Oracle® Retail Active Retail Intelligence

User Guide
Release 13.0.1.1
E38599-02

December 2012

Oracle® Retail Active Retail Intelligence User Guide, Release 13.0.1.1

Primary Author: Kris Lange

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server – Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning, Oracle Retail Demand Forecasting, Oracle Retail Regular Price Optimization, Oracle Retail Size Profile Optimization, Oracle Retail Replenishment Optimization applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by SAP and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Preface

The Oracle Retail Active Retail Intelligence User Guide describes the application user interface and how to navigate through it.

Audience

This document is intended for the users and administrators of Oracle Retail Active Retail Intelligence. This may include merchandisers, buyers, and business analysts.

Active Retail Intelligence Administrators

Most of this guide is directed toward the people in your business who are responsible for configuring and maintaining Active Retail Intelligence. These users define the business rules under which Active Retail Intelligence performs exception management. Ideally, configuring Active Retail Intelligence is a cooperative effort involving the following types of people in your business:

- A business analyst, with both a functional understanding of the business and a technical understanding of the software used in the business.
- A database administrator with database knowledge and Oracle programming skills.

In this cooperative effort, business analysts work with members of the business to define requirements and with database administrators to implement the requirements. A business analyst may take on the major portion of those responsibilities.

If you are responsible for setting up and maintaining Active Retail Intelligence, you should read the entire guide.

Active Retail Intelligence End Users

The Active Retail Intelligence end user receives alerts and takes actions to resolve the alerts. Any user of Oracle Retail applications, such as the Oracle Retail Merchandising System (RMS), could be an Active Retail Intelligence end user.

Related Documents

For more information, see the following documents in the Oracle Retail Document Template Release 13.0.1.1 documentation set:

- Oracle Retail Active Retail Intelligence Operations Guide
- Oracle Retail Active Retail Intelligence Release Notes

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://metalink.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

If you are installing the application for the first time, you install either a base release (for example, 13.0) or a later patch release (for example, 13.0.2). If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Patch documentation can contain critical information related to the base release and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site (with the exception of the Data Model which is only available with the release packaged code):

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement "the Window Name window opens."

> **Note:** This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

```
This is a code sample
    It is used to display examples of code
```

A hyperlink appears like this.

1

Active Retail Intelligence (ARI)

Active Retail Intelligence (ARI) is an exception management and resolution system driven by business rules that you define.

This guide shows you how to:

- Set up and maintain the business rules that drive your Active Retail Intelligence system
- Define users of Active Retail Intelligence and how they can use Active Retail Intelligence
- Define exceptions and events
- View alerts and the events associated with them

Organization and Intended Audiences

Following is a summary of the parts of this user guide, with comments about the intended audience for each part.

This guide is divided into two main parts: Setting Up and Maintaining Active Retail Intelligence, and Using Active Retail Intelligence. In addition, there is a glossary of Active Retail Intelligence terms.

Active Retail Intelligence Setup and Maintenance

Active Retail Intelligence Setup and Maintenance describes how to configure and maintain Active Retail Intelligence. This part of the guide covers the following topics:

Active Retail Intelligence Overview is critical reading for business analysts. It introduces Active Retail Intelligence terminology in context and should serve as a reference point when trying to understand the big picture of how the Active Retail Intelligence application components fit together. It also contains sections on conventions used throughout the manual and common features used in many of the system interfaces.

Metadata Maintenance discusses how to create and maintain the metadata that you will use to build your business rules into the system, and to define exceptions, events, and alerts. This task is intended for technical analysts who understand data modeling, the Oracle Retail Data Model, PL/SQL, and/or any external data system that Active Retail Intelligence is intended to monitor.

Active Retail Intelligence User and Group Administration explains how to create Active Retail Intelligence user and user group definitions. This task is intended for business administrators, though it may be set up initially by business analysts.

Exception and Event Definition describes how to define exceptions to your business rules, and the events associated with those exceptions. This task is intended for business analysts who have a clear understanding of the retailer's practices, the application's capabilities, and some understanding of the data model to be monitored.

Schedules for Exception Scanning and Event Reevaluation explains how to use ARI’s scheduling feature to set up periodic scheduling of processes, customize the scheduling of exception scanning, and set when and how often events are reevaluated to determine whether they are still in effect. As with managing exceptions and events, this task is intended for business analysts.

Language Translation describes how to prepare Active Retail Intelligence to run in one or more target languages. Active Retail Intelligence includes a form for translating strings into the target languages.

Examples contains detailed examples using the Active Retail Intelligence interfaces, focusing primarily on planning the resolution of a business problem and then explaining how Active Retail Intelligence administrators and users would utilize the administrative and user interfaces to implement a solution.

Active Retail Intelligence Use: View Alerts and Events

Using Active Retail Intelligence describes how to use Active Retail Intelligence once the business rules are in place, and exceptions and events are defined and scheduled. Tasks in this section are performed by Active Retail Intelligence end users. This part of the guide includes the following topics:

About Active Retail Intelligence provides a high-level overview of Active Retail Intelligence.

View Alerts explains how to view and take actions on alerts sent to you.

View Events explains how to view and take action on events associated with an event details and enables users to act on events. This task is performed by Active Retail Intelligence end users.

Active Retail Intelligence Glossary

For Active Retail Intelligence administrators, it is critical to understand and distinguish between key terms and concepts in Active Retail Intelligence, such as exceptions and events. A glossary is available for you to review these terms.

Active Retail Intelligence Glossary

Additional Active Retail Intelligence-Related Documentation

The Active Retail Intelligence Overview and other topics within the Setup and Maintenance section of this help system present a good deal of background on where to get started and how to use Active Retail Intelligence. Additional operational information is included in the Active Retail Intelligence Operations Guide, the Oracle Installation Guide, and Oracle Batch Schedule document.

Common User Interface Controls in ARI

This section introduces five common interface controls in ARI screens.

- Multisort
- Multiselect
- Multisearch
- Multifilter
- Workbench

Multisort

Multisort is available on any data blocks that have buttons for column headings. Click the column heading button and the data in the block will be sorted in ascending order based on the column for which the button was clicked. Click the same button again to sort the block by descending order based on the column's contents. Some columns on a block may not be available for Multisort. Others that are available may perform somewhat slowly because the amount of data in the block is large and sorting is inherently slow. The functionality was nevertheless left available to you to use at your discretion rather than taking out functionality on the basis of what some users may consider too slow. With a little experience, you will learn which buttons perform reasonably and are useful. If you feel performance on a sort is critical, a system administrator could add an index to that column, but performance impacts elsewhere would warrant consideration before making such a change.

Multisearch

Multisearch is enabled on many blocks. There is no visual indicator on the block, but the Edit menu will contain a search option. To use Multisearch, click the cursor in the field you want to search on and invoke the search option. Searching will move you to the first record in the block that matches your criteria if you search from the beginning. Otherwise it will move to the first record after the current record (as currently ordered in the block) that matches your criteria. Because of its wide availability, the documentation generally does not mention Multisearch for every block of every screen even when it is applicable.

Multifilter

Multifilter allows users to filter the data in the block displaying only rows that match the filter criteria entered. The matching is partial, not exact, and is based on the values in the column corresponding to the filter field. Filter criteria can be entered for more than one field. The visual clue that Multifilter is available is a row of fields above the block headings that are always editable. You will also see the filter and clear buttons side by side above the filter row. Typically Multisort is available on all Multifilter blocks and is available on the same columns. Multifilter has all of the same issues as Multisort in terms of column availability and performance.

Multiselect

Multiselect allows a user to select multiple rows in a single block by using the Shift and Control keys. This is so users can perform a single action on multiple rows. To use Multiselect, click on a row. Shift click on another row to select all the rows in between, inclusively. Control click on rows to select or deselect the individual rows clicked on (depending on whether they are currently selected) while leaving the other selected rows still selected. The functionality is analogous to the select functionality of other Windows applications. A known issue is that scrolling through a Multiselect block while Multiselecting can cause the first or last row to not appear selected (highlighted). Instead, only the text appears to be selected. This is an issue with the Oracle toolset's scrollbars not being widgets, so it is recommended that Multiselect actions involving scrolling be confirmed after they are performed.

Workbench

A workbench is an interface that contains several related dialogs in one form. A dialog is a screen or series of screens that performs a single logical unit of work, so each dialog could be presented in its own form or window. However, some tasks, though logically independent, are functionally closely related and likely to be performed together as part of a larger work plan. A workbench helps group these functionally related tasks and facilitate switching between them.

The standard workbench interface used in ARI has a common workbench area and a dialog specific area. The common area is a column of vertical buttons on the left side of the screen and a row of buttons across the bottom of the screen. This common area may include other features on specific workbenches, but these are the standard features:

Vertical Buttons

Each of these represents one of the dialogs in the workbench. These are used to switch between dialogs.

OK Button

Saves all changes made in any of the dialogs and exits the workbench.

Cancel Button

Cancels all unapplied changes and exits the workbench. Unapplied changes are any that have been made since the last use of the Apply button or since the form was opened if the Apply button has not been used.

Apply Button

Saves all changes made in any of the workbench's dialogs.

One of the unique features of the workbench is that none of the changes you make in a given dialog are saved to the database until you complete the dialog and use one of the main workbench buttons to either save or cancel your changes. While a dialog is in progress, the workbench buttons are disabled and the dialog specific buttons act on that dialog alone.

Reserving saving of work for the workbench level and making it available only at appropriate points in the dialog opens up a number of new working strategies. You can either do lots of work tentatively with the option of canceling it all, or you can *apply* your work often, reserving Cancel to undo only your most recent changes. Generally, it is a good idea to complete a dialog and apply changes, when leaving the workbench open but moving on to another task, to avoid inadvertently canceling when returning to the workbench and losing all the work completed during that session.

Posting and Rollbacks

OK buttons *within* dialogs of a workbench are described throughout as posting changes. Cancel buttons are described as rolling changes back. Posting is like a temporary save that can still be undone manually (if the editor allows) or by the workbench's main Cancel button until a workbench's main Apply or OK button is used. Rollbacks just undo the changes made on the current screen.

2

ARI Setup and Maintenance

ARI Overview

Starting and Using Active Retail Intelligence

Active Retail Intelligence is accessed from the Active Retail Intelligence start form, or from any of several places in RMS, depending on whether you are using Active Retail Intelligence or setting up and maintaining Active Retail Intelligence.

Dialogs

The Active Retail Intelligence user interface consists of two main parts:

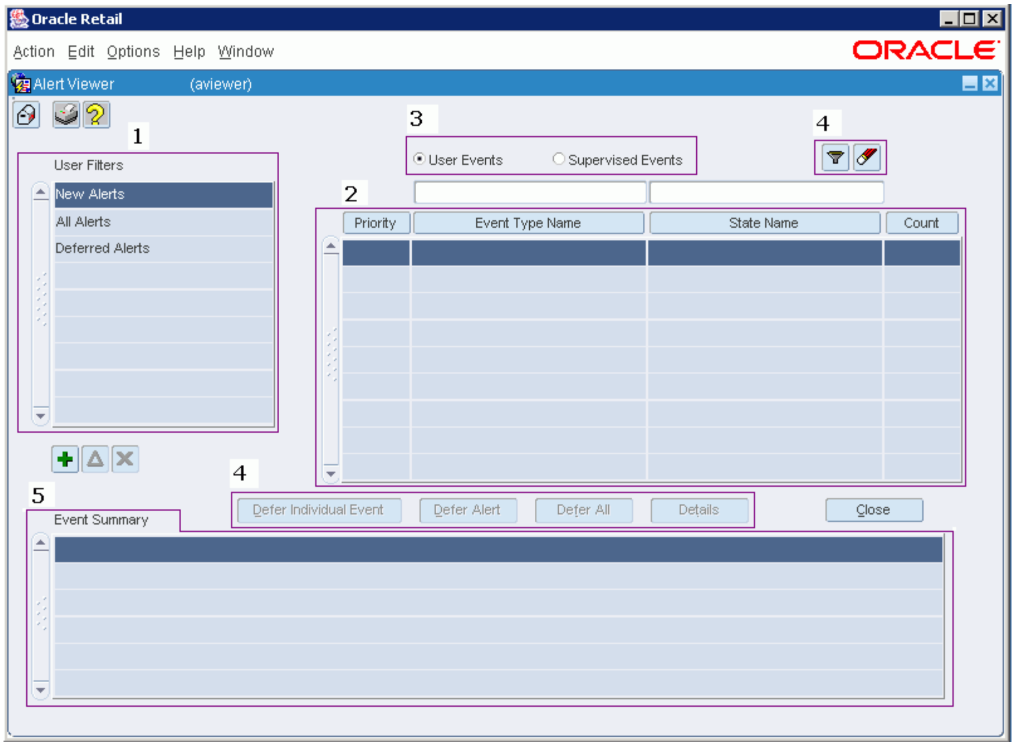
- Dialogs for setting up and maintaining Active Retail Intelligence for your business
- Dialogs and displays for end users to use Active Retail Intelligence after it is set up

Common Buttons and Controls

All of these dialogs share a common set of buttons and use several common user interface controls.

To View Events Assigned To You

1. On the RMS toolbar, if you have new events assigned to you, you will see the **Alert** button on the RMS toolbar. If no new events are assigned to you, the button is displayed as **No Alert** button.
2. Click **Alert** button. The Alert Viewer is displayed, with a summary of the events assigned to you.



Alert Viewer Screen

To Maintain Active Retail Intelligence (Active Retail Intelligence Administrators Only)

Active Retail Intelligence comes with instructions for adding the Active Retail Intelligence administration forms to the main menu for RMS or other Active Retail Intelligence-integrated application. Check with your Oracle system administrator for the method used to access these forms.

Active Retail Intelligence End User Interface

For an end user, using Active Retail Intelligence is largely system-initiated, and consists of using two main viewers or dialogs:

- Alert Viewer
- Event Viewer

Alert Viewer

The Alert Viewer displays all alerts routed to you. In the Oracle Retail Merchandising System, or other Active Retail Intelligence-integrated application, clicking the as or Alert button on the toolbar displays the Alert Viewer.

If you have new alerts, the button is displayed as Alert; otherwise, the button is displayed as . This button is updated every time you open a new window in RMS.

The Alert Viewer shows end users the priority and a count of the number of instances of each event type assigned to them. New alerts can be deferred out of New status, but just opening the alert viewer does not take an alert out of New status. An alert is new until it is deferred or the event it represents is closed.

Event Viewer

Clicking the Details button in the Alert Viewer displays events associated with an alert in the Event Viewer. The Event Viewer displays additional details about events, including actions to drill into the system to further examine the cause of the event, and other actions that you can take to resolve events.

Active Retail Intelligence Setup User Interface

Setting up Active Retail Intelligence and maintaining Active Retail Intelligence involves using several dialogs:

- Metadata administration
- User and group administration
- Exception Manager
- Event Manager
- Scheduling

Security Considerations

Active Retail Intelligence provides no special security features or safeguards. Addressing any site-specific security issues involving Active Retail Intelligence is the customer's responsibility. Security settings in other applications with which Active Retail Intelligence interacts will not be overridden or circumvented by Active Retail Intelligence. Whereas this is generally desirable, it is a consideration when determining to whom Active Retail Intelligence alerts should be routed. Sending an alert to a user who does not have the privileges to take the actions necessary to resolve the event may prove frustrating and counter-productive. Users should be educated about this issue so that they can avoid forward events that have actions with limited access as well.

At a data level, Active Retail Intelligence detection is necessarily done with full access privileges to all data. Individual users with data level security may see different values for some parameters (in particular those involving sums) than the values seen by Active Retail Intelligence. This may cause adverse effects, such as a user looking at an event automatically causing it to close because the user's limited data access causes the event to see values that make Active Retail Intelligence think the exception is no longer an issue when in fact it still is. For this reason, Oracle urges extreme caution when designing Active Retail Intelligence processes that involve users with limited data access. The consequences of missing alerts can be great in an exception driven enterprise, so extra care is needed in the technical analysis of how such Active Retail Intelligence processes will behave.

Metadata Maintenance

Maintaining metadata is a key setup and maintenance task for Active Retail Intelligence.

What is Metadata?

Metadata is data about data; that is, it is a description of the data sources available to Active Retail Intelligence, presented at a level above the actual data descriptions. This simpler level of data detail allows you to deal with the data along functional rather than structural lines. Metadata definitions are descriptions of data tables, with the added convenience of being able to provide additional information that was not in the original definitions of the tables and to logically group like data. Metadata can also be used to describe other data sources such as functions and actions.

Metadata must be defined so that exceptions and events will process properly. Maintaining metadata involves defining and editing the data objects that you will use to set up and maintain rules, events, and actions for your Active Retail Intelligence system.

Metadata Concepts

Active Retail Intelligence metadata consists of several concepts and types of information:

- Parameters
- Parameter types
- Realm types
- Realms
- Lookups

When to Maintain Metadata

Metadata must be defined before you can proceed with defining exceptions and events.

Metadata must be synchronized with and accurately reflect the data model that drives your system, or Active Retail Intelligence will malfunction. This means that any changes made to tables and functions must be reflected in the Active Retail Intelligence metadata.

Because you are modifying the data that drives your Active Retail Intelligence system, metadata maintenance should be performed when there are no Active Retail Intelligence users besides yourself on the system. Consult the Active Retail Intelligence Operations Guide for more information.

Ongoing Metadata Maintenance

Metadata maintenance is one of the most critical tasks in Active Retail Intelligence. No matter how well a business analyst defines exceptions and events, if the metadata does not accurately describe the DDL, Active Retail Intelligence will not work correctly. Fortunately, because metadata reflects data storage specification and code specifications, it is not likely to change often in a redundant environment.

Following are requirements and recommendations for maintaining metadata.

Before Using Active Retail Intelligence

Prior to the use of any Active Retail Intelligence functions, the REALM.SCHEMA field must be updated to reflect that actual schema name in which the functions and tables are deployed. This can be done through the user interface for several 100 realms, but is better done in SQL*Plus by looking at what already has a schema name and validating against the DDL. A couple of update statements should be sufficient.

Adding New Functions

A business analyst may define functions to calculate values not stored on the database or to act as action shells around existing Oracle functionality so that actions available on an event can be created. In these cases, metadata for these new DDL objects can be created at any time without regard to the production schedule. This is generally true of any newly created objects, including new external data sources monitored through the API.

Changing the DDL

When the DDL is changed, the metadata needs to be updated to reflect it immediately. Active Retail Intelligence processes that expect data objects with one definition and instead find a different definition will, as a result, miss processing any exception candidates that might have occurred between the time that the DDL is changed and the metadata is updated.

Updating the metadata alone is not enough to start catching the new exceptions, but it will at least prevent time-wasting error generation. In some cases, the exception being monitored may not be critical enough that a gap in the exception record is even critical, or if it is not real-time monitored, a delay in detecting the exception (which is only periodically monitored anyway) may not be critical. However, in many cases the goal is in fact to catch all real-time exceptions.

In this case, after updating the DDL and the metadata, it is necessary to rerun the ARI_CONTROL_SQL processes to rebuild the monitors and associated support code. Of course, this process must run with no other user logged in or any other process running (as defined in the Oracle Retail Batch Schedule). This means that it may be difficult to rerun the process if the DDL and metadata are updating during the day; but this shouldn't be a problem in a production environment since, in such an environment, DDL changes are not made during production hours anyway.

The recommended procedure for changing the DDL is: run the shutdown process, disconnect users, change DDL, update metadata, run the build process, bring the database back up, and run the restart process.

Creating Function Shells

Many actions within the Oracle Retail Enterprise Suite can be used directly from Active Retail Intelligence simply by describing them in metadata. Others involve user interaction, depending on certain checks made as the process is performed. Generally it is best to take these actions by using ARI's Event Viewer as a gateway into the appropriate GUI and then taking the action, using the traditional method defined in the GUI. In some cases, however, the user interactions have a limited number of outcomes and it may be useful to create a fully automated action that, when demanding user interaction, uses a predefined value.

An example of this is approving a purchase order check the open to buy. Two function shells could be created for approve order, one of which approves even if the OTB is exceeded. One does not proceed if the OTB is exceeded. The user can set up automated rules then to approve even if over the OTB, or can set up rules to approve but not if over the OTB. These action shells are described as two different actions in metadata and behave like two different functions, even thought they are coded around the same base Active Retail Intelligence code. In fact, it is possible to create a single shell with two different INSTANCE IDs in the metadata that have different default values for an input parameter that determines whether to ignore the OTB warning.

Impact of RMS Data Dictionary on Active Retail Intelligence Metadata Administration

Data model irregularities are not unusual in complex systems, since performance, flexibility, and functionality are constant requirements that force many compromises. The same metadata that allows the application's access to the RMS data model also exposes the irregularities of that same data model, unlike any existing RMS interface. This would be true in general for any system monitored by Active Retail Intelligence via metadata, and is not in itself problematic. In fact, the metadata as designed has in some cases been used to build in compensation for these irregularities. Without regard for the benefits or reason for the RMS data model being constructed as it is, following are brief discussions of some of the specific areas of the RMS data model that impact Active Retail Intelligence administration.

Split Location Tables

Even though locations are unique, the location tables are split by location type (store and warehouse). Thus, to monitor certain kinds of location related exceptions, similar but distinct exceptions may need to be defined on more than one table. These exceptions may be the best example (see the Examples chapter) of multiple exceptions driving the same event, which could be done only if the event could get its additional information from the common tables and precisely because the key identifiers are unique across the tables.

DDL Navigation

To define conditions against a unique set of metadata requires that information can only be gathered from tables that are somehow the parents of whatever table is being monitored. You can only navigate "up" the foreign key chain. To navigate downward, to check whether a ITEM exists on an order, or to compute a total, or otherwise check an attribute of detail information that can be treated as a single row of data, requires functions to return calculated fields. Details on setting up these functions are discussed elsewhere in this section and in the Examples section.

Metadata Definition

Active Retail Intelligence installation loads metadata appropriate to the release of RMS with which it is released. However, use of external data systems or custom modifications that add new tables or change existing ones requires metadata maintenance.

- Internal system metadata maintenance
- External system metadata maintenance

Internal System Metadata Maintenance

Internal metadata, describing Active Retail Intelligence, the RMS, or some other integrated Oracle systems sharing the same DDL and owner as these products, is generally best defined before the systems are put into production.

Suppose that, through custom modifications, you add a new table to the RMS or modify an existing table. Before putting the system into production or creating Active Retail Intelligence exceptions and events, you should always make the metadata accurate. This includes deleting parameters representing removed columns, adding parameters representing new columns, and modifying those representing changed columns. Depending on the change, the modification may simply be a change to a parameter type. Consider these examples:

Example 1

Suppose you add new columns to the RMS ITEM tables to categorize items in terms of shelf life and, where applicable, record the duration of that shelf life. Using the auto-discovery feature of the Parameter dialog, you could start at the ITEM table and auto-discovery would successively pre-populate data about these new columns. You would likely create a new parameter type for the shelf-life category, and use an existing parameter type such as *number of days* parameter type to describe the shelf-life duration.

Building on this example, if you added a code to the code_head and code_detail table to describe the various shelf-life categories, you might add a code type to the shelf-life category parameter as well. A code type such as SHLF could contain the codes FOOD and PHARM for food items and pharmaceuticals. If, instead, the number of categorizes were to be dynamically created by the buyers as needed, the categories might be stored on a new shelf-life category table with a code and description column. In this case you would first describe the new table and then use its description column as a decode column for the parameter representing the shelf-life category on the ITEM tables.

Example 2

As another example, consider if you needed all ITEM numbers to be able to have a length of 10 instead of only 8 as is currently the practice in the RMS. After making the changes in the RMS to tables and code, the only change that would be needed for Active Retail Intelligence would be a modification to the ITEM parameter type. Because all parameters that contain ITEMs should already have a parameter type of ITEM and it is the parameter type that contains the data length and type information, not the parameter itself, a single modification to the type modifies the whole system.

The difficulty in the parameter type example is that after the parameter types are in use (to describe parameters), they cannot be modified. Instead, a new parameter type (ITEM - 10, for example) must be created and then all of the ITEMs must be assigned to it so that, no longer part of the definition of ITEM parameters, the parameter type (ITEM-8) can be deleted. Such a mass update is likely easier done in SQL*PLUS by a system administrator, but these illustrations are intended to show you both when and how to use the tools provided and when to use another method.

Challenges of Changing Table Structures During Production

As these examples show, there are inherent difficulties with changing the table structures of the RMS after it has been put into production. Data dependencies make certain changes difficult to make once the RMS system is in use, and so it is with Active Retail Intelligence. As shown in the last example, some of these dependency difficulties exist in Active Retail Intelligence as well. Thus, you should know that pre-production metadata synchronization is critical. Post-production changes, except possibly in the form of new tables, are more difficult (though still possible).

External System Metadata Maintenance

Active Retail Intelligence does not come with any external realms pre-defined. This is because the external feeds that are available depend on the reports an individual retailer is interested in monitoring.

It might be helpful to describe the Active Retail Intelligence realm table as internal metadata and define an exception against it notifying the Active Retail Intelligence Administrator every time the active indicator is set to *NO* on a realm that has an active or future exception defined against it. This would prompt the Active Retail Intelligence Administrator to define a new realm and create a new version of the exception to monitor this new realm.

Defining a realm for a new headline version is just like defining a realm for the first version of a headline. The process of creating a new exception version when the realm changes is discussed in a later example.

Metadata Maintenance Form [metadata]

From the Metadata Maintenance form, you can create, edit, and delete metadata such as parameters, realms, parameter types, and lookup data.

Accessing Views of Metadata

On the left side of the Metadata Maintenance form is a set of buttons for accessing the various views, or screens, for defining and maintaining metadata.

Realm

The button displays the Realm Find view, from which you access the Realm Maintenance view for creating and editing realms.

Parameter

The button displays the Parameter Find view, from which you access the Parameter Maintenance dialog for creating and editing parameters.

Parameter Type

The button displays the Parameter Type Find screen, from which you access the Parameter Type Maintenance for creating and editing parameter types.

Lookup Data

The button displays the Lookup Find view, from which you access the Lookup Data Edit screen for creating and editing lookups.

Committing and Canceling Changes

At the bottom of the Metadata Maintenance form is a set of buttons for committing edits to the various views on the form, and for exiting the form.

OK

Commits all changes and closes the Metadata Maintenance form.

Cancel

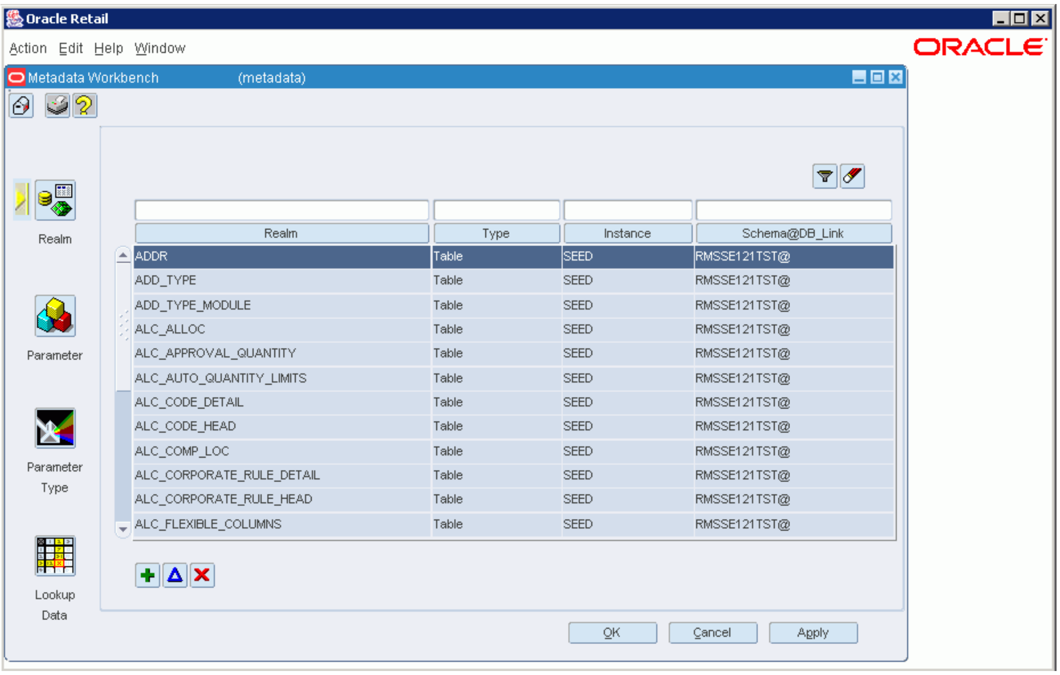
Closes the Metadata Maintenance form without saving any changes.

Apply

Commits all changes. The Metadata Maintenance form remains open.

Realm Find Screen

The Realm Find screen lists all of the active realms in your Active Retail Intelligence system. This view is accessed by clicking the button on the Metadata Maintenance form.



Realm	Type	Instance	Schema@DB_Link
ADDR	Table	SEED	RMSSE121TST@
ADD_TYPE	Table	SEED	RMSSE121TST@
ADD_TYPE_MODULE	Table	SEED	RMSSE121TST@
ALC_ALLOC	Table	SEED	RMSSE121TST@
ALC_APPROVAL_QUANTITY	Table	SEED	RMSSE121TST@
ALC_AUTO_QUANTITY_LIMITS	Table	SEED	RMSSE121TST@
ALC_CODE_DETAIL	Table	SEED	RMSSE121TST@
ALC_CODE_HEAD	Table	SEED	RMSSE121TST@
ALC_COMP_LOC	Table	SEED	RMSSE121TST@
ALC_CORPORATE_RULE_DETAIL	Table	SEED	RMSSE121TST@
ALC_CORPORATE_RULE_HEAD	Table	SEED	RMSSE121TST@
ALC_FLEXIBLE_COLUMNS	Table	SEED	RMSSE121TST@

Metadata Maintenance Form

Field Description

Information displayed about each realm includes:

Realm

The realm name. This field is read-only.

Type

The realm type. This field is read-only.

Instance

The instance ID for the realm. The instance allows you to create more than one version of a realm in case of different database instances of the same realm.

Schema@DB_Link

The concatenation of the database link and schema for the realm, as defined on the Realm Maintenance screen, and displayed in the format schema@db_link.

Sorting and Filtering

You can sort the realms by the categories of information by clicking the column heading above each column.

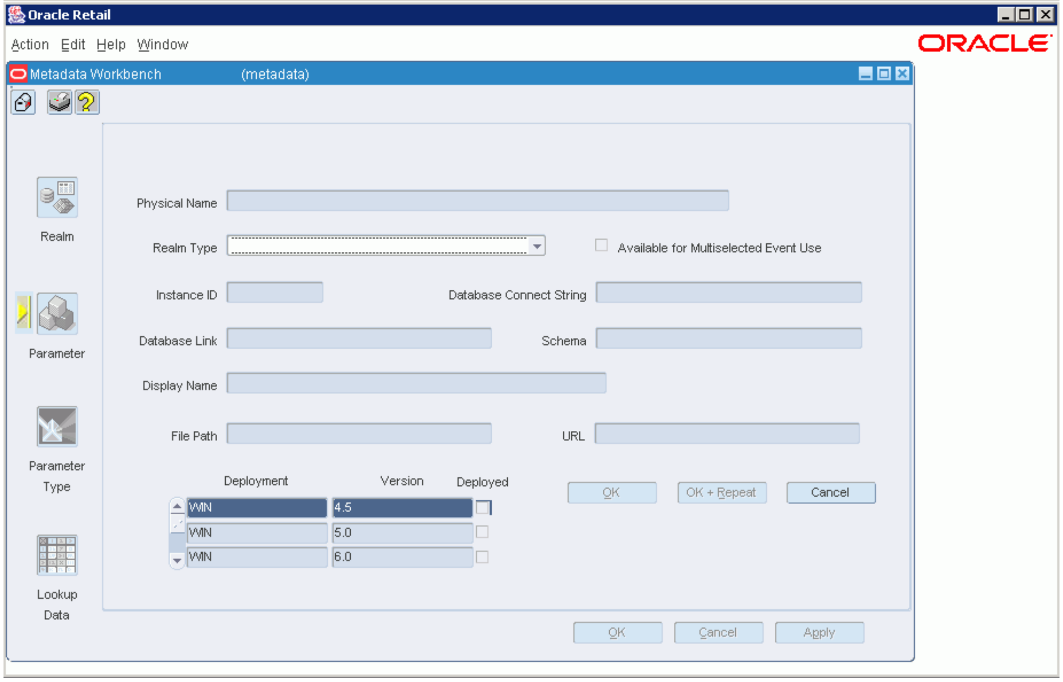
You can filter the view or realms by filling in the fields above each column and clicking the ▽ button. The ✎ button clears any active filter.

Adding, Deactivating, and Editing Realms

The three buttons at the bottom of the list are for adding, deactivating, and editing realms.

Add Button

The ✚ button adds a realm. The Realm Maintenance view is displayed, where you can add the new realm.



Deployment	Version	Deployed
WIN	4.5	☐
WIN	5.0	☐
WIN	6.0	☐

Realm Maintenance View Screen

Delete Button

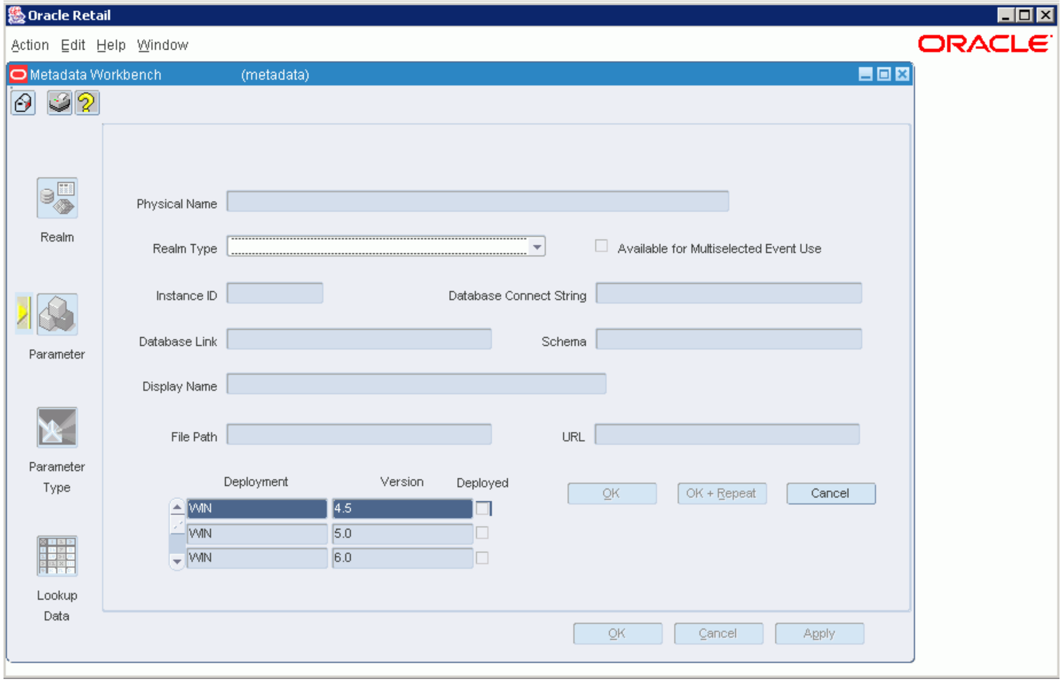
The ✖ button deactivates the selected realm.

Edit Button

The △ button edits the selected realm. The Realm Maintenance screen is displayed.

