

# **R11i Oracle Alert**

**Student Guide**

14492GC11

Production 1.1

September 2000

M013067

**ORACLE<sup>®</sup>**

**Copyright © Oracle Corporation, 2000. All rights reserved.**

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

#### **Restricted Rights Legend**

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of the Education Products group of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Worldwide Education Services, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

Oracle and all references to Oracle Products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

## **Authors**

Clara Jaeckel, Deborah Steffen

## **Technical Contributors and Reviewers**

Andrea Auld, Siu Chang, Richard Drage, Martyn James Kavanagh, Liz MacKelden, Sylvia Pinner, Traci Short, Mike Smith, Adrian Tate, Markus Tjahjadi

**This book was published using:**

**Oracle<sup>®</sup> Tutor<sup>™</sup>**



# Table of Contents

---

<b>Introduction to Oracle Alert.....</b>	<b>1-1</b>
Introduction to Oracle Alert.....	1-2
Course Objectives.....	1-3
Oracle Alert Process Overview.....	1-4
What Is an Exception?.....	1-5
Examples of Event Alerts.....	1-6
Examples of Periodic Alerts.....	1-7
Action Types.....	1-8
Action Levels.....	1-9
Seven Steps to Implementing an Alert.....	1-10
<b>Defining an Event Alert.....</b>	<b>2-1</b>
Defining an Event Alert.....	2-2
Objectives.....	2-3
What Is an Alert?.....	2-4
Business Problem.....	2-5
Research.....	2-6
Alert Terminology.....	2-7
Defining an Event Alert.....	2-8
Defining Event Details.....	2-9
Writing the SQL SELECT Statement.....	2-10
Verifying the SQL SELECT Statement.....	2-11
Specifying Alert Inputs.....	2-12
Defining Alert Outputs.....	2-13
Defining an Event Alert Action.....	2-14
Defining Action Details.....	2-15
Defining Message Action Details.....	2-16
Defining an Alert Action Set.....	2-17
Defining Alert Action Set Members.....	2-18
Testing an Alert.....	2-19
Reviewing Alert History.....	2-20
Reviewing Alert Checks.....	2-21
Reviewing Action Set Checks.....	2-22
Displaying Output Variables.....	2-23
Reviewing Alert Exceptions.....	2-24
Reviewing Alert Actions.....	2-25
Reviewing the Alert Action Log.....	2-26
Reviewing Alert History - Navigation Path Summary.....	2-27
Documenting an Alert.....	2-28
Practice 2-1.....	2-29
Practice 2-2.....	2-31
Summary.....	2-39
<b>Defining a Periodic Alert.....</b>	<b>3-1</b>
Defining a Periodic Alert.....	3-2
Objectives.....	3-3
Business Problem.....	3-4
Research.....	3-5
Defining a Periodic Alert.....	3-6
Defining Periodic Details.....	3-7
Predefined Frequencies.....	3-8
Writing the SQL SELECT Statement.....	3-9

Verifying and Running the SQL SELECT Statement.....	3-10
Specifying Alert Inputs.....	3-11
Defining Alert Outputs.....	3-12
Defining a Periodic Alert Action.....	3-13
Defining Action Details.....	3-14
Defining Message Action Details.....	3-15
Defining a Periodic Alert Action Set.....	3-16
Defining Alert Action Set Members.....	3-17
Modifying Action Set Inputs.....	3-18
Checking a Periodic Alert.....	3-19
Reviewing Alert History.....	3-20
Documenting the Alert.....	3-21
Practice 3-1.....	3-22
Practice 3-2.....	3-24
Practice 3-3.....	3-30
Summary.....	3-36
<b>Summary Thresholds, Copying, and Deleting Alerts.....</b>	<b>4-1</b>
Summary Thresholds, Copying, and Deleting Alerts.....	4-2
Objectives.....	4-3
How a Summary Threshold Works.....	4-4
Business Problem.....	4-5
Creating a Summary Threshold Alert.....	4-6
Defining a Threshold Group.....	4-7
Adding a Threshold Group to an Action Set.....	4-8
Testing and Running the Alert.....	4-9
Checking Actions and Exceptions.....	4-10
Copying Alert Definitions.....	4-11
Copying Alerts.....	4-12
Deleting Alert Definitions.....	4-13
Practice 4-1.....	4-14
Practice 4-2.....	4-16
Summary.....	4-22
<b>Implementing Duplicate Checking.....</b>	<b>5-1</b>
Implementing Duplicate Checking and Action Escalation.....	5-2
Objectives.....	5-3
Duplicate Exceptions.....	5-4
Implementing Duplicate Checking.....	5-5
Saving Alert History.....	5-6
Defining the Default Duplicate Configuration.....	5-7
Modifying the Duplicate Configuration for an Action Set.....	5-8
Suppressing Actions for Duplicate Exceptions.....	5-9
Action Escalation.....	5-10
Implementing Action Escalation.....	5-11
Defining Actions for Escalation.....	5-12
Defining an Escalation Group.....	5-13
Adding an Escalation Group to an Action Set.....	5-14
Suppressing Escalation Group Actions for Duplicate Exceptions.....	5-15
Practice 5-1.....	5-16
Practice 5-2.....	5-18
Summary.....	5-24
<b>Additional Alert Actions and Features.....</b>	<b>6-1</b>
Additional Alert Actions and Features.....	6-2
Objectives.....	6-3
Alert Actions.....	6-4
SQL Statement Script Actions.....	6-5

Critical Outputs in SQL Statement Script Actions .....	6-6
Business Problem.....	6-7
Defining SQL Statement Script Actions.....	6-8
Defining SQL Statement Script Action Details .....	6-9
SQL Statement Script Example .....	6-10
Practice 6-1 .....	6-11
Concurrent Programs .....	6-15
Concurrent Program Action Levels .....	6-16
Concurrent Program Action Processing.....	6-17
Defining Concurrent Program Action Details .....	6-18
Practice 6-2 .....	6-19
Operating System Script Actions.....	6-25
Critical Outputs in Operating System Script Actions .....	6-26
Operating System Script Action Levels.....	6-27
Defining Operating System Script Action Details .....	6-28
Distribution Lists .....	6-29
Using Distribution Lists.....	6-30
Defining Distribution Lists .....	6-31
Periodic Sets .....	6-32
Defining Periodic Sets .....	6-33
Using Periodic Sets.....	6-34
Precoded Alerts.....	6-35
Precoded DBA Alerts .....	6-36
Precoded Purging Alerts .....	6-37
Using Precoded Alerts .....	6-38
Practice 6-3 .....	6-39
Summary.....	6-41
<b>Response Processing .....</b>	<b>7-1</b>
Response Processing.....	7-2
Objectives .....	7-3
Response Processing.....	7-4
Response Processing Terminology .....	7-6
Implementing Response Processing .....	7-9
Business Problem.....	7-10
Response Processing Example.....	7-11
Defining Response Actions .....	7-12
Defining a Response Set .....	7-13
Defining Response Variables.....	7-14
Creating Response Variables .....	7-15
Defining Possible Responses .....	7-16
Defining Valid Responses .....	7-17
Defining Valid Response Actions.....	7-18
Defining Invalid Response Actions .....	7-19
Using Invalid Response Actions.....	7-20
Defining No Response Actions.....	7-21
Attaching a Response Set to the Alert Message Action.....	7-22
Response Processing Message Structure .....	7-23
Defining Generic Response Text.....	7-25
How Response Processing Works .....	7-26
Practice 7-1 .....	7-29
Practice 7-2 .....	7-31
Summary.....	7-41
<b>Identifying Implementation Issues and Responsibilities for Oracle Alert.....</b>	<b>8-1</b>
Identifying Implementation Issues and Responsibilities for Oracle Alert .....	8-2
Objectives .....	8-3

Implementation Steps .....	8-4
Implementation Terminology .....	8-7
Step 1: Reregister Oracle IDs .....	8-9
Step 2: Create Electronic Mail Accounts .....	8-12
Step 3: Register Oracle Alert Mailboxes .....	8-16
Step 4: Configure Concurrent Managers .....	8-19
Step 5: Define Database Links.....	8-21
Step 6: Register Custom Applications .....	8-25
Step 7: Define Custom Application Installations.....	8-27
Step 8: Register Custom Application Tables .....	8-29
Step 9: Define Response Options .....	8-31
Step 10: Create Mail Folders for Response Accounts .....	8-34
Step 11: Define Message Options.....	8-35
Step 12: Start the Periodic Alert Scheduler .....	8-38
Step 13: Start the Response Processor .....	8-41
Planning Responsibilities for Oracle Alert .....	8-44
Oracle Alert Manager Responsibility .....	8-45
Defining an Application Alert Manager Responsibility .....	8-46
Application Alert Manager Responsibility .....	8-47
Practice 8-1 .....	8-48
Summary.....	8-52

# Preface

---

## Profile

### Before You Begin This Course

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

- Thorough knowledge of SQL
- Working experience with navigating Oracle Applications forms and windows

### Prerequisites

- *Introduction to Oracle: SQL and PL/SQL*
- *R11i Navigating Oracle Applications*

### How This Course Is Organized

*R11i Oracle Alert* 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
<i>Oracle Alert User's Guide</i>	A75398-01

### Additional Publications

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



# Typographic Conventions

## Typographic Conventions in Text

Convention	Element	Example
Bold italic	Glossary term (if there is a glossary)	The <i>algorithm</i> 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: <code>debug.set ('I", 300);</code> Directory: <code>bin</code> (DOS), <code>\$FMHOME</code> (UNIX) Filename: Locate the <code>init.ora</code> file. Password: User <code>tiger</code> as your password. Pathname: Open <code>c:\my_docs\projects</code> URL: Go to <code>http://www.oracle.com</code> User input: Enter <code>300</code> Username: Log on as <code>scott</code>
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address ( <i>but</i> Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	Do <i>not</i> save changes to the database. For further information, see <i>Oracle7 Server SQL Language Reference Manual</i> . Enter <code>user_id@us.oracle.com</code> , where <i>user_id</i> 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 <code>SELECT</code> command to view information stored in the <code>LAST_NAME</code> column of the <code>EMP</code> table.

Convention	Element	Example
Arrow	Menu paths	Select File→ Save.
Brackets	Key names	Press [Enter].
Commas	Key sequences	Press and release keys one at a time: [Alternate], [F], [D]
Plus signs	Key combinations	Press and hold these keys simultaneously: [Ctrl]+[Alt]+[Del]

## Typographic Conventions in Code

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

## Typographic Conventions in Navigation Paths

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

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

This simplified path translates to the following:

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

## Notations :

(N) = Navigator

(M) = Menu

(T) = Tab

(I) = Icon

(H) = Hyperlink

(B) = Button

## Typographical Conventions in Help System Paths

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

The following help navigation path, for example—

(Help) General Ledger > Journals > Enter Journals

—represents the following sequence of actions:

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

## Getting Help

Oracle Applications provides you with a complete online help facility.

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

## To display help for a current window:

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

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

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

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

## Searching for Help

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

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

# Introduction to Oracle Alert

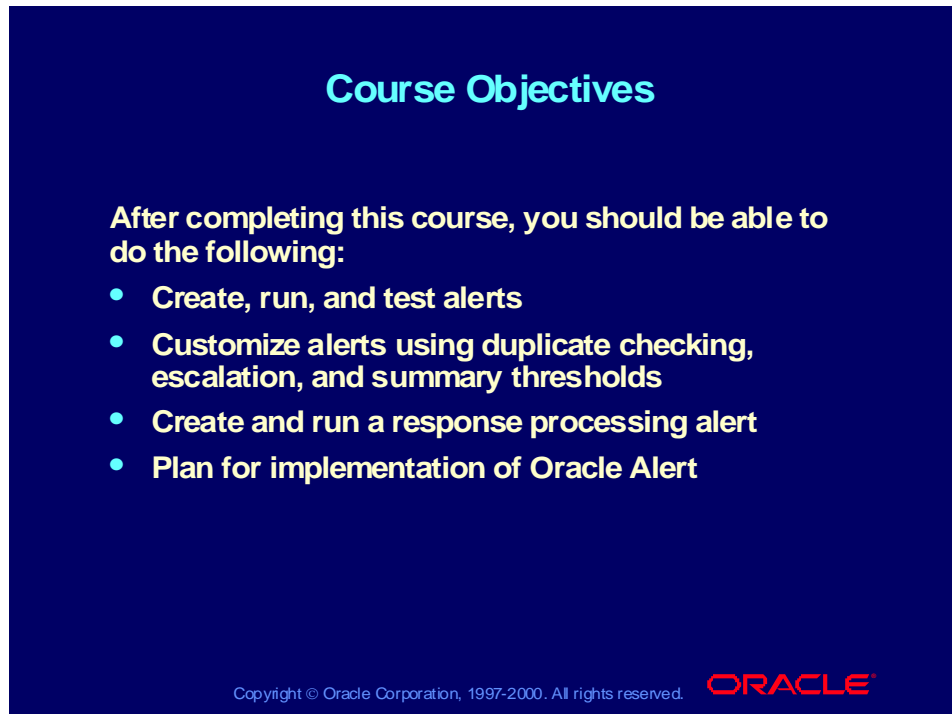
## Chapter 1

## Introduction to Oracle Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

## Course Objectives

---

A dark blue rectangular slide with white text. The title 'Course Objectives' is centered at the top in a light blue font. Below it, the text 'After completing this course, you should be able to do the following:' is followed by a bulleted list of four objectives. At the bottom right, the Oracle logo is displayed in red, and at the bottom left, there is a small copyright notice.

**Course Objectives**

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

- Create, run, and test alerts
- Customize alerts using duplicate checking, escalation, and summary thresholds
- Create and run a response processing alert
- Plan for implementation of Oracle Alert

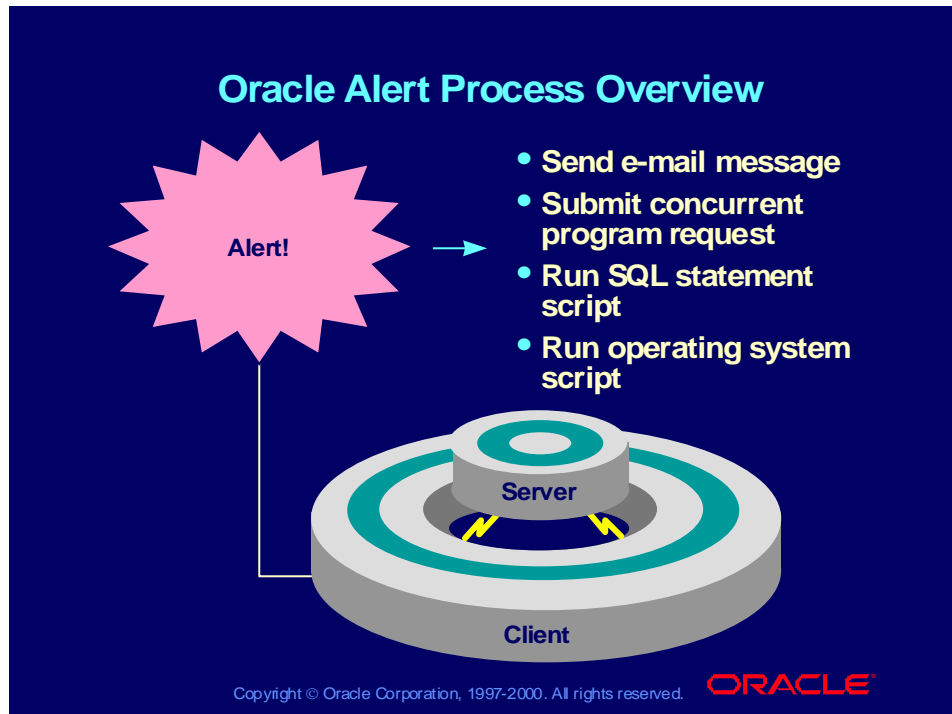
Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Introduction

This course provides instructions for creating exception control solutions for Oracle Applications. Participants will perform hands-on exercises and participate in discussions on how they will use Oracle Alert in the workplace.

## Oracle Alert Process Overview

---



### Oracle Alert Process Overview

How do you find out about important or unusual activity in your database? How do you stay aware of regular, yet critical database events without sorting through lengthy reports?

Alerts monitor your database information and notify you when the condition you have specified is found. You can define Oracle alerts in any Oracle application or custom Oracle application.

You can define one of two types of alerts: event and periodic.

- Event alert - Notifies you of activity in your database as soon as it occurs
- Periodic alert - Checks the database for information according to a schedule that you define

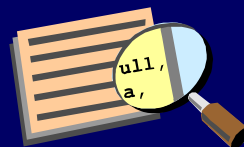


## What Is an Exception?

---

### What Is an Exception?

- An exception is a specified condition found during an alert check.
- For example, an alert checking for users who did not change their passwords within the designated time finds five users that meet the criteria. Each user found is an **exception**.

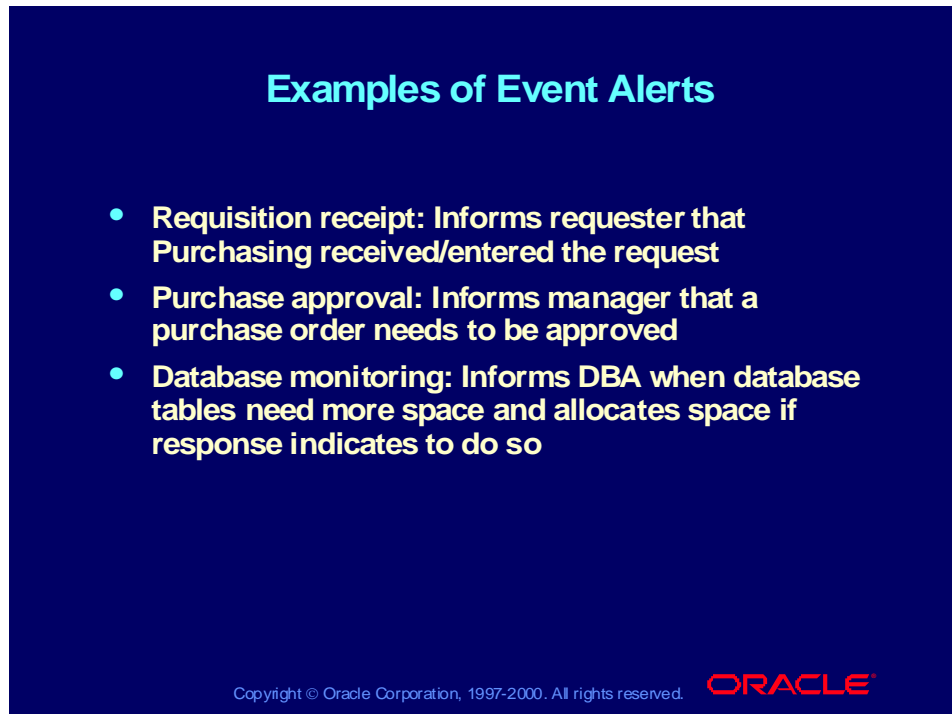


Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Examples of Event Alerts

---



**Examples of Event Alerts**

- **Requisition receipt:** Informs requester that Purchasing received/entered the request
- **Purchase approval:** Informs manager that a purchase order needs to be approved
- **Database monitoring:** Informs DBA when database tables need more space and allocates space if response indicates to do so

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### What Event Alerts Specify

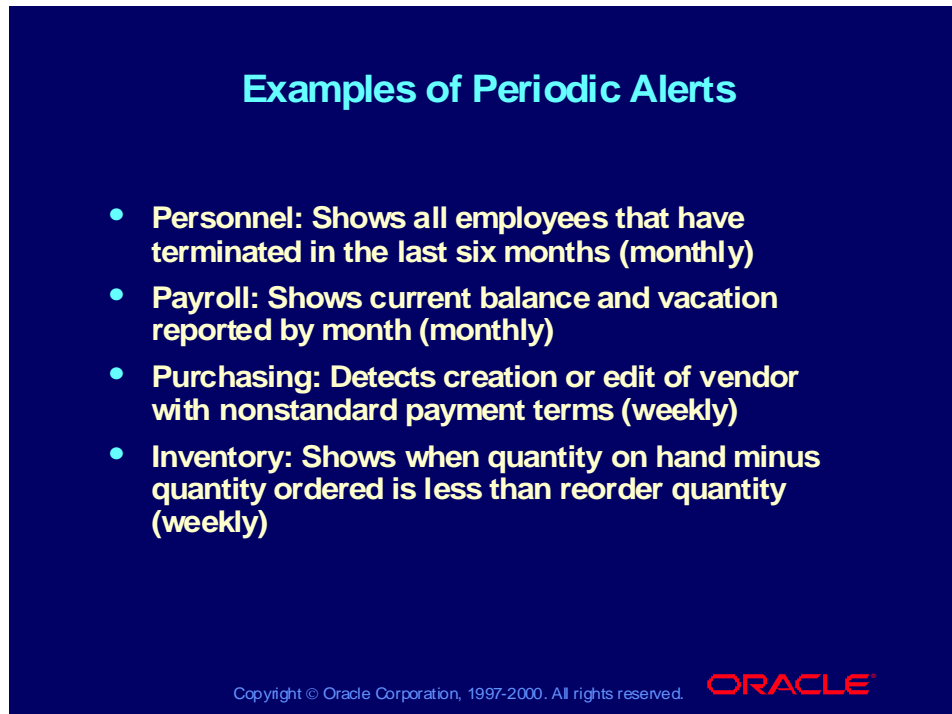
By creating event alerts, you can have an immediate view of the activity in your database, thus keeping up with important or unusual events as they happen.

When you create an event alert, you specify the following:

- A database event that you want to monitor—that is, an insert or an update to a specific database table
- A SQL SELECT statement that retrieves specific database information as a result of the database event
- Actions that you want Oracle Alert to perform as a result of the database event

## Examples of Periodic Alerts

---



**Examples of Periodic Alerts**

- **Personnel:** Shows all employees that have terminated in the last six months (monthly)
- **Payroll:** Shows current balance and vacation reported by month (monthly)
- **Purchasing:** Detects creation or edit of vendor with nonstandard payment terms (weekly)
- **Inventory:** Shows when quantity on hand minus quantity ordered is less than reorder quantity (weekly)

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### What Periodic Alerts Specify

By creating periodic alerts, you can have current measurements of staff and organization performance, so that you can focus on potential trouble spots. You can automate routine transactions.

When you create a periodic alert, you specify the following:

- A SQL SELECT statement that retrieves specific database information
- The frequency with which you want the periodic alert to run the SQL statement
- Actions that you want Oracle Alert to perform once it runs the SQL statement

## Action Types

---

### Action Types

An action is an event that occurs once a monitored database event occurs or once a periodic checking of the database has been performed. Oracle Alert can perform these actions:

- Send an e-mail message
- Submit a concurrent program request
- Run a SQL statement script
- Run an operating system script

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

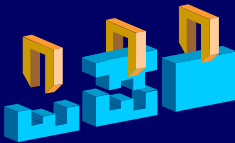
ORACLE

#### Four Action Types

- Messages
  - UNIX Sendmail
  - VMS mail
  - Any MAPI-compliant Windows NT mail application
- Concurrent program request: supply arguments
- SQL statement script
  - Note: The only tables that you can write to directly are custom application tables and open interface tables.
- Operating system script

**Action Levels**

- **Detail:** Perform the action for each occurrence of the condition.
- **Summary:** Perform the action for a group of occurrences of the condition.
- **No Exception:** Perform the action when nothing in the database meets the search criteria.



Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Example

Inform the account manager of invoices on hold

- **Detail Action:** Send a separate e-mail message for each invoice that meets the search criteria of invoices on hold.
- **Summary Action:** Send a single e-mail message listing all invoices that meet the search criteria, or send one summary for each vendor.
- **No Exception Action:** Send an e-mail message stating that nothing in the database is on hold.

## Seven Steps to Implementing an Alert

---



### How to Create an Alert

- 1 Research all the information needed to create your alert.
  - E-mail addresses
  - Required parameters for concurrent programs
  - Table names
  - Column names
  - Application name that owns the table
- 2 Define the alert condition.
  - Name your alert.
  - Select the type.
  - Define event or periodic details.
  - Write your SELECT statement.
  - Verify the SELECT statement.
  - Run the SELECT statement (if a periodic alert).
- 3 Define the actions.
  - Select action level.
  - Select action type.
- 4 Define the action set.
  - Create the sequence of actions.

- Define what to do when an actions fails.
- 5 Run and test your alert by triggering your alert.
  - Event alerts: perform the event.
  - Periodic alerts: use Request—>Check
- 6 Review the alert history.
  - Find checks.
  - Find exceptions.
  - Find actions.
  - View request log.
- 7 Document the alert.





# Defining an Event Alert

## Chapter 2

# Defining an Event Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Objectives

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


- Define event alert conditions
- Use implicit inputs
- Create detail alert actions
- Build an alert action set
- Test the alert
- Evaluate alert history

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

## What Is an Alert?

---

**What Is an Alert?**



- **Alert:** An entity that monitors your business information and notifies you of the information you want
- **Event Alert:** Runs when an insert or an update occurs in a specific Oracle Application or custom Oracle Application database table

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Business Problem

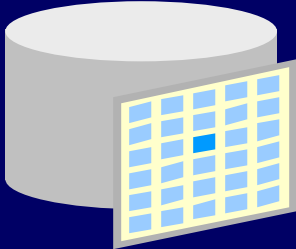
Define an event alert to solve the following business problem for the system administrator:

- **Problem**
  - Need to inform new Oracle Applications users of their new login
  - Need to notify system administrator of new accounts
- **Solution—Create an event alert**
  - Activates when a new application user is created
  - Automatically notifies a new user by e-mail that the user's account is available
  - Sends a confirming message to the system administrator of the new user created

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

**Research**



- Table name
- Application database tables—column names
- Technical reference manuals
- Class materials
- E-mail address books

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Doing Background Research

- Select Help—>Record History to find the name of the table that gets updated when you enter information into a window.
- In the Application Developer responsibility, navigate to the database tables to look up column names.
  - (N) Application—>Database—>Table
- Use your technical reference manual for your application.
- Attend training classes for your application.
- Use e-mail address books to research e-mail usernames.

## Alert Terminology

---

Term	Definition
Input variable	A variable you define in your SQL statement that identifies the conditions for your alert. Input variables start with a colon (:) and usually appear in the WHERE clause.
Output	A value returned by the alert designated by the INTO clause in your SQL statement. It is preceded by an ampersand (&).

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Implicit Input Variables for Event Alerts

Oracle Alert supplies the following values for these implicit inputs as shown below. Any other input values created by your SQL SELECT statement can be changed by you, as you will learn later.

#### **:ROWID**

Contains the ID number of the row where the insert or update that triggers an event alert occurs. This is mandatory for Event Alerts.

#### **:MAILID**

Contains the e-mail username of the person who enters an insert or update that triggers an event alert.

#### **:ORG\_ID**

Contains the organization ID that is selected when the alert runs.

#### **:DATE\_LAST\_CHECKED**

Contains the date and time that the alert was most recently checked.

## Defining an Event Alert

---



**Defining an Event Alert**

Use the Alerts window to enter:

- Application
- Name
- Description
- Event Details
- Keep\_Days
- Select Statement

(N) Alert → Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Defining an Event Alert

Choose the Event region to specify that you are defining an event alert and enter event details.



### Defining Event Details

Use the Event region of the Alerts window to enter event details:

- Application
- Table
- After Insert
- After Update

(N) Alert—>Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE<sup>®</sup>

## Writing the SQL SELECT Statement

---

```
SELECT user_name,  
       email_address  
INTO   &user_name,  
       &email_address  
FROM   fnd_user  
WHERE  rowid = :rowid
```

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Writing the SQL SELECT Statement

In the Select Statement field, enter your SQL SELECT statement. The SQL SELECT statement defines your event alert. You can enter a SELECT statement of up to 64 KB in length.

A single SELECT Statement does the following:

- Defines your alert condition
- Retrieves all information that you need to do the following:
  - Create messages
  - Define SQL statement scripts
  - Define operating system scripts
  - Define concurrent program actions

## Verifying the SQL SELECT Statement

---

### Verifying the SQL SELECT Statement

In the Alerts window , click the Verify button to test the syntax of your SQL SELECT statement.

