Oracle Alert® User's Guide

Release 11

Part No. A56103–01

ORACLE®

Enabling the Information Age™

Oracle Alert User's Guide, Release 11

Part No. A56103–01

Copyright © 1997 Oracle Corporation

All rights reserved.

Primary Author: Siu Chang
Contributors: Dave Hung, Cathleen Trezza, Markus Tjahjadi

Contents

Preface

Welcome to Release 11 of the *Oracle Alert User's Guide.*

This user's guide includes the information you need to work with Oracle Alert effectively. It contains detailed information about the following:

- Overview and reference information
- Oracle Alert implementation suggestions
- Specific tasks you can accomplish using Oracle Alert
- How to use Oracle Alert windows
- Oracle Alert programs
- Oracle Alert functions and features
- Oracle Alert system setup

This preface explains how this user's guide is organized and introduces other sources of information that can help you.

About This User's Guide

This guide is the primary source of information about Oracle Alert. It contains overviews as well as task and reference information. This guide includes the following chapters:

- Chapter 1 gives you an overview of Oracle Alert and its major features.

- Chapter 2 describes the steps necessary to implement Oracle Alert.

- Chapter 3 describes how to define an event or a periodic alert.

- Chapter 4 tells you how to further customize the alerts you define so that they perform the exact kind of exception reporting your organization needs.

- Chapter 5 provides an overview of Oracle Alert history concepts and discusses ways you can use history to enhance your exception reporting capabilities.

- Chapter 6 describes how response processing works and provides a discussion of response processing concepts. This chapter also tells you how to define a response processing alert and how you can implement multiple Oracle Alert installations with response processing.

- Chapter 7 discusses how Oracle Alert checks alerts.

- Chapter 8 describes how you can set up your Oracle Alert installation to meet your organization's exception–reporting needs.

- Chapter 9 lists and describes the precoded alerts that are included in your Oracle Alert installation.

- Finally, appendix A includes information about the menu paths that help you navigate to Oracle Alert's various windows.

This user's guide is available online

All Oracle Applications user's guides are available online, in both HTML and Adobe Acrobat format. (Most other Oracle Applications documentation is available in Adobe Acrobat format.)

The paper and online versions of this manual have identical content; use whichever format is most convenient.

The HTML version of this book is optimized for onscreen reading, and lets you follow hypertext links for easy access to books across our

entire library. The HTML documentation is available from the Oracle Applications toolbar, or from a URL provided by your system administrator.

You can order an Oracle Applications Documentation Library CD containing Adobe Acrobat versions of each manual in the Oracle Applications documentation set. Using this CD, you can search for information, read it onscreen, and print individual pages, sections, or entire books. When you print from Adobe Acrobat, the resulting printouts look just like pages from an Oracle Applications hardcopy manual.

> **Note:** There may be additional material that was not available when this user's guide was printed. To learn if there is a documentation update for this product, look at the main menu on this product's HTML help.

Assumptions

This guide assumes you have a working knowledge of the principles and customary practices of your business area. It also assumes you are familiar with Oracle Alert. If you have never used Oracle Alert, we suggest you attend an Oracle Alert training class available through Oracle Education. (See Other Information Sources for more information about Oracle Alert and Oracle training.)

This guide also assumes that you are familiar with the Oracle Applications graphical user interface. To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide.*

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle provides powerful tools you can use to create, store, change, retrieve and maintain information in an Oracle database. But if you use Oracle tools like SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications forms, you may change a row in one

table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications forms to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. But, if you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

*Consequently, we STRONGLY RECOMMEND that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications tables, unless we tell you to do so in our manuals.*

Other Information Sources

You can choose from many sources of information, including documentation, training, and support services, to increase your knowledge and understanding of Oracle Alert.

Most Oracle Applications documentation is available in Adobe Acrobat format on the *Oracle Applications Documentation Library* CD. We supply this CD with every software shipment.

If this manual refers you to other Oracle Applications documentation, use only the Release 11 versions of those manuals unless we specify otherwise.

Oracle Applications User's Guide

This guide explains how to navigate, enter data, query, run reports, and introduces other basic features of the graphical user interface (GUI) available with this release of Oracle Alert. This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent requests.

You can also access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Oracle Applications Character Mode to GUI Menu Path Changes

This is a quick reference guide for experienced Oracle Applications end users migrating from character mode to a graphical user interface (GUI). This guide lists each character mode form and describes which GUI windows or functions replace it.

Oracle Applications Implementation Wizard User's Guide

If you are implementing more than one Oracle product, you can use the Oracle Applications Implementation Wizard to coordinate your setup activities. This guide describes how to use the wizard.

Installation and System Administration

Oracle Applications Installation Manual

This manual and the accompanying release notes provide information you need to successfully install Oracle Financials, Oracle Public Sector Financials, Oracle Manufacturing, or Oracle Human Resources in your specific hardware and operating system software environment.

Oracle Applications Upgrade Manual

This manual explains how to prepare your Oracle Applications products for an upgrade. It also contains information on finishing the upgrade procedure for each product. Refer to this manual and the *Oracle Applications Installation Manual* when you plan to upgrade your products.

Oracle Applications System Administrator's Guide

This manual provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage processing.

Oracle Alert Technical Reference Manual

The *Oracle Alert Technical Reference Manual* contains database diagrams and a detailed description of Oracle Alert and related applications database tables, forms, reports, and programs. This information helps you convert data from your existing applications, integrate Oracle Alert with non–Oracle applications, and write custom reports for Oracle Alert.

You can order a technical reference manual for any product you have licensed. Technical reference manuals are available in paper format only.

Other Information

Training

Oracle Education offers a complete set of training courses to help you and your staff master Oracle Applications. We can help you develop a training plan that provides thorough training for both your project team and your end users. We will work with you to organize courses appropriate to your job or area of responsibility.

Training professionals can show you how to plan your training throughout the implementation process so that the right amount of information is delivered to key people when they need it the most. You can attend courses at any one of our many Educational Centers, or you can arrange for our trainers to teach at your facility. In addition, we can tailor standard courses or develop custom courses to meet your needs.

Support

From on–site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Alert working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle server, and your hardware and software environment.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 45 software modules for financial management, supply chain management, manufacturing, project systems, human resources, and sales and service management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers, and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 140 countries around the world.

Thank You

Thank you for using Oracle Alert and this user's guide.

We value your comments and feedback. At the end of this manual is a Reader's Comment Form you can use to explain what you like or dislike about Oracle Alert or this user's guide. Mail your comments to the following address or call us directly at (650) 506–7000.

> Oracle Applications Documentation Manager
> Oracle Corporation
> 500 Oracle Parkway
> Redwood Shores, CA 94065
> U.S.A.

Or, send electronic mail to **appsdoc@us.oracle.com**.

Overview of Oracle Alert

This chapter gives you an overview of the major features of Oracle Alert, which:

- Keep you informed of critical activity in your database
- Deliver key information from your applications, in the format you choose
- Provide you with regular reports on your database information
- Automate system maintenance, and routine online tasks

Overview of Oracle Alert

Oracle Alert is your complete exception control solution.

How do you find out about important or unusual activity in your database? How do you stay aware of regular, yet critical database events? You don't have time to sort through reports, yet you need a reliable way to monitor your database.

Oracle Alert facilitates the flow of information within your organization by letting you create entities called *alerts* to monitor your business information and to notify you of the information you want. You can define one of two types of alerts: an *event alert* or a *periodic alert*.

An event alert immediately notifies you of activity in your database as it occurs. When you create an event alert, you specify the following:

- A database event that you want to monitor, that is, an insert and/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. An action can entail sending someone an electronic mail message, running a concurrent program, running an operating script, or running a SQL statement script. You include all the actions you want Oracle Alert to perform, in an action set.

A periodic alert, on the other hand, checks the database for information according to a schedule you define. When you create a periodic alert, you specify the following:

- A SQL Select statement that retrieves specific database information.

- The frequency that you want the periodic alert to run the SQL statement.

- Actions that you want Oracle Alert to perform once it runs the SQL statement. An action can entail sending the retrieved information to someone in an electronic mail message, running a concurrent program, running an operating script, or running a SQL statement script. You include all the actions you want Oracle Alert to perform, in an action set.

By creating event alerts, you can have an immediate view of the activity in your database, so you keep on top of important or unusual events as they happen. By creating periodic alerts, you can have current

measurements of staff and organization performance, so you can zero in on potential trouble spots. You can automate routine transactions, preserving your valuable time for more important issues. Oracle Alert gives you the information you need online, so you do not have to contend with a pile of paperwork.

Basic Business Needs

Oracle Alert gives you the flexibility you need to monitor your business information the way you want. An exception reporting system should:

- Keep you informed of database exception conditions, as they occur.

- Let you specify the exception conditions you want to know about, as often as you want to know about them.

- Keep you informed of exception conditions through a single point—your electronic mail.

- Take predefined actions when it finds exceptions in your database, without user intervention.

- Take the actions you specify, depending upon your response to an alert message.

- Perform routine database tasks automatically, according to the schedule you define.

- Be fully integrated with your electronic mail system.

- Send electronic mail messages to a distribution list you define, so you avoid typing the same set of recipients for many messages.

Major Features

Event Alerts

Event alerts immediately notify you of activity in your database as it happens. You define what a database event is—an insert or an update to a table—and Oracle Alert informs you when it happens. You can modify our precoded alert conditions or simply create your own, and Oracle Alert will send messages or perform predefined actions in an action set when important events occur.

Periodic Alerts

Periodic alerts periodically report key information according to a schedule you define. You can modify our precoded alerts or simply create your own, and Oracle Alert will send messages or perform predefined actions from an action set according to the schedule you set.

You can define periodic alerts on any Oracle Financials, Oracle Manufacturing, Oracle Human Resources, or Oracle Public Sector Financials application as well as any custom Oracle application.

Periodic alerts can be set to run as often as you need during a 24–hour period, or they can be set to run once a month—the frequency is up to you. Used over time, periodic alerts can provide a regular and reliable measure of performance.

For example, you can define a periodic alert for Oracle Purchasing that sends a message to the Purchasing Manager listing the number of approved requisition lines that each purchasing agent placed on purchase orders. You can define this alert to run weekly, and provide performance measurement on a consistent and timely basis.

Easy Alert Definition

Oracle Alert can load the SQL statement for your alert definition from an operating system file, allowing you to automatically perform the functions you currently do by hand. Oracle Alert will also transfer your entire alert definition across databases. You can instantly leverage the work done in one area to all your systems.

Customizable Alert Frequency

With Oracle Alert, you can choose the frequency of each periodic alert. You may want to check some alerts every day, some only once a month, still others only when you explicitly request them. You have the flexibility to monitor critical exceptions as frequently as necessary, even multiple times during a 24–hour period. You can also check less significant exceptions on a more infrequent schedule; for example, once a month.

Customizable Alert Actions

You can define a variety of actions for Oracle Alert to perform based on the exceptions it finds in your database. Oracle Alert can send an electronic mail message, run a SQL script or an operating system script, or submit a concurrent request, or any combination of the above. You can create your own message, SQL script, or operating system script

actions in Oracle Alert, or have Oracle Alert send messages or perform scripts that reside in external files. Each action is fully customizable to the exceptions found in your database, so you have complete flexibility in your exception management.

Detail or Summary Actions

You can choose 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 such that Oracle Alert performs that action for each individual exception found. You can also define a summary action such 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.

No Exception Actions

Oracle Alert can perform actions if it finds no exceptions in your database. You can define Oracle Alert to send electronic mail messages, run SQL scripts or operating system scripts, or submit concurrent requests, or any combination of the above.

Alert History

Oracle Alert can keep a record of the actions it takes and the exceptions it finds in your database, for as many days as you specify. When you ask Oracle Alert to reconstruct alert history you see a complete record of alert activity exactly as it was performed. You can even review all responses Oracle Alert received to your messages and the actions they invoked. Oracle Alert also lets you decide which information you want to review. You can narrow your review criteria so you see only the history you specifically want to examine, without sorting through all the history information available for an alert.

Duplicate Checking

Oracle Alert can search for exceptions that remain in your database over time, and can take certain actions based on the presence of those "duplicate exceptions." You can track exceptions in your database for the length of time that you save history for your alerts.

Action Escalation

You can define a sequence of actions and have Oracle Alert perform the next action in that sequence each time it finds the same exception or exceptions in your database. For example, you can have Oracle Alert send messages of increasing severity if it finds the same exceptions over a period of time. Using action escalation, you can make sure that exceptions needing attention don't languish unattended in your database.

Summary Threshold

Oracle Alert can automatically determine whether to perform a detail or a summary action based on the number of exceptions it finds in your database. If your alert locates few exceptions, it can simply perform detail actions—one for each exception. If your alert locates many exceptions, it can perform a summary action on all of those exceptions. Oracle Alert automatically determines when it should perform a detail or a summary action.

Response Processing

Oracle Alert can take certain predefined actions based on a user's response to an alert message. The response can cause Oracle Alert to send another alert message, run a SQL script or an operating system script, or submit a concurrent request, or any combination of the above. Because Oracle Alert performs response actions automatically, you can delegate routine user transactions to Oracle Alert and thereby increase your organization's efficiency.

Self–Referencing Alerts

You can create an alert that checks for exceptions that are new in your database since the last time the alert was checked. The alert uses its own DATE_LAST_CHECKED value as the start time for checking for new exceptions.

Customizable Options and User Profile

You can specify exactly how you want your Oracle Alert user interface to look and behave. From choosing a printer to specifying the header text in your Oracle Alert messages.

Electronic Mail Independence

Oracle Alert allows you to send electronic mail messages directly to your mail system. Connections to Oracle Office, Unix Mail, and VMS Mail are predefined for you.

☞ **Attention:** References to Oracle*Mail may be used interchangeably with Oracle Office in this guide. Oracle Alert is fully integrated with all Oracle mail products.

2

Implementing Oracle Alert

This chapter describes the steps necessary to implement Oracle Alert.

Implementing Oracle Alert

This section provides you with an overview of each task you need to complete to implement Oracle Alert quickly and easily. You should read this section with the Setting Up Oracle Alert chapter, which provides an overview and a detailed description of each Oracle Alert definition form. See: Setting Up Oracle Alert: page 8 – 2

Before you set up Oracle Alert, you should set up an Oracle Applications System Administrator responsibility. See: Setting Up Oracle Applications System Administrator: *Oracle Applications System Administrator's Guide.*

Setup Checklist

Complete the following steps to set up your installation of Oracle Alert:

❑ Step 1: Reregister ORACLE IDs (Required)

❑ Step 2: Create an Electronic Mail Account for Oracle Alert (Required)

❑ Step 3: Define the Oracle Alert Mailbox (Required)

❑ Step 4: Configure Your Concurrent Managers (Optional)

❑ Step 5: Define Database Links (Optional)

❑ Step 6: Register Custom Applications (Optional)

❑ Step 7: Define Custom Application Installations (Optional)

❑ Step 8 Register Custom Application Tables (Optional)

❑ Step 9: Define Response Accounts (Optional)

❑ Step 10: Create Reviewed and Reviewed_OK Mail Folders for Response Accounts (Optional)

❑ Step 11: Define Message Options (Optional)

❑ Step 12: Start the Periodic Alert Checker (Required)

❑ Step 13: Start the Response Processor (Optional)

Setup Flowchart

While you can set up Oracle Alert in many different ways, and defer optional set up steps until you are ready to use the corresponding functionality, we recommend you use the order suggested in the following flowchart:

Figure 2 – 1 Oracle Alert Setup
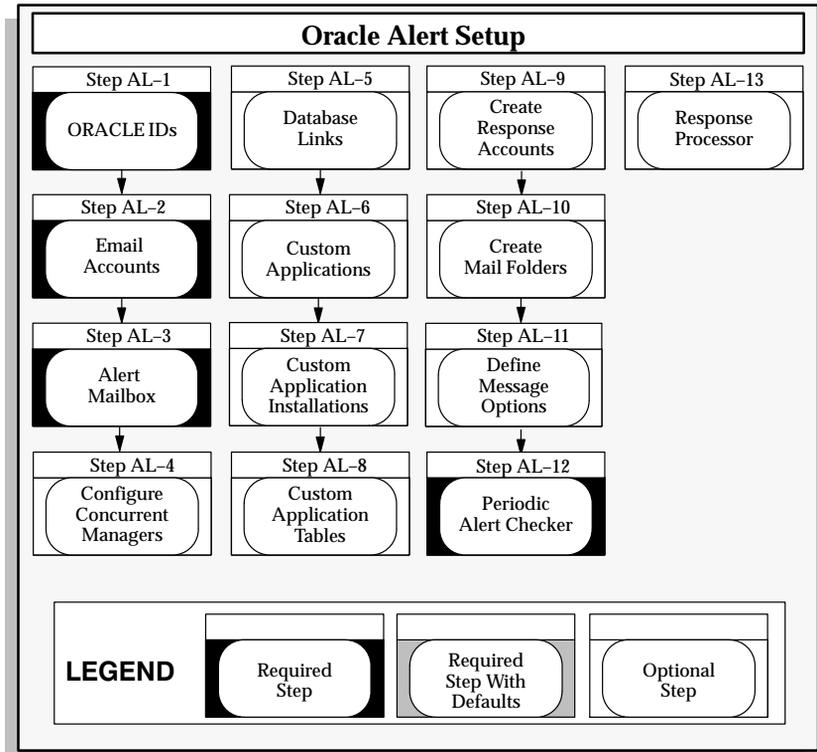
Setup Steps

Step 1   Reregister ORACLE IDs (Required)

After installation, reregister 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.   You need not reregister the APPLSYS ORACLE ID. See: Register ORACLE IDs: *Oracle Applications System Administrator's Guide.*

Step 2   Create an Electronic Mail Account for Oracle Alert (Required)

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 use Oracle InterOffice to set up the accounts through which Oracle Alert can process responses to alert messages.   You can either use the ALERT account set up through the Oracle InterOffice system, or you may define additional accounts to process responses to alert messages.

If you set up Oracle Alert on a single–task architecture such as VMS, you must install Oracle InterOffice on a separate database. This requires a SQL*Net installation. You must specify a SQL*Net alias for your Oracle InterOffice database name. Consult your Oracle InterOffice documentation to learn how to create a mail account.

Step 3   Define the Oracle Alert Mailbox (Required)

Use the Oracle Alert Options form 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 location of your Oracle InterOffice database (if you are using Oracle InterOffice).

If you are using a separate database for your Oracle InterOffice installation, you must specify the SQL*Net alias in the Oracle Office Options alternative region of the Oracle Alert Options form. See: Specifying Oracle Alert Options: page 8 – 3.

Step 4   Configure Your Concurrent Managers (Optional)

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. See: Specializing Managers to Run Only Certain Programs: *Oracle Applications System Administrator's Guide* and Define Concurrent Manager: *Oracle Applications System Administrator's Guide.*

Step 5   **Define Database Links (Optional)**

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.

☞   **Attention:**   You can only create Periodic Alerts on the remote database, not Event Alerts.

Step 6   **Register Custom Applications (Optional)**

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. See: Applications Window: page 8 – 21.

Step 7   **Define Custom Application Installations (Optional)**

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 form. See: Define Application Installations: page 8 – 24.

Step 8   **Register Custom Application Tables (Optional)**

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( ) API's. See: Table Registration API, *Oracle Applications Coding Standards.*

Step 9   **Define Response Accounts (Optional)**

Use the Oracle Alert Options form to tell Oracle Alert the names and passwords of the Oracle Office 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. See: Oracle Alert Options: page 8 – 3.

Step 10 **Create Reviewed and Reviewed_OK Mail Folders for Response Accounts (Optional)**

Create two new folders called Reviewed and Reviewed_OK in each Oracle InterOffice response account that you specify in the Oracle Alert Options form. Consult your Oracle InterOffice documentation to learn how to create a new mail folder.

Step 11 **Define Message Options (Optional)**

You can use the Oracle Alert Options form to create a generic message that Oracle Alert appends to the top or bottom of each alert message it sends. See: Oracle Alert Options: page 8 – 3.

Step 12 **Start the Periodic Alert Checker (Required)**

You can start the *Periodic Alert Checker* using the Schedule Alert Programs form. Oracle Alert submits the *Periodic Alert Checker* to the Concurrent Manager. The *Periodic Alert Checker* then resubmits itself for the next day. See: The Periodic Alert Scheduler: page 7 – 4.

Step 13 **Start the Response Processor (Optional)**

You can start Response Processor from the Schedule Alert Programs form. 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 form. These processes continue to resubmit themselves each day, or according to the interval you defined in the Oracle Alert Options form. See: The Response Processor: page 7 – 8.

Defining Alerts

This chapter tells you how to define alerts in Oracle Alert.

It includes a summary of major alert features and an explanation of the steps you complete to define both periodic and event alerts.

Overview of Alerts

You can create alerts that are as simple or as complex as you need them to be, and you can tailor your alerts in a variety of ways so they perform the kind of exception reporting your organization needs. There are two types of alerts: event and periodic. Both types are defined by a SQL Select statement that you specify.

You can create event alerts that monitor your applications for the exception conditions you specify. You can create periodic alerts that check your database for predefined conditions according to the schedule you determine.

Major Features

Verify SQL

You can verify that your alert's SQL Select statement runs correctly, and returns the data you specify. You can do this verification directly in Oracle Alert immediately after you enter your Select statement – you don't have to suspend your Oracle Alert session or navigate to SQL*Plus.

Specify Installations

You can specify which Application installations you want your alert to run against, so you can control which Application installations your alert checks in a database with multiple Application installations.

Information Routing

With Oracle Alert, you can include a file created by another application as part of an alert message. You can also define an alert that distributes an electronic copy of a report, log file, or any other ASCII file.

Dynamic Message Distributions

Oracle Alert lets you define a message distribution list without knowing ahead of time who the actual individuals on the list will be. Oracle Alert can retrieve the appropriate electronic mail IDs from your application tables, and send the message automatically.

Duplicate Suppression

Oracle Alert can automatically determine which action to perform based on whether it locates the same exceptions during a sequence of alert checks. You can have Oracle Alert perform a different action during each alert check that finds the same database exception.

Customizable Inputs by Action Set

You can further customize your alerts by specifying parameters for each set of actions you define. These parameters, or inputs, provide you with extra flexibility in creating your alerts because you can assign specific values to them. For example, if a vendor delivery is overdue, Oracle Alert can notify the purchasing agent when it is two days late and the purchasing manager when it is seven days late. The number of days late is the input; two and seven are distinct input values you assign for each type of recipient.

Distribution Lists

Oracle Alert lets you create an electronic distribution list that you can use on many messages. At any time, you can add or remove names from your lists, or you can make copies of your lists and use the copies to create new lists.

Standard Message Address Formats

Oracle Alert lets you address messages using easily recognizable symbols: to, cc, bcc, just as you would use when writing a memo or sending electronic mail.

Printed Alert Messages

Oracle Alert lets you send messages to people who do not use electronic mail. You can direct a message to a printer with the recipient's name on the burst page.

Electronic Mail Integration

Oracle Alert is fully integrated with Oracle Office, and uses Oracle Office gateways to process outgoing and incoming electronic mail messages of other electronic mail systems. Oracle Alert can also communicate directly with local electronic mail systems to send outgoing messages if the local sendmail command can accept arguments from Oracle Alert that consists of a filename that indicates

the body of the mail message and a recipient list. Examples of such mail systems include UNIX mail and VMS mail.

Creating a Periodic Alert

To create a periodic alert, you perform the following tasks in the order listed:

- Define your periodic alert and specify its frequency

- Specify the details for your alert

- Define actions for your alert

- Create action sets containing the actions you want your alert to perform

This section focuses on the first task of defining a periodic alert and its frequency and divides the task into smaller sub–tasks.

Before you define a periodic alert, make sure you do the following:

- Define an Oracle Office account for Oracle Alert. See: Setting Up Users and the Directory Service *(Oracle Office Administrator's Guide)*

- Set up a connection between Oracle Alert and Oracle Office. See: Specifying Oracle Alert Options: page 8 – 3.

▶ **To define a periodic alert:**

1. Navigate to the Alerts form.

2. Enter the name of the application that owns the alert in the Application field. This application must reside in the same Oracle database as Oracle Alert.

3. Name the alert (up to 50 characters), and give it a meaningful description (up to 240 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

    Enter a name for the alert that is unique within the application. Use an initial character other than a pound sign (#), a colon (:), or a percentage sign (%).