Realm Maintenance Screen

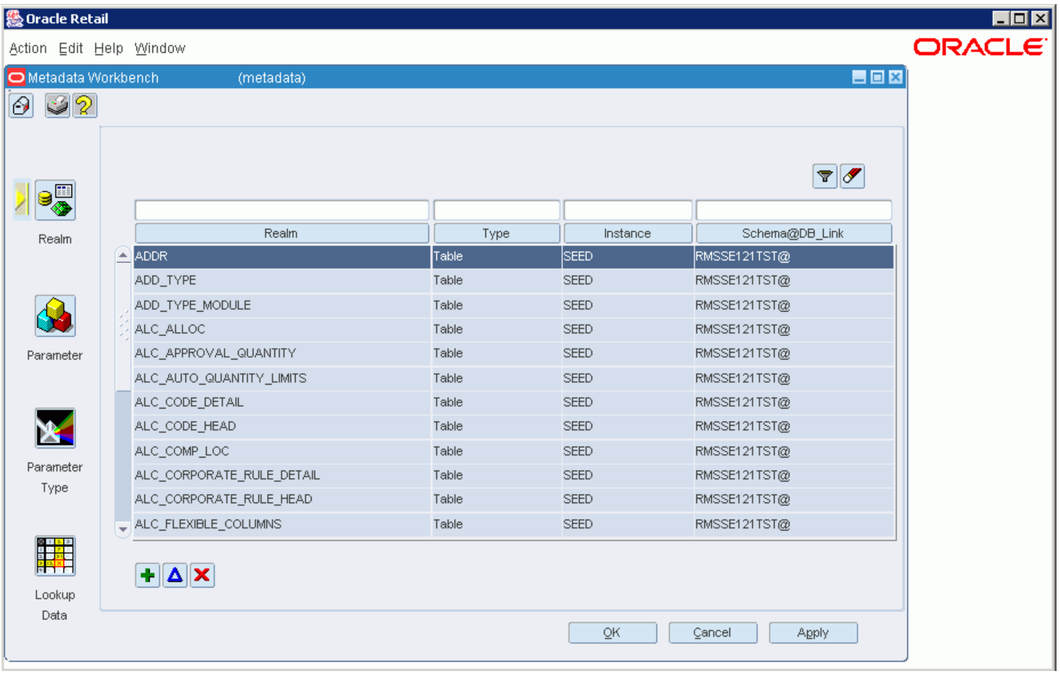
The Realm Maintenance screen is for editing the definition of a realm. This screen is accessed through several actions:



Deployment	Version	Deployed
WIN	4.5	□
WIN	5.0	□
WIN	6.0	□

Realm Maintenance Screen

1. Adding a realm by clicking the ✚ button on the Realm Find screen.



Realm	Type	Instance	Schema@DB_Link
ADDR	Table	SEED	RMSSE121TST@
ADD_TYPE	Table	SEED	RMSSE121TST@
ADD_TYPE_MODULE	Table	SEED	RMSSE121TST@
ALC_ALLOC	Table	SEED	RMSSE121TST@
ALC_APPROVAL_QUANTITY	Table	SEED	RMSSE121TST@
ALC_AUTO_QUANTITY_LIMITS	Table	SEED	RMSSE121TST@
ALC_CODE_DETAIL	Table	SEED	RMSSE121TST@
ALC_CODE_HEAD	Table	SEED	RMSSE121TST@
ALC_COMP_LOC	Table	SEED	RMSSE121TST@
ALC_CORPORATE_RULE_DETAIL	Table	SEED	RMSSE121TST@
ALC_CORPORATE_RULE_HEAD	Table	SEED	RMSSE121TST@
ALC_FLEXIBLE_COLUMNS	Table	SEED	RMSSE121TST@

Realm Find Screen

2. Editing a realm by clicking the △ button on the Realm Find screen.
3. Adding a realm by clicking the ✚ button next to the Realm Name field on the Parameter Maintenance screen.

Field Description

Realm Maintenance has the following fields:

Physical Name

The physical name by which the realm is named in the database. The Physical Name field must be unique and cannot conflict with any active realms.

Realm Type

The source of the realm. A realm type is a set of properties common for all realms of a particular type. The list displays the possible realm types.

There are currently four built-in action types.

- WINACT: Generic Windows action. This action type runs a Windows document. The full file system path of the document (including filename) should be stored in the FILE_PATH column on the REALM table. This document must have an extension that is registered with an application in the Windows Registry. Windows actions can only be run from Client/Server forms.
- WEBACT: Generic Web action. This action type runs a URL, which is stored in the URL column of the REALM table. In Client/Server forms, web actions are run in the default Web browser (the browser that is registered with the .htm/.html file extension). The URL should include the scheme (http:, ftp:, and so on) and hostname (for example, //www.yahoo.com), as well as the path and filename.
- FRMACT: Forms Action. This action type runs an Oracle Form. The name of the Form is stored in the physical name column. Note that this the internal name of the form, not the filename (.fmb). Forms Actions are associated with Forms Deployments (on the table FORMS_DEPLOYMENT). Each Forms Deployment represents a Version of Forms on the Web or on Client/Server. For each Forms Deployment that a Forms action can be run in, a record is inserted into the table ACTION_FORMS_DEPLOYMENT. Windows deployments cannot be run from Web Forms, so if a Forms Action does not have any Web deployments then it cannot be run from the Web.
- PLSQLA: PL/SQL Action. This action type runs a PL/SQL function or procedure on the database.

Instance ID

This field allows you to create more than one version of a realm across more than one database instance.

Database Link

The name for connecting to remote databases. It is available for external tables and views and all PL/SQL realm types.

Schema

The name of the schema that contains the object. The field is available depending on the realm type chosen.

Display Name

The name to be used for the realm on screen displays. The display name must be unique within active realms. If you leave this field blank, the value in the Physical Name is used for the display name.

Can Be Used on Multiselected Events

This checkbox is available only if the realm type is an action. It allows the action to be performed on multiple events at the same time.

File_path

Used for document actions. Stores the full file system path and name of the document associated with the action. The filename should have an extension corresponding to a registered document type in windows. Either this field or the URL field must be filled in for document actions.

URL

Used for document actions. Stores the URL associated with the action. Either this field or the File_path field must be filled in for document actions.

Database_string

Used for forms actions. Stores the database connect string used when running the forms action. If null then the form is run on the same database instance as Active Retail Intelligence.

Forms Deployment Information

This multirecord block is used for forms deployments. The first two fields contain the deployment type (Windows or Web) and forms version for each forms deployment, and are read-only fields. The third column, the available indicator, is a checkbox. This field is enabled for forms actions. Each row that is checked indicates that the forms action can be run in the corresponding deployment. At least one deployment must be selected for each forms action.

Adding a Realm from Realm Find Screen: Enabled and Required Fields

- When entering the screen, everything is disabled except for Realm Type and the Cancel button. Once you choose a realm type, all other fields are enabled. The Can be Used on Multiselected Events indicator is only available if the realm type is an action. If the realm type does not require a database link, then the Database Link field is disabled. For document actions, either the File_path or URL field must be filled in. For forms actions, the Database_string field must be filled in. For Forms actions at least one deployment must be selected.
- Clicking OK validates all data, posts the data to the Active Retail Intelligence database, and closes the Realm Maintenance screen. The Realm Find screen is displayed, showing the newly created realm in the realm list.
- Clicking OK+Repeat does the same things as OK, except that the Realm Maintenance screen is cleared, rather than exited, and everything except for Realm Type is disabled again.
- Clicking Cancel cancels your changes, closes the Realm Maintenance screen, and displays the Realm Find screen.

Editing a Realm from Realm Find Screen: Enabled and Required Fields

- When editing a realm, depending on the realm's type, fields are turned on or off as described in Adding a realm from the Realm Find screen. Realm Type, Instance ID, and the OK+Repeat button are never enabled.
- When the data is queried back, a set of package variables, corresponding to the fields listed below, is filled with the values from the fields. These variables will be used to determine whether anything has changed when the OK button is clicked.
- All items are still changeable, but the consequences of these changes must be assessed:

Item Changed	Action
Physical Name	Notify exceptions & events
Database Link	Notify exceptions & events
Schema	Notify exceptions & events

- When you click OK, the database compares the current values to the persistent package variables. If any differences are found, they are handled according to these rules: when the change you are making will affect some other part of the system, a warning is issued. If you decide to continue with the change, the changes are committed to the database. Otherwise, the operation is simply cancelled. The Realm Maintenance screen is then deactivated, and the Realm Find screen is displayed, showing the realm ID you were editing.
- Clicking Cancel cancels any changes you have made, closes the Realm Maintenance screen, and displays the Realm Find screen.

Adding a Realm from Parameter Maintenance Screen: Enabled and Required Fields

There are only a few differences between adding a realm from the Parameter Maintenance screen and the Realm Find screen:

- The OK < Repeat button is never enabled.
- After clicking OK or Cancel, the screen returns to Parameter Maintenance, not Realm Find.
- If the parameter on the Parameter Maintenance screen is set to use Auto Discover, a realm name may have been passed in. If this has happened, the Physical Name is filled in, and the Display Name is defaulted to the same thing. The Display Name is not editable until a Realm Type is chosen. Physical Name is never editable.

Validation Sequences Performed on this Screen

Several validation sequences occur on this screen:

- Whenever you change the realm type, items are turned on and off appropriately, as described in the discussions of required and enabled fields. If any fields are deactivated, they are cleared as well.

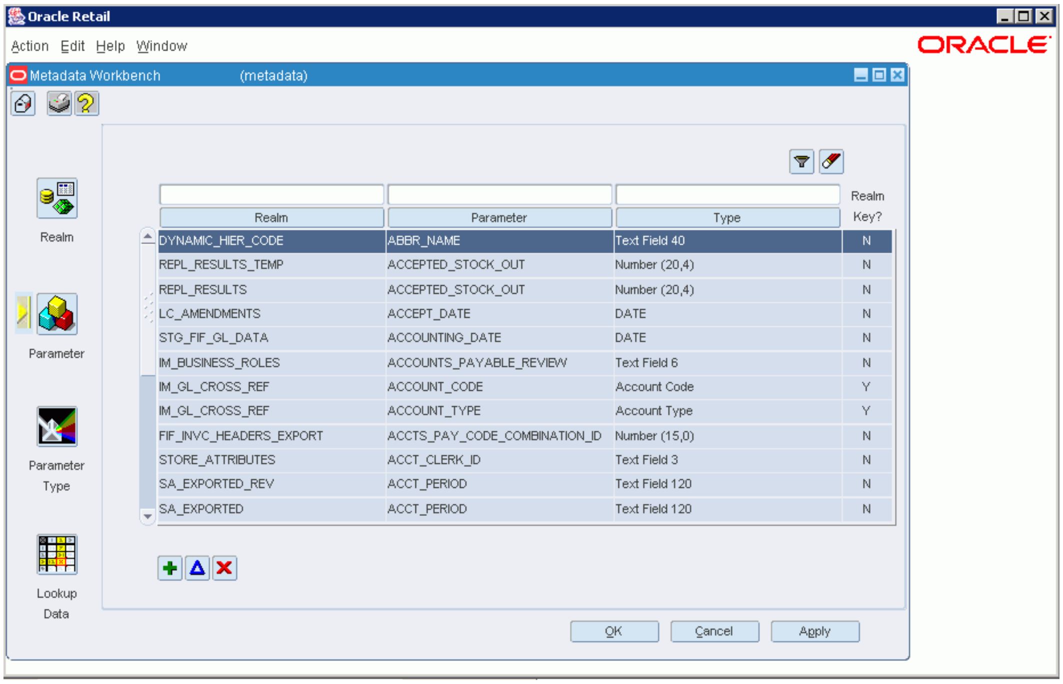
- The Physical Name field is checked to make sure that the name does not conflict with any active realms. If it conflicts with an inactive realm, the sequence number is incremented to one higher than the last version. Otherwise, the sequence number is set to 1. If it conflicts with an active realm, an error is raised. If the Display Name field is empty, and the realm isn't an action, or a function, it is set to the same value as the Physical Name field.
- The Display Name field is checked to make sure that no active realms with the same display name exist already. If the name already exists, an error is raised.

Change Effect Warning Dialog [metadata]

This dialog displays which other objects will be impacted by the current change you are making to metadata. You can choose whether you want to continue with the change or cancel it.

Parameter Find Screen

The Parameter Find screen lists all the parameters defined in your Active Retail Intelligence system, by parameter name, parameter type, and the realm to which the parameter belongs. This view is accessed by clicking the button on the tab bar. When you first open the Metadata Maintenance form, this is the first view displayed.



Realm	Parameter	Type	Realm Key?
DYNAMIC_HIER_CODE	ABBR_NAME	Text Field 40	N
REPL_RESULTS_TEMP	ACCEPTED_STOCK_OUT	Number (20,4)	N
REPL_RESULTS	ACCEPTED_STOCK_OUT	Number (20,4)	N
LC_AMENDMENTS	ACCEPT_DATE	DATE	N
STG_FIF_GL_DATA	ACCOUNTING_DATE	DATE	N
IM_BUSINESS_ROLES	ACCOUNTS_PAYABLE_REVIEW	Text Field 6	N
IM_GL_CROSS_REF	ACCOUNT_CODE	Account Code	Y
IM_GL_CROSS_REF	ACCOUNT_TYPE	Account Type	Y
FIF_INVC_HEADERS_EXPORT	ACCTS_PAY_CODE_COMBINATION_ID	Number (15,0)	N
STORE_ATTRIBUTES	ACCT_CLERK_ID	Text Field 3	N
SA_EXPORTED_REV	ACCT_PERIOD	Text Field 120	N
SA_EXPORTED	ACCT_PERIOD	Text Field 120	N

Metadata Maintenance Form

Field Description

The Parameter Find view has the following fields.

For the data displayed in columns, you can sort the data by clicking the column headings, and sort in reverse order by clicking the column headings again. You can also use the filtering fields and buttons to filter the display of parameters according to the filtering criteria entered in the fields.

Parameter

The name of the parameter. This field is read-only.

Type

The name of the parameter type to which the parameter belongs. This field is read-only.

Realm

The name of the realm to which the parameter belongs. This field is read-only.

Realm Key?

Indicates whether the parameter is part of the key for the realm to which the parameter belongs. This field is read-only.

Sorting

You can sort the realms by the categories of information by clicking the column heading above each column.

Filtering

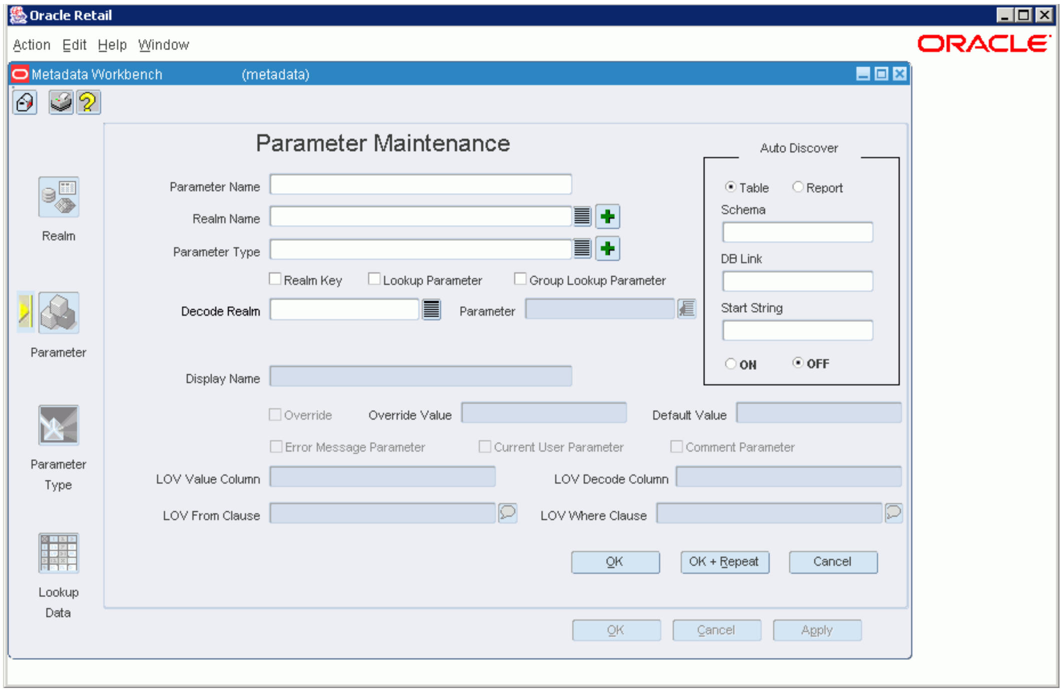
You can filter the view or realms by filling in the fields above each column and clicking the ▽ button. The ✐ button clears any active filter.

Adding, Deactivating and Editing Parameters

The three buttons at the bottom of the list are for adding, deactivating, and editing parameters.

Add Button

The ✚ button adds a parameter. The Parameter Maintenance screen is displayed, where you can add the new parameter.



Parameter Maintenance Screen

Delete Button

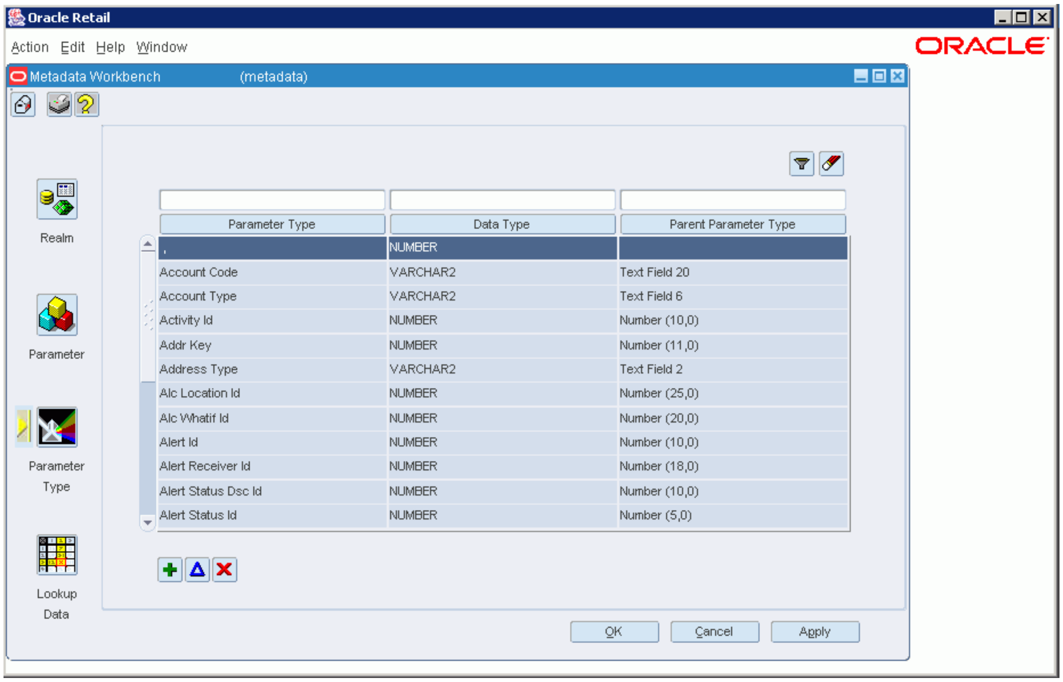
The ✖ button deactivates the selected parameter. You are notified of any impact of deactivating the parameter and asked whether you want to proceed.

Edit Button

The △ button edits the selected parameter. Displays the Parameter Maintenance screen.

Parameter Type Find Screen

The Parameter Type Find screen displays all active parameter types, or parameter types currently defined in your Active Retail Intelligence system. This screen is accessed by clicking the button on the Metadata Maintenance form.



Parameter Type	Data Type	Parent Parameter Type
,	NUMBER	
Account Code	VARCHAR2	Text Field 20
Account Type	VARCHAR2	Text Field 6
Activity Id	NUMBER	Number (10,0)
Addr Key	NUMBER	Number (11,0)
Address Type	VARCHAR2	Text Field 2
Alc Location Id	NUMBER	Number (25,0)
Alc Whatif Id	NUMBER	Number (20,0)
Alert Id	NUMBER	Number (10,0)
Alert Receiver Id	NUMBER	Number (18,0)
Alert Status Dsc Id	NUMBER	Number (10,0)
Alert Status Id	NUMBER	Number (5,0)

Metadata Maintenance Form

Field Description

Information displayed on this screen includes the parameter type ID, parameter type name, data type, and parent parameter type name, displayed in a table of columns. All three displayed columns are sortable by clicking the sort buttons above each column. In addition, all three displayed columns are filterable by filling in the fields above each column and clicking the button.

Parameter Type

The name of the parameter type. This field is read-only.

Data Type

The name of the parameter type's data type. This field is read-only.

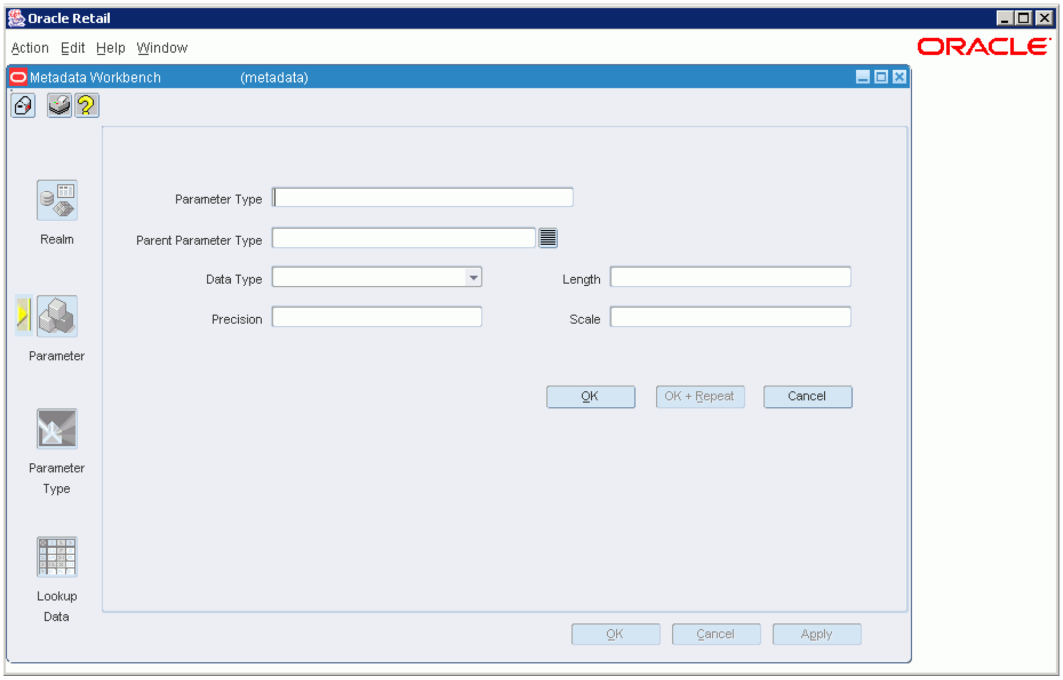
Parent Parameter Type

The name of the parameter type's parent's name. This field is read-only.

Buttons

Add Button

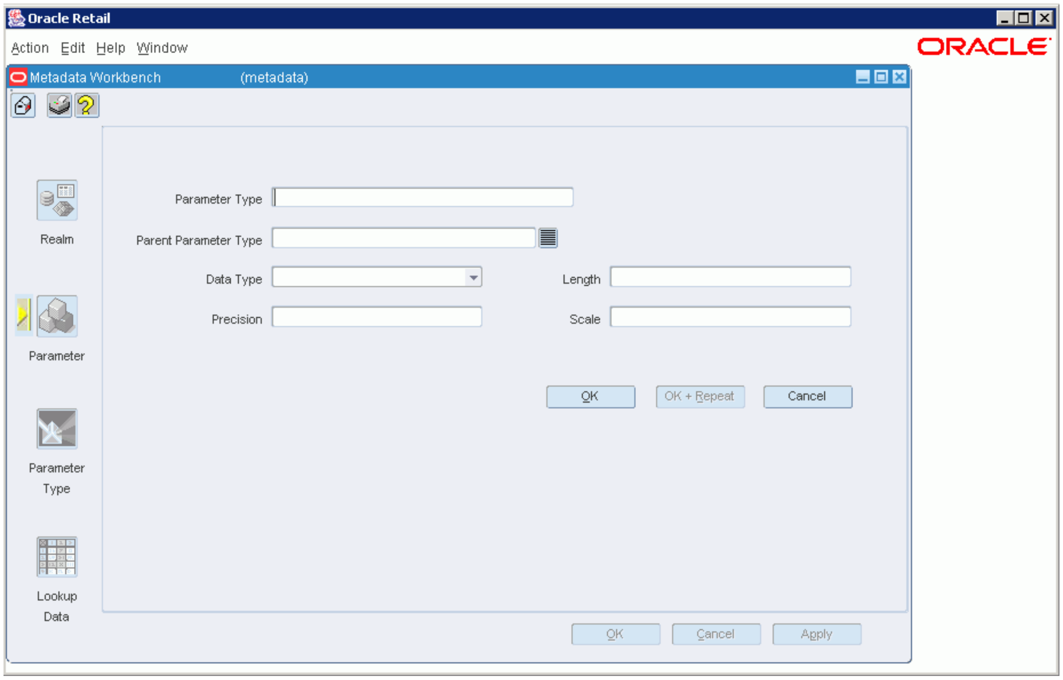
The ✚ button adds a new parameter type. The Parameter Type Maintenance screen is displayed.



Parameter Type Maintenance Screen

Edit Button

The △ button edits the selected parameter type. The Parameter Type Maintenance screen is displayed.

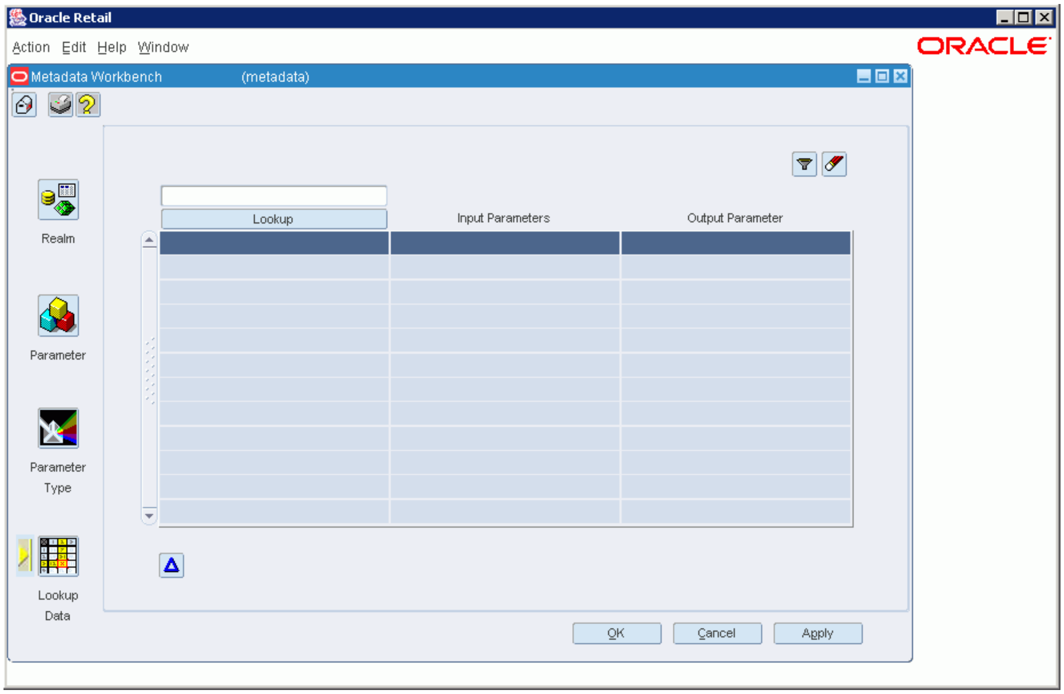


Delete Button

The ✕ button deletes the selected parameter type.

Lookup Find Screen

The Lookup Find screen lists all of the active realms that are lookups on the Active Retail Intelligence system. This screen is accessed by clicking the ▦ button on the Metadata Maintenance form. From this screen, you can view and edit lookups on the Lookup Data Edit screen.



Lookup Find Screen

Field Description

Lookup

This column displays the text of the lookup. This column is sortable by clicking the sort buttons above it. You can filter the view of the Lookup column by filling in the field above it and clicking the ▽ button. To clear an active filter, click the ✎ button. Although the filter field is editable, the list of lookups in the column is read-only. To edit a lookup, select it and click the △ button. The Lookup Data Edit screen is displayed.

Input Parameters

This column displays a list of each lookup’s input parameter set. This set is comprised of the parameters’ names concatenated together, with semicolons separating them.

Output Parameter

The output value column displays a list of each lookup’s output parameter.

Lookup Data Edit Screen

The Lookup Data Edit screen displays the input values and output values for lookups in your system and allows you to edit them. There are two parts to the Lookup Data Edit screen, described below as the Top Half and the Bottom Half.

Field Description - Top Half: Existing Input and Output Values

The top part of the screen displays all the existing input and output values for the lookup. All of the inputs for the retrieved output are retrieved and strung together in the same way that the input parameters were in the header; that is, delimited by semicolons and ordered by parameter ID. Filling in these values may take a significant amount of time. If the output sequence number is 0, this means that this output is the default value, so the input values field for this record will be "Default Value".

When this screen is displayed, a check is made to make sure that for every parameter in the lookup, there is a parameter of the same parameter type with a lookup indicator on 'Y'. If this check fails, then all buttons except for Cancel are disabled and a message is displayed stating that because the form is unable to validate all values, you may not enter any more data until the situation is fixed.

Add Button

The ➕ button adds a new set of input and output parameters and values. The bottom half of the screen is enabled for entering the values.

Edit Button

The △ button edits a selected set of input and output values. When both creating and editing, the top of half of the screen is disabled and the fields in the bottom half of the screen are enabled.

Delete Button

The ✖ button deletes a selected set of input and output values.

OK

Saves changes to the input and output values and closes the screen.

Cancel

Cancels any edits and closes the screen, returning to the Lookup Find screen.

Field Description - Bottom Half: New and Edited Input and Output Values

The bottom half of the screen is for entering new input and output values for the lookup, or editing existing input and output values. Clicking the ➕ button or the △ button moves the cursor to this part of the screen.

Input Parameter

The name of the input parameter.

Value

The value of the input parameter. To display a list of valid values for a parameter, click the LOV button in the value column.

Output Parameter

The name of the output parameter.

Value

The value of the output parameter. To display a list of valid values for a parameter, click the LOV button in the value column.

Apply

Verifies and applies the changes you have made in the lookup editor part of the screen. The input and output value fields are checked to make sure they are not blank, and the database is queried to make sure that the input value set is unique. Control is returned to the upper part of the screen.

Cancel

Cancels any changes to the lookup’s input and output parameters and returns control to the upper half of the screen.

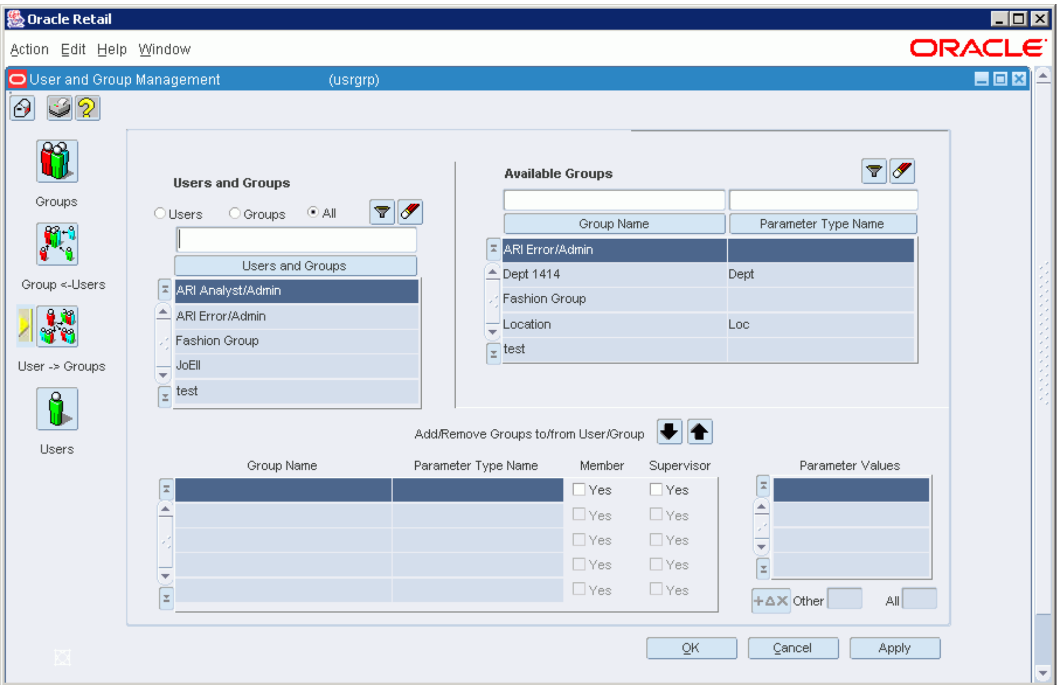
3

ARI User and Group Administration

Setting up Active Retail Intelligence in your business involves defining users and user groups to which events will be routing. In its role as a business rule monitoring tool and messaging/workflow system, Active Retail Intelligence routes event to particular users of the system to be dealt with. To add flexibility to this routing (and to provide a framework for other aspects of the application such as security), users are consolidated into groups. These groups can be set up to mirror aspects of a company's business model, with groups reflecting job functions, for example, Buyers and Invoice Matching Teams. Groups can also be made members of other groups, allowing for the creation of multi-layered group hierarchies to reflect more complex organizations (for example, employee supervision).

The Users and Groups Dialog

Users and groups are defined and maintained through the Users and Groups dialog. This dialog is accessed by choosing Setup+Users and Groups from the Oracle main menu. There are several main aspects to the Users and Groups dialog:



Users and Groups
ARI Analyst/Admin
ARI Error/Admin
Fashion Group
JoEll
test

Group Name	Parameter Type Name
ARI Error/Admin	
Dept 1414	Dept
Fashion Group	
Location	Loc
test	

Users and Groups Screen

- User setup and maintenance
- Group setup and maintenance
- User membership: Mapping a single user or group into multiple groups
- Group membership: Mapping multiple users or groups into a single group
- Maintaining user preferences

Users

Users correspond to actual individuals working in the Active Retail Intelligence system. Oracle user functionality tracks system users. User definitions extend this Oracle user functionality by tracking additional user attributes and preferences, such as the user to whom events should be forwarded in the event that a user is unable to receive alerts, for example, out of the office on vacation.

User Attributes

A user definition is maintained in a table, and includes such attributes as

- The name of the user.
- The name of the user to receive this user's alerts when this user is inactive.
- Notification Information: Miscellaneous information on external notification options, including numbers and addresses for notifying the user via e-mail, fax, and pager.
- List of Group Memberships: All of the groups to which the user belongs.
- Status (Active/Inactive):Whether the user is active or inactive, that is, available or unavailable for event routing in Active Retail Intelligence. A user can be deactivated at any time. If the user has any active events assigned at the time of deactivation, the user is given the option to forward those events on to other users.
- A target user for any automatically forwarded events.

Groups

A group is used to combine users based on some common characteristics. The most typical use for groups is to represent aspects of the company's business model that need to be mapped back to individual users. For example, you can define groups to reflect employee work areas, such as store or department, or to reflect job roles such as buyer roles, or invoice matching teams.

Groups and Event Routing

Within Active Retail Intelligence, user groups are used to route events. A user group or groups can be linked with event conditions, so that when the event conditions are met, the routing process uses that group or groups to build a list of users eligible to receive the event.

This routing group can be setup as either the intersection or the union of all groups associated with the event condition. The intersection or union is taken over all users who are members of those groups either directly or through nested group membership.

From this routing group a user or set of users will be selected for the event assignment, in accordance with the routing rules associated with the event, such as workload-based distribution, random or even-based distribution, or distribution to all group members.

A user group can have a parameter associated with it, and the resolution of state rules may resolve to several groups.

An example of how this routing could work is a group is set up using the store parameter and another just of buyers. By assigning to both the store parameter is evaluated and only users associated with that parameter in the store group and also in the buyer group will fill the resultant routing set, so store buyers are effectively found specific to the instance.

