AppsAudit

Training Guide

Software Version 6.5.5

LogicalApps®

Restricted Rights Legend

Logical Apps provides on-site support as well as remote phone and web support to ensure quick and effective product implementation. To request support, to suggest product enhancements, or to comment on Logical Apps software or documentation, send email to support@logicalapps.com, or contact us at the address or phone number given below.

Document AR002-655A

11/2/05

Logical Apps
15420 Laguna Canyon, Suite 150
Irvine, CA 92618
949.453.9101

Contents

Chapter 1

About AppsAudit

AppsAudit tracks changes to the values of fields in database tables, and displays reports that present information about changes to each field. For each database row in which field values have changed, the report includes the old and new values for each changed field, the ID of the user who made each change, the time at which the change was made, and the type of change.

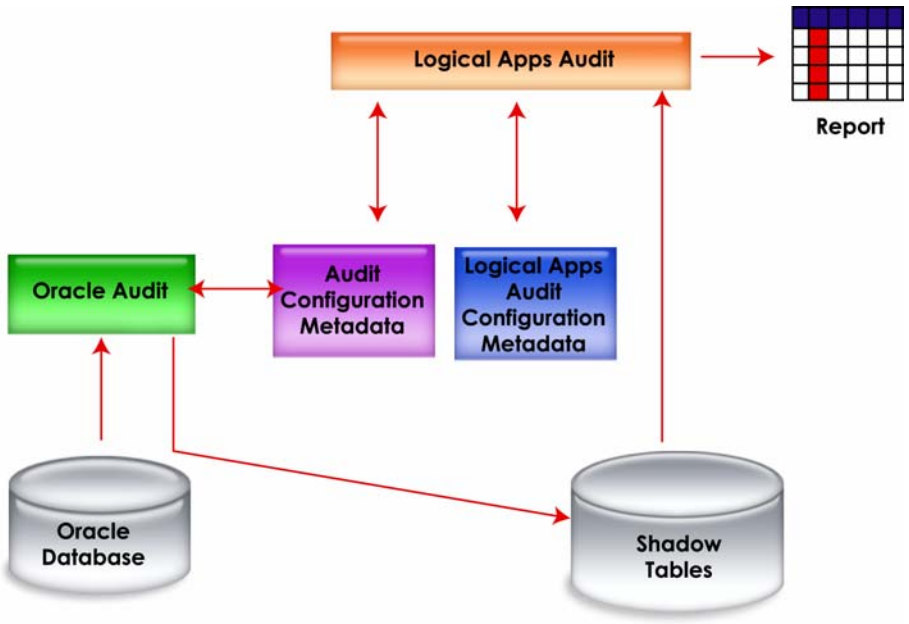
You select the tables you want to include in an audit by assigning them to a group. You refine the audit further by selecting columns from the tables that belong to the group. You can also link audited columns to translations — meaningful values that correspond to the values held in audited tables. For example, a person's actual name might be the translation value when an audited table column holds a numeric ID for the person.

AppsAudit is completely integrated with Oracle Applications. It not only uses Oracle application and database features, but also provides audit and reporting capabilities that Oracle Audit does not offer. The following table lists major differences between AppsAudit and Oracle Audit:

AppsAudit	Oracle Audit
Displays audit results in an AppsAudit Report.	Has no available reports.
Displays translation values for audited fields in reports.	Has no translation capability.
Permits the creation of customized reports based on translated, audited data. AppsAudit includes audited data and current data.	Translated data does not exist to be reported on. Only audited data is stored; therefore the current values would need to be derived.

AppsAudit	Oracle Audit
Creates triggers with When clauses on audited fields for optimal performance.	Builds triggers with no conditions, causing excessive overhead during auditing.
Permits users to view audited data through AppsAudit Online Form.	Provides no way to view audited data online.
Permits the migration of audit groups from one Oracle instance to another.	Has no migration capability.

The following figure depicts AppsAudit integration with Oracle Applications:



Opening AppsAudit

To open AppsAudit:

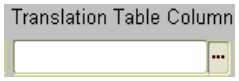
1. Select the Logical Apps AppsRules responsibility in the Oracle Applications list. (Ensure first that the AppsRules responsibility is available to you.)
2. A Logical Apps — AppsRules form appears. In it, click on the AppsAudit tab.

