

Oracle® Alert

User's Guide

Release 12.1

Part No. E12951-03

March 2009

Oracle Alert User's Guide, Release 12.1

Part No. E12951-03

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Siu Chang, Clara Jaeckel

Contributing Author: Dave Hung, Nancy Liu, Steve Mayze, Traci Short, Jan Smith, Cathleen Trezza, Markus Tjahjadi, Melody Yang

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Overview of Oracle Alert

Overview of Oracle Alert.....	1-1
Basic Business Needs.....	1-2
Major Features.....	1-3

2 Defining Alerts

Overview of Alerts.....	2-1
Major Features	2-2
Creating a Periodic Alert.....	2-3
Creating an Event Alert.....	2-8
Specifying Alert Details.....	2-13
Creating Alert Actions.....	2-17
Creating an Action Set for an Alert.....	2-32
Deleting an Alert.....	2-36
Transferring Alert Definitions.....	2-36

3 Customizing Alerts

Overview of Alert Customizations.....	3-1
Defining a Distribution List	3-2
Formatting Summary Message Actions	3-4
Overview of Summary Threshold.....	3-10
Defining a Summary Threshold.....	3-11

Creating Self-Referencing Alerts	3-16
Defining a Periodic Set	3-17

4 Using Oracle Alert History: Duplicate Checking and Action Escalation

Overview of Oracle Alert History.....	4-1
Basic Business Needs.....	4-2
Major Features.....	4-2
Duplicate Checking.....	4-4
Checking for Duplicates.....	4-5
Action Escalation.....	4-8
Defining Escalating Actions.....	4-8
Reviewing Alert Actions.....	4-14
Reviewing Action History	4-14
Reviewing Alert Exceptions.....	4-19
Reviewing Exception History	4-19
Reviewing Alert Checks.....	4-21
Reviewing Alert Check Information	4-21

5 Using Response Processing

Overview of Response Processing.....	5-1
Basic Business Needs	5-1
Major Features	5-2
Defining a Response Processing Alert.....	5-3
Responding to an Alert Message	5-14
How Oracle Alert Processes Responses.....	5-15

6 How Oracle Alert Checks Alerts

How Oracle Alert Checks Alerts	6-1
Overview of Oracle Alert Concurrent Programs	6-2
The Periodic Alert Scheduler	6-3
Checking a Periodic Alert	6-5
Checking an Event Alert	6-6
Processing Alert Response Actions.....	6-7
Processing No Response Actions.....	6-7
Reviewing Alert Check History	6-7

7 Setting Up Oracle Alert

Setting Up Oracle Alert.....	7-1
Specifying Oracle Alert Options	7-2

Defining Boilerplate Message Text.....	7-2
Defining Other Options for Oracle Alert.....	7-4
Setting Up the :MAILID Implicit Input.....	7-6
Defining Alerts for Custom Oracle Applications	7-8
Creating a Responsibility for Your Custom Application Alert Manager	7-9
Oracle Alert and Multiple Application Installations	7-10
Oracle Alert Naming Conventions.....	7-14
General Naming Conventions.....	7-15
Alert-Specific Naming Conventions	7-15
Applications Window.....	7-16
Registering Application Installations	7-16

8 Using Precoded Alerts

Oracle Alert Precoded Alerts	8-1
Oracle Alert DBA Alerts	8-2
Oracle Alert Purging Alerts.....	8-5

9 Implementing Oracle Alert

Overview of Setting Up.....	9-1
Related Product Setup Steps.....	9-1
Setup Flowchart.....	9-2
Setup Checklist	9-2
Setup Steps	9-3

A Menu Navigation Appendix

Oracle Alert Menu Paths.....	A-1
------------------------------	-----

Glossary

Index

Send Us Your Comments

Oracle Alert User's Guide, Release 12.1

Part No. E12951-03

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.1 of the *Oracle Alert User's Guide*.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area
- Oracle Alert.

If you have never used Oracle Alert, Oracle suggests you attend one or more of the Oracle Alert training classes available through Oracle University.

- The Oracle Applications graphical user interface.

To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*. See Other Information Sources for more information about Oracle Applications product information.

See Related Information Sources on page xi for more Oracle Applications product information.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

1 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

2 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.

3 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.

4 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.

5 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.

6 How Oracle Alert Checks Alerts

This chapter discusses how Oracle Alert checks for alert exceptions.

7 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.

8 Using Precoded Alerts

This chapter describes the precoded alerts that are included in your Oracle Alert installation.

9 Implementing Oracle Alert

This chapter describes the steps necessary to implement Oracle Alert.

A Menu Navigation Appendix

This appendix lists the forms available in the Oracle Alert Manager responsibility along with their navigation paths.

Glossary

Related Information Sources

This book is included on the Oracle Applications Documentation Library, which is supplied in the Release 12 Media Pack. You can download soft-copy documentation as PDF files from the Oracle Technology Network at <http://otn.oracle.com/documentation>, or you can purchase hard-copy documentation from the Oracle Store at <http://oraclestore.oracle.com>. The Oracle Applications Release 12 Documentation Library contains the latest information, including any documents that have changed significantly between releases. If substantial changes to this book are necessary, a revised version will be made available on the "virtual" documentation library on My Oracle Support (formerly Oracle*MetaLink*).

If this guide refers you to other Oracle Applications documentation, use only the latest Release 12 versions of those guides.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **Online Help** – The new features section in the HTML help describes new features in 12. This information is updated for each new release of Oracle Alert. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-pack or an upgrade, this document describes the new features. Online help patches are available on My Oracle Support.
- **PDF Documentation** - See the Oracle Applications Documentation Library for current PDF documentation for your product with each release. The Oracle Applications Documentation Library is also available on My Oracle Support and is updated frequently.
- **Oracle Electronic Technical Reference Manual** - The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle Applications product. This information helps you convert data from your existing applications and integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. The Oracle eTRM is available on My Oracle Support.

Related User's Guides

Oracle Alert shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other user's guides when you set up and use Oracle Alert.

You can read the guides online by choosing Library from the expandable menu in your HTML help window, by reading from the Oracle Applications Documentation Library included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at <http://oraclestore.oracle.com>.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to navigate, enter data, query, and run reports using the user interface (UI) of Oracle Applications. This guide also includes information on setting user profiles, as well as running and reviewing concurrent requests.

Installation and System Administration

Oracle Applications Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12, or contemplating significant changes to a configuration. After describing the Oracle Applications architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus the installation and configuration choices that may be available.

Oracle Applications Installation Guide: Using Rapid Install

This book is intended for use by anyone who is responsible for installing or upgrading Oracle Applications. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle Applications Release 12, or as part of an upgrade from Release 11i to Release 12. The book also describes the steps needed to install the technology stack components only, for the special situations where this is applicable.

Oracle Applications Upgrade Guide: Release 11i to Release 12

This guide provides information for DBAs and Applications Specialists who are responsible for upgrading a Release 11i Oracle Applications system (techstack and products) to Release 12. In addition to information about applying the upgrade driver, it outlines pre-upgrade steps and post-upgrade steps, and provides descriptions of product-specific functional changes and suggestions for verifying the upgrade and reducing downtime.

Oracle Applications Patching Procedures

This guide describes how to patch the Oracle Applications file system and database using AutoPatch, and how to use other patching-related tools like AD Merge Patch, OAM Patch Wizard, and OAM Registered Flagged Files. Describes patch types and structure, and outlines some of the most commonly used patching procedures. Part of Maintaining Oracle Applications, a 3-book set that also includes *Oracle Applications Maintenance Utilities* and *Oracle Applications Maintenance Procedures*.

Oracle Applications Maintenance Utilities

This guide describes how to run utilities, such as AD Administration and AD Controller, used to maintain the Oracle Applications file system and database. Outlines the actions performed by these utilities, such as monitoring parallel processes,

generating Applications files, and maintaining Applications database entities. Part of Maintaining Oracle Applications, a 3-book set that also includes *Oracle Applications Patching Procedures* and *Oracle Applications Maintenance Procedures*.

Maintaining Oracle Applications Documentation Set

This documentation set provides maintenance and patching information for the Oracle Applications DBA. *Oracle Applications Maintenance Procedures* provides a description of the strategies, related tasks, and troubleshooting activities that will help ensure the continued smooth running of an Oracle Applications system. *Oracle Applications Maintenance Utilities* describes the Oracle Applications utilities that are supplied with Oracle Applications and used to maintain the application file system and database. It also provides a detailed description of the numerous options available to meet specific operational requirements. *Oracle Applications Patching Procedures* explains how to patch an Oracle Applications system, covering the key concepts and strategies. Also included are recommendations for optimizing typical patching operations and reducing downtime.

Oracle Applications System Administrator's Guide Documentation Set

This documentation set provides planning and reference information for the Oracle Applications System Administrator. *Oracle Applications System Administrator's Guide - Configuration* contains information on system configuration steps, including defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help. *Oracle Applications System Administrator's Guide - Maintenance* provides information for frequent tasks such as monitoring your system with Oracle Applications Manager, administering Oracle E-Business Suite Secure Enterprise Search, managing concurrent managers and reports, using diagnostic utilities including logging, managing profile options, and using alerts. *Oracle Applications System Administrator's Guide - Security* describes User Management, data security, function security, auditing, and security configurations.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle Applications. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Other Implementation Documentation

Oracle Applications Multiple Organizations Implementation Guide

This guide describes the multiple organizations concepts in Oracle Applications. It describes in detail on setting up and working effectively with multiple organizations in Oracle Applications.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle Applications implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This guide also provides information on creating custom reports on flexfields data.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Alert and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

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 Database, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle **STRONGLY RECOMMENDS** that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

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 such as 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, 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 to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. 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.

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

This chapter covers the following topics:

- Overview of Oracle Alert

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 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 Integration

Oracle Alert allows you to send alert e-mail messages through your mail system using the Simple Mail Transfer Protocol (SMTP) for outbound messages and the Internet Message Access Protocol (IMAP) for inbound messages.

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.

This chapter covers the following topics:

- Overview of Alerts
- Creating a Periodic Alert
- Creating an Event Alert
- Specifying Alert Details
- Creating Alert Actions
- Creating an Action Set for an Alert
- Deleting an Alert
- Transferring Alert Definitions

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 leverages the Workflow Notification Mailer to send and receive alert e-mail messages. The notification mailer uses the Simple Mail Transfer Protocol (SMTP) for outbound messages and the Internet Message Access Protocol (IMAP) for inbound messages.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Transferring Alert Definitions, page 2-36

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, page 2-13
- Define actions for your alert, page 2-17
- Create action sets containing the actions you want your alert to perform, page 2-32

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:

- Configure the Workflow Notification Mailer to send and receive e-mail messages according to your alert requirements. See: Setup Steps, page 9-3.
- Specify Oracle Alert options to configure how Oracle Alert checks alerts and handles alert messages. See: Specifying Oracle Alert Options, page 7-2.

To define a periodic alert:

The Alerts Window

Application Name

Description ☒ Enabled

Periodic **Event**

Periodic Details

Frequency **On Demand**

Start Time End Time Check Interval

Keep **0** Days End Date Last Checked

Select Statement

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

Import... Export... Verify Run

Actions Action Sets Response Sets Alert Details

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 7-14.

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. Choose the Periodic tab.
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 6-5.

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 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 4-1
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.

Tip: 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*.

Important: The SQL Select statement must be less than 32K.

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.

Important: The file you import must be less than 32K.

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 2-13.

Related Topics

How Oracle Alert Checks Alerts, page 6-1

Checking a Periodic Alert, page 6-5

Transferring Alert Definitions, page 2-36

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

Deleting an Alert, page 2-36

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, page 2-13
- Define actions for your alert, page 2-17
- Create action sets containing the actions you want your alert to perform, page 2-32

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:

- Configure the Workflow Notification Mailer to send and receive e-mail messages according to your alert requirements. See: Setup Steps, page 9-3.
- Specify Oracle Alert options to configure how Oracle Alert checks alerts and handles alert messages. See: Specifying Oracle Alert Options, page 7-2.

To define an event alert:

The Alert Window: Event Tab

Alerts

Application Name

Description ☒ Enabled

Periodic ☒ Event

Event Details

Application Table

☒ After Insert (B) ☒ After Update

Keep Days End Date Last Checked

Select Statement

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

Import... Export... Verify Run

Actions Action Sets Response Sets Alert Details

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 7-14.

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. Choose the Event tab.
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.

Important: You cannot use a view as the event table for your alert.

Important: 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 6-1.

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 4-1.
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.

Important: The SQL Select statement must be less than 32K.

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.

Tip: 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.

Important: 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 values for the implicit inputs :MAILID and :ORG_ID 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 EMAIL_ADDRESS profile option. 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.

Note: To use the EMAIL_ADDRESS profile option value as the :MAILID value, you must create and set the EMAIL_ADDRESS profile option and perform some additional setup steps. If the EMAIL_ADDRESS profile option has not been created, or if its value 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. See: Setting Up the :MAILID Implicit Input, page 7-6.