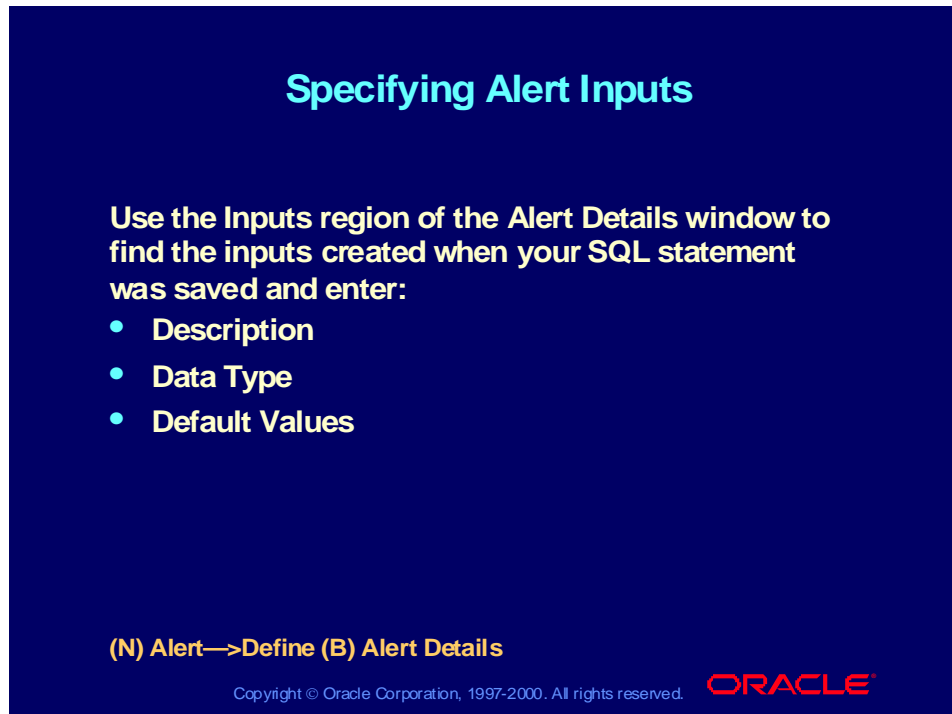
(N) Alert → Define (B) Verify

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Specifying Alert Inputs

---



**Specifying Alert Inputs**

Use the **Inputs** region of the **Alert Details** window to find the inputs created when your **SQL** statement was saved and enter:

- **Description**
- **Data Type**
- **Default Values**

**(N) Alert** → **Define (B) Alert Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

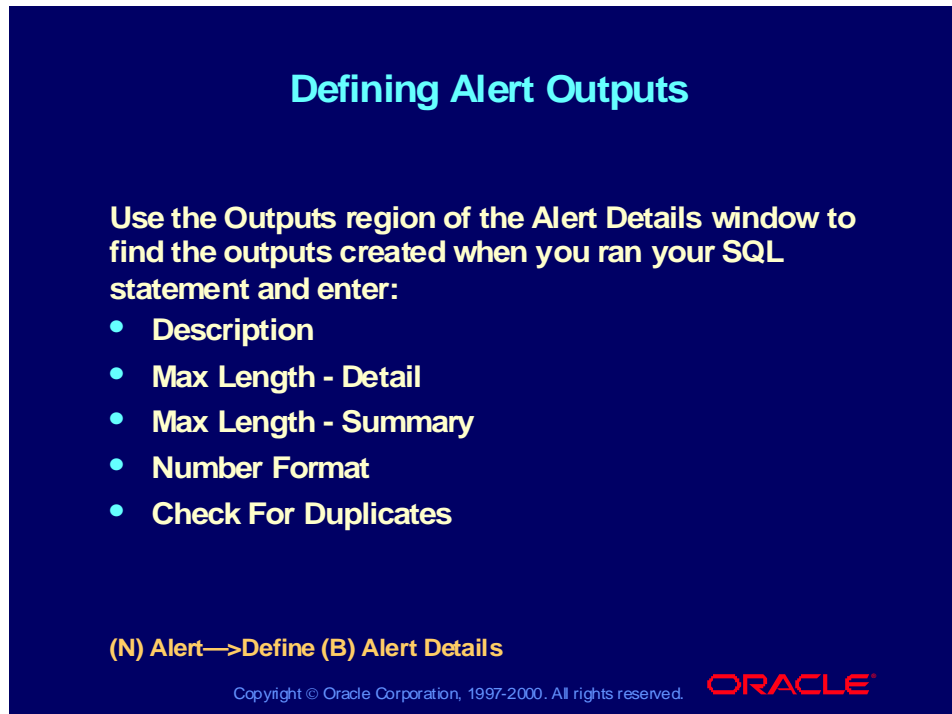
### Alert Details: Inputs/Outputs/Installations Regions

Navigate to the Alert Details window to automatically display the inputs and outputs used in your SQL SELECT statement. You can display one of three alternative regions in this window: Inputs, Outputs, and Installations. Use the Inputs region to set up default values for your alert inputs. Use the Outputs region to define the lengths of the outputs that you can use in your alert actions. Use the Installations region to specify the application installations you want your alert checked against.

- Do a Find All from the menu to see the inputs created when your SQL statement was saved.
- The Inputs region does not display implicit inputs such as :ROWID, because Oracle Alert already knows their types and values.
- You can optionally add a description for each input. You must specify the data type (either character, number, or date) for the input, because Oracle Alert uses the data type to validate the default values you enter in the Default Values field and in the Action Set Inputs alternative region of the Action Sets block.
- Enter a default value for your input. You can enter up to 240 characters. This value automatically becomes the default value for your input in each action set you define.

## Defining Alert Outputs

---



**Defining Alert Outputs**

Use the **Outputs** region of the **Alert Details** window to find the outputs created when you ran your SQL statement and enter:

- **Description**
- **Max Length - Detail**
- **Max Length - Summary**
- **Number Format**
- **Check For Duplicates**

(N) Alert → Define (B) Alert Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining Alert Outputs

The Outputs region displays outputs from the INTO clause of your SQL SELECT statement.

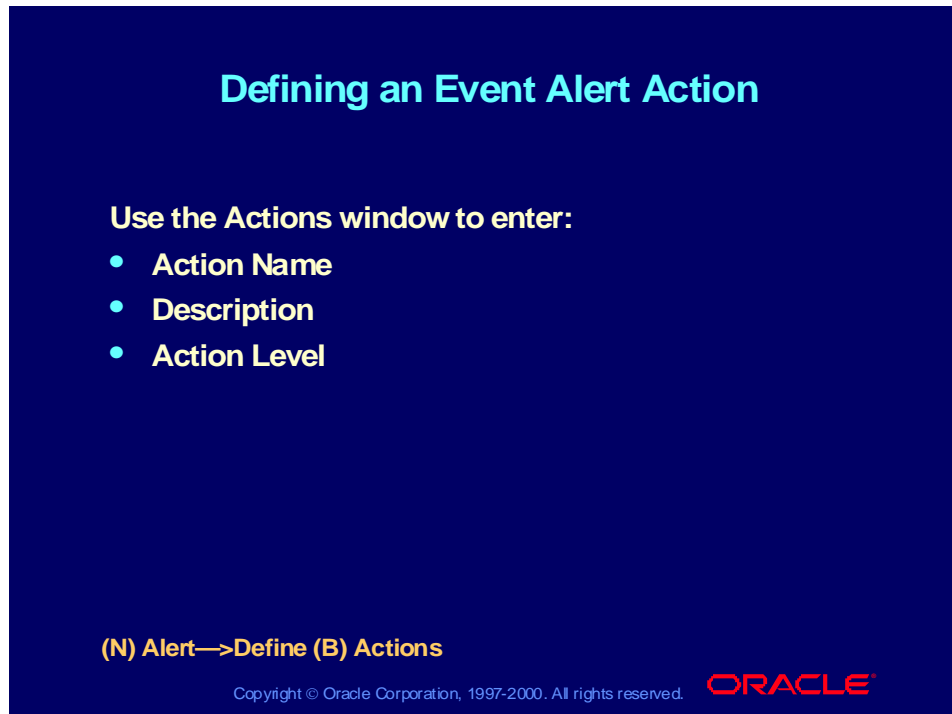
- 1 Do a Find All to see the outputs created when you ran your SQL statement.
- 2 Optionally, you can enter the number of characters of data that you want to display for each output in your actions.
- 3 Save and close the window.

### Defining Alert Installations

The Alert Details Installation region lets you specify the schema and organization names used by Alert to access tables. Query to see the schemas and organizations associated with the application tied to your alert name.

## Defining an Event Alert Action

---



The screenshot shows a dark blue window titled "Defining an Event Alert Action". The text inside the window reads: "Use the Actions window to enter:" followed by a bulleted list: "• Action Name", "• Description", and "• Action Level". At the bottom left, it says "(N) Alert—>Define (B) Actions". At the bottom right, there is the Oracle logo and the text "Copyright © Oracle Corporation, 1997-2000. All rights reserved."

### Defining an Action

For each type of action, you can select three levels of actions.

- **Detail:** Represents one action for each exception found during the alert check.
- **Summary:** Represents one action for a group of exceptions. Groups exceptions either in a single message or in another type of action, depending upon the action and action set definition.
- **No Exception:** Performs when no exceptions are found during an alert check.

### Defining Action Details

Use the Action Details window to select an action type:

- Message
- Concurrent Program
- Operating System Script
- SQL Statement Script

Oracle Alert displays only those fields that are relevant to that action type.

(N) Alert → Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Defining Message Action Details

---

**Defining Message Action Details**

Use the Action Details window to enter:

- To
- Cc
- Bcc
- Subject
- File or Text

(N) Alert → Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Defining Message Action Details

- 1 Enter the recipient's name in the To field.

**Note:** Create a dynamic message by using one of the alert outputs as the recipient. Alert automatically determines the recipient based on the information returned during the alert check. You do not know who the recipient is when you define a dynamic message. Using an alert output as a recipient implies that you are storing an electronic mail name in your application.
- 2 Enter the subject text.

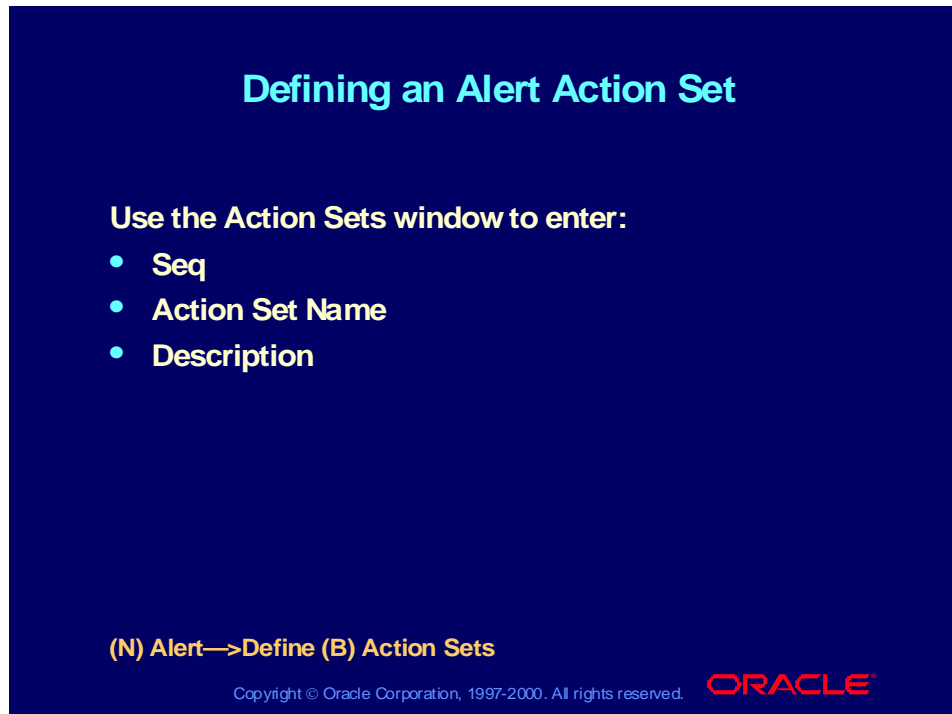
**Note:** You can enter text or outputs (if appropriate) in the subject field. Oracle Alert performs string substitution on the outputs.
- 3 Select the Text option and enter the message text, or select File and retrieve a file already in existence.

**Note:** In a detail message, you enter the text exactly as you want it to appear. You can use outputs within the text. Oracle Alert substitutes the information returned into the message, expanding or contracting the text to provide a formatted message.
- 4 Save and close the window.



## Defining an Alert Action Set

---



The screenshot shows a dark blue window titled "Defining an Alert Action Set". The text inside the window reads: "Use the Action Sets window to enter:" followed by a bulleted list: "• Seq", "• Action Set Name", and "• Description". At the bottom left, it says "(N) Alert → Define (B) Action Sets". At the bottom right, there is the Oracle logo and the text "Copyright © Oracle Corporation, 1997-2000. All rights reserved."

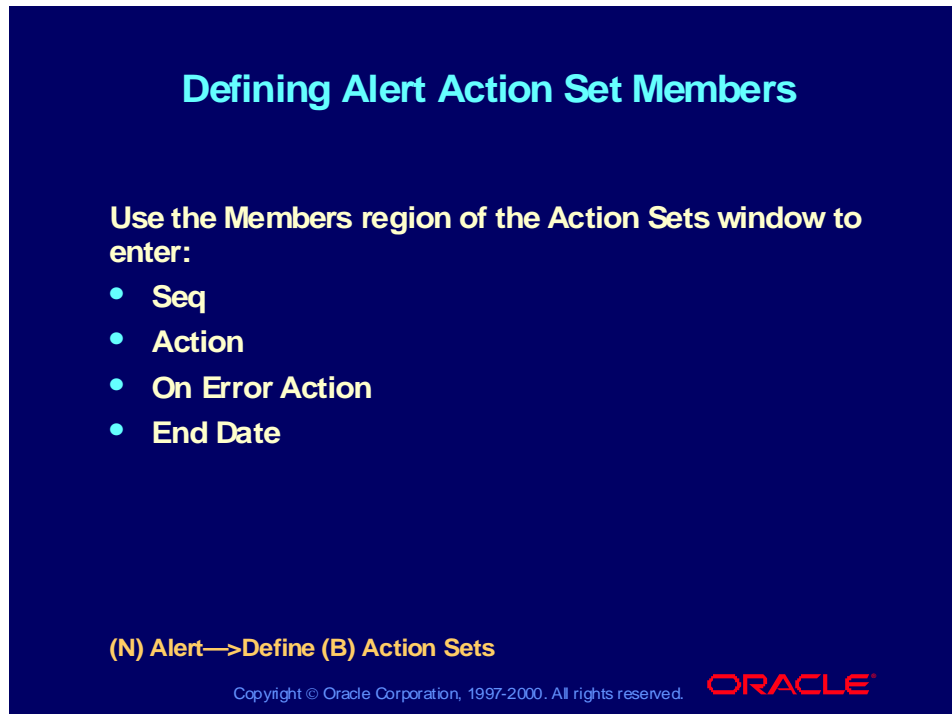
### Defining an Alert Action Set

Oracle Alert can perform an action for an alert only if the action is included in an action set.

- Because you can create more than one action set, you must specify the order in which Oracle Alert runs each action set by entering a sequence number.
- Oracle Alert uses a default sequence number for a new action set.
- If you have more than one action set, you can change the sequence numbers at any time. You can define as many action sets for an alert as you need, but you must have at least one action set.

## Defining Alert Action Set Members

---



**Defining Alert Action Set Members**

Use the **Members** region of the **Action Sets** window to enter:

- **Seq**
- **Action**
- **On Error Action**
- **End Date**

(N) Alert → Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining Alert Action Set Members

#### Create the Sequence of Actions

Determine the order in which Oracle Alert performs the actions in your action set by specifying the sequence number for each action.

#### When an Action Fails

You control what happens to an action set if one of its actions fails. You can abort the action set, continue with another action in the action set, or start a new action set.

## Testing an Alert

---



### Testing Alerts

Event alerts fire off a concurrent request whose status can only be seen under the System Administrator responsibility. To see the request in the System Administrator responsibility, select Concurrent—>Requests from the Navigator. To see the log file, from within the Requests window, choose (M) Tools—>Manager Log.

### Reviewing Alert History

Use the Review Alert History window to enter:

- Application
- Alert Name
- View History Dates

(N) History

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Reviewing Alert Checks

Use the Review Alert Checks window to view:

- Requestor
- Request Date and Time
- Request ID
- Status
- History Purge Date

(N) History (B) Find Checks

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Reviewing Action Set Checks

Use the Action Set Checks region of the Review Alert Checks window to view:

- Action Set Name
- Oracle ID
- Status
- Number of Exceptions
- Number of Actions

(N) History (B) Find Checks

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Displaying Output Variables

---

### Displaying Output Variables

Use the **Display Output Variables** window to select the output variables you want to review.

**(N) History (B) Find Exceptions**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Reviewing Alert Exceptions

Use the Review Alert Exceptions window to view:

- Action Set Name
- Request Date and Time
- Oracle ID
- Alert Exceptions
  - Variable Name
  - Value

(N) History (B) Find Exceptions (B) OK

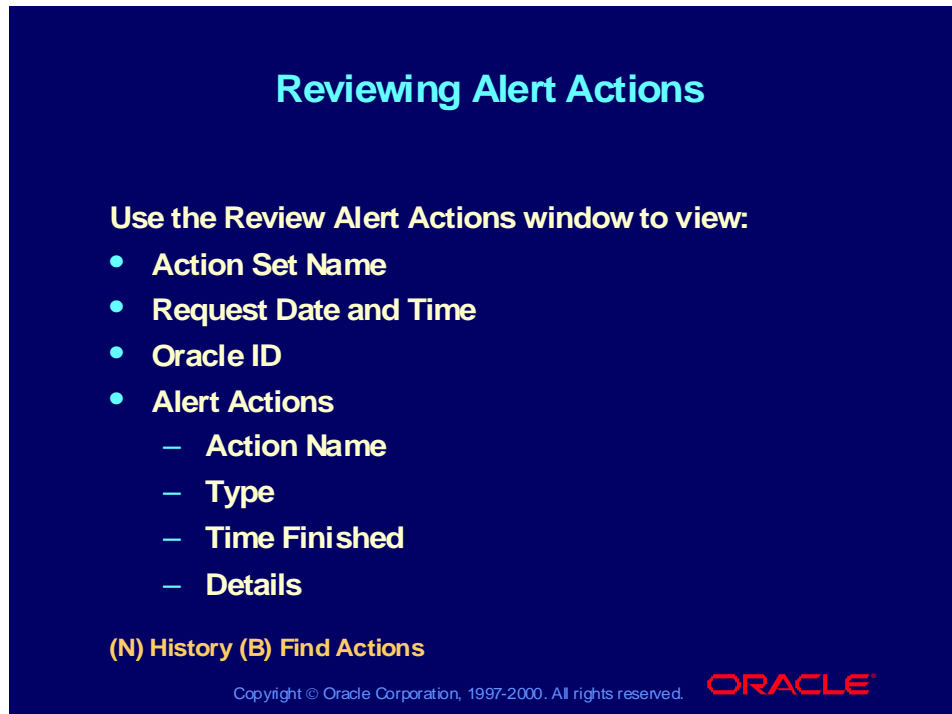
Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Reviewing Alert Actions

---



**Reviewing Alert Actions**

Use the Review Alert Actions window to view:

- Action Set Name
- Request Date and Time
- Oracle ID
- Alert Actions
  - Action Name
  - Type
  - Time Finished
  - Details

(N) History (B) Find Actions

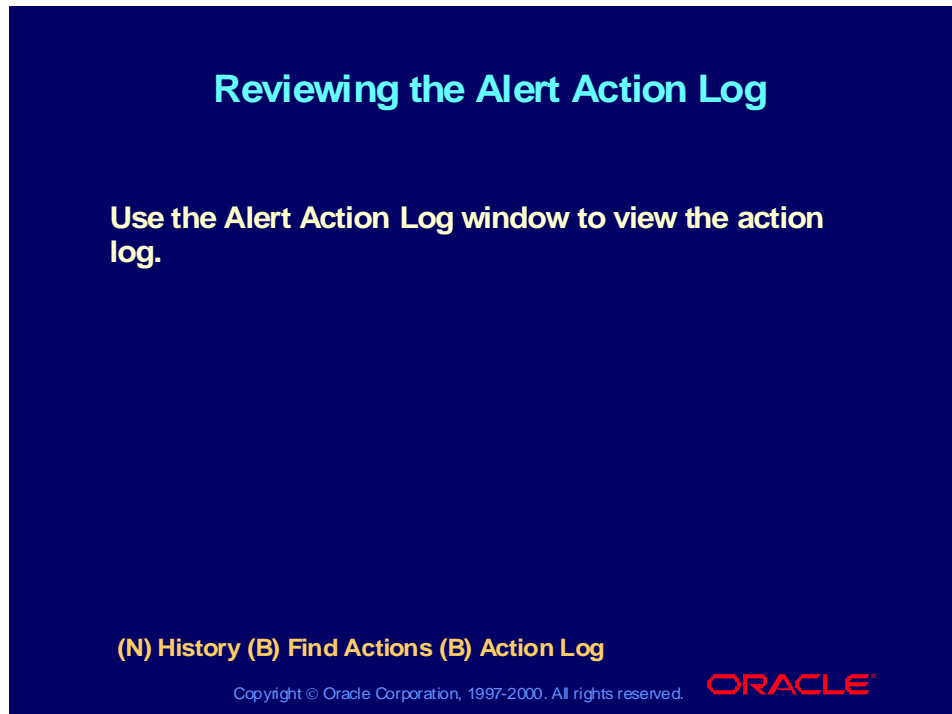
Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Reviewing Alert Actions

- Oracle Alert displays the list of actions that it performed.
- If the alert action is a message, the recipients and subject are displayed.
- If the alert action is a concurrent program, the start date, name, and any arguments passed to the program are displayed.
- If the alert action is an operating system script, any arguments passed to the script are displayed. If the script came from an external file, the name of the file is also displayed.

## Reviewing the Alert Action Log

---



### Reviewing the Alert Action Log

- If there was an error executing the alert, you will see error messages.
- If the alert action is a message defined in Oracle Alert, you can see the message sent.
- If the message text came from an external file, you can view the contents of the file.
- If the alert action is a SQL statement script or an operating system script defined in Oracle Alert, you can view the complete script exactly as it was performed.

## Reviewing Alert History - Navigation Path Summary

---

### Reviewing Alert History - Navigation Path Summary

- **Review Alert Checks**  
(N) History (B) Find Checks
- **Display Output Variables**  
(N) History (B) Find Exceptions
- **Review Alert Exceptions**  
(N) History (B) Find Exceptions (B) OK
- **Review Alert Actions**  
(N) History (B) Find Actions
- **Alert Action Log**  
(N) History (B) Find Actions (B) Action Log
- **Response History Review**  
(N) History (B) Find Actions (B) Response


Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Documenting an Alert

---

**Documenting an Alert**



- Explanations of programs, SQL, scripts**
- Unusual options used**
- Tables  
Column names**
- E-mail  
Usernames**

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Practice 2-1

This practice covers the following topics:

- Defining an alert
- Describing the types of actions that Oracle Alert can perform
- Describing the levels of actions that Oracle Alert can perform

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 2-1 Instructions

Discuss the following questions:

- 1 What is Oracle Alert?
- 2 How does an event alert work?
- 3 What are :ROWID and :MAILID?
- 4 What are the types of actions that Oracle Alert can perform?
- 5 What are the three levels of actions used for?
- 6 How do I determine the order in which my actions will be performed, and what happens if one fails?
- 7 Where can I look to debug my alert?
- 8 How do I determine what schema my alert uses to query database tables?

### Practice 2-1 Solutions

- 1 What is Oracle Alert?

**Oracle Alert is a software tool that monitors your database and notifies you when a condition you have specified is found. It can be used with any Oracle application as well as with any custom Oracle application.**

- 2 How does an event alert work?

**You define the event to be monitored and the actions to be performed when the event occurs.**

- 3 What are :ROWID and :MAILID?  
**:ROWID is an implicit input available in Oracle Alert that returns the current row from the event table. :MAILID contains the e-mail username of the person who enters an insert or update that triggers an event alert.**
- 4 What are the types of actions that Oracle Alert can perform?  
**The action types are: send an e-mail message; submit a concurrent program request; run a SQL statement script; and run an operating system script.**
- 5 What are the three levels of actions used for?  
**The action levels are: Detail, Summary, and No Exception. Detail actions perform a separate action for each exception found. A Summary action performs the action for a group of exceptions found. A No Exception action performs the action when no exceptions are found.**
- 6 How do I determine the order in which my actions will be performed, and what happens if one fails?  
**You put your actions into an action set and define the sequence in which they are performed. You can also specify the action to be taken if an error occurs. Oracle Alert cannot perform an alert action unless it is in an action set.**
- 7 Where can I look to debug my alert?  
**View your alert history request log.  
(N) History (B) Find Checks (M) Tools—>Request Log**
- 8 How do I determine what schema my alert uses to query database tables?  
**View the alert details installations.  
(N) Alert—>Define (B) Alert Details—>Installations**

### Practice 2-2

This practice covers the following topics:

- Defining an alert
- Defining alert actions
- Defining an action set
- Running and testing an alert
- Reviewing history for an alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 2-2 Instructions

#### Practice Notes

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

You will need an e-mail account to receive and reply to the e-mail messages that Oracle Alert generates as a result of your work in these practices. Your instructor will assign you an e-mail address to use. You will use the same e-mail address throughout all the practices in this course.

#### Scenario

In this practice, you will define an event alert that sends a notification to a user each time the user creates a new distribution list in Oracle Alert. Your alert message action needs to include the name of the application to which the distribution list belongs, the name of the list, and all recipients on the list.

#### Step 1: Create the New User

Use the Users window in the System Administrator responsibility to define a new user named ALERTXX. Specify that the new user should be effective from today's date onward. Give the user a description, the password *welcome*, and

the e-mail address assigned to you. Assign the user the following responsibilities:

- Alert Manager
- Application Developer
- General Ledger, Vision Operations (USA)
- System Administrator

Log out of Oracle Applications, and log back in with your new username and password. When you log in for the first time with the new username, you must immediately change your password. Enter *oracle* as your new password.

### Step 2: Research

Use the Record History for the Distribution Lists window in the Alert Manager responsibility to find the table name that you need to code your SQL SELECT statement. Use the Tables window in the Application Developer responsibility to find the column names you need within that table.

### Step 3: Define the Alert

Define an event alert named *XX Distribution List* for the Oracle Alert application, and give the alert a description. Specify the event table and application you found in step 2, and specify that the alert should be triggered after insert on this table but not after updates. Save the alert history for two days. Use the following SELECT statement:

```
SELECT fa.application_name,
       adl.bcc_recipients,
       adl.cc_recipients,
       adl.name,
       :mailid,
       adl.to_recipients
INTO   &application_name,
       &bcc_recipients,
       &cc_recipients,
       &distribution_list_name,
       &mailid,
       &to_recipients
FROM   fnd_application_tl fa,
       alr_distribution_lists adl
WHERE  adl.rowid = :rowid
AND    adl.application_id = fa.application_id
```

**Note:** Because this is an event alert, remember to use the implicit input *:ROWID* in the WHERE clause of your SELECT statement.

After you define your alert, verify your SELECT statement syntax, and review the inputs, outputs, and installations for the alert.

### Step 4: Define the Action



Define a detail action named Send E-mail for the alert, and give the action a description. Specify that the action should send a message to the user who created the new distribution list. Write a subject line and message text for the message. The message text should include the name of the application to which the distribution list belongs, the name of the list, and all recipients on the list.