Group Attributes

Each group definition has several attributes:

- Name: The name of the group:
- List of Members: A list of the users and groups that belong to this particular group.
- Group Type: Whether the group is a simple group or parameter-driven group, described below.

For parameter-driven groups only, there is an additional group attribute, Parameter Type, which is the parameter type associated with the group.

Group Types

There are two types of groups: parameter-driven groups and simple groups.

Parameter-Driven Groups

Parameter-driven groups are groups where membership is driven by a specific parameter value. These group types are used to simplify the assignment of users into what would otherwise be extremely large numbers of group specifications. For example, an assignment may be made to a Store group, driven by a Store Number parameter. Subsequently, when users were made members of the group, they would be assigned a value (or values) for the parameter.

To illustrate, Joe Smith could be made a member of the Store group with a Store Number of 1001. This group assignment means that Joe Smith is eligible to receive events that were routed to the Store group with parameter value 1001. This grouping not only saves the effort of defining a separate group for each store, but it also allows Active Retail Intelligence to route events to a specific users within a group based on the value of an event parameter. For example, a quality control event could be assigned to the Store group, and when a shipment requiring quality control was received it would be routed only to those members of the store group whose Store Number matched that of the store receiving the merchandise.

The parameters used to drive these groups must be of either a character, or a discrete numeric data type. Parameters such as date, or currency values, cannot be used to drive a group since inexact comparisons (>, <, between, and so on) are not possible within the group resolution, although they can be part of Active Retail Intelligence event rules.

Parameter based groups can also be set to use exclusion values, that is, all values except some specified. These exclusion values do not contradict other values a user may have assigned through other group membership: they are just a convenient way to choose most, but not all, values. All values are also an option. Composite groups can also be created, though they can contain at most one group that requires a parameter.

Simple Groups

Simple groups are groups that are not driven by any parameters. Simple groups can be based on values that align with aspects of a company's business model. Such groups may be location based, for example, Northeast, Midwest, Southeast, or function-based, for example, Buyer or Sr Buyer. Simple groups are purely structural, with no internal routing mechanisms.

Making Users and Groups Members of Another Group

Users and groups can be connected by making a user or group a member *of* another group. However, a user or a group cannot be made a member of another user. When a user or a group is made a member of a parameter-driven group, values must be specified for the driving parameter. This will allow for parameter-driven routing within the group. Parameter driven groups cannot be made a member of another group. This ability to nest groups within groups allows for the definition of group hierarchies that can parallel a specific business model, or provide a more general grouping.

When a user or group is added to another group, it must be specified whether the user or group belongs to the parent group in a membership capacity, a supervisory capacity, or both. A user that belongs to a group in a membership capacity has all roles and responsibilities associated with that group. Within Active Retail Intelligence a member of a group is able to receive any events routed to that group. A user that belongs to a group in a supervisory capacity is not necessarily given the responsibilities associated with that group, but is able to monitor activity within the group. For example, within Active Retail Intelligence supervisors of groups will not be eligible to receive events associated with that group, but will be able to oversee any events routed to that group. A group can be designated as supervising another group by making it a member of the other group in a supervisory capacity. However, it is important to note that if the supervisory capabilities are to propagate to the users assigned to that group, the users must also be mapped to the supervisory group in a supervisory capacity. This will allow system administrators to use the same group hierarchies for both event routing and supervisory situations, or to set up completely separate group structures for the two purposes.

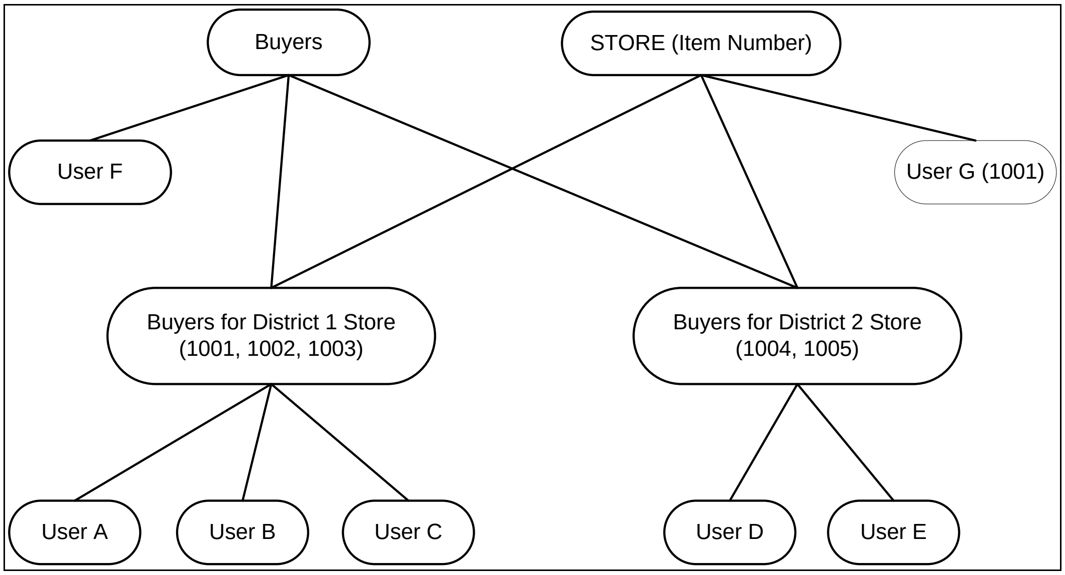
Example Groups

The following are examples of group structures:

Location-Based Group

In this strictly location-based hierarchy, there are four groups, and five users.

The Store group is a parameter driven group, driven by store number. The other groups (District 1 Stores, District 2 Stores, and Central Region Stores) are all simple groups. The District 1 Stores and District 2 Stores groups are members of the Store group, with associated parameter values, while the Central Region Stores group is a member of both the District 1 Stores group and the District 2 Stores group. User A is a member of the Store group directly, with a parameter value of 1001. Users B and C are members of the District 1 Stores group, making them indirectly members of the STORE group with parameter values 1001, 1002, and 1003. Similarly, User D is a member of the District 2 Stores group, and User E is a member of the Central Region Stores group.



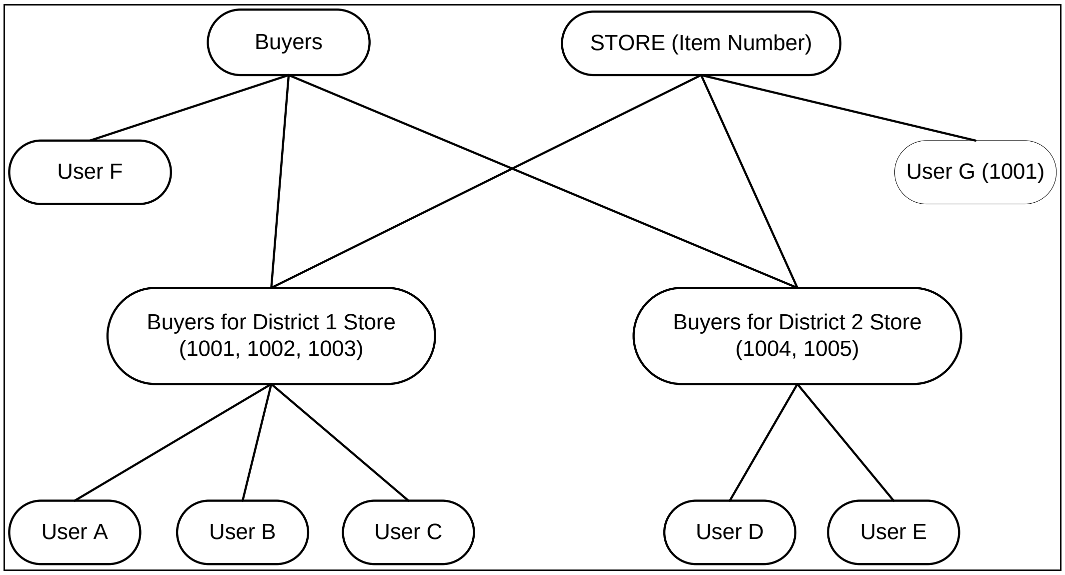
Location-Based Group

This group model allows us to access the data in a number of ways. In the case of event routing, we can start with the routing group and trace through the tree to find all the users eligible to receive that event. For example, if an Out of Stock event occurred at Store 1001, the event could be routed to User A directly; Users B and C through the District 1 Stores group, or to User E through the Central Region Stores group. In the case of event supervision (considering all relationships to be supervisory for this example), we can start with a particular user and trace through the tree in the opposite direction to find which events the user is eligible to supervise. For example, User A would be able to supervise events routed to Store 1001, where User D would be able to supervise events routed to Stores 1004 and 1005. User E would have the ability to supervise the stores in either the District 1 Stores or the District 2 Stores groups through supervision of the Central Region Stores Group.

Location and Business Role-Based Group

In the group configuration, we have set up groups to encompass both location and business role functionality.

Again we have our store group, driven by a store number, and we also have a simple group called Buyer. We can then create two other simple groups, Buyers for District 1 Stores and Buyers for District 2 Stores that are members of both the Store and Buyer groups, thereby combining the aspects and privileges of those groups. In this situation one might route an event to both the Buyer and the Store groups. In this case the set of users eligible to receive the event instances would be the intersection of the Users assigned to the Buyer group and the Store group. This would mean Users A, B, and C would be eligible for any events whose Store parameter value was 1001, 1002, or 1003, and Users D and E would be eligible for events whose Store parameter value was 1004 or 1005. Users F and G would not be eligible for any events of this type, since the event is routed to both the Store and the Buyer groups, and they are not members of both groups.



Role Based Group

Supervision and Groups

It is possible that one group can supervise another; however, because routing groups are determined dynamically, supervision can be elusive, so it may be helpful to assign an entirely distinct set of groups to form a supervision hierarchy. Supervision works such that a supervisor can view all of the events owned by a supervised user. Alternatively, without assigning supervision rights, an Active Retail Intelligence supervisor user attribute allows supervisors to view all events whose signature is contained in the union of all possible event signatures that could have been routed to the supervisor.

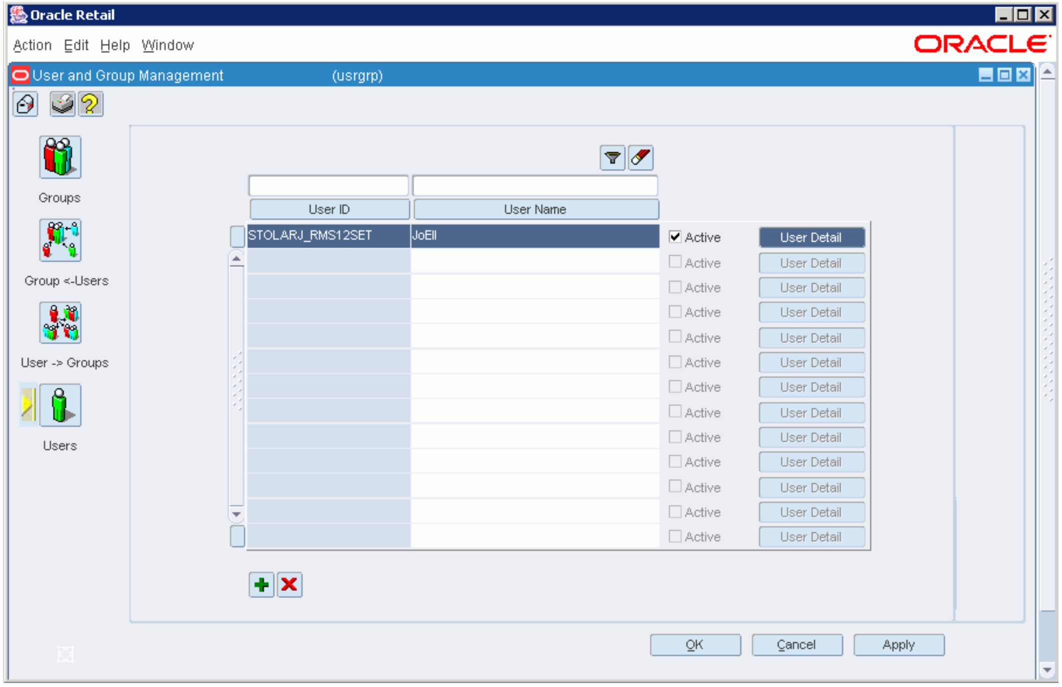
Security Considerations

The previous release of Active Retail Intelligence attempted to implement application level security that allowed supervisors to add and remove users from a user group, but this left a security gap allowing a user-level supervisor to elevate his rights by adding more powerful users to the supervised role. Application security is not implemented elsewhere in the enterprise application at lower than menu level, so it is being removed here for consistency. Any user with access to the user group and user forms can add and remove users from user groups as appropriate.

Windows and Dialogs

User Setup/Maintenance Screen

The User Setup/Maintenance screen displays all users currently defined in the Active Retail Intelligence system. From this screen, you can add and delete users and access the detailed user definition information for users.



User ID	User Name		
STOLARJ_RMS12SET	JoEll	☑ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail

User Setup/Maintenance Screen

Field Description

User ID

The user group ID for the user, which must be a valid Oracle user ID. This field is enabled only for user records that have not been posted to the Active Retail Intelligence database. It is not enabled for existing users.

User Name

The screen name for the user. The user name must be unique in the Active Retail Intelligence system.

Status

Identifies whether the user is active or inactive. This field is always enabled.

User Detail

Displays the detailed user definition for the selected user.

Add Button

The ✚ button Adds a user to the list.

Delete Button

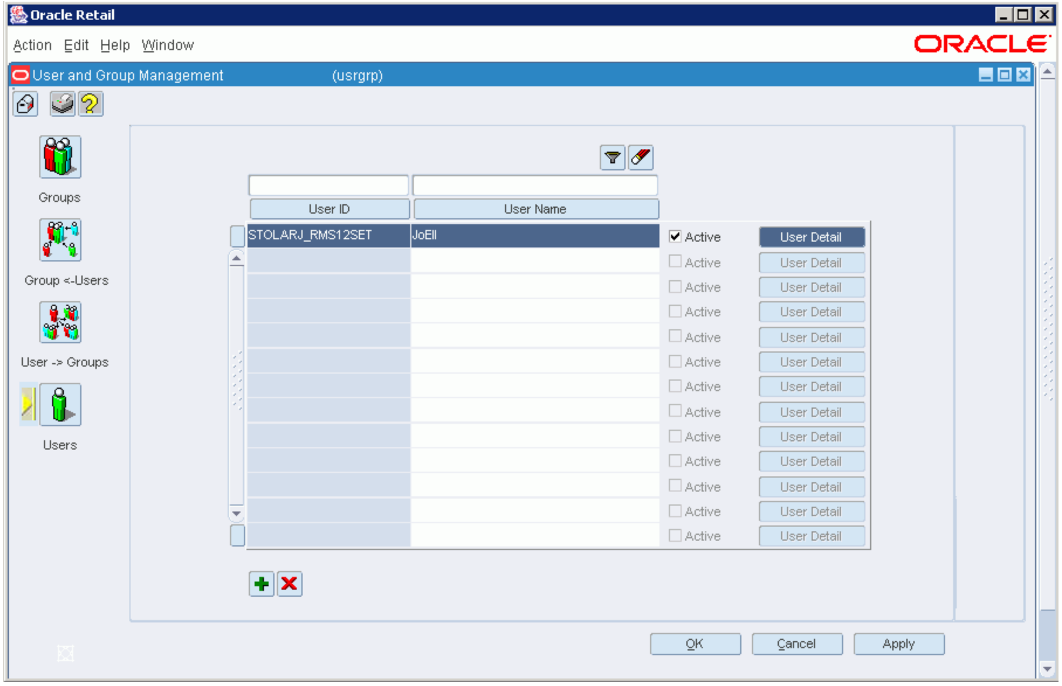
The ✖ button Deletes the selected user from the list. This button is enabled only if the user does not own any active events. If any other users have specified the user you are trying to delete as an auto-forwarding target, a warning is displayed, and those users will lose their auto-forwarding specifications.

Filtering and Sorting

Entering values into the fields above the User ID and User Name columns and clicking the ▼ button filters the records in the user list. Clicking the ✎ button clears the filter criteria and removes the filter from the user list. The user list can be sorted by any of the displayed columns by pressing the button above the column. Pressing the same button multiple times sorts the user list in the reverse order.

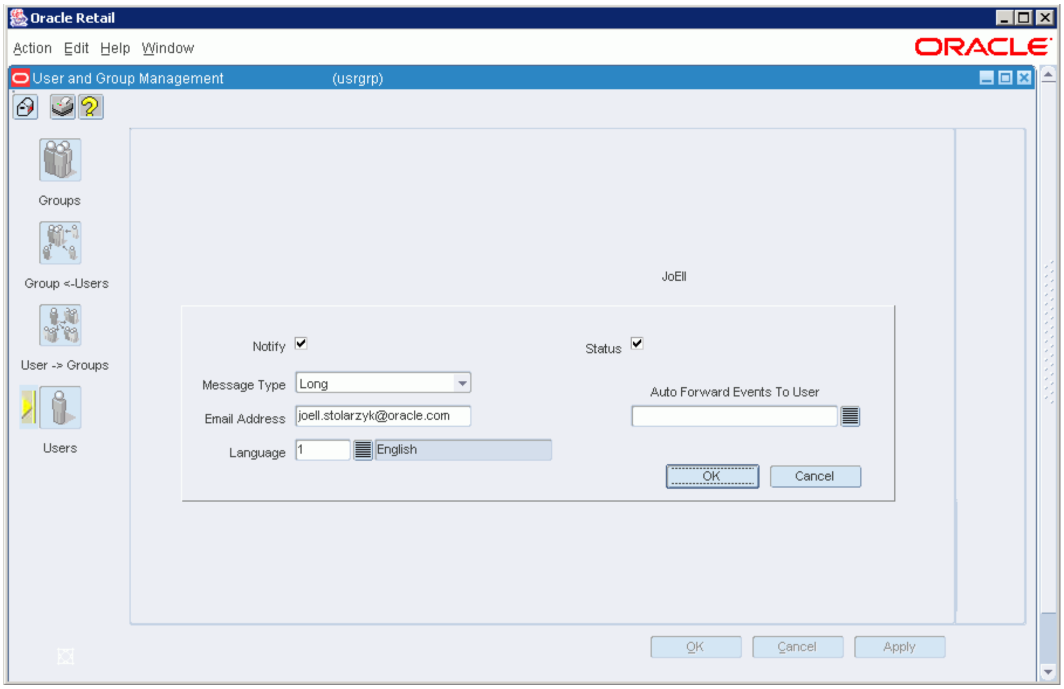
User Details Screen

The User Details screen contains detailed user definition information for an Active Retail Intelligence user. If you are an Active Retail Intelligence administrator, this screen is accessed by clicking the User Detail button for a particular user on the Users screen. If you are an Active Retail Intelligence user, this screen is automatically displayed, with your current user details, when you select Control+Setup+Users and Groups from the RMS main menu.



User ID	User Name		
STOLARJ_RMS12SET	JoEll	☑ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail
		☐ Active	User Detail

User and Group Management Screen



JoEll

Notify ☑

Status ☑

Message Type: Long

Email Address: joell.stolarzyk@oracle.com

Language: 1 English

Auto Forward Events To User:

User Details Screen

Field Description

User Name

The screen name of the current user. This field is read-only.

Notification Checkbox

Indicates whether the user wants to receive email notification of their events. Note that even if the user selects email notification (which is in addition to normal Alert notification), email will be sent only for events that are set up to send email.

Notification Address

The email address for the current user. This field is editable at all times.

Notification Type

The type of email notifications the user would like to receive. Notifications can be either Short or Long. The Short version is kept under 100 characters, and is recommended for sending notifications to cell phones and other embedded devices.

Language

The user's language preference for the entire system. Clicking the list of values button displays a list of all valid languages in the system. Leaving this field blank indicates that the system's primary language will be used.

Status

Indicates whether the user definition is active or inactive. An active user is currently in the office, and therefore eligible to receive events. A user can also be deactivated, which means that the user is not eligible to receive alerts. If the user has any active events assigned at the time of deactivation, the events can be automatically forwarded to another user by specifying a user in the Auto Forward Events to User field.

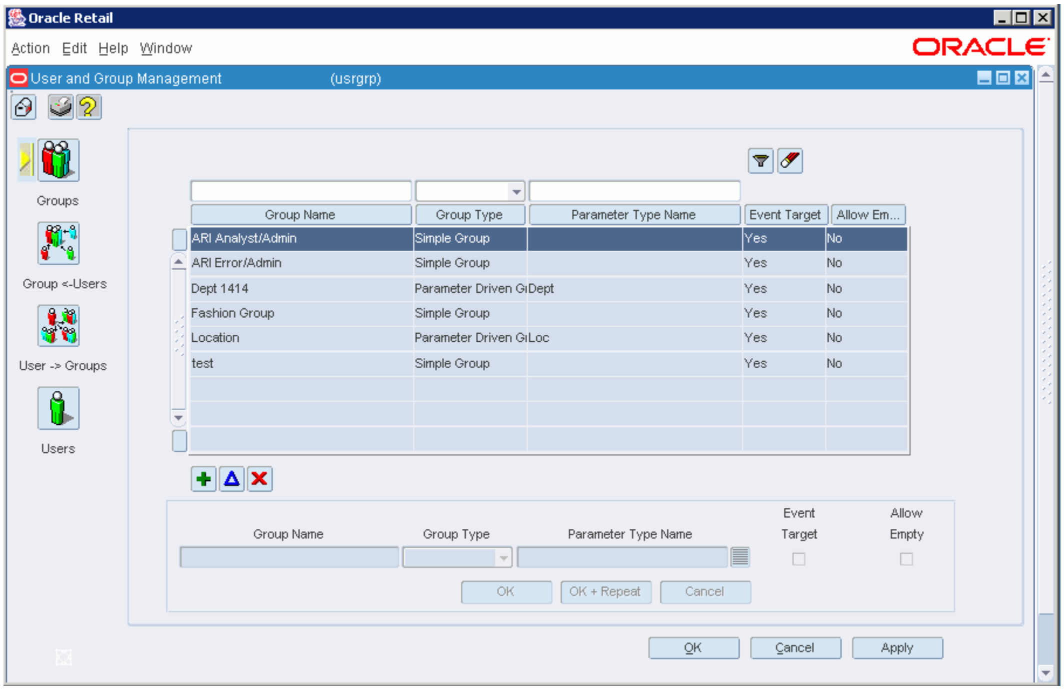
While a user is inactive, it will be unable to receive any events, either new or forwarded. Any events routed to the inactive user are re-routed to the user specified by the forward_to_user attribute. When you deactivate a user, the system is checked to determine whether the user has any active events. If the user has active events, a warning is displayed.

Auto Forward Events to User

Displays the automatic forwarding target, or the name of the user to receive this user's alerts when this user is inactive. Clicking the list of values button displays all users in the system, except the current user.

Group Setup/Maintenance Screen

The Groups screen is for setting up and maintaining user groups.



Group Name	Group Type	Parameter Type Name	Event Target	Allow Em...
ARI Analyst/Admin	Simple Group		Yes	No
ARI Error/Admin	Simple Group		Yes	No
Dept 1414	Parameter Driven G	Dept	Yes	No
Fashion Group	Simple Group		Yes	No
Location	Parameter Driven G	Loc	Yes	No
test	Simple Group		Yes	No

Group Name	Group Type	Parameter Type Name	Event Target	Allow Empty
			☐	☐

Groups Screen

Field Description

Group List

The top portion of the Groups screen is a list of all groups defined in the system in a read-only list. Below the list is an editor in which you can add and edit group definitions.

Filtering and Sorting

The group list can be filtered by entering values into the fields above the columns and clicking the ▼ button. Clicking the ✎ button clears any values from the filter fields and removes the filter from the group list. The group list can be sorted by any of the displayed columns by pressing the button above the column. Pressing the same button multiple times sorts the list in the reverse order.

The ✚ button adds a new group. The cursor will be positioned in the first field of the group editor, Group Name.

Pressing the Add Group (green plus symbol) button will enable the lower block of items, and disable all other action buttons displayed in the form. The user can then enter information into the fields.

Group Name

A unique name by which the group is known in the Active Retail Intelligence system. This field is required.

Group Type

Identifies whether the group is a simple group or a parameter-driven group.

Parameter Type

For parameter-driven groups, this field specifies the parameter type. This field is required for parameter-driven groups, and must be left blank for simple groups. Clicking the list of values button displays a list of valid parameter types from which you can choose. A valid parameter type has a group lookup parameter associated with it. If you enter a value that does not correspond to exactly one parameter type ID, a warning is displayed, prompting you to use the LOV to select a value. Any values entered into this field will restrict the values displayed in the LOV.

Event Target

Indicates whether the group may be used as a target for assigning events. The default setting is Yes (checked).

Allow Empty

Identifies whether it is valid for the routing group to be empty when an event is assigned to this group. This could happen as a result of a simple group having no members, a parameter-driven group having no member associated with a parameter value, or if the routing group is the intersection of multiple groups, of groups that have no members in common. If Yes, then the event will not be assigned to any user. If No, then the event will be assigned to the error group. The default setting is No (unchecked). This field is available only when the Event Target field is set to Yes.

Buttons

Edit Button

The △ button edits the selected group. The group's definition is displayed in the group editor at the bottom of the screen. The user group name field is always editable, the event assignment target checkbox is only editable if Active Retail Intelligence is installed and no events have been assigned to the group. The group type and parameter type fields are only editable if no events or users (or groups) have been assigned to the group.

Delete Button

The ✖ button deletes the selected group. This button is enabled only for groups that do not have events currently being routing to them. A check is made to determine if the any users (or groups) have been assigned to the selected group. If no users (or groups) have been assigned, the group is deleted. If users (or groups) have been assigned to the selected group, a warning is displayed. If you choose to continue, the group and all mappings to that group are deleted.

Users and Groups Dialog [usergrp]

Users and groups are set up and maintained through the Users and Groups dialog.

If you are an Active Retail Intelligence administrator, you can access all the screens of this dialog to perform these tasks.

From this dialog, you can perform these tasks:

- Set up and maintain user definitions
- Set up and maintain user groups
- Assign a user to multiple user groups
- Assign multiple users or groups to a single group
- Maintain user preferences

If you do not have Active Retail Intelligence administrator user privileges, you only have access to the User Details screen, with your own user details displayed.

Buttons

When the Users and Groups dialog is first opened, the Group Setup and Maintenance screen is displayed. You can access other screens of the Users and Groups dialog by clicking the buttons on the left side of the dialog.

Groups

The button displays the Group Setup and Maintenance screen where you can add, edit, and delete user groups.

Groups – Users

The button displays the User Membership screen, where you add and change group assignments for a user.

User – Groups

The button displays the Group Members screen, where users and groups can be added or removed from the selected group.

Users

The button displays the User Setup/Maintenance screen, where you can add, edit, and delete user definitions.

OK

Commits changes to the Active Retail Intelligence database and closes the Users and Groups dialog.

Cancel

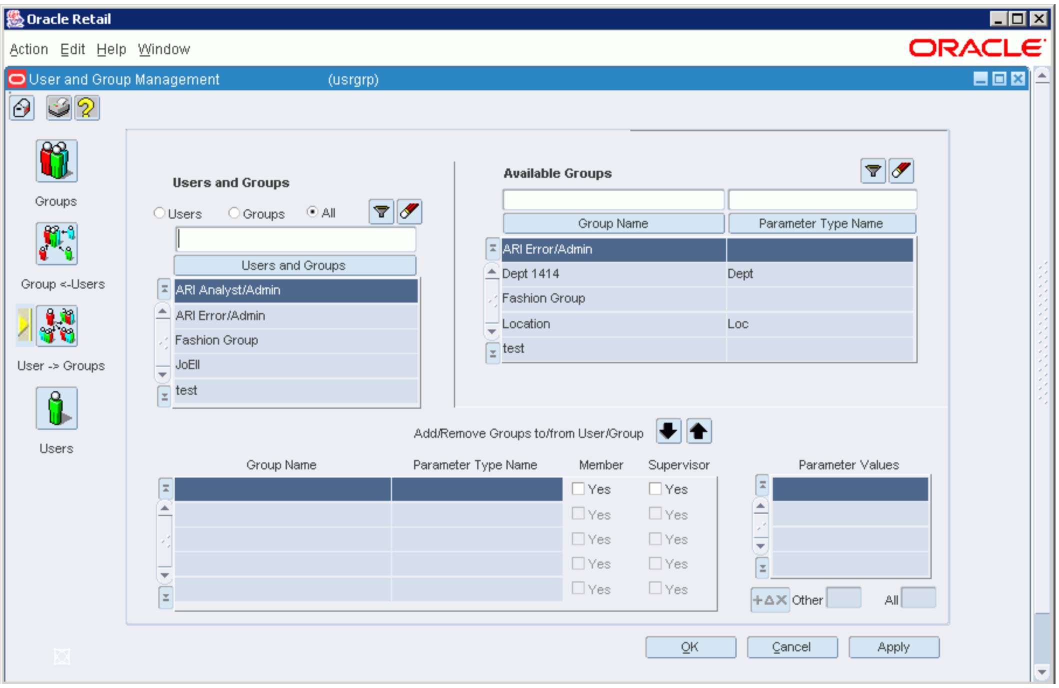
Closes the Users and Groups dialog without saving any changes.

Apply

Commit changes to the Active Retail Intelligence database but keeps the Users and Groups dialog open.

User Membership Screen

The User Membership screen is one of two screens used to link users and groups with groups. In the User Membership screen, you first select a user or group, and then all of the groups to which the user belongs are displayed in the bottom part of the screen. You can then edit the user membership, for example, add or remove groups from the selected user.



Users and Groups
ARI Analyst/Admin
ARI Error/Admin
Fashion Group
JoEll
test

Group Name	Parameter Type Name
ARI Error/Admin	
Dept 1414	Dept
Fashion Group	
Location	Loc
test	

Group Name	Parameter Type Name	Member	Supervisor
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes

Parameter Values

User Membership Screen

Field Description

User/Group List

In this part of the screen, you can view users, groups, or users and groups, by clicking the radio buttons located above the list. Select a user or group from the list. The groups which the chosen user or group is a member of will be displayed in the lower part of the screen. Filtering and sorting buttons are available to aid your search on the users and groups list.

Member of (Group List for Selected User)

Displays the groups to which the selected user belongs.

Parameter Values

Displays any parameter values associated with a particular group.

Available Groups

The upper right part of the screen displays all the groups to which the selected user does not belong, along with the parameter type associated with the group. You can use the ⬆ button to add the user to a selected group.

Up Arrow and Down Arrow Buttons

The ⬆ button removes the selected group from the Members list and returns it to the Available groups list. The ⬇ button adds the group from the Available groups list to the Member of list, making the user a member of that group.

Predefined User Groups

Operating Active Retail Intelligence requires two predefined user groups, the Active Retail Intelligence Analyst/Administrator Group and the Active Retail Intelligence Error/Administrator Group.

These user groups exist for Active Retail Intelligence administration and Active Retail Intelligence error processing. The administrative user group is notified when an event process enters an infinite reevaluation/action loop or can be used as a user-group assignment target for events that require administrative access for any kind of self-monitoring rules that may be setup for Active Retail Intelligence. Likely a client using menu level security would allow Active Retail Intelligence administrators access to the user/user-group form and other Active Retail Intelligence administrative forms. The error user group is used for events that encounter fatal processing errors, and though it is not restricted from use by business analysts, it is difficult to justify when its use as an assignment target (other than in the Active Retail Intelligence code itself) would be appropriate.

Active Retail Intelligence Analyst/Administrator Group

The Active Retail Intelligence Analyst/Administrator user group is a group that receives notification when there are no users in any of the groups to which an event is assigned. The Active Retail Intelligence analyst can use Active Retail Intelligence tools to correct the problem by adding users to the groups that have no users.

Active Retail Intelligence Analysts/Administrators are generally business consultants with technical analysis and/or development skills.

The Active Retail Intelligence Analyst/Administrator user group is notified when an event process enters an infinite reevaluation/action loop. In addition, this group can be used as a user-group assignment target for events that require administrative access for any kind of self-monitoring rules that may be set up for Active Retail Intelligence.

Active Retail Intelligence Error/Administrator Group

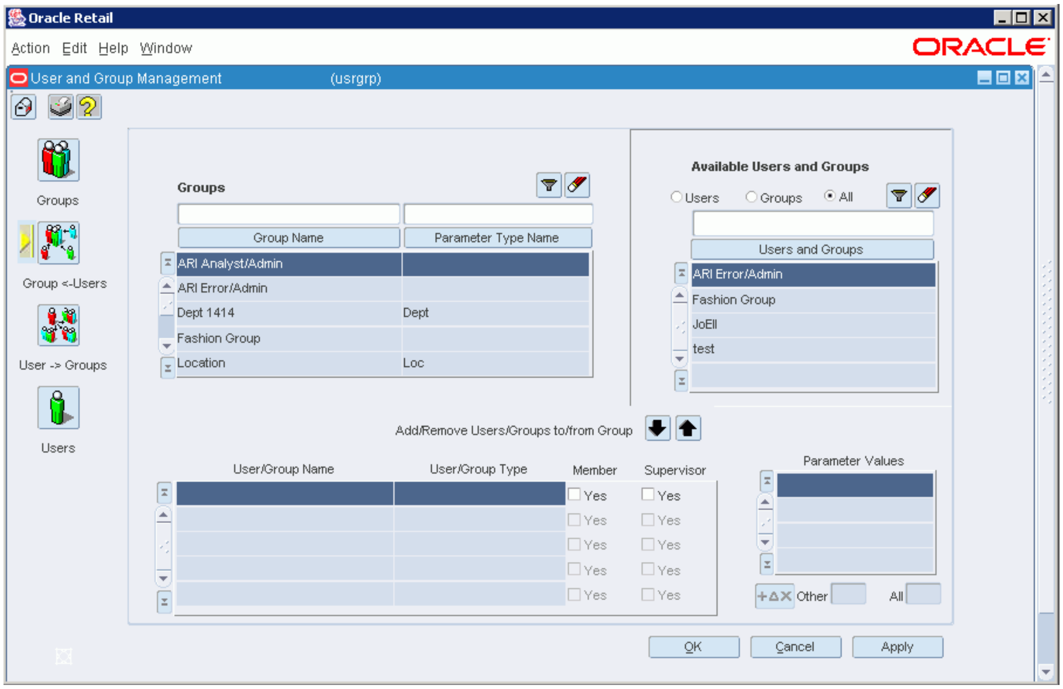
The Active Retail Intelligence Error Administrator Group receives notification when un-handled or fatal processing errors occur anywhere within Active Retail Intelligence. The Active Retail Intelligence code handles routing such events to the Active Retail Intelligence Error user group.

Processing such errors must usually be done by looking into code issues or identifying some problem that has arisen with the database. The people responsible for processing such errors are generally database administrators and/or system administrators.

Although there are no restrictions on your using the Active Retail Intelligence Error Administrator Group as a target for assigning events, it is difficult to justify any situations other than this internal error handling where the Active Retail Intelligence Error user group should be used as an event assignment target.

Group Members Screen

The Group Members screen shows all users and groups that are members of the selected group. Users and groups can then be added or removed from the selected group.



Groups

Group Name	Parameter Type Name
ARI Analyst/Admin	
ARI Error/Admin	
Dept 1414	Dept
Fashion Group	
Location	Loc

Users and Groups
ARI Error/Admin
Fashion Group
JoEll
test

User/Group Name	User/Group Type	Member	Supervisor
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes
		□ Yes	□ Yes

Group Members Screen

In both screens the button below the parameter values block will call up the parameter values screen, in which the user can add or remove parameter values associated with the user group linkage. Also in both screens the user and groups blocks can be filtered and sorted using the fields and buttons above the blocks. The radio group above the users block will filter the block to display only system users, only simple groups, or both.

Field Description

Group List

The upper left part of the screen displays a list of groups and associated parameter types. Select a group from this list. The members of the group and any associated parameters are then displayed in the lower part of the screen. This list can be filtered and sorted using the filter fields and buttons above the list.