The value for :ORG_ID is the ID of the operating unit that the user is running against upon triggering the event alert. If your SQL statement selects from a multiorg partitioned view, Oracle Alert automatically sets the correct operating unit context to retrieve the data and you need not use :ORG_ID in the SQL statement. You can,

however, use :ORG_ID in your SQL statement to select from a table that is partitioned by operating units, that is, a table that has an ORG_ID column.

Oracle Alert does not require that you use :MAILID or :ORG_ID in your Select statement.

Important: The inputs :MAILID, :ORG_ID 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.

Important: The file you import must be less than 32K.

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 2-13.

Related Topics

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

Deleting an Alert, page 2-36

Transferring Alert Definitions, page 2-36

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:

The Alert Details Window: Inputs Tab

Input Name	Description	Data Type	Default Values
		Character	

1. With your alert definition displayed in the Alerts form, choose Alert Details. The Alert Details window that appears contains three tabbed regions: Installations, Inputs, and Outputs.
2. In the Inputs tabbed region, Oracle Alert automatically displays the inputs used in your Select statement, unless they are the implicit inputs: :ROWID, :MAILID, :ORG_ID 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.
 - **ORG_ID**-Contains the organization ID that is selected when the alert runs.
 - **DATE_LAST_CHECKED**-Contains the date and time that the alert was most recently checked.
3. You can optionally add a description for each input, but you must specify the data

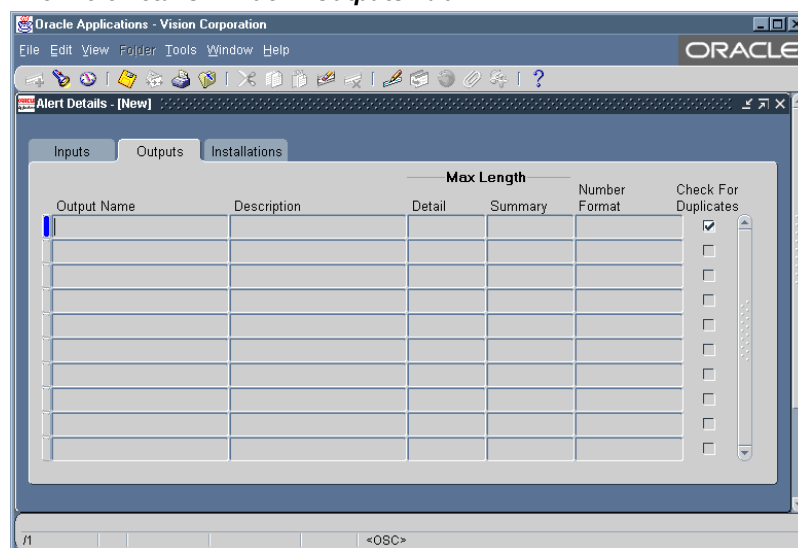
type (either character, number, or date) for the input, because Oracle Alert uses the data type to validate the default values for inputs you enter in the Default Values field and in the Action Set Inputs tabbed region of the Action Sets block.

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.

The Alert Details Window: Outputs Tab



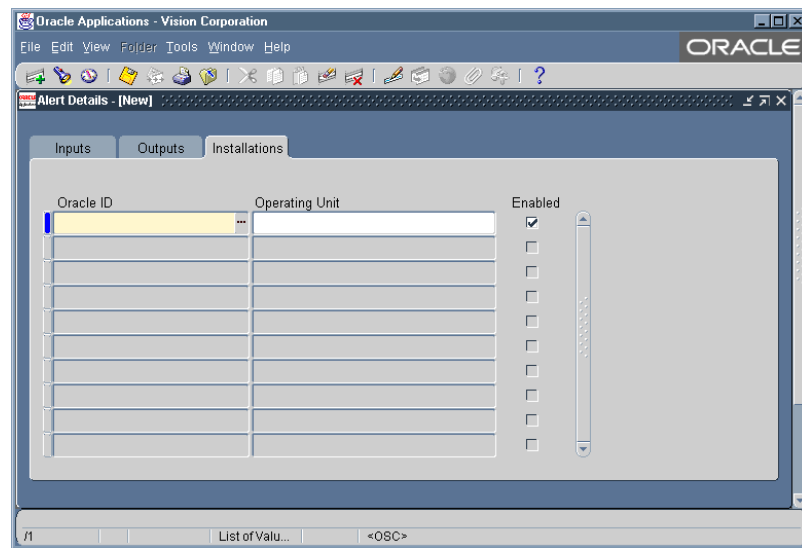
5. In the Outputs tabbed region, Oracle Alert automatically displays the outputs used in your alert Select statement without the ampersand (&) and any numbering format. You can add meaningful descriptions for the outputs.
6. You can specify the maximum number of output characters you want to display in your detail or summary message actions. See: Formatting Summary Message Actions, page 3-4.
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 4-1.

The Alert Details Window: Installation Tab



9. In the Installations tabbed 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. If you have multiple organizations defined in your Oracle Applications schema, you must specify the organization you want the alert to run against in the Operating Unit field.
 - If you do not specify an operating unit for a multiorg schema, then:
 - your event alert will run for the operating unit from which

the event alert is triggered, as specified in the MO:
Operating Unit profile option.

- your periodic alert assumes a non-multiorg schema and simply runs for the Oracle Alert Manager's operating unit.
 - If you want your alert to run against several operating units of the multiorg schema, you must enter them individually in the Installations tabbed region. Delete any records in this region that do not have a valid Oracle ID and operating unit.
11. Check the Enabled field if you want your alert to run against the specified Oracle ID.
 12. 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. .

Related Topics

[Creating an Action Set for an Alert, page 2-32](#)

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.

The Actions Window

Action Name	Description	Action Level
		Detail

Action Details

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 7-14.
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.

Define Message Actions

The screenshot shows the 'Define Message Actions' dialog box. The 'Action Type' is set to 'Message'. The 'To' field is highlighted in yellow. The 'List' field has a dropdown arrow. The 'Reply To' field is empty. The 'Subject', 'Cc', and 'Bcc' fields are empty. The 'Print For User' field is empty. The 'Printer' field is empty. The 'Response Set' field is empty. The 'Response Days' field is empty. The 'File (C)' radio button is selected. The 'Text (D)' radio button is unselected. The 'Column Overflow' is set to 'Wrap'. The 'Max Width' field is empty. The 'Import...' button is at the bottom right.

2. Specify the electronic mail IDs 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.
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 3-2.

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.
4. In the Reply To field, you can optionally specify the address that should receive the reply to the alert message. This address must be either the reply-to address for the Workflow Notification Mailer, the e-mail address for a human in charge of response processing, or the address for a custom processing agent external to the notification mailer. If you leave this field blank, the Workflow Notification Mailer uses the value of its own Reply-to Address configuration parameter instead.
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 5-3.

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 also click on the Import button and import the contents of a file into 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.

Important: 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 32K.

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 2-32.

To complete the action details for a summary message:

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

Complete the Action Details for a Summary Message

The screenshot shows the 'Action Details - [New]' window. The 'Action Type' dropdown is set to 'Message'. The 'List' field is empty. The 'To' field is highlighted in yellow. The 'Subject', 'Cc', and 'Bcc' fields are empty. The 'Print For User' field is empty, and the 'Printer' field is set to 'Printer'. The 'Response Set' field is empty, and the 'Response Days' field is set to 'Response Days'. The 'File (Q)' and 'Text (D)' radio buttons are shown, with 'Text (D)' selected. The main text area contains the following template:

```

=== Enter summary template below this line ===
==
=== Enter summary template above this line ===

```

At the bottom, the 'Column Overflow' dropdown is set to 'Wrap', and the 'Max Width' field is set to '80'. An 'Import...' button is located in the bottom right corner.

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.

Tip: 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 tabbed 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 3-4.

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 7-2.

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 Wrap, Oracle Alert uses as many lines as necessary to display the output value. 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 3-4.

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 tabbed 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.

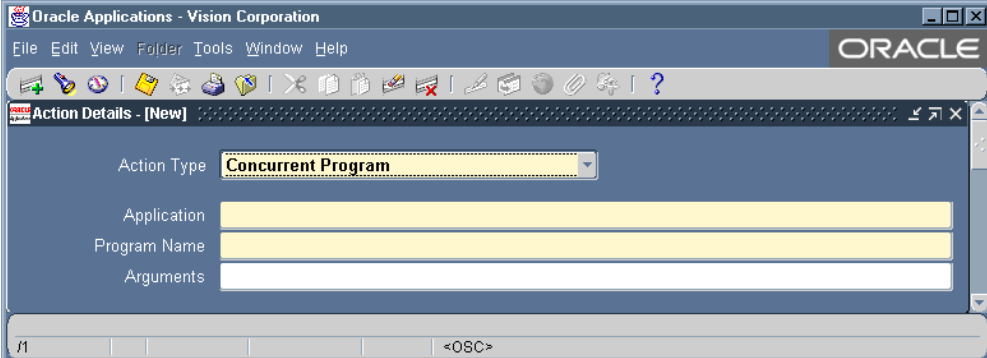
Important: For a summary message actions, the message text that you enter in Oracle alert must be less than 2000 characters, while the message file that you specify must be less than 32K.

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 2-32.

To define concurrent program actions:

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

Define Concurrent Program Actions



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 double 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 2-32.

To define operating system script actions:

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

Define Operating System Script Actions

The screenshot shows a window titled "Action Details - [New]" with a blue header. The "Action Type" dropdown is set to "Operating System Script". The "Application" and "Arguments" fields are empty. The "File" radio button is selected, and the "Text" radio button is unselected. A large text area is below the radio buttons. An "Import..." button is at the bottom right.

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 32K.

6. If you want to enter the script in Oracle Alert, check Text and type the script in the Text field. (You can also click on the Import button and import the contents of a file into 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 system script actions, Oracle Alert performs the operating system script once for each unique combination of critical output values returned by the alert Select statement. 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.

Important: 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 2-32.

To define SQL statement script actions:

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

Define SQL Statement Script Actions

The screenshot shows the 'Action Details - [New]' window in Oracle Applications. The 'Action Type' dropdown is set to 'SQL Statement Script'. Below this, there are three input fields: 'Application', 'Arguments', and a large text area for the script. To the left of the text area are two radio buttons: 'File (A)' (selected) and 'Text (B)'. At the bottom right of the window is an 'Import...' button. The window title bar includes 'Oracle Applications - Vision Corporation' and the Oracle logo.

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.

Important: The SQL script file must be less than 32K.

6. If you want to enter the script in Oracle Alert, check Text and type the SQL script in the Text field. (You can also click on the Import button and import the contents of a file into 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.

Important: 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 3-4.
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 2-32.

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.

Delete an Action

References - Detail msg AS2

Action Groups Action Sets Response Sets

This action is an active member of the following action groups:

Name	Type
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

You can also choose References from the Tools 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 the Action Groups, Action Sets, or Response Sets tab 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.

Related Topics

Defining a Response Processing Alert, page 5-3

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.

The Action Sets Window

Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	

Action Set Details

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 7-14.

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 4-1.

Note: Use the Check for Duplicates check box in the Outputs tabbed 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, choose Action Set Details. In the Action Set Details window, display the Members tabbed region.

The Action Set Details Window - Members Tab

Seq	Action	Type	Summary Threshold	On Error Action	Seq	Enabled	End Date
1				Abort		<input checked="" type="checkbox"/>	
						<input type="checkbox"/>	
						<input type="checkbox"/>	
						<input type="checkbox"/>	
						<input type="checkbox"/>	
						<input type="checkbox"/>	

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.

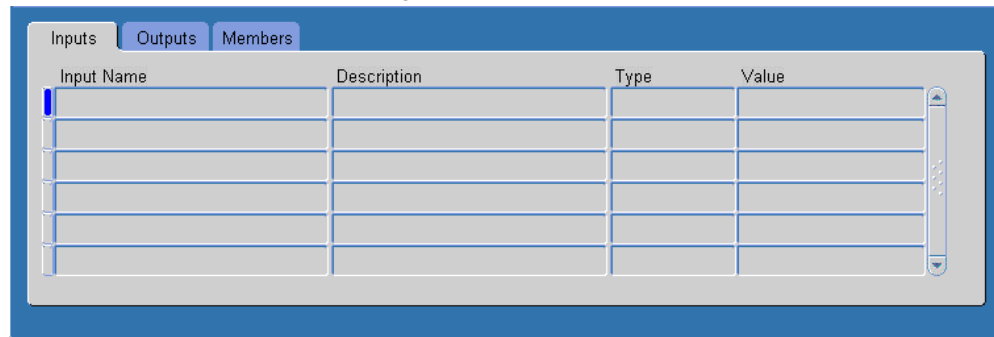
11. You can also assign additional actions for Oracle Alert to take if a particular action in an action set fails.

