
                                    aname => 'FROM_DATE');
    lto_date := wf_engine.GetItemAttrDate(itemtype => itemtype,
                                    itemkey => itemkey,
                                    aname => 'TO_DATE');



    -- delete row from vacation schedule table
    delete from WFVACXX_VACATION_SCHEDULE
    where  REQUESTOR_USERNAME = lrequestor_username
      and  APPROVER_USERNAME = lapprover_username
      and  FROM_DATE = lfrom_date
      and  TO_DATE = lto_date;

    -- no result needed
    resultout := wf_engine.eng_completed||':'||wf_engine.eng_null;
    return;
  end if;


  --
  -- Other execution modes may be created in the future.  Your
  -- activity will indicate that it does not implement a mode
  -- by returning null
  --
  resultout := wf_engine.eng_null;
  return;

exception
  when others then
    -- The line below records this function call in the error system
    -- in the case of an exception.
    wf_core.context('WFVACXX', 'SCHEDULE_UPDATE',
                    itemtype, itemkey, to_char(actid), funcmode);
    raise;
end SCHEDULE_UPDATE;
```

```
-- PROCEDURE NTF_VACATION_PROPOSAL
--
-- <describe the activity here>
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - post-notification function execution mode
--   ('RESPOND','FORWARD','TRANSFER','RUN', 'CANCEL', 'TIMEOUT')
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivity
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure NTF_VACATION_PROPOSAL(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout in out varchar2)
is
     nid               number;
     ntf_responder     varchar2(30);
     ntf_result        varchar2(30);
     ntf_alt_from_date date := '';
     ntf_alt_to_date   date := '';

begin

  --
  -- RESPOND mode - recipient has supplied a response to the notification
  --
  if (funcmode = 'RESPOND') then
```

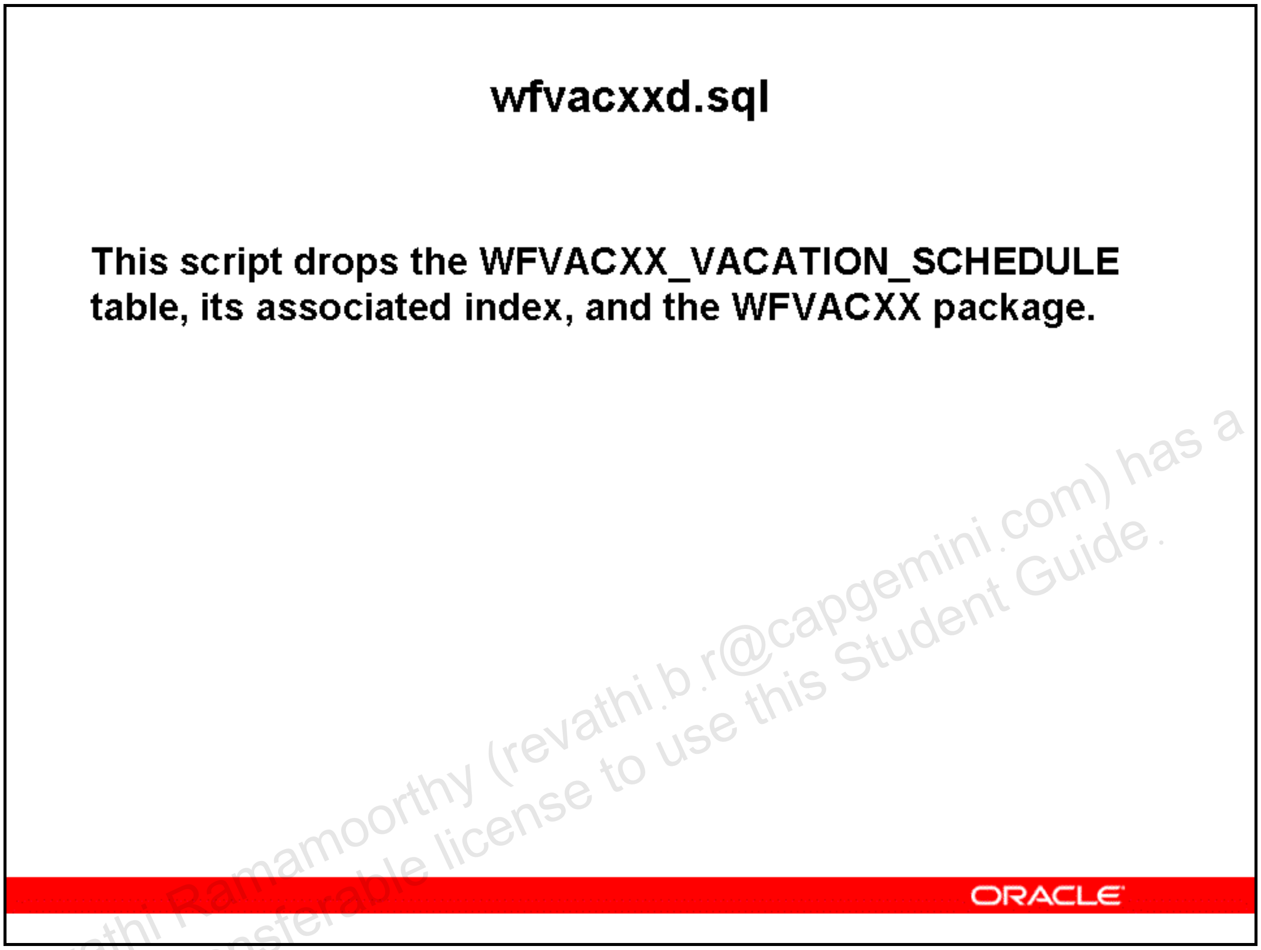
```
      -- get notification id and responder from wf_engine context variables
      nid := WF_ENGINE.CONTEXT_NID;
      ntf_responder := WF_ENGINE.CONTEXT_TEXT;