#### **Step 5: Define the Action Set**

Define an action set named Send E-mail Only for the alert, and give the action set a description. Add your detail action as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

#### **Step 6: Test the Alert**

Define a new distribution list named XX Distribution List for the Oracle Payables application. Give the distribution list a description and enter several To, Cc, and Bcc recipients for the list. The e-mail usernames do not have to be valid, since you will not actually send any messages to this list.

Use your mail system to check your e-mail account and view the message sent by Oracle Alert.

#### **Step 7: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

### **Practice 2-2 Solutions**

#### **Step 1: Create the New User**

- 1 Navigate to the Users window in the System Administrator responsibility.**  
**System Administrator: (N) Security—>User—>Define**
- 2 Enter information to define a new user in the following fields:**
  - **User Name: ALERTXX**
  - **Description: XX New Alert User**
  - **Password: welcome**
  - **Password Expiration: None**
  - **E-Mail: <your e-mail address>**
  - **Effective Dates From: <today's date>**
- 3 Navigate to the Responsibilities region.**
- 4 Select the following responsibilities in the Responsibility fields:**
  - **Alert Manager**
  - **Application Developer**
  - **General Ledger, Vision Operations (USA)**
  - **System Administrator**
- 5 Select today's date in the Effective Dates: From field for each responsibility.**

- 6 Save your work.
- 7 Log out of Oracle Applications, and log back in with your new username and password. When you log in for the first time with the new username, you must immediately change your password. Enter *oracle* as your new password.

### Step 2: Research

- 1 Navigate to the Distribution Lists window in the Alert Manager responsibility.  
Alert Manager: (N) Distribution List
- 2 Select Find All from the View menu to display the distribution list records.
- 3 Select Record History from the Help menu to view information about a record. Note that the table name for these records is ALR\_DISTRIBUTION\_LISTS.
- 4 Navigate to the Tables window in the Application Developer responsibility.  
Application Developer: (N) Application—>Database—>Table
- 5 Query the table name ALR\_DISTRIBUTION\_LISTS in the Oracle Alert application to find the column names you need to define your alert. You will use the following columns:
  - APPLICATION\_ID: The application for which the list was created
  - NAME: The name of the distribution list
  - TO\_RECIPIENTS: The names of the recipients in the TO field of the distribution list
  - CC\_RECIPIENTS: The names of the recipients in the CC field of the distribution list
  - BCC\_RECIPIENTS: The names of the recipients in the BCC field of the distribution list

You will also use the implicit input :MAILID to retrieve the e-mail address of the user who created the list.

### Step 3: Define the Alert

- 1 Navigate to the Alerts window in the Alert Manager responsibility.  
Alert Manager: (N) Alert—>Define
- 2 Enter information to define the alert in the following fields:
  - Application: Oracle Alert
  - Name: XX Distribution List
  - Description: XX Distribution List
  - Enabled: Checked
- 3 Navigate to the Event region.
- 4 Enter the event detail information in the following fields:

- **Application: Oracle Alert**
  - **Table: ALR\_DISTRIBUTION\_LISTS**
  - **After Insert: Checked**
  - **After Update: Not checked**
- 5 Enter 2 in the Keep\_Days field to save history for the alert.**
- 6 Enter the following SELECT statement in the Select Statement field:**

```

SELECT fa.application_name,
       adl.bcc_recipients,
       adl.cc_recipients,
       adl.name,
       :mailid,
       adl.to_recipients
INTO   &application_name,
       &bcc_recipients,
       &cc_recipients,
       &distribution_list_name,
       &mailid,
       &to_recipients
FROM   fnd_application_tl fa,
       alr_distribution_lists adl
WHERE  adl.rowid = :rowid
AND    adl.application_id =
       fa.application_id

```

- 7 Save your work.**
- 8 Click the Verify button to check your SQL for syntax errors.**
- 9 Click the Alert Details button to navigate to the Alert Details window.**
- 10 Navigate to the Inputs region.**  
**Note: This alert does not have any inputs to display because :ROWID and :MAILID are implicit inputs.**
- 11 Navigate to the Outputs region.**
- 12 Select Find All from the View menu to display all the output variables created when you saved your SQL.**
- **APPLICATION\_NAME**
  - **BCC\_RECIPIENTS**
  - **CC\_RECIPIENTS**
  - **DISTRIBUTION\_LIST\_NAME**
  - **MAILID**
  - **TO\_RECIPIENTS**
- 13 Navigate to the Installations region.**

- 14 Select Find All from the View menu to display the schema that Oracle Alert uses to access the tables in the SELECT statement.

#### Step 4: Define the Action

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Enter information to define the action in the following fields:
  - Action Name: Send E-mail
  - Description: Send e-mail notification of new distribution list
  - Action Level: Detail
- 4 Click the Action Details button to navigate to the Action Details window.
- 5 Enter the action detail information in the following fields:
  - Action Type: Message
  - To: &mailid
  - Subject: New distribution list &distribution\_list\_name
- 6 Select the Text option, and enter the following message text:  
A new distribution list called  
&distribution\_list\_name has been created for  
the application &application\_name. The  
recipients are:  
To: &to\_recipients  
Cc: &cc\_recipients  
Bcc: &bcc\_recipients
- 7 Save your work.

#### Step 5: Define the Action Set

- 1 Navigate back to the Alerts window.
- 2 Click the Action Sets button to navigate to the Action Sets window.
- 3 Enter information to define the action set in the following fields:
  - Seq: 1
  - Action Set Name: XX Distribution List Set
  - Description: XX Distribution List Set
  - Enabled: Checked
- 4 Navigate to the Members region.
- 5 Enter information to define the action set member in the following fields:
  - Seq: 1
  - Action: Send E-mail
  - On Error Action: Abort
  - Enabled: Checked
- 6 Save your work.

## **Step 6: Test the Alert**

- 1 Navigate to the Distribution Lists window.**  
**Alert Manager: (N) Distribution List**
- 2 Enter information in the following fields to define the distribution list:**
  - **Application: Oracle Payables**
  - **Name: XX DISTRIBUTION LIST**
  - **Description: XX Distribution List**
- 3 Navigate to the Mail Recipients region.**
- 4 Enter several recipients in the To, Cc and Bcc fields. The e-mail usernames do not have to be valid, since you will not actually send any messages to this list. For example, enter the following recipients:**
  - **To: ahamilton, ajohnson**
  - **Cc: aperkins, asanders**
  - **Bcc: bcochran, berickso**
- 5 Save your work. Oracle Alert submits a concurrent request to create the new distribution list.**
- 6 Navigate to the Requests window.**  
**Alert Manager: (N) Request—>View (B) Find**
- 7 Monitor the request until it is completed.**
- 8 Use your mail system to check your e-mail account and view the message sent by Oracle Alert.**

## **Step 7: Review Alert History**

- 1 Navigate to the Review Alert History window.**  
**Alert Manager: (N) History**
- 2 Select the application and alert name you want to review. In this case, select the following:**
  - **Application: Oracle Alert**
  - **Alert Name: XX Distribution List**
- 3 Click the Find Checks button to navigate to the Review Alert Checks window.**
- 4 Review the information in the Request and Action Set Checks regions. Note the number of exceptions that were found and the number of actions that were executed.**
- 5 Navigate back to the Review Alert History window.**
- 6 Click the Find Exceptions button to navigate to the Display Output Variables window.**
- 7 Review the outputs that will be displayed.**
- 8 Click the OK button to navigate to the Review Alert Exceptions window.**

- 9 Review the exception history for the alert in the Request and Alert Exceptions regions. Note the actual values retrieved by your SELECT statement for the different output variables.**
- 10 Navigate back to the Review Alert History window.**
- 11 Click the Find Actions button to navigate to the Review Alert Actions window.**
- 12 Review the action history for the alert in the Request and Alert Actions regions. Verify that the action you defined was executed.**
- 13 Click the Action Log button to navigate to the Alert Action Log window. For a message action, Oracle Alert displays the message that was sent in the action log.**
- 14 Review the message, and verify that the output variable names were replaced with their values.**
- 15 Click the OK button to close the action log.**

## Summary

---

### Summary

- Event alert conditions are detected by the alert when an insert or update is done to the specified database table.
- Implicit input values are supplied by Oracle Alert.
- The SQL SELECT statement defines your alert condition and retrieves all information needed to perform the action you define.
- A detail action performs one action for each exception found; a summary action groups exceptions; a no exception action performs an action defined when no exceptions are found.
- All actions are put in an action set to be performed.
- Alert history is used to review actions performed by Oracle Alert.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Lesson Summary

In this lesson, you learned to define alert conditions using SQL SELECT statements. Implicit inputs can be used to set up the conditions. You define actions to occur when the alert conditions are found. Place all actions that you define in an action set that you test by triggering the event that would find the condition defined. Alert history shows you the results of your alert actions.





# Defining a Periodic Alert

## Chapter 3

## Defining a Periodic Alert

---

### Defining a Periodic Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Objectives

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

- Define periodic alert conditions
- Determine a periodic alert schedule
- Create No Exception alert actions
- Check a periodic alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Business Problem

Define a periodic alert to solve the following business problem:

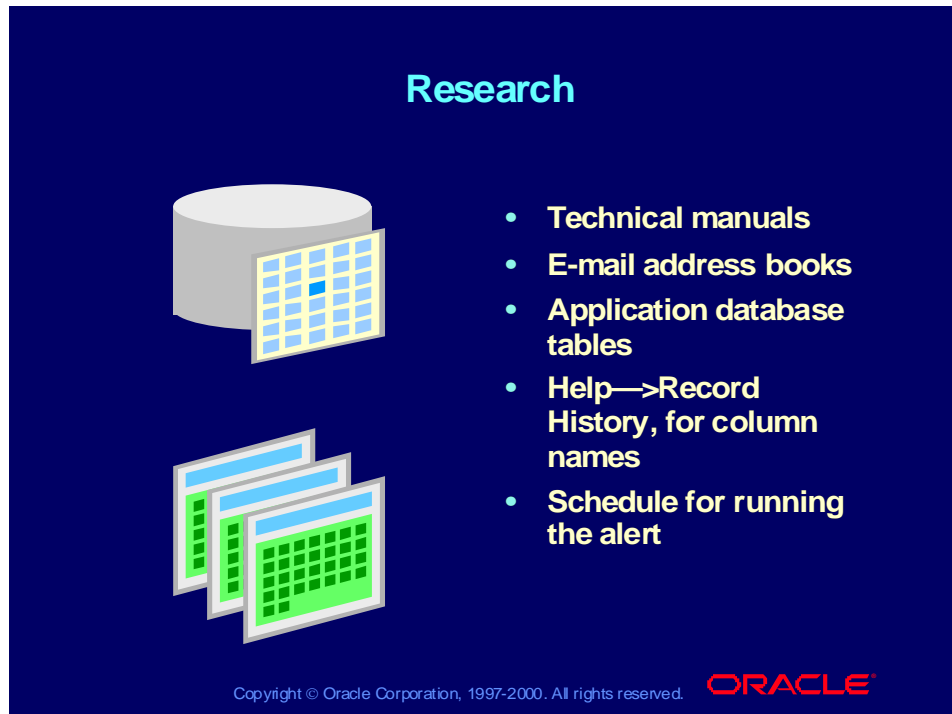
- **Problem**—Every week, management needs to know about all purchases over \$50,000.
- **Solution**—Create a periodic alert:
  - Send a message to management listing all purchase orders created that week where the total dollar amount is \$50,000 or more.
  - Send a message to management when no purchase orders exceed \$50,000.
  - Allow management to change the dollar amount threshold.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Research

---



**Research**

- Technical manuals
- E-mail address books
- Application database tables
- Help—>Record History, for column names
- Schedule for running the alert

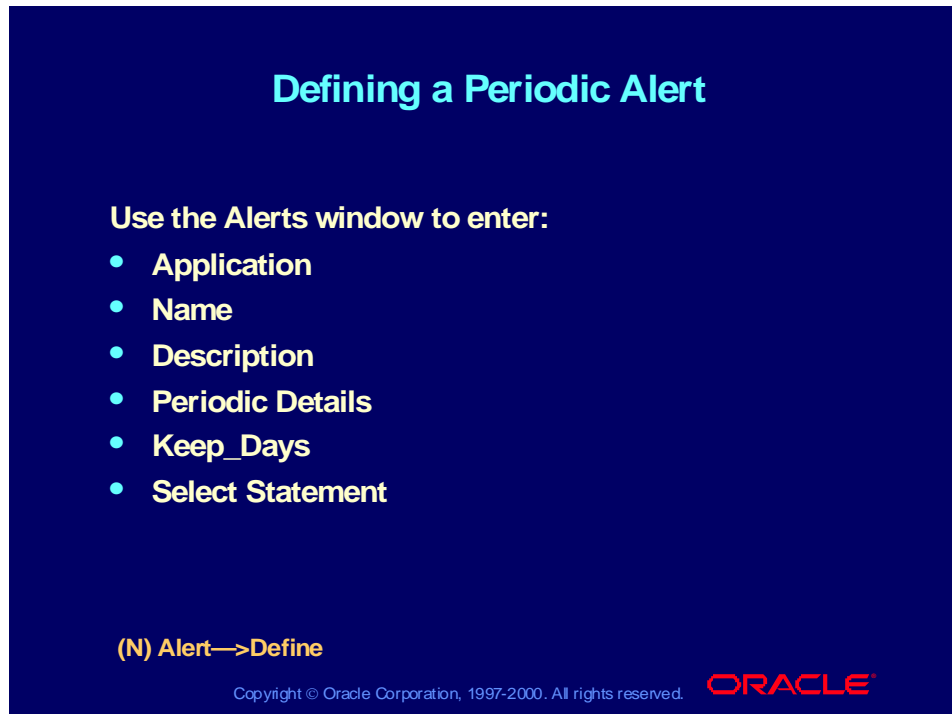
Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Research

- Research all the applications and tables you will need to access to define the alert condition.
- Determine what the alert actions should actually do.
- Determine what any message actions should say.
- Look up the e-mail addresses or distribution list names for message actions that you will define.
- Determine when you want this alert to be performed.
- Refer to any technical manuals needed to help you build your SQL SELECT statement or definition for the alert.

## Defining a Periodic Alert

---



**Defining a Periodic Alert**

Use the Alerts window to enter:

- Application
- Name
- Description
- Periodic Details
- Keep\_Days
- Select Statement

(N) Alert—>Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Defining a Periodic Alert

Choose the Periodic region to specify that you are defining a periodic alert and enter periodic details.

**Note:** By saving history for your alerts, you can define Oracle Alert to check for duplicates—exceptions that existed in your database during a previous alert check. You can define Oracle Alert to perform alert actions when it finds these duplicate exceptions in your database.

### Defining Periodic Details

Use the **Periodic** region of the Alerts window to enter periodic details:

- Frequency
- Start Time
- End Time

(N) Alert—>Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved.


ORACLE

## Predefined Frequencies

---

**Predefined Frequencies**

- On demand
- On day of the month
- On day of the week
- Every *N* calendar days
- Every *N* business days
- Every day
- Every other day
- Every business day
- Every other business day



Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Predefined Frequencies

To run an alert on the last day of the month, specify it as the 31st of the month. If there are fewer days in the month, the alert will run on the last day of the month, whichever date that is.



## Writing the SQL SELECT Statement

---

**Writing the SQL SELECT Statement**

```
SELECT user_name,  
       password_date,  
       :threshold_days  
INTO   &user,  
       &lastday,  
       &numdays  
FROM   fnd_user  
WHERE  sysdate = NVL(password_date,  
                     sysdate) + :threshold_days  
ORDER BY user_name
```

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Writing the SQL SELECT Statement

The SQL SELECT statement retrieves all the data your alert needs to perform the actions you define. Your periodic alert SELECT statement must include an INTO clause that contains one output for each column selected by your SELECT statement. By default, a sample SELECT statement is in the Select Statement field. You may type over it or use parts of it while you build your own SELECT statement.

## Verifying and Running the SQL SELECT Statement

---

### Verifying and Running the SQL SELECT Statement

Use the Alerts window to test your SQL SELECT statement.

- Click the Verify button to verify the SQL syntax. The parsing of your SELECT statement results is displayed in a Note window.
- Click the Run button to run the SQL statement outside Oracle Alert using one of your application's Oracle IDs. The number of rows returned by the SQL statement is displayed in a Note window.

(N) Alert—>Define (B) Verify

(N) Alert—>Define (B) Run

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Specifying Alert Inputs

---

### Specifying Alert Inputs

Use the **Inputs** region of the **Alert Details** window to find the inputs from your **SELECT** statement and enter:

- **Description**
- **Data Type**
- **Default Values**

**(N) Alert** → **Define (B) Alert Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining Alert Outputs

Use the **Outputs** region of the **Alert Details** window to find the outputs from your **SELECT** statement and enter:

- **Description**
- **Max Length - Detail**
- **Max Length - Summary**
- **Number Format**
- **Check For Duplicates**

**(N) Alert** → **Define (B) Alert Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining a Periodic Alert Action

Use the Actions window to enter:

- Action Name
- Description
- Action Level
  - Detail
  - Summary
  - No Exception

**(N) Alert—>Define (B) Actions**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining Action Details

Use the Action Details window to select an action type:

- Message
- Concurrent Program
- Operating System Script
- SQL Statement Script

Oracle Alert displays only those fields that are relevant to that action type.

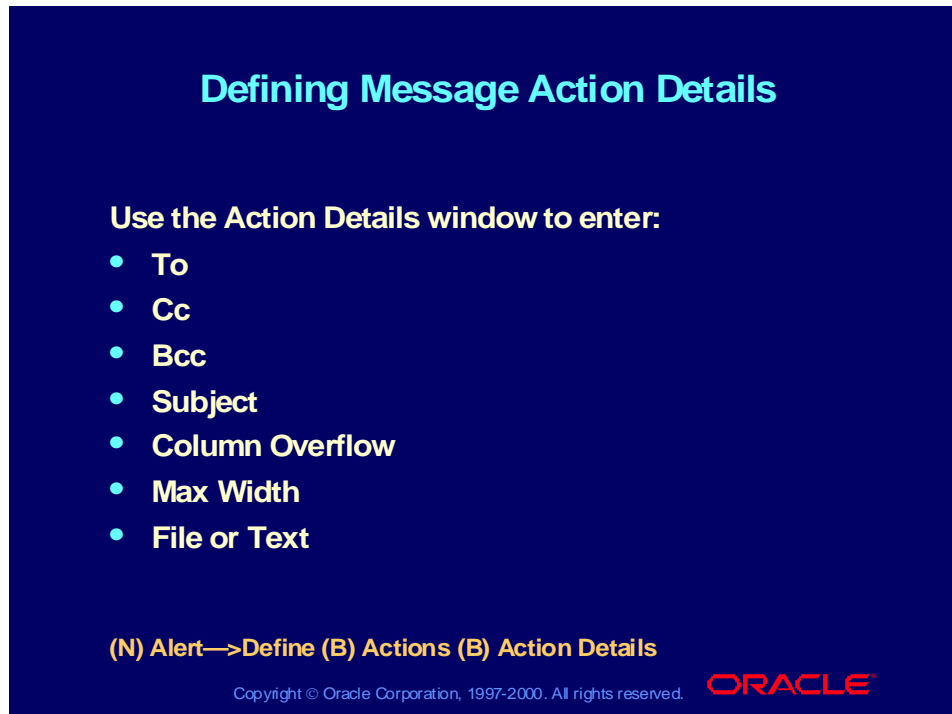
(N) Alert → Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Defining Message Action Details

---



**Defining Message Action Details**

Use the Action Details window to enter:

- To
- Cc
- Bcc
- Subject
- Column Overflow
- Max Width
- File or Text

(N) Alert—>Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Defining Message Action Details

You can enter a number in the Max Width field to define the width of the message as a whole. This is one of two factors that can affect the length of the message. The other factor is the Max Length field, defined in the Outputs region of the Alert Details window.

**Note:** Do not include any outputs in a No Exception message action definition.

### How to Create Exception Messages with Multiple Lines

- 1 To create constant text, type within the template where it says “Enter summary template below this line.”  
**Note:** Make the spacing exact. Be sure you know the size of the output fields and adjust your entry to fit the field.
- 2 Enter critical outputs outside the template.  
**Note:** If your critical output is an e-mail username, a separate e-mail message is sent for each recipient.

## Defining a Periodic Alert Action Set

---

### Defining a Periodic Alert Action Set

Use the Action Sets window to enter:

- Seq
- Action Set Name
- Description
- Suppress Duplicates
- Enabled
- End Date

(N) Alert → Define (B) Action Sets

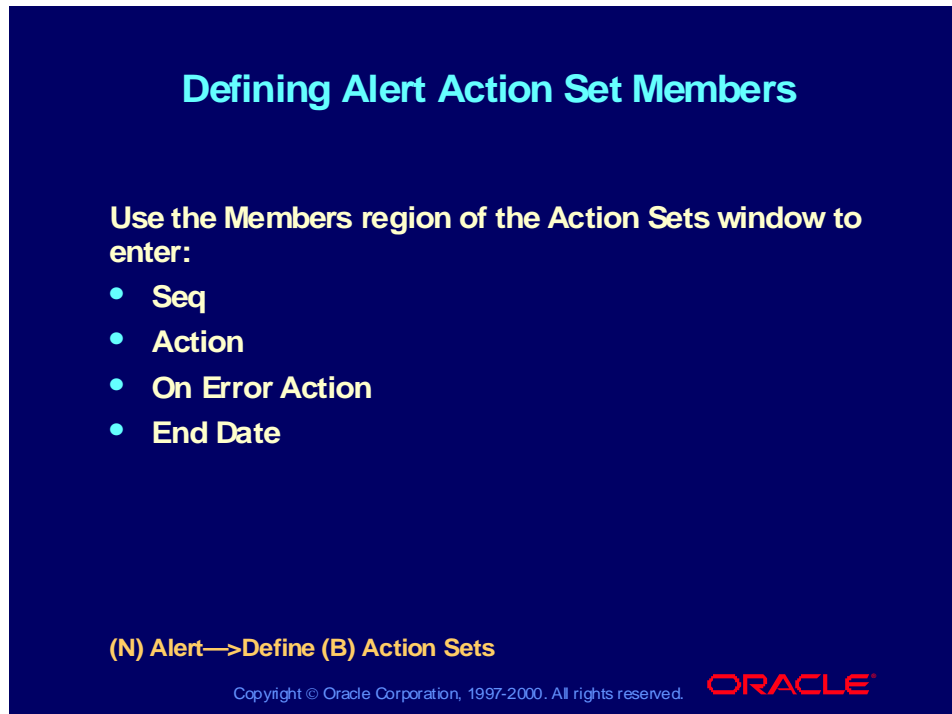
Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Defining Alert Action Set Members

---



**Defining Alert Action Set Members**

Use the **Members** region of the **Action Sets** window to enter:

- **Seq**
- **Action**
- **On Error Action**
- **End Date**

(N) Alert → Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining Alert Action Set Members

**Note:** When you add a member to your action set, Oracle Alert automatically indicates in the **Type** field whether the member is an action, escalation group, or threshold group, and what its action type is (e-mail message, concurrent request, operating system script, or SQL statement script). Oracle Alert also indicates whether the member is enabled.

### Modifying Action Set Inputs

Use the Value field in the Inputs region of the Action Sets window to enter specific input values for the action set that override the default values.

- You can customize the SQL SELECT statement by changing the input values for an action set instead of changing the coding.
- When the inputs are selected into outputs, changing the input values in each action set also changes the output values, allowing you to send messages containing different information.

(N) Alert → Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Checking a Periodic Alert

---

**Checking a Periodic Alert**

Use the Request Periodic Alert Check window to enter:

- Application
- Alert
- Start Date

(N) Request—>Check

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Checking a Periodic Alert

Test your periodic alert by submitting the alert to run in the Request Periodic Alert Check window.

**Note:** After you click the Submit Request button, a request ID appears for the request. Use this request ID to monitor the status of the periodic alert check in the Concurrent Request Summary window. If the periodic alert is not an on-demand periodic alert, it also continues to run according to its defined schedule.

### Reviewing Alert History

Review the alert history for a periodic alert using the same procedures as you use for an event alert.

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Documenting the Alert

**Document your alert so that others can understand how it works and what it does.**

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Practice 3-1

This practice covers the following topics:

- Defining periodic alert conditions
- Identifying differences between event and periodic alerts
- Dealing with exceptions by limiting the exceptions found by the alert and allowing them to be changed as needed

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 3-1 Instructions

Discuss the following:

- 1 How does a periodic alert differ from an event alert?
- 2 What periodic frequency do you use to check on the last day of the month?
- 3 When does a No Exception action perform?
- 4 What are the steps to create a periodic alert?
- 5 Can you use the periodic check window for event alert checking?
- 6 How do you allow management to change its mind about specific exceptions?
- 7 What are six areas of an alert that you can review using history windows?

### Practice 3-1 Solutions

- 1 How does a periodic alert differ from an event alert?

**Periodic alerts check your database for predefined conditions according to a schedule you determine. If you need an alert that runs on a schedule, use a periodic alert. Event alerts monitor your applications for the exception conditions you specify at any time. You do not schedule an event alert.**

- 2 What periodic frequency do you use to check on the last day of the month?  
**If you want your alert checked on the last day of the month, enter the number 31. Oracle Alert checks the alert on the 28th, 29th, 30th, or 31st, whichever is the actual last day of the month.**
- 3 When does a No Exception action perform?  
**When no exceptions are found for a limit amount you specify, this action performs.**
- 4 What are the steps to create a periodic alert?  
**Same steps as an event alert:**  
**Research—In this step determine the schedule for performing this alert.**  
**Define the alert.**  
**Define the actions.**  
**Define the action set.**  
**Run and test the alert.**  
**Review the alert history.**  
**Document the alert.**
- 5 Can you use the periodic check window for event alert checking?  
**No.**
- 6 How do you allow management to change its mind about specific exceptions?  
**Change the value of the input variable.**
- 7 What are six areas of an alert that you can review using history windows?  
**Review the alert checks.**  
**Display the output variables.**  
**Review the alert exceptions.**  
**Review the alert actions.**  
**Review the alert action log.**  
**Review the response history.**

### Practice 3-2

This practice covers the following topics:

- Creating an on-demand alert that sends a message depending on an exception condition
- Formatting messages
- Creating No Exception alert actions
- Creating summary messages
- Checking and testing the alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 3-2 Instructions

#### Scenario

In this practice, you will create an on-demand periodic alert for Oracle Payables that sends you a summary message listing all the invoices that are on active hold for each vendor. For each invoice on hold, the message should show the vendor name, invoice number, hold type, and hold reason.

**Note:** A hold is active only if the value in the `RELEASE_LOOKUP_CODE` column in the `AP_HOLDS` table is null for the invoice that is on hold. You will need the following tables and columns for this alert:

- `PO_VENDORS`: `VENDOR_ID`, `VENDOR_NAME`
- `AP_INVOICES`: `INVOICE_ID`, `INVOICE_NUM`, `VENDOR_ID`
- `AP_HOLDS`: `INVOICE_ID`, `HOLD_LOOKUP_CODE`, `RELEASE_LOOKUP_CODE`, `HOLD_REASON`

#### Step 1: Define the Alert

Define a periodic alert named *XX Invoices on Hold* for the Oracle Payables application, and give the alert a description. Specify that the alert should be run on demand. Save the alert history for two days. Use the following `SELECT` statement:

```
SELECT    aph.hold_reason,
          aph.hold_lookup_code,
          api.invoice_num,
```



```

        pov.vendor_name
INTO      &hold_reason,
        &hold_type,
        &invoice_number,
        &vendor_name
FROM      po_vendors pov,
        ap_invoices api,
        ap_holds aph,
        hr_organization_units hou
WHERE     hou.name = 'Vision Operations'
AND       hou.organization_id = aph.org_id
AND       aph.release_lookup_code IS NULL
AND       aph.invoice_id = api.invoice_id
AND       api.vendor_id = pov.vendor_id
ORDER BY pov.vendor_name

```

After you define your alert, verify your SELECT statement syntax, and run the SELECT statement to check how many records it retrieves. Specify that the maximum length in the summary for the hold reason output is 25 and the maximum length in the summary for the hold type is 20.