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 tabbed region.

The Action Set Details Window - Inputs Tab



The screenshot shows a window titled 'The Action Set Details Window - Inputs Tab'. It has three tabs: 'Inputs' (selected), 'Outputs', and 'Members'. Below the tabs is a table with four columns: 'Input Name', 'Description', 'Type', and 'Value'. The table is empty and has a vertical scrollbar on the right side.

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 tabbed 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 tabbed region.

The Action Set Details Window - Outputs Tab

The screenshot shows a window titled 'The Action Set Details Window - Outputs Tab'. It has three tabs: 'Inputs', 'Outputs', and 'Members'. The 'Outputs' tab is active. It contains a table with two columns: 'Output Name' and 'Description'. There are six rows in the table. To the right of the table is a column labeled 'Check For Duplicates' with checkboxes. The first checkbox is checked, and the others are unchecked. There is also a vertical scrollbar on the right side of the table.

Output Name	Description	Check For Duplicates
		<input checked="" type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

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 tabbed 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.

Related Topics

Overview of Oracle Alert History, page 4-1

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.

Related Topics

Overview of Oracle Alert History, page 4-1

Defining a Response Processing Alert, page 5-3

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 Tools menu of the Alerts form, Transfer Alert.

The Transfer Alert Definitions Window

The screenshot shows a window titled "Transfer Alert Definitions". It has a blue background. At the top, there's a title bar with the text "Transfer Alert Definitions" and standard window controls (minimize, maximize, close). Below the title bar, there are two main sections: "Source Alert" and "Destination Alert". Each section contains three input fields: "Application" (yellow), "Alert" (yellow), and "Database" (white). A "Transfer" button is located at the bottom right of the window.

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.

Related Topics

[Oracle Alert Naming Conventions, page 7-14](#)

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.

This chapter covers the following topics:

- Overview of Alert Customizations
- Defining a Distribution List
- Formatting Summary Message Actions
- Overview of Summary Threshold
- Defining a Summary Threshold
- Creating Self-Referencing Alerts
- Defining a Periodic Set

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, page 3-2 to simplify the maintenance of your alert message recipients
- improve the format of your summary messages, page 3-4 so that they are easier to read
- set a summary threshold, page 3-10 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, page 3-16 to tell you what exceptions are unique since the alert was last checked

- create a periodic set, page 3-17 so that you can check several periodic alerts at once

You can also implement other customizations, such as duplicate checking, page 4-4 and action escalation, page 4-8, by saving alert history. See: Overview of Oracle Alert History, page 4-1.

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.

The Distribution Lists Window

The screenshot shows a web-based form titled "Distribution Lists". The form is organized into several sections. At the top, there are three input fields: "Application" (with a dropdown arrow), "Name", and "Description". Below these is a section titled "Mail Recipients" which contains three input fields: "To", "Cc", and "Bcc". The "To" field is highlighted in yellow. At the bottom, there is a section titled "Print Options" which contains two input fields: "For User" and "Printer". The "Printer" field is highlighted in grey.

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 7-14.
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.

Tip: 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.

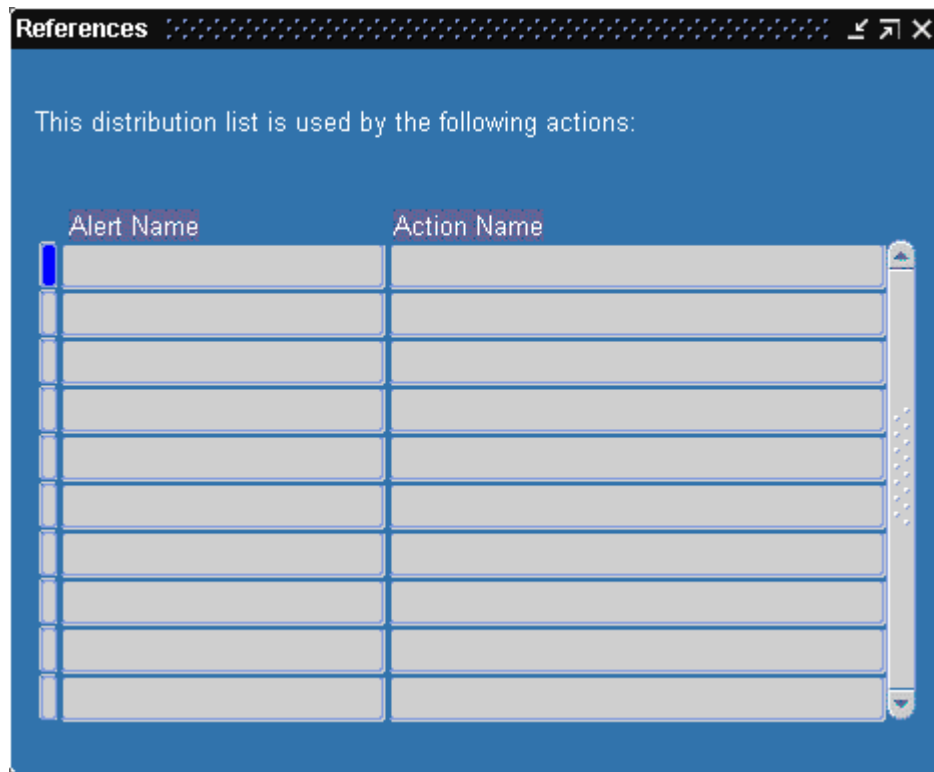
6. In the Print Options region, enter in the For User field, the names of the recipients for whom you want to print a copy of a message. Separate the names with commas and do not use blank spaces within a single name. You can enter as many names as you like, up to 240 characters.
7. Enter in the Printer 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 Tools menu in the Distribution Lists window at any time to display the References window.

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 tabbed 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:

The Alerts Window: Periodic Tab

Alerts

Application Name

Description ☒ Enabled

Periodic Event

Periodic Details

Frequency

Start Time End Time Check Interval

Keep Days End Date Last Checked

Select Statement

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

Import... Export... Verify Run

Actions Action Sets Response Sets Alert Details

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

The Alert Details Window

Output Name	Description	Max Length		Number Format	Check For Duplicates
		Detail	Summary		
					<input checked="" type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

3. In the Outputs tabbed 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.

The Action Details 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 Wrap or Truncate 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.

If you choose Wrap, Oracle Alert uses as many lines as necessary to display the output value. 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.

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.

If you choose column wrap, Oracle Alert appends a hyphen to the end of each line that ends in the middle of a word, to indicate that the word continues on the next line. If a line ends on a space or on the last letter of a word before a space, Oracle Alert does not add a hyphen, but simply wraps the remaining words onto the next line. For example, an output value might appear in a column as follows:

```
Mai-  
  ling  
  list  
  for  
  ale-  
  rts
```

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 22 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 and appends a hyphen to the end of a line when the line break occurs in the middle of a word. 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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

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.

Related Topics

Defining a Summary Threshold, page 3-11

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 2-1.
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 Tools menu.

The Threshold Groups Window

Group Name	Description	Type
		Message

Threshold Group Members			
Action	Level	Enabled	End Date
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

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 7-14.
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.

The Action Sets Window

Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2			<input type="checkbox"/>	<input type="checkbox"/>	
3			<input type="checkbox"/>	<input type="checkbox"/>	
4			<input type="checkbox"/>	<input type="checkbox"/>	
5			<input type="checkbox"/>	<input type="checkbox"/>	

Members							
Seq	Action	Type	Summary Threshold	On Error	Seq	Enabled	End Date
1				Abort		<input checked="" type="checkbox"/>	
2						<input type="checkbox"/>	
3						<input type="checkbox"/>	
4						<input type="checkbox"/>	
5						<input type="checkbox"/>	

11. In the Action field of the Members tabbed 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.

Important: 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

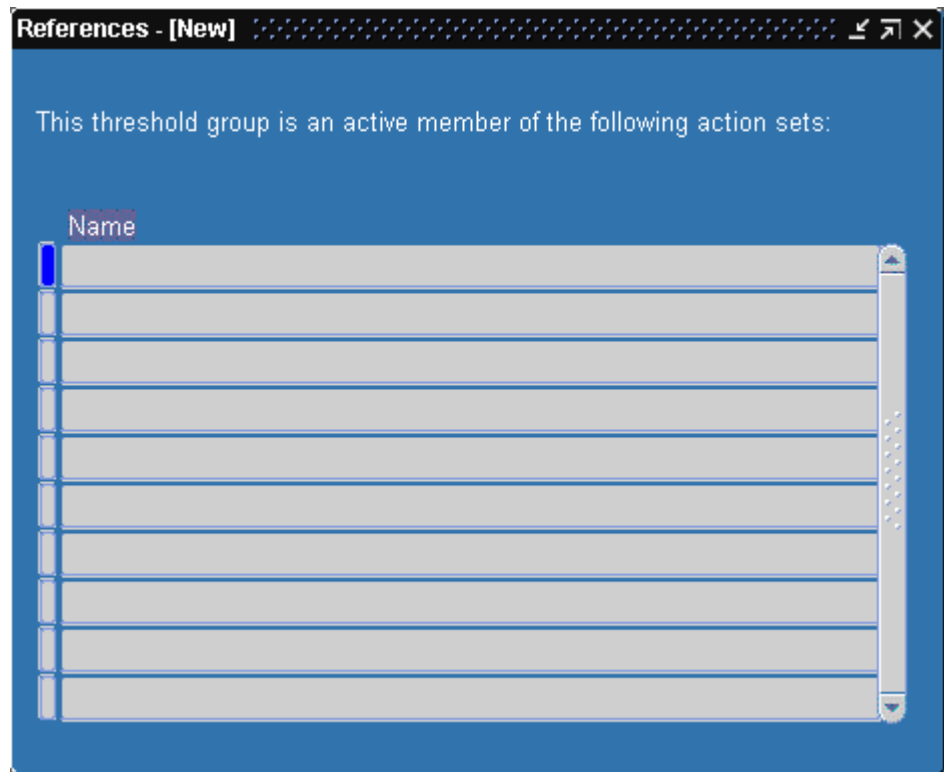
Oracle Alert groups exceptions before it determines whether to perform a detail or a summary action. That is, if you use an output to represent your message action's distribution, Oracle Alert first substitutes the appropriate exception value for the output, then based on the number of exceptions substituted, 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.

The References Window



You can also choose References from the Tools 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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

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. The value of the implicit input :DATE_LAST_CHECKED is character. 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 6-1.

Oracle Alert never displays :DATE_LAST_CHECKED in the Inputs tabbed 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 creation_date > 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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

Overview of Oracle Alert History, page 4-1

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 2-3.

To define a periodic set:

1. Navigate to the Periodic Sets form.

The Periodic Sets Window

The screenshot shows the 'Periodic Sets' window. At the top, there are three input fields: 'Application' (with a dropdown arrow), 'Name', and 'Description'. Below these fields is a section titled 'Periodic Set Members' which contains a table. The table has six columns: 'Seq', 'Application', 'Name', 'Type', 'Enabled', and 'End Date'. The first row of the table is highlighted in blue, and the 'Enabled' checkbox in this row is checked. There are 10 rows in total, with the first row being the header and the remaining 9 rows being data rows. A vertical scrollbar is visible on the right side of the table.

Seq	Application	Name	Type	Enabled	End Date
1				<input checked="" type="checkbox"/>	
2				<input type="checkbox"/>	
3				<input type="checkbox"/>	
4				<input type="checkbox"/>	
5				<input type="checkbox"/>	
6				<input type="checkbox"/>	
7				<input type="checkbox"/>	
8				<input type="checkbox"/>	
9				<input type="checkbox"/>	
10				<input type="checkbox"/>	

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 7-14.

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 than 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 you 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.

Related Topics

Checking a Periodic Alert, page 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.

This chapter covers the following topics:

- Overview of Oracle Alert History
- Duplicate Checking
- Checking for Duplicates
- Action Escalation
- Defining Escalating Actions
- Reviewing Alert Actions
- Reviewing Action History
- Reviewing Alert Exceptions
- Reviewing Exception History
- Reviewing Alert Checks
- Reviewing Alert Check Information

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 detail 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 for an escalation, 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.

Note: Because you can only use detail actions for action escalation, Oracle Alert can only suppress detail actions. However, if you choose to suppress duplicates for an action set, Oracle Alert also flags duplicate exceptions in summary message actions.

Related Topics

Duplicate Checking, page 4-4

Action Escalation, page 4-8

Reviewing Alert Actions, page 4-14

Reviewing Alert Exceptions, page 4-19

Reviewing Alert Checks, page 4-21

Creating Self-Referencing Alerts, page 3-16

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:

The Alerts Window

The screenshot shows the 'Alerts' configuration window. At the top, there are fields for 'Application', 'Name', and 'Description'. Below these is a 'Periodic' tab and an 'Event' tab. The 'Periodic Details' section includes a 'Frequency' dropdown set to 'On Demand', 'Start Time', 'End Time', and 'Check Interval' fields. Below this is a 'Keep' field set to '0' and a 'Days' label. To the right is an 'End Date' field and a 'Last Checked' field. A large text area labeled 'Select Statement' contains the SQL query: `SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME`. To the right of this text area are buttons for 'Import...', 'Export...', 'Verify', and 'Run'. At the bottom of the window are four buttons: 'Actions', 'Action Sets', 'Response Sets', and 'Alert Details'.