4. Select Periodic in the Type field.

5. Check Enabled to enable your periodic alert.

▶   **To set the frequency for a periodic alert:**

1. Select a frequency for your periodic alert. You can choose from nine frequency options:

    - **On Demand**—If you choose this frequency, Oracle Alert checks your periodic alert only when you specify. You do not have to fill in any other field in the Periodic Details block. You check on demand alerts by using the Request Periodic Alert Check window.

    - **On Day of the Month**—If you choose this frequency, Oracle Alert checks your alert on a monthly basis on the day number you enter in the Day field. 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.

    - **On Day of the Week**—If you choose this frequency, Oracle Alert checks your alert on the day of the week you enter in the Day field.

    - **Every N Calendar Days**—If you choose this frequency, enter a value in the Days field. Oracle Alert considers every day a calendar day, and does not skip holidays.

    - **Every Day**—Choosing this frequency is the same as choosing Every N Calendar Days and entering a value of 1 in the Days field.

    - **Every Other Day**—Choosing this frequency is the same as choosing Every N Calendar Days and entering a value of 2 in the Days field.

- **Every N Business Days**—If you choose this frequency, enter a value in the Days field. Oracle Alert lets you choose your business days, but does not skip any holidays. A value of 1 indicates that Oracle Alert should check the alert every business day, and a value of 2 indicates that Oracle Alert should check the alert every other business day. If you enter 3 in the Days field, Oracle Alert checks your periodic alert every three business days. For example, if you enable your alert on a Monday, Oracle Alert checks the alert first on that Monday, then on Thursday, then on the following Tuesday, then the following Friday, and so on.

- **Every Business Day**—Choosing this frequency is the same as choosing Every N Business Days and entering a value of 1 in the Days field.

- **Every Other Business Day**—Choosing this frequency is the same as choosing Every N Business Days and entering a value of 2 in the Days field.

You can also check all periodic alerts at any time, regardless of the frequency you assign them, using the Request Periodic Alert Check window. See: Checking a Periodic Alert: page 7 – 6.

2. Depending on the frequency you choose in the previous step, the Start Time and End Time fields become enabled. Enter 24–hour clock time values in these fields to specify when to start and/or end the periodic alert.

3. You may also specify the number of times within a 24–hour period that Oracle Alert checks your alert. Enter 24–hour clock time values in the Start Time, End Time and Check Interval fields.

   For example, suppose you want Oracle Alert to check your alert every 2 and a half hours between the hours of 9:00 A.M. and 7:00 P.M. Enter 09:00:00 in the Start Time field, 19:00:00 in the End Time field, and 02:30:00 in the Check Interval field.

   If you want Oracle Alert to check your alert just once a day, enter 00:00:00 in the Start Time field and leave End Time blank.

4. Specify a value in the Keep _ Days field to indicate the number of days of exceptions, actions, and response actions history you want to keep for this alert. See: Overview of Oracle Alert History: page 5 – 2

5. Specify a value in the End Date field if you want to disable your alert by a certain date. Note that you can enter a date in this field only if the alert is enabled.

In the Last Checked field, Oracle Alert displays the last date that this exception condition was checked, if it has been checked.

▶ **To enter a SQL Select statement for your periodic alert:**

1. Enter a SQL Select statement that retrieves all the data your alert needs to perform the actions you plan to define.

   Your periodic alert Select statement must include an INTO clause that contains one output for each column selected by your Select statement. Identify any inputs with a colon before the name, for example, :INPUT_NAME. Identify any outputs with an ampersand (&) before the name, for example, &OUTPUT_NAME. Do not use set operators in your Select statement.

   Suggestion: If you want to use an input value in an action for this alert, select the input into an output. Then you can use the output when you define actions for this alert.

   When selecting number columns, Oracle Alert uses the number formats defined in your database. Optionally, you can format your number outputs as real numbers by specifying a SQL*Plus format mask in your Select statement. For each number output, simply add a pound sign (#) and format mask to your output name. For example, if you select purchase price into the output &PRICE, add "#9999.99" after &PRICE for Oracle Alert to display the value to two decimal places. Your number output looks like: &PRICE#9999.99.

   Here is an example of a periodic alert Select statement that looks for users who have not changed their passwords within the number of days specified by the value in :THRESHOLD_DAYS.:

```
SELECT    user_name,
          password_date,
          :THRESHOLD_DAYS
INTO      &USER,
          &LASTDATE,
          &NUMDAYS
FROM      fnd_user
WHERE     sysdate = NVL(password_date,
          sysdate) + :THRESHOLD_DAYS
ORDER BY user_name
```

   Note: Although Oracle Alert does not support PL/SQL statements as the alert SQL statement definition, you can create a PL/SQL packaged function that contains PL/SQL logic and

enter a SQL Select statement that calls that packaged function. For example, you can enter a SQL Select statement that looks like:

```
SELECT package1.function1(:INPUT1, column1)
INTO &OUTPUT1
FROM table1
```

In this example, `package1` is the name of the PL/SQL package and `function1` is the name of user–defined PL/SQL function stored in the package.

To ensure that your PL/SQL function is callable from a SQL statement, verify that the packaged function meets the requirements listed in your *PL/SQL Release 2.1 and Oracle Precompilers Release 1.6 Addendum.*

☞ **Attention:** The SQL Select statement must be less than 64K.

2. Rather than typing in your Select statement, you may also import a SQL Select statement from a file in your operating system. Choose Import... and specify the location and name of the file that contains the SQL Select statement you want to import into your alert definition.

If the Select statement in the file does not contain an Into clause, Oracle Alert automatically inserts an Into clause into the alert Select statement. This default Into clause contains three outputs: &OUTPUT1, &OUTPUT2, and &OUTPUT3. You may want to give these outputs meaningful names.

If the Select statement in the file contains an Into clause, Oracle Alert preserves the Into clause. However, if the Into clause contains an incorrect number of outputs (the number of outputs does not match the number of columns you are selecting), Oracle Alert inserts the default Into clause into the alert Select statement.

☞ **Attention:** The file you import must be less than 64K.

▶ **To verify the SQL statement:**

1. You can verify the accuracy and effectiveness of your Select statement. Choose Verify to parse your Select statement and display the result in a Note window.

2. Choose Run to execute the Select statement in one of your application's Oracle IDs, and display the number of rows returned in a Note window.

3. Once you are satisfied with the SQL statement, save your work.

4. Navigate to the Alert Details window to complete the definition of your periodic alert. See: Specifying Alert Details: page 3 – 16.

See Also

How Oracle Alert Checks Alerts: page 7 – 2

Checking a Periodic Alert: page 7 – 6

Transferring Alert Definitions: page 3 – 39

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Deleting an Alert: page 3 – 38

Creating an Event Alert

To create an event alert, you perform the following tasks in the order listed:

- Define the database events that will trigger your alert
- Specify the details for your alert
- Define actions for your alert
- Create action sets containing the actions you want your alert to perform

This section focuses on the first task of defining the database events that trigger your event alert and divides the task into smaller sub–tasks.

Before you define an event alert, make sure you do the following:

- Define an Oracle Office account for Oracle Alert. See: Setting Up Users and the Directory Service (*Oracle Office Administrator's Guide*)
- Set up a connection between Oracle Alert and Oracle Office. See: Specifying Oracle Alert Options: page 8 – 3.

▶ **To define an event alert:**

1. Navigate to the Alerts form.

2. In the Application field, use the list of values to choose the name of the application that owns the alert. This application must reside in the same Oracle database as Oracle Alert.

3. Name the alert (up to 50 characters), and give it a meaningful description (up to 240 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

   Enter a name for the alert that is unique within the application. Use an initial character other than a pound sign (#), a colon (:), or a percentage sign (%).

4. In the Type field, choose Event.

5. Check Enabled to enable your event alert.

► **To specify an event table:**

1. Specify the name of the application and the database table that you want Oracle Alert to monitor.

   Although the application you enter here need not be the same application that owns the alert, both applications must reside in the same Oracle database and the application that owns the alert has to have Select privileges on the tables listed in the alert Select statement.

   ☞ **Attention:** You cannot use a view as the event table for your alert.

   ☞ **Attention:** Do not define an event alert on the table FND_CONCURRENT_REQUESTS. Oracle Alert submits a concurrent request to the concurrent manager when an event alert is triggered by an insert or update to an event table. For concurrent processing to occur, every submitted concurrent request automatically gets inserted as a row in the FND_CONCURRENT_REQUESTS table. If you define an event alert on this table, you create a situation where the event alert will cause an exception to occur recursively. See: How Oracle Alert Checks Alerts: page 7 – 2.

2. Check After Insert and/or After Update if you want to run your event alert when an application user inserts and/or updates a row in the database table.

3. Specify a value in the Keep _ Days field to indicate the number of days of exceptions, actions, and response actions history you want to keep for this alert. See: Overview of Oracle Alert History: page 5 – 2.

4. Specify a value in the End Date field if you want to disable your alert by a certain date. Note that you can enter a date in this field only if the alert is enabled.

In the Last Checked field, Oracle Alert displays the last date that this exception condition was checked, if it has been checked.

▶ **To enter a SQL Select statement for your event alert:**

1. Enter a SQL Select statement in the Select Statement field that retrieves all the data your alert needs to perform the actions you plan to define.

☞ **Attention:** The SQL Select statement must be less than 64K.

2. Your event alert Select statement must include an INTO clause that contains one output for each column selected by your Select statement.

Identify any inputs with a colon before the name, for example, :INPUT_NAME. Identify any outputs with an ampersand (&) before the name, for example, &OUTPUT_NAME. Do not use set operators in your Select statement.

💡 **Suggestion:** If you want to use an input value in an action for this alert, select the input into an output. Then you can use the output when you define actions for this alert.

When selecting number columns, Oracle Alert uses the number formats defined in your database. Optionally, you can format your number outputs as real numbers by specifying a SQL*Plus format mask in your Select statement. For each number output, simply add a pound sign (#) and format mask to your output name. For example, if you select purchase price into the output &PRICE, add "#9999.99" after &PRICE for Oracle Alert to display the value to two decimal places. Your number output looks like: &PRICE#9999.99.

3. Your event alert Select statement must also include a join from the table where the event transaction occurs to an implicit input we call :ROWID. Oracle Alert supplies the values for implicit inputs. Oracle Alert uses :ROWID to pinpoint the inserted or updated record in the table that caused the event.

☞ **Attention:** You must include :ROWID in your event alert's Select statement Where clause, or Oracle Alert will not be able to locate the row where the event transaction occurred.

Oracle Alert also automatically supplies the value for the implicit input :MAILID when triggering an event alert. The value for :MAILID is the electronic mail address of the application user who triggered the event. This value comes from the E–Mail field in the Users form of the System Administrator responsibility. If the E–Mail field is null, Oracle Alert uses the operating system user ID or the Application username of the user who triggered the event, depending on the value set for the Default User Mail Account option in the Oracle Alert Options form. If you want to send an alert message to that user, or use that user's email username in an action, simply select :MAILID into an output. Oracle Alert does not require that you use :MAILID in your Select statement.

☞ **Attention:** The inputs :MAILID and :ROWID are reserved terms that you should not use for anything other than their intended purposes.

Here is an example of an event alert Select statement that reports the creation of new users:

```
SELECT    user_name,
          :MAILID
INTO      &NEWUSER,
          &USER
FROM      fnd_user
WHERE     rowid = :ROWID
```

Note: Although Oracle Alert does not support PL/SQL statements as the alert SQL statement definition, you can create a PL/SQL packaged function that contains PL/SQL logic and enter a SQL Select statement that calls that packaged function. For example, you can enter a SQL Select statement that looks like:

```
SELECT package1.function1(:INPUT1, column1)
INTO &OUTPUT1
FROM table1
```

In this example, `package1` is the name of the PL/SQL package and `function1` is the name of user–defined PL/SQL function stored in the package.

To ensure that your PL/SQL function is callable from a SQL statement, verify that the packaged function meets the requirements listed in your *PL/SQL Release 2.1 and Oracle Precompilers Release 1.6 Addendum.*

4. Rather than typing in your Select statement, you may also import a SQL Select statement from a file in your operating system. Choose

Import... and specify the location and name of the file that contains the SQL Select statement you want to import into your alert definition.

If the Select statement in the file does not contain an Into clause, Oracle Alert automatically inserts an Into clause into the alert Select statement. This default Into clause contains three outputs: &OUTPUT1, &OUTPUT2, and &OUTPUT3. You may want to give these outputs meaningful names.

If the Select statement in the file contains an Into clause, Oracle Alert preserves the Into clause. However, if the Into clause contains an incorrect number of outputs (the number of outputs does not match the number of columns you are selecting), Oracle Alert inserts the default Into clause into the alert Select statement.

☞ **Attention:** The file you import must be less than 64K.

▶ **To verify the SQL statement:**

1. You can verify the accuracy and effectiveness of your Select statement. Choose Verify to parse your Select statement and display the result in a Note window.

2. Choose Run to execute the Select statement in one of your application's Oracle IDs, and display the number of rows returned in a Note window.

3. Once you are satisfied with the SQL statement, save your work.

4. Navigate to the Alert Details window to complete the definition of your event alert. See: Specifying Alert Details: page 3 – 16.

See Also

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Deleting an Alert: page 3 – 38

Transferring Alert Definitions: page 3 – 39

Specifying Alert Details

Once you define an event or periodic alert in the Alerts window, you need to display to the Alert Details window to complete the alert definition. The Alert Details window includes information such as which Application installations you want the alert to run against, what default values you want your inputs variables to use, and what additional characteristics you want your output variables to have.

▶ **To specify the alert details for an event or periodic alert:**

1. With your alert definition displayed in the Alerts form, choose Alert Details. The Alert Details window that appears contains three alternative regions: Installations, Inputs, and Outputs.

2. In the Inputs alternative region, Oracle Alert automatically displays the inputs used in your Select statement, unless they are the implicit inputs: :ROWID, :MAILID, and :DATE_LAST_CHECKED. The values of the implicit inputs are as follows:

   • **ROWID**—Contains the ID number of the row where the insert or update that triggers an event alert occurs.

   • **MAILID**—Contains the email username of the person who enters an insert or update that triggers an event alert.

   • **DATE_LAST_CHECKED**—Contains the date and time that the alert was most recently checked.

3. You can optionally add a description for each input, but 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 for inputs you enter in the Default Values field and in the
Action Set Inputs alternative region of the Action Sets block.

4. 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.

    For example, if you have an input APPLICATION_NAME, and you
    give it the default value "Oracle General Ledger," Oracle General
    Ledger will also be the value for the input in each action set you
    define, unless you change it in the action set to another value like
    "Oracle Payables."

    Note: When you change an input value in an action set, you
    simply change the value for that action set. The new value does
    not get propagated as a new default value for other occurrences
    of that input.

Alert Details - [New]					
Outputs					
		Max Length		Number	Check For
Output Name	Description	Detail	Summary	Format	Duplicates
					☒
					☐
					☐
					☐
					☐
					☐
					☐
					☐
					☐
					☐
					☐

5. In the Outputs alternative region, Oracle Alert automatically
    displays the outputs used in your alert Select statement without the
    ampersand (&) and any numbering format. You can add
    meaningful descriptions for the outputs.

6. You can specify the maximum number of output characters you
    want to display in your detail or summary message actions. See:
    Formatting Summary Message Actions: page 4 – 6.

7. If your output value is numeric, enter the SQL*Plus format mask in
    the Number Format field.

8. You can also check the Check for Duplicates check box to customize
    the combination of outputs you want to verify as a possible
    duplicate exception each time the alert is checked. A duplicate

exception is an exception that existed in your database during previous alert checks. You can define Oracle Alert to perform certain actions based on the presence of these duplicate exceptions.

Note: You must save history for your alert to use duplicate checking. See: Overview of Oracle Alert History: page 5 – 2.

	Alert Details - [New]

Installations

Oracle ID	Enabled
	☒
	☐
	☐
	☐
	☐
	☐
	☐
	☐
	☐
	☐
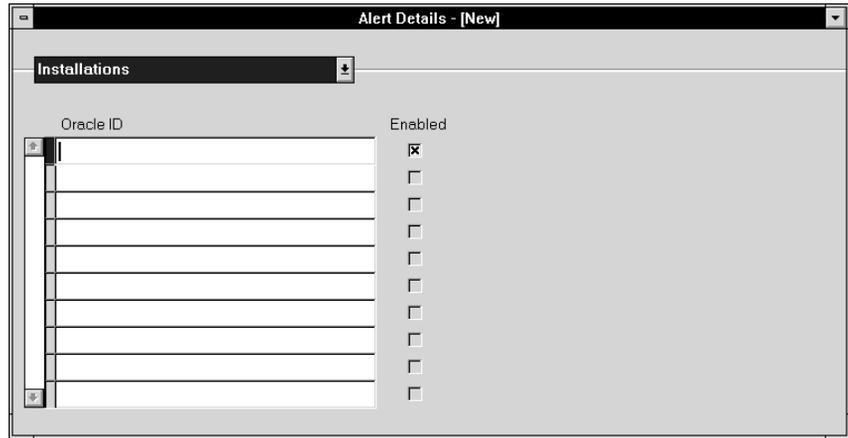
9. In the Installations alternative region, specify an Oracle ID if you want Oracle Alert to check your alert against that specific Oracle ID. You can select only those Oracle IDs that are associated with the application that owns your alert.

   If you do not specify an Oracle ID in this region, Oracle Alert checks your alert against all installations in your database of the application that owns your alert.

10. Check the Enabled field if you want your alert to run against that Oracle ID.

11. Save your changes and close the window.

Once you finish specifying the details for your alert definition, you need to create the actions for your alert. See: Creating Alert Actions: page 3 – 19.

Creating Alert Actions

After you define your alert you need to create the actions you want your alert to perform. There are four types of actions you can create:

- message actions

- concurrent program actions

- operating script actions

- SQL statement script actions

▶ **To create an action for an alert:**

1. Display the alert that you want to create an action for in the Alerts form.
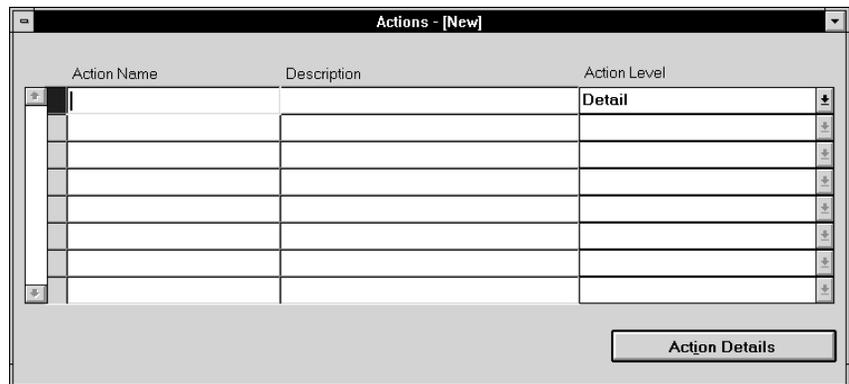
2. Choose Actions to display the Actions window.

3. Enter a name (up to 80 characters) and description (up to 240 characters) for your alert action. See: Oracle Alert Naming Conventions: page 8 – 19.

4. Select a level for your action: Detail, Summary, or No Exception.

During an alert check, a detail action performs once for each individual exception found, a summary action performs once for all exceptions found, and a no exception action performs when no exceptions are found.

5. Choose Action Details to display the Action Details window.

6. Select the type of action you want to create in the Action Type field: Message, Concurrent Program, Operating System Script, or SQL Script.

Depending upon the action level and type you choose, different fields appear in the Action Details window.

▶ **To define message actions:**

1. Choose Message in the Action Type field of the Action Details window if you want the alert to send a message as its action.

2. Specify the electronic mail IDs or Oracle Office public mail lists of the recipients you want to send your message to in the To: field. You can also enter the IDs of recipients to whom you want to send "carbon" (Cc) and "blind" (Bcc) copies of your alert message. If you list more than one recipient in any of these recipient fields, separate each recipient by a space, or a comma, or a combination of the two. You can enter as many recipients as you want, up to 240 characters.

> **Note:** If you integrate with Oracle Office to process your outgoing mail, Oracle Alert first validates with Oracle Office all the electronic mail IDs listed in the To, Cc, and Bcc fields before sending the alert message.

> If an ID is invalid in any of these fields and the action is part of a periodic alert, Oracle Alert returns the message to the <u>user who last updated the periodic alert.</u>

> If an ID is invalid in any of these fields and the action is part of an event alert or an on demand periodic alert, Oracle Alert returns the message to the <u>user who invoked the alert</u>.

In both cases, Oracle Alert determines the user's electronic mail ID from the value entered in the E–Mail field of the Users form (from the System Administrator responsibility). If the value of E–Mail is null, Oracle Alert then checks whether the value for the Default User Mail Account option listed in the Oracle Alert Options form is set to Operating System Login ID or Application Username and sends the message to the appropriate user using one of these two IDs.

This electronic mail ID validation is disabled if you integrate with other electronic mail systems to send outgoing messages.

3. You can also enter in the List field, a distribution list that you define in the Distribution Lists window. See: Defining a Distribution List: page 4 – 3.

    If you enter a distribution list name, Oracle Alert displays all electronic mail IDs on the list in the To field, and does not let you change any values in the To, Cc, Bcc, Print For, or Printer fields. If you want to use an Oracle Office public mail list, enter the list name in any of the recipient fields.

4. You can specify a value in the Reply To field that overrides the default Oracle Alert mail account that appears in the Reply To prompt of message actions. You can enter more than one Reply To mail ID, if appropriate. You can enter up to 240 characters.

    If you integrate with Oracle Office to process your outgoing messages, Oracle Alert uses Oracle Office to first validate the electronic mail IDs listed before sending out the alert message action.

5. You can also have Oracle Alert print your message to the printer you specify in the Printer field. Enter in the Print For User field, the names, or alert outputs or response variables that represent the names of the recipients for whom you want to print a copy of the message. Separate the names by commas and do not use blank spaces within a name.

    Oracle Alert prints the message preceded by a banner page with the name as it appears in the Print For User field. You can enter as many names as you want, up to 240 characters.

6. In the Subject field, enter a brief statement that describes the message's subject matter. You can enter up to 240 characters.

7. You can also enter alert outputs or response variables in any of the alert detail fields. Oracle Alert automatically substitutes the associated output value when checking the alert or the associated response variable value when reading the response. By entering an

output or response variable rather than a mail ID in a recipient field, you can make use of dynamic distribution. See: Defining a Response Processing Alert: page 6 – 4.

For example, if you define an alert that reminds users to change expired passwords, you can define an alert output called &USER to hold the user's electronic mail ID. Enter &USER in the To: field to have Oracle Alert send a message to each user whose password has expired.

8. For a detail message action, you can either send the contents of a file as the message, or write the text of your message in Oracle Alert. If you want to write your message in Oracle Alert, choose the Text option and enter the text in the Text field. You can include any outputs in the message text.

If you want to send the contents of a file, choose the File option and enter the file name (including the full path), or the output from the alert Select statement that represents the file name and location, in the File field. You may use any outputs to construct the file name.

   Note: If this message action has a response set associated with it, you must define the text of the message within Oracle Alert.

9. For detail message actions, Oracle Alert distributes one message for each exception found during an alert check. Place outputs wherever you need to in the message text, to format your message as you like. When Oracle Alert sends the message it substitutes the output variables with the exception values found during the alert check.

10. If this action is a response action, identify your response variables with an ampersand before the name, for example, &VARIABLE_NAME. Oracle Alert reads the value for a response variable from the response message. If the response message does not include the response variable, Oracle Alert uses the default value from your response variable definition.

11. For a no exception message action, Oracle Alert distributes a message when no exceptions are found for an alert check. Do not use outputs to define your no exception message and recipients.

   Note: If you decide to change the Action Level of your action after entering message text in the Text field, you must delete the message text yourself. Oracle Alert does not automatically delete the text for you.

☞ **Attention:** For detail or no exception message actions, the message text that you specify in Oracle Alert must be less than 2000 characters, while the message file that you specify must be less than 64K.

12. For summary message actions, complete your action details by following the instructions listed in the next section.

13. Save your changes.

14. When you are done defining actions, close the Actions and Action Details window and choose Action Sets to define the action set(s) for your alert definition. See: Creating an Action Set for an Alert: page 3 – 33.

▶ **To complete the action details for a summary message:**

1. Choose the Text option. A Summary Message Template appears in the text field.

2. You can place outputs anywhere in the three parts of the summary message body: above the summary template in the opening text, within the Summary Message Template, and after the summary template in the closing text.

3. Enter the opening text of your summary message at the top of the window above the summary template. Use this area to enter any introductory text that you want to precede the columns in your message and to lay out your column headings.