### Step 2: Define the Actions

Define two actions:

- Define a summary action named Invoices on Hold E-mail Notification for the alert, and give the action a description. Specify that the action should send a message to you. Write a subject line and message text for the message. The message text should include the vendor name as well as the invoice number, hold type, and hold reason for each invoice on hold.  
**Note:** Place the critical output outside the summary template area of the message to create a separate summary message for each vendor.
- Define a no exception action named No Invoice on Hold E-mail Notification for the alert, and give the action a description. Specify that the action should send a message to you. Write a subject line and message text for the message. The message text should indicate that no invoices are on hold.

### Step 3: Define the Action Set

Define an action set named E-mails for the alert, and give the action set a description. Add your summary action and your no exception action as members of the action set, and specify that the actions should be aborted if errors occur during execution.

### Step 4: Test the Alert

Submit a periodic alert check for your alert. Use your mail system to check your e-mail account and view the message sent by Oracle Alert.

### Step 5: Review Alert History

Review the history for your alert, including checks, exceptions and actions.

## Practice 3-2 Solutions

### Step 1: Define the Alert

- 1 **Navigate to the Alerts window in the Alert Manager responsibility.**  
**Alert Manager: (N) Alert—>Define**
- 2 **Enter information to define the alert in the following fields:**
  - **Application: Oracle Payables**
  - **Name: XX Invoices on Hold**
  - **Description: XX Invoices on Hold**
  - **Enabled: Checked**
- 3 **Navigate to the Periodic region.**
- 4 **Enter the periodic detail information in the following field:**
  - **Frequency: On Demand**
- 5 **Enter 2 in the Keep\_Days field to save history for the alert.**
- 6 **Enter the following SELECT statement in the Select Statement field:**

```
SELECT    aph.hold_reason,
          aph.hold_lookup_code,
          api.invoice_num,
          pov.vendor_name
INTO      &hold_reason,
          &hold_type,
          &invoice_number,
          &vendor_name
FROM      po_vendors pov,
          ap_invoices api,
          ap_holds aph,
          hr_organization_units hou
WHERE     hou.name = 'Vision Operations'
AND       hou.organization_id = aph.org_id
AND       aph.release_lookup_code IS NULL
AND       aph.invoice_id = api.invoice_id
AND       api.vendor_id = pov.vendor_id
ORDER BY pov.vendor_name
```
- 7 **Save your work.**
- 8 **Click the Verify button to check your SQL for syntax errors.**
- 9 **Click the Run button to check how many records are retrieved by your SELECT statement.**
- 10 **Click the Alert Details button to navigate to the Alert Details window.**

- 11 Navigate to the Outputs region.
- 12 Select Find All from the View menu to display the output variables created when you saved your SQL.
- 13 Enter 25 in the Max Length Summary field for the HOLD\_REASON output.
- 14 Enter 20 in the Max Length Summary field for the HOLD\_TYPE output.
- 15 Save your work.

**Step 2: Define the Actions**

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Enter information to define the first action in the following fields:
  - Action Name: Invoices on Hold E-mail Notification
  - Description: Summary e-mail notification of invoices on hold
  - Action Level: Summary
- 4 Click the Action Details button to navigate to the Action Details window.
- 5 Enter the action detail information in the following fields:
  - Action Type: Message
  - To: <your e-mail address>
  - Subject: Invoices on active hold for &vendor\_name
- 6 Select the Text option, and enter the following message text:
 

```
The following invoices are on hold for vendor
&vendor_name:

    Invoice Number  Hold Type  Hold Reason
=== Enter summary template below this line ===
**      &invoice_number &hold_type &hold_reason
=== Enter summary template above this line ===
```

Note: Since vendor\_name is placed outside of the summary template area, Oracle Alert will consider vendor\_name a critical output and send a separate e-mail message for each vendor.
- 7 Ensure that the Column Overflow field is set to Wrap and the Max Width field is set to 80.
- 8 Save your work.
- 9 Navigate back to the Actions window.
- 10 Enter information to define the second action in the following fields:
  - Action Name: No Invoice on Hold E-mail Notification
  - Description: E-mail notification of no invoices on hold
  - Action Level: No Exception
- 11 Click the Action Details button to navigate to the Action Details window.

**12 Enter the action detail information in the following fields:**

- **Action Type: Message**
- **To: <your e-mail address>**
- **Subject: On Hold Invoice Status**

**13 Select the Text option, and enter the following message text:**

**No invoices are currently on hold.**

**14 Save your work.**

**Step 3: Define the Action Set**

**1 Navigate back to the Alerts window.**

**2 Click the Action Sets button to navigate to the Action Sets window.**

**3 Enter information to define the action set in the following fields:**

- **Seq: 1**
- **Action Set Name: E-mails**
- **Description: E-mail notifications**
- **Enabled: Checked**

**4 Navigate to the Members region.**

**5 Enter information to define the first action set member in the following fields:**

- **Seq: 1**
- **Action: Invoices on Hold E-mail Notification**
- **On Error Action: Abort**
- **Enabled: Checked**

**6 Enter information to define the second action set member in the following fields:**

- **Seq: 2**
- **Action: No Invoice on Hold E-mail Notification**
- **On Error Action: Abort**
- **Enabled: Checked**

**7 Save your work.**

**Step 4: Test the Alert**

**1 Navigate to the Request Periodic Alert Check window.**

**Alert Manager: (N) Request—>Check**

**2 Select the application and alert name you want to check. In this case, select the following:**

- **Application: Oracle Payables**
- **Alert Name: XX Invoices on Hold**

**3 Click the Submit Request button to run the check. A Note window appears to inform you when the periodic alert is scheduled to run.**

**4 Click the OK button to close the Note window.**

- 5 Note your request ID in the Request Periodic Alert Check window.
- 6 Navigate to the Requests window.  
Alert Manager: (N) Request—>View (B) Find
- 7 Monitor the request until it is completed.
- 8 Use your mail system to check your e-mail account and view the messages sent by Oracle Alert.

**Step 5: Review Alert History**

After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:

- Application: Oracle Payables
- Alert Name: XX Invoices on Hold

### Practice 3-3

This practice covers the following topics:

- Defining an input variable for an alert
- Modifying the input variable amount to retrieve different information

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 3-3 Instructions

#### Scenario

In this practice, you will create an on-demand periodic alert for Oracle Payables that sends you a summary message listing all the invoices that are on active hold. This alert will include an input variable that lets you limit the number of invoices retrieved by specifying a minimum invoice amount for the invoices that you want to see. After you create and test the alert with a default input value, you will modify the alert action set to specify a different input value for that set, and test the results of this change.

#### Step 1: Define the Alert

Define a periodic alert named *XX Invoices on Hold with Input Variable* for the Oracle Payables application, and give the alert a description. Specify that the alert should be run on demand. Save the alert history for two days. Use the following SELECT statement:

```
SELECT      aph.hold_reason,
            aph.hold_lookup_code,
            api.invoice_amount,
            api.invoice_num,
            :limit_amount,
            pov.vendor_name
INTO        &hold_reason,
```

---

Copyright © Oracle Corporation, 2000. All rights reserved.

```

        &hold_type,
        &invoice_amount,
        &invoice_number,
        &limit_amount,
        &vendor_name
FROM      po_vendors pov,
        ap_invoices_all api,
        ap_holds_all aph,
        hr_organization_units hou
WHERE     hou.name = 'Vision Operations'
AND      hou.organization_id = aph.org_id
AND      aph.release_lookup_code IS NULL
AND      api.invoice_amount >=
        :limit_amount
AND      aph.invoice_id = api.invoice_id
AND      api.vendor_id = pov.vendor_id
ORDER BY pov.vendor_name

```

After you define your alert, verify your SELECT statement syntax. Specify that the default value for the LIMIT\_AMOUNT variable is 0.00. Run the SELECT statement to check how many records it retrieves.

### **Step 2: Define the Action**

Define a summary action named On Hold Limit E-mail Notification for the alert, and give the action a description. Specify that the action should send a message to you. Write a subject line and message text for the message. The message text should include the vendor name as well as the invoice number, invoice amount, hold type, and hold reason for each invoice on hold.

### **Step 3: Define the Action Set**

Define an action set named E-mail for the alert, and give the action set a description. Add your summary action as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

### **Step 4: Test the Alert**

Submit a periodic alert check for your alert. Use your mail system to check your e-mail account and view the message sent by Oracle Alert.

### **Step 5: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

### **Step 6: Modify the Input for the Action Set**

Modify your action set to specify a value of 5000.00 for the LIMIT\_AMOUNT input. Then repeat steps 4 and 5 to view the results of the change.

## **Practice 3-3 Solutions**

## Step 1: Define the Alert

- 1 Navigate to the Alerts window in the Alert Manager responsibility.  
Alert Manager: (N) Alert—>Define
- 2 Enter information to define the alert in the following fields:
  - Application: Oracle Payables
  - Name: *XX Invoices on Hold with Input Variable*
  - Description: *XX Invoices on Hold with Input Variable - lets user change retrieval limit amount*
  - Enabled: Checked
- 3 Navigate to the Periodic region.
- 4 Enter the periodic detail information in the following field:
  - Frequency: On Demand
- 5 Enter 2 in the Keep\_Days field to save history for the alert.
- 6 Enter the following SELECT statement in the Select Statement field:

```
SELECT      aph.hold_reason,
            aph.hold_lookup_code,
            api.invoice_amount,
            api.invoice_num,
            :limit_amount,
            pov.vendor_name
INTO        &hold_reason,
            &hold_type,
            &invoice_amount,
            &invoice_number,
            &limit_amount,
            &vendor_name
FROM        po_vendors pov,
            ap_invoices_all api,
            ap_holds_all aph,
            hr_organization_units hou
WHERE       hou.name =
            'Vision Operations'
AND        hou.organization_id =
            aph.org_id
AND        aph.release_lookup_code IS
            NULL
AND        api.invoice_amount >=
            :limit_amount
AND        aph.invoice_id =
```



```

                                api.invoice_id
AND                                api.vendor_id = pov.vendor_id
ORDER BY                            pov.vendor_name

```

- 7 Save your work.
- 8 Click the Verify button to check your SQL for syntax errors.
- 9 Click the Alert Details button to navigate to the Alert Details window.
- 10 Navigate to the Inputs region.
- 11 Select Find All from the View menu to display the inputs.
- 12 Enter 0.00 in the Default Values field for the LIMIT\_AMOUNT input.
- 13 Save your work.
- 14 Navigate back to the Alerts window.
- 15 Click the Run button to check how many records are retrieved by your SELECT statement.

#### Step 2: Define the Action

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Enter information to define the action in the following fields:
  - Action Name: On Hold Limit E-mail Notification
  - Description: Summary e-mail notification of invoices on hold with retrieval limit amount
  - Action Level: Summary
- 4 Click the Action Details button to navigate to the Action Details window.
- 5 Enter the action detail information in the following fields:
  - Action Type: Message
  - To: <your e-mail address>
  - Subject: Invoices on hold over &limit\_amount
- 6 Select the Text option, and enter the following message text:
 

```

The following invoices over &limit_amount are on
active hold:
***= Enter summary template below this line ***=
**  Vendor name:      &vendor_name
    Invoice number:   &invoice_number
    Invoice amount:   &invoice_amount
    Hold type:       &hold_type
    Hold reason:     &hold_reason
***= Enter summary template above this line ***=

```
- 7 Save your work.

#### Step 3: Define the Action Set

- 1 Navigate back to the Alerts window.

- 2 Click the **Action Sets** button to navigate to the **Action Sets** window.
- 3 Enter information to define the action set in the following fields:
  - **Seq: 1**
  - **Action Set Name: E-mail**
  - **Description: E-mail notification**
  - **Enabled: Checked**
- 4 Navigate to the **Members** region.
- 5 Enter information to define the action set member in the following fields:
  - **Seq: 1**
  - **Action: On Hold Limit E-mail Notification**
  - **On Error Action: Abort**
  - **Enabled: Checked**
- 6 Save your work.

#### **Step 4: Test the Alert**

- 1 Navigate to the **Request Periodic Alert Check** window.  
**Alert Manager: (N) Request—>Check**
- 2 Select the application and alert name you want to check. In this case, select the following:
  - **Application: Oracle Payables**
  - **Alert Name: XX Invoices on Hold with Input Variable**
- 3 Click the **Submit Request** button to run the check. A **Note** window appears to inform you when the periodic alert is scheduled to run.
- 4 Click the **OK** button to close the **Note** window.
- 5 Note your request ID in the **Request Periodic Alert Check** window.
- 6 Navigate to the **Requests** window.  
**Alert Manager: (N) Request—>View (B) Find**
- 7 Monitor the request until it is completed.
- 8 Use your mail system to check your e-mail account and view the message sent by **Oracle Alert**.

#### **Step 5: Review Alert History**

After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:

- **Application: Oracle Payables**
- **Alert Name: XX Invoices on Hold with Input Variable**

#### **Step 6: Modify the Input for the Action Set**

- 1 Navigate back to the **Alerts** window.
- 2 Click the **Action Sets** button to navigate to the **Action Sets** window.
- 3 Select the **E-mails** action set you defined in step 3.

- 4 Navigate to the Inputs region.**
- 5 Enter 5000.00 in the Value field for the LIMIT\_AMOUNT input.**
- 6 Save your work.**
- 7 Repeat steps 4 and 5 to view the results of modifying the input value for the action set.**

## Summary

---

### Summary

- A periodic alert checks your database for exceptions according to a schedule you define.
- A No Exception message performs when no exceptions are found.
- When defining messages, the critical outputs are inserted outside of the summary template.
- Summary information is placed inside the message's summary template.
- A periodic alert can be submitted using the Request Periodic Alert Check window.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Lesson Summary

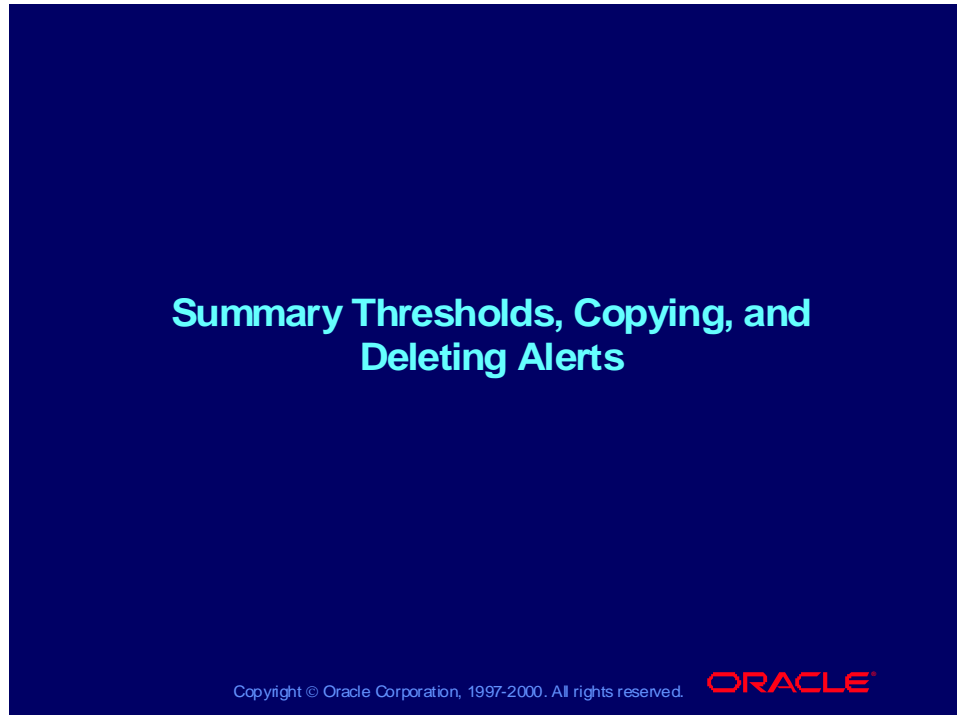
In this lesson, you learned how to define a periodic alert to perform on a defined day or on demand. Input values can be modified, and output lengths and values can be modified in the Inputs and Outputs regions of the Action Sets window. Action details can be defined using exception messages with critical outputs. The placement of the message in the message template field has an effect on the e-mail message and how it is processed.

# **Summary Thresholds, Copying, and Deleting Alerts**

## **Chapter 4**

## Summary Thresholds, Copying, and Deleting Alerts

---



### Objectives

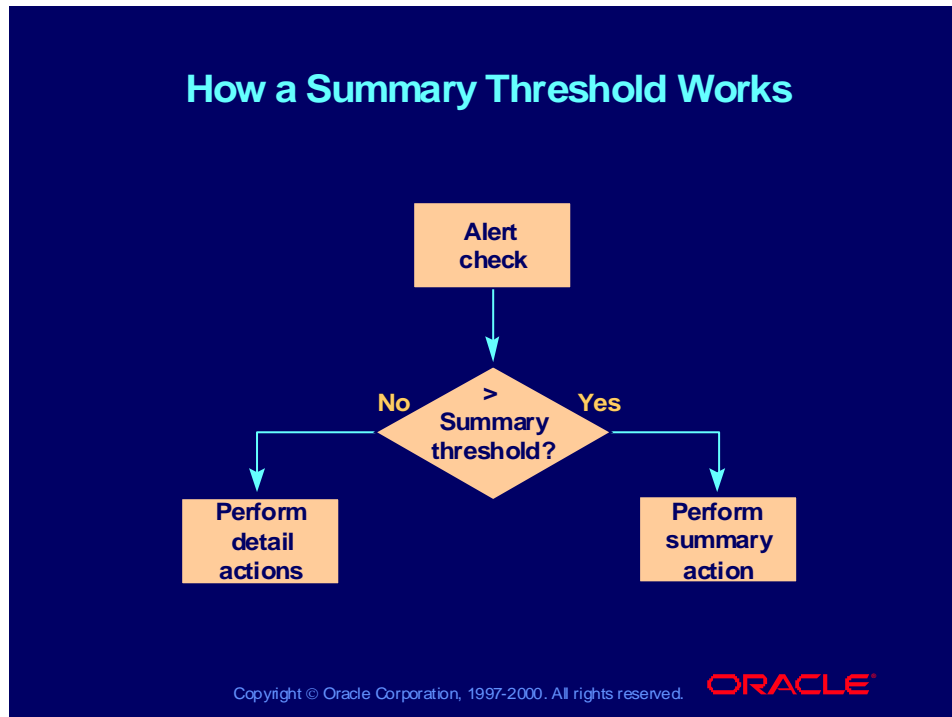
After completing this lesson, you should be able to do the following:

- Define different action levels based on the number of exceptions found using a summary threshold
- Test a summary threshold alert
- Copy alerts across applications
- Describe how to delete alerts

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

## How a Summary Threshold Works

---



### Summary Thresholds

A summary threshold allows you to define when the alert should perform summary actions versus detail actions. Once you define a summary threshold, Oracle Alert can determine whether to perform detail or summary actions for an alert. If the number of exceptions for an alert exceeds the summary threshold value that you define, Oracle Alert performs a summary action; otherwise, it performs a detail action for each exception found.



### Business Problem

Define an alert to solve the following business problem:

- **Problem**—The purchasing manager needs to be notified of overdue items, but wants a summary of the items if more than one is overdue.
- **Solution**—Define an alert and set the summary threshold to 1.

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

## Creating a Summary Threshold Alert

---

### Creating a Summary Threshold Alert

To use a summary threshold for an alert, you must:

- Define the alert
- Define one detail action for the alert
- Define one summary action for the alert
- Add these two actions to a threshold group
- Add the threshold group as a member of the action set for the alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Creating a Summary Threshold Alert

#### Example

- Create a detail message action that sends an overdue item to the requestor.
- Create a summary message action that sends all overdue items to the requestor in one message.

You will add these two actions to a threshold group. You can define either message, SQL statement script, operating system script, or concurrent program request actions, but both the detail action and the summary action must be of the same action type.

## Defining a Threshold Group

---



**Defining a Threshold Group**

Use the Threshold Groups window to enter:

- Group Name
- Description
- Type
- Threshold Group Members

(N) Alert—>Define—>(M) Tools—>Threshold Groups

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining a Threshold Group

Define the threshold group in the Threshold Groups window. In the Action column of the Threshold Group Members region, select one detail action and one summary action from the list of values as members of the group. You can select only actions created for this alert with the same action type as the group.

## Adding a Threshold Group to an Action Set

---

### Adding a Threshold Group to an Action Set

Use the Action Sets window to define an action set. In the Members region, enter:

- **Action:** Select the threshold group name.
- **Summary Threshold:** Enter the summary threshold value.

(N) Alert—>Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Testing and Running the Alert

---

### Testing and Running the Alert

- Request an alert check to test your periodic alert, or perform the necessary actions to trigger your event alert
- Review the alert history, including checks, actions, and exceptions
- Document the alert



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Checking Actions and Exceptions

---

### Checking Actions and Exceptions

- If you have a summary threshold of 3, and 3 exceptions are found, then 3 detail actions will be performed, 1 for each exception.
- If you have a summary threshold of 3, and 4 exceptions are found, 1 summary action will be performed with 4 exceptions listed.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Copying Alert Definitions

---

### Copying Alert Definitions

Use the Transfer Alert Definitions window to enter:

- **Source Alert**
  - Application
  - Alert
  - Database
- **Destination Alert**
  - Application
  - Alert
  - Database

**(N) Alert**→**Define**→**(M) Tools**→**Transfer Alert**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Copying Alerts

- You can transfer an alert definition to another database or make a copy of an existing alert in the same database.
- You must enable the copied alert before you can use it. Find the Alert in the Alerts window and check the Enabled check box.

(N) Alert—>Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

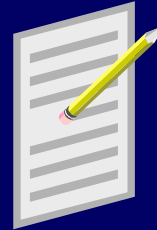


## Deleting Alert Definitions

---

### Deleting Alert Definitions

- You can delete alert definitions in the Alerts window.
- You can delete an alert only if the alert has no history, no enabled action sets, and no outstanding responses.
- If you delete an alert that has actions or response sets defined but no enabled action sets, Oracle Alert deletes the alert and its associated actions and response sets.



(N) Alert—>Define—>(M) Edit—>Delete

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 4-1

This practice covers the following topics:

- Understanding the uses of a summary threshold
- Understanding how to delete an alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 4-1 Instructions

Answer the following questions:

- 1 Why do you use a summary threshold alert?
- 2 Assuming that you implement summary thresholds, what does the summary threshold = 5 do when five exceptions are found?
- 3 What new steps are needed to create an alert with a summary threshold?
- 4 What do you have to do to use an alert that you copy?
- 5 How do you delete an alert?

### Practice 4-1 Solutions

- 1 Why do you use a summary threshold alert?

**To have Oracle Alert perform actions based on a single exception or a combination of exceptions found in your database. You can define a detail action so that Oracle Alert performs that action for each individual exception found. You can also define a summary action so that Oracle Alert performs that action once for each unique combination of exceptions found. You decide which exceptions you want Oracle Alert to consider as a unique combination. You can format a detail or summary message action to display the exception(s) in an easy-to-read message.**

- 2 Assuming that you implement summary threshold, what does the summary threshold = 5 do when five exceptions are found?  
**It performs detail actions for five or less exceptions found. It would perform a summary action for more than five exceptions found.**
- 3 What new steps are needed to create an alert with summary threshold?  
**Define a threshold group that contains the detail and summary messages. Define an action set with the threshold group as a member and give it a summary threshold value.**
- 4 What do you have to do to use an alert that you copy?  
**Click the Enabled check box to use it.**
- 5 How do you delete an alert?  
**Query the alert you want to delete in the Alerts window. Select Edit—>Delete. You can delete an alert only if no history, no enabled action sets, and no outstanding responses exist for the alert. If you delete an alert that has actions or response sets defined but no enabled action sets, Oracle Alert deletes the alert, as well as the associated actions and response sets. Save your work.**

### Practice 4-2

This practice covers the following topics:

- Copying an alert
- Using a summary threshold
- Testing a summary threshold

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 4-2 Instructions

#### Scenario

In this practice, you will copy an existing alert and modify it to create a new alert. You will use your first Invoices on Hold alert as the source alert for the copy. For the new alert, you will define a threshold group with a summary action and a detail action, as well as a summary threshold to specify how many exceptions Oracle Alert can find before it switches from performing the detail action to performing the summary action.

#### Step 1: Copy the Alert

Copy the XX Invoices on Hold alert that you defined in Practice 3-2. Name the new alert XX Threshold Invoices on Hold and enable this alert.

#### Step 2: Define the Actions

Revise the existing summary action named Invoices on Hold E-mail Notification, so that the subject line does not include the &vendor\_name output and the text of the message includes the &vendor\_name output inside the summary template rather than outside the summary template.

Define a detail action named Detail Message for the alert, and give the action a description. Specify that the action should send a message to you. Write a subject line and message text for the message. The message text should include the vendor name, invoice number, hold type, and hold reason for the invoice on hold.

### **Step 3: Define the Threshold Group**

Define a threshold group named E-mail Messages for your alert. Give the threshold group a description and specify that the threshold group actions will be of type Message. Add your detail action and your summary action as members of the group.

### **Step 4: Define Action Sets**

Disable the existing action set for the alert. Define a new action set named Threshold Action Set, and give the action set a description. Add your threshold group as a member of the action set, and specify that the action should be aborted if an error occurs during execution. Specify a summary threshold value for the threshold group that is equal to or larger than the number of records retrieved by your SELECT statement to trigger the detail action.

### **Step 5: Test the Alert**

Submit a periodic alert check for your alert. Use your mail system to check your e-mail account and view the messages sent by Oracle Alert.

### **Step 6: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

### **Step 7: Modify the Summary Threshold**

Disable the threshold group member of your action set. Then add the same threshold group as a new member of the action set, and change the sequence number of the new member to 1. Specify a summary threshold value for the threshold group that is smaller than the number of records retrieved by your SELECT statement to trigger the summary action. Then repeat steps 5 and 6 to view the results of the change.

## **Practice 4-2 Solutions**

### **Step 1: Copy the Alert**

- 1 Navigate to the Alerts window.  
Alert Manager: (N) Alert—>Define**
- 2 Query the *XX Invoices on Hold Notification* alert that you defined in Practice 3-2.**
- 3 Select Transfer Alert from the Tools menu to navigate to the Transfer Alert Definitions window.**
- 4 Enter the following information in the Destination Alert region:**
  - **Application: Oracle Payables**
  - **Alert: *XX Threshold Invoices on Hold***
- 5 Click the Transfer button.**
- 6 Close the Transfer Alert Definitions window.**
- 7 In the Alerts window, query the new *XX Threshold Invoices on Hold* alert.**
- 8 Check the Enabled check box.**
- 9 Save your work.**

- 10 Click the Run button. Note how many records are retrieved by your SELECT statement.

## Step 2: Define the Actions

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Select the existing summary action named Invoices on Hold E-mail Notification.
- 4 Click the Action Details button to navigate to the Action Details window.
- 5 In the Subject field, replace the existing subject line with the following:  

```
Invoices on active hold summary
```
- 6 Select the Text option and replace the existing message text with the following:  

```
The following invoices are on hold:
Vendor Name Invoice Number Hold Type Hold Reason
***= Enter summary template below this line ***=
** &vendor_name &invoice_number &hold_type
&hold_reason
***= Enter summary template above this line ***=
```
- 7 Navigate back to the Actions window.
- 8 Enter information to define a new action in the following fields:
  - Action Name: Detail Message
  - Description: Detail message for threshold alert
  - Action Level: Detail
- 9 Click the Action Details button to navigate to the Action Details window.
- 10 Enter the action detail information in the following fields:
  - Action Type: Message
  - To: <your e-mail address>
  - Subject: Invoice on hold
- 11 Select the Text option, and enter the following message text:  

```
The following invoice is on hold:
Vendor name:      &vendor_name
Invoice number:   &invoice_number
Hold type:        &hold_type
Hold reason:      &hold_reason
```
- 12 Save your work.

## Step 3: Define the Threshold Group

- 1 Navigate back to the Alerts window.