Members

Displays the members (users and groups) of the selected group.

Parameter Values

Displays the parameter values associated with the groups in the Members list.

Available Users and Groups

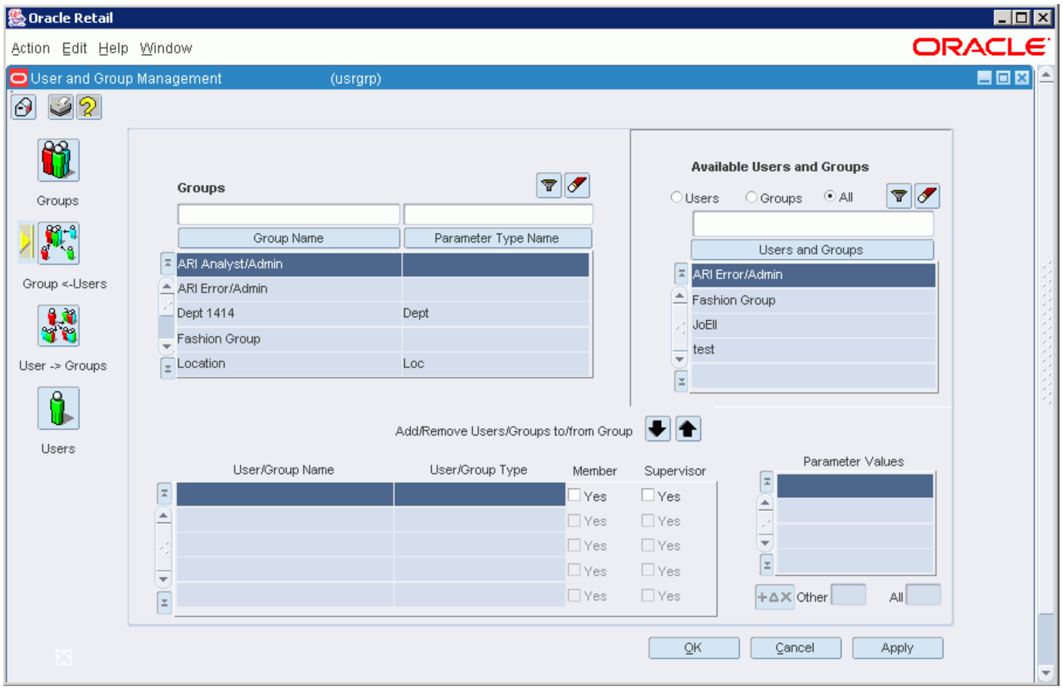
The upper right part of the screen displays all users and groups that are available to be added to the selected group. This list can be filtered and sorted using the filter fields and buttons above the list.

Up Arrow and Down Arrow Buttons

The ⬇ button adds any users or groups selected in the Available Users and Groups list to the selected group. The ⬆ button removes any users or groups selected in the Members list from the group and returns them to the Available Users and Groups list.

Parameter Values Screen

On the Parameter Values screen, you choose values for user or group mappings to parameter-driven groups. This screen is displayed when adding new users (or groups) to a parameter driven group, defining linking for users on the User Membership or Group Members screen, or editing the values for an existing group mapping. On this screen, you can add and delete parameter values for the user/group mapping.



Groups

Group Name	Parameter Type Name
ARI Analyst/Admin	
ARI Error/Admin	
Dept 1414	Dept
Fashion Group	
Location	Loc

Users and Groups
ARI Error/Admin
Fashion Group
JoEll
test

Parameter Values Screen

User and Group Names

The top of the screen displays the user or group name, and the group that is having new members being added to it. If multiple users or groups are being mapped, these fields will display Multiple Users and Multiple groups.

All Values Checkbox

Indicates whether the user/group mapping is valid for all values of the parameter. If the checkbox is checked, the parameter type/name list is blank, because the user will have access to all values for the parameter type.

Other Values Checkbox

Indicates whether the user/group mapping is valid for all values of the parameter that are not assigned to any user. This field can be checked in addition to any individual parameter values that are selected. It is not available if the all values field is set to Yes.

Parameter Value

This field will contain the parameter value associated with the user/group mapping. This field will be enabled for records that have not been written to the database. When a new value is specified for this field, a check will be made against the record group containing all valid values for the parameter type. The list of values associated with this field will display all records in the record group of valid values. Clicking the list of values button displays the list of parameter values from which you can choose.

To add a parameter value to an existing user/group mapping, click the ➕ button. To delete a parameter value from the list, click the ❌ button.

Parameter Description

A description of the parameter value. This field is read-only at all times. After selecting a parameter value, the parameter description field is automatically filled in with the parameter value’s associated parameter description value.

OK

Validates the information you have entered, posts the information to the database, and closes the Parameter Values screen.

Cancel

Cancels any changes and closes the Parameter Values screen.

4

Exception and Event Definition

The main jobs performed by Active Retail Intelligence are detecting exceptions and guiding a user through a set of workflows or events to respond to the exceptions.

Revalidation of External Realm Exceptions

When an exception occurs and an event is created, that event stays active until you close it by your actions or the event fails reevaluation. Events are reevaluated before actions on them are processed, and, depending on how your user preferences are set, as you open the event viewer to examine and act on them. Event reevaluation verifies that the exception data conditions that caused the event are still true.

Clearly, reevaluation is critical because it prevents you from taking action to correct data conditions that no longer exist. For example, if an order is submitted and an event is created, then the order is unsubmitted, and then you try to approve it from Active Retail Intelligence without first reevaluating, you would be trying to approve an order that is no longer submitted. This would cause various processing errors and be a waste of your time, but not nearly as big of an issue as if you had acted to order more stock for an out-of-stock event, only to find that in the meantime other stock had been transferred in.

Usually revalidation is not a concern since it always happens before action processing. However, you can save some time by using the revalidation user option to reevaluate when viewing events as well, since you can weed out many invalidated events before even trying to decide how to process them.

One case where revalidation is an issue is when the realm causing the exception is part of an external system, which cannot be queried nor validated. In this case, there is no way to ensure that you are not acting to correct a problem that no longer exists. In other words, even if it is not so, *external realm exceptions are treated as if they are valid until the event is closed.* The only workarounds are to be aware of the issue and do the following:

- Take advantage of the creation date parameter added by default to all exceptions, and add it to the event as a parameter that must come from the exception. This gives it visibility, so take it into consideration when acting on such events.
- Name the events that are driven by external systems with some convention that users will no require immediate action, or make all such events high priority.
- Manually revalidate exception data by doing additional investigation, outside of the data that Active Retail Intelligence offers you, before taking a corrective action.
- When possible, reserve external monitoring for events that do not require system critical actions, focusing more on informational notifications.
- An issue such as this prompts the question of why revalidation does not work in some other way (internally at least, since it does not work at all externally). For example, why does Active Retail Intelligence not automatically invalidate a purchase order approval event immediately when the order is not submitted back to Worksheet status, and could such a method work for external revalidation?
- Most automatic event invalidation strategies are either administratively cumbersome for you or performance prohibitive for the database, so on-demand revalidation seems the best way to go. Even so, Oracle has a continued interest in investigating ways to auto-invalidate external-realm-exception-driven events or to make those external realms more accessible (able to be queried) for on-demand revalidation.

- Before reevaluation or revalidation all parameters that can be refreshed ARE refreshed. An exception driven by an "external" (non-refreshable) realm is an "external realm exception", though some of its parameters added from other realms may be refreshable, so the idea that an entire exception is external is somewhat incorrect.
- However, the driving realm of an exception does determine how it can be monitored. A non-refreshable (external) driving realm means that the exception monitoring is trickle monitoring (when the realm is an external data feed realm), using the external monitoring APIs.
- Tables deployed in the same database instance (internal) as Active Retail Intelligence can be monitored either in real-time or with batch. Oracle tables deployed in a different database instance (external to the Active Retail Intelligence instance) but connected with an appropriate link can be monitored through batch. In both cases these parameters can be refreshed.

Testing Exception and Event Processes

As you are learning to use Active Retail Intelligence, and even as you gain proficiency, you should try to test all new exception and event processes in a special development environment before using them in production. This is important because of performance issues and so you can ensure that the functionality is correct. Although creating Active Retail Intelligence processes is much simpler than doing customization work on the RMS, if these processes are critical, or destined to become critical to your operation as their definitions become more and more refined, then they should be treated as customizations from the perspective of planning, testing, and management.

Exception Manager Dialog [exmgmt]

The Exception Manager dialog is used to define, view, and maintain exceptions.

Ways to Create New Exceptions

There are several ways to create new exceptions:

- By using the exception wizard, which walks you through the steps for creating exceptions. The wizard for creating exceptions shares many of the screens within the Exception Manager dialog. To use the wizard, click the Wizard button.
- By creating the new exception manually or from scratch; that is, by working through the screens of the Exception Manager and filling in information about the exception.

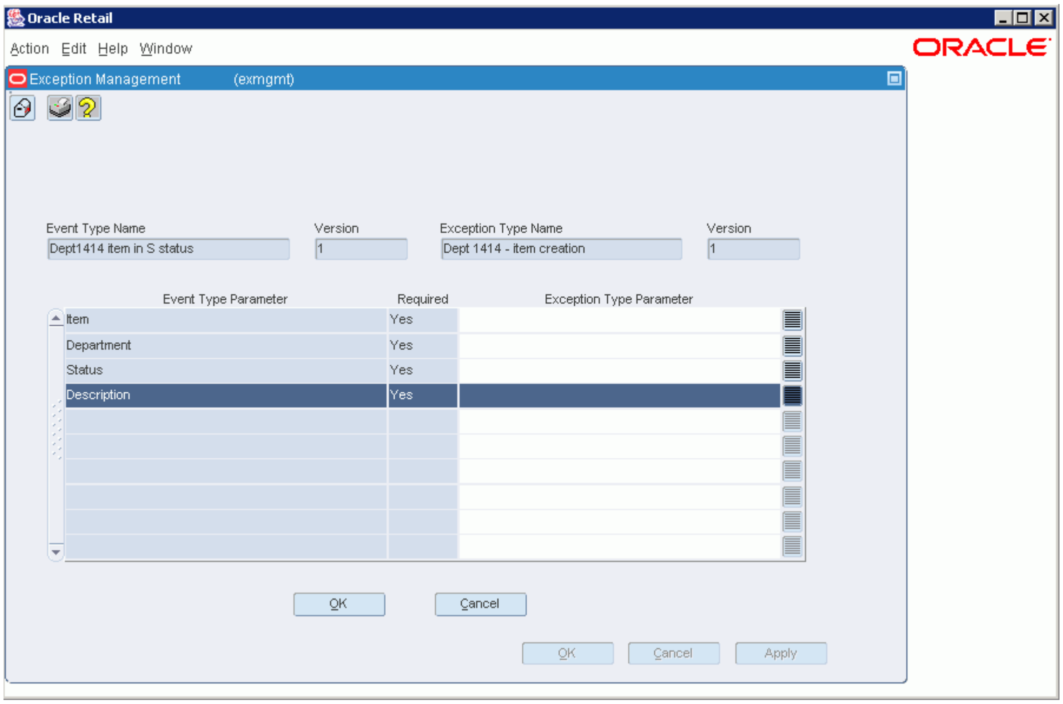
  To create an exception manually, click the ✚ button.
- By versioning, or creating a new exception from a version level of an existing exception, and establishing a date and time at which this new exception version will take effect. To create an exception through versioning, click the 🗎 button.
- By cloning, or copying, an existing exception. Click the 📋 button.

The Exception Manager has several screens for entering data about the exception. The Exception Summary List, the first screen that is displayed when you open the Exception Manager, presents summary information about exceptions. From this screen, you can access the rest of the dialog's functionality.

- Exception Type – Event Type Link Parameter Mapping
- Exception Type – Event Type Link
- Exception Type Conditions
- Exception Type Header
- Exception Type Parameter Mapping
- Exception Type Parameters
- Exception Type Schedules
- Exception Types Summary
- Needed Parameter Types Window [exmgmt]

Exception Type – Event Type Link Parameter Mapping Screen

The Exception Type – Event Type Link Parameter Mapping screen is used to add/edit links between exception types and event types. This screen is available from both the Exception Manager and Event Manager dialogs.



Event Type Name	Version	Exception Type Name	Version
Dept1414 item in S status	1	Dept 1414 - item creation	1

Event Type Parameter	Required	Exception Type Parameter
Item	Yes	
Department	Yes	
Status	Yes	
Description	Yes	

Exception Type – Event Type Link Parameter Mapping Screen

Field Description

Event/Exception Type Header

The header section displays the name and version of the event type and exception type for which a link is being established or edited. None of these fields are editable.

Event Parameters/Required/Exception Parameters

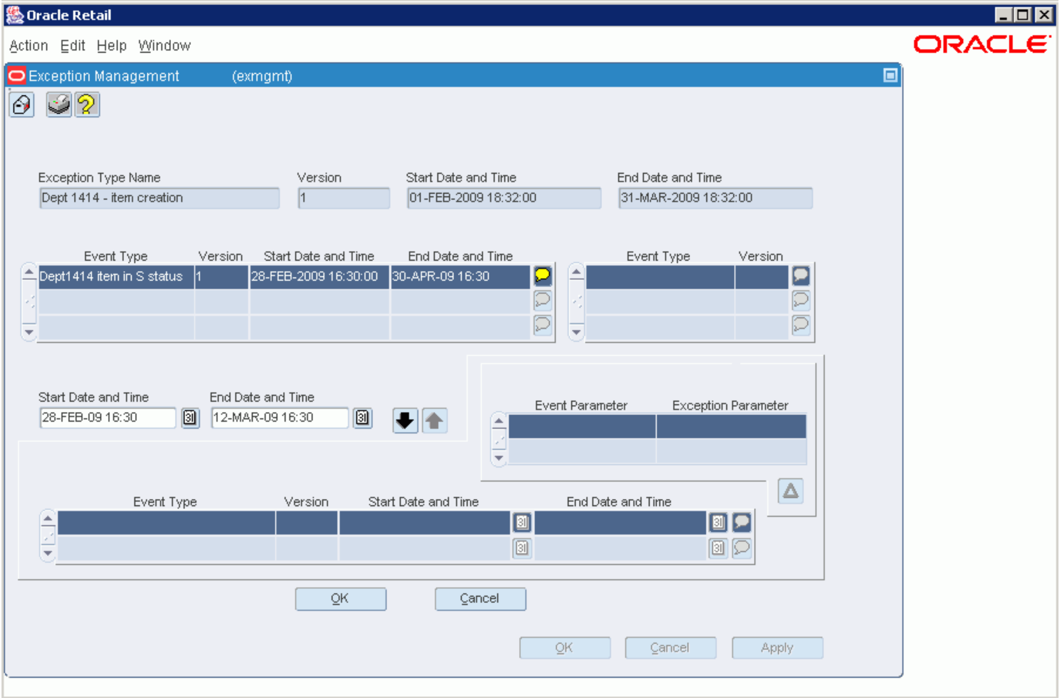
This section lists all of the event parameters for which parameters of corresponding type exist on the linked exception.

Parameters labeled Required must be mapped to complete a valid link between an exception and an event. These are also referred to in Active Retail Intelligence as "event key parameters." The exception parameter being mapped to the event parameter must be of the same type or a more specific type than the event parameter. Non-required parameters may be mapped, in which case the value of the event parameter will be passed from the exception, even though it could also be fetched from an online system (as the event is currently defined).

Use the LOVs and Clear buttons to select exception parameters or clear them from the list.

Exception Type – Event Type Link Screen

The Exception Type – Event Type Link screen is used to add, edit, and delete links between event types and an exception type. The screen has the following sections.



Exception Type Name	Version	Start Date and Time	End Date and Time
Dept 1414 - item creation	1	01-FEB-2009 18:32:00	31-MAR-2009 18:32:00

Event Type	Version	Start Date and Time	End Date and Time
Dept1414 item in S status	1	28-FEB-2009 16:30:00	30-APR-09 16:30

Start Date and Time: 28-FEB-09 16:30

End Date and Time: 12-MAR-09 16:30

Exception Type – Event Type Link Screen

If you are using the exception wizard to define an exception, this is the last screen of the wizard.

Field Description

Viewing Exception Type – Event Type Links

When viewing exception and event type links, only the OK button is enabled, and all fields are read-only.

Maintaining Exception Type – Event Type Links

When using this dialog to add, edit, and delete exception and event type links, you use the following sections of the screen: a header, a list of available events, a list of unavailable events, an add/remove section, a linked events section, and a link mapping section.

Exception Header

The header section displays information about the current exception type. All of these fields are read-only.

Available Event Types

Available event types show event types that are valid, have some overlap in their start and end dates with those of the exception type, and for which the exception type has appropriate parameter types to map to the events key parameters. The comments button in this section displays the event type description in a pop-up comments window.

Unavailable Event Types

Unavailable event types are like available events with respect to validity and an existing overlap period, but the exception is missing one or more of the event's key parameter types. Double-clicking or pressing the key-list-value key displays a list of needed parameter types.

Active Link Period

The active link period is a pair of dates that are used when the exception type and event type are linked to set the active period of the mapping. These dates are used with the Add button. An exception type can be mapped to an event type more than once and the validation for these dates is somewhat complicated. In every case it is necessary to account for the fact that a null end date acts like an infinity end date. The start and end date must be between the minimum of the exception type and event types end dates and the maximum of its start dates. The start date is further restricted in that it must be later than the current date. Finally, if the event type is already mapped to the exception type, the start and end dates must define a period that does not overlap with an existing mapping. If valid date has been entered, the link mapping screen is invoked to complete the process of specifying how the key event type parameters will be populated from the exception type.

Linked Event Types

The linked event types section shows already mapped event types. A radio group toggles between link period dates and event type activity dates. None of the fields are editable except the link start date and end dates. These are both editable, subject to the aforementioned date rules, provided that the start date is always later than the current date. Once the start date had passed only the end date is editable; in which case the end date cannot be set earlier than the current date.

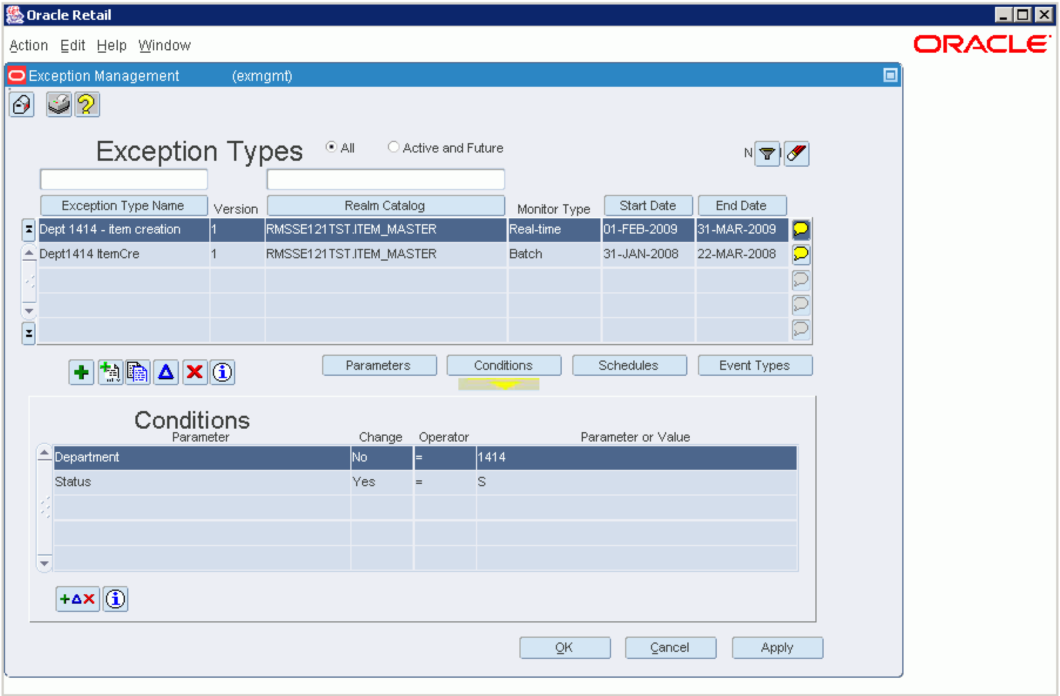
Note also with respect to date validation that when editing already linked events, there is no visibility to the event start and end date so a special error message should be given if the user attempts to choose a date outside of this range. Only events that have not entered the active link period (mapping start date has not yet passed) can be removed. Finally, if the end date has passed, neither date is editable.

Parameter Mapping

The parameter mapping section of the screen shows which event type parameters are mapped from which exception type parameters. The mapping is editable only before the start date of the linkage has passed. The Δ button displays the Exception Type - Event Type Link Parameter Mapping screen.

Exception Type Conditions Screen

The Exception Type Conditions screen is used to add, edit, and delete conditions of an exception. On this screen, you define the conditions that drive the exception and that, when all of them are true, cause an exception instance to occur.



Exception Type Name	Version	Realm Catalog	Monitor Type	Start Date	End Date
Dept 1414 - item creation	1	RMSSE121TST.ITEM_MASTER	Real-time	01-FEB-2009	31-MAR-2009
Dept1414 ItemCre	1	RMSSE121TST.ITEM_MASTER	Batch	31-JAN-2008	22-MAR-2008

Parameter	Change	Operator	Parameter or Value
Department	No	=	1414
Status	Yes	=	S

Exception Type Conditions Screen

Field Description

Exception Header

At the top of the screen is header information about the exception type for which conditions are being defined or edited. This information is read-only.

Exception Conditions

The rest of the screen details information about exception conditions. It consists of an editor for creating or editing conditions, and a list of all the conditions defined for the exception.

Condition Editor

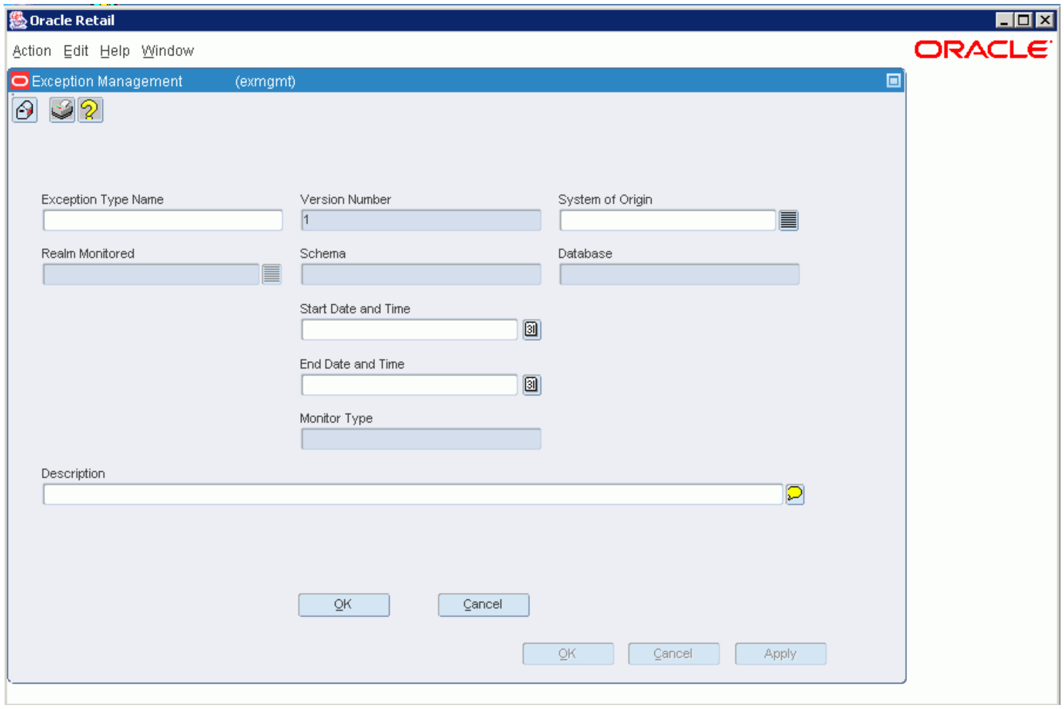
The condition editor is used to add, edit, and delete conditions showing that a parameter changes or that a relationship exists between two parameters or a parameter and a value.

Exception Condition List

The bottom part of the screen displays all the conditions currently defined for the exception. Each entry shows the details of each condition. Conditions are moved to and from this list by the ⬇ and ⬆ buttons.

Exception Type Header Screen

The Exception Type Header screen is used to set up exception type header information. This screen has the following fields:



Exception Type Name

Version Number: 1

System of Origin

Realm Monitored

Schema

Database

Start Date and Time

End Date and Time

Monitor Type

Description

Exception Type Header Screen

Field Description

Exception Name

A unique name by which the exception is known in your system.

Version Number

The version number of the exception.

Realm Type

The kind of realm to be monitored as the exception driver. Different realms allow different kinds of monitoring. For more information on realm types, see the description of the Realm Maintenance screen, which is part of metadata maintenance.

Realm Monitored

The monitored data realm that will drive the exception.

Schema

The database schema that owns the monitored realm. This field is for informational purposes only.

Database

The database instance identifier of the database to which the database schema belongs. This field is for informational purposes only.

Start Date/Time

The date and time at which the exception takes effect.

End Date/Time

The date and time at which the exception expires.

Monitor Type

The type of monitoring used for the exception, which may be batch, real-time, or trickle. All new exceptions have either batch or trickle monitoring until conditions are added that monitor change, in which case the monitoring for the exceptions transitions to real-time monitoring. Whether exceptions are of batch or trickle monitoring type depends on whether they are monitoring "external" data realms, such as External through the API.

Description

An explanation of exception purpose, or any other notes that are appropriate to include with the exception's definition.

Creating Exception Types

When creating the first version of a new exception, the following fields are enabled and editable: Exception name, realm type, realm monitored, description and start and end date. Except for end date, all of these fields are required, as is version number, which is populated to 1.

Editing Exception Types: Active and Required Fields

For future exception types, the following fields are editable: name, start and end date and description.

For active exception types (that is, exception types for which the start date has passed), the start date is not editable.

To change a realm or realm type on a future exception can be accomplished by deleting the future version. This is very high impact, so it is left to be done via deleting, meaning these fields are *not* editable in Edit mode.

Viewing Exceptions: Active and Required Fields

In View mode nothing is editable. The screen returns to the calling screen when called in View mode.

Cloning Exceptions: Active and Required Fields

Cloning acts like Edit mode, except that you are working on a new record that has been created and manually populated with a version number of one and the name is left blank.

Versioning: Active and Required Fields

During versioning based on an exception with an active realm (cloned version), versioning acts just like cloning, except that the version number is the next highest version number based on the exception type.

During versioning that is to be the next version of an exception type, but is not directly based on a known exception (either because the driving realm needs to change either functionally or because it was deactivated) then versioning acts like New mode (new version). This second kind of versioning can be in either standard or wizard format and is in all ways like New mode except that the exception type ID and version number are known.

Start Date, End Date, and Versions

In every instance of working with exception types, start dates cannot be set to be before now (current date and time) or before the end date of the previous exception version, whichever is greater. End dates cannot be set later than the start date of the next version, if one exists, and also become required fields if a later version exists. When creating a new version, if not already set, the previous version’s end date is updated to equal the new start date of the new version and the user is notified of the update.

> **Note:** Changing start and end dates may impact mappings to schedules and events. You are notified of this potential impact before leaving the screen.

Exception Type Parameter Mapping Screen

When you choose a parameter from a realm outside of the exception realm, or one that can be mapped in more than one way, the Exception Type Parameter Mapping screen is displayed to illustrate how the parameter will be queried.

Field Description

Key Parameters Needed to Link to Selected Parameter

On the left side are the key parameters of the realm of the parameter you want to fetch.

Exception Type Parameters Used to Make the Key Links

On the right side are the parameters on the exception that are of the appropriate types to make the mapping. The exception type parameters are displayed using the exception parameter name (alias).

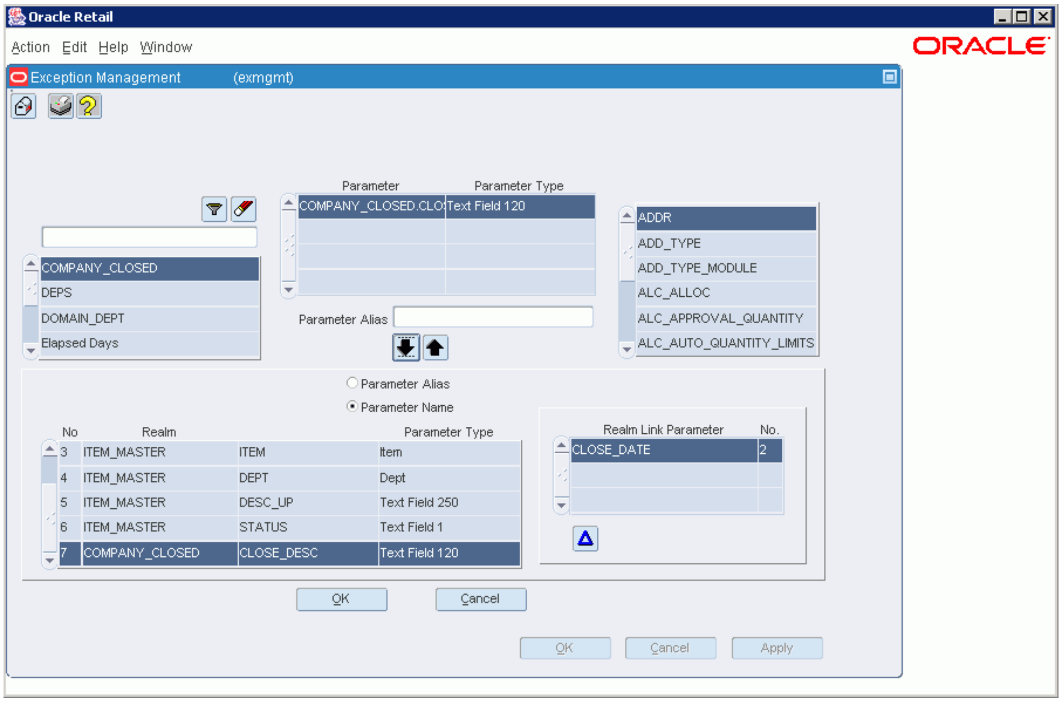
Mapping the Parameters

Using the LOV button on the right of the screen, you select the parameters of the type needed for the key parameters of the realm being mapped. However, the combination must ultimately be unique to allow the completion of the parameter mapping. For key parameters that have only one mapping option from the set of exception type parameters, that value is automatically filled in.

In several cases, this screen is shown even when there is only one mapping combination. This is so that you can be clear about how the data-model navigation is occurring, even when there is only the one mapping option.

Exception Type Parameters Screen

On the Exception Type Parameters screen, parameters are added to the exception. The top half of the screen shows available realms, the parameters of the currently selected realm, and unavailable realms. The bottom half shows the actual parameters on the exception and the links that map those parameters based on other parameters that had been added to the exception previously.



COMPANY_CLOSED
DEPS
DOMAIN_DEPT
Elapsed Days

Parameter	Parameter Type
COMPANY_CLOSED.CLO	Text Field 120

ADDR
ADD_TYPE
ADD_TYPE_MODULE
ALC_ALLOC
ALC_APPROVAL_QUANTITY
ALC_AUTO_QUANTITY_LIMITS

Parameter Alias

○ Parameter Alias
● Parameter Name

No	Realm		Parameter Type
3	ITEM_MASTER	ITEM	Item
4	ITEM_MASTER	DEPT	Dept
5	ITEM_MASTER	DESC_UP	Text Field 250
6	ITEM_MASTER	STATUS	Text Field 1
7	COMPANY_CLOSED	CLOSE_DESC	Text Field 120

Realm Link Parameter	No.
CLOSE_DATE	2

Exception Type Parameters Screen

Field Description

Available Realms and Unavailable Realms Lists

The available and unavailable realm lists show the realms from which you can choose parameters. Realms whose set of key parameters' parameter types is a subset of the parameter type set of the parameters already on the exception are available. Other realms are unavailable. Every time a parameter is added to or removed from an exception, the list of available and unavailable realms is updated.

Available Realms

The Available Realms section displays realms from which parameters can be selected and added to the exception type.

The filter button allows you to filter the view of exception type parameters by the available realms defined in your system.

Using the radio buttons, you restrict the list of available realms. *Monitored* shows only the monitored realm so that you can quickly find the monitored realm, which is probably where most of your parameters will come from. *All* shows all available realms.

Based on your selection, all realms that could be mapped based on parameters already on the exception are displayed.

To filter the view, select a realm from the list and click the ▽ button. To clear an active filter, click the ✎ button.

Unavailable Realms

The unavailable realms list shows realms from which parameters could be selected *if* the appropriate driving parameters already had been added to the exception type. The list of unavailable realms does not include rexternal system realms (which cannot be queried), because no choice of parameters will ever make them available. Such a realm is available only if it is the monitored realm of the exception.

Double clicking one of the unavailable realms in the list or pressing F9 displays the Needed Types screen, which shows a list of parameter types that are needed in order to access an unavailable realm.

Parameters in Available Realm (Parameter/Parameter Type List)

The parameter list between the Available Realms and Unavailable Realms displays the parameters associated with the currently selected available realm, and the parameter type for each parameter. Parameter names are displayed using the format [realm name:parm name].

Parameter Alias

An alias for the parameter. Although it is rare on an exception, it is possible to add the same parameter twice using a different mapping each time. To distinguish these parameters from one another later, when building conditions or mapping an event, the option of creating an alias for each parameter is provided. However, it is typically not necessary and so is not required. Consider the following example:

Suppose you are monitoring the transfer table for transfers between two stores that are not in the same promotion zone. In this case, you would have to add the store promotion zone parameter once for each store. These parameters would be indistinguishable on other screens, which show only the realm/parameter name, unless they were aliased (for example, as *to store promo zone* and *from store promo zone*).

Down Arrow and Up Arrow Buttons

Parameters are added to and deleted from the exception by clicking the ⬇ button and ⬆ button.

The ⬇ (Move Down) button adds the select parameter to the exception. The added parameter is displayed in the Parameters section in the lower half of the screen. If the parameter is not from the monitored realm, or if more than one option to map to the realm is available, then the Exception Parameter-Mapping screen is displayed.

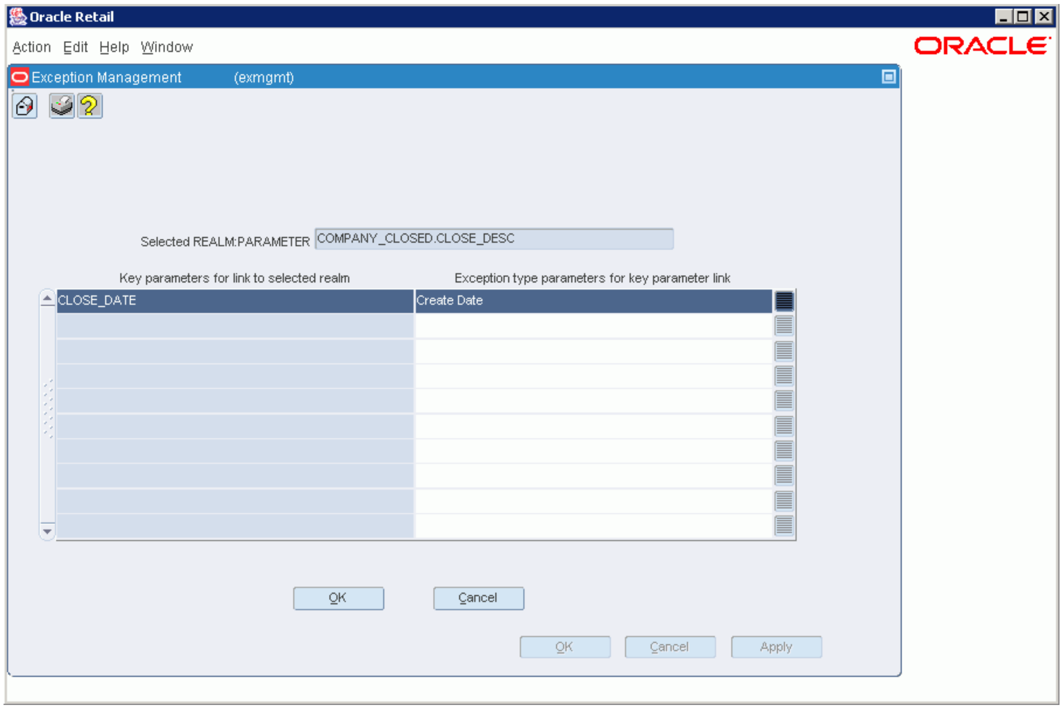
The ⬆ button removes the selected parameter and any of its dependent parameters, link dependencies, and any conditions or event mappings dependent on it from the exception.