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.

The Alert Details Window

Output Name	Description	Detail	Summary	Max Length	Number Format	Check For Duplicates
						<input checked="" type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>

2. Use the Outputs tabbed 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.

The Action Sets Window

Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="checkbox"/>	<input type="checkbox"/>	

Output Name	Description	Check For Duplicates
		<input checked="" type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

2. You can customize the duplicate configuration of each action set by checking the Check for Duplicates field for each output in the Outputs tabbed 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 4-8.
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.

Note: Because you can only use detail actions for action escalation, Oracle Alert can only suppress detail actions. However, if you choose to suppress duplicates for an action set, Oracle Alert also flags duplicate exceptions in summary message actions.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

Action Escalation, page 4-8

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.

The Alerts Window

The screenshot shows the 'Alerts' window with the following fields and controls:

- Application**: Text input field.
- Description**: Text input field.
- Name**: Text input field.
- Enabled**: Checkmark button.
- Periodic / Event**: Tabbed interface with 'Event' selected.
- Event Details**:
 - Application**: Text input field.
 - Table**: Text input field.
 - After Insert (B)**: Checkmark button.
 - After Update**: Checkmark button.
- Keep**: Text input field with value '0'.
- Days**: Text input field.
- End Date**: Text input field.
- Last Checked**: Text input field.
- Select Statement**: Large text area containing the SQL query: `SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME`.
- Buttons**: 'Import...', 'Export...', 'Verify', 'Run'.
- Footer Buttons**: 'Actions', 'Action Sets', 'Response Sets', 'Alert Details'.

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.

The Alert Details Window

Output Name	Description	Detail	Summary	Max Length	Number Format	Check For Duplicates
						<input checked="" type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>

4. Display the Outputs tabbed 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 Tools menu to display the Escalation Groups window.

The Escalation Groups Window

Group Name	Description	Type
		Message

Escalation Group Members			
Action Name	Level	Enabled	End Date
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	

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 7-14.

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.

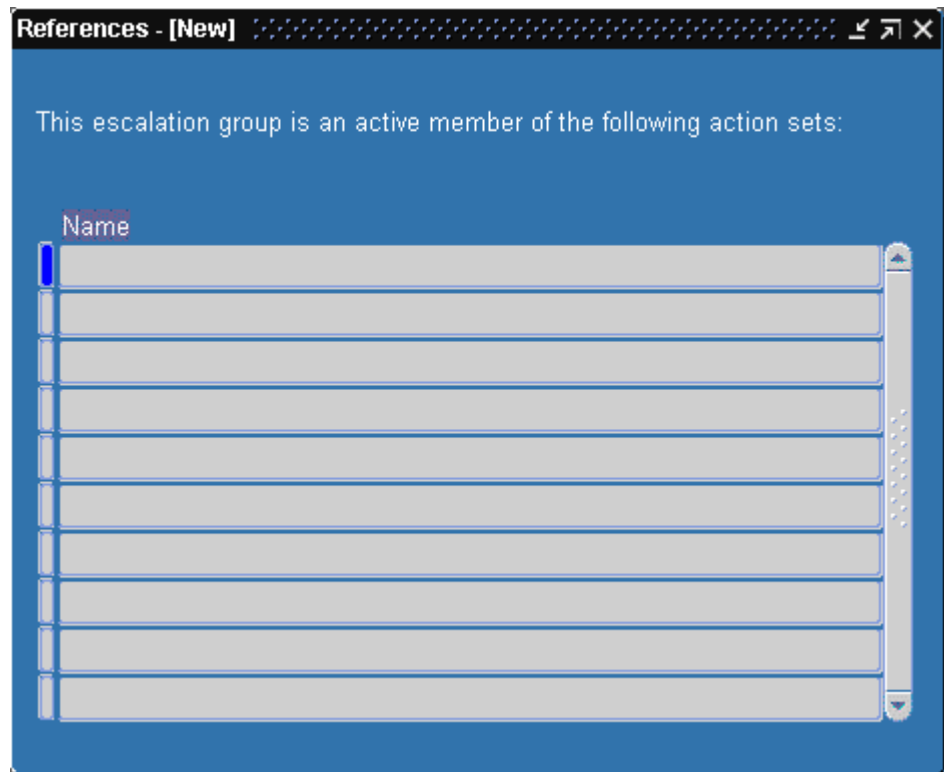
Note: Each action set duplicate configuration takes precedence over the alert duplicate configuration during an alert check.

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 Tools menu in the Escalation Groups window at any time to display the References window.

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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

Checking for Duplicates, page 4-5

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:

The Review Alert History Window

The screenshot shows a window titled "Review Alert History". It contains several input fields and buttons. At the top, there is a "Application" field with a dropdown arrow. Below it is an "Alert Name" field. To the right of the "Alert Name" field are three smaller fields: "Type", "Keep", and "Days". Below these is a "Last Checked" field. A section titled "View History Dates" contains "From" and "To" date input fields. At the bottom, there are three buttons: "Find Checks", "Find Exceptions...", and "Find Actions".

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.

The Review Alert Actions Window

The screenshot shows a window titled "Review Alert Actions - Purge Concurrent Requests". It has a tab labeled "Request". Below the tab is a table with the following columns: "Action Set Name", "Date", "Time", and "Oracle ID". The table contains five rows of data. Below the table is a section titled "Alert Actions" which contains a table with the following columns: "Action Name", "Type", and "Time Finished". This table also contains five rows of data. To the right of the "Alert Actions" table is a "Details" panel with the following fields: "To", "Subject", "Cc", "Bcc", and "File Name". At the bottom of the window are two buttons: "Response" and "Action Log".

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.

The Response History Review Window

Oracle Applications - Vision Corporation

File Edit View Folder Tools Window Help

Response History Review - Purge Concurrent Requests, Response Set, Action

Date Processed	Response	From

Response Actions

Response Action	Action Type	Arguments	Oracle ID	File

Reply Log Action Log

FRM-40350: Query caused no records to be retrieved.

<OSC>

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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Checking a Periodic Alert, page 6-5

Defining a Response Processing Alert, page 5-3

Reviewing Alert Exceptions, page 4-19

Reviewing Alert Checks, page 4-21

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.

The Display Output Variables Window

Display Output Variables - Purge Alert and Action Set Checks

Output variables to display:

Output Variable Name	Enabled
ACTION_SET_CHECK_ID	<input checked="" type="checkbox"/>
ALERT	<input checked="" type="checkbox"/>
ALERT_CHECK_ID	<input checked="" type="checkbox"/>
ALERT_ID	<input checked="" type="checkbox"/>
APPLICATION	<input checked="" type="checkbox"/>
APPLICATION_ID	<input checked="" type="checkbox"/>
AS_CHECK	<input checked="" type="checkbox"/>
A_CHECK	<input checked="" type="checkbox"/>

OK Cancel

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.

The Review Alert Exceptions Window

The screenshot shows a window titled "Review Alert Exceptions - Purge Alert and Action Set Checks". It has a "Request" tab. Below the tab is a table with four columns: "Action Set Name", "Date", "Time", and "Oracle ID". The first row contains the text "Purge old checks", "25/AUG/1997", "14:00:02", and "APPS". Below this table is a section titled "Alert Exceptions" which contains a table with three columns: "Seq", "Variable Name", and "Value". The "Seq" column has five rows with values 1 through 5. The "Variable Name" and "Value" columns are empty for all rows.

Action Set Name	Date	Time	Oracle ID
Purge old checks	25/AUG/1997	14:00:02	APPS

Seq	Variable Name	Value
1		
2		
3		
4		
5		

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.

Related Topics

- Creating a Periodic Alert, page 2-3
- Creating an Event Alert, page 2-8
- Checking a Periodic Alert, page 6-5
- Reviewing Alert Actions, page 4-14
- Reviewing Alert Checks, page 4-21

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.

The Review Alert Checks Window

Request					
Requestor	Date	Time	Request ID	Status	History Purge Date
SYSADMIN	25/AUG/1997	14:00:02	192383	Error	

Action Set Checks			Number of	
Action Set Name	Oracle ID	Status	Exceptions	Actions
Purge old checks	APPS	Error	1269	2073

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 Tools 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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Checking a Periodic Alert, page 6-5

Reviewing Alert Actions, page 4-14

Reviewing Alert Exceptions, page 4-19

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.

This chapter covers the following topics:

- Overview of Response Processing
- Defining a Response Processing Alert
- Responding to an Alert Message
- How Oracle Alert Processes Responses

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: To enable response processing, ensure that you have performed the required setup steps. See: Setup Steps, page 9-3.

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.

Related Topics

Defining a Response Processing Alert, page 5-3

Defining a Response Processing Alert

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

1. Define an alert, page 5-3
2. Create an alert message action that solicits a response, page 5-3
3. Create response actions, page 5-4
4. Define an action set, page 5-5
5. Create a response set, page 5-5
6. Define possible responses, page 5-7
7. Attach the response set to the alert message action, page 5-11

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 2-1.

Tip: 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 5-19.

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 2-17.

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

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 tabbed 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 2-17.

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.

Tip: 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 2-32.
3. Save your work.

To create a response set for a response processing alert:

1. Choose Response Sets in the Alerts window.

The Response Sets Window

Response Set Name	Description	Enabled	End Date
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	

Response Actions

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 7-14.
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.

The Response Variables Window

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).

Important: 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:

The Valid Responses Window

Responses	
Response Name	Text

Actions					
Seq	Action	Description	Type	Enabled	End Date
				<input checked="" type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	

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.

Important: 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.

The Invalid Response Actions Window

Seq	Action	Description	Type	Enabled	End Date
				<input checked="" type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	
				<input type="checkbox"/>	

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.

Note: You must run the No Response Action Processor concurrent program to check for alert messages whose specified response days have passed without receiving a valid response. See: Processing No Response Actions, page 6-7.

The No Response Actions Window

Seq	Action	Description	Type	Enabled	End Date
1				<input checked="" type="checkbox"/>	
2				<input type="checkbox"/>	
3				<input type="checkbox"/>	
4				<input type="checkbox"/>	
5				<input type="checkbox"/>	
6				<input type="checkbox"/>	
7				<input type="checkbox"/>	
8				<input type="checkbox"/>	
9				<input type="checkbox"/>	
10				<input type="checkbox"/>	

18. The No Response Actions window is similar to the Invalid Response Actions window. Follow the same steps to define a no response follow-up action, page 5-10.

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.

The Action Details Window

Oracle Applications - Vision Corporation

File Edit View Folder Tools Window Help

ORACLE

Action Details - [New]

Action Type: Message

List: [Field] Reply To: [Field]

To: [Field]

Subject: [Field]

Cc: [Field]

Bcc: [Field]

Print For User: [Field] Printer: [Field]

Response Set: [Field] Response Days: [Field]

☒ File (C) [Field]

☐ Text (D) [Field]

Column Overflow: Wrap Max Width: [Field]

Import...

11 List of Valu... <OSC>

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.

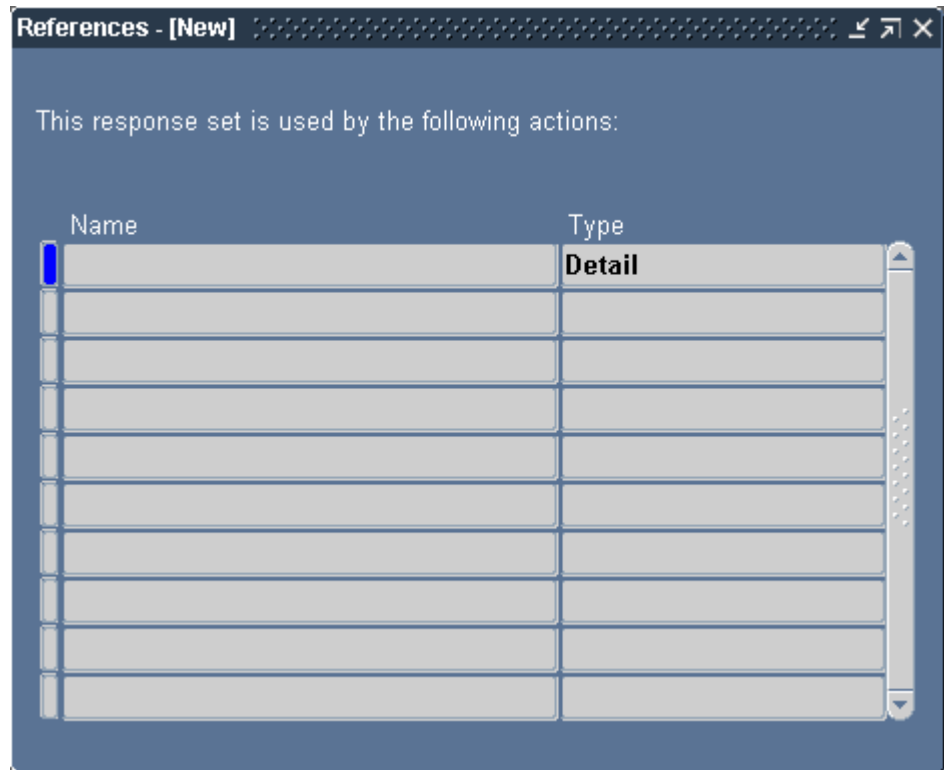
4. In the Response Days field, specify the number of days that you want Oracle Alert to wait before it performs the no response follow-up actions defined in your response set for each outstanding exception.