Suggestion: You can use any output values in the opening text of your summary message. Use outputs like current date (if you want to display the date that the exception was found) in the

introductory text of the message to avoid repeating the date as a separate column in the summary.

4. Leave the summary template lines exactly as you find them. Move the cursor between the two template lines and enter your outputs in this space, lined up below your column headings. Do not use spaces 1 and 2 within the summary template lines, as specified by the two asterisks (**); Oracle Alert reserves these to mark duplicate exceptions. You have up to 178 spaces across in which to lay out your outputs, and you can vary this value by specifying a number of characters in the Max Width field. (Oracle Alert defaults to the value of the Maximum Summary Message Width option defined in the Oracle Alert Options form.)

   The number of characters your message displays for each output depends on the number of characters you allot for each output in the Max Length field in the Outputs alternative region of the Alert Details window. The ampersand (&) of the output defines the left margin of an output's column area. Oracle Alert defines the right margin of an output's column area as two spaces to the left of the next column. Oracle Alert left–justifies any character or date outputs within that output's column area. For number outputs, Oracle Alert right–justifies the data within the column area. If the output in the rightmost column is defined as a number, Oracle Alert right–justifies the data in a column 15 characters wide. If the numeric data is wider than the column width allotted, Oracle Alert displays '###' in the message. See: Formatting Summary Message Actions: page 4 – 6.

5. If you want to use multiple lines per exception, position your outputs on as many lines as you need, in the positions you want. If you want a blank line between each set of exceptions, leave a blank line between your outputs and the lower summary template line.

6. Enter the closing text below the summary message template. You can use any outputs in your closing text.

   If you define a Message Action Header and/or a Message Action Footer in the Oracle Alert Options form, this generic text attaches to the top (before any message–specific text begins) and bottom (after any message–specific text ends) of your message text, respectively. If you define a header and footer, Oracle Alert adds it to every message constructed, including messages for which Oracle Alert is expecting a response. See: Defining Boilerplate Message Text: page 8 – 7.

7. Choose Wrap or Truncate in the Column Overflow field to indicate whether Oracle Alert should wrap column values that exceed the

space allotted to them or truncate them. If you choose Truncate, Oracle Alert truncates a column at the start of the next column, or at the page width (right margin), if it is the last column. See: Formatting Summary Message Actions: page 4 – 6.

8. Enter the width of your message in the Max Width field. The default value is defined by the Maximum Summary Message Width option listed in the Other Options alternative region of the Oracle Alert Options form. You cannot define a message wider than this maximum, but you can define a narrower message. Oracle Alert uses this value to determine the location of the right margin in the summary message.

9. Oracle Alert distributes one summary message for each unique combination of critical output values found during an alert check. Outputs that you place outside the Summary Message Template are considered critical outputs. Within the message, Oracle Alert summarizes all the exceptions found during an alert check for those outputs located within the Summary Message Template.

10. If this action is a response action, identify your response variables with an ampersand before the name, for example, &VARIABLE_NAME. Oracle Alert reads the value for a response variable from the response message. If the response message does not include the response variable, Oracle Alert uses the default value from your response variable definition.

> **Note:** If you decide to change the Action Level of your action after entering message text in the Text field, you must delete the message text yourself. Oracle Alert does not automatically delete the text for you.

☞ **Attention:** For a summary message actions, the message text that you enter in Oracle alert must be less than 2000 characters, while the message file that you specify must be less than 64K.

11. Save your changes.

12. When you are done defining actions, close the Actions and Action Details window and navigate to the Action Sets window to define action sets for your alert definition. See: Creating an Action Set for an Alert: page 3 – 33.

▶ **To define concurrent program actions:**

1. Choose Concurrent Program in the Action Type field of the Action Details window.

```
┌──────────────────────────────────────────────────────────────────────┐
│ ▭                    Action Details - [New]                         ▼ │
├──────────────────────────────────────────────────────────────────────┤
│   Action Type │Concurrent Program              │ ▣│                    │
│                                                                        │
│      Application │                                              │      │
│   Program Name │                                                │      │
│      Arguments │                                                │      │
│                                                                        │
└──────────────────────────────────────────────────────────────────────┘
```

2. Enter the name of the application that owns the concurrent program
    you want to define as an action.

3. Enter the name of the concurrent program.

4. Enter any arguments for the concurrent program in the Arguments
    field, each separated by a space.

    You can use output variables from the alert SQL statement or
    response variables to dynamically pass arguments to your
    concurrent program. If your concurrent program action uses
    arguments that select character or date data, place single quotes
    around your argument so that Oracle Alert can correctly pass the
    data.

5. If this action is a response action, identify your response variables
    with an ampersand before the name, for example,
    &VARIABLE_NAME. Oracle Alert reads the value for a response
    variable from the message response and reads the value of an alert
    output from the original alert message, then substitutes the value
    when sending the message. If the response message does not
    include the response variable, Oracle Alert uses the default value
    from your response variable definition.

6. For detail concurrent program actions, Oracle Alert submits the
    concurrent program request once for each exception found during
    an alert check.

7. For 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.

8. For 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.

9. Save your changes.

10. When you are done defining actions, close the Actions and Action
    Details window and navigate to the Action Sets window to define

action sets for your alert definition. See: Creating an Action Set for an Alert: page 3 – 33.

▶ **To define operating system script actions:**

1. Choose Operating System Script in the Action Type field of the Action Details window.

2. 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.

3. Enter any arguments you want to pass to the operating script in the Arguments field, each separated by a space.

   You can use output variables from the alert SQL statement or response variables to dynamically pass arguments to your operating system script. If your operating system script action uses arguments that select character or date data, place single quotes around your argument so that Oracle Alert can correctly pass the data.

4. Choose File if the operating system script resides in a file.

5. If the operating system script is in a file, and you do not specify a value in the Application field, as specified above, 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.

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

Note: The operating script file must be less than 64K.

6. If you want to enter the script in Oracle Alert, check Text and type the script in the Text field. The script must be less than 2000 characters.

   Identify your alert outputs with an ampersand before the name, for example, &OUTPUT_NAME.

7. If this action is a response action, identify your response variables with an ampersand before the name, for example, &VARIABLE_NAME. Oracle Alert reads the value for a response variable from the message response and reads the value of an alert output from the original alert message, then substitutes the value when sending the message. If the response message does not include the response variable, Oracle Alert uses the default value from your response variable definition.

8. For detail operating system script actions, Oracle Alert performs the operating system script once for each exception found during an alert check.

9. For summary operating script actions, Oracle Alert performs the operating system script once for each unique combination of critical output values returned by the alert Select statement. If the operating system script is found in a file, the outputs used as the file name and arguments are considered critical outputs. If the operating system script is found in the Text field, all outputs in the operating system script are considered critical outputs.

10. For 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.

   ☞ **Attention:** When performing an operating system action, Oracle Alert executes the commands with the privileges of the operating system user that started the Concurrent Manager.

11. Save your changes.

12. When you are done defining actions, close the Actions and Action Details window and navigate to the Action Sets window to define action sets for your alert definition. See: Creating an Action Set for an Alert: page 3 – 33.

▶ **To define SQL statement script actions:**

1. Choose SQL Statement Script in the Action Type field of the Action Details window.

2. If the SQL statement script you want to execute is in a file located in an Application's base path *sql* directory, specify that Application name in the Application field.

3. Enter any arguments you want to pass to the SQL statement script in the Arguments field, each separated by a space.

    You can use output variables from the alert SQL statement or response variables to dynamically pass arguments to your SQL statement script.

4. Choose File if the SQL statement script resides in a file.

5. If the SQL statement script is in a file, and you do not specify a value in the Application field, as indicated in above, then enter the full path of the SQL statement script, or the output from the alert SQL statement that represents the file name and location in the File field.

    Note: You can use PL/SQL for your SQL statement script action.

    Attention: The SQL script file must be less than 64K.

6. If you want to enter the script in Oracle Alert, check Text and type the SQL script in the Text field.

Identify your alert outputs with an ampersand before the name, for example, &OUTPUT_NAME. If your SQL statement script action uses outputs that select character or date data, place single quotes around your output so that Oracle Alert can correctly pass the data. If a single quote lies within your character string, Oracle Alert adds a second single quote. For example, if your character string is **resume'**, Oracle Alert displays this data as **resume"**. Follow SQL format conventions in the text of your SQL statement action, and place a semicolon (;) or a forward slash (/) at the end of each SQL statement.

☞ **Attention:** The SQL script you enter must have less than 2000 characters.

7. If this action is a response action, identify your response variables with an ampersand before the name, for example, &VARIABLE_NAME. Oracle Alert reads the value for a response variable from the message response and reads the value of an alert output from the original alert message, then substitutes the value when sending the message. If the response message does not include the response variable, Oracle Alert uses the default value from your response variable definition.

8. For detail SQL statement script actions, Oracle Alert performs the SQL statement script once for each exception found during an alert check.

9. For summary SQL statement script actions, Oracle Alert performs the SQL statement script once for each unique combination of critical output values returned by the alert Select statement. If the SQL statement script is found in a file, the outputs used as the file name and arguments are considered critical outputs. If the SQL statement script is found in the Text field, all outputs used in the SQL statement are considered critical outputs. See: Formatting Summary Message Actions: page 4 – 6.

10. For no exception SQL statement script actions, Oracle Alert performs the SQL statement script if it finds no exceptions during an alert check. Do not use outputs when defining your no exception SQL statement script actions.

11. Save your changes.

12. When you are done defining actions, close the Actions and Action Details window and navigate to the Action Sets window to define

action sets for your alert definition. See: Creating an Action Set for an Alert: page 3 – 33.

► **To delete an action:**

1. Select the action you wish to delete in the Actions window of the Alerts form.

2. Choose from the Edit menu, Delete Record.

   You can delete an action if there is no history for the action and the action is not an enabled member of an action set, response set, or action group (such as an escalation or threshold group).

3. If the action is an enabled member of an action group, action set, or response set, an error message followed by a References window appears.
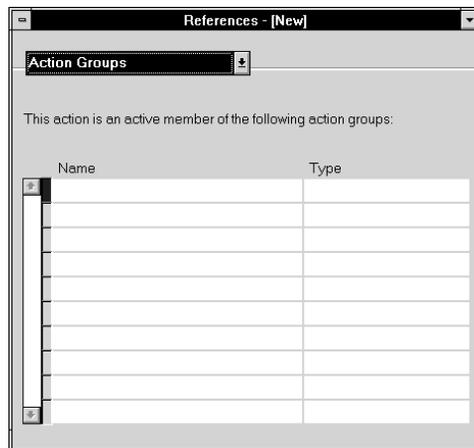
   You can also choose References from the Special menu in the Actions window at any time to display the References window.

4. The References window shows you where the action is referenced. Choose Action Groups, Action Sets, or Response Sets to display the list of action groups, action sets, or response sets that reference this action.

   You must navigate to the appropriate window and disable or remove the action from those< action groups, action sets, or response sets that reference the action before you can delete the action.

5. Once you delete your action, save your changes.

See Also

Defining a Response Processing Alert: page 6 – 4

Creating an Action Set for an Alert

Once you create your alert actions, you must include them in an enabled action set for Oracle Alert to perform during an alert check. An action set can include an unlimited number of actions and any combination of actions and action groups (such as escalation or threshold groups). You can define as many action sets as you want for each alert. Oracle Alert executes the alert Select statement once for each action set you define. During each action set check, Oracle Alert executes each action set member in the sequence you specify.

▶ **To create an action set for an alert:**

1. Display the alert that you want to create an action set for in the Alerts form.

2. Choose Action Sets.

3. In the Action Sets window that appears, enter a sequence number in the Seq field and name your action set (up to 30 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

   The sequence number lets you order the execution of action sets during an alert check. The sequence number automatically defaults to the next available number. You can accept or change the default.

4. Give the action set a meaningful description (up to 240 characters).

5. Check Suppress Duplicates if you want Oracle Alert to suppress the actions in this action set if the exception found is a duplicate that occurred during the last alert check. Note that you first have to save history to use this function.

If you check Suppress Duplicates and the action set includes a summary message action, Oracle Alert marks with an asterisk (*) any duplicate exceptions included in the summary message.

If you check Suppress Duplicates and the action set includes an escalation group whose members are detail actions, Oracle Alert performs the detail action that corresponds to the appropriate escalation level. Once Oracle Alert performs the action that corresponds to the highest escalation level, it ceases performing actions within that escalation group. If you uncheck Suppress Duplicates, then once Oracle Alert performs the action that corresponds to the highest escalation level, it continues to perform that same action during each alert check. See: Overview of Oracle Alert History: page 5 – 2.

> **Note:** Use the Check for Duplicates check box in the Outputs alternative regions of the Alert Details and Action Sets windows to tell Oracle Alert which outputs to consider when checking for duplicates.

6. Enable the action set.

7. Enter a value in End Date if you want to disable the action set by a certain date.

8. To add the actions, escalation groups, and/or threshold groups to your current action set, display the Members alternative region.

9. Assign each action set member a sequence number so that you can specify the order in which the members are performed.

Oracle Alert defaults the next available sequence number, but you can change the number if, for example, you want to re–sequence the actions in the action set. You can assign any sequence numbers you want, keeping in mind that Oracle Alert performs the actions so that two members with the same sequence number are performed in a random order.

10. Display the list of predefined actions, escalation groups, or threshold groups in the Action field to choose the members you want to include in the action set.

When you add a member to your action set, Oracle Alert automatically displays in the Type field, whether the member is an action, escalation group, or threshold group, and what its action

type is (message, concurrent request, operating script, or SQL statement script). Oracle Alert also indicates whether the member is enabled.

11. You can also assign additional actions for Oracle Alert to take if a particular action in an action set fails.

    If an action fails, you can choose to have Oracle Alert abort the action set, continue processing other action set members, or process another action set.

    If you select Set: in the On Error Action field, Oracle Alert leaves the current action set and performs the action set whose sequence number you specify in the Seq field when an error occurs. If you select Member:, Oracle Alert performs the action set member whose sequence number you specify in the Seq field.

12. Enter a value in End Date to disable the action set member by a certain date.

13. Display the Inputs alternative region.
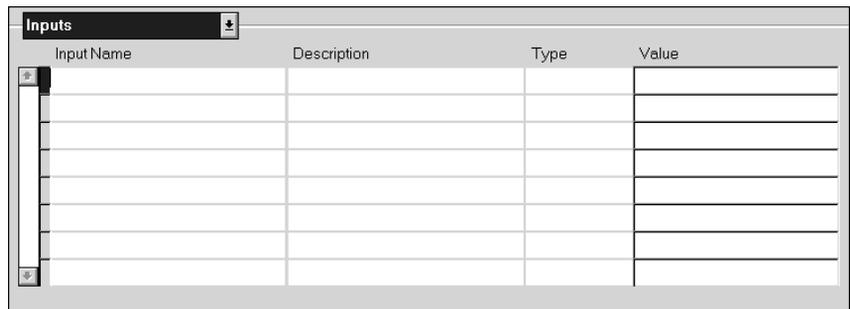
14. The input values you assign at the alert level become the default values for each action set. You can change the default input values for a particular action set by using the Inputs alternative region.

    Note: When you change an input value in an action set, you simply change the value in that action set. The new value does not get propagated as a new default value for other occurrences of that input.

15. Display the Outputs alternative region.

16. You can customize the type of duplicate output you want Oracle Alert to check for at the action set level by checking the Check for Duplicates check box. Indicate, as you did in the Outputs alternative region of the Alert Details window, the outputs you want Oracle Alert to consider when checking for duplicates.

For example, an alert showing overdue password changes might include three outputs: User Name, Date Password Changed, and Current Date. If you want Oracle Alert to consider only User Name and Date Password Changed when checking for duplicates, you uncheck the Check for Duplicates check box for the Current Date output. Since the value of Current Date will always be different each time the alert is checked, an exception would never be considered a duplicate if Current Date is included in the duplicate configuration.

> **Note:** The duplicate check configuration you define at the action set level overrides the duplicate configuration you define at the alert level.

17. Save your work. Your alert is ready for use.

▶ **To delete an action set:**

1. Select the action set you wish to delete in the Action Sets window of the Alerts form.

2. Choose from the Edit menu, Delete Record.

You can delete an action set if there is no history saved for the action set.

3. Save your work.

See Also

Overview of Oracle Alert History: page 5 – 2

Deleting an Alert

▶ **To delete an alert:**

1. Query the Alerts form for the alert you wish to delete.

2. Choose from the Edit menu, Delete Record.

   You can delete an alert only if there is no history, no enabled action sets, and no outstanding responses 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.

3. Save your work.

See Also

Overview of Oracle Alert History: page 5 – 2

Defining a Response Processing Alert: page 6 – 4

Transferring Alert Definitions

You can use the Transfer Alert Definitions window to transfer an alert definition to another database, or make a copy of an existing alert.

▶ **To transfer or copy an alert definition:**

1. Choose from the Special menu of the Alerts form, Transfer Alert.
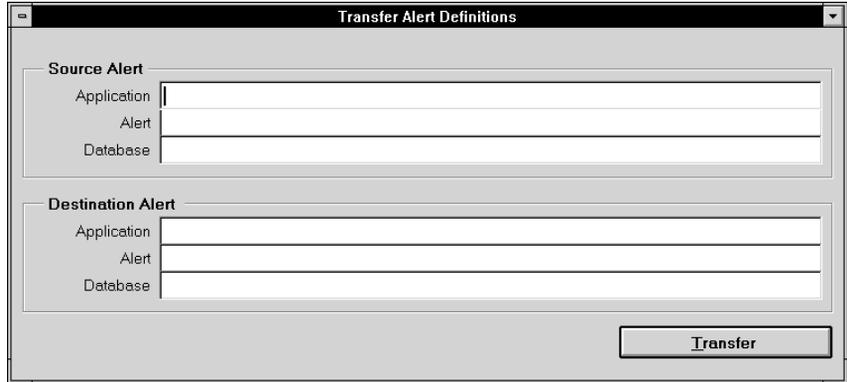
2. In the Source Alert region of the Transfer Alert Definition window, enter:

   • the name of the application that owns the alert you want to transfer

   • the name of the alert you want to transfer

   • the username and password of the database where the alert you want to transfer resides

   Also include any necessary SQL*Net syntax to indicate where your database resides. You may use SQL*Net V1 or SQL*Net V2 syntax. In the Database field, enter the username, password, and SQL*Net syntax in the following format:

   username/*password@SQL*Net_syntax*

   Here is an example of SQL*Net V1 syntax to specify the database location:

   ```
   sysadmin/friday@T:testing:db2
   ```

   Here is an example of SQL*Net V2 syntax to specify the database location:

   ```
   sysadmin/friday@db2
   ```

3.   Similarly, in the Destination Alert region, enter:

- the name of the application you want to transfer the alert to

- the name you want to assign to the new alert

- the database where you want the new alert to reside

4.   Choose Transfer to complete the alert transfer.

See Also

Oracle Alert Naming Conventions: page 8 – 19

4

Customizing Alerts

This chapter tells you how to further customize the alerts you define so that they perform the exact kind of exception reporting your organization needs.

Overview of Alert Customizations

Whether you create simple or complex alerts, you can further tailor the alerts to best suit your needs. For example, you can:

- create distribution lists to simplify the maintenance of your alert message recipients

- improve the format of your summary messages so that they are easier to read

- set a summary threshold so that you automatically get several detail messages or one summary message, depending on the number of exceptions an alert returns

- create a self–referencing alert to tell you what exceptions are unique since the alert was last checked

- create a periodic set so that you can check several periodic alerts at once

You can also implement other customizations, such as duplicate checking and action escalation, by saving alert history. See: Overview of Oracle Alert History: page 5 – 2.

Defining a Distribution List

Distribution lists let you predefine 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.

▶ **To define a distribution list:**

1. Navigate to the Distribution Lists form.

2. Enter in the Application field, the name of the application that owns the distribution list.

3. Enter a name for the distribution list that is unique within the application (up to 30 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

4. Enter a description of the distribution list (up to 240 characters).

5. In the Mail Recipients region, enter the electronic mail IDs of the recipients, separated by a space, or a comma (,), or a combination of the two. You can enter as many mail IDs as you want (up to 240 characters).

   The To field lets you list primary recipients. The Cc field lets you list recipients to whom you want to send a copy of the message. The Bcc field lets you list recipients to whom you want to send a blind copy of the message, that is, recipients in the To and Cc field cannot see who the recipients listed in the Bcc field are.

   Suggestion: We do not recommend using alert outputs in a distribution list. Since 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 could result in an error when used in another alert.

6. In the Print Options region, enter in the For User field, the names of the recipients for whom you want to print a copy of a message. Separate the names with commas and do not use blank spaces within a single name. You can enter as many names as you like, up to 240 characters.

7. Enter in the Printer field, the name of the printer to which you want Oracle Alert to direct the message. For each name you specify in the For User field, the printer prints a copy of the message preceded by a burst page with the name specified.

▶ **To delete a distribution list:**

1. Display the existing distribution list in the Distribution Lists form.

2. Choose Delete Record from the Edit menu.

3. If the distribution list is currently used in a message action, an error message followed by a References window appears.

You can also choose References from the Special menu in the Distribution Lists window at any time to display the References window.
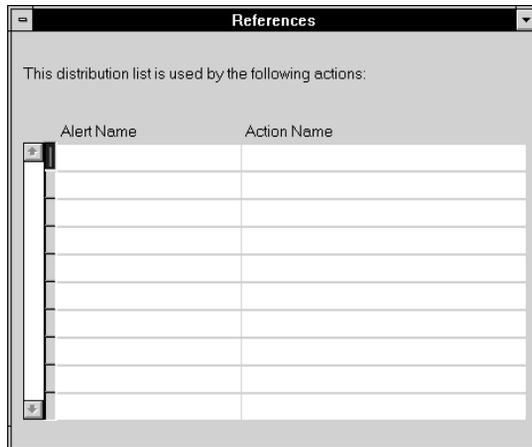
4. The References window shows you which actions reference the current distribution list.

You must navigate to the Action Details window of the Alerts form and remove the distribution list from those actions that reference it before you can delete the distribution list.

5. Once you delete your distribution list, save your changes.

Formatting Summary Message Actions

Oracle Alert can perform a summary message action on a combination of the exceptions found during an alert check, rather than on each exception found. You determine the combination of exceptions, and define the summary message action you want Oracle Alert to perform for each combination. You can take advantage of sophisticated formatting options in summary message actions and create alert messages that look exactly how you want them to look.

You have complete control over the layout of your summary message. When you create your alert you may need a particular message format; for example, you may want to recreate the layout of a purchase order in your alert message. You can define your summary message action so that Oracle Alert sends a separate summary message for each purchase order found during an alert check.

You specify the format of a summary message in two Oracle Alert windows:

- In the Outputs alternative region of the Alert Details window in the Alerts form

- In the Action Details block of the Actions window in the Alerts form

▶ **To format summary message actions:**

1. Display the alert that you want to format summary actions for in the Alerts form.

2. Choose Alert Details.

3. In the Outputs alternative region, specify the maximum number of characters that you want Oracle Alert to display for each output used in your alert. You indicate how many characters of data you want Oracle Alert to display for each output in detail actions, and how many characters you want Oracle Alert to display in summary actions.

4. Save your changes.

5. Choose Actions in the Alerts form.

6. Select the summary message action you want to format.

7. Choose Action Details in the Actions window.

8. Enter a value in the Max Width field to specify the maximum width of your message. Select 80 for portrait format, 132 for landscape format, and 180 for compressed landscape format. The default is 80.

9. Choose Truncate or Wrap in the Column Overflow field to specify whether you want Oracle Alert to wrap exception data that exceeds your defined column widths.

   A "column" in a message action is the space you allot for Oracle Alert to display the data it substitutes for an output. Use column wrap so that you can display large amounts of data within your summary message.

   When Oracle Alert constructs a message, each column starts at the ampersand (&) of the output name and ends two characters before the next output. Oracle Alert always leaves one blank space between columns.

   Oracle Alert considers the rightmost column in an alert message to start with the ampersand of the output name and end with the last character allowed by the defined message width. If the last column contains numeric data, the width of that column can be a maximum of 15 characters. If you want a numeric column to be wider than 15 characters, place the column in a location other than the last column of the message.

Note: Oracle Alert does not wrap numeric data within a column. If the number cannot fit within the column, Oracle Alert displays "###" across the width of the column. You should expand the width of this column within the message definition.

Oracle Alert ignores any columns that you place outside the defined message width. For example, if you define a message width of 132 characters, then place an output beginning in column 140, Oracle Alert ignores it and does not include that information in the message.