Parameters Associated with Exception

The parameters list on the bottom of the screen shows the parameters on the exception, in the order in which they were added.

Parameter Mapping Link

The Parameter Mapping Link section shows, for the currently selected exception parameter, how that parameter is mapped based on the parameters already on the exception.



Selected REALM:PARAMETER COMPANY_CLOSED.CLOSE_DESC

Key parameters for link to selected realm	Exception type parameters for key parameter link
CLOSE_DATE	Create Date

Parameter Mapping Link Screen

In this list, the Realm Link parameter column shows the key parameter, of the selected parameter's realm, that was needed to map to that realm. The Link Number column displays the number of the parameter on the exception that was used to make that mapping. For example:

You base an exception off of order status changing by monitoring the ordering realm. In the parameters screen creation date, creation user ID, and the key parameter of the order realm, *order number*, are already added to the exception. When the order status is added, the realm link parameter is shown as *order number*, since that is the key parameter needed to get the status column from the order header table, and the link number is the sequence number of order number parameter already added to the exception.

Alias Name/Parameter Name Selections

The Alias Name/Parameter Name radio group selections determine whether the list of parameters will be displayed by parameter alias or parameter name.

Exception Type Parameters List

The exception type parameters are displayed by parameter type name, realm/logical realm name, and parameter name.

Parameter Mapping Link

This part of the screen shows the dependencies by which a selected parameter is mapped to the exception. Included here are the realm:parm name of the needed input parameter for the realm of the selected parameter, and the parameter number that was used as the input to this realm. Clicking the △ button displays the Exception Type Parameter Mapping screen, where you can specify details about the mapping between the parameters and the exception.

Viewing Exception Type Parameters: Active and Required Fields

In this mode only the parameters and link blocks are populated with data. Other blocks are entirely disabled. Everything is read-only.

Maintaining Exception Type Parameters: Active and Required Fields

In this mode, you are adding, editing, and deleting exception type parameters.

Any available parameter can be added to the list of parameters on the exception type by using the down (add) arrow. Exception type parameters must have unique names. If no alias is specified the parameter name is used, though uniqueness must be checked. When adding a parameter, the mapping for the parameter must be specified. If it can only be specified one way (based on existing parameter types and those required by the realm) then the mapping is performed automatically. Otherwise the parameter-mapping screen is invoked.

There are several special conditions that must be noted when adding a parameter. Parameters that are added from this exception realm must be added as invariant and require no special mapping. This is also true of any parameters mapped from the monitored exception realm if and only if that realm does not have any key parameters.

Removing Exception Type Parameters

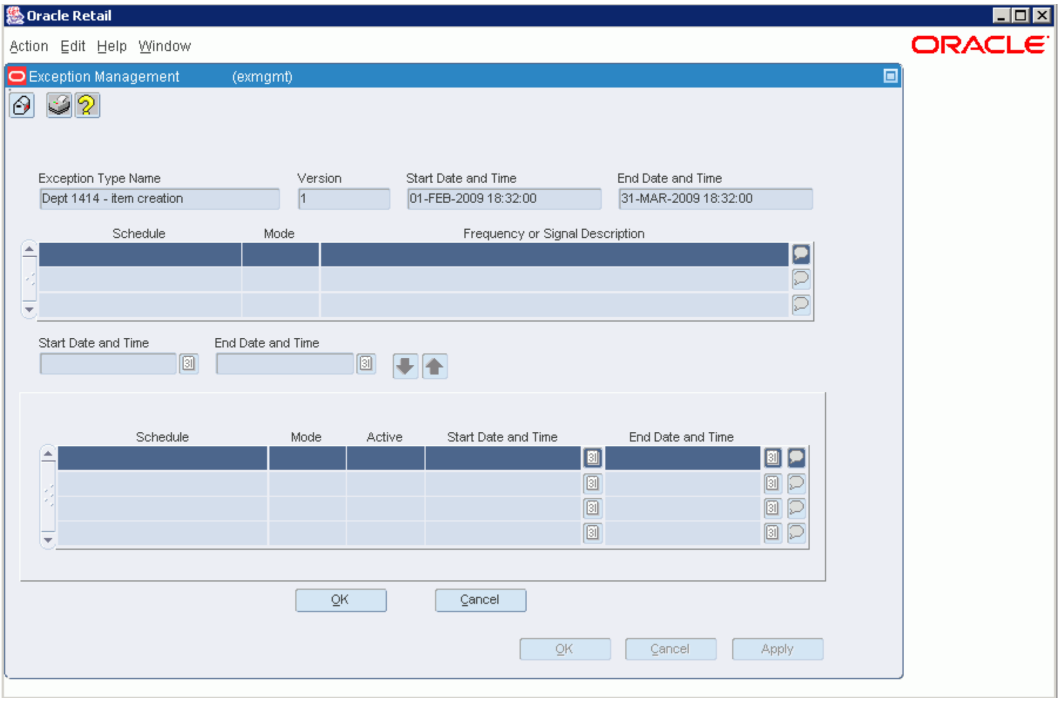
Removing parameters has several implications:

- Realm key parameters of the monitored realm (which are pre-added when the exception is created) cannot be removed.
- Any parameters that are linked via a removed parameter must also be removed, and in fact removed first, as must their dependents, and so on.
- All dependent event type mappings to the exception must be removed, as must any conditions depending on the removed parameters. In the case of key parameters that cannot be removed, you are informed of the reason.

If an attempt to remove a parameter affects dependent parameters, conditions, or event types, a warning is displayed asking whether you want to continue with the removal.

Exception Type Schedules Screen

This Exception Type Schedules screen is used to add, edit, and delete exception type schedules. The screen is divided into several sections, a header, a list of available schedules, an add/remove section and a schedules section.



Exception Type Name: Dept 1414 - item creation

Version: 1

Start Date and Time: 01-FEB-2009 18:32:00

End Date and Time: 31-MAR-2009 18:32:00

Exception Type Schedules Screen

Field Description

Exception Header Information

The header section displays information about the current exception type. This information is read-only.

Available Schedules

The Available Schedules section shows a list of all active schedules. This list includes the schedule type and a brief description of the frequency of the schedule or the signal associated with the schedule, as specified in the schedule's definition.

Active Link Period

This is the period when the schedule is used to drive this exception's monitors. Scheduling exception monitoring is achieved by applying fixed schedules over a period of time. It may be that during certain periods, you would like to use a more or less frequent monitoring schedule. Setting an appropriate link period enables this kind of variation in monitoring.

The active link period must fall within the exception type's active period and the start date cannot be set to be before the current date. Also, the link period cannot overlap any periods already established for the same schedule. When removing a schedule its active link period is copied into this active link period section and that schedule is selected in the available schedules block (unless it has become inactive before it was removed).

Down Arrow Button

The ⬇ button adds, or attaches, a schedule to the exception. The schedule is then displayed in the Schedules list in the lower part of the screen.

Up Arrow Button

The ⬆ button deletes the schedule from the exception's definition. The schedule is then displayed in the Available Schedules list.

Schedules

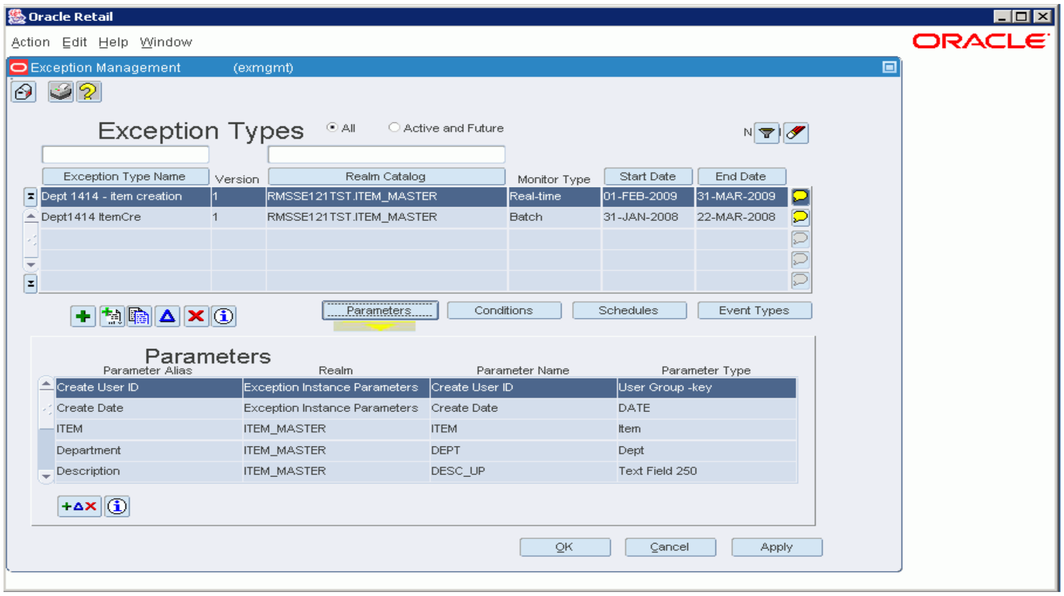
The Schedules section contains a list of all the schedules mapped to the exception. The start and end dates are editable, provided they have not already passed and in no case can they be set before the current date. These are in fact date/times, not just dates. Once a start date has occurred, the date cannot be removed or changed.

The ⬇ button and ⬆ button are for adding and removing schedules to and from the exception. The ⬆ button is essentially a delete button, but instead of just deleting it moves the schedule up into the Available Schedules list, highlights it and keeps the link period intact so you can "undo" the action by re-adding the schedule to the exception.

Schedules determine when batch scans are run for batch monitored exceptions, and thus are only meaningful for batch-monitored exceptions. Schedules for trickle and real-time monitored exceptions are not used in this Active Retail Intelligence release, though they may be defined. The active link period of a schedule determines the period during which the schedule will be used to drive exception monitoring. Multiple schedules with overlapping periods will all drive the exception. In this way schedules of different periods may be reused across multiple exceptions, and sophisticated monitoring patterns may be built up from a combination of simple schedules.

Exception Types Summary Screen

The Exception summary displays a list all exceptions defined in your Active Retail Intelligence system.



Exception Type Name	Version	Realm Catalog	Monitor Type	Start Date	End Date
Dept 1414 - item creation	1	RMSSE121TST.ITEM_MASTER	Real-time	01-FEB-2009	31-MAR-2009
Dept1414 ItemCre	1	RMSSE121TST.ITEM_MASTER	Batch	31-JAN-2008	22-MAR-2008

Parameters

Parameter Alias	Realm	Parameter Name	Parameter Type
Create User ID	Exception Instance Parameters	Create User ID	User Group -key
Create Date	Exception Instance Parameters	Create Date	DATE
ITEM	ITEM_MASTER	ITEM	Item
Department	ITEM_MASTER	DEPT	Dept
Description	ITEM_MASTER	DESC_UP	Text Field 250

Exception Type Summary Screen

Field Description

Filtering the View of Exceptions

Using the ▽ button and the list box above the list of exceptions, you can choose to display all exceptions, including expired exceptions, or only active and future-active exception types. The ✎ button clears any active filter.

Exception Types List

The top part of the screen lists all the exception types in summary form. All the information in this screen is read-only. From this summary list, you can create new exceptions and edit existing exception definitions.

- **Realm catalog:** Displayed in the format schema.realm@dblink, this value shows the name of the realm, its logical schema, and its link to another database (if applicable).
- **Monitor Type:** Identifies the type of monitoring for the exception: batch, trickle, or real-time.
- **Start Date/Time:** The date and time the exception starts being monitored.
- **End Date/Time:** The date and time exception monitoring ends.
- **Exception Name:** The name by which the exception is referred to in the Active Retail Intelligence system.
- **Version:** The version number of the exception.

Buttons

Several action buttons are available to go from this screen that allow creation, editing and deletion of exception types.

The Wizard button starts the wizard for creating a new event type. This button is active at all times.

The ✚ button creates a new exception type manually. This means you work through the screens of the Exception Manager yourself to define the new exception type, rather than using the wizard. This button is active at all times.

The ⊞ button creates a new version of an existing exception type, and establishes a date and time for the new version to take effect. The version button is available at all times that an exception is displayed in the summary list.

The ⧉ button clones, or copies, an existing exception type.

The Δ button edits an exception type that is not yet active.

The ✖ button deletes an expired or not-yet-active exception type.

The ⓘ button displays additional header information about an existing event type.

Exception Details

The lower half of the screen shows details about the currently selected exception in the top half of the screen, depending on which button is selected in the center of the screen, parameters, conditions, schedules or events.

Parameters

Clicking the Parameters button displays the parameters for the selected exception type, as defined on the Exception Type Parameters screen.

Conditions

Clicking the Conditions button displays the conditions associated with the exception, as defined on the Exception Type Conditions screen. For more information on setting up conditions for an exception, click the following link.

Schedules

Clicking the Schedules button displays the schedules that are attached to the exception type, and indicates which schedules are active now, in the future, or in the past, as defined on the Exception Type Schedules screen. From this list of schedules, you can add, edit, and delete schedules for active and future-active exceptions types. The comments box on the right side of each schedule item displays a description of the schedule.

Event Types

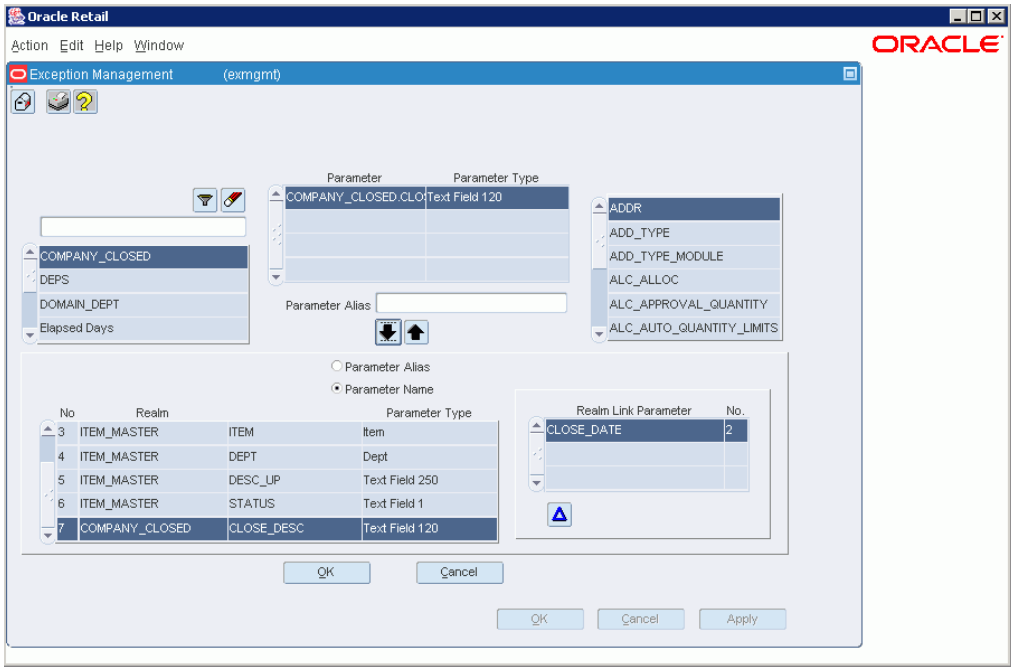
Clicking the Event Types button displays the events that are linked to the exception and the window when the link will be valid, as defined on the Exception Type-Event Type Link screen. From this list of event types, you can add, edit, and delete links to event types.

The window of time when the link is valid depends on the link period, and the exception type and event type's active dates and is computed dynamically for display here. Like schedules, event mappings may be edited in for all exceptions that are active and future active. The comments box on the right side of each event type displays a description of the event type.

Needed Parameter Types Window [exmgmt]

The Needed Parameter Types window is displayed in several situations when defining exception types:

1. In the Exception Type Parameters screen, when you double-click on or press F9 on an unavailable realm.



COMPANY_CLOSED
DEPS
DOMAIN_DEPT
Elapsed Days

Parameter	Parameter Type
COMPANY_CLOSED.CLO	Text Field 120

ADDR
ADD_TYPE
ADD_TYPE_MODULE
ALC_ALLOC
ALC_APPROVAL_QUANTITY
ALC_AUTO_QUANTITY_LIMITS

Parameter Alias

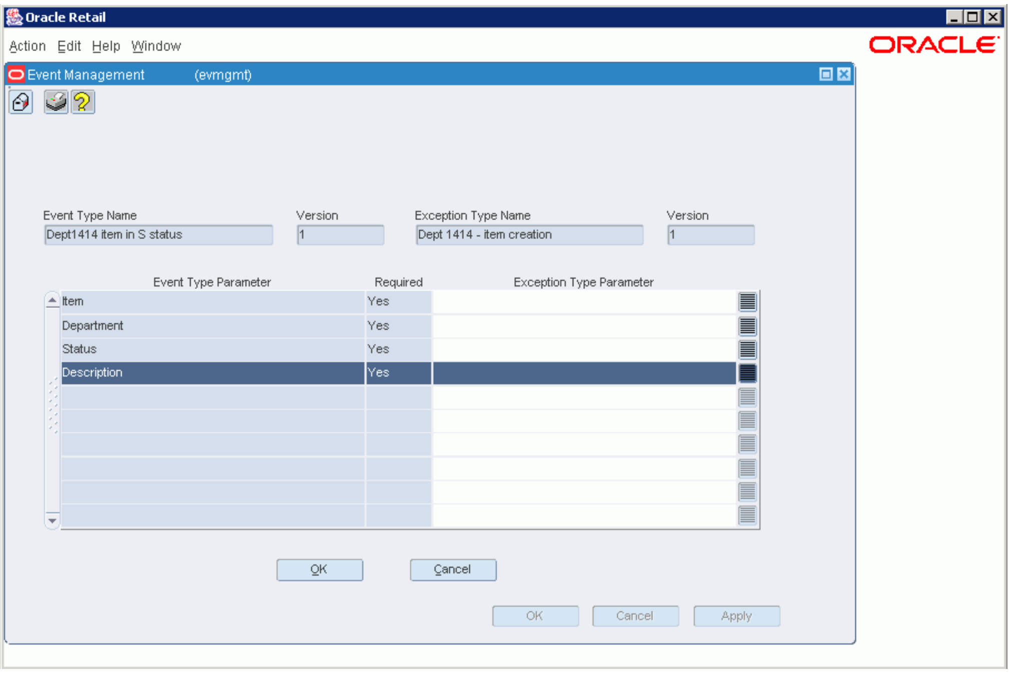
○ Parameter Alias
● Parameter Name

No	Realm		Parameter Type
3	ITEM_MASTER	ITEM	Item
4	ITEM_MASTER	DEPT	Dept
5	ITEM_MASTER	DESC_UP	Text Field 250
6	ITEM_MASTER	STATUS	Text Field 1
7	COMPANY_CLOSED	CLOSE_DESC	Text Field 120

Realm Link Parameter	No.
CLOSE_DATE	2

Exception Type Parameters Screen

2. In the Exception Type – Event Type Link Parameter Mapping screen, when you are attempting to map to an unavailable realm by double-clicking or pressing F9 on the realm.



Event Type Name: Dept1414 item in S status
Version: 1
Exception Type Name: Dept 1414 - item creation
Version: 1

Event Type Parameter	Required	Exception Type Parameter
Item	Yes	
Department	Yes	
Status	Yes	
Description	Yes	

Exception Type – Even Type Link Parameter Mapping Screen

Condition Editor

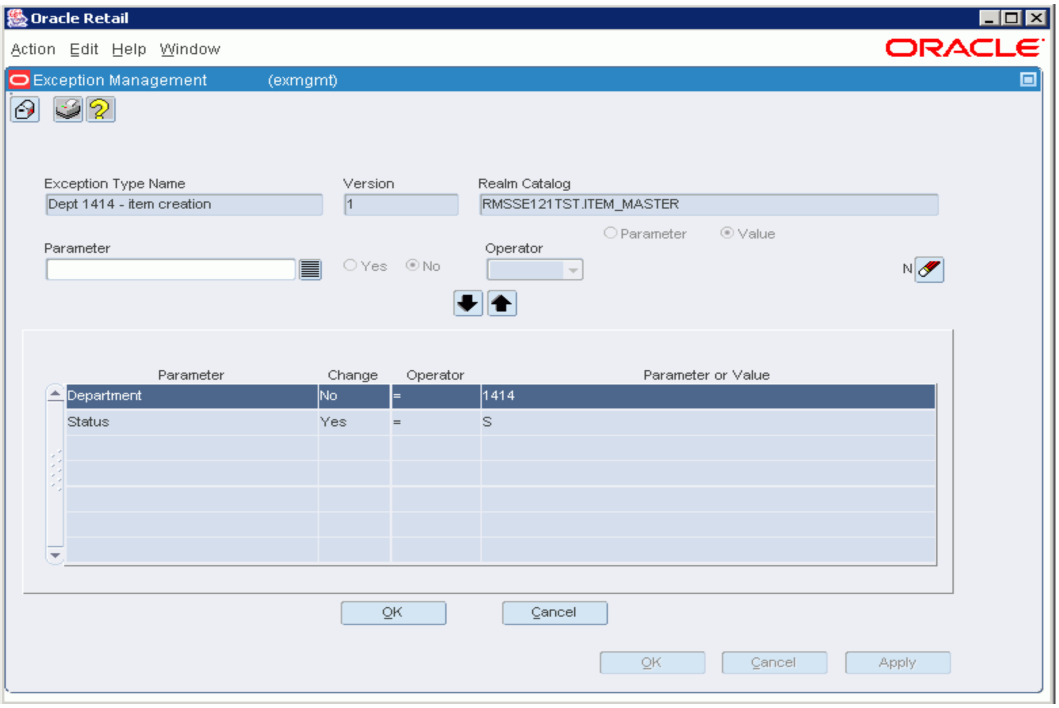
Several exception and event management screens involve defining conditions associated with the exception or event, including;

- Exception Type Conditions
- Event Type State – Event Attribute Rules
- Event Type – State Rules
- Event Type – Closure Rules

Following is a description of how to create and edit conditions on these screens.

How the Condition Editor Works

In all the screens that involve conditions, you use the Condition Editor to add, edit, and delete conditions showing that a parameter changes or that a relationship exists between two parameters or a parameter and a value. The basic layout of a condition is a comparison between a parameter and a value or another parameter, using an operator such as = or >. Exception definitions have an additional setting for the condition, Monitor Change, which specifies whether you want to be notified when the parameter change occurs. It is also possible for a parameter to change and have a relationship condition simultaneously.



Exception Type Name: Dept 1414 - item creation

Version: 1

Realm Catalog: RMSSE121TST.ITEM_MASTER

Parameter: ○ Yes ● No

Operator: ○ Parameter ● Value

Parameter	Change	Operator	Parameter or Value
Department	No	=	1414
Status	Yes	=	S

Condition Editor Screen

All conditions must be between a parameter and a value or two parameters, but never between two constant values. At least one parameter must be chosen, so when the condition editor is empty only the first parameter field and its LOV are enabled. Once a parameter is selected (only varchar2, date, number, and Boolean data types are available in the LOV), the other fields are enabled. When the first parameter field is cleared, the condition editor reverts to the original state.

Once the condition is defined, you add it to the list of exception or event-rule conditions below the editor by clicking the ⬇ button. When you need to change a condition, you click the ⬆ button to move the condition into the editor, and when done changing the condition, you click the ⬇ to move it back into the condition list for the exception or rule.

The conditions defined for an exception or an event is treated as logical AND conditions. That is, when a monitor detects candidate exception instances, or when an event is evaluated against the rules defined for it, the exception or event is tested against all of these conditions. If they are not all true, the candidate is discarded and ignored without further processing. For example, if all the conditions for candidate instance are true, then the candidate instance becomes an actual exception instance, at which time linked events are created.

Field Description

The condition editor is displayed above the list of existing conditions, and consists of the following fields and controls.

Parameter

The parameter on which the condition is based. This parameter is compared to a value or another parameter. Click the LOV button to display allowable parameters for the exception or event.

The parameter chosen to drive the condition determines:

- Which operators are available
- For exceptions, whether the exception can be monitored online
- Whether other parameters or special values can be used for the comparison created.

All parameters allow comparison to NULL values. Only parameters from the monitored realm, if the realm can be monitored in real time, allow change detection. Some parameters have a constrained set of valid values that can be used for comparison, such as the store number parameter only allowing valid store numbers. Also, the data type of the parameter, character, Boolean, date, or number, determines which comparisons can be made to other parameters (that is, those of the same data type). Actually, many times such comparisons are not meaningful, unless the two parameter types are the same, such as comparing two store numbers, as opposed to comparing a dollar amount and a store number.

Monitor Change (for Exception Type Conditions Screen Only)

The monitor change radio button indicates whether or not the condition will be real-time monitored. In other words, do we want to monitor the field changing to the entered value (for example, order status changing to 'S') or all fields that contain that value (for example, order status = S).

Monitoring a change is only possible on parameters that come from the monitored realm, and even then only on a realm with active, query-able data. So, monitoring change is impossible for external systems. For internal systems, however, whether an exception monitors transactions online or through a nightly batch sweep depends on whether at least one of the conditions is set to monitor a change.

Operator

The operator used to compare the parameter with a value or another parameter.

For Exceptions only, an operator is required for all conditions except for those in which a change is being monitored.

Available operators depend on the data type of the specified parameter. Specifically, text parameters do not support inequality operators. In addition, any exception parameters with a Long or Long Raw Data type are not available for defining conditions, since they cannot be used to perform comparisons anyway. The only reason to add such parameters to an exception is to pass them from a non-queried data source through to an event as additional information to present to users in the event interface.

For all data types, if you use the operators IS and IS NOT, the value must be specified as NULL.

For varchar2 and Boolean data types, the only other options are = and !=.

For date and number data types, the operators <, <=, >, >=, =, and != are also available.

Character and Boolean parameters allow = and != operators.

Numbers and dates allow inequality operators as well.

Parameter or Value Radio Buttons

Specifies whether the selected parameter is being compared to a parameter or a parameter value. The default setting is Value, with the Value field set up for the appropriate data type.

A parameter can only be compared to another parameter when a parameter of the same type exists on the exception or event (that is, if there are available parameters to compare the parameter against). Otherwise, the parameter must be compared to a value.

Parameter or Value

This field specifies the parameter or value to which the selected parameter is being compared.

Depending on the data type of the parameter, the controls on this field change. In some cases, when making a value comparison, a list item, a date button, or an LOV button is available to help you choose an appropriate value. For example, for dates, a date button is displayed to enable entering a date. For parameters with a code type, clicking the LOV button displays a list of appropriate code values from which you can choose.

Down Arrow Button

The ⬇ button adds the condition in the condition editor into the condition list for the exception or rule. The fields of the condition editor remain filled with the values for the condition. You can either continue to define more conditions on the same parameter, or click the ✏ button to clear the fields.

Up Arrow Button

The ⬆ button removes the condition from the condition list, and puts it in the condition editor. Click this button to edit or delete existing conditions.

Clear Button

The ✏ button clears the condition editor fields.

Conditions List

Below the condition editor is a list of the conditions currently defined for the exception or rule. The conditions defined for an exception or event are treated as logical AND conditions, as described earlier in this topic.

Adding a Condition

1. Choose a parameter from the list.
2. (For exceptions only) Specify whether you want to be notified of the change to the parameter.
3. Choose an operator from the list.
4. Choose whether you want to compare the parameter to a value or another parameter.
5. Specify the value or parameter to which the parameter will be compared.
6. Click the ⬇ button. The condition is added to the condition list below the editor.

Editing a Condition

1. Select the condition from the condition list.
2. Click the ⬆ button to move it into the condition editor.
3. Modify the definition as needed, for example, changing the operator or the value to which the parameter is compared.
4. When done, click the ⬇ button to add the condition back to the condition list for the exception or rule.

Deleting a Condition

1. Select the condition from the condition list.
2. Click the ⬆ button to move it into the condition editor. The exception condition is left in the edit section, since the only way to edit a condition is remove it, change it, then add it back to the exception conditions.
3. If you want to remove an exception condition and then add another, remove the condition, and then change the first parameter. Or, click the ⌫ button to clear the fields and start with a new exception condition.

Event Manager Dialog [evmgmt]

The Event Manager dialog is used to define, view, and maintain events.

Ways to Create New Events

There are several ways to create new events:

- By using the event wizard, which walks you through the steps for creating events. The wizard for creating events shares many of the screens within the Event Manager dialog. To use the wizard, click the Wizard button.
- By creating the new event manually; that is, by working through the screens of the Event Manager and filling in information about the event. To create an event manually, click the ✚ button.

- By versioning, or creating a new event from a version level of an existing event, and establishing a date and time at which this new event version will take effect. To create an event through versioning, click the button.
- By cloning, or copying, an existing event. Click the button.

The Event Manager has several screens for entering data about events. The Event Summary List, the first screen that is displayed when you open the Event Manager, presents summary information about exceptions. From this screen, you can access the rest of the dialog's functionality.

- The Event Viewer [arieview]
- Event Type Revalidation Schedules
- Event Type State – Action
- Event Type Header
- Event Type Parameter Mapping
- Event Type Parameters
- Event Viewer Layouts
- Event Type – Exception Type Link
- Event Type – State Rules
- Event Type States
- Event Types Summary
- Needed Parameter Types Window [evmgmt]

The Event Viewer [arieview]

The Event Viewer displays all the events of a particular event type that are assigned to you.

The Event Viewer has several parts:

Field Description

Event Type and Event State

At the top of the screen are the event type and state.

Events Assigned to You

Below the event type and event state are listed the events of that type that have been assigned to you. This part of the Event Viewer includes several items:

List of Events

The top portion of the Event Viewer lists the events of a particular event type assigned to you, along with the critical details of the event. The most important of these details is the event priority of the event. High-priority events are indicated by an exclamation mark on an envelope. Normal priority events are indicated by a plain envelope with no exclamation mark.

You can sort these events by parameter, by clicking on the column headings for the event details.

Reevaluate Events Button

Clicking the Reevaluate Events button below the event list checks the events list to ensure that all your events are current and should still be assigned to you.

Refresh Events Button

Refreshes the display of the event list.

User Events Button

Clicking the User Events button displays all events assigned to you by the Active Retail Intelligence administrator.

Supervised Events Button

Clicking the Supervised Events button displays events assigned to you as an Active Retail Intelligence supervisor. This button is activated only if you have Active Retail Intelligence supervisor user privileges.

Event Messages

Below the event list are messages about the event selected in the list. The messages include such things as why the event was created, a list of every user who has forwarded the event, messages included with the forwarded event, and errors encountered during the event's lifecycle.

Details of Event

At the bottom left corner of the screen are further details about the selected event. The event details tell you more about the event and allow you to go to the event's source.

The event details are displayed with a label for the detail in the left column, and the detail value in the right column.

If the event detail is an image or a long-text file, the detail's value in the right column is a hypertext link. Clicking the hypertext link displays the image or text file in a separate viewer.

Expand Button

If the text of the details is cut off, click the Expand button to display the event details in a larger window. Pressing F9 with focus on an event detail does the same thing.

Displaying Event Details: Images and Long-Text Files

In the Detail section of the Event Viewer, there are two types of details that display differently than normal details.

- Images, or pictures attached to the event.
- Long text, or text over 2,000 characters and under 65,534 characters that is attached to the event. Long text is usually text such as articles or long descriptions.

Images and long text have special viewers. Clicking the hypertext link in the value column, or pressing F9 when the focus is in the hypertext link, displays the image or long text in these special viewers.

Both long text and images show a hypertext link in the value column. You click this hypertext link, or press F9 when you are in the field, to go to the image viewer or the long text viewer.

Details of Events

Within the Image Viewer window, the image is displayed as well as possible within the limits of the window space available.

Once the image viewer is open, you can save the image to a file for further manipulation. To do this, type a valid Windows filename in the filename box and select a file type from the file type drop-down list. Then click the Save button to save the image.

For text displayed in the long text viewer, you can select the text and use the Edit—Copy function to copy it into the Windows clipboard and then paste it into your favorite word processing application.

Actions

The bottom right corner of the event viewer displays a list of actions associated with the selected event type, version, and state. These actions are set up by Active Retail Intelligence administrators when defining events.

Actions are performed by selecting an action from the list and clicking the Do Action button.

One action commonly found on events is the close action, which deletes the event. Deleting the event is useful if you wish to ignore the event or act on the event outside of the Event Viewer, perhaps in a way not specified in the list of actions.

Do Action Button

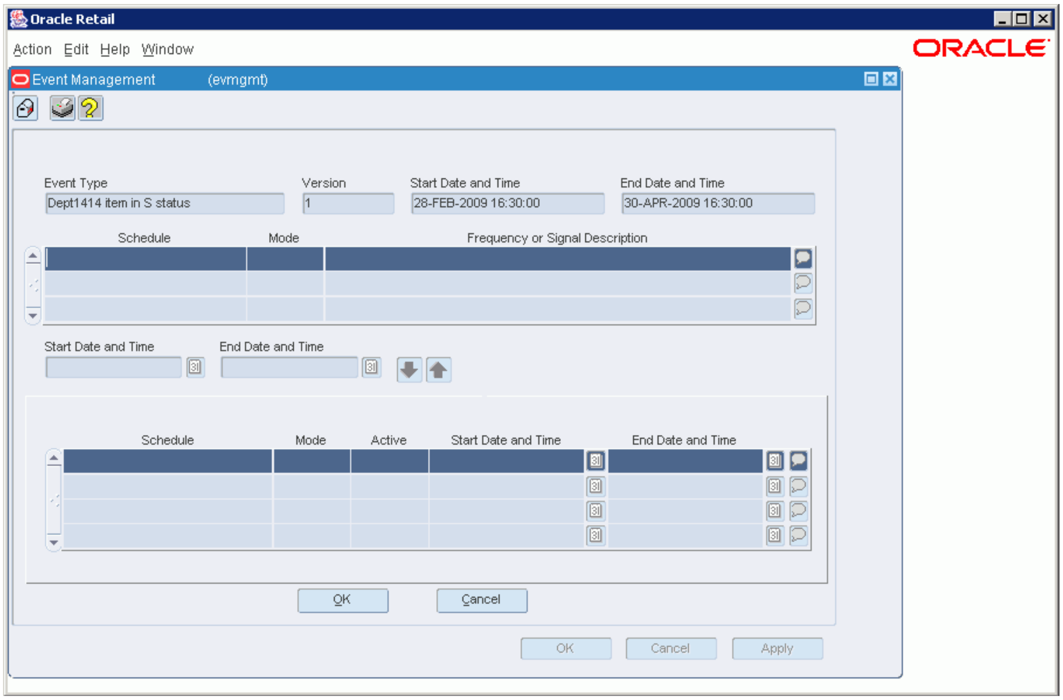
Performs the selected action. Actions can vary greatly by event, and are set up by the business analysts at your site. The action may take you to portions of the merchandising system to perform actions such as approving purchase orders or creating new purchase orders, or cause a message about the action to be displayed.

Revalidate Events

- Search in the Event Viewer
- Sort events and event details
- Display complete event message or detail
- Save an event detail image to a file
- Perform an action on an event

Event Type Revalidation Schedules Screen

The Event Type Revalidation Schedules screen is used to add, edit, and delete schedules for reevaluating the event type. The schedules are subject to the start and end date constraints that you specify on this screen. The active link period means that this schedule will be used to drive periodic reevaluation of this event (according to the schedule) between the link start and end dates. If there is no end date then the reevaluation will continue indefinitely. The link period start and end dates are editable provided they are not in the past, and in no case can they be set before the current date.