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 the No Response Action Processor runs 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.

The References Window

You can also choose References from the Tools 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.

Related Topics

Creating a Periodic Alert, page 2-3

Creating an Event Alert, page 2-8

Creating Alert Actions, page 2-17

Creating an Action Set for an Alert, page 2-32

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 5-18.

To respond to an alert message:

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

Important: You must use Reply-with-Original or Forward to include the original message and message handle with your response, for Oracle Alert to locate the message handle and interpret the response.

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.

Important: 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.

How Oracle Alert Processes Responses

Oracle Alert leverages the Workflow Notification Mailer to send alert e-mail messages and process responses.

Processing Responses with the Workflow Notification Mailer

The Workflow Notification Mailer is a Java program provided by Oracle Workflow that performs e-mail send and response processing using the JavaMail API. Oracle Alert leverages the Workflow Notification Mailer to send alert e-mail messages and process responses.

Before implementing Oracle Alert response processing, check that the Workflow Notification Mailer is set up and that its configuration meets your alert requirements. In particular, ensure that the Workflow Notification Mailer has inbound processing enabled and has an e-mail account for receiving messages on an IMAP4 compliant mail server. You may also want to check that the From and Reply-to Address mailer configuration parameters are set to appropriate values for alert messages. See: *Setting Up a Notification Mailer, Oracle Workflow Administrator's Guide*.

The Workflow Notification Mailer uses three folders in its e-mail account for response processing: one to receive incoming messages, one to store processed messages, and one to store discarded messages. When the Workflow Notification Mailer logs into its IMAP e-mail account, it checks the inbox folder for messages. If a message exists, the notification mailer reads the message, checking for an identifying string called the notification ID (NID) line. If the NID line contains an Oracle Alert message handle, the Workflow Notification Mailer identifies the e-mail as an alert response message. See: *Message Handles*, page 5-18.

Important: An alert recipient must use Reply-with-Original or Forward to include the original message with her or his response so that Oracle Workflow and Oracle Alert can locate the message handle and interpret

the response.

The Workflow Notification Mailer also verifies that From address of the response message matches the e-mail address for the recipient of the original alert. Otherwise, the notification mailer moves the message to the discard folder and treats it as an unsolicited message. In some cases the notification mailer sends a warning back to the sender of the response message. For more information about how the notification mailer handles unsolicited messages, see: Inbound Notification Mailer Processing, *Oracle Workflow Administrator's Guide*.

The Workflow Notification Mailer uses a Java-based e-mail parser to interpret the text of the message and create an XML representation of it. Then the Workflow Notification Mailer places the XML message, in the form of a business event called `oracle.apps.alr.response.receive`, on an Oracle Workflow Business Event System agent called `WF_NOTIFICATION_IN`. Finally, the notification mailer moves the e-mail message to its processed folder.

Note: If the parser finds a string of text in the message that matches a tag pattern specified in the mailer configuration parameters, the Workflow Notification Mailer immediately moves the message to the discard folder instead. You can use tags to identify unusual messages such as bounced, returned, or auto-reply messages in order to skip unnecessary invalid response processing and avoid a perpetual loop of invalid warning messages and bounced or auto-reply messages. See: *Tags, Oracle Workflow Administrator's Guide*.

Oracle Workflow provides an agent listener named Workflow Inbound Notifications Agent Listener that runs on the `WF_NOTIFICATION_IN` agent to continue processing for messages placed on that agent. When the Workflow Inbound Notifications Agent Listener dequeues an `oracle.apps.alr.response.receive` event message from `WF_NOTIFICATION_IN`, it executes a seeded Oracle Alert subscription that processes the response.

Oracle Alert reads the XML response message and tries to match the first word or the first 30 continuous characters of the response to any of the defined valid responses. After determining whether the response is valid or invalid, Oracle Alert submits the Alert Action Processor concurrent program to perform the appropriate actions. See: *Processing Alert Response Actions*, page 6-7.

If the actual response matches a valid response, then the Alert Action Processor performs the actions defined for that valid response. After the Alert Action Processor performs the actions for that valid response, Oracle Alert closes the response so that no other responses are processed for that message.

If the actual response does not match any of the valid responses in that response set, the Alert Action Processor performs the actions associated with the invalid response. The Alert Action Processor performs the invalid response actions each time Oracle Alert

receives an invalid response within the response days specified in the original message action.

You must run the No Response Action Processor concurrent program to check for alert messages whose specified response days have passed without receiving a valid response. After the specified response days have passed, if Oracle Alert never received any response to the alert message or if it received only invalid responses, the No Response Action Processor performs the no response follow-up actions for each message and closes responses for all messages in that response set. See: Processing No Response Actions, page 6-7.

Oracle Alert Response Processing Receive Event

The Workflow Notification Mailer uses this event to handle an incoming alert response message.

Internal Name	oracle.apps.alr.response.receive
Status	Enabled
Owner Name	Oracle Alert
Owner Tag	ALR
Customization Level	Limit

Oracle Alert provides one default subscription to the oracle.apps.alr.response.receive event. When this event is received, this subscription processes the response and performs the appropriate actions as specified in the alert definition. The subscription is enabled by default. The following table lists the properties defined for this subscription.

Subscription Properties

Subscription Property	Value
System	<local system>
Source Type	External
Event Filter	oracle.apps.alr.response.receive
Phase	50
Status	Enabled
Rule Data	Key

Subscription Property	Value
Action Type	Custom
On Error	Stop and Rollback
PL/SQL Rule Function	alr_mlr_package.response
Priority	Normal
Owner Name	Oracle Alert
Owner Tag	ALR
Customization Level	Limit

Message Handles

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.

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:

```
<Oracle_Alert_installation_number>.<message_number>.<Oracle_ID>.  
<organization_ID>
```

For example:

```
2.347.3.5
```

The message handle is visible to the message recipient as a number at the end of the message.

Note: The Oracle Alert installation number, which you can review in the Response Processing tabbed region of the Oracle Alert Options form, is set to 1 by default.

In an alert message sent by the Workflow Notification Mailer, the message handle is included within an identifying string called the notification ID (NID) line. The NID line uses the following format:

```
NID[<message_handle>/0000@<mailer_node_name>]
```


where *<message_handle>* is the Oracle Alert message handle and
<mailer_node_name> is the mailer node name used to identify the notification mailer
that should process the message.

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.

Related Topics

Overview of Oracle Alert History, page 4-1

Reviewing Alert Exceptions, page 4-19

Reviewing Alert Actions, page 4-14

Defining a Response Processing Alert, page 5-3

How Oracle Alert Checks Alerts, page 6-1

How Oracle Alert Checks Alerts

This chapter discusses how Oracle Alert checks for alert exceptions.

This chapter covers the following topics:

- How Oracle Alert Checks Alerts
- Overview of Oracle Alert Concurrent Programs
- The Periodic Alert Scheduler
- Checking a Periodic Alert
- Checking an Event Alert
- Processing Alert Response Actions
- Processing No Response Actions
- Reviewing Alert Check History

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, page 6-3
- Check Periodic Alert, page 6-5
- Check Event Alert, page 6-6
- Alert Action Processor, page 6-7
- No Response Action Processor, page 6-7

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.

Related Topics

Reviewing Alert Check History, page 6-7

Overview of Concurrent Processing, *Oracle Applications System Administrator's Guide - Maintenance*

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
- **Alert Action Processor** - Performs response actions when Oracle Alert receives an e-mail response to an alert message
- **No Response Action Processor** - Checks for alert messages whose specified response days have passed without receiving a valid response and performs the no response follow-up 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 tabbed 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:

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: *Monitoring Requests, 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.

Related Topics

Creating a Periodic Alert, page 2-3

Action Escalation, page 4-8

Defining a Summary Threshold, page 3-11

Checking a Periodic Alert

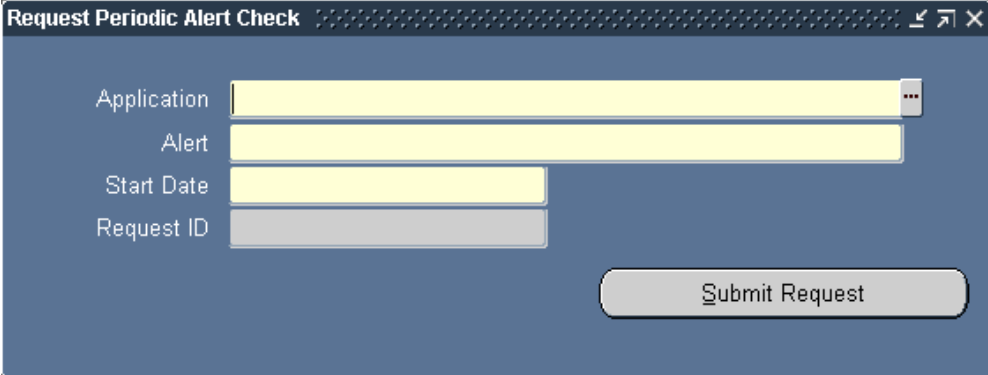
When you want to check an on demand periodic alert or any other periodic alert or periodic set, you use the Request Periodic Alert Check form to submit a request to the concurrent manager to run a concurrent program called Check Periodic Alert (ALECDC). The concurrent manager runs this request according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

The Request Periodic Alert Check Window



The screenshot shows a web-based form titled "Request Periodic Alert Check". The form is set against a blue background. It includes the following elements:

- Application:** A dropdown menu with a yellow background and a small "..." icon on the right.
- Alert:** A text input field with a yellow background.
- Start Date:** A date and time selection field with a yellow background.
- Request ID:** A text input field with a light gray background.
- Submit Request:** A rounded rectangular button with a light gray background and the text "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 Requests window.

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

Related Topics

Creating a Periodic Alert, page 2-3

Defining a Periodic Set, page 3-17

Monitoring Requests, *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.

Related Topics

Creating an Event Alert, page 2-8

Specializing Managers to Run Only Certain Programs, *Oracle Applications System Administrator's Guide - Configuration*

Concurrent Managers Window, *Oracle Applications System Administrator's Guide - Configuration*

Processing Alert Response Actions

When Oracle Alert receives an e-mail response to an alert message, after determining whether the response is valid or invalid, Oracle Alert submits the Alert Action Processor (ALPPWF) concurrent program to perform the appropriate actions.

If the actual response matches a valid response, then the Alert Action Processor performs the actions defined for that valid response. After the Alert Action Processor performs the actions for that valid response, Oracle Alert closes the response so that no other responses are processed for that message.

If the actual response does not match any of the valid responses in that response set, the Alert Action Processor performs the actions associated with the invalid response. The Alert Action Processor performs the invalid response actions each time Oracle Alert receives an invalid response within the response days specified in the original message action.

Note: You do not need to submit the Alert Action Processor manually. Oracle Alert submits the Alert Action Processor automatically when an alert response is received.

You can use the Requests window to review the concurrent requests for this program. See: Monitoring Requests, *Oracle Applications User's Guide*.

Processing No Response Actions

You must run the No Response Action Processor (ALPPNR) concurrent program to check for alert messages whose specified response days have passed without receiving a valid response. After the specified response days have passed, if Oracle Alert never received any response to the alert message or if it received only invalid responses, the No Response Action Processor performs the no response follow-up actions for each message and closes responses for all messages in that response set.

Use Standard Request Submission windows to schedule the No Response Action Processor. This program does not require any parameters. You should run No Response Action Processor at least once a day. See: Running Reports and Programs, *Oracle Applications User's Guide*.

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 4-21.

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 Tools 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 8-1.

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.

This chapter covers the following topics:

- Setting Up Oracle Alert
- Specifying Oracle Alert Options
- Defining Boilerplate Message Text
- Defining Other Options for Oracle Alert
- Setting Up the :MAILID Implicit Input
- Defining Alerts for Custom Oracle Applications
- Creating a Responsibility for Your Custom Application Alert Manager
- Oracle Alert and Multiple Application Installations
- Oracle Alert Naming Conventions
- Applications Window
- Registering Application Installations

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, page 7-2
- How to set up the :MAILID implicit input, page 7-6
- How to define alerts for custom Oracle applications, page 7-8

- How to use Oracle Alert with multiple application installations, page 7-10
- How to use Oracle Alert naming conventions, page 7-14

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 tabbed regions that each contain multiple options you can set.