Suppose you have an output with a defined maximum length of 60, and you place the output in the Summary Message Template. You format your summary message such that there are 21 characters between the first character of the output and first character of the next output, keeping in mind that Oracle Alert always leaves one blank space between columns. During an alert check, the exception data that Oracle Alert substitutes for the output is more than 60 characters long. Because you have turned column wrap on, Oracle Alert displays 60 characters of the data on three lines of 20 characters each. Oracle Alert "wraps" the data so that 60 characters of it is displayed in your summary message.

10. Constant text is any non–output text that you include in the summary message. You can place constant text within the summary message template instead of, or in addition to, any column headings you include outside the summary template. For example, you may want to precede a money column with "$," or you may want to label a column "Purchase order line:".

11. Oracle Alert automatically prints your summary messages in an appropriate format. If a message is narrower than 80 characters, Oracle Alert prints it in a standard portrait orientation. If a message is between 81 and 132 characters wide, Oracle Alert prints it in landscape orientation, with the 132 columns along the long edge of the paper. If a message is between 133 and 180 characters wide, Oracle Alert also prints it in a landscape orientation, but compresses the font or character size. You never need to worry about not being able to see the information in your messages.

12. Save your changes.

Example

Suppose that you are creating an alert that sends a message to the responsible purchasing agent informing that agent of unapproved

purchase orders. You have several possible message formats you can define.

The Select Statement

Your alert Select statement looks like this:

```
SELECT    h.po_number,
          v.vendor,
          h.description,
          a.email_address,
          l.line_number,
          l.item,
          l.quantity,
          l.price
INTO      &PO,
          &VENDOR,
          &DESC,
          &AGENT,
          &LINE,
          &ITEM,
          &QTY,
          &PRICE
FROM      po_headers h,
          po_lines l,
          po_vendors v,
          po_agents a
WHERE     h.header_id       =       l.header_id,
          h.agent_id        =       a.agent_id,
          h.vendor_id       =       v.vendor_id,
          h.approved_flag   =       'N'
```

Your Summary Message Format

You can define your summary message to include all of a purchasing agent's purchase orders into one summary message, or you can send a separate summary message for each purchase order.

For a single summary message per purchasing agent, you can define your summary message action as shown below. Notice the use of constant text within the summary message template.

```
To: &AGENT
Cc:
Bcc:
Subject: Your unapproved purchase orders
```

```
=**= Enter summary template below this line =**=
** PO Number:    &PO    Vendor:       &VENDOR
   Description: &DESC
   Line:         &LINE  Item:         &ITEM
   Quantity:     &QTY   Price:        $&PRICE

=**= Enter summary template above this line =**=
```

Using this definition, Oracle Alert will send one message to each purchasing agent that lists that agent's unapproved purchase orders. Each exception that Oracle Alert lists in the message will be an individual purchase order line from an unapproved purchase order.

If you want to send a separate summary message to the purchasing agent for each unapproved purchase order, you can define your summary message action like this:

```
To: &AGENT
Cc:
Bcc:
Subject: Unapproved purchase order &PO

   PO Number:    &PO    Vendor:       &VENDOR
   Description: &DESC

   Line    Item Description          Quantity      Price
   ----    ----------------          --------      -----
=**= Enter summary template below this line =**=
** &LINE  &ITEM                      &QTY          $&PRICE
=**= Enter summary template above this line =**=
```

In this example, Oracle Alert constructs a distinct message for each unapproved purchase order. That is, Oracle Alert constructs a separate message for each *unique combination* of exceptions substituted for the outputs you place *outside* the summary message template. You can change the construction of the message simply by moving an output inside or outside the summary message template.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Creating Alert Actions: page 3 – 19

Overview of Summary Threshold

Oracle Alert can automatically determine whether to perform a detail or summary action, depending upon the number of exceptions found by the alert Select statement. If you define a summary threshold, Oracle Alert performs a detail action for each exception found by the Select statement, but if the number of exceptions found exceeds the summary threshold, Oracle Alert performs a summary action. You need to first define a detail and a summary action, include them in a threshold group, and then specify a summary threshold.

You may find that defining a summary threshold is useful for your alerts that return varying amounts of data from your database. If your alert typically returns few exceptions, but on occasion returns many exceptions, Oracle Alert can automatically switch from performing a detail action for each exception to performing a summary action for each group of exceptions. Oracle Alert performs the most efficient alert action automatically, without requiring further intervention from you.

Defining a Summary Threshold

To define a summary threshold, you first create an alert in the Alerts form and define one detail action and one summary action for the alert in the Actions window of the Alerts form.

▶ **To define a summary threshold for your alert:**

1. Navigate to the Alerts form and create a new alert or display an existing alert that you want to define a summary threshold for. See: Overview of Alerts: page 3 – 2.

2. In the Alerts form, choose Actions.

3. In the Actions window, define a detail action and a summary action that you want Oracle Alert to perform depending on the number of exceptions it finds for your alert. You will add these two actions to a threshold group. You can define message, SQL statement script, operating system script, or concurrent program request actions, but both detail and summary actions must be of the same action type.

4. In the Alerts form, choose Threshold Groups from the Special menu.

5. In the Threshold Groups window, give your threshold group a name (up to 80 characters) and description (up to 240 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

6. In the Type field, choose the type of action you want to use in the group (Message, SQL Statement Script, Operating System Script, or Concurrent Program).

7. In the Action field of the Threshold Group Members block, display a list of the predefined actions for this alert. Choose a detail action and a summary action to include in this threshold group.

   You can choose only actions created for this alert with the same action type as the group.

8. Check the Enabled field to enable the action in the threshold group.

9. Enter a current or future date in the End Date field if you want to disable an action in the threshold group by a certain date.

10. Choose Action Sets from the Alerts window.

11. In the Action field of the Members alternative region, enter the threshold group you want to include in your alert's action set.

12. In the Summary Threshold field, enter a number that represents the maximum number of exceptions that the alert Select statement can find before Oracle Alert switches from performing the detail action in the threshold group to performing the summary action. You can define a summary threshold for each threshold group that resides in an action set.

   For example, suppose you define an action set with a summary threshold of five. If, during an alert check, the alert Select statement locates between one and five exceptions, Oracle Alert performs the threshold group's detail action once for each exception found. If the Select statement locates six or more exceptions, however, Oracle Alert performs the summary action once for all the exceptions found.

   ☞ **Attention:** You can only define a summary threshold for an action set member that is a threshold group. You cannot define a summary threshold for an action or an escalation group.

13. Save your work. Your alert is now ready to use summary threshold.

▶ **To define a summary threshold with dynamic distribution:**

■ If you are combining dynamic distribution with summary threshold, keep in mind that Oracle Alert groups exceptions before it determines whether to perform a detail or a summary action. That is, if you use an output to represent your message action's distribution, Oracle Alert first substitutes the appropriate exception value for the output, then based on the number of exceptions substituted, performs either the detail or summary action in the threshold group.

For example, suppose you define a message action that uses an output to dynamically distribute information to a group of managers, and you define a summary threshold of 3 for your action set. Suppose that during an alert check, the alert Select statement returns 7 exceptions, 3 of which must go to one manager, 4 of which must go to another manager. Using the defined summary threshold of 3 exceptions, Oracle Alert sends three detail messages to the first manager, and one summary message reporting 4 exceptions to the second manager.

▶ **To delete a threshold group:**

1. Display the existing threshold group in the Threshold Groups window of the Alerts form.

2. Choose Delete Record from the Edit menu.

3. If the threshold group is an active member of an action set, an error message followed by a References window appears.
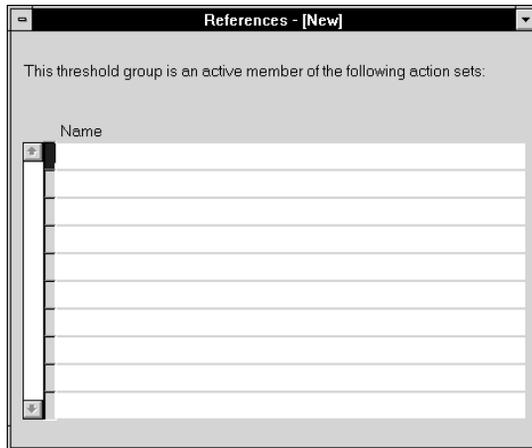
You can also choose References from the Special menu in the Threshold Groups window at any time to display the References window.

4. The References window shows you which action sets reference the current threshold group.

   You must navigate to the Action Sets window and remove or disable the threshold group from those action sets before you can delete the threshold group.

5. Once you delete your threshold group, save your changes.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Creating Self–Referencing Alerts

You can create an alert that checks your database for only those exceptions that are new since the last time the alert was checked – without saving history. A "self–referencing" alert uses its own "date–last–checked" value as the starting point when it checks your database for new exceptions.

A self–referencing alert compares the value of a date and time column with the value of an implicit input called :DATE_LAST_CHECKED in the WHERE clause of the alert Select statement. An implicit input is an input that Oracle Alert automatically provides the value for. When checking a self–referencing alert, Oracle Alert automatically provides the date and time as the value for :DATE_LAST_CHECKED.

When Oracle Alert checks your alert, it uses the date and time information in the column you specify in your Where clause to determine whether the exceptions returned by the alert Select statement were created or updated since the date the alert was last checked.   Your self–referencing alert returns only those exceptions that occurred more recently than the date and time value of :DATE_LAST_CHECKED. Then, Oracle Alert updates the value of :DATE_LAST_CHECKED with the date and time information from the current alert check. See: How Oracle Alert Checks Alerts: page 7 – 2.

Oracle Alert never displays :DATE_LAST_CHECKED in the Inputs alternative region of either the Alert Details window or the Action Sets window in the Alerts form.

▶ **To create a self–referencing alert:**

■ You create a self–referencing alert the same way you would create any other alert, but you tailor your alert Select statement by adding to the Where clause a comparison between the values in the implicit input :DATE_LAST_CHECKED and the column that contains the date and time information that you want your alert to reference.

Example

Suppose your Select statement contains the following lines:

```
.
.
WHERE ...
AND TO_DATE(creation_date, 'DD-MON-YYYY HH24:MI:SS')
  > TO_DATE(:DATE_LAST_CHECKED, 'DD-MON-YYYY
  HH24:MI:SS')
```

In this example, Oracle Alert compares the date and time information in the column CREATION_DATE with the date and time information in :DATE_LAST_CHECKED to determine which exceptions are new.

☞ **Attention:** The format of the date and time information in the column in the Where clause *must* match the format of the date and time information in :DATE_LAST_CHECKED. The format of the date and time information stored in the DATE_LAST_CHECKED column is:

```
DD–MON–YYYY HH24:MI:SS
```

☞ **Attention:** If you choose to use the TO_DATE function to modify date and time information, you must use it to modify both the column your alert references and the DATE_LAST_CHECKED column.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Overview of Oracle Alert History: page 5 – 2

Defining a Periodic Set

You can create a set of periodic alerts that Oracle Alert checks simultaneously. Use the Request Periodic Alert Check window to check the periodic set. Note that each periodic alert you include in a periodic set continues to run according to its individually defined frequency.

Before you define a periodic set, make sure you define one or more periodic alerts. See: Creating a Periodic Alert: page 3 – 5.

▶ **To define a periodic set:**

1. Navigate to the Periodic Sets form.

2. Enter the name of the application that owns the alerts you want to define as a set.

    You can define alerts as members of a set only if they are periodic alerts owned by this application. However, the Oracle Alert Manager responsibility can define and check a periodic set that includes periodic alerts from any application, provided that Oracle Alert is the application you select in this field.

3. Enter the name of your periodic alert set. The set name cannot be the same as the name of any other periodic set or alert owned by this application. See: Oracle Alert Naming Conventions: page 8 – 19.

4. Enter a description of your periodic set (up to 240 characters).

5. Assign the periodic alerts to the set in the Periodic Set Members block. You can include as many periodic alerts in your periodic set as you want, and you can even include periodic sets in a periodic set.

6. Oracle Alert automatically enters the next available number in the Seq field. You can accept the default number or change it. A sequence number need not be unique unless you prefer a particular check order. Oracle Alert checks all of the sequence 1 members first, then all of the sequence 2 members, then all of the sequence 3 members, and so on.

7. Enter the name of the application that owns the alert in the Application field. If the application that owns the periodic set is any application other that Oracle Alert, this application is the same as the application that owns the periodic set. You can only enter another application in this field if the application that owns the periodic set is Oracle Alert.

8. Enter the name of a periodic alert or periodic set in the Name field. You can choose from the list of periodic alert and periodic sets owned by the application that owns the set. You cannot add a disabled alert to a periodic set.

   The Type field displays Alert if the member is a single alert, and Periodic Set if the member is another periodic set.

9. Check Enable if your want Oracle Alert to check this member when checking the periodic set. If you uncheck this check box, Oracle Alert ignores this member when checking this periodic set.

   Note: This Enabled check box does not affect the enabled status of the periodic alert itself as defined in the Alerts form or of the periodic set itself as defined in the Periodic Sets block.

10. Enter a current or future date in the End Date field if you want Oracle Alert to disable this periodic set member by a certain date. After this date, Oracle Alert does not check this alert or set when checking the periodic set.

11. Save your work.

See Also

Checking a Periodic Alert: page 7 – 6

5

Using Oracle Alert History: Duplicate Checking and Action Escalation

This chapter defines Oracle Alert History and discusses ways you can use alert history to enhance your exception reporting capabilities.

This section includes a summary of the major Oracle Alert History features, including a discussion of alert history concepts. This chapter also explains how you use alert history to accomplish various tasks, including:

- Using Duplicate Checking
- Using Action Escalation
- Reviewing Alert Actions
- Reviewing Alert Exceptions

Overview of Oracle Alert History

When you review Oracle Alert history, not only can you see a record of the exceptions that Oracle Alert found during an alert check, but you can also review the exact text of the action that Oracle Alert performed. Oracle Alert even preserves a record of the actions that you no longer use so that you can review the history of your exception reporting. And, you can review history for your response processing alerts, and see a complete record of responses received and response actions performed.

If you are saving history for an alert, you can define your alert to look for exceptions that existed during previous alert checks – known as duplicate exceptions. You can have Oracle Alert take certain actions based on the presence of these duplicate exceptions. And, you can create a series of escalating actions for Oracle Alert to perform when it finds the same exception over a period of time. Oracle Alert performs each action in sequence, then continues to perform the last action until the exception is removed from your database.

Basic Business Needs

In your business, you should be able to:

- Review exception reporting activity over any period of time.

- Review any exception found in your database during a particular period of time.

- Review the exact text of an alert message that Oracle Alert sent, even if you no longer use that message.

- Review any actions that Oracle Alert performed during an alert check, even if you no longer use those actions.

- Review responses received and response actions performed for your response processing alerts.

- Have Oracle Alert check for exceptions that remain in your database over a specific period of time.

- Have Oracle Alert perform certain actions if an exception exists in your database during consecutive alert checks.

- Define a series of actions, each action of an increasing severity level, and have Oracle Alert perform one action each time it encounters the same exception in your database.

Major Features

Review Complete Action History

You can see a complete reconstruction of the actions taken by your alert during an alert check. You can see all the actions your alert performed, or you can enter search criteria to narrow the range of actions you want to review. When you review each action's history, you see the action exactly as it was performed by Oracle Alert, including the complete text of any message action, and the complete script of any SQL script or operating script action.

Review Complete Exception History

You can also review the exceptions found during an alert check.   You can see all the exceptions found for your alert, or just those found for a particular action set. And, you can specify which particular exceptions you want to review. You do not have to sort through all of the exception information for a particular alert – you can simply choose to omit certain outputs, or restrict the length of other outputs, to make your history review more efficient.

Review Complete Response History

Reconstruct complete history of the responses received and response actions performed for your response processing alerts.

Review by Range of Dates

Depending upon how much alert history you choose to preserve, you can review all exceptions found in your database over the period of time you specify. Or, you can look through all the history you are preserving and review all exceptions or actions for a particular alert.

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 of your alert outputs.

Escalating Actions

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 a duplicate exception, it performs the next level detail action. For every action level, you can define a different action. For each alert, you can define an unlimited number of escalation levels.

Duplicate Suppression

Once it has performed the highest level action, Oracle Alert can continue performing that action during each subsequent alert check; or with duplicate suppression, can cease performing actions altogether. Duplicate suppression lets you define Oracle Alert to perform each level of action in a group of escalating actions once only.

See Also

Creating Self–Referencing Alerts: page 4 – 17

Duplicate Checking

By saving history for your alerts, you can define Oracle Alert to check for duplicates – that is, 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. With duplicate checking, you can monitor outstanding items in your database until they are resolved. You can spend your time handling important business events, instead of following up on outstanding issues.

Checking for Duplicates

▶ **To check for duplicates:**

1. Save history for your alert by entering a number in the Keep _ Days field of the Alerts form. Oracle Alert saves history for your alert for the number of days you specify.

> **Note:** You need to save 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 history for the previous alert check is not retained. If you are defining a response set for your alert, save history for the

number of response days in addition to the number of days
you want to retain history.

2. Use the Outputs alternative region in the Alert Details window of
   the Alerts form to define which outputs Oracle Alert should use
   when it checks for duplicates.

3. Check the Check For Duplicates check box if you want Oracle Alert
   to consider an output's exception value when it checks for
   duplicates.

▶ **To check for duplicates at the action set level:**

1. You can also choose to have Oracle Alert check for duplicates at the
   action set level. When you define an action set, Oracle Alert
   automatically creates the duplicate configuration from the default
   duplicate configuration of the alert that you defined in the previous
   task.