Event Type: Dept1414 item in S status
Version: 1
Start Date and Time: 28-FEB-2009 16:30:00
End Date and Time: 30-APR-2009 16:30:00

Event Type Revalidation Schedules Screen

Field Description

Event Type Header

The header section displays information about the current event type. These fields are read-only.

Available Schedules

Available Schedules shows a list of all schedules that are available to attach to the schedule. To attach them to the event type, click the ⬇ button.

Active Link Period

The Active Link Period establishes how long the schedule is attached to the event type. This period is defined by start and end dates and times.

The active link period must fall within the event type's active period and the start date/time cannot be set to be before the current date (sysdate). Also, the link period cannot overlap any periods already established for the same schedule.

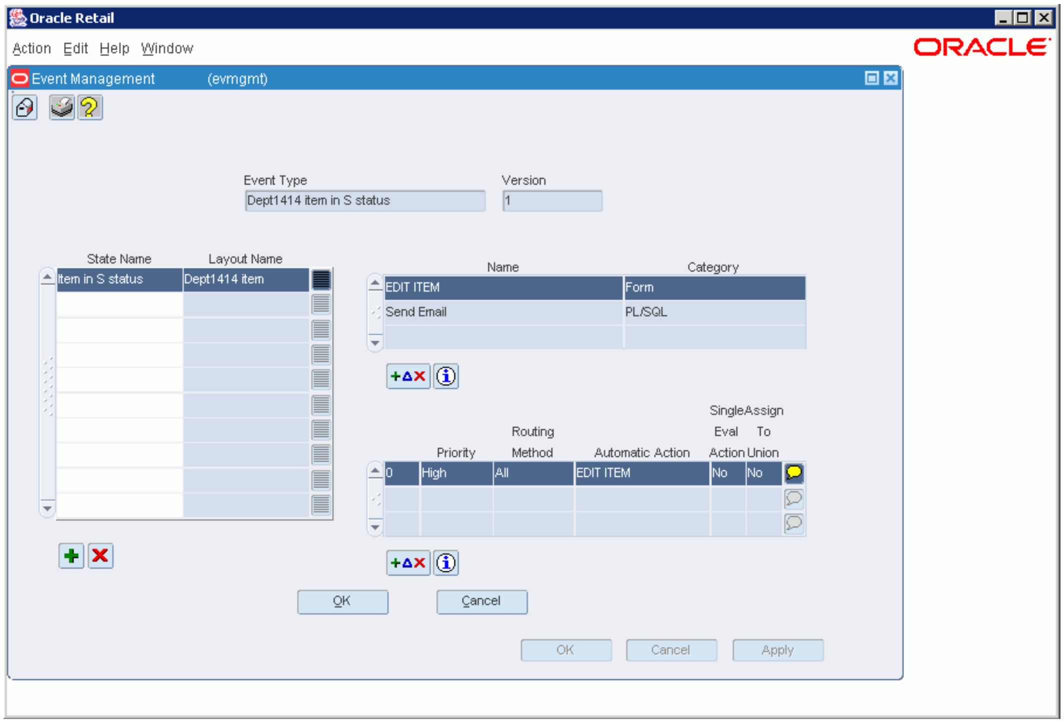
When removing a schedule, its active link period is copied into this active link period section and that schedule is selected in the Available Schedules list, unless it has become inactive before it is removed, in which case it is simply removed.

Schedules

The Schedules part of the screens shows all the schedules currently attached to the event type. To add an available schedule to this list, click the ⬇ button. To remove a schedule from this list, click the ⬆ button.

Event Type State – Action Screen

The Event Type State – Action screen defines the actions associated with the event state. This screen is divided into five sections.



Event Type: Dept1414 item in S status

Version: 1

State Name	Layout Name
Item in S status	Dept1414 item

Name	Category
EDIT ITEM	Form
Send Email	PL/SQL

	Priority	Routing Method	Automatic Action	SingleAssign Eval Action	To Union
0	High	All	EDIT ITEM	No	No

Event Type State – Action Screen

Field Description

Header

The header section is read-only.

Actions

The Actions section shows actions currently mapped to the state.

The up and down arrows can be used to reorder the actions. The resulting order will be used when displaying the list of actions in both the Event Manager and the Event Viewer.

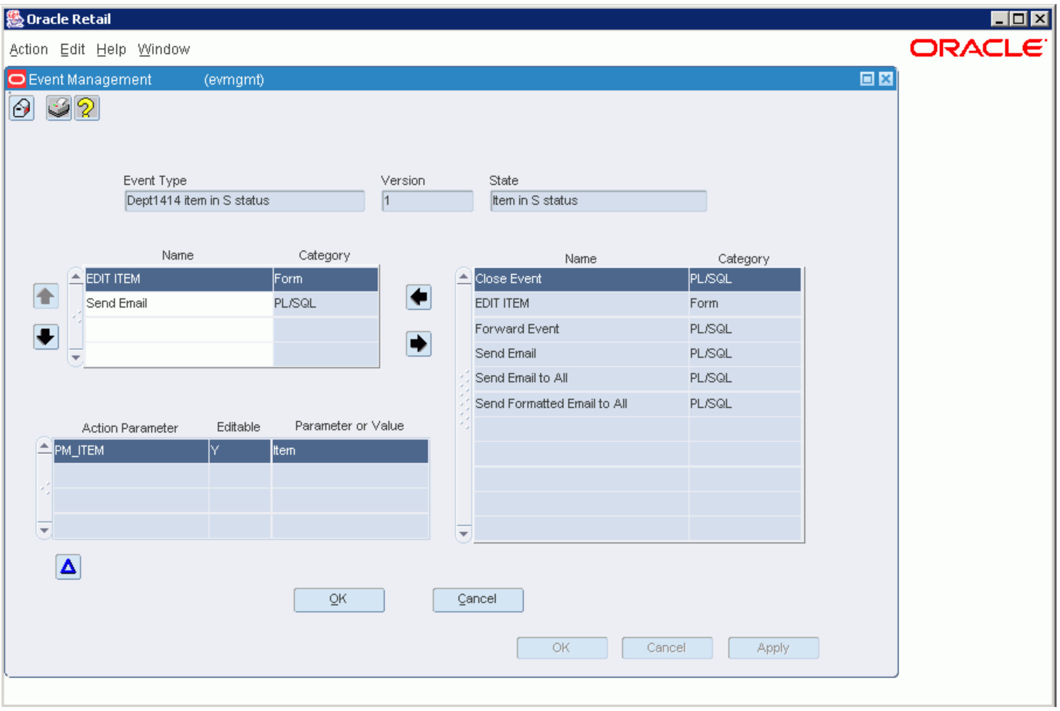
Add/Remove Actions Buttons

The ➡ button and ⬅ buttons in the middle of the screen are for adding and removing actions for the event state.

Available Actions

The available action section shows actions that can be added. The parameter mapping area shows how event parameters have been mapped to add the currently selected action to the event.

Actions can be added or removed at any time. However, removing and action may modify an attribute rule by removing it from being an auto-action for that rule. In this case, you are notified and prompted to proceed. Adding an action opens the Parameter mapping screen, unless all of the parameters of the action are override, error, or current user parameters.



Event Type	Version	State
Dept1414 item in S status	1	Item in S status

Name	Category
EDIT ITEM	Form
Send Email	PL/SQL

Name	Category
Close Event	PL/SQL
EDIT ITEM	Form
Forward Event	PL/SQL
Send Email	PL/SQL
Send Email to All	PL/SQL
Send Formatted Email to All	PL/SQL

Action Parameter	Editable	Parameter or Value
PM_ITEM	Y	Item

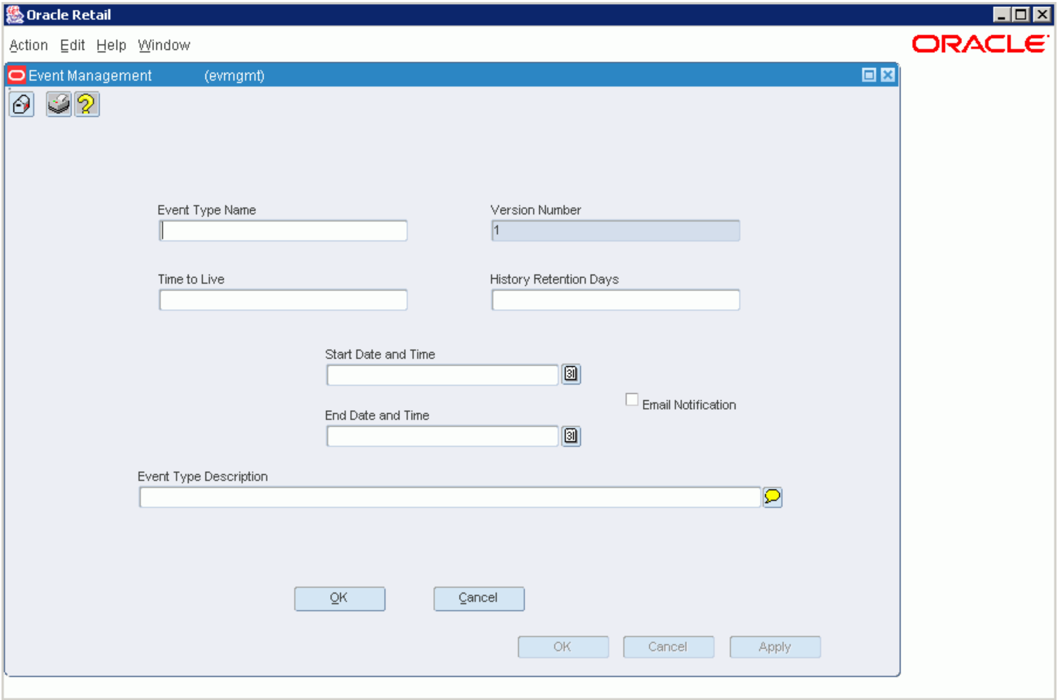
Parameter Mapping Screen

For the actions that are already mapped to the event state, the action names are editable at any time. These action names must be unique within a given event type, state, and action designation. Uniqueness is necessary to allow an action to be added multiple times (potentially with different overriding parameter values).

Parameter mappings can be edited by invoking the parameter mapping screen, but only if there are any parameters that can be mapped (not having at least one override/error/current user parameter).

Event Type Header Screen

The Event Type Header screen defines header information, such as the event name, its lifecycle, and start and end times. The Event Type Header has the following fields:



Event Type Header Screen

Field Description

Event Type Name

A unique name for the event type.

Version Number

The version number of the event.

Time to Live

The number of days that the event "lives" after it is closed. This will prevent a creation of an identical even during this period.

History Retention Days

The number of days that history for the event is retained in the Active Retail Intelligence database.

Start Date and Time

The date and time at which the event becomes active.

End Date and Time

The date and time at which the event becomes obsolete.

Event Type Description

A text description of the event and its purpose, including such information as what the event type does, or the exceptions to which the event type is mapped. The comments button on the right side of the description opens a comment window for entering or editing the description.

Notification Checkbox

Indicates whether the event supports email notification when the event occurs.

Creating Event Types

When you are creating version one of a new event, the following fields are all enabled and editable: Event Type Name, Time to Live, History Retention Days, Event Type Description, and Start Date and Time, and End Date and Time are all enabled and editable.

Except for End Date and Time, all of these fields are required, as is Version Number, which is given a value of 1.

Editing Event Types

For future event types, all displayed fields except Version Number are editable. For active event types (start date has passed), the Start Date and Time is not editable.

Viewing Event Types

When viewing event types, all fields are read-only.

Cloning Event Types

Cloning acts like Edit mode, except that you are working on a new record that has been created and manually populated with a version number of one and the name is left blank.

Versioning Event Types

During versioning based on an event that is not in Disabled (D) status, versioning acts just like cloning, except that the version number is the next highest version number based on the event type’s highest existing version number, also, start and end date are both blank.

During versioning that is to be the next version of an event type, but is not directly based on a known event type (because previous versions have been disabled), versioning acts like New mode (new version). This second kind of versioning can be either in standard or wizard format and is in all ways just like New mode, except that the event type ID and version number are known.

Start Date, End Date, and Versions

In every case, start dates cannot be set to be before or equal to the current date or before or equal to the end date of the previous event version, whichever is greater. End dates cannot be set later than or equal to the start date of the next version, if one exists, and also become required fields if a later version exists. When creating a new version, if not already set, the previous version’s end date is updated to one second less than the new start date of the new version and the user is notified of the update.

Changing start and end dates map impact mapping to schedule, exceptions, and related events, and this must be addressed when changes are saved to the screen (OK, NEXT, BACK). These date related items must be dropped if they fall out of range, or updated accordingly if they are still partially within range. Ideally any tracking tables used by various linking processes (re-evaluation, and so on) would be updated as well.

Event Type Parameter Mapping Screen

When you choose a parameter from a realm outside of the event realm, or one that can be mapped in multiple ways, The Event Type Parameter Mapping screen is displayed to illustrate how the parameter will be queried.

The left side of the Event Type Parameter Mapping screen displays a list of the key parameters of the newly mapped parameter’s realm. The right side shows the parameters of the event that are of the appropriate types to make the mapping. The event type parameters are displayed using the event parameter name (alias).

Note that in several cases this screen is shown even when there is only one mapping combination. This is so that you can be clear about how the data-model navigation is occurring, even when there is only the one mapping option.

Using the LOV, you can select from all of the parameters or parameter types of the type needed for the key parameters of the realm being mapped. However, the combination must ultimately be unique to allow the completion of the parameter mapping. For key parameters that have only one mapping option from the set of event type parameters, that value is automatically filled in.

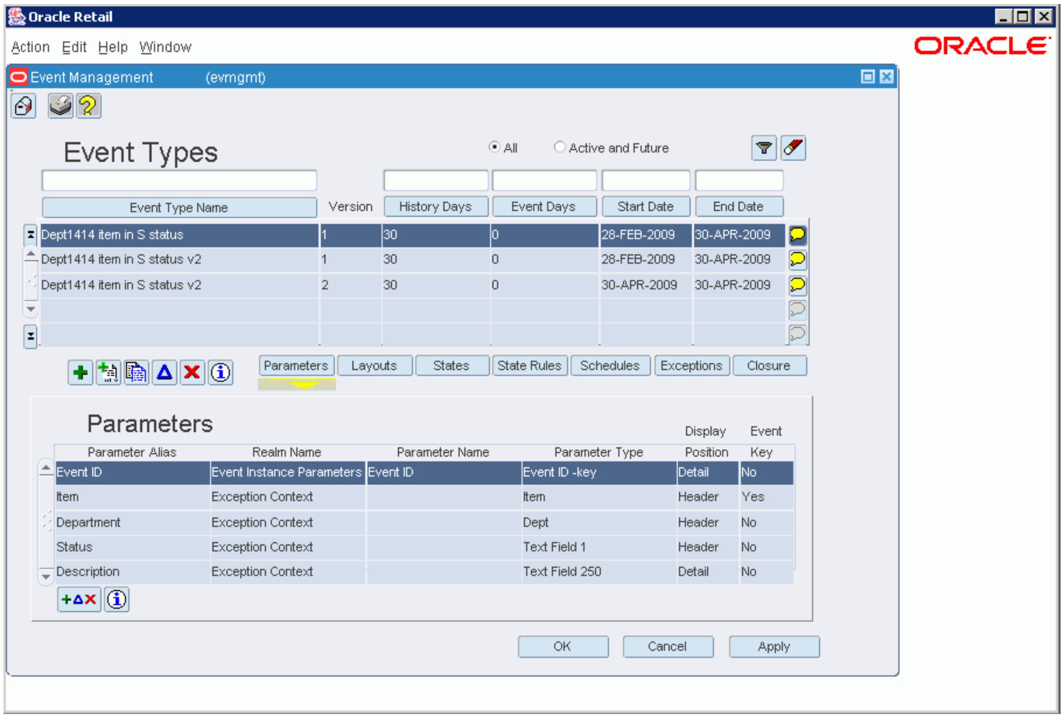
Issues

The reason for overriding an event parameter with an exception parameter, when the event parameter is not required to come from the exception, is that the event may be linked to multiple exceptions. In some cases, the information on one of the exceptions may be seen as more up-to-date than whatever data the event would have re-queried had it been caused by its other driving exception.

For example, an external data source exception and an internal exception may both cause the same event, but the one from the external data source provides information which is more accurate than the internal system (such as a recalculated open-to-buy). In this case, you might prefer to use the external calculation as opposed to the internally accessible data when the event is generated from the external data source. Note, however, that if the internal exception occurs later than the external one, the value will be overwritten with the newer internal value anyway. Thus in cases where parameters other than event key parameters come from two different exceptions, it is best to use a one-to-one exception-event mapping to ensure that the external data source values are maintained in the rare cases where preserving those values is critical.

Event Type Parameters Screen

The Event Type Parameters screen is used to add, edit, and delete parameters associated with an event. The screen is divided into six main sections.



Event Types

● All ○ Active and Future

Event Type Name	Version	History Days	Event Days	Start Date	End Date
Dept1414 item in S status	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	2	30	0	30-APR-2009	30-APR-2009

Parameters

Parameter Alias	Realm Name	Parameter Name	Parameter Type	Display Position	Event Key
Event ID	Event Instance Parameters	Event ID	Event ID -key	Detail	No
Item	Exception Context		Item	Header	Yes
Department	Exception Context		Dept	Header	No
Status	Exception Context		Text Field 1	Header	No
Description	Exception Context		Text Field 250	Detail	No

Event Type Parameters Screen

Field Description

Available Realms/All Parameter Types/Exception Context Buttons

This section of the dialog is a set of radio buttons for switching between editing parameters or purely parameter types. Your choice here affects what is displayed in the list in the middle center of the screen.

- Available Realms: Displays realms from which parameters may be selected and added to the event type.
- All Parameter Types: Displays a list of all parameter types in the system.
- Exception Context: Displays the parameters of a specific exception. However, only these parameters' types are actually of interest. Any parameter type can be added to an event. However, since parameter types play a critical role in exception mapping, the exception mapping context can be useful for restricting the list.

The parameter type list is queried each time the exception context is changed. It is also queried every time the radio group is set to either Exception context or All Parameter Types. Parameter names are blank when All Parameter Types is selected and use the exception parameter names when exception context is selected.

Available Realms List

The Available Realms section displays realms from which parameters can be selected and added to the event type.

Parameters List

The section in the top center displays the parameters of the currently selected realms or all parameter types or the parameters of the specified exception, depending on the radio group selection.

Event Key

This field helps determine to which exception type the event type can be mapped. The exception must possess the same parameter types as those parameter types to which the event type’s key parameters belong.

Display

Determines whether a parameter should be shown in the header or in the detail section of the Event Viewer. If header is chosen, the parameter will be a field that can be sorted in the Event Viewer.

Parameter Alias

An alias for the parameter. It is possible to add the same parameter twice using a different mapping each time. To distinguish these parameters from one another later, when building conditions, the option of creating an alias for each parameter is provided.

Unavailable Realms List

The unavailable realms list shows realms from which parameters could be selected *if* the appropriate driving parameters already had been added to the event type. Double-clicking a realm or pressing F9 displays a list of needed parameter types to be able to access an unavailable realm.

Parameters

The Parameters section shows the parameters already mapped to the event.

Parameter Mapping Link

The Parameter Mapping Link section shows the link basis by which a given parameter is mapped to the event. That is, it shows the mapping dependencies of the selected parameter. The mappings are displayed as the key parameters for the parameter’s realm, and the number of the event type parameter used to invoke each key parameter. To modify the parameter mappings, click the △ button.

Adding Parameters to Event Types

Any available parameter or parameter type can be added to the list of parameters on the event type by using the down (add) arrow. Event type parameters must have unique names. If no alias is specified, the parameter name is used unless it is a parameter type out of exception context. In this case, the parameter type name is used. When adding a parameter, the mapping for the parameter must be specified. If it can only be specified one way (based on existing parameter types and those required by the realm), then the mapping is performed automatically. Otherwise, the parameter-mapping screen is invoked.

Several special conditions apply when adding a parameter:

- Parameters that are added from this event realm require no special mapping.
- Parameter types do not require mapping.
- When automapping parameters, unique names must be derived. The Display attribute is set to Detail and the Event Key attribute is set to No.

Removing Parameters and Parameter Types from Event Types

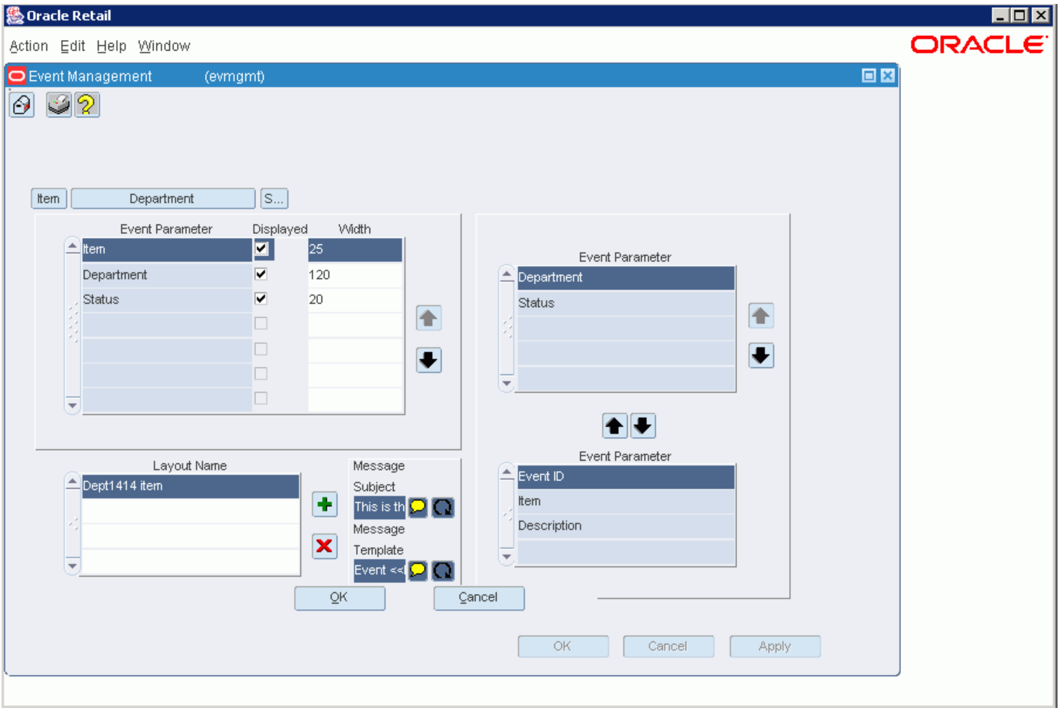
By pressing the up (remove) arrow, you can remove parameters from the event type. However, removing parameters/types has several implications. Specifically, when a parameter or parameter type is removed, the following must also occur:

- Actions mapped with parameters that cannot be overridden must be removed.
- States that have all actions removed must be removed from the state rules. If that means the default state is removed, all state rules must be removed.
- Conditions of attribute rules must be removed and the entire rule must be removed if no conditions remain and it is not the default (rule #0).
- Conditions of state rules must be removed, and if no conditions remain and it is not the default state rule, the entire state rule must be removed.
- Related event links must be removed if no parameter relation remains.
- Exception mappings must be removed if no parameter mappings remain.
- Conditions of closure rules must be removed.

If an attempt to remove a parameter affects dependent parameters, conditions, or event types, a warning is displayed asking whether you want to continue with the removal.

Event Viewer Layouts Screen

The Event Viewer Layout screen defines the layouts for instances of this event type when the events are displayed in the Event Viewer. Each layout can be used as the default layout for one or more Event States.



Item	Department	S...

Event Parameter	Displayed	Width
Item	☑	25
Department	☑	120
Status	☑	20
	☐	
	☐	
	☐	
	☐	

Event Parameter
Department
Status

Event Parameter
Event ID
Item
Description

Layout Name
Dept1414 item

Message Subject: This is th

Message Template: Event <<

Event Viewer Layout Screen

The ✚ button adds a new layout. The ✖ button deletes a new layout. You cannot delete a layout if it is selected as the default layout for any Event Type State.

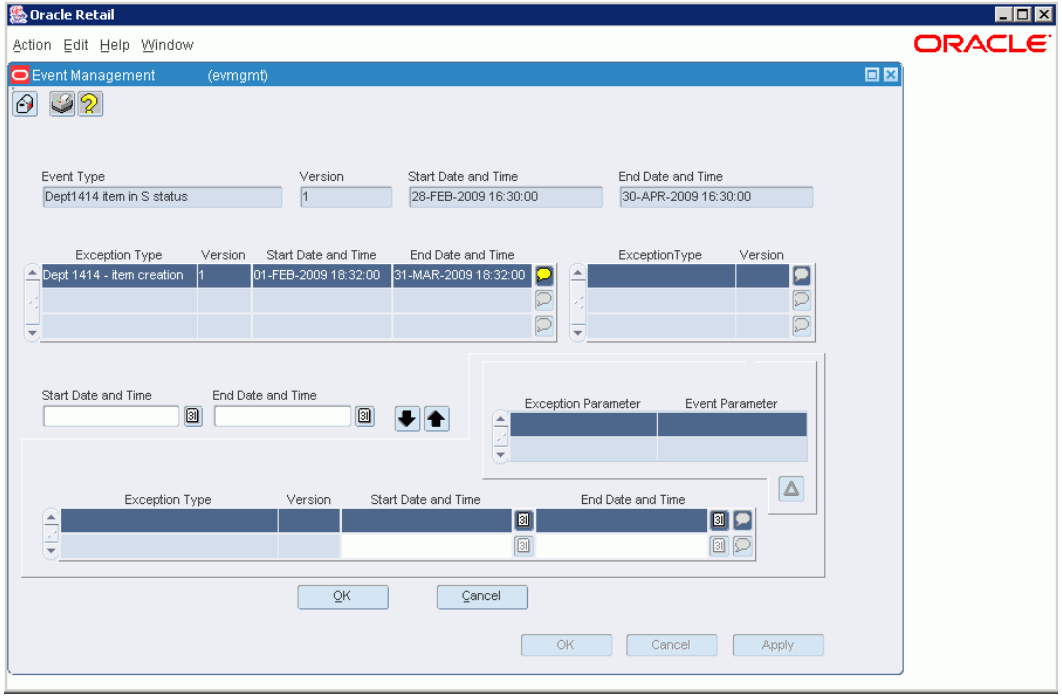
For each layout you can adjust the column width and order of the header columns, and determine which header columns will be displayed by default. As you change the layout for a header column, the Preview area at the topic of the screen shows how the parameter will look in the Event viewer. Users can adjust all of the settings in the Event Viewer.

You can also select which parameters will be displayed in the detail section and the order in which they will be displayed. The detail settings are not adjustable in the event viewer.

You can adjust the order of both the header and detail parameters by using the up and down arrows next to the appropriate block. For each layout you must specify at least one header parameter and one detail parameter to be displayed in order to exit this screen.

Event Type – Exception Type Link Screen

The Event Type – Exception Type Link screen is used to add, edit, and delete links between an event type and exception types.



Event Type	Version	Start Date and Time	End Date and Time
Dept1414 item in S status	1	28-FEB-2009 16:30:00	30-APR-2009 16:30:00

Exception Type	Version	Start Date and Time	End Date and Time
Dept 1414 - item creation	1	01-FEB-2009 18:32:00	31-MAR-2009 18:32:00

Event Type – Exception Type Link Screen

To link an event to an exception, the exception must be able to provide parameter mappings (by parameters of the same type) for all of the event's parameters that are declared as coming from an exception context. Optionally, additional parameters can be fed to the event from the exception, overriding whatever method the event would normally use to fetch those other values. For more information on this, see the Event Parameters Screen section of the Event Manager dialog.

Other conditions for triggering an event from an exception are that the active dates must overlap and the definition of the event being mapped must be complete. All of these factors determine whether an event will show in the available or unavailable mapping area, or not at all. Also, there is no simple window that can be popped up to explain that a mapping cannot be made as there is for parameters with the Needed Parameter Types window.

The screen is divided into several sections:

Field Description

Header

The header section displays information about the current event type. None of these fields are editable.

Available Exception Types

The Available Exception Types list shows exception types that are valid to link to the event type. These exception types have some overlap in their start and end dates with those of the event type, and for which the exception type has appropriate parameter types to map to the events key parameters. The comments button in this section displays the exception type description in a pop-up comments window.

Unavailable Exception Types

Unavailable exception types are like available exceptions with respect to validity and an existing overlap period, but they are missing one or more of the event’s key parameter types. Double-clicking or pressing F9 on an unavailable exception type displays a list of needed parameter types.

Down Arrow and Up Arrow Buttons

The ⬇ button adds an available exception type to the list of linked exception types. The ⬆ button removes the selected exception type from the linked exception types list and returns it to the list of available exception types.

Active Link Period

The active link period is an editable pair of dates that are used when the event type and exception type are linked to set the active period of the mapping. These dates are used with the ⬇ button.

An event type can be mapped to an exception type more than once and the validation for these dates is somewhat complicated. In every case, it is necessary to account for the fact that a null end date acts like an infinity end date. The start and end dates must be between the minimum of the event type and exception types end dates and the maximum of its start dates. The start date is further restricted in that it must be later than the current date. Finally, if the exception type is already mapped to the event type, the start and end dates must define a period that does not overlap with an existing mapping. If valid dates have been entered, the link mapping screen is invoked to complete the process of specifying how the key event type parameters will be populated from the exception type.

Linked Exception Types

The Linked Exception Types section shows already mapped exception types. The Link Period and Exception Active selections toggles the display between link period dates and exception type activity dates.

The only editable fields in this list are the link start date and end dates. These dates are subject to the date rules for the Active Link Period, and the Start Date is editable only if the date is later than the current date. Once the start date had passed, only the end date is editable; in this case, the end date cannot be set earlier than the current date.

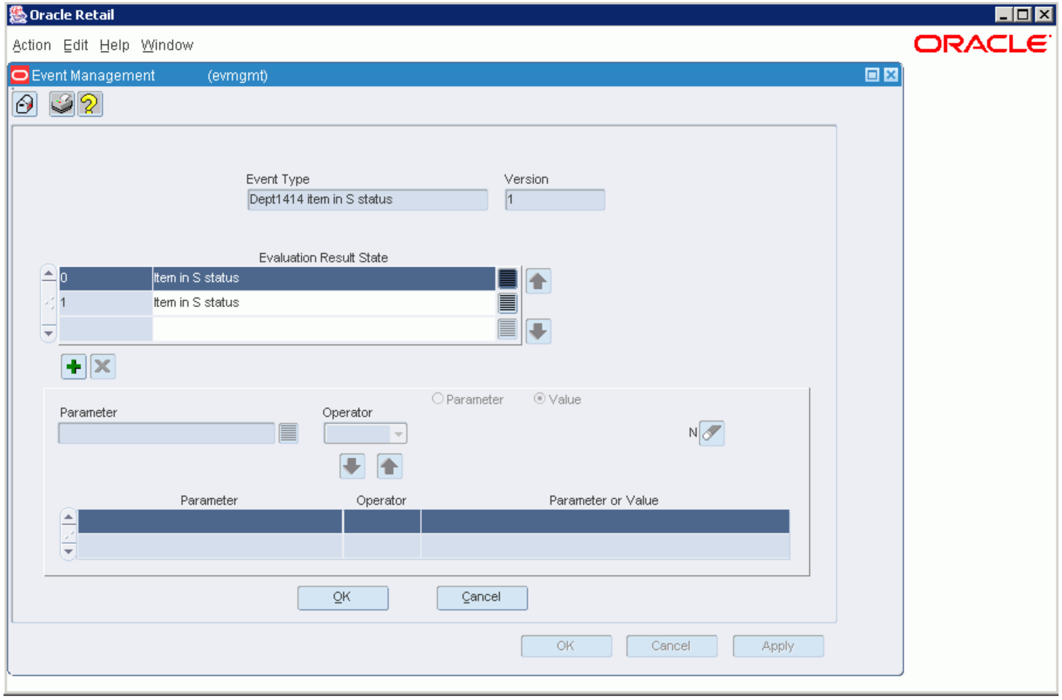
When editing already linked exceptions, if you try to choose a date outside the start and end dates, an error is displayed. Only exceptions that have not entered the active link period (that is, exceptions for which the mapping start date has not yet passed) can be removed. If the end date has passed, both dates are read-only.

Parameter Mapping

The parameter mapping section shows which event type parameters are mapped from which exception type parameters. The mapping is editable only before the start date of the linkage has passed. Clicking the Δ button displays the Exception-Event Link Mapping screen, where you can edit the link mapping for the selected event type-exception type pair.

Event Type – State Rules Screen

The Event Type – State Rules screen is used to add, edit, and delete rules to assign to an event's state, and to change the conditions that make up those rules. The result of a set of rules being true is that a state is assigned to the event.



Event Type: Dept1414 item in S status
Version: 1

0	Item in S status
1	Item in S status

○ Parameter ● Value

Event Type – State Rules Screen

When the screen is launched, if a default rule does not already exist, one is added by clicking on the Add in the +ΔX. A state must be assigned for each rule. The default state cannot be deleted or moved, though buttons allow changing the order of all other rules. Except for the default state, all rules must have at least one condition. When you click the OK, Next, or Back button on this screen, the rules are checked to determine whether they have at least one condition. This check is done by looping through all rules and offering to delete any but the default that do not have conditions.

The screen is divided into several sections, as described below:

Field Description

Event Type Information

The top of the screen displays the name and version number of the event type associated with the state rules.

State Assignment Rules

The second section lists the states that will be assigned and the sequence in which the rules to assign those states will be tested.

Conditions

This section is used for adding, editing, and deleting the conditions that make up the state rule. It consists of a condition editor, controls for moving conditions into and out of the condition editor, and a list of conditions currently defined for the state rule.

Condition Editor

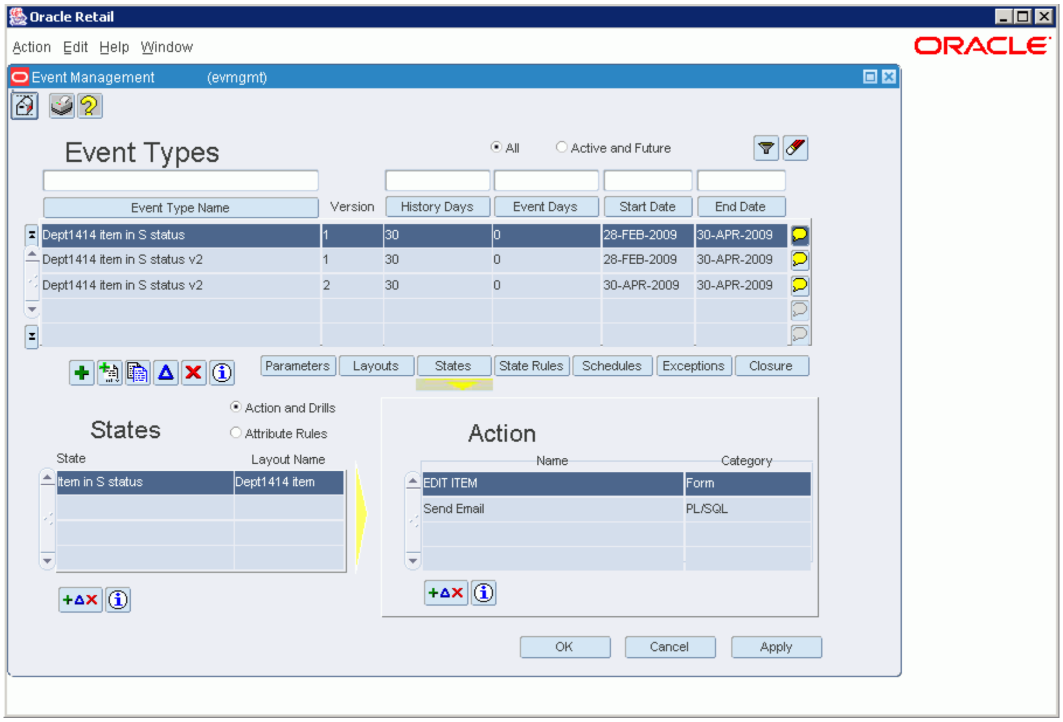
The condition editor is for defining conditions that make up the state rule.