- Message Elements, page 7-2- to specify the boilerplate text you want to appear in your message actions.
- More Options, page 7-4- to specify other miscellaneous options for Oracle Alert.

The Mail Systems, Mail Server Options, and Response Processing tabbed regions are no longer used.

Note: Oracle Alert-specific options appear only in the Oracle Alert Options form. Profile 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 Administrator's Guide - Maintenance*.

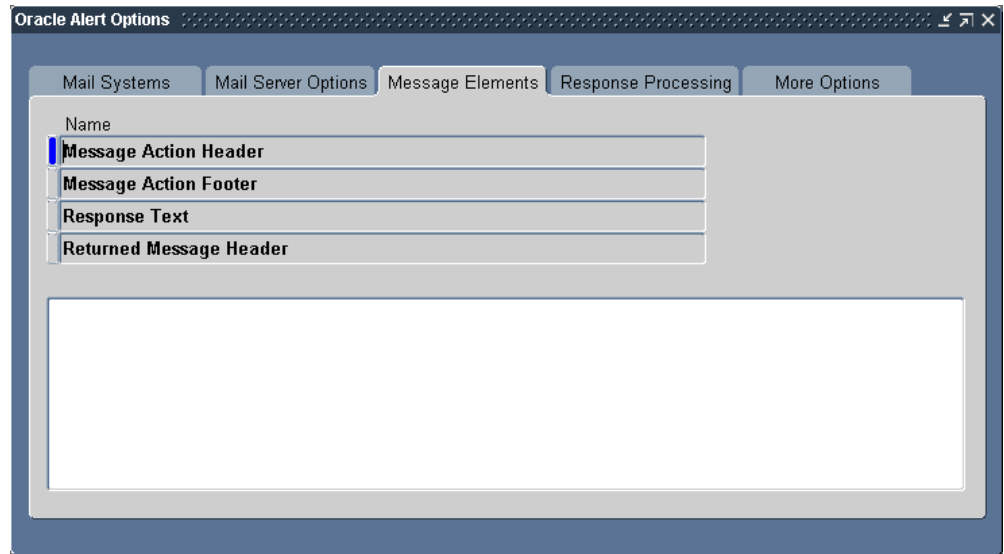
Defining Boilerplate Message Text

You can use the Message Elements tabbed 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 tabbed region.

The Oracle Alert Options Window: Message Elements Tab



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.

Related Topics

Overview of Response Processing, page 5-1

Defining Other Options for Oracle Alert

You can use the More Options tabbed 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 tabbed region.

The Oracle Alert Options Window: More Options Tab

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	/apptop/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 of the EMAIL_ADDRESS profile option 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 EMAIL_ADDRESS profile

	option has not been created or if its value 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 <i>alrmmsg</i> , located in the <i>bin</i> 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 <i>ls</i> .
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	This option is no longer used.

Setting Up the :MAILID Implicit Input

The :MAILID implicit input lets you access the electronic mail address of the application user who triggered an event alert. The value for :MAILID comes from the EMAIL_ADDRESS profile option. Before you can use the :MAILID input in your alerts, you must create and set the EMAIL_ADDRESS profile option and perform some additional setup steps.

If the EMAIL_ADDRESS profile option has not been created, or if its value 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.

To set up the :MAILID implicit input

1. Navigate to the Oracle Alert Options form. In the More Options tabbed region, set the Default User Mail Account option to the default value you want, either Operating System Login ID or Application Username. See: Defining Other Options for Oracle Alert, page 7-4.

2. If you have multiple organizations defined in your Oracle Alert schema, ensure that you specify the organization against which each alert should run. Navigate to the Alert Details window for your alert, and in the Installations tabbed region, enter the operating unit you want for each Oracle ID. Delete any records in this region that do not have a valid Oracle ID and operating unit. See: Specifying Alert Details, page 2-13.

3. Create two new profile options named RESP_ID and EMAIL_ADDRESS. In the Application Developer responsibility, navigate to the Profiles window. Enter the following information to create the first profile option:

- Name-RESP_ID
- Application-Oracle Alert
- User Profile Name-RESP ID
- Description-Responsibility ID

Enter the following information to create the second profile option:

- Name-EMAIL_ADDRESS
- Application-Oracle Alert
- User Profile Name-EMAIL ADDRESS
- Description-User's email address

See: Defining New User Profile Options, *Oracle Applications Developer's Guide*.

4. Set the RESP_ID profile option at responsibility level for each responsibility from which you trigger event alerts. In the System Administrator responsibility, navigate to the System Profile Values window. Query the RESP_ID profile option at responsibility level for the responsibility you want, and set the profile option value to the responsibility ID.

You can query the FND_RESPONSIBILITY_TL table to find the responsibility ID for a responsibility. For example, to find the responsibility ID for the Alert Manager responsibility, use the following command:

```
select responsibility_name, responsibility_id
  from fnd_responsibility_tl
 where responsibility_name like '%Alert%';
```

See: Setting User Profile Options, *Oracle Applications System Administrator's Guide - Maintenance*.

5. Set the EMAIL_ADDRESS profile option at user level for each user. In the System Administrator responsibility, navigate to the System Profile Values window. Query

the EMAIL_ADDRESS profile option at user level, and set the profile option value to the user's email address.

See: Setting User Profile Options, *Oracle Applications System Administrator's Guide - Maintenance*.

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 AutoUpgrade 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.
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 7-16.

4. Register your custom application's Oracle ID with Oracle Alert. See: Registering Application Installations, page 7-16.
5. Once you complete these steps you can define periodic alerts for your custom application. See: Creating a Periodic Alert, page 2-3.

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.

Important: 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 7-16.
3. Register your custom application's Oracle ID with Oracle Alert. See: Registering Application Installations, page 7-16.
4. Use the Table Registration API to register the table(s) that you want Oracle Alert to monitor for inserts and updates.
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 2-8.

Related Topics

Creating a Responsibility for Your Custom Application Alert Manager, page 7-9

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:

1. 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 - Configuration*.

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.

The following table shows which windows to use to perform the steps to add a new application-Oracle ID pair to the Alert Manager responsibility data group. Navigate to these windows using the System Administrator responsibility.

Window Names to Add a New Application-Oracle ID Pair to Alert Manager Responsibility

Window	Step
Applications	1. Register a new application.
Concurrent Programs	2. Make a copy of the new concurrent program and link the new application to it.
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.

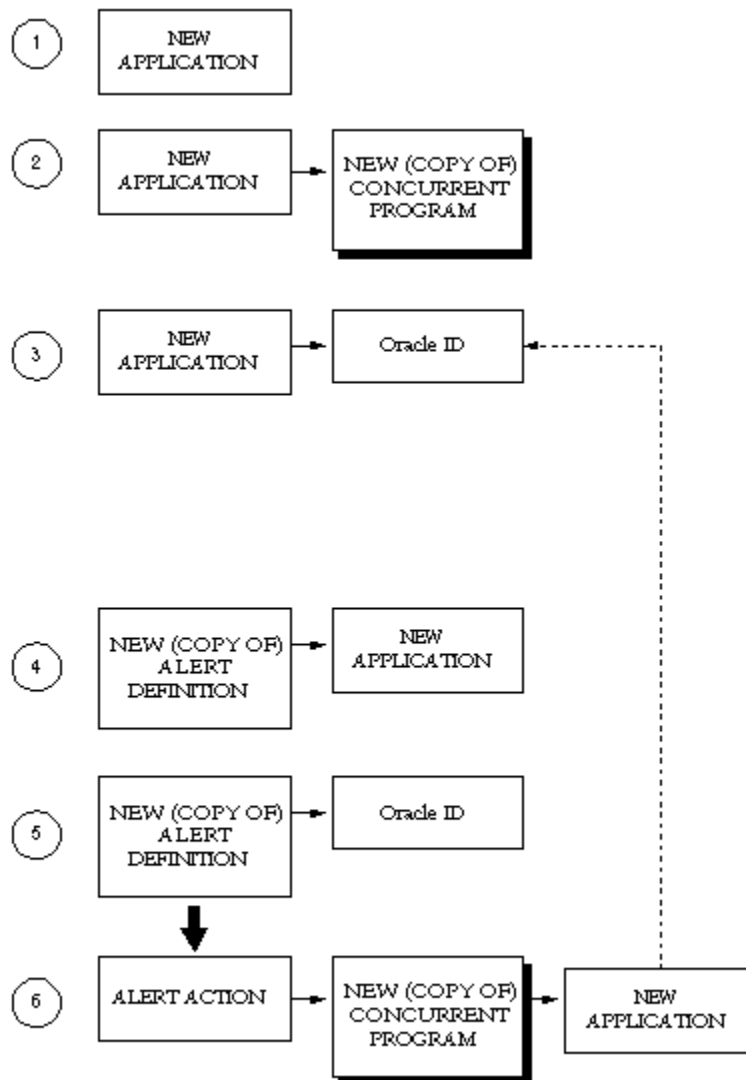
The following table shows which windows to use to perform the steps to direct an alert action to a new application-Oracle ID pair. Navigate to these windows using the Alert Manager responsibility.

Window Names to Direct an Alert Action to a New application-Oracle ID Pair

Window	Step
Transfer Alert Definition	4. Transfer (copy) your alert definition and link it with the new application.
Alert Details	5. Link your new alert definition with the Oracle ID you want to connect to.
Action Details	6. Link the concurrent request action with the new application.

The following chart shows the objects that you link in each step.

Process Flow Diagram of Connecting a Concurrent Program to Multiple Oracle IDs



When the concurrent request action runs, the concurrent program connects to the desired Oracle ID, due to the Alert Manager's Data Group.

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 7-16.
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 - Configuration*.
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 - Configuration*.
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 Tools 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 tabbed 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.

Related Topics

Defining Data Groups, *Oracle Applications System Administrator's Guide - Configuration*

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

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

Example: Concurrent Requests

Example

Purchasing Manager

- When naming a distribution list, make it plural unless it will always refer to only one person.

Example

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

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

Example: Payroll managers at headquarters

Related Topics

Using a List of Values, *Oracle Applications User's Guide*

Applications Window

Use the Applications Window to register your application name, application short name, application basepath, and application description with Oracle Application Object Library. For more information, see: Applications Window, *Oracle Applications System Administrator's Guide - Configuration*.

Registering Application Installations

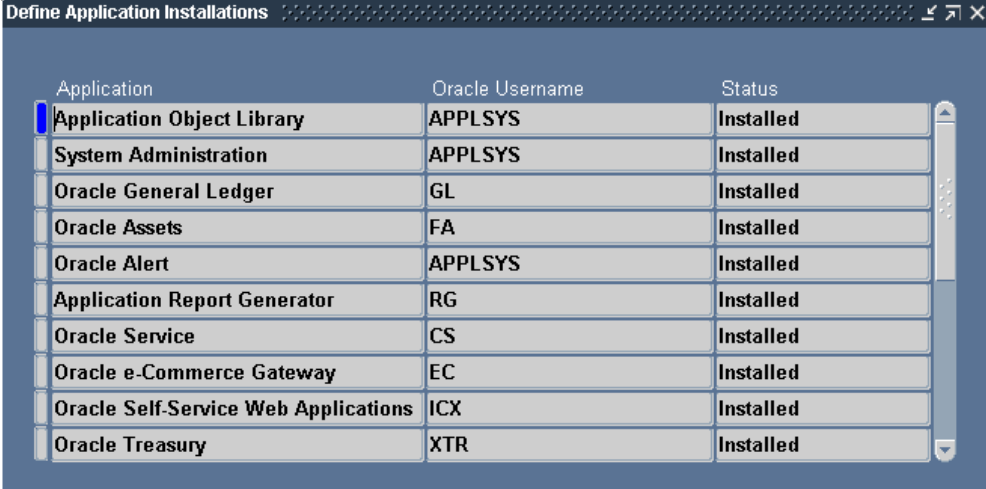
If you wish to define alerts for a custom application, you need to register the installation of the custom application with Oracle Applications first. A custom application is any application that was not installed using AutoUpgrade and resides in the same database as Oracle Alert, or any application 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 7-16.

To register a custom application installation:

1. Navigate to the Define Application Installations form.

The Define Application Installations Window



The screenshot shows a window titled "Define Application Installations". Inside the window is a table with three columns: "Application", "Oracle Username", and "Status". The table lists several Oracle applications, all of which are marked as "Installed".

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 Service	CS	Installed
Oracle e-Commerce Gateway	EC	Installed
Oracle Self-Service Web Applications	ICX	Installed
Oracle Treasury	XTR	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.