If you close the AppsRules form, you can reopen AppsAudit:

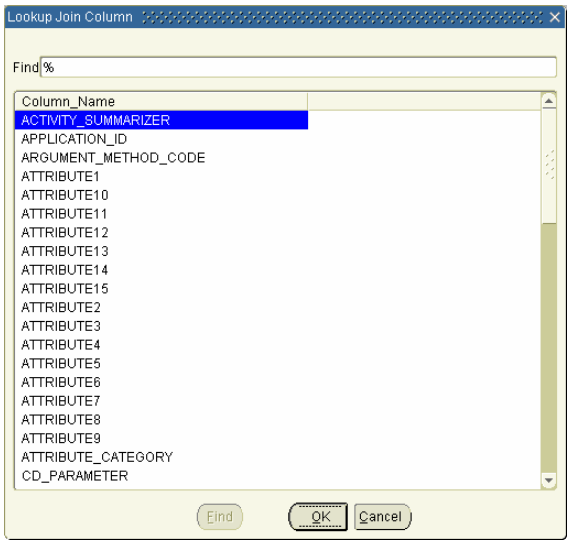
1. In the Logical Apps Navigator, click on the AppsRules option and then on the Open button. (Or, double-click on the AppsRules option.)
2. Once, again, click on the AppsAudit tab in the AppsRules form.

Making Selections in AppsAudit

When you click in a list of values box to enter data, AppsAudit presents a list icon to the right of the box. It looks like an ellipsis:



When you click on this icon, AppsAudit opens a window in which you can produce a filterable list of values that may be entered in the text box.



To use this window:

1. Click in the Find box and type a string of characters for which AppsAudit should search. You can use the percent sign (%) to stand for any string of characters. For example, the string *AKD* would return all entries that begin with those letters, while the string *%AKD* would return all entries in which those letters appear in any position. The percent sign alone would return all possible entries.
2. Click on the Find button. AppsAudit returns a list of items that match your search criteria.
3. Scroll through the list and click on the item you want.
4. Click on the OK button.

Chapter 2

Defining Audit Groups

To be audited, a table must belong to an audit group, and so an essential step in the auditing process is to create (or modify) a group. When you start AppsAudit, it opens by default to the form in which you complete this step.

Creating an audit group involves the following:

- Giving the group a name.
- Assigning the group a "state," which designates whether auditing is requested, enabled, or disabled.
- Selecting the database tables that are to belong to the group. Once you add a table to a group (and save the addition), it cannot be removed.
- Engaging security by determining the Oracle responsibilities that are allowed to run audit reports for the group.
- Saving the group.

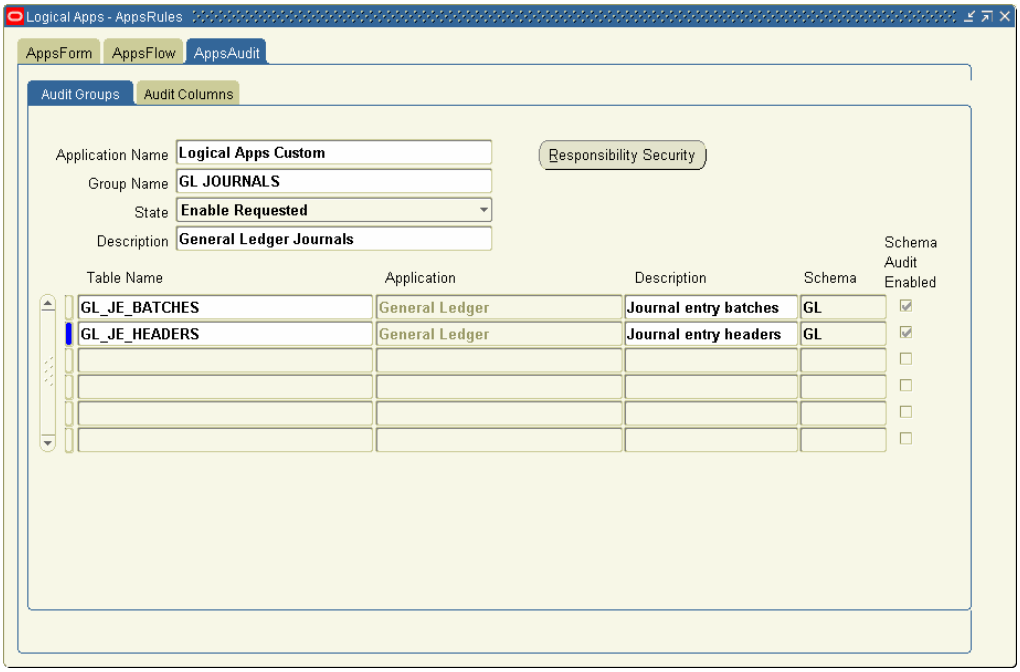
For a table to belong to a group, its entire schema must be defined as audit-enabled. Unlike Oracle Audit, AppsAudit performs this setup step automatically as you add tables to a group.