Event Type Rule Condition List

This section lists the conditions of the rule selected in the State Assignment Rules list.

Event Type States Screen

The Event Type States screen and the set of screens associated with it is used to maintain or add, edit, and delete, the states names of an event and the layouts of the state.



● All ○ Active and Future

Event Type Name	Version	History Days	Event Days	Start Date	End Date
Dept1414 item in S status	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	2	30	0	30-APR-2009	30-APR-2009

● Action and Drills ○ Attribute Rules

State	Layout Name
Item in S status	Dept1414 item

Name	Category
EDIT ITEM	Form
Send Email	PL/SQL

Event Type States Screen

As the data set of an event refreshes, it may be that the data set changes in a way that the event is used to track several stages of a process. For each state a layout must be selected. It will be used as the default layout when events in that state are displayed in the event viewer.

Typically at each step in a process, different actions are appropriate. Also, the rules used to assign the events priority, routing, and any automatic actions may be different at each step of the process. States provide the functionality to support tracking multi-stage processes.

From the Event Type States screen, you can create the event states. On subsequent screens, you define actions to be mapped to the states, and attribute rules defined to dynamically assign priority, routing method, assignment group, and automatic action attributes that apply whenever the event is re-evaluated.

The screen is divided into three sections: States, Actions, and Attribute Rules.

Field Description

States

The States section displays the state names of the event. State names are editable at any time.

The ✚ button adds a new state.

The ✕ button deletes the selected state.

Actions

The Actions section displays the actions of a given state. Using the +△✕ button, you can edit the actions.

Attribute Rules

The Attribute rules section displays the attributes that could be assigned to the event and the order in which the underlying rules used to assign the attributes are evaluated. Using the +△✕ button, you can edit the attribute rules.

Adding and Deleting States

States can be added or deleted directly on this screen. Deleting a state that is used in a state rule results in deleting the rule as well. If this implies deleting the default rule, then all exception mappings must be removed from the event. A different warning should be given to you in each case, asking whether you would like to proceed.

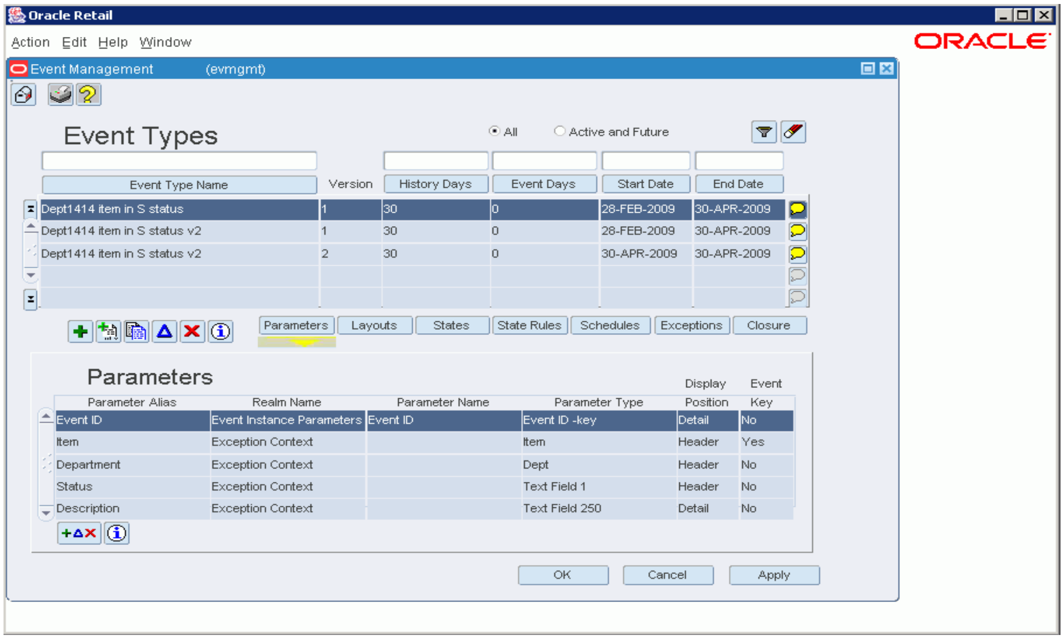
When you click OK on the dialog, Next, or Back when using the event wizard, a message displays indicating that any states that do not have at least one action and a default attribute rule will be deleted.

Related Screens

- Event Type State – Action
- Event Type State – Action Parameter Mapping
- Event Type State – Event Attribute Rules
- Event Type State – Event Attribute Rules: Assignment Groups

Event Types Summary Screen

The event summary screen displays all the exception types defined in your system. All the information in this screen is read-only.



Event Types

● All ○ Active and Future

Event Type Name	Version	History Days	Event Days	Start Date	End Date
Dept1414 item in S status	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	2	30	0	30-APR-2009	30-APR-2009

Parameters

Parameter Alias	Realm Name	Parameter Name	Parameter Type	Display Position	Event Key
Event ID	Event Instance Parameters	Event ID	Event ID -key	Detail	No
Item	Exception Context		Item	Header	Yes
Department	Exception Context		Dept	Header	No
Status	Exception Context		Text Field 1	Header	No
Description	Exception Context		Text Field 250	Detail	No

Event Summary Screen

Field Description

Filtering the View of Events

This view may be filtered. Using the radio buttons and the button, you can choose to display all events, including expired events, or only active and future events. The button clears any active filter.

Event Types List

The top part of the screen lists all the event types in summary form. All the information in this screen is read-only. From this summary list, you can create new exceptions and edit existing exception definitions.

Buttons

Several action buttons are available on this screen that allow creation, editing and deletion of event types.

From left to right the buttons are used to:

The Wizard button starts the wizard for creating a new event type. This button is active at all times.

The button creates a new event type manually. This means you work through the screens of the Event Manager yourself to define the new event type, rather than using the wizard. This button is active at all times.

The button creates a new version of an existing event type, and establishes a date and time for the new version to take effect. The version button is available at all times that an event is displayed in the summary list.

The 📋 button clones, or copies, an existing event type.

The Δ button edits an exception type that is not yet active.

The ✗ button deletes an expired or not-yet-active event type.

The ⓘ button displays additional header information about an existing event type.

Event Details

The lower half of the screen shows details about the currently selected event in the top half of the screen, depending on which button is selected in the center of the screen: parameters, states, state assignment rules, schedules, exceptions, or closure rules.

Parameters

The first stacked detail screen for the bottom half of the summary view is the parameters screen. This screen is the default when the form is first launched. This screen shows the parameters for the selected event type and has a button for accessing the add/edit/delete parameters screen. This button is available for the same records that the edit button for the header is available for and disabled for past and active events. The info button is available whenever the version button is available.

Layouts

The Layouts detail lists the layouts that have been set up for displaying instances of this event type in the Event Viewer.

States

The States detail displays states and state rules defined for the event type. The +Δ✗ button displays the state editing screen. The ⓘ button is enabled whenever the versioning button is available. The Δ button is enabled only for future exceptions.

Clicking the Actions button displays all of the actions associated with the event state on the right side of the state detail.

Clicking the Attribute Rules button displays the possible results of conditional evaluations in terms of priority assignment, routing method, automatic action and, behind the comments button, group assignments.

State Assignment Rules

The state assignment rules detail shows both the evaluation sequence and the state that is assigned if all of the conditions are true.

Schedules for Revalidation

The Schedules for Revalidation detail shows which schedules are attached to the event type, and which of them are active now, in the future or in the past. These values for active are updated using the view the schedule block is based on each time the query is performed.

Schedules can be added/edited/deleted for all active and future active event types. The comments box displays the schedule description. The ⓘ button is available when the selected event type is not blank.

Exception Types

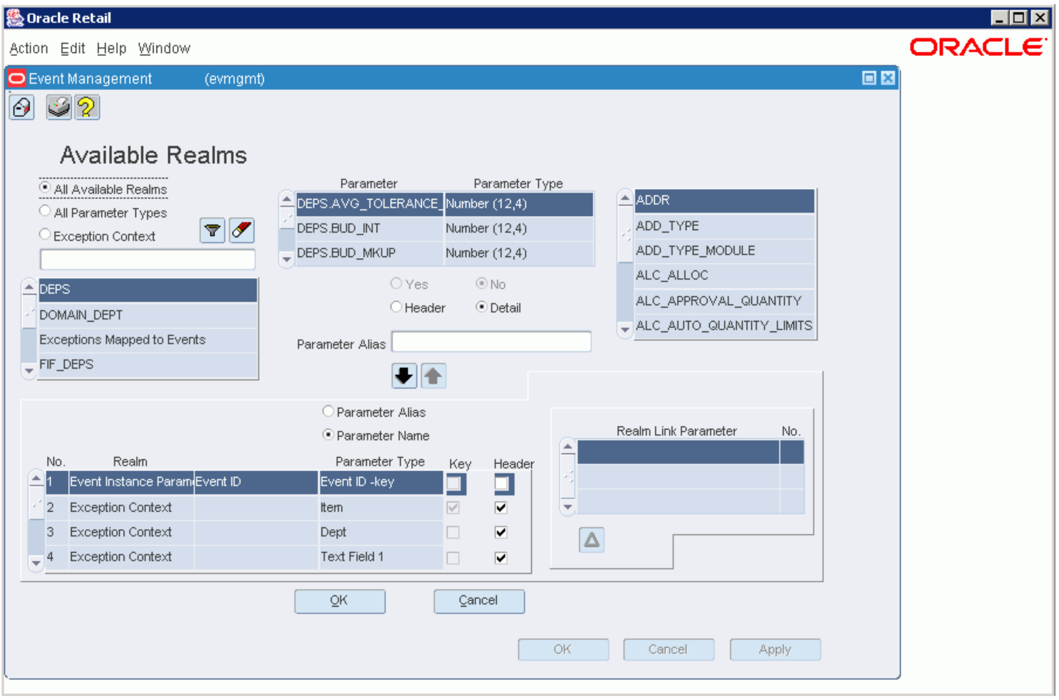
The Exception Types detail shows the exception types that are linked to the event type and the window when the link will be valid. Like schedules, exception type mappings may be edited in for all event types that are active and future active. The comments button displays a comments window with a description of the exception type.

Closure Rules

The Closure Rules detail shows the conditions evaluated to determine whether the event is closed. The evaluation sequence is just a rule number and the order of evaluation is not important since if any of the sets of conditions is true the event will be closed. Closure rules cannot be edited for active exceptions.

Needed Parameter Types Window [evmgmt]

The Needed Types window is displayed in several situations when defining and editing event types:



● All Available Realms
○ All Parameter Types
○ Exception Context

DEPS
DOMAIN_DEPT
Exceptions Mapped to Events
FIF_DEPS

Parameter	Parameter Type
DEPS.AVG_TOLERANCE_	Number (12,4)
DEPS.BUD_INT	Number (12,4)
DEPS.BUD_MKUP	Number (12,4)

ADDR
ADD_TYPE
ADD_TYPE_MODULE
ALC_ALLOC
ALC_APPROVAL_QUANTITY
ALC_AUTO_QUANTITY_LIMITS

○ Yes ● No

○ Header ● Detail

Parameter Alias

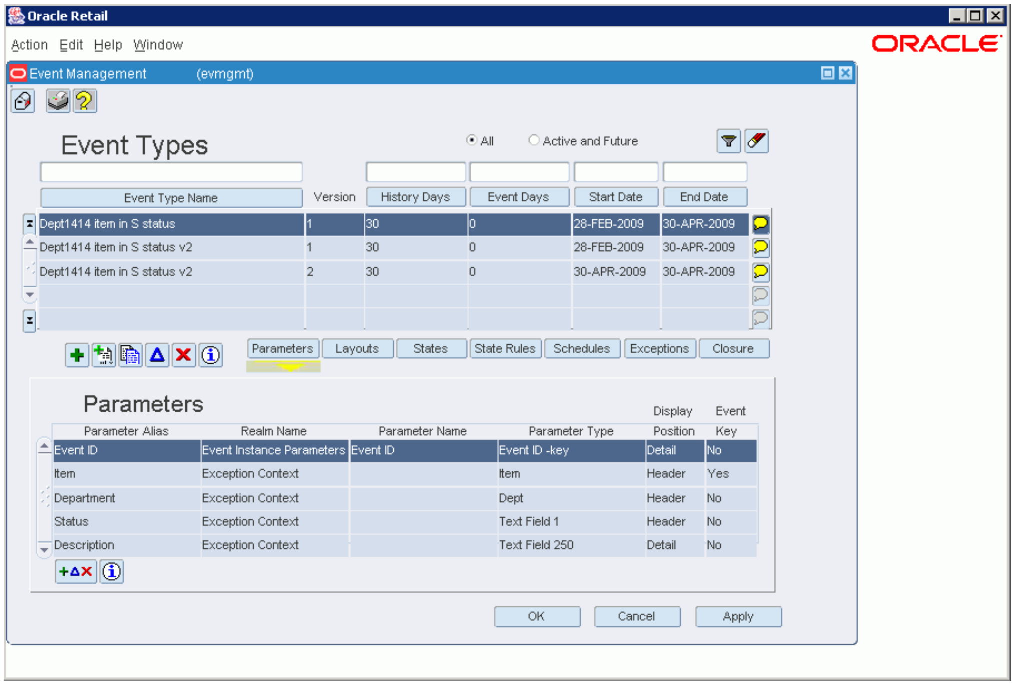
○ Parameter Alias
● Parameter Name

No.	Realm		Parameter Type	Key	Header
1	Event Instance Param	Event ID	Event ID -key	□	□
2	Exception Context		Item	☑	☑
3	Exception Context		Dept	□	☑
4	Exception Context		Text Field 1	□	☑

Realm Link Parameter	No.

Needed Types Window

- When defining event parameters and parameter types, this window is available from the Event Type Parameters screen. In this case, this window shows a list of additional parameter types needed to make that realm available.



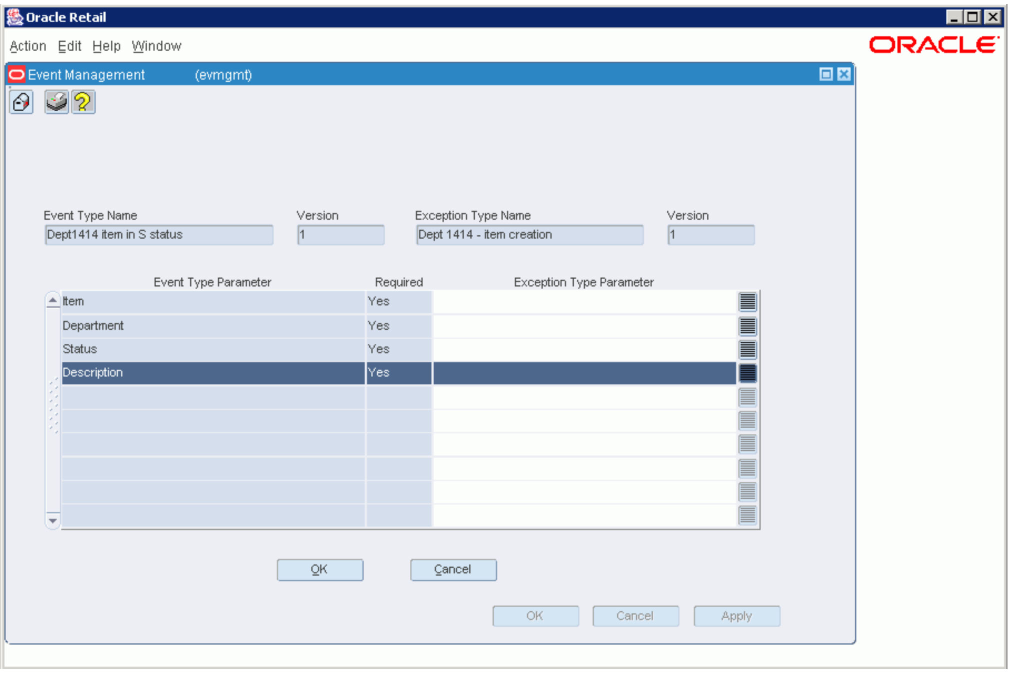
Event Type Name	Version	History Days	Event Days	Start Date	End Date
Dept1414 item in S status	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	1	30	0	28-FEB-2009	30-APR-2009
Dept1414 item in S status v2	2	30	0	30-APR-2009	30-APR-2009

Parameters

Parameter Alias	Realm Name	Parameter Name	Parameter Type	Display Position	Event Key
Event ID	Event Instance Parameters	Event ID	Event ID -key	Detail	No
Item	Exception Context		Item	Header	Yes
Department	Exception Context		Dept	Header	No
Status	Exception Context		Text Field 1	Header	No
Description	Exception Context		Text Field 250	Detail	No

Event Type Parameters Screen

- When defining event-exception links, this window is available from the Exception Type - Event Type Link Parameter Mapping screen. In this case, the window shows which parameters are needed on an exception to make a given event available for mapping.



Event Type Name: Dept1414 item in S status
Version: 1
Exception Type Name: Dept 1414 - item creation
Version: 1

Event Type Parameter	Required	Exception Type Parameter
Item	Yes	
Department	Yes	
Status	Yes	
Description	Yes	

Exception Type – Event Type Link Parameter Mapping Screen

Condition Editor

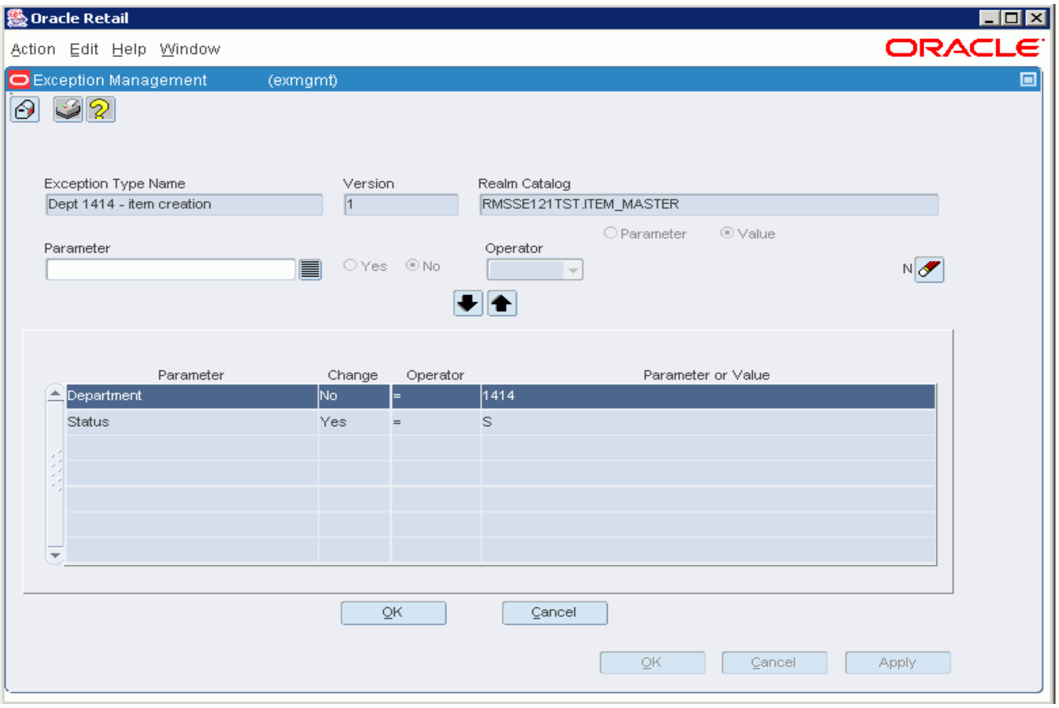
Several exception and event management screens involve defining conditions associated with the exception or event, including:

- Exception Type Conditions
- Event Type State – Event Attribute Rules
- Event Type – State Rules
- Event Type – Closure Rules

Following is a description of how to create and edit conditions on these screens.

How the Condition Editor Works

In all the screens that involve conditions, you use the Condition Editor to add, edit, and delete conditions showing that a parameter changes or that a relationship exists between two parameters or a parameter and a value. The basic layout of a condition is a comparison between a parameter and a value or another parameter, using an operator such as = or >. Exception definitions have an additional setting for the condition, Monitor Change, which specifies whether you want to be notified when the parameter change occurs. It is also possible for a parameter to change and have a relationship condition simultaneously.



Exception Type Name: Dept 1414 - item creation
Version: 1
Realm Catalog: RMSSE121TST.ITEM_MASTER
○ Parameter ● Value
Parameter:
○ Yes ● No
Operator:

Parameter	Change	Operator	Parameter or Value
Department	No	=	1414
Status	Yes	=	S

Condition Editor Screen

All conditions must be between a parameter and a value or two parameters, but never between two constant values. At least one parameter must be chosen, so when the condition editor is empty only the first parameter field and its LOV are enabled. Once a parameter is selected (only varchar2, date, number, and Boolean data types are available in the LOV), the other fields are enabled. When the first parameter field is cleared, the condition editor reverts to the original state.

Once the condition is defined, you add it to the list of exception or event-rule conditions below the editor by clicking the ⬇ button. When you need to change a condition, you click the ⬆ button to move the condition into the editor, and when done changing the condition, you click the ⬇ to move it back into the condition list for the exception or rule.

The conditions defined for an exception or event is treated as logical AND conditions. That is, when a monitor detects candidate exception instances, or when an event is evaluated against the rules defined for it, the exception or event is tested against all of these conditions. If they are not all true, the candidate is discarded and ignored without further processing. For example, if all the conditions for candidate instance are true, then the candidate instance becomes an actual exception instance, at which time linked events are created.

Field Description

The condition editor is displayed above the list of existing conditions, and consists of the following fields and controls.

Parameter

The parameter on which the condition is based. This parameter is compared to a value or another parameter. Click the LOV button to display allowable parameters for the exception or event.

The parameter chosen to drive the condition determines:

- Which operators are available
- For exceptions, whether the exception can be monitored online
- Whether other parameters or special values can be used for the comparison created.

All parameters allow comparison to NULL values. Only parameters from the monitored realm, if the realm can be monitored in real time, allow change detection. Some parameters have a constrained set of valid values that can be used for comparison, such as the store number parameter only allowing valid store numbers. Also, the data type of the parameter, character, Boolean, date, or number, determines which comparisons can be made to other parameters (that is, those of the same data type). Actually, many times such comparisons are not meaningful, unless the two parameter types are the same, such as comparing two store numbers, as opposed to comparing a dollar amount and a store number.

Monitor Change (for Exception Type Conditions Screen Only)

The monitor change radio button indicates whether or not the condition will be real-time monitored. In other words, do we want to monitor the field changing to the entered value (for example, order status changing to 'S') or all fields that contain that value (for example, order status = S).

Monitoring a change is only possible on parameters that come from the monitored realm, and even then only on a realm with active, query-able data. So, monitoring change is impossible for external systems. For internal systems, however, whether an exception monitors transactions online or through a nightly batch sweep depends on whether at least one of the conditions is set to monitor a change.

Operator

The operator used to compare the parameter with a value or another parameter.

For Exceptions only, an operator is required for all conditions except for those in which a change is being monitored.

Available operators depend on the data type of the specified parameter. Specifically, text parameters do not support inequality operators. In addition, any exception parameters with a Long or Long Raw Data type are not available for defining conditions, since they cannot be used to perform comparisons anyway. The only reason to add such parameters to an exception is to pass them from a non-queried data source through to an event as additional information to present to users in the event interface.

For all data types, if you use the operators IS and IS NOT, the value must be specified as NULL.

For varchar2 and Boolean data types, the only other options are = and !=.

For date and number data types, the operators <, <=, >, >=, =, and != are also available.

Character and Boolean parameters allow = and != operators.

Numbers and dates allow inequality operators as well.

Parameter or Value Radio Buttons

Specifies whether the selected parameter is being compared to a parameter or a parameter value. The default setting is Value, with the Value field set up for the appropriate data type.

A parameter can only be compared to another parameter when a parameter of the same type exists on the exception or event (that is, if there are available parameters to compare the parameter against). Otherwise, the parameter must be compared to a value.

Parameter or Value

This field specifies the parameter or value to which the selected parameter is being compared.

Depending on the data type of the parameter, the controls on this field change. In some cases, when making a value comparison, a list item, a date button, or an LOV button is available to help you choose an appropriate value. For example, for dates, a date button is displayed to enable entering a date. For parameters with a code type, clicking the LOV button displays a list of appropriate code values from which you can choose.

Down Arrow Button

The ⬇ button adds the condition in the condition editor into the condition list for the exception or rule. The fields of the condition editor remain filled with the values for the condition. You can either continue to define more conditions on the same parameter, or click the ⌫ button to clear the fields.

Up Arrow Button

The ⬆ button removes the condition from the condition list, and puts it in the condition editor. Click this button to edit or delete existing conditions.

Clear Button

The ⌫ button clears the condition editor fields.

Conditions List

Below the condition editor is a list of the conditions currently defined for the exception or rule. The conditions defined for an exception or event are treated as logical AND conditions, as described earlier in this topic.

Adding a Condition

1. Choose a parameter from the list.
2. (For exceptions only) Specify whether you want to be notified of the change to the parameter.
3. Choose an operator from the list.
4. Choose whether you want to compare the parameter to a value or another parameter.
5. Specify the value or parameter to which the parameter will be compared.
6. Click the ⬇ button. The condition is added to the condition list below the editor.

Editing a Condition

1. Select the condition from the condition list.
2. Click the ⬆ button to move it into the condition editor.
3. Modify the definition as needed, for example, changing the operator or the value to which the parameter is compared.
4. When done, click the ⬇ button to add the condition back to the condition list for the exception or rule.

Deleting a Condition

1. Select the condition from the condition list.
2. Click the ⬆ button to move it into the condition editor. The exception condition is left in the edit section, since the only way to edit a condition is remove it, change it, then add it back to the exception conditions.
3. If you want to remove an exception condition and then add another, remove the condition, and then change the first parameter. Or, click the ✎ button to clear the fields and start with a new exception condition.

5

Schedules for Exception Scanning and Event Revalidation

Schedule Dialog [schedule]

From the Schedule dialog, you can add, edit, and delete schedules.

Filtering Buttons

You can filter the view of schedules by using the filtering buttons and fields above the schedule list.

Filter Button

The ⏷ button applies the filter values specified in the filter fields and re-queries the list of schedules to restrict the view to those meeting the filtering criteria.

Clear Filter Button

The ✎ button clears the filtering criteria at the top of the column, and returns the view of schedules to the normal view.

Filter Fields

These fields allow you to enter filtering criteria for the view of schedules.

The Schedule List

Below the filtering buttons is a list of schedules currently defined in your Active Retail Intelligence system. This list has the following fields.

- Schedule Name: The name of the schedule, such as “General batch scans.”
- Schedule Description: Tells what kind of schedule and its signal text, single occurrence time, or repeat frequency details.
- Excl**usion Window:** Identifies the exclusion window during which the schedule will not execute. If the first time is greater than the second the exclusion is from the first time through midnight until the second time on the following day.

The ✚ button adds a schedule.

The ✖ button deletes the selected schedule, unless it cannot be deleted because it is associated with either an exception or an event.

A schedule can only be deleted if it is not currently in use by an active exception type definition or an event type definition. If the schedule you are attempting to delete is in use, an error message is displayed, listing the names of the event or exception definitions that are using the schedule. Once those definitions have had their schedules reassigned or cleared, the schedule can be deleted. Assigning and reassigning schedules to exceptions and events is performed in the mapping screens for the Exception Manager and Event Manager dialogs.

Schedule Details

The bottom part of the dialog is used for adding schedules

Schedule Name

The name of the schedule.

Schedule Type

Repeating, Once or Signal Driven.

Execute-on Signal

For a Signal Driven schedule this is the text string that must be sent to the accept signal API to execute the schedule.

Execute Once At

Execute once at indicates the date and time when a schedule will execute. An exclusion window used with this type of schedule forces the schedule to wait until after the exclusion window. This may seem like something that would not apply to an execute-once schedule, but an exclusion window can act as a safeguard against delayed execution (due to the scheduler not being active) occurring at an undesirable time (such as during the RMS batch window, for example).

Repeat Every

Select a whole number of hours, days, weeks, or months. Depending on the interval selected additional fields for selecting the day of the week or month and the hour and minute of the day or minutes past the hour will become enabled for specification.

Exclusion Window

This window is available for load balancing of Active Retail Intelligence exceptions or simplifying otherwise complex scheduling processes. When incrementing to the next execution date, repeating schedules increment to the next valid date beyond the exclusion window. Signal driven and execute-once schedules that are supposed to execute immediately or during the exclusion window execute as soon as the exclusion window ends. Repeating schedules execute at the exclusion window end if they are delayed (due to an inactive schedule process, not typical in production) from their normal time until after the exclusion window begins.

OK

Commits changes and returns to the summary view, re-enabling the schedule dialog main buttons that had been disabled by adding a new schedule.

OK + Repeat

Posts the added schedule and begins the addition of a new one.

Cancel

Discards the schedule currently being added and re-enables the schedule dialog main buttons.

Schedule Dialog Confirm and Cancel Buttons

OK

Commits all changes made in the Schedule dialog and closes the dialog.

Cancel

Cancels all changes made in the Schedule dialog since the last Apply and closes the dialog.

Apply

Commits all edits made in the Schedule dialog, leaving the dialog open.

Schedules for Exception Scanning and Event Reevaluation

Schedules serve two purposes:

- You can attach a schedule to a batch (periodic) exception type that governs when batch scanning of the exception type occurs. Note that exception scans can be done throughout the day so that the Active Retail Intelligence administrator can do some manual load balancing of Active Retail Intelligence processes.
- You can attach a schedule to an event type that defines how frequently events of that type are automatically reevaluated by the system. Volatile events that use data that is changed often or by many users or components of the system may benefit from periodic reevaluation both to save users having to reevaluate manually and to reduce their alert inbox traffic by removing resolved events before the user even sees them. As a rule events shouldn't be terribly volatile, but there may be some applications for it.

Active Retail Intelligence has a schedule process that every few minutes checks all schedules and, for those that are due to execute (or past due if the Active Retail Intelligence schedule process has been inactive for a period of time). Active Retail Intelligence then finds the associated exceptions that need scanned and events that need reevaluated and executes the scans and reevaluations accordingly.

Schedule Types

There are three types of schedules:

- Recurring: This kind of schedule starts at a specified day and time, and then recurs every “x” hours, days, weeks or months. A schedule that occurs on the 29th, 30th, or 31st of the month will not execute in a month that does not have a day with that number. After execution the schedule is incremented according to the repeat interval until the next execution date is incremented to some date and time later than the current time. By using multiple recurring schedules together (because an exception or event can have multiple schedules) a sophisticated recurring schedule pattern can be created from these simple schedules.
- Once: A schedule can be set up to be executed once at a specific day and time. Provided that the Active Retail Intelligence scheduler is running (it should run constantly in production) the schedule will be executed within a few minutes of the requested time. If the scheduler is not running when the schedule is due, it will execute as soon as the scheduler is restarted.
- Signal Driven: A signal driven schedule accepts a signal by calling the PL/SQL application programmer interface “ARI_SCHEDULE_SQL.ACCEPT_SIGNAL” and sending the exact signal text that the schedule requires. This will cause the schedule

to set the next execute date to “right now”, and so, provided the scheduler is running it will run immediately, or as soon as the scheduler is restarted.

This API allows linking of one process to another. Consider that you may want to monitor for a certain exception only after some daily or weekly process occurs, and always after it occurs. However, it may be that the process doesn't occur on a rigorous schedule, so you can modify the process to call this API when it completes and so thereby link Active Retail Intelligence to the other process.

Using Signals with Schedules

In a schedule definition, a signal is simply a text field passed to a method defined in the scheduler package, such as “nightly batch run completed.” Using a signal for a schedule, you can have a simple SQL*Plus command send the signal from a shell script as:

```
SQL>execute scheduler.send_signal(‘nightly batch run completed’)
```

It is the responsibility of the business analyst to keep the signal text consistent between the external processes and the internal schedules, as well as to coordinate the text messages used in internal definitions. That is, if two exception scans are driven from the ‘nightly batch run completed’ signal, it is the analyst’s responsibility to make sure the text strings are properly specified in the exception schedules.

Specifying an Exclusion Window

Schedules have an option of setting an exclusion window. An exclusion window applies to the schedule and thus to exception and events using the schedule. For events, an exclusion window means that an event will not reevaluate, at least not based on the schedule, during that period. For an exception, it means an hourly scan, for example, will be hourly but not during the exclusion window. A signal sent during an exclusion window will set the next execution date to the end of the exclusion window, but will not execute immediately.

You can use an exclusion window to keep an event out of the way during periods of heavy processing on related data. For example, you could disable the reevaluation of existing low-inventory events during nightly point-of-sale updates. The reevaluation of the event is then forced when the event emerges from the exclusion window. Exclusion windows don't apply to reevaluations in response to user action.

6

Language Translation

Language Translation Window [aritrans]

The Language Translation window allows you to translate one or more values of the same type from a primary language into a target language.

The Internationalization function provides support for organizations with a requirement for two or more languages. A user can translate and view most descriptive text into another valid language using the tl_lang form. Alternately, the product is shipped with the ability to run multilingually any of the eight supported release 12 languages: French, German, Spanish, Japanese, Korean, Chinese Simplified, Chinese Traditional, and Brazilian Portuguese.

All descriptive text fields should be entered first in the primary language. If a value has been translated to a language that matches the preferred language of the current user, it is displayed in the user's preferred language. If it has not been translated, the user sees the information in the primary language. These translations are supported in the end user interfaces (Event Viewer, Alert Viewer), but not in the administrative forms which are single language (very few users will have admin access).

If an organization supports a multiple-language environment any of the eight supported languages can be made to correspond with each the users' ids. (One primary language per user id. This language setting is set on the ari_user_attrib table.)

Field Descriptions – Search Window

Action

Select the task that you want to perform.

- To translate elements of the same type, select **New**.
- To view all translatable elements of the same type, select **View**.
- To edit elements of the same type that have already been translated, select **Edit**.
- To edit all translatable items of the same type, select **Edit All**.

Language to Translate Into

Enter the ID of the destination language, or click the LOV button and select a language. This is a required field.

Description Type to Translate

Click the LOV button and select a data element. This is a required field.

Selection Criteria From

Enter the primary language string to limit the query. This field is used to limit the lower range of values displayed. This field is optional.

Selection Criteria To

Enter the primary language string to limit the query. This field is used to limit the upper range of values displayed. This field is optional.

Field Descriptions – Edit/View Window

Selected Element

Displays the string to be translated in case it is too long to be easily read in the multi-record block. This field is read-only in all modes.