    -- if the approver rejects the vacation proposal then he/she must
    -- provide an alternate date window

    -- retrieve the notification result
    ntf_result := wf_notification.GetAttrText(nid,'RESULT');
    if (ntf_result = 'REJECTED') then

      -- retrieve the alternate vacation dates
      ntf_alt_from_date :=
      wf_notification.GetAttrDate(nid,'ALT_FROM_DATE');

      ntf_alt_to_date   :=
      wf_notification.GetAttrDate(nid,'ALT_TO_DATE');

      if (ntf_alt_from_date is null
        or ntf_alt_to_date is null) then
           -- raise an error
           resultout  := wf_engine.eng_error||':'||wf_engine.eng_null;
           wf_core.Raise('Provide Alternate Dates');
           return;
      end if;
      if (ntf_alt_from_date > ntf_alt_to_date) then
          -- raise an error
          resultout  := wf_engine.eng_error||':'||wf_engine.eng_null;
          wf_core.Raise('From Date before To Date');
          return;
      end if;
    end if;
    resultout  := wf_engine.eng_completed||':'||ntf_result;
    return;
  end if;

  --
  -- TRANSFER mode - recipient attempting to Transfer notification
  --
  if (funcmode = 'TRANSFER') then

    -- don't allow transfer
    -- raise an error
    resultout  := wf_engine.eng_error||':'||wf_engine.eng_null;
    wf_core.Raise('Transfer not allowed');
    return;
```

```
end if;

--
-- FORWARD mode - recipient attempting to Delegate notification
--
if (funcmode = 'FORWARD') then

  -- delegate allowed
  null;
  -- set resultout to null to indicate that the mode is not implemented
  resultout := wf_engine.eng_null;
  return;
  end if;

--
-- RUN mode - in post-notification function, response to notification
--            already processed and accepted
--
if (funcmode = 'RUN') then
  -- if implemented, your run code goes here
  null;
  -- set resultout to null to indicate that the mode is not implemented
  resultout := wf_engine.eng_null;
  return;
end if;

--
-- TIMEOUT mode - recipient has allowed the notification to timeout
--
if (funcmode = 'TIMEOUT') then
  -- if implemented, your timeout code goes here
  null;
  -- set resultout to null to indicate that the mode is not implemented
  resultout := wf_engine.eng_null;
  return;
end if;


--
-- CANCEL mode - activity 'compensation'
--
-- This is in the event that the activity must be undone,
-- for example when a process is reset to an earlier point
-- due to a loop back.
--
if (funcmode = 'CANCEL') then
```