This chapter covers the following topics:

- Oracle Alert Precoded Alerts

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.

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 alerts, and suggestions on how to use them to enhance your system performance.

Related Topics

Oracle Alert DBA Alerts, page 8-2

Oracle Alert Purging Alerts, page 8-5

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.

- Frequenc: 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 - Maintenance.

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 5-3.

Implementing Oracle Alert

This chapter describes the steps necessary to implement Oracle Alert.

This chapter covers the following topics:

- Overview of Setting Up
- Related Product Setup Steps
- Setup Flowchart
- Setup Checklist
- Setup Steps

Overview of Setting Up

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 7-1

Related Product Setup Steps

Perform the following steps to implement Oracle Alert. These steps are discussed in detail in the Setting Up sections of other Oracle product user's guides.

Setup Underlying Oracle Applications Technology:

You need to complete the following setup steps:

- Perform system-wide setup tasks such as configuring concurrent managers and printers.
- Manage data and function security, which includes setting up users and responsibilities.

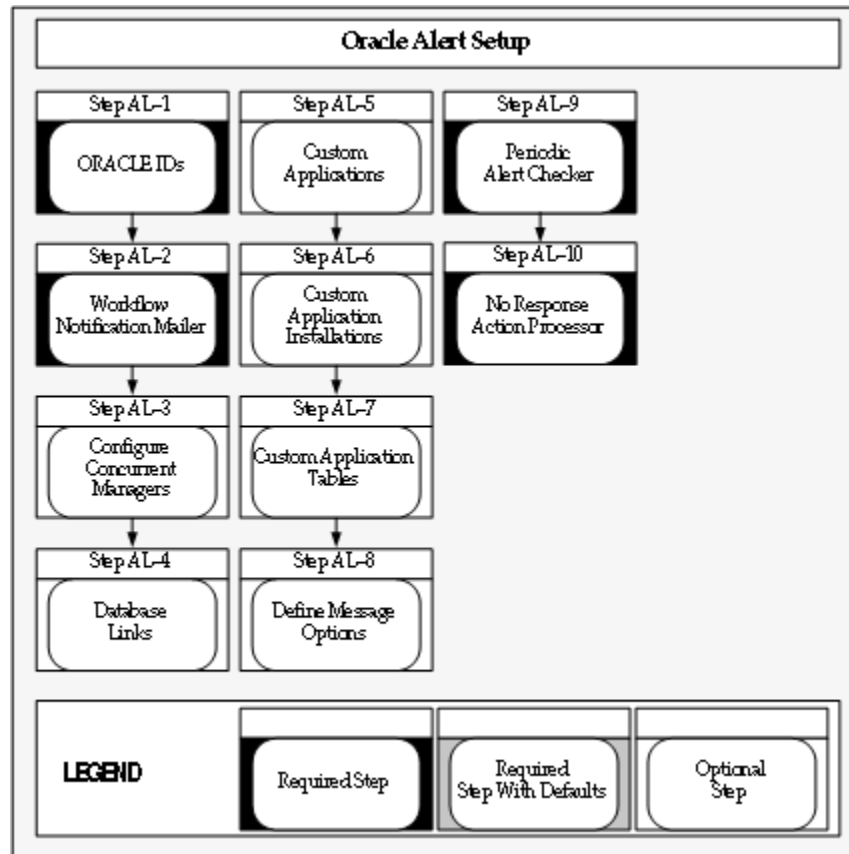
See: Setup Checklist, *Oracle Applications System Administrator's Guide - Configuration*.

Setup Flowchart

Some of the steps outlined in this flowchart and setup checklist are **Required** and some are Optional. Required step with Defaults refers to setup functionality that comes with pre-seeded, default values in the database; however, you should review those defaults and decide whether to change them to suit your business needs. If you want or need to change them, you should perform that setup step. You need to perform **Optional** steps only if you plan to use the related feature or complete certain business functions.

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:

Oracle Alert Setup



Setup Checklist

The following table lists Oracle Alert setup steps and whether a step is optional or

required. After you log on to Oracle Applications, complete these steps to implement Oracle Alert:

Setup Checklist

Step No.	Required	Step
Step 1	Required	Reregister ORACLE IDs, page 9-4
Step 2	Required	Set Up the Workflow Notification Mailer, page 9-4
Step 3	Optional	Configure Your Concurrent Managers, page 9-5
Step 4	Optional	Define Database Links, page 9-5
Step 5	Optional	Register Custom Applications, page 9-5
Step 6	Optional	Define Custom Application Installations, page 9-6
Step 7	Optional	Register Custom Application Tables, page 9-6
Step 8	Optional	Define Message Options, page 9-6
Step 9	Required	Start the Periodic Alert Checker, page 9-6
Step 10	Required	Schedule the No Response Action Processor, page 9-6

Setup Steps

For each step, we include a Context section that indicates whether you need to repeat the step for each set of books, set of tasks, inventory organization, HR organization, or other operating unit under Multiple Organizations.

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.

Context: You need to perform this step only once.

See: Register ORACLE IDs, *Oracle Applications System Administrator's Guide - Configuration*.

2. Set Up the Workflow Notification Mailer (Required)

Oracle Alert leverages the Workflow Notification Mailer to send alert e-mail messages and process responses. Check that the Workflow Notification Mailer is set up and that its configuration meets your alert requirements.

For example, if you are implementing Oracle Alert response processing, ensure that the Workflow Notification Mailer has inbound processing enabled and has an e-mail account for receiving messages on an IMAP4 compliant mail server.

You may also want to check that the From and Reply-to Address mailer configuration parameters are set to appropriate values for alert messages.

Note: Alert messages are not affected by mailer configuration parameters that are specific to workflow notifications, such as workflow message templates.

To send an outbound alert message, Oracle Alert places an XML representation of the message on an Oracle Workflow Business Event System agent called WF_NOTIFICATION_OUT. The Workflow Notification Mailer dequeues the message from this agent, converts the XML message into a MIME (Multipurpose Internet Mail Extensions) encoded message, and sends the message by the Simple Mail Transfer Protocol (SMTP).

To avoid unnecessary processing, if the Workflow Notification Mailer cannot deliver an outbound alert e-mail message because the recipient's e-mail address is invalid, the notification mailer adds the e-mail address to its invalid e-mail address list and does not attempt to send any further messages to that address.

Note: The notification mailer can store up to 100 e-mail addresses in its invalid e-mail address list. If the notification mailer encounters additional invalid addresses when the list is already full, the notification mailer removes the oldest addresses from the list and adds the new addresses in their place. Also, the notification mailer clears the list by removing all addresses whenever you stop and restart the mailer.

To handle an inbound response to an alert from a user, the Workflow Notification Mailer reads the response message and places an XML representation of it on an Oracle Workflow Business Event System agent called WF_NOTIFICATION_IN. An Oracle Workflow agent listener named Workflow Inbound Notifications Agent Listener dequeues the message from this agent and executes an Oracle Alert subscription that checks the validity of the response and submits the Alert Action Processor concurrent program to perform the appropriate actions. See: Processing Responses with the Workflow Notification Mailer, page 5-15.

Note: The Workflow Notification Mailer uses a special e-mail parser to interpret the syntax of alert messages. The e-mail parsers specified in the mailer configuration parameters do not affect alert response processing.

On the notification mailer agents, alert messages are identified by a correlation ID of ALR.

Context: You need to perform this step only once.

See: Setting Up a Notification Mailer, *Oracle Workflow Administrator's Guide*.

3. 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.

Context: You need to perform this step only once.

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

4. 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.

Important: You can only create Periodic Alerts on the remote database, not Event Alerts.

Context: You need to perform this step only once.

See: SQL*Net User's Guide for your protocol for additional information.

5. 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.

Context: You need to perform this step only once.

See: Applications Window, page 7-16.

6. 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.

Context: You need to perform this step only once.

See: Define Application Installations, page 7-16.

7. 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()` APIs.

Context: You need to perform this step only once.

8. 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.

Context: You need to perform this step only once.

See: Oracle Alert Options, page 7-2.

9. 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.

Context: You need to perform this step only once.

See: The Periodic Alert Scheduler, page 6-3.

10. Schedule the No Response Action Processor (Required)

Schedule the No Response Action Processor concurrent program using the Standard Request Submission windows. This program does not require any parameters. You should run No Response Action Processor at least once a day. See: Processing No Response Actions, page 6-7.

Context: You need to perform this step only once.

Menu Navigation Appendix

This appendix lists the forms available in the Oracle Alert Manager responsibility along with their navigation paths.

This appendix covers the following topics:

- Oracle Alert Menu Paths

Oracle Alert Menu Paths

This table shows the first window title and default menu path for each Oracle Alert form. In addition, we provide a page 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: *User***-Refer to the *Oracle Applications User's Guide* for a complete form description.
- **See: *SysAd***-Refer to the *Oracle Applications System Administrator's Guide - Configuration* for a complete form description.
- **See: *AppDev***-Refer to the *Oracle Applications Developer's Guide* for a complete form description.

Oracle Alert Menu Paths

First Window Title	Standard Menu Path
Alerts, page 2-1	Alert Define
Applications: See: <i>SysAd</i>	System Applications
Completed Requests: See: <i>User</i>	Other Requests View

First Window Title	Standard Menu Path
Concurrent Requests: See: <i>User</i>	Other Concurrent
Concurrent Requests: See: <i>SysAd</i>	Request View (with SystemAdministrator functionality)
Define Application Installations, page 7-16	System Installations
Distribution List, page 3-2	Distribution List
Oracle Alert Options, page 7-2	System Options
Periodic Set, page 3-17	Alert Periodic Set
Personal Profile Values: See: <i>User</i>	Other Profile
Profiles: See: <i>AppDev</i>	Profile
Request Periodic Alert Check, page 6-5	Request Check
Request Set: See: <i>User</i>	Other Requests Set
Review Alert History, page 4-14	History
Schedule Alert Programs, page 6-2	Request Schedule
Submit Requests: See: <i>User</i>	Other Requests Submit
System Profile Values: See: <i>SysAd</i>	Profile System

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 Action Processor (ALPPWF)

The Oracle Alert concurrent program that performs alert response actions when Oracle Alert receives an e-mail response to an alert message.

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 AutoUpgrade, you can also create your own application installation for your custom applications.

AutoUpgrade

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 the output value.

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 AutoUpgrade, 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 tabbed 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 email username 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. 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.

No Response Action Processor (ALPPNR)

The Oracle Alert concurrent program that checks for alert messages whose specified response days have passed without receiving a valid response and performs the no response follow-up actions.

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.

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 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.

Workflow Notification Mailer

An Oracle Workflow service component that sends e-mail notifications to users through a mail application, and processes e-mail responses. Oracle Alert leverages the Workflow Notification Mailer to send and receive alert e-mail messages. The notification mailer uses the Simple Mail Transfer Protocol (SMTP) for outbound messages and the Internet Message Access Protocol (IMAP) for inbound messages.

Index

A

- Action escalation, 4-8
 - customizing action sets, 4-12
 - feature description, 4-3
- Action groups, 2-32
- Action history
 - feature description, 4-2
 - reviewing, 4-14
- Actions
 - action escalation
 - feature description, 4-3
 - action sets, 2-32
 - concurrent program actions, 2-24
 - creating, 2-17
 - deleting, 2-30
 - detail actions, 2-18
 - detail concurrent program actions, 2-25
 - detail message actions, 2-20
 - detail operating system script actions, 2-27
 - detail SQL statement script actions, 2-29
 - distribution lists, 2-18
 - duplicate suppression
 - feature description, 4-3
 - feature description, 1-4
 - finding references to, 2-31
 - invalid response, 5-9
 - level, 2-18
 - message actions, 2-18, 2-18
 - entering message text, 2-20
 - that solicit responses, 5-3
 - no exception actions, 2-18
 - no exception concurrent program actions, 2-25
 - no exception message actions, 2-21
 - no exception operating system script actions, 2-27
 - no exception SQL statement script actions, 2-29
 - no response follow-up actions, 5-10
 - operating system script actions, 2-25
 - response actions, 5-4
 - reviewing action history, 4-14
 - SQL statement script actions, 2-27
 - summary actions, 2-18
 - using, 3-4
 - summary concurrent program actions, 2-25
 - summary message actions, 2-23
 - formatting, 3-4
 - summary operating system script actions, 2-27
 - summary SQL statement script actions, 2-29
- Action set checks
 - purging old, 8-6
 - reviewing history, 6-8
- Action sets, 2-32
 - creating action sets, 2-32
 - customizing for action escalation, 4-12
 - customizing for duplicate checking, 4-6
 - defining a summary threshold, 3-13
 - deleting, 2-35
 - duplicate configuration
 - for action escalation, 4-12
 - for duplicate checking, 4-6
 - duplicate suppression
 - feature description, 2-2
 - inputs