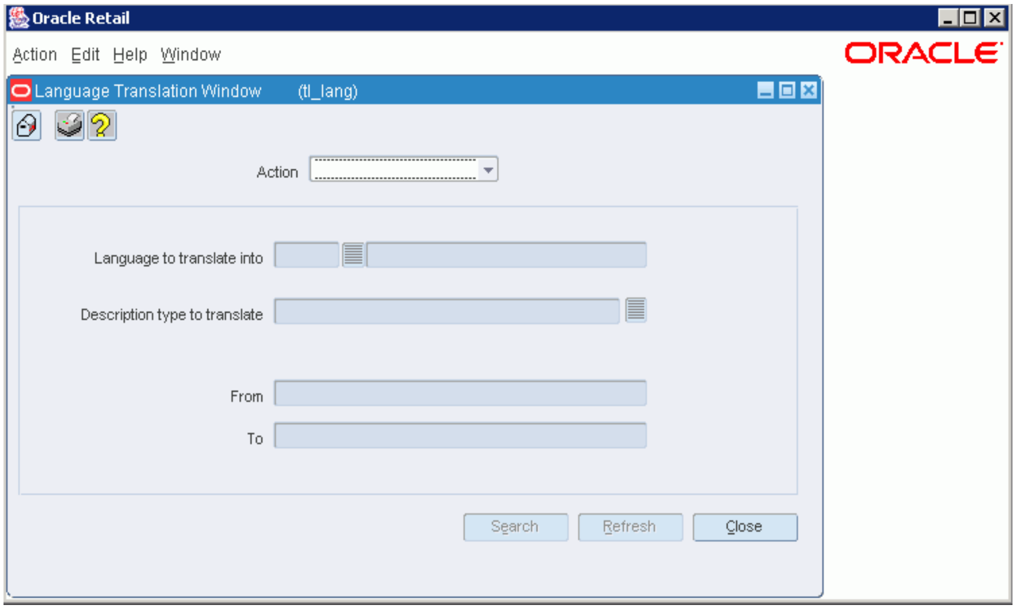
Destination Language

Enter the corresponding translation, the translated value, for each string displayed in the primary language column.

- Translate values into a target language
- Edit translations in a target language
- View translations in a target language

Translate Values into a Target Language

1. Access the Language Translation window. The default location in the RMS main menu is Control > System > Language Translation. Contact your system administrator for details on accessing this form in your own application. To access it from the ARI application, in the Main Menu click on ARI and select Translations from the Menu options. The Language Translation Window is displayed.


Oracle Retail
Action Edit Help Window
ORACLE
Language Translation Window (tl_lang)
Action
Language to translate into
Description type to translate
From
To
Search
Refresh
Close

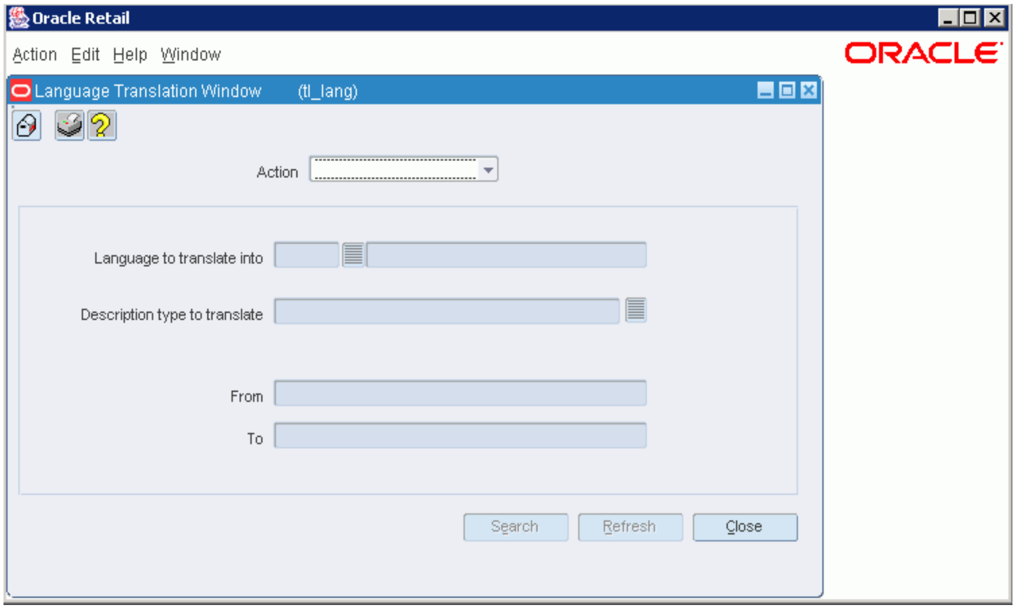
Language Translation Window

2. In the Action field, select New.
3. In the Language to Translate Into field, enter the ID of the destination language, or click the LOV button and select a language.
4. In the Description Type to Translate field, enter the ID of the type of description to be translated, or click the LOV button and select a description type.
5. To restrict the search, enter the range of elements that you want to include in the translation in the From and To fields. These entries are in the primary language. These fields are optional.
6. Click the Search button. The elements for the selected description type are displayed in the left field of the table in their primary language.

7. In the right field of the table, enter the translation for each item in the left field.
8. Click OK to exit.

Edit Translations in a Target Language

1. Access the Language Translation window.
   The default location in the RMS main menu is Control > System > Language Translation. Contact your system administrator for details on accessing this form in your own application. To access it from the ARI application, In the Main Menu click on ARI and select Translations from the Menu options. The Language Translation Window is displayed.


Oracle Retail
Action
Edit
Help
Window
ORACLE
Language Translation Window (tl_lang)
Action
Language to translate into
Description type to translate
From
To
Search
Refresh
Close

Language Transition Window

2. In the Action field, select:
   – **Edit** to edit existing translations, or
   – **Edit All** to create new and edit existing translations for the selected description type.
3. In the Language to Translate Into field, enter the ID of the destination language, or click the LOV button and select a language.
4. In the Description Type to Translate field, enter the ID of the type of description to be translated, or click the LOV button and select a description type.
5. To restrict the search, enter the range of elements that you want to include in the translation in the From and To fields. These entries are in the primary language. These fields are optional.
6. Click the Search button. The elements for the selected description type are displayed in the left field of the table in their primary language.
7. In the right field of the table, enter or edit the translation for each item in the left field.
8. Click OK to exit.

Scheduled Batch Monitored Processes

Some processes are not driven by transactions so they cannot be monitored online. These processes are monitored nightly after all of the POS uploads and other batch processes have been run, but before the date is advanced to the next working day. Essentially, they run once a day at the very end of the day, which is a factor to consider when setting the exception conditions in terms of how many days old something is. Note that, as with online monitoring, batch monitoring is available for internal data systems such as RMS and Active Retail Intelligence.

Time is a special consideration since exceptions that cannot be monitored based on a transaction occurring usually involve such things as data sitting on a table with nothing happening to it, which is of interest only because of the passage of time. An example of a date function computing the number of days between two date parameters is in the topic on defining functions. That function is one of the primary factors in defining at least one condition of many batch processed exceptions.

Aside from no parameter being set to monitor a change, and the frequent use of a date difference function, batch processes are defined similarly to online processes (see that section). In some cases it is unclear, for performance reasons (as discussed in the Overview), whether an internal system exception should be defined as online or batch.

It is unclear because either solution could have performance issues, so testing should be done when the key table to monitor is an extraordinarily large one. However, as a rule, if performance is acceptable with online monitoring and the exception can be defined appropriately, try that definition first.

Example

An example of when trying online first is not even an option is the case of monitoring purchase orders that have been in Submitted status for a certain period of time. From a timing perspective, if a purchase order is submitted on a given day, such as the first of a month, and you want to know when it is over seven days old, the purchase order will not show up as over seven days old until the late-night batch process of the eighth. This means you will not likely find out until you come into work on the morning of the ninth, at which time it may seem to be eight days old, not seven, which some can be confusing but is easily adjusted for in the exception definition.

Similarly, if you want a batch exception to start monitoring on the third, it will do so, but the actual process will not run until late on the third and you will not see any results from it until the fourth.

View Translations in a Target Language

1. Access the Language Translation window. The default location in the RMS main menu is Control > System > Language Translation. Contact your system administrator for details on accessing this form in your own application. To access it from the ARI application, In the Main Menu click on ARI and select Translations from the Menu options. The Language Translation Window is displayed.
2. In the Action field, select View.
3. In the Language to Translate Into field, enter the ID of the language that you want to view, or click the LOV button and select a destination language.
4. In the Description Type to Translate field, enter the ID of the type of description that you want to view, or click the LOV button and select a description type.
5. To restrict the search, enter the range of elements that you want to view in the From and To fields. These entries are in the primary language. These fields are optional.

6. Click the Search button. The elements for the selected description type are displayed in the left field of the table in their primary language.
7. Translations for the elements appear in the right field of the table.
8. Click OK to exit.

7

ARI Use: View Alerts and Events

View Alerts

The Alert Viewer interface alerts you to new events assigned to you. It is your first look at events to which you are assigned individually or as part of a group. From the Alert Viewer, you link to more details about the event and actions you can perform to resolve the event, as displayed in the Event Viewer.

Access Alerts

The Alert Viewer is accessed either through the RMS toolbar or main menu. If there are events assigned to you, the button for accessing the Alert Viewer is displayed as Action button. Otherwise, the button is displayed as No Action. The menu choice is Action>ARI>Alert Viewer.

Filter Events

Events displayed in the Alert Viewer are selected using a filter. Filters consist of an event type, state, and other criteria, such as date and priority of the event.

Using the Alert Viewer, you can:

- Retrieve events by applying default or user-defined filters to event data.
- Add and maintain user-defined filters.

For example, you can filter the view of events to see events from a certain date, of a certain state, or a combination of filtering criteria.

Filter Event Data in the Alert Viewer

Using filters in the Alert Viewer, you can retrieve event data according to a number of criteria, such as event type, state, date, and priority. Filters restrict the event data displayed in the Alert Viewer.

Filter Types

There are two types of filters available in the Alert Viewer

- Default filters. These are filters that will exist for all users, such as New Alerts, All Alerts, and Deferred Alerts.
- User-created filters. In addition to the default filters, you can define, save and reuse filters for your own use.

Available Criteria for Filter Creation

When setting up a user-defined filter, you can filter by the following criteria:

- Event
- State
- Event Creation Date
- Priority
- Whether the alerts are deferred or not deferred

Displaying Owned and Supervised Events

If you are supervisor of other Active Retail Intelligence users, you can view events that you supervise in addition to the events routed to you by using the radio buttons at the top of the Alert Viewer.

- Owned events: Individual or group based assignments.
- Supervised events: Displays events that you supervise.

The default setting for the event type list is Owned Events.

The Alert Viewer Dialog [aviewer]

This screen displays a list of all the new events for you, sorted by event type. A priority indicator, to the left of the event types' names, shows the highest priority currently active of the given event type. To the right of the event types' names is a count of the total number of events of that type that are currently active.

User Filters

The upper left part of the screen displays a list of filters available to apply to the list of event types. This list includes both user-defined and system-provided filters. Selecting a filter from this list applies the filter to the event type list on the right side of the screen.

The buttons below the User Filters list are used to create, modify, and delete user-defined filters. The ✚ button adds a new filter. The current display of the Alert Viewer is replaced by the Alert Viewer Filter Definition Dialog. On this dialog, you can modify the filter name and criteria. The △ button edits a selected filter. The ✖ button deletes a selected filter from the list.

Event Type List

The list of events assigned to you is displayed on the upper right part of the screen. The information displayed about each event includes:

- The priority of the event
- The name of the event
- The event state
- The event count, or number of instances of the event

There is a user-level option that controls the re-query settings for this list. It can be accessed through the Re-query on Entry option on the Options menu. If the option is selected, the list will be refreshed whenever the user returns to the Alert Viewer. Otherwise, it is only refreshed when the filter or display settings are changed. This setting is useful if you leave the Alert Viewer open in the background and want to see your current list of alerts whenever you return the form.

Owned Events/Supervised Events

These radio buttons toggle the display of event types between events that you own (Owned Events) and events that you supervise (Supervised Events), as determined by your user and group definitions configured in Active Retail Intelligence. The default setting for the event type list is Owned Events.

Filtering and Sorting Buttons

To temporarily filter the event type list to a particular event or state, use the filter fields and button above the event type list. The button clears any filtering.

To sort the events types by priority, event type name, state, or count, click the column headings.

Defer Alert Button

Marks the alert of the selected event type and state to be deferred. You have the option to defer an alert. That is, you can view all New Alerts and decide to perform actions on them immediately, or defer an alert or alerts and resolve them later. If you choose to defer alerts, you can still view those deferred alerts by using the Deferred Alerts filter.

Defer All Button

Marks all alerts in the event type list to be deferred. To view these deferred alerts, choose the Deferred Alerts filter.

Defer Individual Event Button

Marks the event currently selected in the Event Summary block at the bottom of the screen to be deferred. The other events under the same alert will not be deferred. To view the deferred event, choose the Deferred Alerts filter.

Details Button

Launches the Event Viewer, which displays the current events of the selected event. If multiple versions of an event are active, a window asking you which version to show is displayed. Multiple versions can be active because some open events of an older version can still exist even as the newer active version is the only one being created currently. Selecting a version is necessary because the Event Viewer can only display one version at a time, and from a processing perspective, older versions of events are like a different event type.

Close

Closes the Alert Viewer.

Event Summary

The Event Summary displays summary information about the events listed in the event type list, such as the status and the values of the key parameters of the event.

Alert Viewer Filter Definition Dialog [aviewer]

When you add or edit a user-defined filter for events displayed in the Alert Viewer, this screen is displayed. On this screen, you define the criteria for the filter.

Field Definitions

Filter Name

The name of the filter. This name is displayed in the User Filters list on the Alert Viewer. This value is required.

Sort By

Identifies the field used to sort alerts that meet the filtering criteria in the Alert Viewer. You must choose a value.

Filter Criteria Area

The middle part of the screen identifies the filter criteria. This filter limits the display of alerts in the Alert Viewer. All the filter criteria fields have lists from which you can choose values.

Descriptions of the possible filtering criteria follow. The Event Name, State Name, and Priority must be valid types, states, or priorities that exist in the Active Retail Intelligence database.

Event Type

Limits the display of alerts to the specified event type.

State

Limits the display of alerts to the specified state for the event type.

Event Creation Date: After and Before

Limits the display of alerts to events that fall within the range of dates selected.

Priority

Limits the display of alerts to events of the selected priority.

Deferred

Limits the display of alerts to deferred or not deferred alerts.

Add Filter Criteria Button

Saves the filter criteria in the Active Retail Intelligence database.

Filter Summary Area

The list at the bottom of the screen displays the filtering criteria in summary form. This list shows all edits to the criteria used for this filter. You can use this list to create and reuse versions of the filter. Using the help0003.bmp button, you can also delete particular edits, or delete all edits and create an entirely new set of filtering criteria.

Delete Button

The ✖ button deletes the currently selected record in the filter criteria.

OK Button

Saves edits to the filter criteria, closes the screen, and returns you to the Alert Viewer.

Cancel Button

Cancels all edits, closes the Alert Viewer Filter Definition Dialog, and returns you to the Alert Viewer.

Event Version Selection Dialog [aviewer]

This screen is displayed when you click the Details button for an event that has event instances from multiple versions that are active simultaneously.

Multiple versions can be active because some open events of an older version can still exist even as the newer active version is the only one being created currently. Selecting a version is necessary because the Event Viewer can only display one version at a time, and from a processing perspective, older versions of events are like a different event type.

Procedures

Access the Alert Viewer

On the RMS toolbar, if you have events assigned to you, you will see the Alert button.

If no new events are assigned to you, the button is displayed as . Click this button to display the alerts.

Apply Filter to Event Data

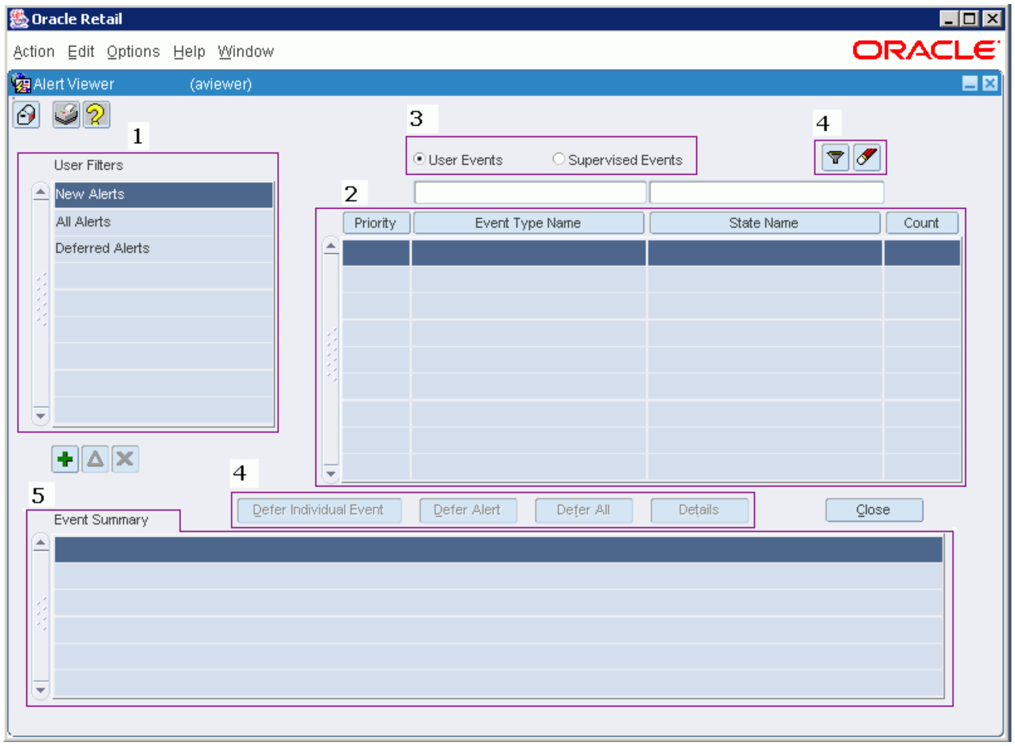
In the Alert Viewer, choose a filter from the User Filters list.

-OR-

Create a new user filter that limits the display according to the desired criteria.

Open an Event in the Event Viewer

1. In the Alert Viewer, select an event you want to view from the event type list.



Alert Viewer Screen

2. Click the **Details** button.
3. If there are multiple versions of the selected event type and state, a list of versions is displayed. Select a version and click **OK**.
4. The event is displayed in the Event Viewer.

Defer Alerts

- Defer a Single Alert Routed to You
  a. In the list of alerts on the Alert Viewer, select the alert.
  b. Click the **Defer** button. To view the deferred alert again, choose the **Deferred Alerts** filter.
- Defer All Alerts Routed to You
  a. In the Alert Viewer, select the alert.
  b. Click the **Defer All** button. To view these deferred alerts again, choose the **Deferred Alerts** filter.
- Defer Individual Event Routed to You
  a. In the Alert Viewer, select the alert for the appropriate event type and version.
  b. Select the event you want to defer from the Event Summary block
  c. Click the **Defer Individual Events** button. To view this deferred event again, choose the **Deferred Alerts** filter.

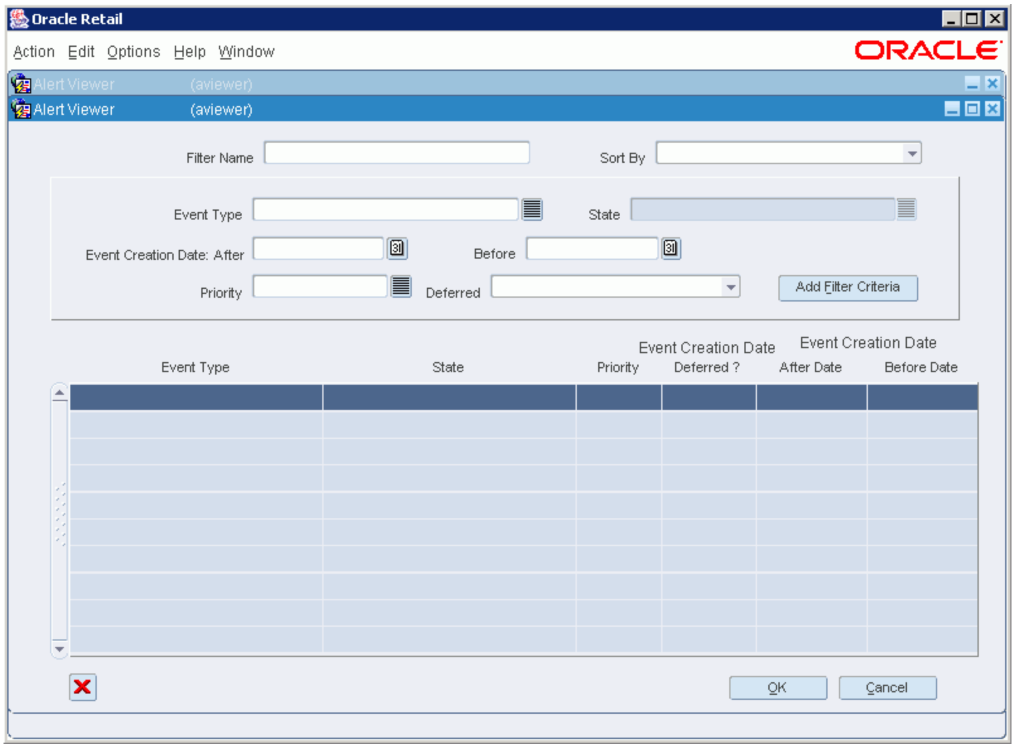
View Supervised Events

In the Alert Viewer, click the Supervised Events radio button above the event list. The display is changed to only those events that you supervise.

For more information on filtering alerts in the alert viewer, click the following link.

Create a New Filter

1. In the Alert Viewer, click the + button below the User Filters list. The Alert Viewer Filter Definition Dialog is displayed.



Alert Viewer Filter Definition Dialog

2. Specify a name for the filter in the Filter Name field, and specify how the alerts that meet the filtering criteria should be sorted in the Alert Viewer.
3. Specify the filtering criteria. Make the filter as restrictive as possible, so that the filter will display only the event data that you want.
4. Click Add Filter Criteria.
5. Click OK to close the Alert Viewer Filter Definition Dialog. The filter will be displayed in the User Filters list on the Alert Viewer.

Modify a User Defined Filter

1. In the Alert Viewer, select a filter from the User Filters list.
2. Click the △ button. The Alert Viewer Filter Definition Dialog is displayed.
3. Modify the filtering criteria as needed.
4. Click Add Filter Criteria. The edits to the filter are displayed in the list at the bottom of the screen.
5. Click OK to close the Alert Viewer Filter Definition Dialog.

Delete a User-Defined Filter

1. In the Alert Viewer, select a filter from the User Filters list.
2. Click the ✖ button.

View Events

The Event Viewer allows you to work with the specific events of a selected event type and state that were assigned to you.

Revalidate Events

Revalidating events ensures that all your events are current and should still be assigned to you. Events are reevaluated in two ways: automatically when you start the Event Viewer, and through the Revalidate Events button once the Event Viewer is open.

Automatic Event Reevaluation

When you open the Event Viewer, Active Retail Intelligence automatically attempts to reevaluate the currently selected event.

Revalidate Events Buttons

Once the Event Viewer is open, you can revalidate events by clicking the Revalidate Events buttons to do all events or only the currently selected one. A refresh button re-queries in case new events have occurred since you entered the form.

Action Dialog [arieview]

The Action Dialog is displayed when taking an action requires specifying data. For example, creating a transfer to balance inventory may show a list of parameters related to creating a transfer, such as the destination and source locations and the transfer quantity. Some of these parameters may be editable, which you can edit on this dialog. There may also be an editable comments field to attach comments to the action, such as, “I chose this action to resolve the out of stock issue because…”

Click the Do Action button from the Event Viewer screen to access the Action dialog. At the top of the Action dialog, the action you are about to perform is displayed. Directly below that, a description of the action is displayed.

In the center of the screen is a list of parameters that will be used to take the action. Required parameters appear first and are displayed with a yellow background. Optional parameters follow and are displayed with a white background. Parameters that are read-only are displayed last and have a gray background. Click the comment button to display the full text of the parameter value if it is too long to fit on the screen. Note that you cannot directly edit parameters within the comment box.

When you click inside a parameter field, the Show List button will be enabled if there is a list associated with the parameter. If this button is enabled, click it or press the F9 key to bring up the list associated with the value and you will be able to select a value.

Beneath the parameter list is the optional note to attach. If the action being taken allows comments, the note will be attached to the action as a comment. If the action being taken will route the event to another user, the note will be displayed as a user message from you.

To perform the action, click the Do Action button.

Expanded Details Dialog

The Expanded Details dialog is a mirror of the Detail section of the Event Viewer, expanded for easier viewing. To access it, click the Expand button on the Detail section.

Glossary

Action

A procedure that can be called via the event user interface to resolve the exception that triggered the event. Actions may include tasks such as approve order, create transfer, and so on. Actions are usually PL/SQL procedures, and must always be described by metadata so the Active Retail Intelligence system can use them. Some actions provide the user with a gateway into more information about the event, or a manual action. An example of such an action would be opening the purchase ordering dialog in Edit mode to examine an order in more detail prior to approving it or so that it can be approved manually.

Alert

A notification to a user that an event of a specific type (that is, an instance of a specific event definition) has occurred. This notification comes through the integrated ARI/RMS interface. Also known as event notification

Active Retail Intelligence supervisor

A type of user group where the users have supervisory privileges over other Active Retail Intelligence users. Events are assigned to groups. Some groups can be designated as supervisory. The two hierarchies overlay one another so an event may be assigned to users via either their membership in a supervisory group or their membership in a regular group. For each end user, the way this assignment occurs determines whether the user sees the event as a supervisor or a user. A user's supervision rights are determined by the group hierarchy used to assign the event, not on user A supervising user B. The advantage of this setup is that only events pertaining to issues a supervisor oversees are routed to a supervisor, not all events. Suppose you do not want user A to see all of user B's events, only those that pertain to issues user A oversees. User B has another boss, user C, who supervises other work user B does. The group hierarchy and event routing setup in Active Retail Intelligence makes routing to both the user and the two supervisors possible.

Batch monitoring

Exception monitoring for a system that is performed on a batch or recurring basis. Contrast with real-time monitoring and trickle monitoring. Business rules Definitions about how your business operates that are configured into the Active Retail Intelligence system. Business rules circumscribe the entire exception management process from monitoring through to resolution. They define not only what constitutes exceptions, but also the workflow process that should be followed when exceptions occur to determine whether a user should be notified or an automatic action should be executed.

Cloning

Basing the definition of an exception or event on another exception or event definition.

Closure Rules

A set of predefined conditions that must occur before an event can be moved to the closed state. Normally, there will be a single closure rule set that requires that the exception no longer be true. However, a special case might be a notification-only event whose only purpose is to highlight that an exception has occurred. In this case, there may be a single rule set with closure rules, which depend on the user to close the event manually.

Event

A collection of data passed from an exception's data set, plus additional information that can be determined through related metadata. Against this combined data set, events contain rules about how to route event notifications or alerts, which users or user groups to route them to, and at what priority. Events are defined through the Event Manager.

Event Instance

Organizational objects that contain all of the rules, states, and actions required for workflow processing. An event instance consists of the following attributes and objects: A copy of the exception instance that generated the event instance. The definitions and rules from the corresponding event type;

Priority of the event.

State of the event.

User assignments for the event.

Event Reevaluation

The process of determining whether an event is still open and what its state is. This process involves evaluating a set of closure rules that define the conditions under which an event can be closed.

Event Rules

Rules sets used to assign an event instance to an event state. Event rules are an attribute of the event type.

Event State Rules

A rule set used to route and assign priority to an event when it is in a particular state.

Event Schedule

A set of date, time, and cycle information that specifies when an event should be reevaluated. Schedules are defined in the Schedule dialog, and attached to events through the Event Manager dialog.

Event Type

A series of attributes and objects that define how an event instance will be presented and processed. The event type contains the following attributes and objects:

- Parameters
- Actions
- States
- Event rules
- Closure rules
- Schedules
- Exception type

Exception

A condition in the data set of a monitored system that is of interest to the business analyst or other user. Detecting exceptions is a key focus of the Active Retail Intelligence system.

Exception Instance

When an Active Retail Intelligence monitor detects a condition defined by an exception type. The exception instance contains the actual data values monitored that satisfied the specific conditions defined by the exception type. An exception type is a data object that describes another data object, the exception instance.

Exception Management

The process of monitoring data in a system for exceptions to a set of business rules, notifying users that an exception has occurred, and facilitating the exception resolution process.

Exception Schedule

A set of date, time, and cycle information that specifies when the system is scanned for instances of the exception. If an exception is not change-driven, it must be scheduled. Schedules are defined in the Schedule dialog, and attached to events through the Exception Manager dialog.

Exception Type

A definition of a data condition in a monitored system that is of interest to the business analyst or other user.

External Monitoring

Exception monitoring on a system that is deployed on a separate database from Active Retail Intelligence.

Function

Any calculated field that can be defined using PL/SQL that, after being defined as a metadata function, is accessible to exceptions and events just as any other parameter is. The function's input parameters act like the key parameters of a realm in terms of making the function available and the function's output acts just like any other parameter selected from a table. To the end user, the difference between functions and realm key parameters is transparent. They are all just parameters attached to the event and having some value. However, to the business analyst who would like to show an average or do some other data calculation, the ability to describe functions as metadata is extremely useful in terms of presenting additional information for more sophisticated decision analysis.

Group

A set of Active Retail Intelligence users to which alerts can be routed as a group. User groups are designed to allow easy event routing. A user group may have a parameter associated with it, and the resolution of state rules may resolve to several groups.

Key Parameters

Unique identifying parameters within a realm.

Layout

Each Event Type State has an associated layout, which determines the default settings for the event header and parameter detail displays in the Event Viewer when viewing events in that state.

Lookup

An Active Retail Intelligence metadata object. Lookups create virtual tables in which a set of coordinates (inputs) defines a specific value (output). The output value is assigned a parameter type and a lookup acts like a parameter in a manner similar to a function (they are accessible through their input parameters, not through tables' primary keys). Lookups prove particularly useful in their ability to define the routing of events to specific roles, depending upon the values of the event's parameters. As an oversimplified example, one lookup might be Store Manager, which is set up as a role ID parameter type (for its output) that accepts store number as its input. If such a lookup were attached as a parameter to an event, a rule could be built to route to the store manager when a specific store-related event occurs, without having to specify a specific rule for each store.

Metadata

Data about data. In Active Retail Intelligence, metadata is a non-technical translation or description of the data sources available to the Active Retail Intelligence system, such as the data dictionary and other computer language objects, that has been abstracted to a level above the actual data specifications. The purpose of the abstraction is to allow the business analyst to deal with the data in terms of its functionality rather than its structure. Metadata is an insulating layer that gives business analysts direct access to the system's internal structures without having to fully understand the programming languages or other technical aspects of the underlying systems. Metadata is the key to shifting the focus away from programming and onto rule building. Active Retail Intelligence metadata includes realms, parameters, parameter types, and lookups.

Metadata Administration

The process of defining and editing metadata such as realms, realm groups, parameter types, parameters, lookups, functions, and actions. This definition and editing is done through Active Retail Intelligence’s Metadata Maintenance dialog.

On-Demand

An option for reevaluating event types, where the event isn't reevaluated except when a user accesses it through the Event Viewer.

On-Schedule

An option for reevaluating event types, where the event periodically reevaluates its rule set to move towards resolution, according to the date, time, and cycle details specified in an event schedule.

On-Signal

An option for reevaluating event types or validating exception types, where the event reevaluation or exception validation is controlled by an external process sending a signal to Active Retail Intelligence to initiate the process. The signal can be as simple as a simple text string such as "Batch processing completed" that is sent from the external process to Active Retail Intelligence. Active Retail Intelligence has a programming object called the schedule application programming interface (API) that sends this string into the scheduler process to trigger the schedule.

Parameters

Data identifiers such as a supplier number or an order number. A parameter is an identifier, rather than the actual data value, such AH23D or 12345. For example, a parameter is a column in the RMS. An example of a specific data set is the set of parameters order number, order status, and total cost, where the parameters are related by the fact that they reference values from the same set of data defining a single order. A related data set is defined in terms of parameters, not in terms of specific values.

Parameter Types

Classifications that group parameters by the kind of data they represent. These classifications recognize that a supplier number is always a supplier number, whether it is on the supplier table, the SKU supplier table, or the order header table in some vendor performance report. Parameter types enable the linking of realms through key parameters.

Periodic Monitoring

Monitoring a system for exceptions by periodically scanning the system's temporary tables and processing the data in the tables, row by row. Contrast with real-time monitoring.

Realm

The actual data source that contains the parameter. A realm is frequently the name of an Oracle table or linked table.

Realm Type

Defines the basic attributes of a realm. Realm types include Oracle tables, local or remote views of data, and functions.

Real-Time Monitoring

Exception monitoring for a system that occurs during real-time system processing. Real-time monitoring is only available when a business rule that drives an exception management system identifies an exception that is driven by a change in the state of data.

Routing

Sending events to specific individuals for processing. You can establish routing rules that specify how events are routed to users and user groups.

Routing Group

A user group that is linked with event conditions, so that when the event conditions are met, the routing process will use that group (or groups) to build a list of users eligible to receive the event. This routing group is the intersection of all users who are members (either directly or through nested group membership) of all groups associated with the event condition. From this routing group, a user or set of users is selected for the event assignment, according to the routing rules associated with the event (for example, randomly, workload, or all users).

Rule

In Active Retail Intelligence, in general, a rule is a statement about your business practices to which the data monitored by Active Retail Intelligence is subject. Business rules govern the entire exception management process. Rules trigger the exception and guide the process of building, evaluating, resolving, and closing the event. Events defined in Active Retail Intelligence have a more specific set of rules that govern the states of the event and the rules for closing the event. See event rules, state rules, and closure rules. Whenever rules are used (closure rules, event state rules and state attribute rules) the rules are evaluated sequentially starting with number 1. For a rule to be true, all of its conditions must be true. The first rule to be found true is used, which makes rule order an important factor. If all rules are false, then rule 0 (the default) is used.

Schedule

A specification of time, date, and cycle information that can be attached to exception and event types. For exception types, a schedule can be attached to an exception type to govern when batch scanning of the exception type occurs. For event types, a schedule can be attached to an event type that defines how frequently events of that type are reevaluated. Schedules are defined in the Schedule dialog, and attached to exceptions and events through the Schedules screen of the Exception Manager and Event Manager dialog.

Scheduler

An Active Retail Intelligence component that manages the execution of the batch scans and the frequency at which events are automatically reevaluated. The scheduler is responsible for monitoring the schedule objects and re-evaluating events as called for by the schedule. It is only responsible for those events whose schedule object contains an on schedule component.

Schedule-Driven

Event reevaluation or exception validation that is performed on a periodic basis, as set in the event or exception definition. Contrast with signal-driven.

Signal-Driven

Event reevaluation or exception validation that is controlled by an external process sending a signal to Active Retail Intelligence. The signal can be as simple as a text field, such as "Nightly batch run completed," that passed to the scheduler component in Active Retail Intelligence. Contrast with schedule-driven.

State

An object that provides organization to the workflow associated with an event type. When Active Retail Intelligence launches an event, event rules are used to assign the event to a state. An event resides in only one state at any given time in the life of the event. The state represents a location in the exception resolution process, and also defines the actions available for resolving the exception or transitioning to the next state. The definition of a state includes:

A set of state rules that determine the routing of an event and its priority. Each state rule contains an optional action that is executed if the rule evaluates to TRUE. Default values for the ownership group, priority and optional action for events in this state. These values are used whenever the assignment rules fail to find a suitable owner or explicitly set a priority for the event.

A list of the user actions available in that state. For example, when an order-tracking event is in the awaiting approval state, the option to not approve the order should not be available. Likewise, if the order has been created and is now in the approved state, the option to not approve the order should be visible until the first receipt is made against the order. After the first receipt, the event enters yet another state, partially received, that allows cancellation of the outstanding quantities on order but not non-approval.

State Rules

A set of rules that govern the transition of an event type from one state to another. These rules are expressed as a set of conditions that must be met for the state transition to occur.

User

In Active Retail Intelligence, a user is an object representing the individual users working in the system. An Active Retail Intelligence user definition consists of such things as data relating to the individual, preferences for working in the system, information on advanced options for notification, status, and any specific rules for routing events to that user.

User Group

A set of Active Retail Intelligence users to which alerts can be routed as a group. User groups are designed to allow easy event routing. A user group may have a parameter associated with it, and the resolution of state rules may resolve to several groups.

Versioning

The process of attaching a version number to each edit of an exception or event definition. Versioning allows users to know exactly when the old and newly modified versions of the exception or event are active.