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ =                         Action Sets - [New]                              ▼   │
├──────────────────────────────────────────────────────────────────────────────┤
│                                                              Suppress       ⊠  │
│     Seq   Action Set Name      Description              Duplicates  Enabled  End Date │
└──────────────────────────────────────────────────────────────────────────────┘
```

2. You can customize the duplicate configuration of each action set by checking the Check for Duplicates field for each output in the Outputs alternative region of the Action Sets window.

   Note: Each action set duplicate configuration takes precedence over the alert duplicate configuration during an alert check.

▶ **To flag duplicates in detail messages:**

1. To flag for duplicates and have Oracle Alert perform a different detail action each time it finds the same exception during an alert check, you need to customize your alert to use a feature called action escalation. See: Action Escalation: page 5 – 9.

2. If you want Oracle Alert to perform the same detail action each time it finds the same exception during an alert check, you need to turn off duplicate checking by unchecking the Check for Duplicates check box in the Alert Details window and in the Action Sets window of the Alerts form.

▶ **To flag duplicates in summary messages:**

1. You can flag duplicate exceptions within summary messages by specifying that you want to suppress duplicates at the action set level. When it sends the summary message, Oracle Alert includes the duplicate exception in the summary message, but flags each duplicate exception found with an asterisk (*).

2. To suppress duplicates at the action set level, check the Suppress
    Duplicates check box in the Action Sets window of the Alerts form.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Action Escalation: page 5 – 9

Action Escalation

You can define a set of escalating detail actions, called an escalation group, for Oracle Alert to perform when it finds the same exceptions during consecutive alert checks. Oracle Alert performs a different detail action each time it encounters the same exception, so you can define actions that correspond to increasing severity levels.

For example, you can define a series of detail message actions for Oracle Alert to distribute if it encounters the same exception during sequential alert checks. You can define each message to go to a different, and higher level manager. Oracle Alert sends one message each time it finds the same exception during an alert check, notifying the next level of management, until it sends the highest level message to the senior manager. Oracle Alert continues to send the highest level message to the senior manager until the exception is removed from your database.

Defining Escalating Actions

Before you define an escalating action, make sure you first define an alert, and save history for at least one day longer than the number of days between alert checks.

You also want to define at least two detail alert actions, one that you want Oracle Alert to perform during a first alert check, and a second that you want Oracle Alert to perform during a subsequent alert check if it finds the same exceptions.

▶ **To define escalating actions:**

1. Navigate to the Alerts form and create a new alert or display an existing alert that you want to define an escalating action for.

2. Save history for your alert by entering a number in the Keep _ Days field of the Alerts form. Oracle Alert saves history for your alert for the number of days you specify.

   Note: You need to save history for your alert for at least one day longer than the frequency of the alert if you want to use action escalation. Oracle Alert cannot check for duplicates if history for the previous alert check is not retained.

3. Choose Alert Details.

4. Display the Outputs alternative region.

5. Check the Check for Duplicates check box for an output if you want Oracle Alert to consider that output's exception value when it checks for duplicates. The default value is checked.

6. In the Alerts form, choose Actions.

7. In the Actions window, create the actions you want Oracle Alert to perform if it finds duplicate exceptions across alert checks. You can define message, SQL statement script, operating system script, or concurrent program request actions.

   Note: You can use only detail level actions for action escalation, but you can define as many detail actions as you like.

8. While displaying the Alerts form, choose Escalation Groups from the Special menu to display the Escalation Groups window.

9. Define an escalation group by giving it a name (up to 80 characters) and description (up to 240 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

10. Specify the escalation group type as either Message, SQL Statement Script, Operating System Script, or Concurrent Program, in the Type field.

   You can define as many escalation groups as you like, but an escalation group can contain only detail level actions. And, actions within an escalation group must be of the same action type; for example, one escalation group can contain message actions only.

11. In the Escalation Group Members block, display the list of predefined actions that you can add to the escalation group in the Action Name field. The list contains only actions created for this alert that are of the same type as the escalation group type.

12. Oracle Alert assigns an escalation level to each action in the escalation group and automatically defaults to the next available number for each successive action. You can change the escalation level of an action.

   Each time Oracle Alert checks the alert and finds the same exception, it performs the next higher level of action in the escalation group. Once it reaches the highest enabled action level Oracle Alert continues to perform that action until the exception is removed from your database.

13. Check Enabled to enable the escalation group member.

14. Enter a current or future date in the End Date field if you want an escalation group member to be disabled by a certain date.

15. Save your work.

▶ **To perform escalated actions:**

1. Include your escalation group in an action set for Oracle Alert to perform the escalated actions.

2. Save your work. Your alert is now ready to perform escalated actions.

▶ **To invoke duplicate suppression:**

1. For each action set you define, you can choose to have Oracle Alert "suppress" duplicates. That is, you can have Oracle Alert stop performing the actions in an escalation group once it has performed the highest level action.

2. If you check the Suppress Duplicates check box in the Actions Sets window of the Alerts form, Oracle Alert performs each action in the escalation group. Once it has performed the highest level action, Oracle Alert no longer performs any action within that escalation group, even if the exception remains in your database.

   For example, suppose you define actions of levels 1 and 2. When Oracle Alert first detects an exception, it performs the level 1 action. If, during the next alert check, Oracle Alert finds that same exception, Oracle Alert performs the level 2 action. If, during the

third alert check, the duplicate exception still exists, Oracle Alert does not perform the level 2 action again.

3. If you uncheck 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.

4. If your action set includes a summary message action and you check the Suppress Duplicates check box, Oracle Alert includes any duplicate exception in the summary message, but flags each duplicate found with an asterisk (*).

▶ **To customize your action set's duplicate configuration:**

■ You can tailor your alert's duplicate configuration at the action set level. When you define an action set, Oracle Alert automatically creates the duplicate configuration from the default duplicate configuration of the alert. You can customize the duplicate configuration of each action set by altering the Check For Duplicates check box in the Action Sets window. Oracle Alert uses this duplicate configuration when checking for duplicate exceptions.

> **Note:** Each action set duplicate configuration takes precedence over the alert duplicate configuration during an alert check.

▶ **To delete an escalation group:**

1. Display the existing escalation group in the Escalation Groups window of the Alerts form.

2. Choose Delete Record from the Edit menu.

3. If the escalation group is an active member of an action set, an error message followed by a References window appears.
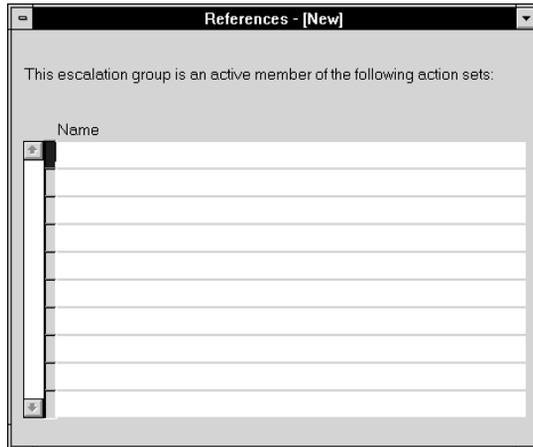
You can also choose References from the Special menu in the Escalation Groups window at any time to display the References window.

4. The References window shows you which action sets reference the current escalation group.

    You must navigate to the Action Sets window and remove or disable the escalation group from those action sets before you can delete the escalation group.

5. Once you delete your escalation group, save your changes.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Creating Alert Actions: page 3 – 19

Creating an Action Set for an Alert: page 3 – 33

Checking for Duplicates: page 5 – 5

Reviewing Alert Actions

You can review the history of any or all actions performed by Oracle Alert by using the Review Alert Actions window in the Review Alert History form. You can also use this window to review the actions performed for your response processing alerts.

Reviewing Action History

Before you can review any action or response action history for an alert, you must first save history for the alert for at least one day and have Oracle Alert check that alert.

▶ **To review action history:**

1. Navigate to the Review Alert History form.

2. Enter the name of the application that owns the alert and the name of the alert. Oracle Alert automatically displays the alert type, the number of days of history kept, and when the alert was last checked.

3. To narrow down the amount of history shown for your alert, you can specify a range of dates. If you do not specify a range of dates, Oracle Alert displays all actions performed for this alert for the entire period of saved history.
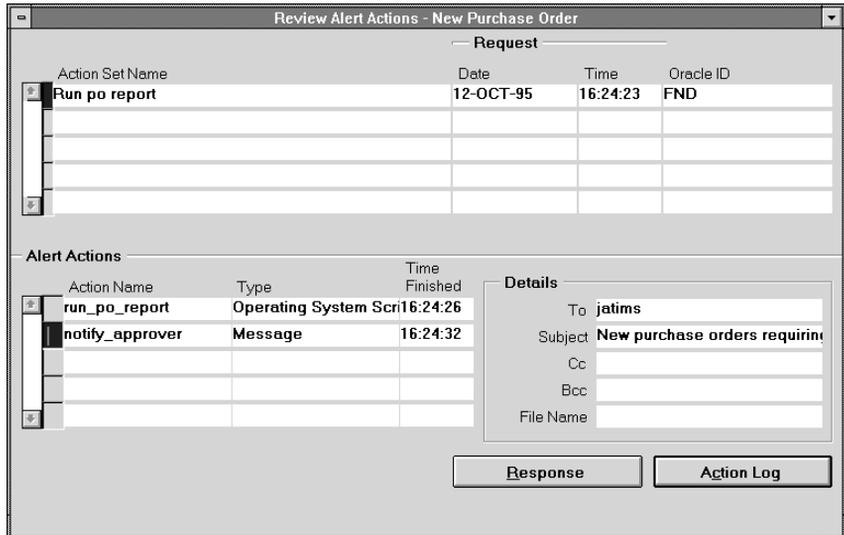
4. Choose Find Actions to display the Review Alert Actions window. The Review Alert Actions window displays the history of action sets that were run: the date and time when the action set completed and the Oracle ID that the action set ran against.

The Alert Actions block also displays each action member in the action set and provides details of that action.

5. If the alert action member is a message action and the message was defined in Oracle Alert, you can see a complete replicate of the message, exactly as it appeared to the recipient(s) by choosing Action Log. Oracle Alert opens a window that displays the entire message.

If the message text originated from an external file, Oracle Alert does not reconstruct the message, instead Oracle Alert displays the contents of the file.

6. If the alert action member is a concurrent program, Oracle Alert displays the date the concurrent request was started, the name of the application that owns the concurrent program, the name of the concurrent program, and any arguments that were passed to the concurrent program.

7. If the alert action member is an operating system script, the alert action Details displays any arguments passed to the script and if the script originated from an external file, the name of the file.

If the script was defined in Oracle Alert, you can see a complete replicate of the script, exactly as it was performed. Choose Action

Log to display the entire operating system script in a separate window.

8. If the alert action member is a SQL statement script and the SQL statement script was defined in Oracle Alert, you can see a complete replicate of the script, exactly as it was performed. Choose Action Log to display the entire SQL statement script in a separate window.

▶ **To review response history:**

1. Choose Response in the Review Alert Actions window to see the responses received and response actions performed by Oracle Alert when reviewing history for your response processing alerts.
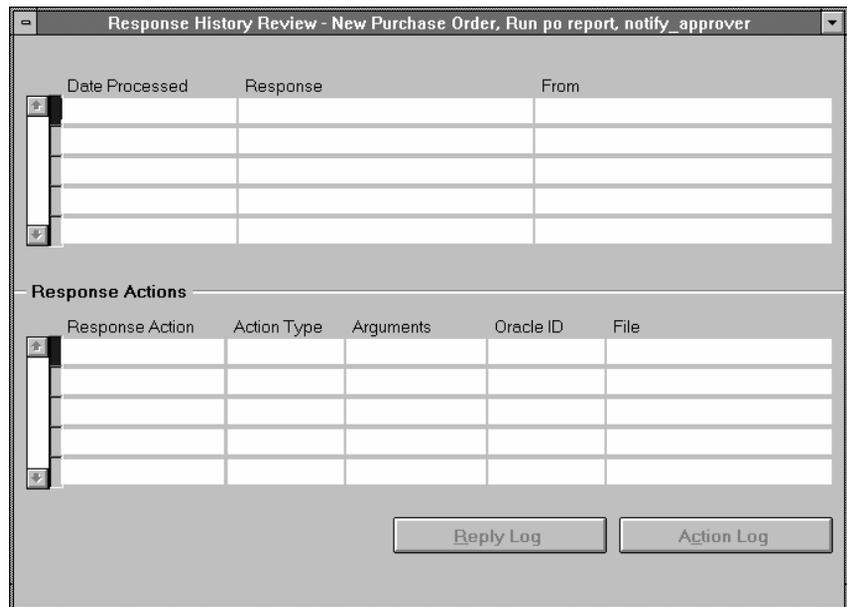
2. The Response History Review window shows the complete text of responses received in reply to an alert message action. Oracle Alert displays the date and time it received each response, the type of response it received (valid, invalid, or no response), and the name of the respondent. When you choose Reply Log, a window displays the entire response.

3. When you choose Action Log, a window displays the attached message action that originally solicited the response.

The message appears exactly as it was sent but includes the current version of the response text as it is defined in the Oracle Alert Options form. If you edit your response text between the time Oracle Alert sent the original message and the time you reconstruct the message, the reconstructed message will include the current version of the response text. Similarly, Oracle Alert displays output values within the currently defined Max Length settings.

4. You can review a complete record of the response actions performed by Oracle Alert when it received a particular response, if those responses are defined in Oracle Alert.

   The Response Actions block displays all the actions performed for the current response received. Oracle Alert displays the name of the action, the action type, any arguments used by the action; the Oracle ID where the SQL statement was executed, if your action was a SQL script action; and the name of the file if your action originated in an external file.

   Choose Action Log to open a window and display the text of the action, if the action is a message action, operating system script, or SQL statement script action that is defined in Oracle Alert. For concurrent request response actions, Oracle Alert displays the name of the concurrent program that was submitted to the concurrent manager.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Checking a Periodic Alert: page 7 – 6

Defining a Response Processing Alert: page 6 – 4

Reviewing Alert Exceptions

You can review any or all exceptions found by Oracle Alert during an alert check by using the Review Alert Exceptions window in the Review Alert History form. You can review all exceptions found for an alert, or you can review just those for a particular action set.

Reviewing Exception History

Before you can review any exception history for an alert, you must first save history for the alert for at least one day and have Oracle Alert check that alert.

▶ **To review exception history:**

1. Navigate to the Review Alert History form.

2. Enter the name of the application that owns the alert and the name of the alert. Oracle Alert automatically displays the alert type, the number of days of history kept, and when the alert was last checked.

3. You can enter a range of dates if you want to narrow your review to exceptions found during a particular period of time. If you do not specify a range of dates, Oracle Alert displays all exceptions found for this alert for the entire period of saved history.

4. Choose Find Exceptions.

5. In the Display Output Variables window, check the output variable values you want to review for each exception. Choose OK to display the Review Alert Exceptions window.

6. The Review Alert Exceptions window displays the name of the action set run for each exception, when the action set ran and completed and the Oracle ID that the action set ran against. The Alert Exceptions block displays the output values for each exception.

See Also

Creating a Periodic Alert: page 3 – 5

Creating an Event Alert: page 3 – 11

Checking a Periodic Alert: page 7 – 6

Reviewing Alert Checks

You can review alert check information by using the Review Alert Checks window in the Review Alert History form. Oracle Alert saves alert check information regardless of whether you save history for an alert.

Reviewing Alert Check Information

Before you can review any alert check information for an alert, you must first save history for the alert for at least one day and have Oracle Alert check that alert.

▶ **To review alert check information:**

1. Navigate to the Review Alert History form.

2. Enter the name of the application that owns the alert and the name of the alert. Oracle Alert automatically displays the alert type, the number of days of history kept, and when the alert was last checked.

3. You can enter a range of dates if you want to narrow your review of alert checks to a particular period of time. If you do not specify a range of dates, Oracle Alert displays the information for all the alert checks made for this alert for the entire period of saved history.

4. Choose Find Checks to display the Review Alert Checks window.

5. If the alert is a periodic alert that was checked on demand, the Requestor field displays the username of the application user who requested the alert check. If the alert is a periodic alert that was submitted by the Periodic Alert Checker, the Requestor field displays the username of the application user who started the concurrent manager. If the alert is an event alert, the Requestor field displays the username of the application user who caused the event.

6. The Review Alert Checks window also displays the date and time when the alert check completed, the concurrent request ID for the alert check, and the status of the alert check. The possible statuses are:

   - **Completed**—The alert check is complete
   - **Running**—Oracle Alert is currently checking the alert
   - **Pending**—The alert is in the concurrent queue waiting to run
   - **Error**—The alert check did not complete successfully

7. The History Purge Date field displays the date on which the Periodic Alert Checker purged the history for this alert. If no date appears, then the history was not purged for this alert.

8. Choose Request Log from the Special menu of the Review Alert Checks window to view the complete log file of the check request.

9. The Action Set Checks block displays the history of the action set checks performed for each alert check. The history includes the name of the action set that was checked, the Oracle ID that the

action set was checked against, and the status of the action set check. In addition, this block also shows the number of exceptions found and the number of actions performed during the action set check.

See Also

Using Response Processing

This chapter tells you everything you need to know about response processing in Oracle Alert.

This section includes a summary of the major response processing features and a discussion of response processing concepts. This chapter also contains task essays that explain how you perform typical response–processing tasks, including:

- Defining a response processing alert

- Implementing multiple Oracle Alert installations with response processing

Overview of Response Processing

Oracle Alert can process responses to your alert messages. When Oracle Alert receives a response to a specific alert message, it automatically performs the actions you define. Optionally, respondents can supply values that Oracle Alert uses to perform these actions. Response processing lets you automate routine user–entry transactions, streamlining your organization's operations.

> **Note:** You must have an installation of Oracle Office and define at least one mailbox in Oracle Office if you want to use response processing. Within that response processing mailbox, you must also create two folders called Reviewed and Reviewed_OK. See: Specifying Oracle Alert Options: page 8 – 3.

Basic Business Needs

While using Oracle Alert, you should be able to:

- Define a set of possible responses to a particular alert message

- Define one or more actions for Oracle Alert to take when it receives a particular response

- Let the responder to an alert message provide values that Oracle Alert uses when performing response actions

- Define one or more actions for Oracle Alert to take if Oracle Alert receives no response by a specific date

- Define one or more actions for Oracle Alert to take if Oracle Alert receives a response it cannot understand

- Save a complete record of response actions taken, for as many days as you specify

- Use response processing with multiple Oracle Alert installations

Major Features

User–Definable Responses

Oracle Alert lets you define the valid responses that a recipient of an alert message can send back to Oracle Alert. If you define responses

for an alert message, Oracle Alert includes the valid responses in the message, so that one of them can serve as a template for the recipient's actual response.

Response Actions

Oracle Alert lets you define actions that it performs automatically when it receives a particular response to an alert message. You can define actions that execute a SQL statement or operating system script, submit a concurrent program request, or send an electronic mail message.

Response Variables and Response Variable Values

Oracle Alert lets you define responses that may include response variables whose values are provided by the responder. Oracle Alert uses these response variable values when performing the actions associated with the response.

No Response Follow–Up Actions

Oracle Alert lets you define follow–up actions that Oracle Alert takes if the message recipient does not respond. Oracle Alert waits for a response to an alert message for the number of days you specify. If Oracle Alert receives no response within that time period, then Oracle Alert automatically initiates the follow–up actions.

Defining a Response Processing Alert

You must complete the following tasks in the order listed to define a response processing alert:

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

► **To define a response processing alert:**

1. Navigate to the Alerts window.

2. Create either an event or a periodic alert. See: Overview of Alerts: page 3 – 2.

   Suggestion: Include in your alert Select statement all the outputs you need to create the alert message action that solicits a response. Also make sure you include all the outputs you need to create the actions to the valid and invalid responses you plan to define.

   Note: You do not have to save history to use response processing, but you can save history for your response processing alerts. See: Response Processing History: page 6 – 17.

3. Save your work.

► **To create a message action that solicits a response:**

1. Choose Actions in the Alerts window.

2. Define a message action in the Actions and Action Details windows that solicits a response from the recipient. Your message can be either a summary or detail level action. See: Creating Alert Actions: page 3 – 19.

3. Format the message however you like, and include the outputs that represent the exceptions you want. See: Formatting Summary Message Actions: page 4 – 6.

When Oracle Alert sends out your message, it automatically appends two pieces of text:

- **Response Text**—tells the recipient exactly how to reply to the alert message. This text originates from the Message Elements alternative region of the Oracle Alert Options form. Your Oracle Alert installation is supplied with the following default text, which you can customize to your needs:

```
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:
```

- **Valid Response Text**—lists the possible responses that the recipient must reply with. You define these possible responses in the Valid Responses block of the Response Sets window, which we describe later in this essay. Oracle Alert appends this Valid Response Text to your message, so the recipient knows the exact text he or she must respond with.

To summarize, the alert message that solicits a response contains three sections of text. In order of appearance in the message, they are:

- Text you create in the Action Details block located in the Actions window of the Alerts form that reports the exceptions found, and asks for a response

- Response text that originates from the Oracle Alert Options form that instructs the recipient how to reply to the message

- Valid response text of possible valid responses, one of which the recipient must include in his or her response

4. Save your work.

▶ **To create the response actions:**

1. Display the Actions window of the Alerts form.

2. Define the response actions in this window. Response actions are the actions you want Oracle Alert to take when it receives a reply to the alert message action you created in the previous task. You define your response actions before defining your responses.

Create as many response actions as you need. See: Creating Alert Actions: page 3 – 19.

> **Note:** You can define response actions only as detail level actions.

3. In any of the response actions you define, you can use both outputs and response variables in the action details. Response variables are variables whose values are assigned by the respondent.

   Identify your response variables with an ampersand before the name, for example, &VARIABLE_NAME. Oracle Alert reads the value for a response variable from the message response and reads the value of an alert output from the original alert message, then substitutes the values when executing the action. If the response message does not include the response variable, Oracle Alert uses the default value from your response variable definition.

   Suggestion: You can create actions for Oracle Alert to take when it receives a response it does not recognize, or when it does not receive a response at all. For example, define a message action that tells the recipient that Oracle Alert did not understand their response.

4. Save your work.

▶ **To define an action set for a response processing alert:**

1. Choose Action Sets in the Alerts window.

2. Define an enabled action set that contains the alert message action that you want to use to solicit a response. See: Creating an Action Set for an Alert: page 3 – 33.

3. Save your work.

▶ **To create a response set for a response processing alert:**

1. Choose Response Sets in the Alerts window.

Use the Response Sets window to define the set of valid, invalid, and no response responses your alert message might receive, and tell Oracle Alert which response action it should perform when it receives one of those responses.

2. Enter a response set name (up to 30 characters) and description (up to 240 characters). See: Oracle Alert Naming Conventions: page 8 – 19.

3. Check Enabled to enable your response set. If you uncheck Enabled, you cannot attach this response set to a message action to solicit responses. If the response set is already attached to a message action, you can uncheck Enabled to prevent Oracle Alert from processing any new responses for that alert.

4. Enter a current or future date in the End Date field to disable your response set by a certain date.

5. Save your work.

6. Choose Response Variables in the Response Sets window to define the response variables that you used in the response actions that you created earlier.

Variable Name	Description	Type	Max Length	Default
		Character		

7. Enter a name for the response variable (up to 30 characters), omitting the ampersand. Enter a description for your response variable (up to 240 characters).

☞ **Attention:** Response variables and outputs cannot have the same names. When you define a response variable, Oracle Alert verifies that the name you assign does not conflict with any defined output names. Oracle Alert displays an error message if your response variable name conflicts with an output name.

8. Specify the data type of the response variable: Character, Number, or Date, in the Type field. Oracle Alert uses the data type to validate default values you enter in the Default field and responses from message recipients.

9. Enter the maximum number of characters you want to display for this response variable in the Max Length field. If the response variable value contains more characters than the maximum length you specify, Oracle Alert:

- Truncates character and date data

- Displays a row of pound signs (######) for numeric data

10. Enter a default value to a response variable in case a respondent does not specify a value for the response variable.

11. Save your work.

▶ **To define the possible responses to a message action:**

1. Choose Valid in the Response Sets window to define your valid responses. A valid response is a response that you expect from an alert message recipient, and one that initiates the alert actions you specify. You can define any number of valid responses as you may have several possible courses of action you allow the user to take.

2. Enter a name for your valid response (up to 240 characters).

3. In the Text field, enter the actual text that you want the respondent to reply with. This becomes the Valid Response Text that Oracle Alert appends to the original alert message.

   If you are defining several valid responses for the same alert, make the first word of each valid response unique. When interpreting an actual response, Oracle Alert uses the first word to identify which valid response it is. In the original alert message, Oracle Alert displays the text of every valid response you define.

   ☞ **Attention:** Your response must not include a blank line, so limit your response to one paragraph. Oracle Alert interprets a blank line as the end of the response.

4. If you want the respondent to specify a particular value for a response variable in her/his response, enter the response variable in the Text field, followed by an equal sign and a question mark between a set of quotation marks (="?").

   Note: Including a response variable in your response text is optional.

For example, suppose you have an alert that sends a message soliciting a reorder quantity from its recipient. You first specify a numeric response variable called &QUANTITY in the alert SQL Select statement and define the variable in the Response Variable window. Then you specify the text of your 'Reorder' valid response as:

```
Reorder QUANTITY="?"
```

Oracle Alert appends this text to the bottom of your alert message, and expects a response beginning with the word 'Reorder', and including the characters QUANTITY="n", where *n* is the amount the recipient wants to order. When Oracle Alert performs any actions that include the response variable &QUANTITY, Oracle Alert substitutes the value of *n* for the response variable.

Your valid response can include as many response variables as you need.

5. In the Actions block of the Valid Responses window, you assign one or more of the response actions you defined earlier for each of the valid responses that you define in the Responses block.

6. Enter a number in the Seq field representing the sequence in which you want this action to perform relative to the other actions identified for this response.

7. Choose a response action from the list of predefined actions in the Action field. If you select an action that uses alert outputs or response variables that are not defined for this alert or response set, a warning message appears.

8. Check Enabled to enable the action.

9. Enter a current or future date in the End Date field to disable this action by a certain date.

10. Save your work.

11. You can optionally add invalid response actions to your response set by choosing Invalid from the Response Sets window. An invalid response action can be a message action that tells the respondent their reply was not understood.

Seq	Action	Description	Type	Enabled	End Date
				☒	
				☐	
				☐	
				☐	
				☐	
				☐	
				☐	
				☐	
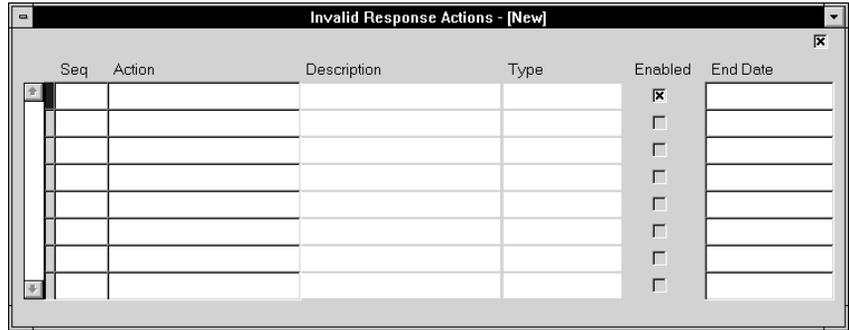
12. In the Invalid Response Action window, enter a number in the Seq field representing the sequence in which you want this action to perform relative to the other actions for the invalid response.

13. Choose a response action from the list of predefined actions in the Action field. If you select an action that uses alert outputs or response variables that are not defined for this alert or response set, a warning message appears.

> **Note:** You can also choose the original alert message action as an Invalid Response Action to resend the original message that asks for a response.

14. Check Enabled to enable the action.

15. Enter a current or future date in the End Date field to disable this action by a certain date.

16. Save your work.

17. You can also optionally choose None from the Response Sets window to add No Response follow–up actions for Oracle Alert to perform if the recipient does not respond at all.

No Response Actions - [New]

Seq	Action	Description	Type	Enabled	End Date
				☒	
				☐	
				☐	
				☐	
				☐	
				☐	
				☐	
				☐	
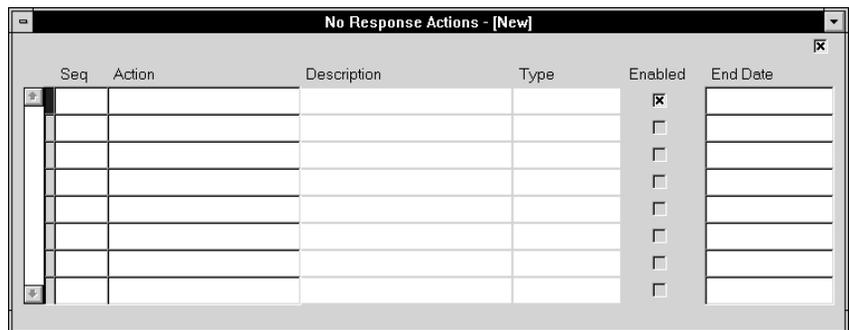
18. The No Response Actions window is similar to the Invalid Response Actions window. Follow steps 12 – 16 to define a no response follow–up action.

▶ **To attach the response set to the alert message action:**

1. For Oracle Alert to perform response processing on an alert, you must attach your response set to a message action. Go to the Actions window in the Alerts form.

2. Display the Action Details window for the message action that you want to use to solicit a response.

3. In the Response Set field, enter the name of your response set. Oracle Alert uses this response set to compose the response template that it appends to this message and to process the responses it receives for this message.

   You cannot remove a response set from a message action if there is an outstanding response.

4. In the Response Days field, specify the number of days that you want Oracle Alert to wait before it performs the no response follow–up actions defined in your response set for each outstanding exception.

   For example, suppose you enter 1 in this field. Oracle Alert waits one full day after sending the message for a response from the

original message recipient. If no response is received, it performs the no response action the first time it reads its mail on the following day. The default value for this field is 7 days. If you leave this field blank, Oracle Alert never performs the no response actions on the outstanding responses.

5. Save your work. Your response processing alert is ready for use.

▶ **To delete a response set:**

1. Display the existing response set in the Response Sets window of the Alerts form.

2. Choose Delete Record from the Edit menu.

3. If the response set is attached to a message action, an error message followed by a References window appears.
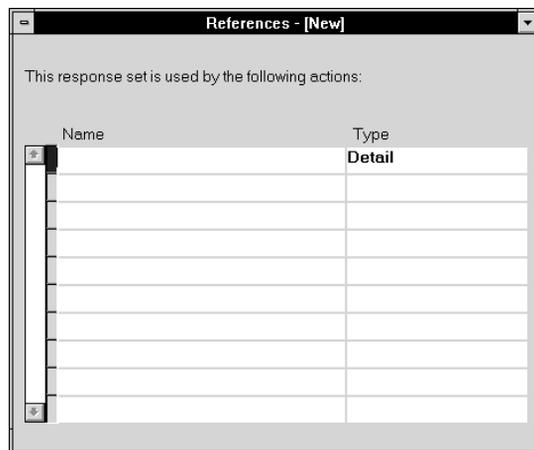
You can also choose References from the Special menu in the Response Sets window at any time to display the References window.

4. The References window shows you which actions reference the current response set.

You must navigate to the Action Details window and remove the response set from the message action that references the response set before you can delete the response set.

5. Once you delete your response set, save your changes.

> **Note:** You also cannot delete a response set if it has outstanding responses.

See Also

Responding to an Alert Message

When Oracle Alert sends the original alert message action that solicits a response, it automatically appends a unique identifier – a message handle – to the end of the message. Oracle Alert uses this message handle to identify responses to the message. If a response does not include the original message (with its message handle), Oracle Alert cannot perform any of the response actions for that alert. See: Message Handles: page 6 – 19.

▶ **To respond to an alert message:**

1. Use your mail program to reply to the alert message action.

2. Your response must match one of the possible valid responses listed in the original message action for Oracle Alert to recognize it as a valid response.

3. Oracle Alert considers any message that contains the appropriate message handle and that does not match one of the possible valid responses, an invalid response.

4. If the original message action includes a response variable as part of the valid response text, Oracle Alert expects you to supply a response variable value in your reply. For example, if the valid response text is:

```
Print at PRINTER="?"
```

You need to reply with the exact same text, but replace the question mark with the value you want Oracle Alert to use. For example:

```
Print at PRINTER="UNX138"
```

5. If you do not supply a response variable value, Oracle Alert uses the default value assigned to the response variable. If a default value was not assigned to the response variable, Oracle Alert performs the response action, but without using a response variable value.

☞ **Attention:** Oracle Alert identifies response variables in the text of a response by the word that immediately precedes the equal sign (=) in the text. Oracle Alert cannot identify a response variable if no equal sign follows it. If the respondent supplies a response variable value, but does not include the response variable itself, Oracle Alert cannot process the response.

The Response Processor

The Response Processor is the Oracle Alert concurrent program that processes responses to alert messages. The Response Processor runs according to the schedule you define in the Response Processing alternative region of the Oracle Alert Options form. You define the Response Processor's start time, end time, and check interval. If you do not define a schedule, the Response Processor runs every day at midnight. See: Defining Response Processing Options: page 8 – 9.

You activate the Response Processor from the Schedule Alert Programs form. You can either start the Response Processor immediately, or schedule it to start later. When you start the Response Processor, Oracle Alert submits it as a concurrent request to the concurrent manager to run at the next scheduled interval. If the Response Processor gets deleted from the concurrent queue, simply restart it from the Schedule Alert Programs form, and this will resubmit the Response Processor to the Concurrent Manager. See: To activate or deactivate the Response Processor: page 7 – 10.

How Oracle Alert Processes Responses

To use response processing, each Oracle Alert installation must have at least one defined Oracle Office account. That Oracle Office account is also called a *response mail account*. You define this account in Oracle Alert using the Oracle Office Accounts alternative region in the Oracle Alert Options form. See: Defining Oracle Office Accounts: page 8 – 5.

Each response account must have three folders defined in Oracle Office: the Inbox folder, the Reviewed folder, and the Reviewed_OK folder. Oracle Alert uses these folders to process responses. Consult your Oracle Office documentation to learn how to create a new folder in a mail account.

When the Response Processor runs, it moves all the messages in the Inbox folder to the Reviewed folder. Oracle Alert recognizes a message in a response account as a response message, by whether the message contains a message handle. Once the messages are in the Reviewed folder, Oracle Alert looks for messages that contain message handles. If the recipient used Reply–with–Original or Forward to include the original message with her or his response, then Oracle Alert can locate the message handle and interpret the response. See: Message Handles: page 6 – 19.

Once Oracle Alert recognizes a response message, it reads the response and tries to match the first word or the first 30 continuous characters of the response to any of the defined valid responses. If Oracle Alert finds that the actual response matches a valid response, then it performs the actions defined for that valid response. Once Oracle Alert performs the actions for that valid response, it closes the response so that no other responses are processed for that message, then moves the message to the Reviewed_OK folder.

If the actual response does not match any of the valid responses in that response set, Oracle Alert performs the actions associated with the invalid response, but leaves the message in the Reviewed folder.

Oracle Alert performs the invalid response actions each time it receives an invalid response within the follow–up days specified in the original message action. After the number of follow–up days have passed, Oracle Alert performs the no response follow–up actions for each invalid response that never received a valid response.

If Oracle Alert never receives a valid response to the alert message, it performs the no response follow–up actions for each unreturned message once the number of response days has passed, and closes responses for all messages in that response set.

Each time the Response Processor completes, the Inbox of Oracle Alert's response account is empty.

> **Note:** Oracle Alert provides you with a precoded alert that you can use to periodically purge the Reviewed and Reviewed_OK mail folders. See: Oracle Alert Precoded Alerts: page 9 – 2.

Response Processing History

You can review a complete record of the exceptions found and response actions performed in Oracle Alert, if you save history for your response processing alert. When you define a response processing alert, the value you enter in the Keep _ Days field of the Alerts form determines the amount of history you save. The number you enter in this field should be in addition to the number of response days you enter in the Action Details block of the Actions window.

For example, if you want Oracle Alert to save history for 7 days after it performs the response actions, enter 7 plus the number of follow–up response days you specified for the alert message.

See Also

Multiple Oracle Alert Installations with Response Processing

Oracle Alert can process responses to alert messages generated from different Oracle Alert installations. By uniquely identifying each message that is associated with a response set, Oracle Alert can determine which response set a message is linked to, and perform the appropriate response actions.

Multiple Oracle Alert Installations

Depending upon the complexity of your system, you may choose to have multiple Oracle Alert installations. If your applications reside in different databases, you may want to install Oracle Alert in each database. You can then define periodic and event alerts to monitor application transactions in each database.

Maintaining separate Oracle Alert installations can help you enhance your system's security. If, for example, you maintain your organization's human resources applications in a separate database, you can install Oracle Alert to monitor exceptions in that database.

Message Handles

Oracle Alert automatically attaches a message handle – a unique identifier – to each message that is associated with a response set. A message handle is composed of a number that identifies the Oracle Alert installation where the message originated, and a number that identifies the message itself. The format of a message handle as it appears in an alert message is:

```
Message #:<Oracle Alert installation number.message number>
```

For example:

```
Message #: 2.347
```

The message handle is visible to the message recipient as a number in the lower left hand corner of the message.

> **Note:** You define your Oracle Alert installation number in the Response Processing alternative region of the Oracle Alert Options form. Each Oracle Alert installation must be assigned a unique number. See: Defining Response Processing Options: page 8 – 9.

How the Response Processor Uses Message Handles

Each Oracle Alert installation has a defined response mail account. This definition resides in the Oracle Office Options alternative region of the Oracle Alert Options form. See: Defining Oracle Office Accounts: page 8 – 5.

> **Note:** You can also define a single response mail account to be the response mail account for several Oracle Alert installations.

Each response account you define has three folders that it uses to process responses: the Inbox folder, the Reviewed folder, and the Reviewed_OK folder. When the Response Processor runs, it looks in the Inbox folder and moves all the messages it finds to the Reviewed folder, one at a time. Once the messages are in the Reviewed folder, Oracle Alert looks for messages that contain message handles.

> ☞ **Attention:** A respondent must use Reply–with–Original or Forward to include the original message with her or his response, for Oracle Alert to locate the message handle and interpret the response.

If a response account receives a response generated by a different Oracle Alert installation, that Response Processor will ignore the response because the message handle references a different Oracle Alert installation. The Response Processor moves the response back to the Inbox until the correct Response Processor recognizes the message handle and processes the appropriate response actions.

See Also

How Oracle Alert Checks Alerts: page 7 – 2

How Oracle Alert Checks Alerts

This chapter discusses how Oracle Alert checks for alert exceptions.

How Oracle Alert Checks Alerts

Oracle Alert checks your database for the exceptions you want to know about using several concurrent programs:

- Periodic Alert Scheduler
- Check Periodic Alert
- Check Event Alert
- Response Processor

Oracle Alert also maintains a history of all alert checks done by these concurrent programs so that you can see a complete record of alert checks performed for your alert.

See Also

Overview of Concurrent Processing
(*Oracle Applications User's Guide*)

Overview of Oracle Alert Concurrent Programs

Concurrent managers are components of the Oracle Alert concurrent processing facility that monitor and run time–consuming, non–interactive tasks without tying up your terminal. Whenever you submit a concurrent request in Oracle Alert, such as checking an on–demand alert, a concurrent manager processes that request in the background, letting you perform an unlimited number of tasks simultaneously.

The concurrent manager processes the following concurrent programs for Oracle Alert:

Periodic Alert Scheduler	Resides in the concurrent queue and runs every 24 hours at 12 AM, when it submits requests for all periodic alerts that are scheduled to run during the next 24 hours
Check Periodic Alert	Submits a request to the concurrent manager when you check an on demand alert
Check Event Alert	Submits a request to the concurrent manager when an insert or an update to an event table occurs

Response Processor	When active, the Response Processor resides in the concurrent queue and runs according to the interval you define. It checks the response mail account for responses to alert message actions, and performs the appropriate response actions.

If for some reason you need to inactivate or activate the concurrent manager, you can use the Schedule Alert Programs form.

Before you use the Schedule Alert Programs form for the first time, make sure you specify the following in the More Options alternative region of the Oracle Alert Options form:

- Concurrent Manager Name (default is STD)

- Concurrent Manager Startup Command (default is STARTMGR)

▶ **To activate or deactivate the concurrent manager:**

Schedule Alert Programs		
Program	**Status**	**Request ID**
Internal Manager	Active	
Periodic Alert Scheduler	Active	221994
Response Processor	Active	221992

Active Date [   ]

[**Deactivate**]   [Activate]

1. Navigate to the Schedule Alert Programs form. The Status field shows whether the concurrent manager is active or not.

2. Select Internal Manager.

3. If the status is Deactivated, choose Activate to start the concurrent manager immediately.

4. If the status is Active, choose Deactivate to terminate any running requests and deactivate the concurrent manager immediately.

   Note: You can activate or deactivate the concurrent manager on the current date only.

The Periodic Alert Scheduler

The Periodic Alert Scheduler (ALEPPE) is a concurrent program that automatically checks your scheduled periodic alerts.

How the Periodic Alert Scheduler Works

The Periodic Alert Scheduler runs at each day at 12 AM. When it runs, the Periodic Alert Scheduler looks for all periodic alerts scheduled to be checked that day, and then submits one concurrent request for each scheduled check.

For example, suppose you have two periodic alerts scheduled to be checked today. You scheduled the first alert for 6 AM, and the second for every two hours between 10 AM and 6 PM. The Periodic Alert Scheduler submits three requests. The first request is for the first alert at 6 AM. The second request is for the second alert at 10 AM, with a resubmission interval of two hours thereafter, and an end time of 6 PM. The last request is for the Periodic Alert Scheduler to run again at 12 AM. Every time Oracle Alert checks an alert successfully, it inserts that day's date in the Last Checked field of the Alerts form.

The Periodic Alert Scheduler also checks for any alerts that were scheduled to run the previous day but failed. It does this by looking for alerts that do not have the date and time of the latest scheduled check in the Last Checked field. The Periodic Scheduler resubmits each failed alert to the concurrent manager every day until the alert runs successfully (that is, the error causing it to fail is corrected) or is disabled.

How Oracle Alert Checks an Alert

Oracle Alert performs each action in an action set in sequence. During an alert check, Oracle Alert checks each action set for any defined input values, then substitutes the input values into the alert Select statement and executes it. Oracle Alert then performs each member of the action set in sequence, substituting the defined output variables with the exception data returned by the Select statement.

- If the action set contains a message action, Oracle Alert constructs and sends the message(s).

- If the action set contains a SQL statement script action, Oracle Alert executes the SQL statement script.

- If the action set contains an Operating System script action, Oracle Alert executes the commands in the script provided that the user who started the concurrent manager has adequate

privileges to execute all operating system commands defined in the action.

- If the action set contains a concurrent program request action, Oracle Alert submits the request to the concurrent manager.

If one of the action set members is an escalation group, Oracle Alert determines which action to perform based on that action's defined escalation level. If one of the action set members is a threshold group, Oracle Alert performs either the detail or the summary action, depending upon the summary threshold defined and the number of exceptions returned by the Select statement.

▶ **To activate or deactivate the Periodic Alert Scheduler:**

1. Once you have successfully installed Oracle Alert, you need to verify that the Periodic Alert Scheduler is active. Choose View Requests from the Help menu to display the Requests window.

   You should see a concurrent request with a program name called Periodic Alert Scheduler. Its status should be Pending. See: Using the Requests Window, *Oracle Applications User's Guide.*

   You can also navigate to the Concurrent Request form and perform a general query to see if the Periodic Alert Scheduler program is in the concurrent queue. See: Using the Concurrent Request Windows, *Oracle Applications User's Guide.*

2. If the Periodic Alert Scheduler's status is not Pending, navigate to the Schedule Alert Programs form.

3. Select Periodic Alert Scheduler.

4. Choose Activate to start the Periodic Alert Scheduler immediately or enter a value in the Active Date field and then choose Activate to schedule it to start later.

   When you start the Periodic Alert Scheduler, Oracle Alert submits it as a concurrent request to the concurrent manager to run every 24 hours at 12 AM. Every time the Periodic Alert Scheduler runs, it submits a request to the concurrent manager to be run again the next day at 12 AM.

5. If the Periodic Alert Scheduler gets deleted from the concurrent queue for any reason, simply restart it from the Schedule Alert Programs form.

6. If you need to deactivate the Periodic Alert Scheduler, choose Deactivate.

See Also

Creating a Periodic Alert: page 3 – 5

Action Escalation: page 5 – 9

Defining a Summary Threshold: page 4 – 12

Checking a Periodic Alert

When you want to check an on demand periodic alert or any other periodic alert or periodic set, you use the Request Periodic Alert Check form to submit a request to the concurrent manager to run a concurrent program called Check Periodic Alert (ALECDC). The concurrent manager runs this request according to its priority in the concurrent queue.

The Check Periodic Alert concurrent program checks a periodic alert and all its enabled action sets.

▶ **To check a periodic alert or periodic set:**

1. Define a periodic alert or periodic set.

2. Navigate to the Request Periodic Alert Check form.