- feature description, 2-2
- Alert Action Processor, 5-16, 6-2, 6-7
- Alert check history
 - reviewing, 4-21, 6-7
- Alert checks, 6-1
 - purging old, 8-6
 - reviewing, 4-21
 - reviewing history, 6-7
- Alert customizations
 - overview, 3-1
- Alert definitions
 - transferring, 2-36
 - using PL/SQL, 2-7, 2-12
- Alert options
 - response text, 7-3
 - summary message width, 7-6
- Alerts
 - action escalation
 - feature description, 4-3
 - actions
 - feature description, 1-4
 - application installations
 - feature description, 2-2
 - application that owns the alert, 2-4, 2-9
 - copying, 2-36
 - defining
 - from external files, 2-7, 2-12
 - defining alert types, 2-4, 2-9
 - deleting, 2-36
 - detail actions
 - feature description, 1-4
 - duplicate checking
 - feature description, 4-3
 - duplicate suppression
 - feature description, 4-3
 - enabling, 2-4, 2-9
 - event alerts
 - feature description, 2-1
 - Select statement, 2-12
 - frequency
 - feature description, 1-3
 - history
 - overview, 4-1
 - saving, 2-6, 2-10
 - how alerts are checked, 6-1
 - inputs, 2-14
 - naming conventions, 7-14

- outputs, 2-15
- overview, 2-1
- periodic alerts
 - feature description, 2-1
 - frequency, 1-3
 - Select statement, 2-7
- precoded, 8-1
- response processing alerts
 - defining, 5-3
- Select statement, 2-6, 2-10
 - for event alerts, 2-12
- self-referencing, 3-16
- specifying Oracle IDs, 2-16
- summary actions
 - feature description, 1-4
 - using, 3-4
- transferring definitions, 2-36
- Verify SQL
 - feature description, 2-2

- Alert Select statement
 - for event alerts, 2-10
 - for periodic alerts, 2-6, 2-7
- Application installations
 - defining, 7-16
 - feature description, 2-2
 - specifying Oracle IDs, 2-16

B

- Blind copy recipients
 - of message actions, 2-19

C

- Check Event Alert, 6-2
 - how it works, 6-6
- Check Periodic Alert, 6-2
 - how it works, 6-5
- Column Overflow, 2-23
- Column wrap
 - in summary messages, 3-7
- Concurrent managers
 - activating, 6-2
 - optimizing for event alerts, 6-6
- Concurrent program actions, 2-24
 - detail concurrent program actions, 2-25
 - no exception concurrent program actions, 2-25
- Concurrent programs, 6-2

- Alert Action Processor, 6-2, 6-7
- Check Event Alert, 6-2, 6-6
- Check Periodic Alert, 6-2, 6-5
- No Response Action Processor, 6-2, 6-7
- Periodic Alert Scheduler, 6-2
- Concurrent request actions
 - reviewing history, 4-16
 - summary concurrent program actions, 2-25
- Concurrent requests
 - purging old, 8-5
- Constant text, 3-8
- Critical outputs
 - in summary actions, 2-23, 2-25, 2-27, 2-29
- Custom applications
 - and Oracle Alert, 7-8
 - registering, 7-8, 7-9, 7-16
- Custom tables
 - registering, 7-9

D

- Database link
 - creating, 7-8
- DATE_LAST_CHECKED, 2-14
 - in self-referencing alerts, 2-6, 2-10, 3-16
- DBA alerts, 8-2
- Detail actions
 - detail concurrent program actions, 2-25
 - detail message actions, 2-20
 - detail operating system script actions, 2-27
 - detail SQL statement script actions, 2-29
 - feature description, 1-4
 - for action escalation, 4-10
- Detail concurrent program actions, 2-25
- Detail message actions, 2-20
 - flagging duplicates, 4-7
- Detail operating system script actions, 2-27
- Detail SQL statement script actions, 2-29
- Distribution lists, 2-18
 - deleting, 3-3
 - feature description, 2-2
 - finding references to, 3-4
- Distribution Lists
 - defining, 3-2
- Duplicate checking, 4-4
 - customizing action sets, 4-6
 - duplicate configuration, 4-6

- feature description, 4-3
- outputs, 2-35
- Duplicate configuration
 - customizing, 2-34
 - for action escalation, 4-10, 4-12
 - for duplicate checking, 4-6
- Duplicate exceptions
 - description, 4-3
- Duplicate suppression
 - feature description, 2-2, 4-3
 - in detail messages, 4-7
 - in summary messages, 4-7
 - outputs, 2-16, 2-35
- Dynamic distribution, 2-20
 - feature description, 2-2
 - outputs, 2-18, 2-20
 - with summary threshold, 3-13

E

- Electronic mail integration
 - feature description, 2-3
- Escalation groups
 - creating, 4-10
 - defining, 4-9
 - deleting, 4-13
 - finding references to, 4-13
- Event alert request handling
 - configuring concurrent managers, 6-6
- Event alerts
 - defining, 2-8
 - event table, 2-9, 2-11
 - feature description, 2-1
 - for custom applications, 7-9
 - how checked, 6-6
 - inputs, 2-10
 - ROWID, 2-11
 - Select statement-example, 2-10, 2-12
- Event table
 - event alerts, 2-9, 2-11
 - insert, 2-10
 - update, 2-10
- Exception history
 - feature description, 4-3
 - outputs, 4-3, 4-20
 - reviewing, 4-19
- Exceptions

reviewing, 4-19

F

Footers

defining message action footers, 2-23

Frequency of alerts

on demand, 2-4

periodic alerts, 2-4

H

Headers

defining message action headers, 2-23

History

action history

feature description, 4-2

and action escalation, 4-9

and duplicate checking, 4-5

exception history

feature description, 4-3

of concurrent request actions, 4-16

of message actions, 4-16

of operating system script actions, 4-16

of SQL statement script actions, 4-17

overview, 4-1

response history, 4-17

feature description, 4-3

reviewing actions, 4-14

reviewing action set checks, 6-8

reviewing alert checks, 4-21, 6-7

reviewing exceptions, 4-19

saving, 2-6, 2-10

I

Implicit Input

ORG_ID, 2-14

Implicit inputs, 2-14

DATE_LAST_CHECKED, 2-14, 3-16

MAILID, 2-11, 2-14

ORG_ID, 2-11

ROWID, 2-11, 2-14

Inputs

action sets

feature description, 2-2

defining default values, 2-15

displaying, 2-14

implicit inputs, 2-14

in event alerts - example, 2-12

in periodic alerts - example, 2-7

Invalid response actions

defining, 5-9

M

MAILID, 2-11, 2-14

Menus

menu paths for all forms, A-1

Message actions, 2-18, 2-18

defining headers and footers, 2-23

detail message actions, 2-20

distribution lists

feature description, 2-2

dynamic distribution

feature description, 2-2

entering message text, 2-20

no exception message actions, 2-21

printing, 2-19

responding to, 5-14

reviewing history, 4-16

summary message actions, 2-23

formatting, 3-4

that solicit responses, 5-3

Message handles, 5-18

Message text

entering message text, 2-20

N

No exception actions

no exception concurrent program actions, 2-25

no exception message actions, 2-21

no exception operating system script actions,
2-27

no exception SQL statement script actions, 2-
29

No exception message actions, 2-21

No exception operating system script actions, 2-
27

No exception SQL statement script actions, 2-29

No Response Action Processor, 5-17, 6-2, 6-7

No response follow-up actions, 5-12

defining, 5-10

feature description, 5-2

O

- On demand alerts, 2-4
 - how checked, 6-5
- Operating system script actions, 2-25
 - detail operating system script actions, 2-27
 - no exception operating system script actions, 2-27
 - reviewing history, 4-16
 - summary operating system script actions, 2-27
- Oracle Alert
 - implementing, 9-1
 - menu paths for all forms, A-1
 - overview, 1-1
- Oracle Alert Manager responsibility
 - for defining periodic sets, 3-17
- Oracle Alert Options, 7-2
- Oracle Alert Response Processing Receive event, 5-17
- Oracle IDs
 - registering, 7-9, 7-9
 - specifying, 2-16
- Oracle Workflow, 9-4
- ORG_ID, 2-11, 2-14
- Outputs
 - displaying, 2-15
 - display length, 2-15
 - Duplicate suppression, 2-16
 - event alerts - example, 2-12
 - formatting numeric data, 2-6, 2-11
 - in alert Select statement
 - Select statement, 2-6, 2-10
 - in distribution lists, 3-3
 - in periodic alert - example, 2-7
 - in summary messages, 2-22
 - number formats, 2-6, 2-11, 2-15
 - reviewing exception history
 - Exception history, 4-3

P

- Periodic alerts
 - defining, 2-3
 - feature description, 2-1
 - frequency, 2-4
 - feature description, 1-3
 - Select statement-example, 2-6, 2-7

- Periodic Alert Scheduler, 6-2
 - activating, 6-4
 - how it works, 6-3
- Periodic sets
 - checking, 6-5
 - defining, 3-17
- PL/SQL
 - in SQL statement script actions, 2-28
- PL/SQL packaged functions, 2-7, 2-12
- Precoded alerts, 8-1
 - DBA alerts, 8-2
 - Purge Alert and Action Set Checks, 6-8
 - Purge Concurrent Requests, 8-5
- Printing
 - distribution lists, 3-3
 - message actions, 2-19
 - feature description, 2-3
 - summary messages, 3-8
- Purge Alert and Action Set Checks
 - precoded alerts, 6-8
- Purge Concurrent Requests
 - precoded alert, 8-5

R

- References window, 2-31, 3-4, 3-15, 4-13, 5-13
- Register
 - custom tables, 7-9
- Response actions, 5-4
 - feature description, 5-2
 - reviewing history, 4-18
- Response history
 - feature description, 4-3
- Response processing
 - Alert Action Processor, 6-7
 - associating with messages, 5-11, 5-12
 - defining a response processing alert, 5-3
 - how it works, 5-15
 - message handles, 5-18
 - No Response Action Processor, 6-7
 - no response follow-up actions, 5-12
 - feature description, 5-2
 - no response follow-up days, 5-12
 - overview, 5-1
 - responding to messages, 5-14
 - response actions
 - feature description, 5-2

- response sets, 5-5, 5-11, 5-12
 - message actions, 2-20
- response text, 4-18, 7-3
- response variables, 5-2, 5-6
 - feature description, 5-2
 - in concurrent request actions, 2-25
 - in message actions, 2-20, 2-24, 5-5
 - in operating script actions, 2-27
 - in SQL statement script actions, 2-29
 - in valid response text, 5-8
- response variable values
 - feature description, 5-2
- reviewing history, 4-17
- valid responses, 5-8
 - feature description, 5-2
- with the Workflow Notification Mailer, 5-15

Response sets

- creating, 5-5
- finding references to, 5-13
- message actions, 2-20

Response text

- defining, 5-4

Response variables, 5-6

- defining, 5-6
- feature description, 5-2
- in valid response text, 5-8

Response variable values

- definition, 5-2

ROWID, 2-11, 2-14

S

Select statement

- for event alerts, 2-10
- for periodic alerts, 2-6
- inputs, 2-6, 2-10
- number formats, 2-6, 2-11, 2-15
- outputs, 2-6, 2-10
- using PL/SQL, 2-7, 2-12

Self-referencing alerts, 3-16

- checking, 3-16
- creating, 3-16
- use of DATE_LAST_CHECKED, 2-6, 2-10

SQL statement script actions, 2-27

- detail SQL statement script actions, 2-29
- no exception SQL statement script actions, 2-29

- reviewing history, 4-17
- summary SQL statement script actions, 2-29

Summary actions

- feature description, 1-4
- summary concurrent program actions, 2-25
- summary message actions, 2-23
- summary operating system script actions, 2-27
- summary SQL statement script actions, 2-29
- using, 3-4

Summary concurrent program actions, 2-25

- critical outputs, 2-25

Summary message actions, 2-23

- column wrapping, 2-23, 3-7
- constant text, 3-8
- critical outputs, 2-23
- defining output lengths, 3-5
- flagging duplicates, 4-7
- formatting, 2-22, 3-4
- maximum width, 2-22, 2-23
- printing, 3-8
- summary message template, 2-23
- using column wrap, 3-7
- using outputs, 2-22

Summary message template, 2-23

Summary operating system script actions, 2-27

- critical outputs, 2-27

Summary SQL statement script actions, 2-29

- critical outputs, 2-29

Summary threshold

- action sets, 3-13
- and dynamic distribution, 3-13
- defining, 3-11
- threshold groups, 3-10

T

Tables

- registering, 7-9

Threshold groups

- defining, 3-11
- deleting, 3-14
- finding references to, 3-15

Transferring alert definition, 2-36

V

Valid responses

- defining, 5-8

- feature description, 5-2
- Valid response text
 - defining, 5-4
- Verify SQL
 - feature description, 2-2
 - when defining alerts, 2-7, 2-13

W

- Workflow Notification Mailer, 9-4