Note
Be sure the AuditTrail:Activate profile option has been set to Yes.

After a group is defined and saved, you need to select columns for auditing from each of the group's tables. This process is competed in a distinct form (and is discussed in Chapter 3).

Assigning a Group Name and State

To set up an audit group, ensure that the Audit Groups tab is selected. This is the default when you start AppsAudit:



An Application Name field includes the entry *Logical Apps Custom*. This is the name of a "container" for the audit groups you create. You cannot change this value. (Other containers exist for "seeded" audit groups, and you can use Application Name values as parameters to search for those groups. See "Querying an Audit Group" on page 8.)

Enter the following values in the Audit Groups form:

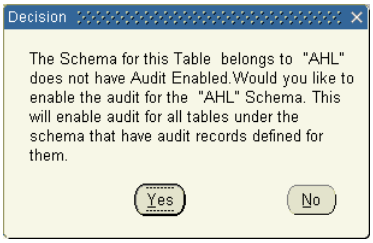
1. In the Group Name field, type the name you want to assign to the group. This field accepts uppercase letters and numerals; it does not accept lowercase letters.
2. In the State list box, select one of the following entries:
   - Enable Requested: Auditing is requested but not yet enabled. This is the default value, and is the appropriate entry for a new audit group.
   - Enabled: The system can conduct audits on the group. If the State box is set to Enable Requested, AppsAudit automatically changes the value to Enabled when you select a Create Audit Objects option from the LogicalApps Utilities menu (see page 14).
   - Disable—Interrupt Audit: Auditing for the group is paused, and audit objects, including audit data, are preserved in shadow tables.
   - Disable—Prepare for Archive: Auditing for the group is stopped, and current data for each row is stored in shadow tables, which can be archived.
   - Disable—Purge Table: Auditing for the group is stopped, and audit objects, including audit data, are dropped.
3. In the Description text box, type a description of the audit group.

Adding Tables to the Audit Group

In each row of the grid in the lower half of the Audit Groups form, use the Table Name list of values to select a database table you want to include in the audit group.

When you select a table, AppsAudit automatically supplies appropriate values for the remaining fields and check box in the row:

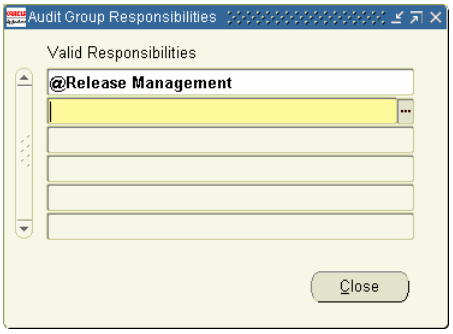
- Application: AppsAudit provides the name of the Oracle Application that stores data in the selected table. You cannot change this value.
- Description: AppsAudit displays a description of the database table. You can modify the description if you wish.
- Schema: AppsAudit supplies the name of the schema (database organizational structure) of which this table forms a part. You cannot change this value.
- Schema Audit Enabled: AppsAudit checks this box if the schema is audit-enabled. If you select a table whose schema is not audit-enabled, AppsAudit displays a Decision dialog in which you can select Yes to enable the schema for auditing or No to leave the schema as it is. If you select No, however, AppsAudit does not accept the table into the audit group. As a result, the Schema Audit Enabled box is checked for every table you add to the group.



Setting Security

To specify users who are permitted to run audit reports for the tables in this group:

1. Click on the Responsibility Security button. An Audit Group Responsibilities form appears:



2. Select any number of responsibilities (one in each field). To run an audit report, a user must be assigned one of the responsibilities you've selected.
3. Click on the Close button.

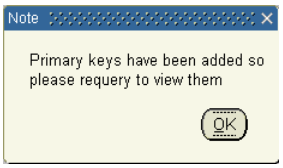
If you choose not to select responsibilities in the Audit Group Responsibilities form, any user can run audit reports on the tables in this group.

Note
A table can belong to more than one group. If you limit the responsibilities that have access to a group, and one of its tables exists in other groups, then the same limits apply to the other groups as well.

Saving the Group

When you have finished setting up the group, save it:

1. Click on File in the Oracle Applications menu bar, then on Save in the File menu.
2. As you save the group, a message informs you that primary keys have been added, but that you need to requery the group to view the primary keys. Click on the OK button to clear the message.