- 2 **Select Threshold Groups from the Tools menu to navigate to the Threshold Groups window.**
- 3 **Enter information to define the threshold group in the following fields:**
  - **Group Name: E-mail Messages**
  - **Description: E-mail messages for threshold alert**
  - **Type: Message**
- 4 **Navigate to the Threshold Group Members region.**
- 5 **Enter the following information to define the first threshold group member:**
  - **Action: Detail Message**
  - **Enabled: Checked**
- 6 **Enter the following information to define the second threshold group member:**
  - **Action: Invoices on Hold E-mail Notification**
  - **Enabled: Checked**
- 7 **Save your work.**

#### **Step 4: Define Action Sets**

- 1 **Navigate back to the Alerts window.**
- 2 **Click the Action Sets button to navigate to the Action Sets window.**
- 3 **Disable the existing E-mails action set by clearing the Enabled check box.**
- 4 **Enter information to define a new action set in the following fields:**
  - **Seq: 2**
  - **Action Set Name: Threshold Action Set**
  - **Description: Threshold action set**
  - **Enabled: Checked**
- 5 **Navigate to the Members region.**
- 6 **Enter information to define the action set member in the following fields:**
  - **Seq: 1**
  - **Action: E-mail Messages**
  - **On Error Action: Abort**
  - **Enabled: Checked**
- 7 **In the Summary Threshold field, enter a summary threshold value for the threshold group that is equal to or larger than the number of records retrieved by your SELECT statement to trigger the detail action.**
- 8 **Save your work.**

#### **Step 5: Test the Alert**

- 1 **Navigate to the Request Periodic Alert Check window.**  
Alert Manager: (N) Request—>Check
- 2 **Select the application and alert name you want to check. In this case, select the following:**
  - **Application: Oracle Payables**
  - **Alert Name: XX Threshold Invoices on Hold**
- 3 **Click the Submit Request button to run the check. A Note window appears to inform you when the periodic alert is scheduled to run.**
- 4 **Click the OK button to close the Note window.**
- 5 **Note your request ID in the Request Periodic Alert Check window.**
- 6 **Navigate to the Requests window.**  
Alert Manager: (N) Request—>View (B) Find
- 7 **Monitor the request until it is completed.**
- 8 **Use your mail system to check your e-mail account and view the messages sent by Oracle Alert.**

#### **Step 6: Review Alert History**

**After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:**

- **Application: Oracle Payables**
- **Alert Name: XX Threshold Invoices on Hold**

#### **Step 7: Modify the Summary Threshold**

- 1 **Navigate back to the Alerts window.**  
Alert Manager: (N) Alert—>Define
  - 2 **Query your alert.**
  - 3 **Click the Action Sets button to navigate to the Action Sets window.**
  - 4 **Select your Threshold Action Set.**
  - 5 **Navigate to the Members region.**
  - 6 **Disable the existing E-mail Messages member by clearing the Enabled check box.**
  - 7 **Enter information to define a new action set member in the following fields:**
    - **Seq: 1**
    - **Action: E-mail Messages**
    - **On Error Action: Abort**
    - **Enabled: Checked**
- Note: You may need to change the sequence number to 1 manually.**
- 8 **In the Summary Threshold field, enter a summary threshold value for the threshold group that is smaller than the number of records retrieved by your SELECT statement to trigger the summary action.**
  - 9 **Save your work.**



**10 Repeat steps 5 and 6 to view the results of modifying the summary threshold.**

## Summary

---

### Summary

- To define a summary threshold, you first create an alert in the Alerts window and define one detail action and one summary action for the alert.
- A threshold group will contain the two actions as members.
- The summary threshold represents the maximum number of exceptions that the alert can find before it switches from performing the detail action to performing the summary action.
- You can delete an alert only if the alert has no history, no enabled action sets, and no outstanding responses.
- After you copy an alert, you must enable the alert before you can use it.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Lesson Summary

In this lesson, you learned how to let Oracle Alert determine whether to perform detail or summary actions for an alert by defining a summary threshold. If the number of exceptions for an alert exceeds the summary threshold value that you define, Oracle Alert performs a summary action; otherwise, it performs a detail action for each exception found.

You can copy alert definitions. After you copy an alert, you must enable the alert before you can use it.

You can only delete an alert if the alert has no history, no enabled actions sets, and no outstanding responses. In an upcoming lesson, you will learn what a response is.

# Implementing Duplicate Checking

## Chapter 5

# Implementing Duplicate Checking and Action Escalation

---

## Implementing Duplicate Checking and Action Escalation

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Objectives

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

- Check for duplicate exceptions
- Define an action escalation alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Duplicate Checking

You can define an alert to check for exceptions that remain in your database over time. Each time Oracle Alert encounters an exception it found during a previous alert check, it considers that exception a “duplicate” and can perform alert actions based on the presence of that duplicate exception. You determine which outputs Oracle Alert should consider when checking for duplicates, and you can turn duplicate checking “off” for any or all alert outputs.

## Duplicate Exceptions

---

<b>Duplicate Exceptions</b>			
<b>Run Number</b>	<b>Vendor Name</b>	<b>Invoice Number</b>	<b>Hold Type</b>
1	ABC Corp.	1234	QTYREC
2	XYZ Corp.	6789	QTYREC
2	** ABC Corp.	1234	QTYREC

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Implementing Duplicate Checking

To implement duplicate checking while defining an alert, you should:

- Save alert history
- Define a default duplicate configuration at the alert level
- Optionally modify the duplicate configuration at the action set level
- Test your alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Saving Alert History

- While defining the alert in the Alerts window, use the Keep\_Days field to enter the number of days for which you want Oracle Alert to save the alert history.
- You must save the history for your alert for at least one day longer than the frequency of the alert if you want to use duplicate checking. Oracle Alert cannot check for duplicates if the history for the previous alert check is not retained.

(N) Alert—>Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



### Defining the Default Duplicate Configuration

Use the **Outputs** region of the **Alert Details** window to define the duplicate exception conditions:

- Select the outputs that you want to check for duplicate exceptions by enabling the **Check for Duplicates** option for those outputs.
- This configuration becomes the default for all action sets you create for this alert.

**(N) Alert**—>**Define (B) Alert Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Modifying the Duplicate Configuration for an Action Set

### **Modifying the Duplicate Configuration for an Action Set**

- Oracle Alert automatically creates the duplicate configuration for a new action set from the default duplicate configuration you defined for the alert.
- You can modify the duplicate configuration for the action set in the Outputs region of the Action Sets window by enabling or disabling the Check for Duplicates option for each output.
- Each action set duplicate configuration takes precedence over the default alert duplicate configuration during an alert check.

**(N) Alert—>Define (B) Action Sets**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Suppressing Actions for Duplicate Exceptions

---

### Suppressing Actions for Duplicate Exceptions

- Check the **Suppress Duplicates** check box in the **Action Sets** window if you want Oracle Alert not to perform the detail actions in this action set for duplicate exceptions.
- If you check **Suppress Duplicates** for an action set that includes a summary message action, Oracle Alert still includes duplicate exceptions in the summary message, but marks the duplicates with an asterisk (\*).

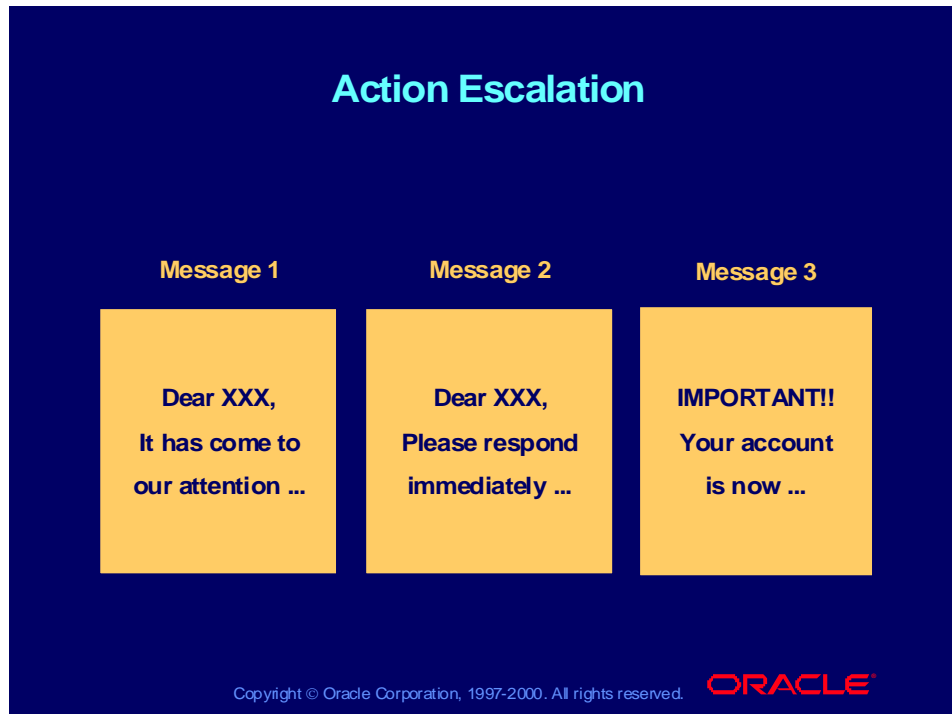
(N) Alert—>Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Action Escalation

---



### Action Escalation

You can create a set of actions—each action of a different level—that Oracle Alert performs if it finds the same exception during consecutive alert checks. Each time that Oracle Alert finds the same duplicate exception, it performs the next level detail action. For every escalation level, you can define a different action. For each alert, you can define an unlimited number of escalation levels.

### Implementing Action Escalation

To implement action escalation while defining an alert, you should:

- Save alert history
- Define the duplicate configuration
- Define actions for different escalation severity levels
- Add the actions to an escalation group
- Add the escalation group as a member of the action set for the alert
- Test your alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining Actions for Escalation

- Use the Actions window to define actions for escalation.
- Define an action for each escalation severity level you will use. Each time the same exception is found, Oracle Alert will perform the action at the next higher escalation severity level.
- Escalation actions must have an action level of Detail.

**(N) Alert—>Define (B) Actions**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Defining an Escalation Group

---

### Defining an Escalation Group

Use the Escalation Groups window to enter:

- Group Name
- Description
- Type (All member actions must be of the same type)
- Escalation Group Members
  - Action Name
  - Level (Escalation severity level)

(N) Alert—>Define—>(M) Tools—>Escalation Groups

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Adding an Escalation Group to an Action Set

---

### Adding an Escalation Group to an Action Set

Use the Action Sets window to define an action set. In the Members region, select the escalation group name in the Action field.

(N) Alert—>Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Suppressing Escalation Group Actions for Duplicate Exceptions

---

### Suppressing Escalation Group Actions for Duplicate Exceptions

- If you check the **Suppress Duplicates** check box in the **Action Sets** window, Oracle Alert performs each action in the escalation group once. After performing the highest level action, Oracle Alert no longer performs any action within that escalation group, even if the exception remains in your database.
- If you do not check the **Suppress Duplicates** check box, Oracle Alert performs each action at the appropriate level, but when it reaches the highest level action it continues to perform that action as long as the exception remains in your database.

(N) Alert—>Define (B) Action Sets

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Practice 5-1

This practice covers the following topics:

- Understanding duplicate checking
- Understanding action escalation

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 5-1 Instructions

Discuss the following questions:

- 1 What is duplicate checking?
- 2 How do you check for duplicates?
- 3 What is action escalation?
- 4 What is an escalation group?
- 5 What is the rule for saving history?
- 6 Explain Suppress Duplicates for a detail action and for a summary action.

### Practice 5-1 Solutions

- 1 What is duplicate checking?  
**An alert feature that checks for exceptions that remain in your database over time. Each time Oracle Alert encounters an exception it found during a previous alert check, it considers that exception a duplicate.**
- 2 How do you check for duplicates?  
**You determine which outputs Oracle Alert should consider when checking for duplicates by enabling the Check For Duplicates check box in the Alert Details window.**
- 3 What is action escalation?

**You can create a set of actions—each action of a different severity level—that Oracle Alert performs if it finds the same exception during consecutive alert checks.**

- 4 What is an escalation group?

**An escalation group defines the severity level of each action that you created.**

- 5 What is the rule for saving history?

**Save history at least one day longer than the frequency of the alert. Oracle Alert cannot check for duplicates if history for the previous alert check is not retained.**

- 6 Explain Suppress Duplicates for a detail action and for a summary action.

**If Suppress Duplicates is enabled, Oracle Alert will not execute the action set if the exceptions found are duplicates.**

**For a summary message Oracle Alert places asterisks next to duplicate exceptions in the summary message.**

**For detail actions that are members of an escalation group (within the action set), Oracle Alert performs the action that corresponds to the appropriate escalation level. Once Alert performs the action that corresponds to the highest escalation level, it stops performing actions within that escalation group.**

### Practice 5-2

This practice covers the following topics:

- Modifying a periodic alert to use duplicate checking
- Defining an action escalation group

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 5-2 Instructions

#### Scenario

In this practice, you will define an alert with an escalation group and use duplicate checking to send escalated messages at increasing severity levels each time the same exceptions are found. You will define messages at three different escalation severity levels.

#### Step 1: Define the Alert

Define a periodic alert named *XX Action Escalation* for the Oracle Payables application, and give the alert a description. Specify that the alert should be run on demand. Save the alert history for two days. Use the following `SELECT` statement:

```
SELECT    aph.hold_reason,
          aph.hold_lookup_code,
          api.invoice_num,
          pov.vendor_name
INTO      &hold_reason,
          &hold_type,
          &invoice_number,
          &vendor_name
FROM      po_vendors pov,
```

```

        ap_invoices api,
        ap_holds aph,
        hr_organization_units hou
WHERE   hou.name = 'Vision Operations'
AND     hou.organization_id = aph.org_id
AND     aph.release_lookup_code IS NULL
AND     aph.invoice_id = api.invoice_id
AND     api.vendor_id = pov.vendor_id
ORDER BY pov.vendor_name

```

After you define your alert, verify your SELECT statement syntax, and run the SELECT statement to check how many records it retrieves.

### **Step 2: Define the Actions**

Define three detail actions for the alert, named First Notice to Manager, Second Notice to Manager, and Third Notice to Manager. Give each action a description. Specify that each action should send a message to you. Write subject lines and message text for the messages. In each case, the message text should include the vendor name, invoice number, and hold type for the invoice on hold, and the text for the second and third messages should request more rapid responses.

### **Step 3: Define the Escalation Group**

Define an escalation group named Invoice on Hold Escalation for your alert. Give the escalation group a description and specify that the escalation group actions will be of type Message. Add your three detail actions as members of the group, each with a different escalation severity level.

### **Step 4: Define the Action Set**

Define an action set named Invoices on Hold Escalation, and give the action set a description. Add your escalation group as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

**Note:** Only the escalation group needs to be a member of the action set. You do not need to add the individual actions as members of the action set.

### **Step 5: Test the Alert**

Submit three periodic alert checks for your alert. After each check, use your mail system to check your e-mail account and view the messages sent by Oracle Alert. Verify that you receive the escalated messages you defined.

### **Step 6: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

## **Practice 5-2 Solutions**

### **Step 1: Define the Alert**

- 1 Navigate to the Alerts window in the Alert Manager responsibility.  
Alert Manager: (N) Alert—>Define**

- 2 Enter information to define the alert in the following fields:
  - Application: Oracle Payables
  - Name: XX Action Escalation
  - Description: XX Action Escalation
  - Enabled: Checked
- 3 Navigate to the Periodic region.
- 4 Enter the periodic detail information in the following field:
  - Frequency: On Demand
- 5 Enter 2 in the Keep\_Days field to save history for the alert.
- 6 Enter the following SELECT statement in the Select Statement field:

```

SELECT    aph.hold_reason,
          aph.hold_lookup_code,
          api.invoice_num,
          pov.vendor_name
INTO      &hold_reason,
          &hold_type,
          &invoice_number,
          &vendor_name
FROM      po_vendors pov,
          ap_invoices api,
          ap_holds aph,
          hr_organization_units hou
WHERE     hou.name = 'Vision Operations'
AND      hou.organization_id = aph.org_id
AND      aph.release_lookup_code IS NULL
AND      aph.invoice_id = api.invoice_id
AND      api.vendor_id = pov.vendor_id
ORDER BY pov.vendor_name

```

**Note:** This is the same SELECT statement as the one used for the XX Invoices on Hold alert in Practice 3-2. You can save time by copying and pasting the text of the SELECT statement if you wish.

- 7 Save your work.
- 8 Click the Verify button to check your SQL for syntax errors.
- 9 Click the Run button to check how many records are retrieved by your SELECT statement.

#### Step 2: Define the Actions

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Enter information to define the first action in the following fields:

- **Action Name: First Notice to Manager**
  - **Description: First Notice to Manager**
  - **Action Level: Detail**
- 4 **Click the Action Details button to navigate to the Action Details window.**
  - 5 **Enter the action detail information in the following fields:**
    - **Action Type: Message**
    - **To: <your e-mail address>**
    - **Subject: First Notice of Invoice on Hold**
  - 6 **Select the Text option, and enter the following message text:**  
`Invoice &invoice_number from &vendor_name is on &hold_type hold for the reason: &hold_reason.`
  - 7 **Save your work.**
  - 8 **Navigate back to the Actions window.**
  - 9 **Enter information to define the second action in the following fields:**
    - **Action Name: Second Notice to Manager**
    - **Description: Second Notice to Manager**
    - **Action Level: Detail**
  - 10 **Click the Action Details button to navigate to the Action Details window.**
  - 11 **Enter the action detail information in the following fields:**
    - **Action Type: Message**
    - **To: <your e-mail address>**
    - **Subject: Second Notice of Invoice on Hold**
  - 12 **Select the Text option, and enter the following message text:**  
`Invoice &invoice_number from &vendor_name is still on &hold_type hold. Please resolve this as quickly as possible as we value our good relationship with &vendor_name.`
  - 13 **Save your work.**
  - 14 **Navigate back to the Actions window.**
  - 15 **Enter information to define the third action in the following fields:**
    - **Action Name: Third Notice to Manager**
    - **Description: Third Notice to Manager**
    - **Action Level: Detail**
  - 16 **Click the Action Details button to navigate to the Action Details window.**
  - 17 **Enter the action detail information in the following fields:**
    - **Action Type: Message**

- **To:** <your e-mail address>
- **Subject:** Third Notice of Invoice on Hold -- ATTENTION!

**18** Select the Text option, and enter the following message text:

`Invoice &invoice_number from &vendor_name is  
STILL on &hold_type hold. Please contact  
&vendor_name and resolve this within 24  
hours.`

**19** Save your work.

### **Step 3: Define the Escalation Group**

- 1** Navigate back to the Alerts window.
- 2** Select Escalation Groups from the Tools menu to navigate to the Escalation Groups window.
- 3** Enter information to define the escalation group in the following fields:
  - **Group Name:** Invoice on Hold Escalation
  - **Description:** Escalation messages for invoices on hold
  - **Type:** Message
- 4** Navigate to the Escalation Group Members region.
- 5** Enter the following information to define the first escalation group member:
  - **Action Name:** First Notice to Manager
  - **Level:** 1
  - **Enabled:** Checked
- 6** Enter the following information to define the second escalation group member:
  - **Action Name:** Second Notice to Manager
  - **Level:** 2
  - **Enabled:** Checked
- 7** Enter the following information to define the third escalation group member:
  - **Action Name:** Third Notice to Manager
  - **Level:** 3
  - **Enabled:** Checked
- 8** Save your work.

### **Step 4: Define the Action Set**

- 1** Navigate back to the Alerts window.
- 2** Click the Action Sets button to navigate to the Action Sets window.
- 3** Enter information to define the action set in the following fields:
  - **Seq:** 1
  - **Action Set Name:** Invoices on Hold Escalation



- **Description: Invoices on hold escalation**
  - **Enabled: Checked**
- 4 **Navigate to the Members region.**
  - 5 **Enter information to define the action set member in the following fields:**
    - **Seq: 1**
    - **Action: Invoice on Hold Escalation**
    - **On Error Action: Abort**
    - **Enabled: Checked**
  - 6 **Save your work.**

#### **Step 5: Test the Alert**

- 1 **Navigate to the Request Periodic Alert Check window.**  
**Alert Manager: (N) Request—>Check**
- 2 **Select the application and alert name you want to check. In this case, select the following:**
  - **Application: Oracle Payables**
  - **Alert Name: XX Action Escalation**
- 3 **Click the Submit Request button to run the check. A Note window appears to inform you when the periodic alert is scheduled to run.**
- 4 **Click the OK button to close the Note window.**
- 5 **Note your request ID in the Request Periodic Alert Check window.**
- 6 **Navigate to the Requests window.**  
**Alert Manager: (N) Request—>View (B) Find**
- 7 **Monitor the request until it is completed.**
- 8 **Use your mail system to check your e-mail account and view the messages sent by Oracle Alert.**
- 9 **Repeat steps 1 through 8 in this section to check your alert a second time. Verify that you receive the second-level message you defined in your escalation group.**
- 10 **Repeat steps 1 through 8 in this section again to check your alert a third time. Verify that you receive the third-level message you defined in your escalation group.**

#### **Step 6: Review Alert History**

**After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:**

- **Application: Oracle Payables**
- **Alert Name: XX Action Escalation**

## Summary

---

### Summary

- By saving alert history, you can have Alert check for exceptions that existed in your database during a previous alert check.
- You can flag duplicate exceptions within summary messages by specifying that you want to suppress duplicates at the action set level.
- You can create escalation groups to perform actions at increasing escalation severity levels each time an exception is found.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Lesson Summary

In this lesson, you learned how to define an alert to check exceptions that remain in your database over time. You can set up the conditions to check over time and you can optionally turn the duplicate checking off.

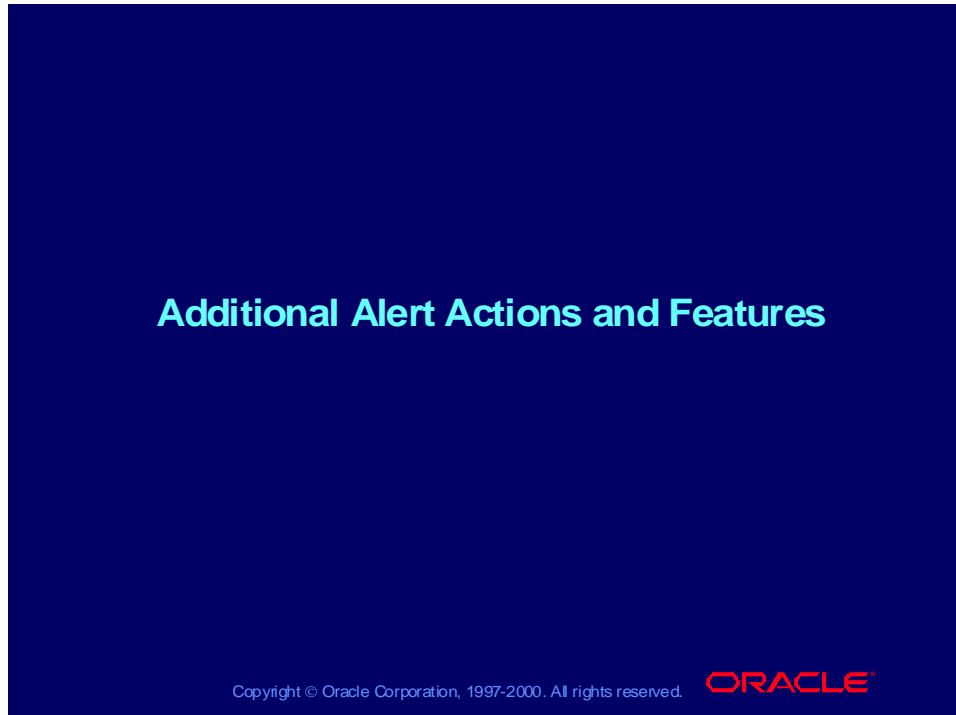
Action escalation is used to have Alert perform different actions each time the same exception is found over time. These actions are placed in an escalation group.

# **Additional Alert Actions and Features**

## **Chapter 6**

## Additional Alert Actions and Features

---



## Objectives

---

### Objectives

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

- Create, run, and test a SQL statement script action for an alert
- Create, run and test a concurrent program action
- Describe how to use an operating system script action
- Discuss how you would use distribution lists
- Define a periodic set and determine when you would use one
- Identify the uses of precoded alerts

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Alert Actions

Oracle Alert can perform four types of alert actions during an alert check or as a result of processing a response:

- Message
- SQL statement script
- Concurrent program
- Operating system script

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### SQL Statement Script Actions

- Oracle Alert can run a SQL statement script as an alert action.
- You can provide the SQL statement script as text in Oracle Alert or as a separate file.

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Critical Outputs in SQL Statement Script Actions

- If the SQL statement script is provided as text in Oracle Alert, Oracle Alert considers all the outputs included in the SQL statement to be critical outputs.
- If the SQL statement script is provided as a file, Oracle Alert considers the outputs used as the file name and arguments (those outputs used in the File and Arguments fields of the Action Details window) to be critical outputs.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



### Business Problem

Define a periodic alert to solve the following business problem:

- **Problem**—You want to clean out periodically old alert and action set checks from the database.
- **Solution**—Create a periodic alert.
  - Automatically deletes old alert checks
  - Automatically deletes old action set checks

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Using a SQL Script Action

A periodic alert is used to clean out the old alert checks because it can be scheduled to run when needed. The SQL statement action would be used to do the delete from the ALR\_ACTION\_SET\_CHECKS table.

### Defining SQL Statement Script Actions

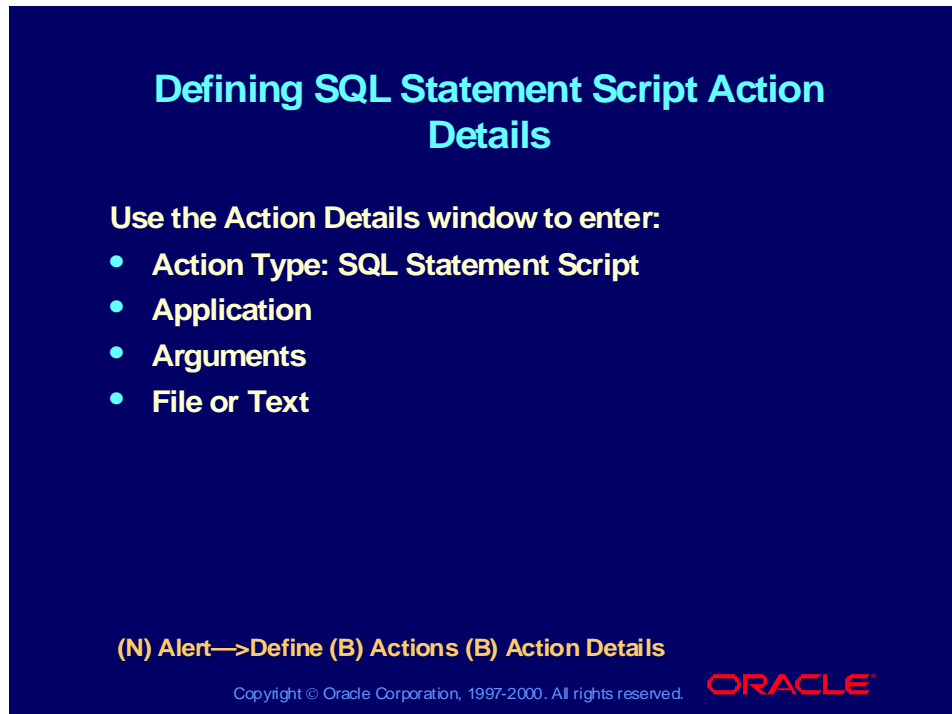
- Define an alert
- Define alert details
- Define actions
- Define SQL statement script action details
- Define action sets
- Test your alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Defining SQL Statement Script Action Details

---



**Defining SQL Statement Script Action Details**

Use the Action Details window to enter:

- Action Type: SQL Statement Script
- Application
- Arguments
- File or Text

(N) Alert—>Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

### Defining SQL Statement Script Action Details

- If the SQL statement exists in a file, select the File button, and enter the location and name of the SQL Statement.
- If you are going to write the SQL statement directly in Oracle Alert, select the Text button, and enter the SQL statement in the text box.
  - Note:** Arguments for the script can be added in the Arguments field.
- End the SQL statement with a slash (/) or a semicolon (;) to execute. SQL statements can be stacked in the script.