```
┌─────────────────────────────────────────────────────────────────┐
│ □            Request Periodic Alert Check                    ▼   │
├─────────────────────────────────────────────────────────────────┤
│                                                                   │
│   Application  ┌───────────────────────────────────────────┐     │
│                └───────────────────────────────────────────┘     │
│        Alert   ┌───────────────────────────────────────────┐     │
│                └───────────────────────────────────────────┘     │
│   Start Date   │ 29-AUG-1995 00:00:00      │                     │
│   Request ID   ┌───────────────────────────┐                     │
│                └───────────────────────────┘                     │
│                              ┌──────────────────────────────┐    │
│                              │      Submit Request           │   │
│                              └──────────────────────────────┘    │
│                                                                   │
└─────────────────────────────────────────────────────────────────┘
```

3. Enter the name of the application that owns the periodic alert or periodic set you want to check.

4. Enter the name of the periodic alert or periodic set you want to check.

5. Enter the date and time that you want the periodic alert or periodic set to be checked. The default is the current date, but you can specify a future date and time, using the format DD–MON–YYYY HH24:MI:SS.

6. Choose Submit Request to submit a request to the concurrent manager to run the concurrent program Check Periodic Alert. 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 form.

   Note: If the periodic alert is not an on–demand periodic alert, it also runs according to its defined schedule.

See Also

Creating a Periodic Alert: page 3 – 5

Defining a Periodic Set: page 4 – 19

Using the Concurrent Request Windows
(*Oracle Applications User's Guide*)

Checking an Event Alert

How Check Event Alert Works

Once you define an event alert to monitor a table for inserts and/or updates, any insert or update to the table will trigger the event alert. When an insert or update to an event table occurs, Oracle Alert submits to the concurrent manager, a request to run a concurrent program called Check Event Alert (ALECTC). The concurrent manager runs this request according to its priority in the concurrent queue. When the request is run, Check Event Alert executes the alert Select statement. If the Select statement finds exceptions, Check Event Alert performs the actions defined in the enabled action set(s) for the alert. If the Select statement does not find any exceptions, Check Event Alert performs the No Exception actions in the enabled action set(s) for the alert.

How to Optimize Event Alert Request Handling

You can have your system administrator configure your 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.

See Also

Creating an Event Alert: page 3 – 11

Specializing Managers to Run Only Certain Programs
(*Oracle Applications System Administrator's Guide*)

Concurrent Managers Window
(*Oracle Applications System Administrator's Guide*)

The Response Processor

The Response Processor (ALPPIM) is the Oracle Alert concurrent program that processes responses to an alert message. The Response Processor runs according to the schedule you define in the Response Processing alternative region of the Oracle Alert Options form. You define the Response Processor's start time, end time, and check intervals. If you do not define a response processing schedule, the

Response Processor runs every day at midnight. See: Defining Response Processing Options: page 8 – 9.

☞ **Attention:** If you change the response processor interval in the Oracle Alert Options form, you need to restart the Response Processor in the Schedule Alert Programs form. Oracle Alert submits the Response Processor to the concurrent manager once for each response account according to the new schedule.

How the Response Processor Interprets Responses

Each Oracle Alert installation has a defined Response Mail Account. This definition resides in the Oracle Office Options alternative region of the Oracle Alert Options form. See: Defining Oracle Office Accounts: page 8 – 5.

Each Oracle Office response account has three electronic folders that it uses to process responses: the Inbox folder, the Reviewed folder, and the Reviewed_OK folder. When the Response Processor runs, it looks in the Inbox folder and moves all the messages it finds to the Reviewed folder. Once the messages are in the Reviewed folder, the Response Processor looks for messages that contain message handles. For Oracle Alert to recognize a message in the Inbox folder of the response account as a response message, that message must contain a message handle. If the recipient used Reply–with–Original or Forward to include the original message with her or his response, then Oracle Alert can locate the message handle and interpret the response.

Once Oracle Alert recognizes a message by its message handle, it reads the response and tries to match the first word or the first 30 continuous characters of the response to any of the defined valid responses. If Oracle Alert finds that the actual response matches a valid response, then it performs the actions defined for that valid response. Once Oracle Alert performs the actions for that valid response, it closes the response so that no other responses are processed for that message, then moves the message to the Reviewed_OK folder.

If the actual response does not match any of the valid responses in that response set, Oracle Alert performs the actions associated with the invalid response, but leaves the message in the Reviewed folder. Oracle Alert performs the invalid response actions each time it receives an invalid response within the follow–up days specified in the original message action. After the number of follow–up days have passed, Oracle Alert performs the no response follow–up actions for each outstanding response.

If Oracle Alert never receives a response to the alert message, it performs the no response follow–up actions for each unreturned message once the number of response days has passed, and closes responses for all messages in that response set.

Each time the Response Processor completes, the Inbox of Oracle Alert's response account is empty.

> **Note:** Oracle Alert provides you with a precoded alert that you can use to purge the Reviewed and Reviewed_OK mail folders. See: Oracle Alert Precoded Alerts: page 9 – 2.

▶ **To activate or deactivate the Response Processor:**

1. Navigate to the Schedule Alert Programs form.

Schedule Alert Programs		
Program	**Status**	**Request ID**
Internal Manager	Active	
Periodic Alert Scheduler	Active	221994
Response Processor	Active	221992

Active Date [   ]

[**D**eactivate]   [**A**ctivate]

2. Select Response Processor.

3. Choose Activate to start the Response Processor immediately or enter a value in the Active Date field and then choose Activate to schedule it to start later.

   When you start the Response Processor, Oracle Alert submits it as a concurrent request to the concurrent manager to run at the next scheduled interval.

4. If the Response Processor gets deleted from the concurrent queue for any reason, simply restart it from the Schedule Alert Programs form.

Note: If you change the Response Processor interval in the Oracle Alert Options form, use the Schedule Alert Programs form to restart the Response Processor.

5. If you need to deactivate the Response Processor, choose Deactivate.

See Also

Overview of Response Processing: page 6 – 2

Reviewing Alert Check History

You can review the results of any alert check by using the Review Alert Checks window in the Review Alert History form. You can review all checks for an alert, or just those that occurred during a range of dates. See: Reviewing Alert Checks: page 5 – 21.

Oracle Alert provides you with complete information about each alert check. You see the ID of the application user who submitted the alert check request, the date the request was run, the time the request completed, the request ID, and the status of the request. If the history of the alert check has been purged, Oracle Alert displays the purge date. You can also choose to review the complete log file for the check request by choosing Request Log from the Special menu in the Review Alert Checks window.

Review Action Set Check History

Oracle Alert also displays information about the action sets checked during an alert check. You can review the number of exceptions found as well as the number of actions Oracle Alert performed for each action set.

Purging Alert Check and Action Set Check History

You can keep your Oracle Alert tables clear of old alert check and action set check data by using Oracle Alert's precoded Purge Alert and Action Set Checks alert. This is a periodic alert that you can set to run regularly, or that you can check on–demand whenever you like. See: Oracle Alert Precoded Alerts: page 9 – 2.

Setting Up Oracle Alert

This chapter describes additional ways you can set up your Oracle Alert installation to meet your organization's exception–reporting needs.

Setting Up Oracle Alert

Before you define an alert, you may want to review this section to make sure you have specified the necessary options to customize your Oracle Alert installation. This section describes:

- How to set up and maintain your Oracle Alert options

- How to define alerts for custom Oracle applications

- How to use Oracle Alert with multiple application installations

- How to use Oracle Alert naming conventions

Specifying Oracle Alert Options

You use the Oracle Alert Options form to define and maintain the options Oracle Alert uses when checking your alerts. This form contains five alternative regions that each contain multiple options that you can set:

- **Mail Systems**—to specify the mail systems you want to use with Oracle Alert to send outgoing mail.

- **Oracle Office Options**—to define the options and Oracle Office accounts you want Oracle Alert to use to send messages and process responses.

- **Message Elements**—to specify the boilerplate text you want to appear in your message actions.

- **Response Processing**—to specify the options you use with response processing.

- **More Options**—to specify other miscellaneous options for Oracle Alert.

Before you use this form to define Oracle Office for Oracle Alert, you should make certain that your Oracle Office Administrator performs the following steps:

- Installs Oracle Office in a database accessible via SQL*Net from the database in which Oracle Alert is installed, or in the same database in which Oracle Alert is installed.

- Defines an Oracle Office electronic mail ID from which all Oracle Alert messages will be sent.

- Notifies you of the Oracle Office electronic mail ID, the remote password, and the network connect string, if necessary, for accessing the Oracle Office database from Oracle Alert.

    Note: Oracle Alert–specific options appear only in the Oracle Alert Options form. Profiles options that are common to all Oracle Applications appear in the Personal Profile Values and System Profile Values forms. See: Common User Profile Options, *Oracle Applications User's Guide*. See: Profile Options in Oracle Application Object Library, *Oracle Applications System Adminstrator's Guide.*

Defining Mail Systems for Oracle Alert

You can define the electronic mail system you want to use with Oracle Alert for outgoing mail.

▶ **To define a mail system for Oracle Alert:**

1. Navigate to the Oracle Alert Options form.

2. Display the Mail Systems alternative region.

3. The Name field displays the names of the electronic mail systems you can use with Oracle Alert. Oracle Alert predefines three mail systems for you:

   - Oracle Office

   - Unix mail

   - VMS mail

4. Add the names of any other local mail systems you want to use.

   Note: Oracle Alert can communicate directly with a local electronic mail system to send outgoing mail if the local sendmail command can accept arguments from Oracle Alert that consists of a filename that indicates the body of the mail message and a recipient list.

5. In the Command field, enter the operating system command you use to invoke your mail system.

For Unix Mail, Oracle Alert displays the full path name of the executable file. For VMS mail, Oracle Alert displays the command *MAIL.* Oracle Office does not require any value in this field.

6. In the Parameters field, enter the parameters you want to pass to your mail system.

☞ **Attention:** If you use another mail system, Oracle Alert will append the name of the file where the message body is written and a comma separated list of message recipients after any parameters you define. Please ensure that your mail system can accept this information.

7. Check the In Use check box to enable the mail system. You can enable only one mail system at a time. Oracle Alert displays the name of the enabled system first, then displays the remaining systems alphabetically by name.

8. Save your changes.

Defining Oracle Office Accounts

You can specify the Oracle Office accounts you want Oracle Alert to use to send messages and process responses. The Response Processor can only process responses from an Oracle Office account. The Response Processor processes responses in all mail accounts that you define in the Oracle Office Options alternative region of the Oracle Alert Options form. You can define as many application–specific accounts for Oracle Alert to use as you need.

▶ **To define Oracle Office options and accounts:**

1. Navigate to the Oracle Alert Options form.

2. Display the Oracle Office Options alternative region.

3. In the Value field for the Oracle Office Database Name option, enter a SQL*Net connect string or alias that accesses your Oracle Office database. See the *SQL*Net User's Guide* for your protocol.

   If you want to access a local Oracle Office database or if Oracle Office is installed in the same database as Oracle Alert, simply leave the value for this option blank.

4. For the Mail Priority option, choose the value Low, Normal, or High to flag the message action as either low, normal, or high priority once it is received by a recipient in Oracle Office. Note that this option does not affect the way the message is processed or delivered.

5. In the Accounts block, Oracle Alert automatically displays in the first two rows, two default Oracle Office accounts: the account Oracle Alert uses to send messages, and the account Oracle Alert uses to process responses.

   You can define additional Oracle Office accounts in this block to process responses that are specific to an Oracle Application.

   Note: For response processing, if an alert message action does not have a Reply To value specified, Oracle Alert displays one of two following values in the Reply To field when it sends a message:

   – The name of an application–specific account specified in the Oracle Office Options alternative region, if one is supplied for the application that owns the alert

– Or, the name of the default response mail account specified in the Oracle Office Options alternative region, if no application–specific account is specified for the application that owns the alert

6. In the Application field, enter the name of the application for which you want to specify an Oracle Office account for Oracle Alert to use to process responses for that application.

7. In the Installation field, specify the Oracle ID where the application you want to process responses for resides. Oracle Alert displays Send Mail Account and Response Mail Account in this field for the two default accounts.

8. In the Username and Password fields, enter the username and password of the mail account you want Oracle Alert to use for this application. For security, Oracle Alert does not display the password.

9. Save your changes.

Defining Boilerplate Message Text

You can use the Message Elements alternative region of the Oracle Alert Options form to define constant text that you want to appear in your message actions.

▶ **To define constant text in your message actions:**

1. Navigate to the Oracle Alert Options form.

2. Display the Message Elements alternative region.
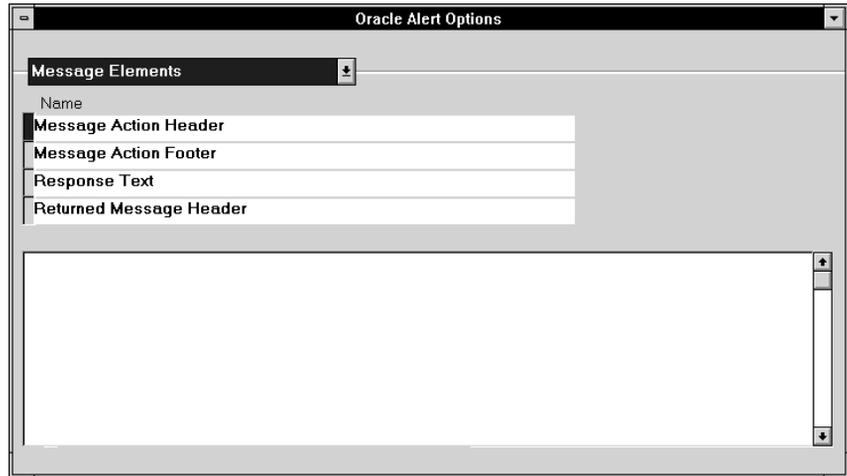
3. Select the message text option you want to specify constant text for. There are four message text options that you can edit:

Message Action Header
Standard text that appears in the beginning of every message action you define in Oracle Alert. (Oracle Alert does not add a header to a message action that is an external file.)

Message Action Footer
Standard text that Oracle Alert appends to the end of every message action you define in Oracle Alert. (Oracle Alert does not append a footer to a message action that is an external file.)

Response Text
Standard text that instructs your message recipients how to respond to alert messages. Oracle Alert appends this text to every message it sends that is associated with a response set. After appending this text to the end of the message, Oracle Alert lists the valid responses defined for the message. Oracle Alert supplies you with default Response Text; you can use this default text or modify it to create your own.

Returned Message Header
Standard text that lets respondents know that their reply to an alert message was not understood, and that they must reply again. Oracle Alert adds this text to the beginning of every Return Original Message invalid response action. Oracle Alert supplies default Returned Message Header text; you can use this default text or modify it to create your own.

Note: If you edit the text for the Message Action Header, Message Action Footer, or Response Text options, messages that you reconstruct using the Review Alert Actions window will show the current text, not the version of the text that was sent with the original message.

4. Enter or edit the text for the selected option in the Text field below.

5. Save your changes.

See Also

Overview of Response Processing: page 6 – 2

Defining Response Processing Options

You can use the Response Processing alternative region in the Oracle Alert Options form to specify your response processing options.

▶ **To define response processing options:**

1. Navigate to the Oracle Alert Options form.

2. Display the Response Processing alternative region.
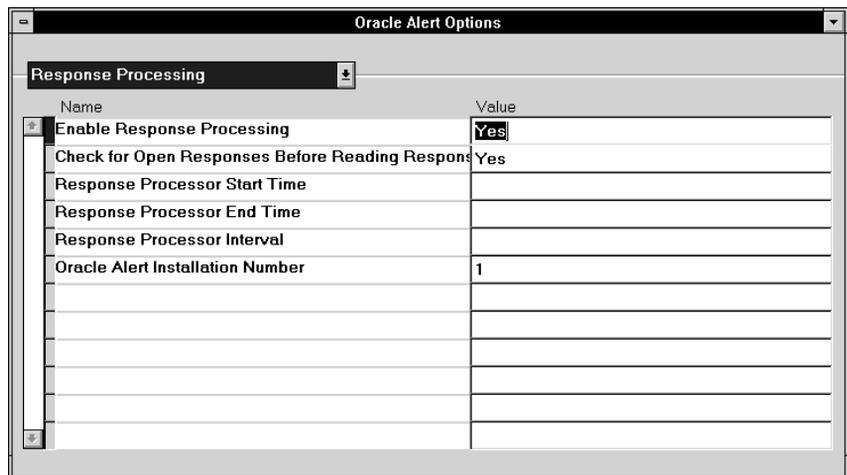
3. Select the option whose value you want to specify or edit.

4. Enter the value for the option in the Value field.

5. There are six response processing options you can modify:

Enable Response Processing Use this option to enable response processing in this installation of Oracle Alert. Enter Yes or No.

Check for Open Responses Before Reading Responses Use this option to enhance performance efficiency. Enter Yes if you want Oracle Alert to open its Oracle Office mailboxes only if there are open responses. The default is Yes.

Response Processor Start Time Use this option to specify the time of day that Oracle Alert first submits the Response Processor to the concurrent manager. The format is HH24:MI.

Response Processor End Time Use this option to specify the time of day that Oracle Alert last submits the Response Processor to the concurrent manager. The format is HH24:MI.

Response Processor Interval Use this option to specify the number of minutes between submissions of the Response Processor.

> **Note:** If you do not enter a Start Time, End Time, and Interval for the Response Processor, Oracle Alert submits the program at midnight of each day.

> **Note:** If you change the Response Processor interval, restart the Response Processor in the Schedule Alert Programs form. Oracle Alert will submit the Response Processor once for each response account according to the new schedule. See: To activate or deactivate the Response Processor: page 7 – 10.

Oracle Alert Installation Number Use this option to specify the number of this Oracle Alert installation. Oracle Alert uses this number when multiple Oracle Alert installations are using the same mailboxes for response processing. This lets Oracle Alert determine where each alert message originated from. Each installation must be uniquely numbered. The default is 1.

See Also

Overview of Response Processing: page 6 – 2

Defining Other Options for Oracle Alert

You can use the More Options alternative region in the Oracle Alert Options form to define other miscellaneous options for your Oracle Alert installation.

▶ **To define other miscellaneous options:**

1. Navigate to the Oracle Alert Options form.

2. Display the More Options alternative region.
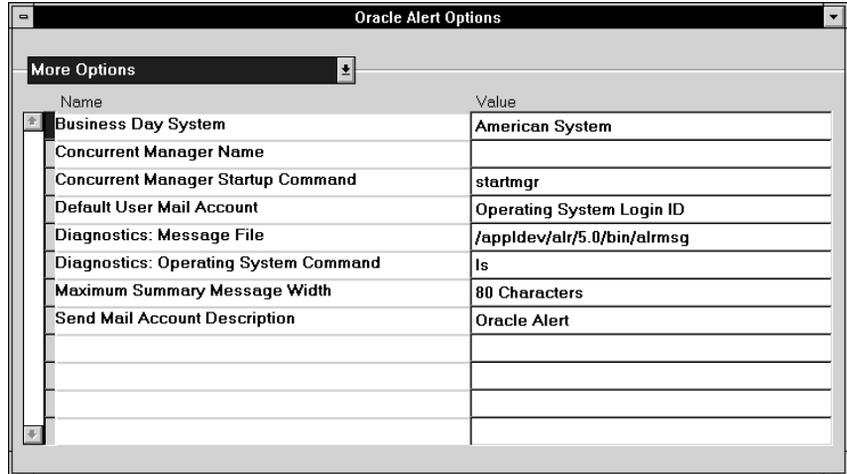
Oracle Alert Options	
More Options	
Name	Value
Business Day System	American System
Concurrent Manager Name	
Concurrent Manager Startup Command	startmgr
Default User Mail Account	Operating System Login ID
Diagnostics: Message File	/appdev/alr/5.0/bin/alrmsg
Diagnostics: Operating System Command	ls
Maximum Summary Message Width	80 Characters
Send Mail Account Description	Oracle Alert

3. Select the option whose value you want to specify or edit.

4. Enter the value for the option in the Value field.

5. There are eight miscellaneous options you can modify:

Business Day System   Use this option to select the method for determining business days. The American system starts with Sunday and defines Monday through Friday as business days. The European system starts with Monday and defines Monday through Friday as business days. The Middle Eastern System starts with Monday and defines Monday through Saturday as business days.

Concurrent Manager Name   Use this option to specify the name of the concurrent manager.

Concurrent Manager Startup Command	Use this option to specify the name of the operating system command that starts the concurrent manager. The default value is *startmgr*.
Default User Mail Account	When a user triggers an event alert, Oracle Alert uses the value entered in the E–Mail field of the Users form (available from the System Administrator responsibility) for that user as the default value for the :MAILID implicit input. Use this option to define the value you want Oracle Alert to use if the E–Mail field is null. You can choose Operating System Login ID, or Application Username for the user who triggered the event.
Diagnostics: Message File	Use this option to specify the full path and file name of the diagnostics message file. The suggested file is called *alrmsg*, located in the *bin* subdirectory under your Alert directory.
Diagnostics: Operating System Command	Use this option to specify the operating system command used in operating system actions in the predefined diagnostic alerts. The default value is *ls*.
Maximum Summary Message Width	Use this option to specify the maximum number of characters your printers can print across the top of a page. Oracle Alert uses this value as the default message width when you define summary message actions using the Action Details window of the Alerts form. Your choices are 80, 132, or 180 characters. Depending on your printer, 80 usually results in a portrait–mode report (short end of paper across top of page), 132 and 180 usually result in a landscape–mode report (long end of paper across top of page).
Send Mail Account Description	Use this option to enter the text you want Oracle Alert to display as the sender of alert message actions. When you install Oracle Alert, the default value is *Oracle Alert*, which you can modify.

Defining Alerts for Custom Oracle Applications

Oracle Alert integrates fully with your custom Oracle applications, so you have a quick and easy way to monitor the exceptions in your custom Oracle databases.

A custom application is an application that was not installed using AutoInstall or, any application that resides outside the database where Oracle Alert resides. For example, an Oracle Application that resides in a different database from Oracle Alert is considered a custom application. Using database links, Oracle Alert can monitor any custom application as if it resides in the same database as Oracle Alert.

▶   **To define a periodic alert for your custom application:**

1. Use SQL*Net to create a database link from the Oracle Alert database to the custom Oracle database:

    ```
    CREATE DATABASE LINK linkname
    CONNECT TO username IDENTIFIED BY password
    USING 'connect string';
    ```

    Note: Replace `'connect string'` with a SQL*Net V1 connect string or a SQL*Net V2 alias. Refer to the *SQL*Net User's Guide* for your protocol for more information.

    Here is an example if you are using SQL*Net V1:

    ```
    CREATE DATABASE LINK CUSTOM_APPL
    CONNECT TO SCOTT IDENTIFIED BY TIGER
    USING 't:hqfin:CUSTOM_DB';
    ```

    Note: If your custom application resides in the same database as Oracle Alert, you need not define a database link.

2. Optionally, you can create synonyms for each table in your custom application so that you can reference the table using the synonym instead of the table name and database link. That way, the users defining alerts and actions do not need to know the location of the custom application, just the table names. See: *SQL Language Reference Manual.*

3. Register your custom application with Oracle Alert so that it is available in the list of values for the Application field of the Alerts form. See: Applications Window: page 8 – 21.

4. Register your custom application's Oracle ID with Oracle Alert. See: Registering Application Installations: page 8 – 24.

5. Once you complete these steps you can define periodic alerts for your custom application. See: Creating a Periodic Alert: page 3 – 5.

▶ **To define an event alert for your custom application:**

1. If you want to create event alerts for your custom Oracle applications, you need to complete the following steps.

☞ **Attention:** If you want to create event alerts for Oracle Applications that reside on another database, you must install Oracle Alert on that database.

2. Register your custom application with Oracle Alert so that it appears in the list of applications available in the Alerts form. See: Applications Window: page 8 – 21.

3. Register your custom application's Oracle ID with Oracle Alert. See: Registering Application Installations: page 8 – 24.

4. Use the Table Registration API to register the table(s) that you want Oracle Alert to monitor for inserts and updates. See: Table Registration API, *Oracle Applications Coding Standards*.

5. Once you complete these steps you can define event alerts for your custom application. Oracle Alert creates the necessary insert or update database trigger in your custom application table for your event alert. See Creating an Event Alert: page 3 – 11.

Creating a Responsibility for Your Custom Application Alert Manager

Define a product specific responsibility for your custom application and assign this responsibility to the person who creates the alerts for the custom application. That way, only this person can define and maintain the alerts, actions, and response sets for your custom application. Your System Administrator can do this for you.

▶ **To create a responsibility for your custom application alert manager:**

■ Connect an application–specific responsibility to the Oracle ID of the application, not to Oracle Alert's Oracle ID. See: Define Responsibility, *Oracle Applications System Administrator's Guide.*

Oracle Alert and Multiple Application Installations

Normally, when a periodic alert action runs against more than one installation of an application, any concurrent program that is requested by the alert action will connect to the same application–Oracle ID pair each time the program runs.

That is because the concurrent program connects to an Oracle ID based on the data group assigned to the Alert Manager responsibility. The application that owns the concurrent program is always paired with an Oracle ID in the data group.

If you want the concurrent program requested by an alert action to connect to more than one Oracle ID, you need to perform two tasks. First, you need to create an additional application–Oracle ID pair for the Oracle ID you want the alert action to connect to and add it to the data group assigned to the Alert Manager responsibility. Second, you need to copy (transfer) the alert definition so that its alert action can be directed towards the new application–Oracle ID pair.

Figure 8 – 1

Designating a Concurrent Request Action to a New Application–
Oracle ID Pair

System Administrator Tasks

Windows:

Applications
1. Register a new application.

NEW APPLICATION

Concurrent Programs
2. Make a copy of the concurrent program and link the new application to it.

NEW APPLICATION → NEW (COPY OF) CONCURRENT PROGRAM

Data Groups
3. In the Data Group associated with the Alert Manager responsibility, pair the new application with the Oracle ID you want it to connect to.

NEW APPLICATION → Oracle ID
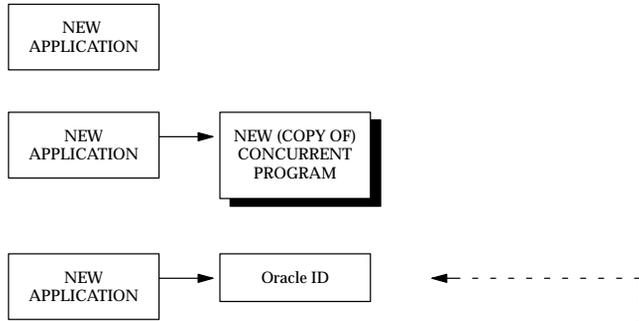
Alert Manager Tasks

Windows:

Transfer Alert Definition
4. Transfer (copy) your alert definition and link it with the new application.

NEW (COPY OF) ALERT DEFINITION → NEW APPLICATION

Alert Details
5. Link your new alert definition with the Oracle ID you want to connect to.

NEW (COPY OF) ALERT DEFINITION → Oracle ID

Action Details
6. Link the concurrent request action with the new application.

ALERT ACTION → NEW (COPY OF) CONCURRENT PROGRAM → NEW APPLICATION
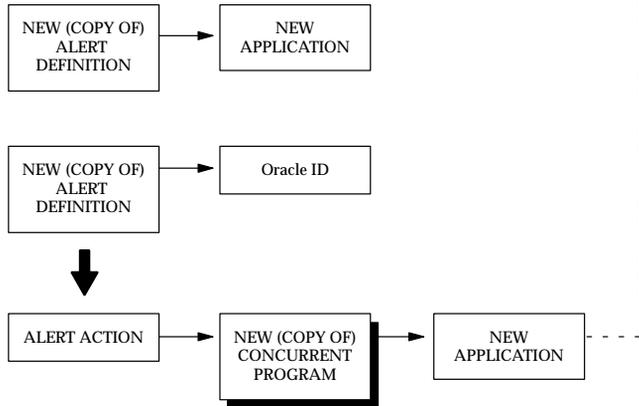
When the concurrent request action runs, the concurrent program connects to the desired Oracle ID, courtesy of the Alert Manager's Data Group.

Defining Data Groups
(*Oracle Applications System Administration Reference Manual*)

▶ **To add a new application–Oracle ID pair to the Alert Manager responsibility data group:**

1. Sign on to Oracle Applications and choose the System Administrator responsibility.

2. Navigate to the Applications form.

3. Use this form to register a new application with Oracle Applications, which includes creating an application name and short name, identifying a base path, and so on. See: Applications Window: page 8 – 21.

4. Save your changes.

5. Navigate to the Concurrent Programs form.

6. Query the concurrent program you wish to run from your periodic alert action and choose Copy to...

7. In the Copy to window, enter the information necessary to copy the concurrent program to a new concurrent program name and associate it with the new application you just registered. See: Concurrent Programs Window, *Oracle Applications System Administrator's Guide.*

8. Choose OK to copy the concurrent program.

9. Navigate to the Data Groups form.

10. Query the data group that the Oracle Alert Manager responsibility is associated with.

11. In the Application–Oracle User Name Pairs block, insert a new row to add a new application–Oracle ID pair. Add the new application you just registered, and associate it with the application Oracle ID you wish to run your concurrent program against. See: Data Groups, *Oracle Applications System Administrator's Guide.*

12. Save your changes.

▶ **To direct an alert action to a new application–Oracle ID pair:**

1. Switch to the Alert Manager responsibility.

2. Navigate to the Alerts form.

3. Query for the periodic alert whose alert action you want to direct to the new application–Oracle ID pair.

4. Choose from the Special menu, Transfer Alert.

5. In the Transfer Alert Definitions window, copy (transfer) your periodic alert definition to a new periodic alert. You do not have to change the name of the alert, although you may want to change the name to indicate which Oracle ID it is connecting to. Associate the new alert with the new application you registered in the previous task.

6. Associate the periodic alert with the same database by entering the database username and password. Include any necessary SQL*Net syntax with the database information.

   Note: On the VMS platform, this may cause an error condition to occur. If this occurs, consult your VMS database administrator.

7. Choose Transfer to complete the transfer of the alert definition.

8. Query for your new alert definition in the Alerts form.

9. Choose Alert Details.

10. Display the Installations alternative region in the Alert Details window.

11. To change the installation that your new periodic alert runs against, enter in the Oracle ID field, the Oracle ID that you recently added to the Alert Manager responsibility's data group and check Enabled.

12. Save your changes.

13. Choose Actions from the Alerts form.

14. Select the concurrent request action that you want to run against the new Oracle ID and choose Action Details.

15. Update the application to the new application you registered in the previous task.

16. Update the program name to the new concurrent program you copied in the previous task.

17. Save your changes.

Oracle Alert Naming Conventions

This section provides you with guidelines for naming your alerts, actions, action sets, escalation groups, and threshold groups. These guidelines help you give your alerts and alert components names that are unique, yet intuitive and easily understood by you and other Oracle Alert users.

General Naming Conventions

Follow these general naming conventions throughout Oracle Alert to take advantage of Oracle Alert's information search capabilities and list of values features.

Unique Initial Letters

Develop names with unique initial letters so you can autoreduce a list of values when entering names in fields that contain lists. Instead of naming two alerts Invoice Receipt Holds and Invoice Quantity Holds, choose names that adequately describe the alert but avoid the conflict of similar initial letters. Simply name them Receipt Holds and Quantity Holds. To select the Receipt Holds alert, you need to enter only one letter, R, to uniquely identify the alert and let Oracle Alert automatically enter the full name, Receipt Holds for you.

While avoiding similar initial letters is not always possible, the fewer characters you need to enter to establish a unique choice, the faster you can enter or query information. Entering action set names of Auditor and Administrator is easier than entering action set names of Payables Auditor and Payables Administrator. In the former case, you need only enter two characters, Au or Ad, to choose the correct action set.

⚠ **Warning:** Use characters other than the pound sign (#), the colon (:), or the percent sign (%) for your initial letter; as these characters are reserved for use in Enter Query mode.

Key Words