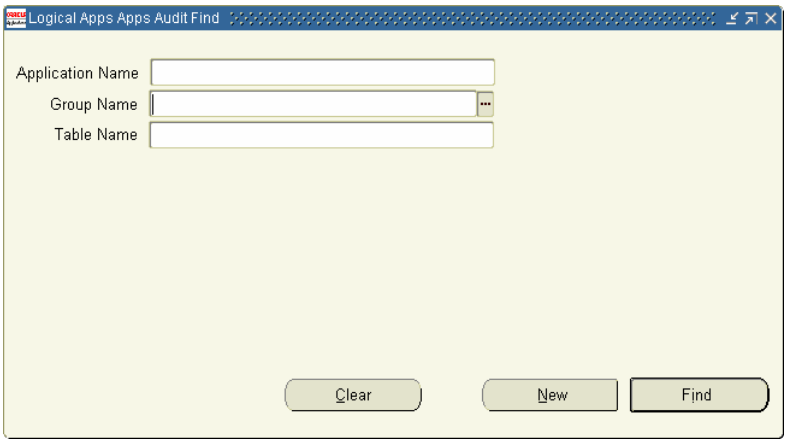


To requery the group, first clear it from the Audit Groups form — position the cursor in the Group Name field, and then click on Edit in the menu bar, Clear in the Edit menu, and Record in the submenu that appears. Then use the Find feature (see the next section) to reload the group you are creating. (AppsAudit displays the primary keys in the form that's accessible from the Audit Columns tab; see Chapter 3).

Querying an Audit Group

To find an audit group, use the AppsAudit Find window:

1. Click on View in the menu bar, then on Find in the View menu. Or click on the Find button, which is located second from the left in the tool bar and looks like a flashlight. The following form appears:



2. From the lists of values, select any combination of the following values:
   - Application Name: The name of the application that contains a group you want to locate. (Remember that in this context, *application* means a virtual "container" that holds a selection of audit groups.) All of the groups you create belong to a Logical Apps Custom application, but you may have access to others that contain "seeded" audit groups, which would be configured by Logical Apps and provided with AppsAudit.
   - Group Name: The name of the group you want to locate.
   - Table Name: The name of a table within a group you want to locate.
3. Click on the Find button.

Depending on the parameters you specify, the search may uncover more than one group. For example, if you provide only a table name and the table belongs to more than one group, the search returns all the groups to which the table belongs. In such a case, ensure that the cursor is positioned in the Group Name field of the Audit Groups form (this is the default after you use the Find tool) and press the up- or down-arrow button to move from one group to another.

The Find form also contains a New button. If you click on it rather than the Find button, any values entered in the Audit Groups form are cleared and you can create a new group.

Chapter 3

Defining Audit Columns and Translation Data

After you specify the tables that belong to a group, you need to select columns from each table. For a given column in an audit table, you can also specify a translation value — a corresponding column in a lookup table. This lookup column would contain meaningful values that match up with values in the audited tables, such as a person's actual name in place of a numeric identifier.

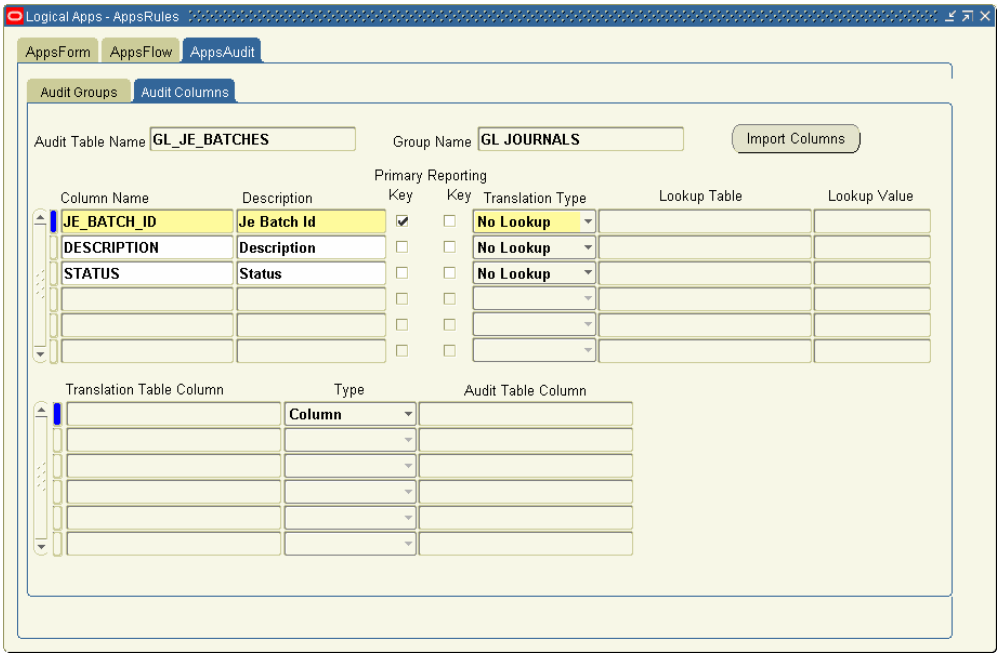
A table can belong to more than one audit group, and the selection of columns for a table is identical in all the groups to which it belongs. If you add columns for a table in one group, the same columns are added for that table in its other groups. If you add a table to a group, any column selections made earlier in other groups apply in the new group by default. Translation-value settings configured for a table in one group apply in all the groups to which the table belongs. Once you add columns for a table (and save the addition), the columns cannot be deleted.