## SQL Statement Script Example

---

### SQL Statement Script Example

```
DELETE FROM alr_action_set_checks
WHERE      application_id = &APPLICATION_ID
AND        alert_id = &ALERT_ID
AND        alert_check_id = &ALERT_CHECK_ID
AND        check_id = &ACTION_SET_CHECK_ID
/
```

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Practice 6-1

This practice covers the following topics:

- Defining a periodic alert
- Defining a SQL statement script action

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 6-1 Instructions

#### Scenario

In this practice, you will define an alert with an action that runs a SQL statement script for an exception. The alert will select the new user you created in Practice 2-2 and add the default company fax number to the user's information in FND\_USER.

#### Step 1: Define the Alert

Define a periodic alert named *XX SQL Action Alert* for the Oracle Alert application, and give the alert a description. Specify that the alert should be run on demand. Save the alert history for two days. Use the following SELECT statement:

```
SELECT user_id,  
       user_name  
INTO   &USERID,  
       &USERNAME  
FROM   fnd_user  
WHERE  user_name = 'ALERTXX'
```

After you define your alert, verify your SELECT statement syntax, and run the SELECT statement to check how many records it retrieves.

#### Step 2: Define the Action

Define a detail action named SQL Statement Script Action for the alert. Give the action a description. Specify that the action should run a SQL statement script to add the default company fax number, 650-555-1234, to the information in FND\_USER for your new user from Practice 2-2.

### **Step 3: Define the Action Set**

Define an action set named Add Fax, and give the action set a description. Add your detail action as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

### **Step 4: Test the Alert**

Submit a periodic alert check for your alert. Use the Users window in the System Administrator responsibility to verify that Oracle Alert added the fax number to your user information.

### **Step 5: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

## **Practice 6-1 Solutions**

### **Step 1: Define the Alert**

- 1 Navigate to the Alerts window in the Alert Manager responsibility.  
Alert Manager: (N) Alert—>Define**
- 2 Enter information to define the alert in the following fields:**
  - **Application: Oracle Alert**
  - **Name: XX SQL Action Alert**
  - **Description: XX SQL Action Alert**
  - **Enabled: Checked**
- 3 Navigate to the Periodic region.**
- 4 Enter the periodic detail information in the following field:**
  - **Frequency: On Demand**
- 5 Enter 2 in the Keep\_Days field to save history for the alert.**
- 6 Enter the following SELECT statement in the Select Statement field:**

```
SELECT user_id,  
       user_name  
INTO   &USERID,  
       &USERNAME  
FROM   fnd_user  
WHERE  user_name = 'ALERTXX'
```
- 7 Save your work.**
- 8 Click the Verify button to check your SQL for syntax errors.**
- 9 Click the Run button to check how many records are retrieved by your SELECT statement.**

### **Step 2: Define the Action**

- 1 **Navigate back to the Alerts window.**
- 2 **Click the Actions button to navigate to the Actions window.**
- 3 **Enter information to define the action in the following fields:**
  - **Action Name: SQL Statement Script Action**
  - **Description: SQL statement script action**
  - **Action Level: Detail**
- 4 **Click the Action Details button to navigate to the Action Details window.**
- 5 **Enter the action detail information in the following field:**
  - **Action Type: SQL Statement Script**
- 6 **Select the Text option, and enter the following SQL statement script:**

```
UPDATE FND_USER
SET    fax = '650-555-1234'
WHERE user_id = &USERID;
```
- 7 **Save your work.**

### **Step 3: Define the Action Set**

- 1 **Navigate back to the Alerts window.**
- 2 **Click the Action Sets button to navigate to the Action Sets window.**
- 3 **Enter information to define the action set in the following fields:**
  - **Seq: 1**
  - **Action Set Name: Add Fax**
  - **Description: Add fax number with SQL statement script**
  - **Enabled: Checked**
- 4 **Navigate to the Members region.**
- 5 **Enter information to define the action set member in the following fields:**
  - **Seq: 1**
  - **Action: SQL Statement Script Action**
  - **On Error Action: Abort**
  - **Enabled: Checked**
- 6 **Save your work.**

### **Step 4: Test the Alert**

- 1 **Navigate to the Request Periodic Alert Check window.**  
**Alert Manager: (N) Request—>Check**
- 2 **Select the application and alert name you want to check. In this case, select the following:**
  - **Application: Oracle Alert**
  - **Alert Name: XX SQL Action Alert**

- 3 Click the **Submit Request** button to run the check. A **Note window** appears to inform you when the periodic alert is scheduled to run.
- 4 Click the **OK** button to close the **Note window**.
- 5 Note your request ID in the **Request Periodic Alert Check** window.
- 6 Navigate to the **Requests** window.  
Alert Manager: (N) Request—>View (B) Find
- 7 Monitor the request until it is completed.
- 8 Navigate to the **Users** window in the **System Administrator** responsibility.  
System Administrator: (N) Security—>User—>Define
- 9 Query your user.
- 10 Verify that the **Fax** field contains the fax number added by Oracle Alert.

#### **Step 5: Review Alert History**

After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:

- **Application: Oracle Alert**
- **Alert Name: XX SQL Action Alert**



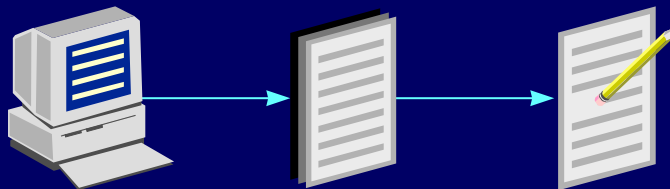
## Concurrent Programs

---

### Concurrent Programs

A concurrent program is an executable file that performs a specific task, such as reversing a journal entry or generating a report.

#### Reversing Journals



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Concurrent Program Action Levels

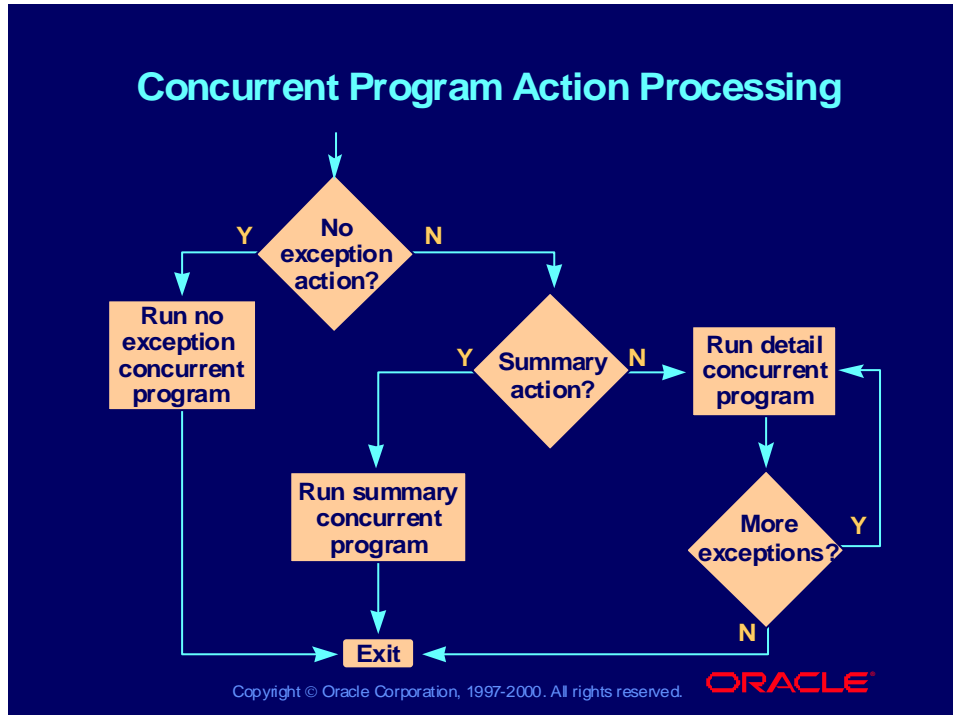
- **Detail concurrent program actions**—Oracle Alert submits the concurrent program request once for each exception found during an alert check.
- **Summary concurrent program actions**—Oracle Alert submits the concurrent program request once for each unique combination of critical output values. Outputs used as arguments are considered critical outputs.
- **No exception concurrent program actions**—Oracle Alert submits the concurrent program request if it finds no exceptions during an alert check. Do not use outputs when defining your no exception concurrent program actions.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Concurrent Program Action Processing

---



## Defining Concurrent Program Action Details

---

### Defining Concurrent Program Action Details

Use the Action Details window to enter:

- Action Type: Concurrent Program
- Application
- Program Name
- Arguments: Enter arguments separated by spaces

(N) Alert—>Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 6-2

This practice covers the following topics:

- Defining a periodic alert
- Defining a concurrent program action

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 6-2 Instructions

#### Scenario

In this practice, you will define an alert with an action that runs a concurrent program. The alert action will use the Reverse Journals concurrent program to create reversing entries for journal entries marked for reversal with a category of Accrual. You will also create a sample journal entry against which you can test your alert.

#### Step 1: Create a Journal Entry

- 1 Navigate to the Enter Journals window in the General Ledger, Vision Operations (USA) responsibility. The Find Journals window appears.  
General Ledger, Vision Operations (USA): (N) Journals—>Enter
- 2 In the Find Journals window, click the New Journal button to navigate to the Journals window.
- 3 Enter information to create the journal entry in the following fields:
  - Journal: XX Test Journal Entry
  - Category: Accrual
  - Description: XX Test Journal Entry
- 4 Enter information to create the first journal entry line as follows:
  - Line: 10
  - Account: 01-410-5800-1300-000

- Debit: 500.00
- 5 Enter information to create the second journal entry line as follows:
  - Line: 20
  - Account: 01-000-2410-1300-000
  - Credit: 500.00
- 6 Save your work.
- 7 Click the More Details button to navigate to the More Details window.
- 8 Navigate to the Reverse region.
- 9 In the Period field, select the next period after the current period.
 

**Note:** Do not click the Reverse Journal button.
- 10 Save your work.

### Step 2: Define the Alert

Define a periodic alert named *XX Create Reversing Journals* for the Oracle General Ledger application, and give the alert a description. Specify that the alert should be run on demand. Save alert history for two days. Use the following SELECT statement:

```
SELECT name ,
       je_header_id,
       period_name
INTO   &journal_entry_name ,
       &journal_header_id ,
       &period_name
FROM   gl_je_headers
WHERE  accrual_rev_flag = 'Y'
AND    je_category = 'Accrual'
AND    name = 'XX Test Journal Entry'
```

After you define your alert, verify your SELECT statement syntax, and run the SELECT statement to check how many records it retrieves.

### Step 3: Define the Action

Define a detail action named *Create Reversals* for the alert. Give the action a description. Specify that the action should run the concurrent program *Reverse Journals* in the Oracle General Ledger application, with the argument *&journal\_header\_id*.

### Step 4: Define the Action Set

Define an action set named *Generate Reversing Entries*, and give the action set a description. Add your detail action as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

### Step 5: Test the Alert

Submit a periodic alert check for your alert. Review your journal entry and its reversing entry in Oracle General Ledger to verify that the original journal entry has been reversed.

## Step 6: Review Alert History

Review the history for your alert, including checks, exceptions and actions.

### Practice 6-2 Solutions

#### Step 1: Create a Journal Entry

- 1 **Navigate to the Enter Journals window in the General Ledger, Vision Operations (USA) responsibility. The Find Journals window appears.**  
**General Ledger, Vision Operations (USA): (N) Journals—>Enter**
- 2 **In the Find Journals window, click the New Journal button to navigate to the Journals window.**
- 3 **Enter information to create the journal entry in the following fields:**
  - **Journal: XX Test Journal Entry**
  - **Category: Accrual**
  - **Description: XX Test Journal Entry**
- 4 **Enter information to create the first journal entry line as follows:**
  - **Line: 10**
  - **Account: 01-410-5800-1300-000**
  - **Debit: 500.00**
- 5 **Enter information to create the second journal entry line as follows:**
  - **Line: 20**
  - **Account: 01-000-2410-1300-000**
  - **Credit: 500.00**
- 6 **Save your work.**
- 7 **Click the More Details button to navigate to the More Details window.**
- 8 **Navigate to the Reverse region.**
- 9 **In the Period field, select the next period after the current period.**  
**Note: Do not click the Reverse Journal button.**
- 10 **Save your work.**

#### Step 2: Define the Alert

- 1 **Navigate to the Alerts window in the Alert Manager responsibility.**  
**Alert Manager: (N) Alert—>Define**
- 2 **Enter information to define the alert in the following fields:**
  - **Application: Oracle General Ledger**
  - **Name: XX Create Reversing Journals**
  - **Description: XX Create Reversing Journals**
  - **Enabled: Checked**
- 3 **Navigate to the Periodic region.**
- 4 **Enter the periodic detail information in the following field:**
  - **Frequency: On Demand**

- 5 Enter 2 in the **Keep\_Days** field to save history for the alert.
- 6 Enter the following **SELECT** statement in the **Select Statement** field:

```
SELECT name ,
        je_header_id,
        period_name
INTO    &journal_entry_name ,
        &journal_header_id,
        &period_name
FROM    gl_je_headers
WHERE   accrual_rev_flag = 'Y'
AND     je_category = 'Accrual'
AND     name = 'XX Test Journal Entry'
```

- 7 Save your work.
- 8 Click the **Verify** button to check your SQL for syntax errors.
- 9 Click the **Run** button to check how many records are retrieved by your **SELECT** statement.

### Step 3: Define the Action

- 1 Navigate back to the **Alerts** window.
- 2 Click the **Actions** button to navigate to the **Actions** window.
- 3 Enter information to define the action in the following fields:
  - **Action Name:** Create Reversals
  - **Description:** Runs the Reverse Journals program
  - **Action Level:** Detail
- 4 Click the **Action Details** button to navigate to the **Action Details** window.
- 5 Enter the action detail information in the following field:
  - **Action Type:** Concurrent Program
  - **Application:** Oracle General Ledger
  - **Program Name:** Reverse Journals
  - **Arguments:** &journal\_header\_id
- 6 Save your work.

### Step 4: Define the Action Set

- 1 Navigate back to the **Alerts** window.
- 2 Click the **Action Sets** button to navigate to the **Action Sets** window.
- 3 Enter information to define the action set in the following fields:
  - **Seq:** 1
  - **Action Set Name:** Generate Reversing Entries
  - **Description:** Generate reversing journal entries
  - **Enabled:** Checked



- 4 **Navigate to the Members region.**
- 5 **Enter information to define the action set member in the following fields:**
  - **Seq: 1**
  - **Action: Create Reversals**
  - **On Error Action: Abort**
  - **Enabled: Checked**
- 6 **Save your work.**

#### **Step 5: Test the Alert**

- 1 **Navigate to the Request Periodic Alert Check window.**  
**Alert Manager: (N) Request—>Check**
- 2 **Select the application and alert name you want to check. In this case, select the following:**
  - **Application: Oracle General Ledger**
  - **Alert Name: XX Create Reversing Journals**
- 3 **Click the Submit Request button to run the check. A Note window appears to inform you when the periodic alert is scheduled to run.**
- 4 **Click the OK button to close the Note window.**
- 5 **Note your request ID in the Request Periodic Alert Check window.**
- 6 **Navigate to the Requests window.**  
**Alert Manager: (N) Request—>View (B) Find**
- 7 **Monitor the alert request until it is completed and Oracle Alert submits a new request for the Reverse Journals program.**
- 8 **Monitor the Reverse Journals request until it is completed.**
- 9 **Navigate to the Enter Journals window in the General Ledger, Vision Operations (USA) responsibility. The Find Journals window appears.**  
**General Ledger, Vision Operations (USA): (N) Journals—>Enter**
- 10 **In the Find Journals window, enter %XX% in the Journal field.**
- 11 **Enter Unposted in the Posting field in the Status region.**
- 12 **Click the Find button to navigate to the Enter Journals window.**
- 13 **Select your original journal entry and click the Review Journal button to navigate to the Journals window.**
- 14 **Click the More Details button to navigate to the More Details window.**
- 15 **Verify that the Status field in the Reverse region displays a status of Reversed.**
- 16 **Navigate back to the Enter Journals window.**
- 17 **Select your reversing journal entry and click the Review Journal button to navigate to the Journals window.**
- 18 **Review the reversing journal entry.**

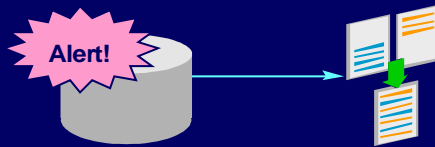
#### **Step 6: Review Alert History**

**After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:**

- **Application: Oracle General Ledger**
- **Alert Name: XX Create Reversing Journals**

### Operating System Script Actions

- Oracle Alert can execute an operating system script as an alert action.
- You can provide the operating system script as text in Oracle Alert or as a separate file.
- When performing an operating system script action, Oracle Alert executes the commands with the privileges of the operating system user that started the Concurrent Manager.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Critical Outputs in Operating System Script Actions

---

### Critical Outputs in Operating System Script Actions

- If the operating system script is provided as text in Oracle Alert, Oracle Alert considers all the outputs included in the operating system script to be critical outputs.
- If the operating system script is provided as a file, Oracle Alert considers the outputs used as the file name and arguments (those outputs used in the File and Arguments fields of the Action Details window) to be critical outputs.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Operating System Script Action Levels

- **Detail operating system script actions—Oracle Alert performs the operating system script once for each exception found during an alert check.**
- **Summary operating system script actions—Oracle Alert performs the operating system script once for each unique combination of critical output values returned by the alert select statement.**
- **No exception operating system script actions—Oracle Alert performs the operating system script if it finds no exceptions during an alert check. Do not use outputs when defining your no exception operating system script actions.**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Defining Operating System Script Action Details

---

**Defining Operating System Script Action Details**

Use the Action Details window to enter:

- **Action Type: Operating System Script**
- **Application**
- **Arguments: Enter arguments separated by spaces**
- **File or Text**

(N) Alert—>Define (B) Actions (B) Action Details

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining Operating System Script Action Details

If the operating system script you want to execute is in a file located in an application's base path BIN directory, specify that application name in the Application field. If the operating system script is in a file, and you do not specify a value in the Application field, then enter the full path of the operating system script, or the output from the alert SQL statement that represents the file name and location in the File field. The operating script file must be less than 64K.

**Note:** Oracle Alert cannot substitute values into output variables located in an operating system script file.

If you want to enter the script in Oracle Alert, the script must be less than 2,000 characters. Identify your alert outputs with an ampersand before the name, for example, &ORACLE\_ID.

### Distribution Lists

**A distribution list is a predefined set of electronic mail usernames and printer instructions that you can use on message actions instead of re-entering all the recipient names each time.**



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Using Distribution Lists

---

### Using Distribution Lists

- **Do not use outputs on distribution lists. Sharing lists across alerts can result in errors if you use outputs.**
- **When you use a list on a message action, you cannot alter the recipients on that message action.**

  
**ORACLE**<sup>®</sup>

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

### Using Distribution Lists

Distribution lists help you simplify the creation of message actions by predefining a set of message recipients for use on many actions. If a recipient changes, you need only adjust it in the distribution list, not in the individual message actions.

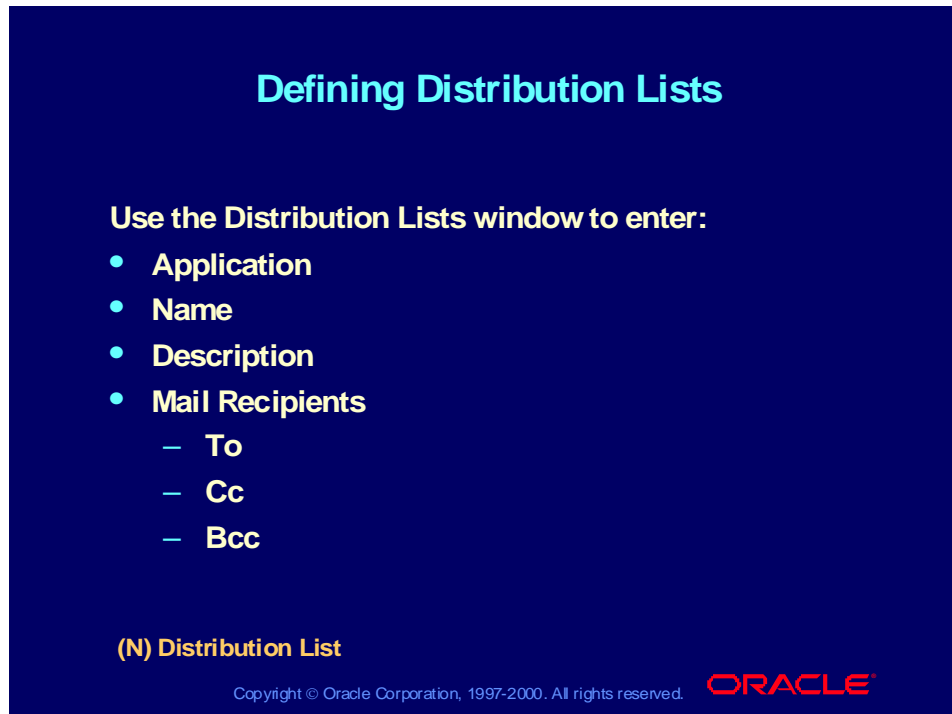
### Ways to Distribute a Message Action

- Enter the user usernames you want to send a message to in the To, Cc, and Bcc fields of the Message Recipients region of the Distribution Lists window.
- Use an Oracle Alert distribution list that you define. All user names assigned to the To, Cc, and Bcc fields on the list automatically appear in the To, Cc, and Bcc fields of the Message Recipients region.
- Initially select an Oracle Alert distribution list in the Message region, then delete the list name, and edit the set of usernames on the message.



## Defining Distribution Lists

---



**Defining Distribution Lists**

Use the Distribution Lists window to enter:

- Application
- Name
- Description
- Mail Recipients
  - To
  - Cc
  - Bcc

(N) Distribution List

Copyright © Oracle Corporation, 1997-2000. All rights reserved. ORACLE

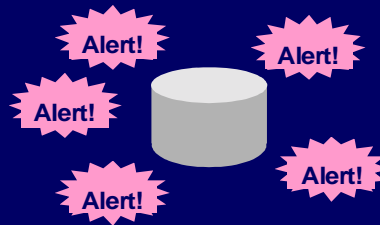
### Defining Distribution Lists

- 1 In the Application field, enter the name of the application that owns the distribution list.
- 2 Enter a name for the distribution list that is unique within the application.
- 3 Enter a description of the distribution list.
- 4 In the Mail Recipients region, enter the electronic mail usernames of the recipients, separated by a space, or a comma (,), or a combination of the two. You can enter as many mail usernames as you want (up to 240 characters).

**Note:** We do not recommend using alert outputs in a distribution list. Because you can use distribution lists for many alerts, and each alert can have different outputs, an output you use in a distribution list for one alert can result in an error when used in another alert.

### Periodic Sets

- A periodic set is a set of periodic alerts that Oracle Alert checks simultaneously.
- Periodic sets let you perform on-demand alert checking more efficiently.
- Each periodic alert you include in a periodic set continues to run according to its individually defined frequency.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Defining Periodic Sets

Use the Periodic Sets window to enter:

- Application
- Name
- Description
- Periodic Set Members
  - Seq
  - Application
  - Name
  - End Date

(N) Alert—>Periodic Set

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Using Periodic Sets

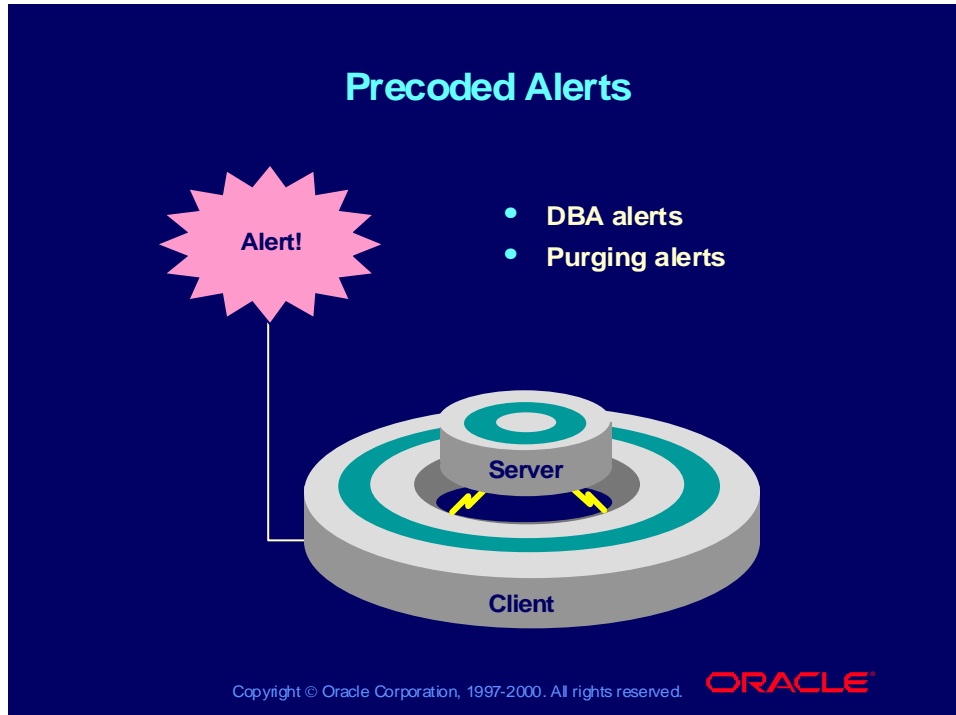
- Before you can define a periodic set, you must define one or more periodic alerts.
- If the application for the set is Oracle Alert, the members can be owned by any application. If the application is anything other than Oracle Alert, the members must be owned by that application.
- You can include as many periodic alerts in your periodic set as you want. You can also include other periodic sets as members of a periodic set.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Precoded Alerts

---



### Precoded DBA Alerts

- **Tables and indexes unable to allocate another extent**
- **Users who are nearing their tablespace quota**
- **Tablespaces without adequate free space**
- **Tables and indexes that are too large or are fragmented**
- **Tables and indexes that are near their maximum extents**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Precoded Purging Alerts

- Delete concurrent requests older than a specified number of days.
- Delete alert checks and action set checks older than a specified number of days.

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Using Precoded Alerts

- **Select Find All from the View menu to find the precoded alert you want to use.**
- **All precoded alerts are periodic alerts, so you can customize the alert frequencies to determine how often they run.**
- **You can define inputs at the alert level to customize your precoded alerts. You can also define your input values at the action set level and create multiple action sets that target different input values.**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



## Practice 6-3

---

### Practice 6-3

This practice covers the following topics:

- Understanding how a SQL statement script works
- Deciding how to use distribution lists at your company
- Defining a periodic set and determining when to use one
- Identifying the uses of precoded alerts

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 6-3 Instructions

Answer the following questions:

- 1 What is a SQL statement script action?
- 2 What is the difference between supplying the SQL statement script in a file and as text in Oracle Alert?
- 3 What must end SQL statement scripts?
- 4 Can you create one action with multiple SQL statements?
- 5 How do you dynamically pass arguments to a SQL statement script action?
- 6 What is a periodic set and how can it benefit you?
- 7 Can you use public electronic mail lists in your message actions?
- 8 Where do you put alert distribution lists in your message actions?
- 9 Discuss with a class member how you use precoded alerts.

### Practice 6-3 Solutions

- 1 What is a SQL statement script action?  
**An action that Oracle Alert performs during an alert check that runs a SQL statement script.**
- 2 What is the difference between supplying the SQL statement script in a file and as text in Oracle Alert?

---

Copyright © Oracle Corporation, 2000. All rights reserved.