```
    -- if implemented, your cancel code goes here
    null;
    -- set resultout to null to indicate that the mode is not implemented
    resultout := wf_engine.eng_null;
    return;
  end if;


  --
  -- Other execution modes may be created in the future.  Your
  -- activity will indicate that it does not implement a mode
  -- by returning null
  --
  resultout := wf_engine.eng_null;
  return;

exception
  when others then
    -- The line below records this function call in the error system
    -- in the case of an exception.
    wf_core.context('WFVACXX', 'NTF_VACATION_PROPOSAL',
                    itemtype, itemkey, to_char(actid), funcmode);
    raise;
end NTF_VACATION_PROPOSAL;

-- PROCEDURE VACATION_SCHEDULED
--
-- Report vacation scheduled for the current Vacation Proposal requestor
--
-- IN
--   document_id  - string that uniquely identifies the document
--   display_type - text/html or text/plain
-- OUT
--   document     - outbound text buffer
--   document_type- outbound document type of text/html, text/plain, or ''

procedure VACATION_SCHEDULED(
     document_id   in varchar2,
     display_type  in varchar2,
     document      in out varchar2,
     document_type in out varchar2) is

  cursor vacation_schedule (xrequestor in varchar2) is
     select approver_username,
            to_char(from_date,'Day DD Month YYYY') from_displayed,
            to_char(to_date,'Day DD Month YYYY') to_displayed
     from   wfvacxx_vacation_schedule
     where  requestor_username in
            (select name from wf_users
             where display_name = xrequestor)
     order by from_date, to_date;

begin
     if display_type = 'text/html' then
       document_type := 'text/html';
       document :=
         '<BR><BR><LEFT><TABLE BORDER CELLPADDING=5 BGCOLOR=#FFFFFF>'||
         '<TR BGCOLOR=#83C1C1>'||
         '<TH>From Date</TH>'||
         '<TH>To Date</TH>'||
         '<TH>Approver</TH>'||
         '</TR>';
     else
       document_type := 'text/plain';
       document :=
         chr(10)||rpad('From Date',28)||
         rpad('To Date',28)||
         rpad('Approver',30)||
         chr(10);
     end if;

     -- build table body with data
     for schedule_rec in vacation_schedule(document_id) loop
       if display_type = 'text/html' then
         document := document||
           '<TR>'||
           '<TD>'||schedule_rec.from_displayed||'</TD>'||
           '<TD>'||schedule_rec.to_displayed||'</TD>'||
           '<TD>'||
           wf_directory.getroledisplayname(schedule_rec.approver_username)
           ||'</TD>'||
           '</TR>';
       else
         document := document||
           rpad(schedule_rec.from_displayed,28)||
           rpad(schedule_rec.to_displayed,28)||
           rpad(
           wf_directory.getroledisplayname(schedule_rec.approver_username)
           ,30)||
           chr(10);
       end if;
     end loop;
     -- close the table
     if display_type = 'text/html' then
       document := document||
         '</TABLE></LEFT><BR>';
     end if;

  return;

exception
  when others then
    -- The line below records this procedure call in the error system
    -- in the case of an exception.
    wf_core.context('WFVACXX', 'VACATION_SCHEDULED',
                    document_id, display_type);
    raise;
end VACATION_SCHEDULED;

-- PROCEDURE CHECK_APPROVER
--
-- Compares the value of the Vacation Proposal Requestor to the Approver.
-- If the Approver = Requestor, return result Y (Yes)
-- If the Approver <>Requestor, return result N (No)
--
-- IN
--   itemtype  - type of the current item
--   itemkey   - key of the current item
--   actid     - process activity instance id
--   funcmode  - function execution mode ('RUN', 'CANCEL', 'TIMEOUT', ...)
-- OUT
--   resultout
--       - COMPLETE[:<result>]
--           activity has completed with the indicated result
--       - WAITING
--           activity is waiting for additional transitions
--       - DEFERED
--           execution should be deferred to background
--       - NOTIFIED[:<notification_id>:<assigned_user>]
--           activity has notified an external entity that this
--           step must be performed.  A call to wf_engine.CompleteActivity
--           will signal when this step is complete.  Optional
--           return of notification ID and assigned user.
--       - ERROR[:<error_code>]
--           function encountered an error.
procedure CHECK_APPROVER(
     itemtype  in varchar2,
     itemkey   in varchar2,
     actid     in number,
     funcmode  in varchar2,
     resultout in out varchar2)
is
     lrequestor_username varchar2(30);
     lapprover_username  varchar2(30);
     wf_yes              varchar2(1) := 'Y';
     wf_no               varchar2(1) := 'N';
begin