You select columns and specify translation values in the form available when you click the Audit Columns tab in AppsAudit. First, though, click on the Audit Groups tab and ensure the following:

- You have saved the audit group with which you want to work, and then reloaded it. (See "Querying an Audit Group" on page 8.)
- You have selected the table whose columns you want to prepare for auditing. (In the grid that lists tables, click on the small rectangle to the extreme left of a table row. Doing so turns the rectangle blue, an indicator that the table is selected.)

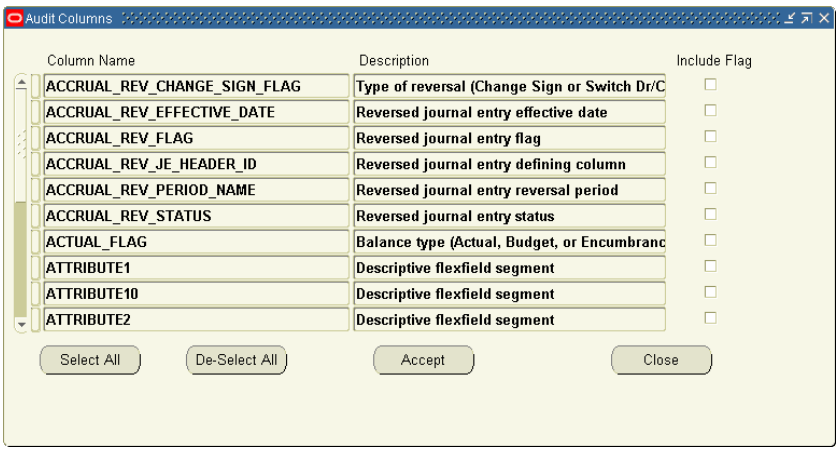
Selecting Columns

Click on the Audit Columns tab. The following form appears, displaying the names of the audit group and audit table that you selected in the Audit Groups tab. If the table has never before been included in a group, the form also shows values for the table's primary-key columns. If the table belongs to other groups, the form shows values for the columns selected in those groups.



You can select additional columns for auditing, one by one, by selecting each in the next available list-of-values box under Column Name. Alternatively, you can complete these steps to select audit columns:

1. Click on the Import Columns button. The following Audit Columns form appears, listing all of the columns in the table you've selected.



2. Click the Include Flag checkbox for each of the columns you want to audit. Or, click on the Select All button to include all columns. (You can also click on a selected check box to clear it, or the De-Select All button to clear all check boxes.) Owing to Oracle Audit limits, you can select a maximum of 248 columns.

3 Click on the Accept button.

Notes

Once columns have been selected and the group has been saved, the columns cannot be deleted from the group definition.

For key information to be used, tables must be registered to the Applications. Oracle seeds this information for its e-business suite tables. Mappings that use custom database tables should be registered, through the use of Oracle APIs.