**If the statement is text, Oracle Alert considers all the outputs included in the SQL statement to be critical outputs.**

**If the statement is in a file, Oracle Alert considers the critical outputs to be those in the File and Arguments field.**

**Note: Oracle Alert cannot substitute output values for outputs located in a SQL statement script that resides in an external file.**

- 3 What must end SQL statement scripts?

**A semicolon or a slash.**

- 4 Can you create one action with multiple SQL statements?

**Yes, SQL statements can be stacked in the script.**

- 5 How do you dynamically pass arguments to a SQL statement script action?

**Enter outputs in the Arguments field in the Action Details window.**

- 6 What is a periodic set and how can it benefit you?

**Use periodic alerts to check several alerts simultaneously. Each periodic alert you include in a periodic set still continues to run according to its defined schedule. This method is more efficient than checking each alert individually.**

- 7 Can you use public electronic mail lists in your message actions?

**Yes, you can use an e-mail distribution list by entering the list name in the To field of the Message Recipients region.**

- 8 Where do you put alert distribution lists in your message actions?

**In the To, Cc, or Bcc fields in the Mail Recipients region.**

- 9 Discuss with a class member how you use precoded alerts.

**Consider using the purging or DBA alerts. Decide which ones you or your staff could use. You can systematically monitor your system for potential tablespace, disk space, and allocation problems, making your database administrators more efficient and improving database performance.**

**Occasionally, you will want to purge your database of obsolete concurrent requests, alert checks, and action set checks. Oracle Alert provides two alerts that let you periodically remove old files, freeing up valuable tablespace and improving database performance.**

### Summary

- A SQL statement action performs a SQL statement as the alert action.
- A concurrent program action is an executable file that performs a task as the alert action.
- An operating system script action performs an operating system script as the alert action.
- Distribution lists simplify the creation of message actions by predefining a set of message recipients for use on actions.
- DBA and purging alerts are precoded Oracle alerts that can help you optimize your database and clean up old alert data.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



# Response Processing

## Chapter 7

## Response Processing

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Objectives

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

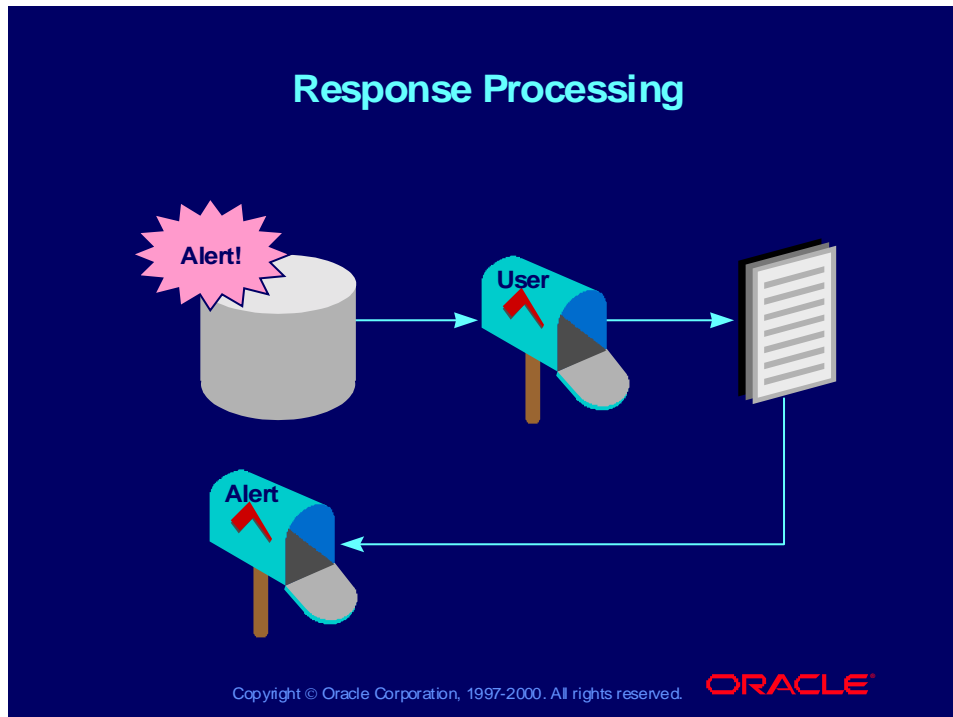
- List response processing uses
- Describe how response processing works
- Define response actions
- Define response sets
- Define response variables
- Define possible responses (valid, invalid, no response)
- Attach a response set to an alert message

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Response Processing

---





### Response Processing

- **Response processing is a feature of Oracle Alert that processes replies from alert recipients.**
- **A response processing alert sends a message that solicits a response from the recipient as its alert action.**
- **Oracle Alert automatically takes predefined actions based on the reply that the user returns.**
- **Response processing helps eliminate operator intervention for routine transactions.**
- **Oracle Alert supports both UNIX Sendmail and Windows NT MAPI-compliant mail applications for response processing.**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Response Processing Terminology

---

## Response Processing Terminology

Term	Definition
Response action	A message, a SQL statement script, an operating system script, or a concurrent program request that you define and that Oracle Alert sends or performs when it receives a response to a response processing alert.
Response set	The set of instructions for valid responses, invalid responses, and no responses to an alert message. A response set need not include invalid or no response responses.
Response variable	A variable whose value is specified by the message recipient.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Response Processing Terminology

---

## Response Processing Terminology

Term	Definition
Valid response	A response that you define and associate with an alert message. You define all responses that are valid for the alert message. Each response includes a set of actions that you want Oracle Alert to perform when the message recipient returns an actual response that matches a valid response. Oracle Alert includes all the valid responses in the alert message, one of which serves as the template for the recipient's actual response.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

# Response Processing Terminology

---

## Response Processing Terminology

Term	Definition
Invalid response	A set of actions that you want Oracle Alert to perform if it receives an actual response that does not match a valid response.
No response	A set of actions that you want Oracle Alert to perform if it does not receive a response within the number of days you specify.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Implementing Response Processing

- Define an alert
- Define an alert message action that solicits a response
- Add the message action to an action set
- Define response actions
- Define a response set
- Define possible responses
- Attach the response set to the alert message action

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Business Problem

Define a response processing alert to solve the following business problem:

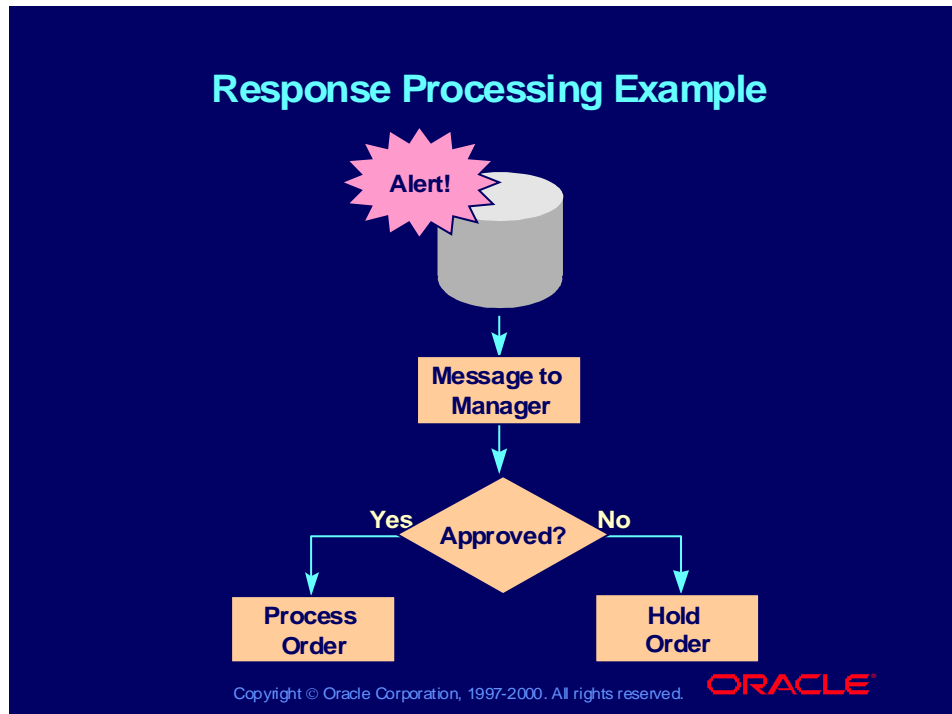
- **Problem**—Managers need to know about and approve orders over a certain amount.
- **Solution**—Create a response processing alert:
  - Alert sends a message to the manager with the order information. If the manager responds “Approved,” the valid response action releases the order for processing and add the comment “Response Processing Approved”.
  - If the manager responds with anything else, or does not respond, the invalid and no response actions hold the order and add the comment “AUTOCREATED—ORDER OVER LIMIT”.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Response Processing Example

---



### Defining Response Actions

- Use the Actions window and the Action Details window to define response actions, following the same procedures as for alert actions.
- Define response actions for valid responses. You can also optionally define an invalid response action and a no response action.
- Response actions must have an action level of Detail.
- You can use alert outputs or response variables in a message or SQL statement script action.

**(N) Alert—>Define (B) Actions**

**(N) Alert—>Define (B) Actions (B) Action Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



### Defining a Response Set

Use the Response Sets window to enter:

- Response Set Name
- Description
- End Date

**(N) Alert**→**Define (B) Response Sets**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining Response Variables

Use the Response Variables window to enter:

- Variable Name
- Description
- Type
- Max Length
- Default

(N) Alert → Define (B) Response Sets (B) Response Variables

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Creating Response Variables

- Create a response variable if you cannot anticipate the exact value of a response.
- Define response variables for values that you want the responder to provide, such as quantity or price.
- Provide a default value for a response variable in case the responder does not return a value for the variable.
- Note that a response variable is not an output.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Defining Possible Responses

- You can define actions for three types of responses:
  - Valid responses
  - Invalid response
  - No response
- You can define as many valid responses as you want.

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

### Defining Valid Responses

Use the Valid Responses window to enter:

- Response Name
- Text
  - This text is the actual text of a valid response.
  - You can use response variables in the text.
  - Alert reads only the first thirty characters in the body of the e-mail message.

(N) Alert → Define (B) Response Sets (B) Valid

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Defining Valid Response Actions

---

### Defining Valid Response Actions

Use the Actions region of the Valid Responses window to enter:

- Seq
- Action
- End Date

Assign as many actions as necessary to represent this response.

(N) Alert—>Define (B) Response Sets (B) Valid

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Defining Invalid Response Actions

---

### Defining Invalid Response Actions

Use the Invalid Response Actions window to enter:

- Seq
- Action
- End Date

Assign as many actions as necessary to follow up on an invalid response.

(N) Alert → Define (B) Response Sets (B) Invalid

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Using Invalid Response Actions

- You do not need to specify text for an invalid response.
- You can select the original alert message action as an invalid response action to re-send the original message that solicits a response.
- You may want to monitor invalid responses to see whether users need training in responding to alert messages.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



## Defining No Response Actions

---

### Defining No Response Actions

Use the No Response Actions window to enter:

- Seq
- Action
- End Date

Assign as many actions as necessary to follow up on a lack of response. You do not need to specify text for a lack of response.

(N) Alert—>Define (B) Response Sets (B) None

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Attaching a Response Set to the Alert Message Action

### **Attaching a Response Set to the Alert Message Action**

Use the Action Details window to enter:

- Reply To (mail username of your e-mail response processing account)
- Response Set
- Response Days (number of days Oracle Alert waits before performing the no response actions in the response set)

**(N) Alert—>Define (B) Actions (B) Action Details**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Response Processing Message Structure

---

### Response Processing Message Structure

- **Message text**—The body of the detail message you defined
- **Generic response text**—Boilerplate text that appears after the message text of each alert message that solicits a response
- **Valid responses**—The text of each valid response you defined in the response set
- **Message handle**—A message identifier assigned by Oracle Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Example of Response Processing Message Structure

-----  
Received: January 30, 2000 11:08  
Sent: January 30, 2000  
From: Oracle Alert <ALRTEST.US.ORACLE.COM>  
To: berickso@us.oracle.com  
Subject: Large Order Entered for Computer Services  
1010  
Reply-to: ALRTEST  
Priority: High

Dear Approver,

A large sales order was booked today. This order is awaiting your approval.

Order Number: 1144

Customer: Computer Services, 1010

Order Total: \$259,792,425.00

Thank you.

ORACLE BUSINESS AGENT (Alert)

---

Copyright © Oracle Corporation, 2000. All rights reserved.

Your response to this mail note will be processed electronically. For successful processing, you must:

- Include this note in your reply by using your mail system's Reply With Original feature.

- Make your actual response EXACTLY like ONE of the following; however, you should replace the question mark (?) with the value you want to specify. Choose one of the following responses:

Approved

Approve and reset the approval threshold to  
NEW\_LIMIT="?"

Message #:1.100324

-----End of Message-----

## Defining Generic Response Text

---

### Defining Generic Response Text

Use the **Message Elements** region of the **Oracle Alert Options** window to enter:

- **Name: Response Text**
- **Text to include in response processing messages**

**(N) System—>Options**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## How Response Processing Works

---

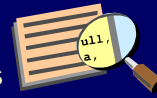
### How Response Processing Works

Oracle Alert sends out a message that solicits a response from an e-mail mailbox. Recipients reply to the alert message, returning it to the response mailbox indicated in the original message's Reply To field. Oracle Alert then performs the following tasks depending on the response received:



- Valid response

- Oracle Alert processes the valid response and performs the actions associated with the response.
- Oracle Alert moves the messages from the Inbox to the REVIEWED\_OK folder.



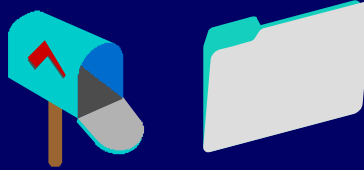
Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### How Response Processing Works

#### Invalid response

- If Oracle Alert does not recognize a response but the response has a message handle, Oracle Alert classifies it as an invalid response and performs the invalid response actions.
- Oracle Alert moves all invalid response messages with a message handle to the REVIEWED folder in the response mailbox.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### How Response Processing Works

- **No response after X days**
  - If on the X+1 day, Oracle Alert reviews the INBOX and still finds no response, it performs the No Response follow-up actions for the outstanding response.
- **Other messages**
  - If Oracle Alert reads a message in the INBOX and does not find a message handle, it moves the message to the REVIEWED folder.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



### Practice 7-1

This practice covers the following topics:

- Understanding response processing
- Defining responses and response actions

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 7-1 Instructions

Answer the following questions:

- 1 What is response processing?
- 2 What is a response variable?
- 3 What is a response set?
- 4 Where do you create response actions?
- 5 What are the possible types of responses?
- 6 What are response days?
- 7 Where do you link a response set to the alert?

### Practice 7-1 Solutions

- 1 What is response processing?

**Response processing is a feature that processes replies from alert recipients. A response processing alert sends a message that solicits a response from the recipient as its alert action. Oracle Alert automatically takes predefined actions based on the reply that the user returns. This feature helps eliminate operator intervention for routine transactions.**

- 2 What is a response variable?

**A variable whose value is specified by the message recipient.**

- 3 What is a response set?  
**The set of instructions for valid responses, invalid responses, and no responses to an alert message. A response set need not include invalid or no response actions.**
- 4 Where do you create response actions?  
**In the Actions window.**
- 5 What are the possible types of responses?  
**Valid, invalid, or no response.**
- 6 What are response days?  
**The number of days you want Oracle Alert to wait before performing the No Response follow-up actions.**
- 7 Where do you link a response set to the alert?  
**In the Action Details window.**

## Practice 7-2

---

### Practice 7-2

This practice covers the following topics:

- Defining an alert that uses response processing
- Responding to an alert message

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 7-2 Instructions

#### Scenario

In this practice, you will define an alert that solicits a response from a user and then uses response processing to either update or preserve the user's fax number in FND\_USER, based on the response.

You will need the following table and columns for this alert:

FND\_USER: EMAIL\_ADDRESS, FAX, USER\_ID, USER\_NAME

#### Step 1: Define the Alert

Define a periodic alert named *XX Fax Change* for the Oracle Alert application, and give the alert a description. Specify that the alert should be run on demand. Save the alert history for two days. Use the following SELECT statement:

```
SELECT email_address ,
       fax ,
       user_id ,
       user_name
INTO   &EMAIL ,
       &FAX ,
       &USERID ,
       &USERNAME
FROM   fnd_user
```

---

Copyright © Oracle Corporation, 2000. All rights reserved.

```
WHERE user_name = 'ALERTXX'
```

After you define your alert, verify your SELECT statement syntax, and run the SELECT statement to check how many records it retrieves.

### **Step 2: Define the Initial Action**

Define a detail action named Initial Message for the alert, and give the action a description. This action will be the initial action performed by the alert. Specify that the action should send a message to the e-mail address of the user retrieved by the SELECT statement. Write a subject line and message text for the message. The message text should include the user name and current fax number for the user, and should ask if the user wants to update the fax number.

### **Step 3: Define the Action Set**

Define an action set named Initial Message Action Set for the alert, and give the action set a description. Add the Initial Message action as a member of the action set, and specify that the action should be aborted if an error occurs during execution.

### **Step 4: Define the Response Set**

Define a response set named Fax Change for the alert, and give the response set a description. Define a response variable named NEWFAX for this response set, and give the response variable a description and a type of Character.

### **Step 5: Define the Response Actions**

Define three actions for the alert that Oracle Alert can perform after receiving a valid response from a user:

- Define a detail action named Update FND\_USER, and give the action a description. This action will be performed if the user responds with a new fax number. Specify that the action should run a SQL statement script to update the FAX column for the user in FND\_USER with the new fax number specified in the user's response.
- Define a detail action named Fax Change Confirmation Message, and give the action a description. This action will be performed if the user responds with a new fax number. Specify that the action should send a message to the e-mail address of the user retrieved by the SELECT statement. Write a subject line and message text for the message. The message text should confirm that the fax number has been changed and should include both the previous fax number and the new fax number.
- Define a detail action named No Fax Change Confirmation Message, and give the action a description. This action will be performed if the user responds that the fax number should not be changed. Specify that the action should send a message to the e-mail address of the user retrieved by the SELECT statement. Write a subject line and message text for the message. The message text should confirm that the fax number will not be changed.

### **Step 6: Define the Responses**

Define two valid responses for the Fax Change response set:

- Define a valid response named Change Fax with the following text:

Change my fax number to NEWFAX = "?"

Assign the actions Update FND\_USER and Fax Change Confirmation Message as the response actions for this valid response.

- Define a valid response named No Change with the following text:

Do not change my fax number

Assign the action No Fax Change Confirmation Message as the response action for this valid response.

Also, assign the action "Return original message" as the response action for invalid responses.

### **Step 7: Attach the Response Set to the Initial Action**

Attach the Fax Change response set to the Initial Message action. Specify that Oracle Alert should wait for responses for one day.

### **Step 8: Review System Options for Response Processing**

Review the Oracle Alert options related to response processing, including:

- Response Text
- Enable Response Processing: Verify that this option is set to Yes.
- Response Processor Start Time
- Response Processor End Time
- Response Processor Interval: Ensure that this option is set to a brief interval, such as 5 minutes.

### **Step 9: Verify Response Processor Activation**

Use the Schedule Alert Programs window to verify that the Response Processor is active, and activate it if it is not.

### **Step 10: Test the Alert**

Submit a periodic alert check for your alert. Use your mail system to check your e-mail account and view the message sent by Oracle Alert. Reply to the alert message, following the response instructions to change your fax number to a new number such as 650-555-6789.

Use your mail system to check your e-mail account again and view the confirmation message sent by Oracle Alert. Use the Users window in the System Administrator responsibility to verify that Oracle Alert updated your fax number.

Optionally check your alert a second time, and use your mail system to check your e-mail account and view the message sent by Oracle Alert. This time, however, reply to the alert message following the response instructions to indicate that you do not want to change your fax number. Use your mail system to check your e-mail account again and view the confirmation message sent by Oracle Alert.

### **Step 11: Review Alert History**

Review the history for your alert, including checks, exceptions and actions.

## **Practice 7-2 Solutions**

### Step 1: Define the Alert

- 1 Navigate to the Alerts window in the Alert Manager responsibility.  
Alert Manager: (N) Alert—>Define
- 2 Enter information to define the alert in the following fields:
  - Application: Oracle Alert
  - Name: XX Fax Change
  - Description: XX Fax Change
  - Enabled: Checked
- 3 Navigate to the Periodic region.
- 4 Enter the periodic detail information in the following field:
  - Frequency: On Demand
- 5 Enter 2 in the Keep\_Days field.
- 6 Enter the following SELECT statement in the Select Statement field:

```
SELECT email_address,  
       fax,  
       user_id,  
       user_name  
INTO   &EMAIL,  
       &FAX,  
       &USERID,  
       &USERNAME  
FROM   fnd_user  
WHERE  user_name = 'ALERTXX'
```
- 7 Save your work.
- 8 Click the Verify button to check your SQL for syntax errors.
- 9 Click the Run button to check how many records are retrieved by your SELECT statement.

### Step 2: Define the Initial Action

- 1 Navigate back to the Alerts window.
- 2 Click the Actions button to navigate to the Actions window.
- 3 Enter information to define the initial alert action in the following fields:
  - Action Name: Initial Message
  - Description: Initial message to user for fax number change
  - Action Level: Detail
- 4 Click the Action Details button to navigate to the Action Details window.
- 5 Enter the action detail information in the following fields:
  - Action Type: Message

- **To: &EMAIL**
  - **Subject: Do you want to update your fax number?**
- 6 Select the Text option, and enter the following message text:**
- Currently we have the following information for you:
- User name: &USERNAME
- Fax number: &FAX
- If you want to update your fax number, please reply with the new fax number.
- 7 Save your work.**

**Step 3: Define the Action Set**

- 1 Navigate back to the Alerts window.**
- 2 Click the Action Sets button to navigate to the Action Sets window.**
- 3 Enter information to define the action set in the following fields:**
  - **Seq: 1**
  - **Action Set Name: Initial Message Action Set**
  - **Description: Initial message action set**
  - **Enabled: Checked**
- 4 Navigate to the Members region.**
- 5 Enter information to define the action set member in the following fields:**
  - **Seq: 1**
  - **Action: Initial Message**
  - **On Error Action: Abort**
  - **Enabled: Checked**
- 6 Save your work.**

**Step 4: Define the Response Set**

- 1 Navigate back to the Alerts window.**
- 2 Click the Response Sets button to navigate to the Response Sets window.**
- 3 Enter information to define the response set in the following fields:**
  - **Response Set Name: Fax Change**
  - **Description: Response set for fax change**
  - **Enabled: Checked**
- 4 Save your work.**
- 5 Click the Response Variables button to navigate to the Response Variables window.**
- 6 Enter information to define the response variable in the following fields:**
  - **Variable Name: NEWFAX**

- **Description:** New fax number sent by responder
- **Type:** Character

7 Save your work.

#### **Step 5: Define the Response Actions**

1 Navigate back to the Alerts window.

2 Click the Actions button to navigate to the Actions window.

3 Enter information to define a new action in the following fields:

- **Action Name:** Update FND\_USER
- **Description:** Update user's fax number in FND\_USER
- **Action Level:** Detail

4 Click the Action Details button to navigate to the Action Details window.

5 Enter the action detail information in the following field:

- **Action Type:** SQL Statement Script

6 Select the Text option, and enter the following SQL statement script:

```
UPDATE FND_USER
SET     fax = '&NEWFAX'
WHERE  user_id = &USERID;
```

7 Save your work.

8 Navigate back to the Actions window.

9 Enter information to define another new action in the following fields:

- **Action Name:** Fax Change Confirmation Message
- **Description:** Confirmation of fax number change
- **Action Level:** Detail

10 Click the Action Details button to navigate to the Action Details window.

11 Enter the action detail information in the following fields:

- **Action Type:** Message
- **To:** &EMAIL
- **Subject:** Your fax number has been updated

12 Select the Text option, and enter the following message text:

```
Your fax number has been updated in FND_USER
from &FAX to &NEWFAX.
```

13 Save your work.

14 Navigate back to the Actions window.

15 Enter information to define another new action in the following fields:

- **Action Name:** No Fax Change Confirmation Message
- **Description:** Confirmation of no change to fax number
- **Action Level:** Detail



- 16 Click the **Action Details** button to navigate to the **Action Details** window.
- 17 Enter the action detail information in the following fields:
  - **Action Type:** Message
  - **To:** &EMAIL
  - **Subject:** Your fax number will not be changed
- 18 Select the **Text** option, and enter the following message text:  
Your response has been received, and your fax number in FND\_USER will not be changed.
- 19 Save your work.

**Step 6: Define the Responses**

- 1 Navigate back to the **Alerts** window.
- 2 Click the **Response Sets** button to navigate to the **Response Sets** window.
- 3 Select the **Fax Change** response set.
- 4 Click the **Valid** button to navigate to the **Valid Responses** window.
- 5 Navigate to the **Responses** region.
- 6 Enter information to define the first valid response in the following fields:
  - **Response Name:** Change Fax
  - **Text:** Change my fax number to NEWFAX = "?"
- 7 Navigate to the **Actions** region.
- 8 Enter information to define the first valid response action in the following fields:
  - **Seq:** 1
  - **Action:** Update FND\_USER
  - **Enabled:** Checked
- 9 Enter information to define the second valid response action in the following fields:
  - **Seq:** 2
  - **Action:** Fax Change Confirmation Message
  - **Enabled:** Checked
- 10 Save your work.
- 11 Navigate back to the **Responses** region.
- 12 Enter information to define the second valid response in the following fields:
  - **Response Name:** No Change
  - **Text:** Do not change my fax number
- 13 Navigate to the **Actions** region.

**14 Enter information to define the valid response action in the following fields:**

- **Seq: 1**
- **Action: No Fax Change Confirmation Message**
- **Enabled: Checked**

**15 Save your work.**

**16 Navigate back to the Response Sets window.**

**17 Click the Invalid button to navigate to the Invalid Response Actions window.**

**18 Enter information to define the invalid response action in the following fields:**

- **Seq: 1**
- **Action: Return original message**
- **Enabled: Checked**

**19 Save your work.**

#### **Step 7: Attach the Response Set to the Initial Action**

**1 Navigate back to the Alerts window.**

**2 Click the Actions button to navigate to the Actions window.**

**3 Select the Initial Message action.**

**4 Click the Action Details button to navigate to the Action Details window.**

**5 In the Response Set field, select the Fax Change response set from the list of values.**

**6 Enter 1 in the Response Days field.**

**7 Save your work.**

#### **Step 8: Review System Options for Response Processing**

**1 Navigate to the Oracle Alert Options window.**

**Alert Manager: (N) System—>Options**

**2 Navigate to the Message Elements region.**

**3 Select the Response Text message element.**

**4 Review the generic response text that Oracle Alert will append to the initial message action, explaining to the user how to respond.**

**5 Navigate to the Response Processing region.**

**6 Verify that the Enable Response Processing option is set to Yes.**

**7 Review the settings for the following options:**

- **Response Processor Start Time**
- **Response Processor End Time**
- **Response Processor Interval**

**Note: Normally, you would set the Response Processor Interval option to an appropriate interval for your business needs, such**

as 120 minutes. However, for the purposes of this practice, you should ensure that the **Response Processor Interval** option is set to a brief interval, such as 5 minutes, so that you can verify the results of your work without delay.

- 8 If you have made any changes to the system option settings, save your work.

#### **Step 9: Verify Response Processor Activation**

- 1 Navigate to the **Schedule Alert Programs** window.  
**Alert Manager: (N) Request—>Schedule**
- 2 Verify that the **Response Processor** program has a status of **Active**.  
If the **Response Processor** has a status of **Deactivated**, select it and click the **Activate** button. In the **Schedule Alert Programs** window that appears, leave the **Active Date** field blank and click the **OK** button to activate the **Response Processor** immediately.