  --
  -- RUN mode - normal process execution
  --
  if (funcmode = 'RUN') then

    -- retrieve requestor, approver
    lrequestor_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                     itemkey => itemkey,
                                     aname => 'REQUESTOR');
    lapprover_username := wf_engine.GetItemAttrText(itemtype => itemtype,
                                     itemkey => itemkey,
                                     aname => 'APPROVER');

    if lrequestor_username <> lapprover_username then
      resultout  := wf_engine.eng_completed||':'||wf_no;
    else
      resultout  := wf_engine.eng_completed||':'||wf_yes;
    end if;
    return;
  end if;


  --
  -- CANCEL mode - activity 'compensation'
  --
  -- This is in the event that the activity must be undone,
  -- for example when a process is reset to an earlier point
  -- due to a loop back.
  --
  if (funcmode = 'CANCEL') then

    -- no result needed
    resultout := wf_engine.eng_completed||':'||wf_engine.eng_null;
    return;
  end if;
```

```
    --
    -- Other execution modes may be created in the future.  Your
    -- activity will indicate that it does not implement a mode
    -- by returning null
    --
    resultout := wf_engine.eng_null;
    return;

  exception
    when others then
      -- The line below records this function call in the error system
      -- in the case of an exception.
      wf_core.context('WFVACXX', 'CHECK_APPROVER',
                      itemtype, itemkey, to_char(actid), funcmode);
      raise;
  end CHECK_APPROVER;

  -- PROCEDURE SELECTOR
  --
  -- Examines the value of item key to select
  -- which process to run.
  -- If the first four characters of itemkey:
  --   is 'CNTR', then run WFVACXX_PROCESS
  --   is 'SELF', then run WFVACXX_ALTERNATE_PROCESS
  -- Note:  This logic is contrived for class practice only.
  --        The selector function is expected to use the itemkey as
  --        the primary key to retrieve supporting application data.
  --        The Application data retrieved would be used to determine
  --        which process is appropriate to start.
  --
  -- IN
  --   itemtype  - type of the current item
  --   itemkey   - key of the current item
  --   actid     - process activity instance id
  --   command   - execution mode ('RUN', 'SET_CTX','TEST_CTX',...)
  -- OUT
  --   resultout
  --       - name of process to run
  --       - ERROR[:<error_code>]
  --           function encountered an error.
  procedure SELECTOR(
       itemtype  in varchar2,
       itemkey   in varchar2,
       actid     in number,
       command   in varchar2,
       resultout in out varchar2)
  is
       litemkey varchar2(30) := itemkey;
  begin

    --
    -- RUN mode - determine which process to run
    --
    if (command = 'RUN') then

      if UPPER(substr(litemkey,1,4)) = 'CNTR' then
        resultout  := 'WFVACXX_PROCESS';
      elsif UPPER(substr(litemkey,1,4)) = 'SELF' then
        resultout  := 'WFVACXX_ALTERNATE_PROCESS';
      else
        resultout  := wf_engine.eng_error||':'||wf_engine.eng_null;
        wf_core.Raise('Invalid item key');
      end if;
      return;
    end if;


    --
    -- SET_CTX mode
    --
    if (command = 'SET_CTX') then

      -- no result needed
      resultout := wf_engine.eng_null;
      return;
    end if;

    if (command = 'TEXT_CTX') then

      -- no result needed
      resultout := wf_engine.eng_null;
      return;
    end if;