Develop names with memorable key words, or parts of words, somewhere in the name. For example, to take advantage of Oracle Alert's information search capabilities, you might describe all alerts that relate to invoices so that they have the word "invoice" somewhere in the name. Then, when you search for these alerts, you can simply

enter your search criteria as "%invoice%" to retrieve all your alerts on invoices.

Alert–Specific Naming Conventions

Names

- Capitalize the first letter of every word.

- When naming an alert, do not use the word "Alert" in the name (e.g., New Employee Alert). If possible, use the name of the alert as the subject heading for any associated message actions.

  Example: New Employee Notification

- When naming an action set, do not use the words "Action" or "Set". Relate the name of the action set to either the types of actions or the type of recipients. Try to limit the name to one or two words.

  Example: Concurrent Requests
  Purchasing Manager

- When naming a distribution list, make it plural unless it will always refer to only one person.

  Example: Collections Managers

Descriptions

- Capitalize only the first letter of the first word.

- When writing an alert description, make the first word a verb.

  Example: Informs the appropriate staff of new employees

- When writing a distribution list description, do not use the word "List". Instead, use the description to fully explain the use.

  Example: Payroll managers at headquarters

See Also

Using a List of Values
(*Oracle Applications User's Guide*)

Applications Window

When you define a custom application, you supply several pieces of information to Oracle Applications. You must register your application name, application short name, application basepath, and application description with Oracle Application Object Library. Oracle Application Object Library uses this information to identify application objects such as responsibilities and forms as belonging to your application. This identification with your custom application allows Oracle Applications to preserve your application objects and customizations during upgrades. The application basepath tells Oracle Application Object Library where to find the files associated with your custom application.

You can use your custom application to name your custom menus, concurrent programs, custom responsibilities, and many other custom components. For some objects, the application part of the name only ensures uniqueness across Oracle Applications. For other components, the application you choose has an effect on the functionality of your custom object.

Prerequisites

❑ If your application resides in a database other than the database where Oracle Alert resides, you must create a database link.

Applications Block

When you register a custom application, you provide the information Oracle uses to identify it whenever you reference it. Although you can change the name of an application, doing so may cause a change in the application code where you hardcode your application name. For example, if you pass program arguments through the menu that have application name hardcoded, you will also have to update them.

☞ **Attention:** You should not change the name of any application that you did not develop, as you cannot be sure of the consequences. You should never change the name of any Oracle Applications application, because these applications may contain hardcoded references to the application name.

Application

This user–friendly name appears in lists seen by application users.

Short Name

Oracle Applications use the application short name when identifying forms, menus, concurrent programs and other application components. The short name is stored in hidden fields while the name displays for users.

Your short name should not include spaces. You use an application short name when you request a concurrent process from a form, and when you invoke a subroutine from a menu.

💡 **Suggestion:** Although your short name can be up to 50 characters, we recommend that you use only four or five characters for ease in maintaining your application and in calling routines that use your short name. To reduce the risk that your custom application short name could conflict with a future Oracle Applications short name, we recommend that your custom application short name begins with "XX".

Basepath

Enter the name of an environment variable that represents the top directory of your application's directory tree. Oracle Applications search specific directories beneath the basepath for your application's executable files and scripts when defining actions that reside in external files.

In general, your application's basepath should be unique so that separate applications do not write to the same directories.

However, you may define custom applications that will be used only for naming your custom responsibilities, menus and other components. In this case, you can use the basepath of the Oracle application that uses the same forms as your application. For example, if you are defining a Custom_GL application, you could use the GL_TOP basepath for your custom application.

See: Development Environment
    (*Oracle Applications Installation Manual)*

Registering Application Installations

If you wish to define alerts for a custom application, you need to register the installation of the custom application with Oracle Applications first. A custom application is any application that was not installed using Autoinstall and resides in the same database as Oracle Alert, or any application that resides in a database other than the one in which Oracle Alert is installed.

Before you can register an installation, you must first register your custom application. See: Applications Window: page 8 – 21.

▶ **To register a custom application installation:**

1. Navigate to the Define Application Installations form.
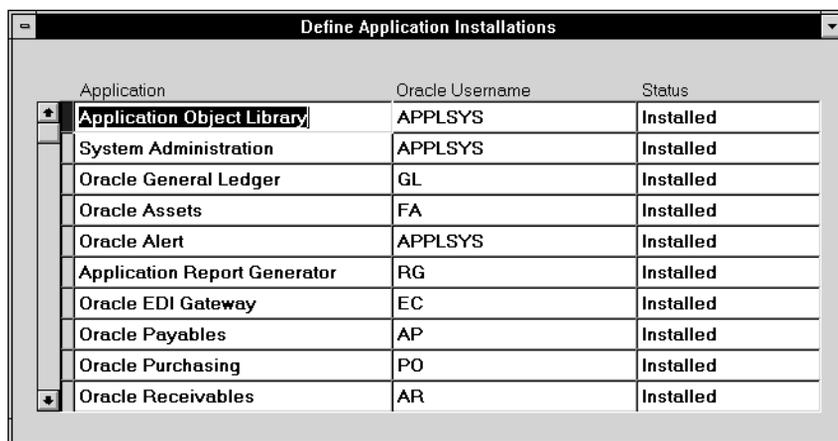
Application	Oracle Username	Status
Application Object Library	APPLSYS	Installed
System Administration	APPLSYS	Installed
Oracle General Ledger	GL	Installed
Oracle Assets	FA	Installed
Oracle Alert	APPLSYS	Installed
Application Report Generator	RG	Installed
Oracle EDI Gateway	EC	Installed
Oracle Payables	AP	Installed
Oracle Purchasing	PO	Installed
Oracle Receivables	AR	Installed

2. Choose from the list of values in the Application field, the name of the custom application installation you are registering.

3. Choose from the list of values in the Oracle Username field, the Oracle ID associated with this custom application.

    If you are registering a custom application installation that resides in a different database, choose Oracle Alert's Oracle ID as the Oracle Username for your application installation.

4. Oracle Alert automatically displays *Custom* in the Status field.

5. Save your changes.

Using Precoded Alerts

This chapter describes the precoded alerts that are included in your
Oracle Alert installation.

Oracle Alert Precoded Alerts

Your Oracle Alert installation contains predefined alerts that are designed to help you manage your database and the data you generate when you use Oracle Alert. Oracle Alert provides eight alerts that systematically monitor your system for potential tablespace, disk space, and allocation problems, making your Database Administrators more efficient, and increasing 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 increasing database performance. Oracle Alert also provides an alert that clears your Oracle Alert electronic mail folders of older messages, keeping your sendmail and response mail accounts to a manageable size.

Customizable Alert Frequencies

All the precoded alerts are periodic alerts, so you determine how often they run. Set them to run daily, weekly, or monthly, according to your needs.

Customizable Alert Inputs

You can use inputs to customize your precoded alerts. You can also define your input values at the action set level, so you can create multiple action sets that target different input values. You can create as many action sets as you need.

This section gives you an overview of these eleven alerts, and suggestions on how to use them to enhance your system performance.

Oracle Alert DBA Alerts

Oracle Alert Database Administration (DBA) alerts help you manage your database by notifying you regularly of:

- 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

If Oracle Alert finds the database exceptions specified in a DBA alert, it sends you a message summarizing all exceptions found. If Oracle Alert finds no exceptions, it sends you a message reporting that no exceptions were found. Oracle Alert keeps you notified of the status of your database, even if it is unchanging.

The Applications DBA application owns the Oracle Alert DBA alerts. This lets Oracle Alert perform the DBA alerts for every database instance you create, even those that reside outside Oracle Alert's database.

The following descriptions list the customizable frequency and inputs of each DBA alert.

Tables Unable to Allocate Another Extent

This alert looks for tables where the next extent is larger than the largest free extent.

Frequency	Every N Calendar Days
Inputs	Table Name
	Oracle Username

Indexes Unable to Allocate Another Extent

This alert looks for indexes where the next extent is larger than the largest free extent.

Frequency	Every N Calendar Days
Inputs	Index Name
	Oracle Username

Users Near Their Tablespace Quota

This alert detects users that are near their tablespace quota.

Frequency	Every N Calendar Days
Inputs	Oracle Username
	Tablespace Name
	Check minimum percent free space remaining
	Check maximum percent space used
	Minimum free space remaining (in bytes)

Maximum percent space used

Tablespaces Without Adequate Free Space

This alert looks for tablespaces without a specified minimum amount of free space.

Frequency	Every N Calendar Days
Inputs	Tablespace Name
	Check total free space remaining
	Check maximum size of free extents available
	Maximum size of free extents available (in bytes)
	Minimum total free space remaining (in bytes)

Indexes Too Large or Fragmented

This alert detects indexes that exceed a specified number of blocks or extents.

Frequency	Every N Calendar Days
Inputs	Index Name
	Oracle Username
	Check maximum number of blocks
	Check maximum number of extents
	Maximum number of blocks
	Maximum number of extents

Tables Too Large or Fragmented

This alert detects tables that exceed a specified number of blocks or extents.

Frequency	Every N Calendar Days
Inputs	Table Name
	Oracle Username
	Check maximum number of blocks
	Check maximum number of extents
	Maximum number of blocks

Maximum number of extents

Tables Near Maximum Extents

This alert searches for tables and indexes that are within a specified
number of extents of their maximum extents.

Frequency	Every N Calendar Days
Inputs	Table Name
	Oracle Username
	Minimum number of extents remaining

Indexes Near Maximum Extents

This alert searches for tables and indexes that are within a specified
number of extents of their maximum extents.

Frequency	Every N Calendar Days
Inputs	Index Name
	Oracle Username
	Minimum number of extents remaining

Oracle Alert Purging Alerts

Two of the Oracle Alert precoded alerts are designed to help you
manage the data you generate when you use Oracle Alert. While using
Oracle Alert you should be able to:

- Automatically delete concurrent requests older than a specified
  number of days

- Automatically clean out alert checks and action set checks that
  are older than a specified number of days

The following descriptions list the customizable frequency and inputs
of each purging alert.

Purge Concurrent Requests

This alert looks for concurrent requests and their log and out files that
are older than the number of days you specify, and runs a concurrent
program that deletes them. If you enter a concurrent program name
input, you should use the program name (located in the column

USER_CONCURRENT_PROGRAM_NAME in the table
FND_CONCURRENT_REQUESTS), and not the optional description
that may accompany the concurrent program name in the Concurrent
Requests form.

Alert Type	Periodic
Periodicity	Every N Calendar Days
Inputs	Application Name
	Concurrent Program Name
	Number of days since concurrent request was submitted to the concurrent manager
Operating System Program	Deletes log file, out file, and corresponding record of each concurrent request
Arguments	Concurrent request ID

Note: This precoded alert uses the concurrent program called
Purge Concurrent Request and/or Manager Data Program as
its action. See: Purge Concurrent Request and/or Manager
Data Program, *Oracle Applications System Administrator's Guide.*

Purge Alert and Action Set Checks

This alert looks for alert and action set checks older than the number of
days you specify, and runs a SQL statement script that deletes them.

Alert Type	Periodic
Periodicity	Every N Calendar Days
Inputs	Application Name
	Number of days since alert check

Note: Oracle Alert will not delete alert checks and/or action
set checks for a response processing alert that has open
responses. See: Defining a Response Processing Alert: page
6 – 4.

Oracle Alert Purge Mail Alert

One of the Oracle Alert precoded alerts is designed to help you keep
your Oracle Office folders to a manageable size. In particular, if you
are using response processing, you will want to keep your response

account(s) clear of old messages. While using Oracle Alert you should be able to:

- Automatically delete old, obsolete mail messages from your Oracle Office accounts defined for Oracle Alert

- Specify which Oracle Office accounts and the Oracle Office folders you want to clear of old messages

- Determine which messages you want to delete

The following description provides the customizable frequency and inputs of the purge mail alert.

Purge Oracle Office Messages

Frequency	Weekly
Inputs	Expiration Days
	Folder
	Oracle Office Account

A

Menu Appendix

This appendix lists the forms available in the Oracle Alert Manager responsibility along with their navigation paths.

Oracle Alert Menu Paths

This section shows you the first window title and default menu path for each Oracle Alert form. In addition, we provide a page number reference for the description of each form in this manual, or a reference for the descriptions of forms that are located in other manuals.

See...	Refer to this manual for a complete form description
User	*Oracle Applications User's Guide*
SysAd	*Oracle Applications System Administrator's Guide*

First Window Title	Standard Menu Path
Alerts: page 3 – 2	Alert Define
Applications: page 8 – 21	System Applications
Completed Requests: *User*	Other Requests View
Concurrent Requests: *User*	Other Concurrent
Concurrent Requests: *SysAd*	Request View (with System Administrator functionality)
Define Application Installations: page 8 – 24	System Installations
Distribution List: page 4 – 3	Distribution List
Oracle Alert Options: page 8 – 3	System Options
Periodic Set: page 4 – 19	Alert Periodic Set
Personal Profile Values: *User*	Other Profile
Request Periodic Alert Check: page 7 – 6	Request Check
Request Set: *User*	Other Requests Set
Review Alert History: page 5 – 15	History
Schedule Alert Programs: page 7 – 3	Request Schedule
Submit Requests: *User*	Other Requests Submit

Glossary

:DATE_LAST_CHECKED An Oracle Alert implicit input that contains the date and time that an alert was last checked. Oracle Alert automatically provides the value for :DATE_LAST_CHECKED. You must use :DATE_LAST_CHECKED to create self–referencing alerts.

Action A message, a SQL statement script, an operating system script, or a concurrent program request that Oracle Alert performs during an alert check or as a result of processing a response.

Action Definition The set of instructions you provide Oracle Alert that tell it how to perform an action. For example, whether you define your action as detail, summary, or no exception is part of the action's definition.

Action Escalation An Oracle Alert feature that allows you to create a set of detail actions of escalating severity levels for Oracle Alert to perform if it finds the same exception during consecutive alert checks. Oracle Alert performs one action during each alert check.

Action Group The collection of actions that comprise a threshold group or an escalation group. An action group can be included in the action set for an alert.

Action History A record of the actual actions performed for each action set check that includes the value substituted for each output.

Action Set The set of inputs, outputs, actions, and action groups for an alert that define the instructions you want Oracle Alert to perform during an action check.

Action Set Check One execution of an alert's Select statement.

Actual Response The reply that an alert message recipient sends to Oracle Alert. Ideally, the actual response matches a defined valid response.

Alert A SQL Select statement that checks your database for certain exceptions and performs actions based on the exceptions found.

Alert Check   One completed request of Check Periodic Alert (ALECDC) or Check Event Alert (ALECTC).

Alert Check History Information that Oracle Alert keeps about each alert check, such as date and time, number of exceptions found, and whether the alert check completed successfully.

Application   A unique Oracle name that identifies a set of forms, reports, and programs.

Application Installation   A unique combination of an application and an Oracle User. Automatically created for Oracle Applications by AutoInstall, you can also create your own application installation for your custom applications.

Application Response Account   An Oracle Office account where users return alert messages with their responses, also known as the response processing Inbox. When the response processor runs, it moves all the messages in this account to the Reviewed folder before determining which messages are responses.

AutoInstall   The Oracle Applications automated installation program.

Check Event Alert (ALECTC)   The Oracle Alert concurrent program that submits a request to the concurrent manager to run an event alert.

Check Periodic Alert (ALECDC)   The Oracle Alert concurrent program that submits a request to the concurrent manager to run a periodic alert that you check on demand.

Column   The data in a message that represents the value Oracle Alert substituted for a single output. A summary message can include many columns, each one using a specified number of characters across, and as many lines down as are needed to fit the information from all the exceptions.

Column Wrap   Column formatting that uses the first character of the column as the left margin and one space before the first character of the next column as the right margin. Oracle Alert uses as many lines as necessary to display or print the output value. For example, if the width of the column in the message is 20 characters, and the length of the selected output is 60 characters, Oracle Alert displays the output information on three lines of 20 characters each.

Concurrent Manager   An Oracle Applications component that manages the queuing of concurrent requests and the operation of concurrent programs.

Concurrent Process   An instance of running a concurrent program. Each time you submit a request, a concurrent manager processes your request, starts a concurrent process, and runs a concurrent program.

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

Concurrent Queue   A list of concurrent requests awaiting completion by a concurrent manager. Each concurrent manager has a queue of requests waiting in line to be run. If your system administrator sets up your Oracle Application to have simultaneous queuing, your request can wait to run in more than one queue.

Concurrent Request   A request to Oracle Alert to complete a task.

Concurrent Request Action   A concurrent request that Oracle Alert submits during an alert check or while processing a response.

Constant Text   Any characters, other than outputs, placed within the Summary Message Template. For example, you can put a dollar sign ($) in the first space of a column that displays money. Also known as boilerplate text.

Critical Output   Output used in an alert summary action. Oracle Alert performs one summary action for each unique combination of critical outputs. For a summary message action, that includes any output that resides outside the summary message template. For a summary concurrent request action, that includes any output used as an argument to a concurrent program request. For a summary operating system script action or summary SQL statement script action, that includes any output used as a file name or as an argument in an external script file or any output used in a script stored in the database.

Custom Oracle Application   ”A custom application is an application that was not installed using AutoInstall, or any application that resides outside the database where Oracle Alert resides. For example, any of the Oracle Applications that resides in a different database than Oracle Alert is considered a custom application.”

Database Link   A logical name you define within your database for another (remote) database that may or may not reside on a different machine. Using SQL*Net, your database can use this name to access information in the remote database as if that information were part of your database.

Detail Action   An action that Oracle Alert performs once for each exception found by the alert Select statement. You must define detail actions if you want to use action escalation.

Distribution List   A predefined list of electronic mail IDs and printer instructions that you can use across multiple message actions, rather than entering the complete set of IDs for each action.

Duplicate Checking   An alert that you define to check for exceptions that remain in your database over time. An exception is a “duplicate” if Oracle Alert encountered the same exception during a previous alert check.

Duplicate Check Period   The number of days of history for an alert that Oracle Alert checks when determining whether an exception is a duplicate.

Duplicate Configuration   The set of outputs that are used to determine whether an exception is a duplicate. You define a duplicate configuration for an alert in the Outputs alternative region located in the Alert Details window of the Alerts form.

Duplicate Exception   An exception that was found during the previous alert check.

Duplicate Suppression   Oracle Alert can suppress the action for an alert if subsequent checks of that alert locate the same exceptions.

Dynamic Distribution   You can use an output to represent a list of electronic mail IDs. When you define a message action with dynamic distribution, Oracle Alert sends the message to all the user IDs on the list.

Enabled Alert   An alert that Oracle Alert performs during an alert check.

Escalation Group   A set of detail actions of the same action type. Oracle Alert performs one of the actions during an alert check of an alert that uses action escalation. Which action Oracle Alert performs depends upon the escalation level of the action, and whether the exceptions found during the alert check are duplicates.

Event Alert   An alert that runs when an insert or update occurs on a database table that you specify. For example, you can define an event alert that immediately sends a message to the system administrator when a user creates a new alert.

Event Table   A database table where the insert or update that triggers your event alert occurs. An event alert fires when an insert or update occurs on the event table.

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

Exception History   The exception values found during an alert check.

Exception Reporting   An integrated system of alerts and actions that focuses attention on time–sensitive or critical information, streamlines your communication channels, shortens your reaction time, and eliminates information clutter. Exception reporting communicates information either by electronic mail or paper reports.

Extent   Logical database space, expressed as a specific number of contiguous data blocks, obtained in a single allocation, used to store a specific type of information.

Implicit Input   A predefined input for which Oracle Alert automatically provides the type and value. The three Oracle Alert implicit inputs are: :MAILID, :ROWID, and :DATE_LAST_CHECKED. The value of each implicit input comes from the alert whose Select statement it belongs to. Oracle Alert never displays implicit inputs outside the Select statement.

Input   A optional parameter that you can use to customize the data retrieved by the alert Select statement. You can set an input to different values depending upon to whom and when the alert is sent. For example, an alert testing for users who need to change their passwords uses the number of days between password changes as an input. You need not specify inputs when you define an alert.

Invalid Response   A response that Oracle Alert receives in reply to an alert message that does not match any specified valid response.

Landscape Format   The message format that displays or prints 132 characters across the top of the page.

MAILID   Represents the application logon ID of the user who enters an insert or update that triggers an event alert.

Message Action   An electronic mail message that Oracle Alert distributes during an alert check or while processing a response.

Message Handle   A unique number that Oracle Alert assigns to an alert message that solicits a response. Oracle Alert places the number in the lower left–hand corner of the message. The message handle must be included in the actual response, for Oracle Alert to process it correctly. In other words, respondents to an alert message must include the original message with their reply.

Multi–Line Exception   The value of an output that spans more than one physical line in a message.

No Exception Action   An action Oracle Alert performs when no exceptions are found during an alert check.

No Response   The actions you want Oracle Alert to perform if the recipient does not respond within the number of Response Days specified in the Action Details window of the Alerts form.

Operating System Script Action   An operating system script that Oracle Alert executes during an alert check or while processing a response.

Output   A variable whose value changes based on the outcome of an alert check. You use outputs in your alert actions to display exception data, or to pass parameters to SQL scripts and operating scripts and concurrent program requests.

Output Value   An exception found by the alert select statement that provides the value for an output.

Periodic Alert   An alert that checks your database for the conditions defined in your Select statement, according to a schedule you determine.

Periodic Alert Scheduler (ALEPPE)   A concurrent program that runs every midnight. It submits, to the concurrent manager, each periodic alert that is scheduled to be checked that day. Then, it resubmits itself to run at midnight. The Periodic Alert Checker also deletes any expired alert history.

Portrait Format   A message format that displays or prints 80 characters across the top of the page.

Process Incoming Messages (ALPPIM)   The Oracle Alert concurrent program that reads the messages in the Oracle Alert response account Inbox and performs the predefined actions for those that received a reply. Also known as the response processor.

Recipient   Anyone that receives an alert message. A recipient may receive a message via electronic mail–in which case the recipient is represented by an electronic mail ID, or via a printer–in which case the recipient is represented by an identifying word that Oracle Alert prints on the burst page of the printed message.

Response Days   The number of days you want Oracle Alert to wait before it takes the No Response Follow–up Actions you specify. You indicate the number of response days in the Response Days field of the Action Details window of the Alerts form.

Response History   Information that Oracle Alert keeps about the responses received and response actions performed for your response processing alerts.

Response Processing   An Oracle Alert feature that lets Oracle Alert process a user's response to an electronic mail message, and perform a set of predefined actions.

Response Processor (ALPPIM)   An Oracle Alert concurrent program that processes responses to alert messages. Also see the term "processing incoming messages".

Response Processor Interval   The frequency with which Oracle Alert submits the response processor to the concurrent manager. You define a start time, an end time, and the number of minutes between.

Response Set   The set of response variables, actions, valid responses, invalid responses, and no response responses defined for a response processing alert.

Response Template   A template containing predefined valid response text. When responding to an alert message action that solicits a response, the alert recipient copies this predefined text template and substitutes the appropriate values for any response variables.

Response Text   Standard text that Oracle Alert appends to every alert message that is associated with a response set. After appending this text to the end of the message, Oracle Alert lists the valid responses defined for the message. Use the Response Text to tell your message recipients how to respond to the alert message.

Response Variable   A variable whose value is supplied by a respondent to an alert message.

Response Variable Value   The actual value a respondent supplies for a response variable. For example, if REORDER_QUANTITY is a response variable, a recipient assigns a numeric value to it; for example, 200. That numeric value is the response variable value.

ROWID   The Oracle Alert implicit input that represents the row ID number in the event table where the insert or update that triggers an event alert occurs.

Select Statement   A SQL Select statement that defines the specific database conditions you want Oracle Alert to monitor.

Self–Referencing Alert   An alert that uses its own Date Last Checked value during an alert check to report only the exceptions that are new since the last alert check.

SQL Statement Script Action   A SQL statement script that Oracle Alert executes during an alert check or while processing a response.

Summary Action   An action that represents multiple exceptions found in the database.

Summary Message Template   The template you use to format summary message actions. The template resides in the Text field of the Action Details window of the Alerts form. The outputs you place within the template provide a summary of all exception values found for those outputs during an alert check, within a single message. (As opposed to a detail message, where one message is sent for each exception found.)

Summary Threshold   The number of exceptions found during an alert check that determines whether Oracle Alert will perform the detail action in a Threshold Group, or the summary action.

Threshold Group   One detail action and one summary action of the same action type that you want Oracle Alert to choose between when performing an alert check for an alert that uses summary threshold. Oracle Alert performs the detail action if, during an alert check, it finds fewer exceptions than the summary threshold number defined in the Action Sets window of the Alerts form. Oracle Alert performs the summary action if, during an alert check, it finds the number of exceptions to be equal to or more than the summary threshold number.

Valid Response   A response that you define in Oracle Alert and assign actions to. If an actual response from a recipient matches a valid response, Oracle Alert performs the assigned actions.

Index

A

D

E

F

Footers, defining message action footers, 3 – 24
Forms, Applications, 8 – 21
Frequency of alerts
  on demand, 3 – 6
  periodic alerts, 3 – 6

H

Headers, defining message action headers, 3 – 24
History
  action history, feature description, 5 – 3
  and action escalation, 5 – 10
  and duplicate checking, 5 – 5
  exception history, feature description, 5 – 3
  of concurrent request actions, 5 – 16
  of message actions, 5 – 16
  of operating system script actions, 5 – 16
  of SQL statement script actions, 5 – 17
  overview, 5 – 2
  response history, 5 – 17
    feature description, 5 – 3
  reviewing action set checks, 7 – 11
  reviewing actions, 5 – 15 to 5 – 28
  reviewing alert checks, 5 – 21, 7 – 11
  reviewing exceptions, 5 – 19
  saving, 3 – 7, 3 – 12

I

Implicit inputs, 3 – 16
  DATE_LAST_CHECKED, 3 – 16, 4 – 17
  MAILID, 3 – 14, 3 – 16
  ROWID, 3 – 13, 3 – 16
Inbox folder, 6 – 16
Inputs
  action sets, feature description, 3 – 3
  defining default values, 3 – 17
  displaying, 3 – 16
  implicit inputs, 3 – 16
  in event alerts – example, 3 – 14
  in periodic alerts – example, 3 – 8

Installation number, for Oracle Alert, 8 – 10
Installations, multiple Oracle Alert installations, 6 – 19
Invalid response actions, defining, 6 – 10

M

Mail ID validation, in Oracle Office, 3 – 20
MAILID, 3 – 14, 3 – 16
Menus, menu paths for all forms, A – 2
Message actions, 3 – 20
  defining headers and footers, 3 – 24
  detail message actions, 3 – 22
  distribution lists, feature description, 3 – 3
  dynamic distribution, feature description, 3 – 2
  entering message text, 3 – 22
  no exception message actions, 3 – 22
  printing, 3 – 21
  responding to, 6 – 15
  reviewing history, 5 – 16
  summary message actions, 3 – 25
    formatting, 4 – 6
  that solicit responses, 6 – 4
Message handles, 6 – 19
Message text, entering message text, 3 – 22

N

No exception actions
  no exception concurrent program actions, 3 – 26
  no exception message actions, 3 – 22
  no exception operating system script actions, 3 – 28
  no exception SQL statement script actions, 3 – 30
No exception message actions, 3 – 22
No exception operating system script actions, 3 – 28
No exception SQL statement script actions, 3 – 30
No response follow–up actions, 6 – 12
  defining, 6 – 11

feature description, 6 – 3

Reader's Comment Form

Oracle Alert User's Guide
A56103–01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

_____

_____

_____

_____

_____

_____

_____

_____

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (650) 506–7000   Fax: (650) 506–7200

If you would like a reply, please give your name, address, and telephone number below:

_____

_____

_____

Thank you for helping us improve our documentation.