#### **Step 10: Test the Alert**

- 1 Navigate to the **Request Periodic Alert Check** window.  
**Alert Manager: (N) Request—>Check**
- 2 Select the application and alert name you want to check. In this case, select the following:
  - **Application: Oracle Alert**
  - **Alert Name: XX Fax Change**
- 3 Click the **Submit Request** button to run the check. A **Note** window appears to inform you when the periodic alert is scheduled to run.
- 4 Click the **OK** button to close the **Note** window.
- 5 Note your request ID in the **Request Periodic Alert Check** window.
- 6 Navigate to the **Requests** window.  
**Alert Manager: (N) Request—>View (B) Find**
- 7 Monitor the request until it is completed.
- 8 Use your mail system to check your e-mail account and view the message sent by **Oracle Alert**.
- 9 Reply to the alert message, following the response instructions to change your fax number.  
Include the original message in your reply to ensure that the reply contains the message handle that **Oracle Alert** uses to identify the message. You can copy and paste the valid response text to ensure that your response matches the defined response text exactly.  
However, you should replace the question mark in your response with a new fax number, such as 650-555-6789.
- 10 Use your mail system to check your e-mail account again and view the confirmation message sent by **Oracle Alert**.
- 11 Navigate to the **Users** window in the **System Administrator** responsibility.

**System Administrator: (N) Security—>User—>Define**

- 12 Query your user.**
- 13 Verify that the Fax field contains the fax number updated by Oracle Alert.**
- 14 Optionally repeat steps 1 through 8 in this section to check your alert a second time. This time, however, reply to the alert message following the response instructions to indicate that you do not want to change your fax number.**
- 15 Use your mail system to check your e-mail account again and view the confirmation message sent by Oracle Alert.**

**Step 11: Review Alert History**

**After you finish testing your alert, review the alert history as described in practice 2-2. Specify the following application and alert name to review:**

- Application: Oracle Alert**
- Alert Name: XX Fax Change**

### Summary

- **Response processing is a feature that processes replies from alert recipients.**
- **A response processing alert sends a message that solicits a response from the recipient as its alert action.**
- **Oracle Alert takes actions based on valid, invalid, and no response actions of the user.**
- **The valid, invalid, and no response actions are defined in a response set.**
- **For Oracle Alert to perform response processing on an alert, you must attach your response set to a message action.**

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**



# **Identifying Implementation Issues and Responsibilities for Oracle Alert**

## **Chapter 8**

# Identifying Implementation Issues and Responsibilities for Oracle Alert

---

## Identifying Implementation Issues and Responsibilities for Oracle Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

---

Copyright © Oracle Corporation, 2000. All rights reserved.



### Objectives

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

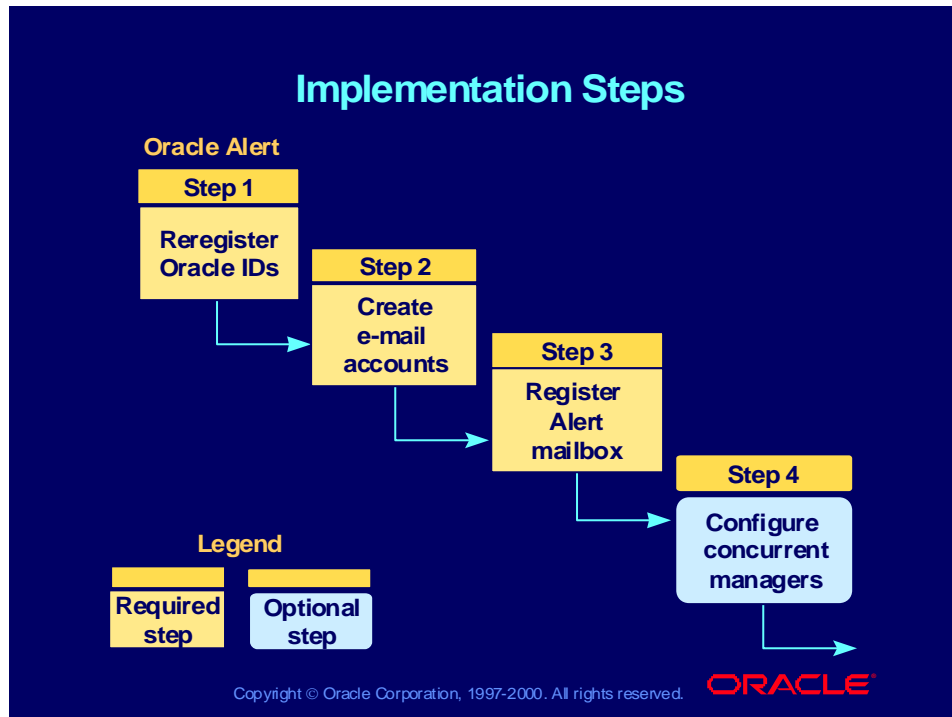
- **Plan a successful implementation of Oracle Alert**
- **Configure Oracle Alert for your business**
- **Identify your business needs and map them to responsibilities**
- **Determine whether you can use the predefined Oracle Alert Manager responsibility or create a new application Alert Manager responsibility**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

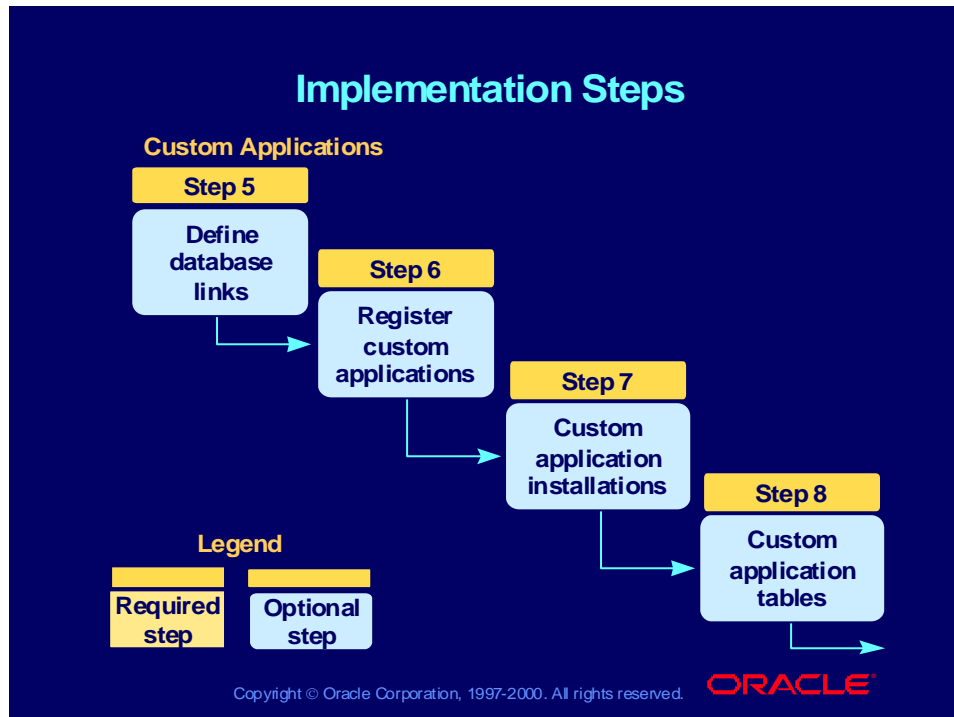
# Implementation Steps

---



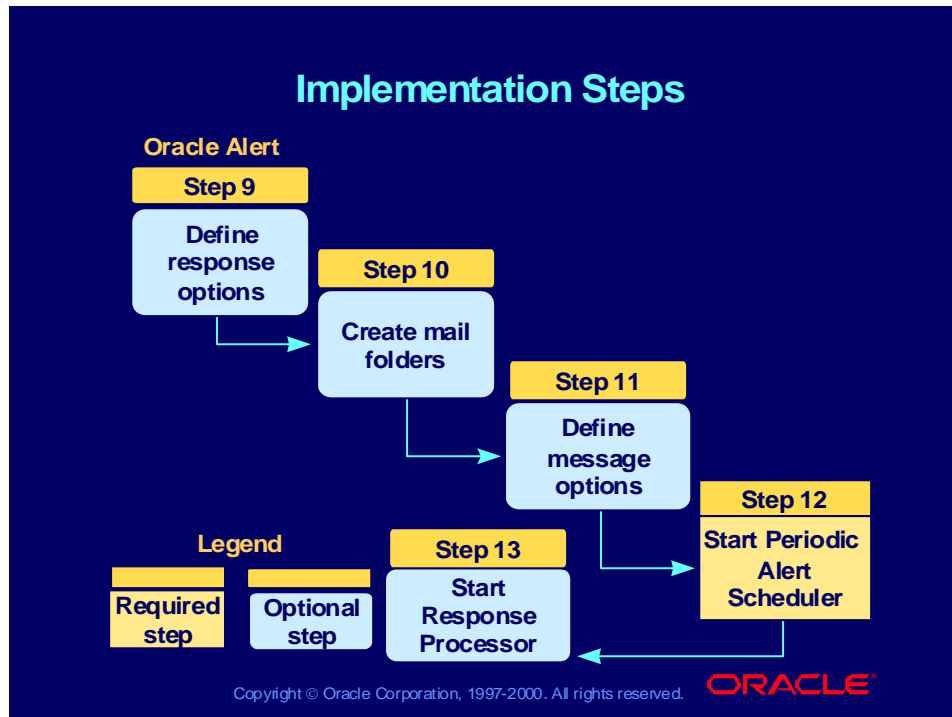
# Implementation Steps

---



# Implementation Steps

---



# Implementation Terminology

---

## Implementation Terminology

Term	Definition
Custom application	<p>Any Oracle Application residing in the same database as Oracle Alert that was not installed with Rapid Install</p> <p>or</p> <p>Any custom Oracle application residing in the same database as Oracle Alert</p> <p>or</p> <p>Any application residing in a different database from Oracle Alert</p>

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Implementation Terminology

---

### Implementation Terminology

Term	Definition
Database link	An object stored in the local database that identifies a remote database, a communication path to that database, and optionally, a username and password
Application installation	A unique combination of an application and an Oracle ID

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 1: Reregister Oracle IDs

---

### Step 1: Reregister Oracle IDs

After installing Oracle Alert, reregister any Oracle IDs that were not registered during installation and for which you want to create alerts.

- Include any custom Oracle IDs for which you want to create alerts.
- You do not need to reregister the APPLSYS Oracle ID.
- The Concurrent Manager must recognize these Oracle IDs.
- Oracle Alert must recognize these Oracle IDs in order to check alerts based on tables that reside outside its own Oracle ID.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 1: Reregister Oracle IDs

---

### Step 1: Reregister Oracle IDs

Use the Oracle Users window to enter:

- Database User Name
- Password
- Privilege
- Install Group
- Description

The Concurrent Manager executes the Grant and Synonym scripts for the Oracle IDs.

System Administrator: (N) Security—>ORACLE—>Register

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**



## Step 1: Reregister Oracle IDs

---

### Step 1: Reregister Oracle IDs

**Key implementation questions:**

- Are all Oracle IDs for which you want to create alerts registered?
- When you create new Oracle IDs, have you registered them?
- Have you registered all Oracle IDs for custom applications that will be using alerts?

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 2: Create Electronic Mail Accounts

---

### Step 2: Create Electronic Mail Accounts

Create an electronic mail account from which Oracle Alert can send mail.

- Define at least one electronic mail account as the Oracle Alert mailbox. You may want to name this account ALERT.
- This mail account is the sender of messages for alert message actions.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 2: Create Electronic Mail Accounts

---

### Step 2: Create Electronic Mail Accounts

You can define the account to send messages in any of the following mail systems:

- UNIX Sendmail
- VMS mail
- MAPI-compliant Windows NT mail application
- Any mail system supported by UNIX gateways



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE<sup>®</sup>

## Step 2: Create Electronic Mail Accounts

---

### Step 2: Create Electronic Mail Accounts

Define accounts to send and receive responses if you want to use response processing.

- Oracle Alert supports both UNIX Sendmail and Windows NT MAPI-compliant mail applications for response processing.
- You can use either the same account or different accounts to send messages and to process responses.
- Set up either one central response account, or a separate response account for each application.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 2: Create Electronic Mail Accounts

---

### Step 2: Create Electronic Mail Accounts

#### Key implementation questions:

- Is your e-mail application properly installed?
- Are the necessary gateways in place?
- Will you use response processing?
- Will you use response processing with more than one application?
- Do you need to enable decentralized administration of the response mail?
- Is response mail security an issue?



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 3: Register Oracle Alert Mailboxes

---

### Step 3: Register Oracle Alert Mailboxes

Register the electronic mail accounts you want to use with Oracle Alert.

- Identify the appropriate command to invoke the electronic mail system.
- Determine the number of mail accounts you want to use.
- Oracle Alert automatically displays two default mail accounts, the Send Mail Account and the Response Mail Account.
- You can define as many additional application-specific response accounts as you need.
- Identify the username and password of each mail account.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 3: Register Oracle Alert Mailboxes

---

### Step 3: Register Oracle Alert Mailboxes

Use the Mail Systems region of the Oracle Alert Options window to enter:

- Name
- Command
- Parameters
- In Use

(N) System—>Options

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 3: Register Oracle Alert Mailboxes

---

### Step 3: Register Oracle Alert Mailboxes

Use the Mail Server Options region of the Oracle Alert Options window to enter:

- Mail Database/Server Name
- Accounts
  - Application
  - Installation
  - User Name
  - Password

(N) System→Options

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Step 4: Configure Concurrent Managers

---

### Step 4: Configure Concurrent Managers

Your system administrator should configure concurrent managers to optimize event alert request handling.

- Define one concurrent manager to run only the Check Event Alert program (ALECTC).
- Define all other concurrent managers to run all other programs (except the Check Event Alert program).

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 4: Configure Concurrent Managers

---

### Step 4: Configure Concurrent Managers

#### Key implementation questions:

- Will you be defining many event alerts?
- Do you need event alerts to be processed immediately once an event is triggered?
- Can you meet your business needs using a periodic alert and :date\_last\_checked?

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 5: Define Database Links

---

### Step 5: Define Database Links

Create database links for applications where necessary.

- Define a database link if you want to define periodic alerts for an application that resides in a database outside the database where Oracle Alert is installed. (To define event alerts for an application, the application must reside in the same database where Oracle Alert is installed.)
- Define a database link for an Oracle Application if that application is installed in a different database than Oracle Alert.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 5: Define Database Links

---

### Step 5: Define Database Links

- Define a database link for an Oracle Application if that application is installed in the same database as Oracle Alert, but in a different Oracle Application Object Library context than Oracle Alert.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 5: Define Database Links

---

### Step 5: Define Database Links

**Key implementation questions:**

- **Do you want to use Oracle Alert with an application on a machine where Oracle Alert is not installed?**
- **Do you want to use Oracle Alert with an application on a database where Oracle Alert is not installed?**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 5: Define Database Links

---

### Step 5: Define Database Links

- **Syntax**

```
CREATE DATABASE LINK linkname  
CONNECT TO username IDENTIFIED BY password  
USING 'connect string';
```

- **Example**

```
CREATE DATABASE LINK BUGDB  
CONNECT TO SCOTT IDENTIFIED BY TIGER  
USING 'T:WRVMS:V60BUG';
```

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 6: Register Custom Applications

---

### Step 6: Register Custom Applications

Register any custom applications you want to use with Oracle Alert.

- You must register your custom application to tell Oracle Alert that your application exists.
- Custom applications include:
  - Oracle applications not installed using Rapid Install
  - Custom Oracle applications
  - Any applications residing in a different database from Oracle Alert

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 6: Register Custom Applications

---

### Step 6: Register Custom Applications

Use the Applications window to enter:

- Application
- Short Name
- Basepath: The environment variable that represents the basepath of your custom application so that Oracle Alert can locate your SQL statement script and operating system script external files
- Description

(N) System—>Applications

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Step 7: Define Custom Application Installations

---

### Step 7: Define Custom Application Installations

If you registered a custom application in the previous step, you must also register the custom application's installation so that Oracle Alert knows which Oracle ID to connect to.

- Register applications that reside in the same database as Oracle Alert and that are not installed with Rapid Install.
- Register installations of applications that reside in databases other than Oracle Alert's database.
- When you register a custom application that resides in another database, use Oracle Alert's Oracle ID.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 7: Define Custom Application Installations

---

### Step 7: Define Custom Application Installations

Use the Define Application Installations window to enter:

- Application
- Oracle Username
- Status

(N) System—>Installations

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 8: Register Custom Application Tables

---

### Step 8: Register Custom Application Tables

If you have a custom application with which you want to define event alerts, and you registered the application and application installation in the previous steps, you need to register the tables and columns in your custom application by calling the `AD_DD.register_table()` and `AD_DD.register_column()` APIs.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Step 8: Register Custom Application Tables

See Table Registration API, *Oracle Applications Developer's Guide*.

## Step 8: Register Custom Application Tables

---

### Step 8: Register Custom Application Tables

- This step is required for defining event alerts for non-Oracle applications that reside in the same database as Oracle Alert.
- This step informs Oracle Alert of the names of your custom application tables so that they appear in the of values for the Table field in the Alerts window for event alerts.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 9: Define Response Options

---

### Step 9: Define Response Options

Control how Oracle Alert processes responses by defining response options. Note that the response accounts and response mailboxes must already be defined as indicated in Steps 2 and 3.

- The Response Processor runs every 24 hours at midnight by default if these options are left blank:
  - Response Processor Start Time
  - Response Processor End Time
  - Response Processor Interval
- You can change the Response Processor Interval to process mail more frequently.
- If the Response Processor gets off schedule, it will always run at the next scheduled interval.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 9: Define Response Options

---

### Step 9: Define Response Options

Use the Response Processing region of the Oracle Alert Options window to enter:

- Enable Response Processing
- Check for Open Responses Before Reading Response Accounts
- Response Processor Start Time
- Response Processor End Time
- Response Processor Interval
- Oracle Alert Installation Number

(N) System—>Options

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 9: Define Response Options

---

### Step 9: Define Response Options

**Key implementation questions:**

- Do you want to use response processing?
- How often do you need responses to be processed?

Copyright © Oracle Corporation, 1997-2000. All rights reserved. **ORACLE**

## Step 10: Create Mail Folders for Response Accounts

---

### Step 10: Create Mail Folders for Response Accounts

Create two new folders, **Reviewed** and **Reviewed\_OK**, in your electronic mail response account.

- Create these two folders in every electronic mail response account that you define in the Oracle Alert Options window.
- Oracle Alert uses these folders to manage the responses it receives for different alert messages.
- Consult your electronic mail application documentation to learn how to create the folders.



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Step 11: Define Message Options

---

### Step 11: Define Message Options

Create generic message elements that Oracle Alert appends to the alert messages it sends.

- Define the Message Action Header and Message Action Footer to save time when defining message actions.
- Define the Response Text to clearly communicate reply instructions to message recipients.
- Define the Returned Message Header to let respondents know that their reply to an alert message was not understood and that they must reply again.
- Oracle Alert provides a generic version of the Response Text and Returned Message Header that you can use or edit.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 11: Define Message Options

---

### Step 11: Define Message Options

Oracle Alert inserts the generic message elements in the following positions:

- The Message Action Header is displayed at the beginning of the message text of every message.
- The Message Action Footer is displayed at the end of the message text—but before the response text—of every message.
- The Response Text is displayed at the end of each alert message that solicits a response.
- The Returned Message Header is displayed at the beginning of every Return Original Message invalid response message action.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 11: Define Message Options

---

### Step 11: Define Message Options

Use the **Message Elements** region of the **Oracle Alert Options** window to enter:

- **Message Action Header**
- **Message Action Footer**
- **Response Text**
- **Returned Message Header**

**(N) System**—>**Options**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 12: Start the Periodic Alert Scheduler

---

### Step 12: Start the Periodic Alert Scheduler

After you have successfully installed Oracle Alert, you must verify that the Periodic Alert Scheduler is active to enable automatic periodic alert submission.

- Select **Requests** from the **View** menu to find requests and display the **Requests** window.
- You should see a concurrent request for the program name **Periodic Alert Scheduler**. Its phase should be **Pending**.
- If the phase of the **Periodic Alert Scheduler** is not **Pending**, start the program from the **Schedule Alert Programs** window.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 12: Start the Periodic Alert Scheduler

---

### Step 12: Start the Periodic Alert Scheduler

Use the Schedule Alert Programs window to:

- Select the Periodic Alert Scheduler program.
- Click the Activate button.

Never deactivate the Periodic Alert Scheduler.

(N) Request → Schedule

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 12: Start the Periodic Alert Scheduler

---

### Step 12: Start the Periodic Alert Scheduler

The Periodic Alert Scheduler runs automatically every day at midnight and performs the following actions:

- Purges expired history
- Submits periodic alerts that are scheduled to be checked that day
- Resubmits itself to run for the next day

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

## Step 13: Start the Response Processor

---

### Step 13: Start the Response Processor

After you have successfully installed Oracle Alert, start the Response Processor so that Oracle Alert automatically reads and processes the mail in the response account.

- The Response Processor also submits itself once for each response account that you define in the Oracle Alert Options window.
- The Response Processor runs automatically every day at midnight, or according to the schedule you define in your response options.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Step 13: Start the Response Processor

---

### Step 13: Start the Response Processor

Use the Schedule Alert Programs window to:

- Select the Response Processor program
- Click the Activate button

(N) Request → Schedule

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE



## Step 13: Start the Response Processor

---

### Step 13: Start the Response Processor

- Each time the Response Processor runs, it moves mail from the Inbox folder to the Reviewed folder of the response account.
- The Response Processor then reads the messages and does the following:
  - Acts on valid responses, then moves those messages to the Reviewed\_OK folder
  - Acts on invalid responses, and leaves those messages in the Reviewed folder
  - Acts on a lack of response if the follow-up period has expired

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Planning Responsibilities for Oracle Alert

- **Identify your business needs:**
  - Who will be defining alerts?
  - Will the user writing alert **SELECT** statements be different from the user modifying alert message actions?
- **Will you use response processing?**
  - If so, do you want to decentralize administration of the mail accounts?
  - Will you divide the Alert responsibility by application?



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Oracle Alert Manager Responsibility

The standard Oracle Alert Manager responsibility includes:

- All alert windows
- All action windows
- Response set windows
- All history windows
- All definition windows
- Access to data and history for all applications
- Permission to submit periodic alerts on demand for any application



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

## Defining an Application Alert Manager Responsibility

---

### Defining an Application Alert Manager Responsibility

Use the Responsibilities window to define new Alert Manager responsibilities if you want to divide your alert responsibilities by application.

System Administrator: (N) Security→Responsibility→Define

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Application Alert Manager Responsibility

An Application Alert Manager responsibility includes:

- All alert windows
- All action windows
- Response set windows
- All history windows
- The Oracle Alert Options window with access to the response account for this application
- Access to data and history for only this application
- Permission to submit periodic alerts on demand for this application



Copyright © Oracle Corporation, 1997-2000. All rights reserved.

ORACLE

### Practice 8-1

**This practice covers the following topics:**

- **Identifying the items you need to define to implement Oracle Alert**
- **Understanding any additional considerations for implementing Oracle Alert**

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**

### Practice 8-1 Instructions

Using the following list, identify steps you will implement and any considerations or research you need to do first to accomplish these steps.

- 1 (Required) Reregister Oracle IDs.
- 2 (Required) Create an electronic mail account for Oracle Alert.
- 3 (Required) Define the Oracle Alert mailbox.
- 4 (Optional) Configure your concurrent managers.
- 5 (Optional) Define database links.
- 6 (Optional) Register custom applications.
- 7 (Optional) Define custom application installations.
- 8 (Optional) Register custom application tables.
- 9 (Optional) Define response accounts.
- 10 (Optional) Create Reviewed and Reviewed\_OK mail folders for response accounts.
- 11 (Optional) Define message options.
- 12 (Required) Start the Periodic Alert Scheduler.
- 13 (Optional) Start the Response Processor.
- 14 (Optional) Define alert responsibilities.

## Practice 8-1 Solutions

- 1 (Required) Reregister Oracle IDs.  
**Identify any Oracle IDs that were not registered during Alert's installation and for which you want to create alerts. This should include any custom Oracle IDs for which you want to create alerts.**
- 2 (Required) Create an electronic mail account for Oracle Alert.  
**Before you set up Oracle Alert, your Mail System Administrator must create at least one electronic mail account for the Oracle Alert mail user. We suggest that you name the account ALERT. This user is the originator of alert messages. If you are using Response Processing, you must set up the accounts through which Oracle Alert can process responses to alert messages. You can either use the default ALERT account, or you may define additional accounts to process responses to alert messages.**  
**Consult your electronic mail application documentation to learn how to create a mail account.**
- 3 (Required) Define the Oracle Alert mailbox.  
**Use the Oracle Alert Options window to tell Oracle Alert the name and password of the electronic mail account you want to use as the sender of alert messages, as well as the operating system command to start your mail system.**
- 4 (Optional) Configure your concurrent managers.  
**Have your system administrator configure your concurrent managers for optimal event alert request handling. Define one concurrent manager to run only the Check Event Alert program (ALECTC). Define all other concurrent managers to run all other programs except the Check Event Alert program.**
- 5 (Optional) Define database links.  
**If you want to use Oracle Alert with an application that resides on a database other than the database where Oracle Alert is installed, you can use SQL\*Net to connect Oracle Alert to that database. Define a database link that represents the Oracle User in the remote database. Refer to the *SQL\*Net User's Guide* for your protocol for additional information.**
- 6 (Optional) Register custom applications.  
**If you have a custom application with which you want to use Oracle Alert, use the Applications window to register the application with Oracle Alert.**
- 7 (Optional) Define custom application installations.  
**If you have a custom application with which you want to use Oracle Alert, and you registered the application in the previous**

**step, you need to define the application installation using the Define Application Installations window.**

- 8 (Optional) Register custom application tables.  
**If you have a custom application with which you want to define event alerts, and you registered the application and application installation in the previous steps, you need to register the tables and columns in your custom application by calling the AD\_DD.register\_table() and AD\_DD.register\_column() APIs.**
- 9 (Optional) Define response accounts.  
**Use the Oracle Alert Options window to tell Oracle Alert the names and passwords of the e-mail accounts you want to use to process responses. You can also define the Response Processor interval. All occurrences of the Response Processor for each response account use this interval.**
- 10 (Optional) Create Reviewed and Reviewed\_OK mail folders for response accounts.  
**Create two new folders called Reviewed and Reviewed\_OK in each response account that you specify in the Oracle Alert Options window. Consult your electronic mail application documentation to learn how to create a new mail folder.**
- 11 (Optional) Define message options.  
**You can use the Oracle Alert Options window to create generic message elements that Oracle Alert appends to the top or bottom of each alert message it sends.**
- 12 (Required) Start the Periodic Alert Scheduler.  
**You can start the Periodic Alert Scheduler using the Schedule Alert Programs window. Oracle Alert submits the Periodic Alert Scheduler to the Concurrent Manager. The Periodic Alert Scheduler then resubmits itself for the next day.**
- 13 (Optional) Start the Response Processor.  
**You can start Response Processor from the Schedule Alert Programs window. Oracle Alert submits the program Response Processor to the Concurrent Manager. That program then submits itself once for each response account you defined in the Oracle Alert Options window. These processes continue to resubmit themselves each day, or according to the interval you defined in the Oracle Alert Options window.**
- 14 (Optional) Define alert responsibilities.  
**A responsibility in Oracle Applications is a level of authority that determines how much of an application's functionality a user can use, what requests and concurrent programs the user can run, and which applications' data those requests and concurrent programs can access. Oracle Applications provides a set of predefined responsibilities that you can use. You can**



**also define your own responsibilities if the ones provided do not meet your needs. You may want to consider dividing the responsibilities.**

### Summary

- After Alert is installed, you need to reregister any Oracle IDs that were not registered during the Alert installation.
- Create at least one electronic mail account to originate alert messages.
- If you are using response processing, you must set up the accounts through which Oracle Alert can process responses to alert messages.
- To optimize alert request handling, define one concurrent manager to run only the Check Event Alert program and other concurrent managers to run all other programs.
- Create an implementation plan that considers whether any of the optional implementation steps are needed for your installation.

Copyright © Oracle Corporation, 1997-2000. All rights reserved.

**ORACLE**