    --
    -- Other execution modes may be created in the future.  Your
    -- selector will indicate that it does not implement a mode
    -- by returning null
    --
    resultout := wf_engine.eng_null;
    return;

  exception
    when others then
      -- The line below records this function call in the error system
      -- in the case of an exception.
      wf_core.context('WFVACXX', 'SELECTOR',
                      itemtype, itemkey, to_char(actid), command);
      raise;
  end SELECTOR;

  end WFVACXX;
  /

  show errors package body WFVACXX
  commit;
  --exit;
```

wfvacxxd.sql



wfvacxxd.sql

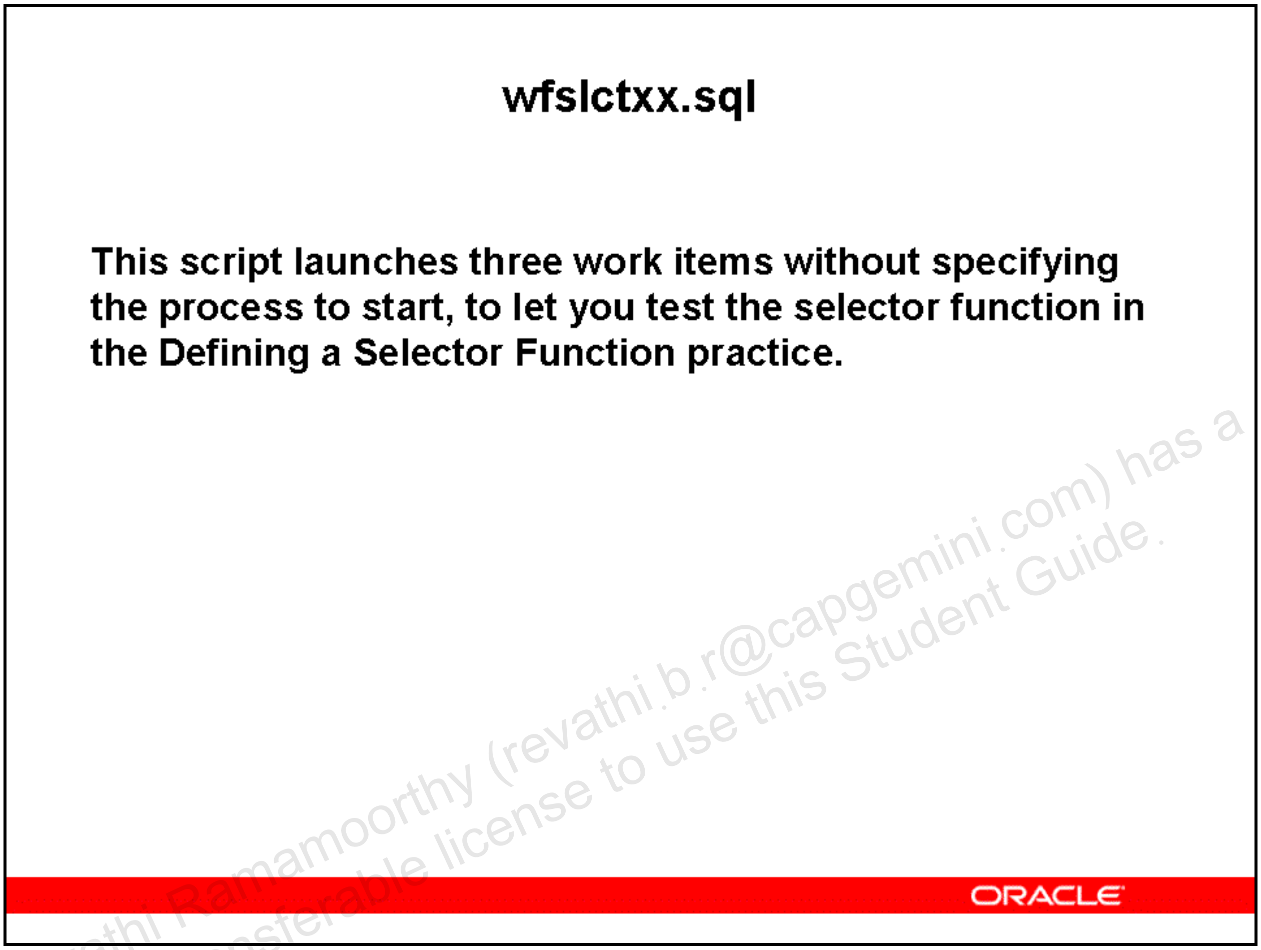
This script drops the WFVACXX_VACATION_SCHEDULE table, its associated index, and the WFVACXX package.

wfvacxxd.sql