Adding to Column Definitions

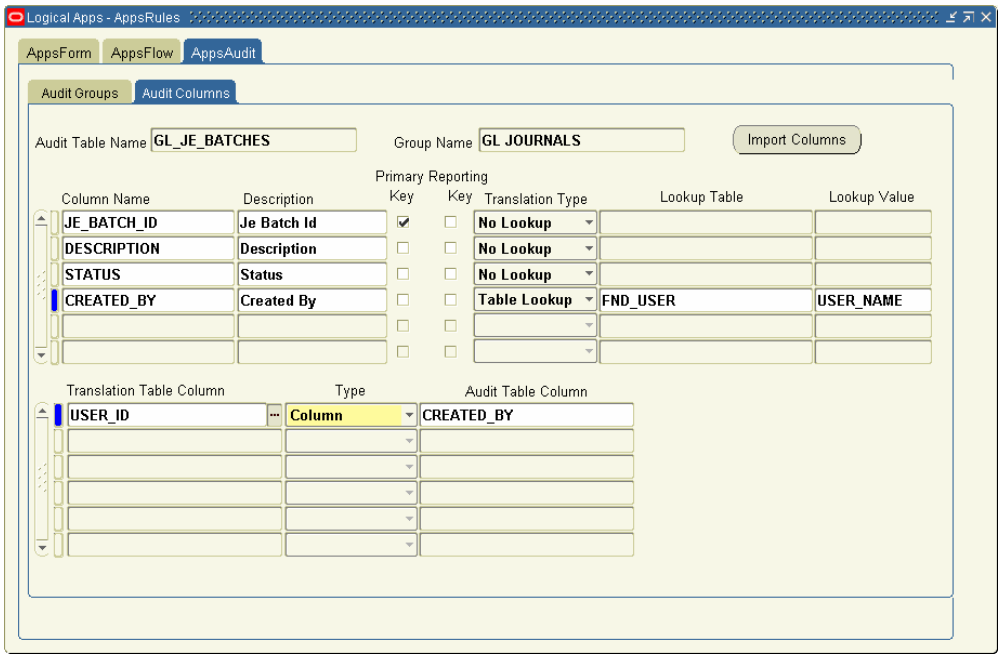
Regardless of the method you use to select columns, AppsAudit automatically fills in a description for each column and selects a No Lookup translation type by default. (See "Setting Up Translations," below, for more on translation type selections.)

Moreover, if a column is a primary key, AppsAudit has already selected the Primary Key check box; you need not alter the default selections for this check box.

For each column, select the Reporting Key check box if you want information about fields in the column to appear in reports even if field values have not changed. Clear the check box if you want information about column fields to appear in reports only for field values that have changed.

Setting Up Translations

If you want audit reports to display actual values from an audited column, select No Lookup in its Translation Type list of values.



If, however, you want audit reports to display a translation value for an audited column, join it to a corresponding column in a lookup table. Typically, you would specify a linkage among three columns:

- The first is the column that contains an audited value. In the example illustrated above, this is CREATED_BY in the GL_JE_BATCHES table.
- The second is a lookup-table column that contains an identifying value — the same value as in the audited table. In the example illustrated above, this is USER_ID in the FND_USR table.
- The last is a column in the lookup table that contains the translation value. In the example illustrated above, this is USER_NAME in the FND_USR table.

To create this linkage:

1. In the Translation Type list of values, select Table Lookup.
2. In the Lookup Table list of values, select the name of the lookup table you want.
3. In the Lookup Value list of values, select the name of the lookup-table column that contains translation values for the audited column.
4. Move to the lower grid and, in the Translation Table Column list of values, select the lookup-table column that contains the identifying value.
5. In the Type list of values, select the value *Column*.
6. In the Audit Table Column list of values, select once again the column from the audited table that contains the audited value.

In the lower grid, you can complete as many rows as you like to create a translation value as complex as you like. The rows have an AND relationship — all must be true for a value to be returned.

To change translation values (once they've been saved), you must delete the existing values and then save the audit group before selecting new values. If you attempt to replace old translation values directly with new values, AppsAudit presents a message stating that you must first delete the old values and save the deletion.

Saving Your Work

Once you've finished selecting columns and defining translation values, save the new configuration: click on File in the menu bar, then on Save in the File menu. Or, click on the Save icon, located first on the left in the toolbar.

Activating the Audit

To enable AppsAudit to begin auditing the group you have defined, click on LogicalApps Utilities in the menu bar, then on AppsAudit Create Audit Objects in the LogicalApps Utilities menu.

At this point, two concurrent programs run:

- An AuditTrail Update Tables program updates any changes made to existing audit groups and creates new audit objects required by new audit groups.
- An LA AppsAudit Update Audit Objects program optimizes the triggers to include When clauses for the selected columns.

You must select the AppsAudit Create Audit Objects menu option in the following circumstances:

- You have created a new group.
- You have added a table to an existing group.
- You have added a column in an existing group.

You need not select the AppsAudit Create Audit Objects menu option in the following circumstances:

- You have added or changed a lookup table or a lookup value.
- You have added or changed lookup join information.
- You have changed a description.