```
/*=======================================================================+
 |  Copyright (c) 1995 Oracle Corporation Redwood Shores, California, USA|
 |                            All rights reserved.                        |
 +=======================================================================+
 | FILENAME
 |   wfvacxxd.sql
 |
 | DESCRIPTION
 |  Drop Workflow Vacation Schedule index, table and package.
 | NOTES
 |   This file is a SAMPLE that should be modified with your own
 |   names before installing.  Names that include
 |   XX should be replaced with values for your implementation.
 |
 *=======================================================================*/
```

```
/* $Header$ */
whenever sqlerror continue;
--
drop   index WFVACXX_VACATION_SCHEDULE_N1;
--
drop   table WFVACXX_VACATION_SCHEDULE;
--
drop   package WFVACXX;

commit;

--exit
```

wfslctxx.sql



wfslctxx.sql

This script launches three work items without specifying the process to start, to let you test the selector function in the Defining a Selector Function practice.

wfslctxx.sql

```
/*=========================================================================+
 |  Copyright (c) 2004 Oracle Corporation Redwood Shores, California, USA|
 |                          All rights reserved.                         |
 +=========================================================================+
 | FILENAME
 |   wfslctxx.sql
 |
 | DESCRIPTION
 |  Launch three work items without specifying the process to start,
 |    to test the selector function in the Defining a Selector Function
 |    practice
 |
 | NOTES
 |   This file is a SAMPLE that should be modified with your own
 |   values before running.  Names that include
 |   XX should be replaced with values for your implementation.
 |   Additionally, you should replace the following with appropriate
 |   values for your tests:
 |
 |  <requestor_username> - Vacation requestor
 |  <approver_username>  - Vacation approver
 |  <from_date>          - Vacation from date
 |  <to_date>            - Vacation to date
 |
 *=========================================================================*/

/* $Header$ */
whenever sqlerror continue;

--
-- Create and start a work item with an item key that begins with CNTR
--

Exec WF_ENGINE.CreateProcess('WFVACXX', 'CNTR211XX');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'CNTR211XX', 'REQUESTOR',
     '<requestor_username>');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'CNTR211XX', 'APPROVER',
     '<approver_username>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'CNTR211XX', 'FROM_DATE',
     '<from_date>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'CNTR211XX', 'TO_DATE',
'<to_date>');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'CNTR211XX', 'VACEVENTKEY',
'211XXE1');

Exec WF_ENGINE.StartProcess('WFVACXX', 'CNTR211XX');

--
-- Create and start a work item with an item key that begins with SELF
--

Exec WF_ENGINE.CreateProcess('WFVACXX', 'SELF211XX');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'SELF211XX', 'REQUESTOR',
     '<requestor_username>');
```

```
Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'SELF211XX', 'APPROVER',
     '<approver_username>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'SELF211XX', 'FROM_DATE',
     '<from_date>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'SELF211XX', 'TO_DATE',
'<to_date>');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'SELF211XX', 'VACEVENTKEY',
'211XXE2');

Exec WF_ENGINE.StartProcess('WFVACXX', 'SELF211XX');

--
-- Create and start a work item with an item key that begins with TEST
--
-- If you use the selector function provided in the sample solutions,
-- the CreateProcess procedure should return a ORA-20002: [Invalid itemkey]
-- error, and the following procedures should also error out
-- because the work item does not exist.
--

Exec WF_ENGINE.CreateProcess('WFVACXX', 'TEST211XX');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'TEST211XX', 'REQUESTOR',
     '<requestor_username>');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'TEST211XX', 'APPROVER',
     '<approver_username>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'TEST211XX', 'FROM_DATE',
     '<from_date>');

Exec WF_ENGINE.SetItemAttrDate('WFVACXX', 'TEST211XX', 'TO_DATE',
'<to_date>');

Exec WF_ENGINE.SetItemAttrText('WFVACXX', 'TEST211XX', 'VACEVENTKEY',
'211XXE3');

Exec WF_ENGINE.StartProcess('WFVACXX', 'TEST211XX');

--
commit;

--exit
```