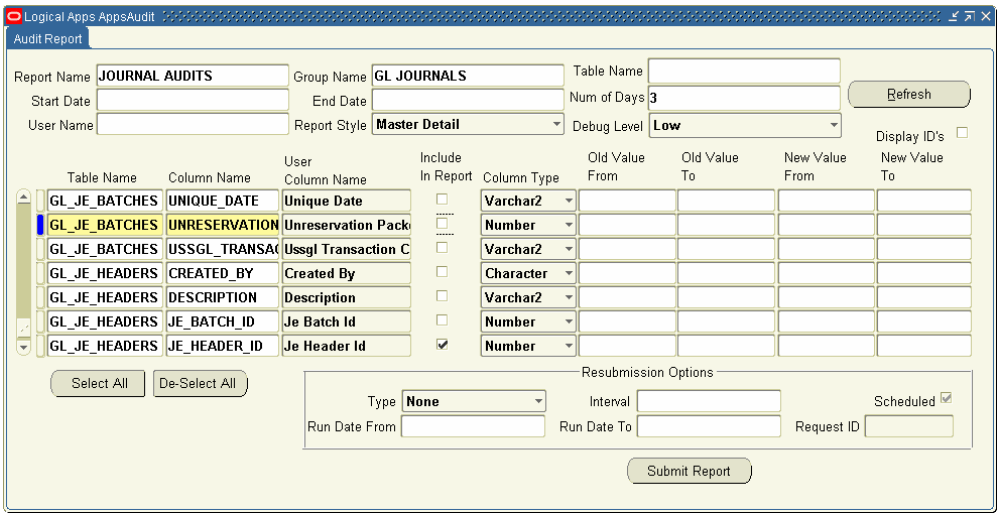
Chapter 4

Reporting

AppsAudit lets you save report definitions and run reports based on translated audit data. You may create a new report definition or query a saved report definition. You can also select filter criteria for the report based on ranges of old and new values. Reports are executed via the concurrent request queue.

Defining Reports

To run reports, click on LogicalApps Utilities in the menu bar, and then on AppsAudit Report in the Utilities menu. The following Audit Report form appears:



Audit Report Header

To create a new report definition, begin by completing entries in the header area of the Audit Report form:

1. In the Report Name box, type a name for the report.
2. Make selections in either or both of the Group Name and Table Name lists of values. (These fields provide access to groups, and tables within groups, that are either associated with responsibilities you have been assigned, or not associated with responsibilities at all.)
   - If you make a selection in the Group Name field, the Table Name field displays only tables belonging to the group you've chosen. You can select one of those tables to report only on auditing for it, or you can leave the Table Name field blank to report on auditing for all the tables in the group.
   - If you leave the Group Name field blank, the Table Name field displays tables belonging to all audit groups to which you have access. Select one of the tables to report on auditing for it.
3. Specify a span of time the report should cover. Do one of the following:
   - Type or select starting and ending dates in the Start Date and End Date boxes. Use the format DD-MMM-YY HH:MM:SS — for example, 1:30 P.M. on April 28, 2005 would be rendered as 28-Apr-05 13:30:00. (Note that the Start Date and End Date boxes default to the current date.)
   - Type a value in the Num of Days box. AppsAudit subtracts this value from the date the report is run and uses the resulting date as the start of a date range. The end date is the date the report is run.
4. In the User Name list of values, select a user to report only on data changes made by that user. Or leave the box blank to report on data changes by all users.
5. In the Report Style list box, determine the format of the report output by selecting either of the following:
   - Master Detail presents report information as blocks of data, each of which lists a table, information about its primary keys, and then a row of data about each of the audited fields that has changed.
   - Delimited File presents report information as records of data changes, each using a tilde to separate the individual values that make up a record.
6. In the Debug Level list box, select High or Low. AppsAudit maintains data about both report generation and the audit "package" — the core generation of audit data from which a report gathers information. It places this data in a log table, where it may be reviewed if problems need to be resolved. The High debug-level value produces more detail and retains the data in the table. The Low value produces less detail and allows log data to be purged. Typically, select Low. (This setting does not alter the information presented in an audit report.)
7. Select the Display IDs check box if you want the report to identify primary keys, or clear the check box if not.

Audit Report Detail

Once the header information is in place, click the Refresh button to select tables and columns for inclusion in the report. In response, AppsAudit fills the grid in the Audit Report form with entries for all columns selected for auditing:

- If you selected a table name as you completed the report header, AppsAudit displays all the audit columns from that table.
- If you selected a group name as you completed the report header, AppsAudit displays all the audit columns from all the tables of the group.

For each, AppsAudit supplies the Table Name, Column Name, and User Column Name, and selects an Include in Report check box. In a Column Type list box, AppsAudit also provides the data type for the column; however, if the report is to present data from a lookup table, the data type shown here reflects the lookup value. For instance, if the audited column is VENDOR_ID but a lookup table is to provide VENDOR_NAME in the report, the data type is given as *Character*, not *Number*.

You can refine this automatically generated selection of columns:

- Choose the columns you want in the report. For each column, click on the Include in Report check box to remove the check mark and therefore exclude the column from the report. Or click on the check box again to reinsert the check mark and reinclude the column in the report.

  Alternatively, click the Select All button to include all the columns or the De-Select All button to exclude all the columns. Note, however, that you cannot exclude columns defined as primary keys.

- Define ranges of values that filter report entries.

  For any column, type entries in the Old Value From and Old Value To boxes to report only data changes that begin with a value in the range you specify. For example, if a numeric field contained the value 6 and the data were changed, a record of that change would nevertheless be excluded from a report if you specify old-value-from and -to entries of 0 and 5.

  Similarly, type entries in the New Value From and New Value To boxes to report only data changes that end with a value in the range you specify.

Note
Filter values for two or more columns have an AND relationship — all must be true for the filter to return a result. For example, if you define an old-value filter for a name column that permits only the name *Smith* (if that name is placed in both the From and To boxes), and an old-value filter for a date column that permits only a single date (if that date is placed in both the From and To boxes), then the report would display audited data only for records concerning Smith on the specified date.

Resubmission Options

You can schedule a report to be run repeatedly. To do so, select values in the block of the Audit Report form labeled Resubmission Options:

- In the Type list box, select Hours, Days, or Weeks to designate the unit of time you use to define an interval at which the report is to be rerun. Or, keep the default selection, None, to prevent the report from being rerun.
- In the Interval box, type a number that expresses the interval at which the report is to be run. For example, if you type 5 here and select Hours in the Type list box, the report is resubmitted every five hours.
- In the Run Date From and Run Date To boxes, type or select dates and times that mark the beginning and end of the period in which the report should be run repeatedly. Again, use the format DD-MMM-YY HH:MM:SS — for example, 1:30 P.M. on April 28, 2005 would be rendered as 25-Apr-05 13:30:00.

AppsAudit fills in these values:

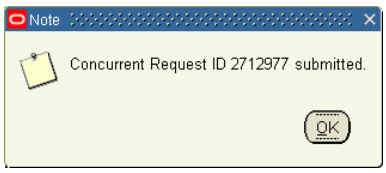
- The Scheduled check box indicates that the reporting cycle you have defined is being implemented. AppsAudit checks it when you submit the report. Once the date in the Run Date To box has passed, AppsAudit clears the Scheduled box.
- The Request ID text box displays an identifying number for the most recent submission of the report. For instance, if a report runs four times per day, this box would at the end of the day display the fourth request ID.

Saving a Report

When you are done selecting header, detail, and resubmission options, save the report definition: click on File in the menu bar and then on Save in the File menu.

Submitting and Viewing Reports

Once you have defined a report, click on the Submit Report button in the Audit Report form. When you do, a message similar to the following one indicates that the report is being generated:

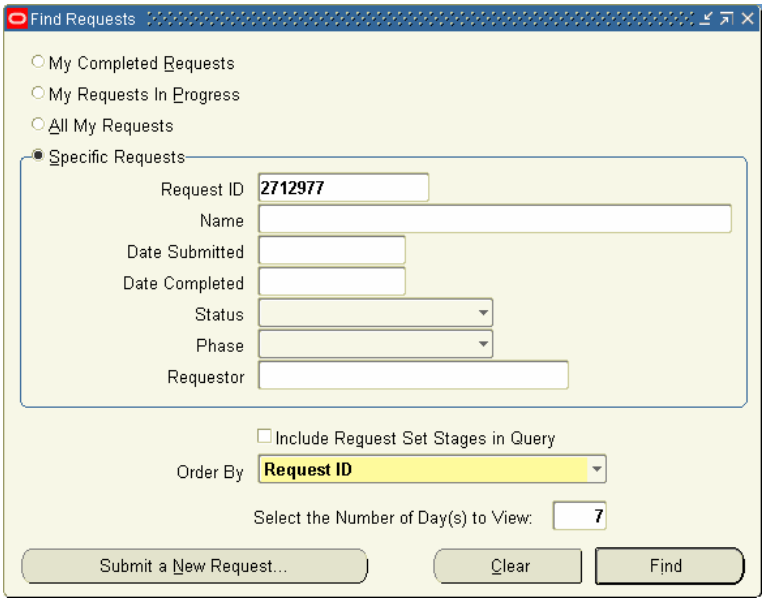


Note the request ID number, then click on the OK button to close the message.

Note
If two reports are submitted, the second does not begin until the first has finished running. This is because audit data is placed in a temporary table until it is no longer needed for generating the report, then purged; a second report must wait so that it does not overwrite temporary-table for the first.

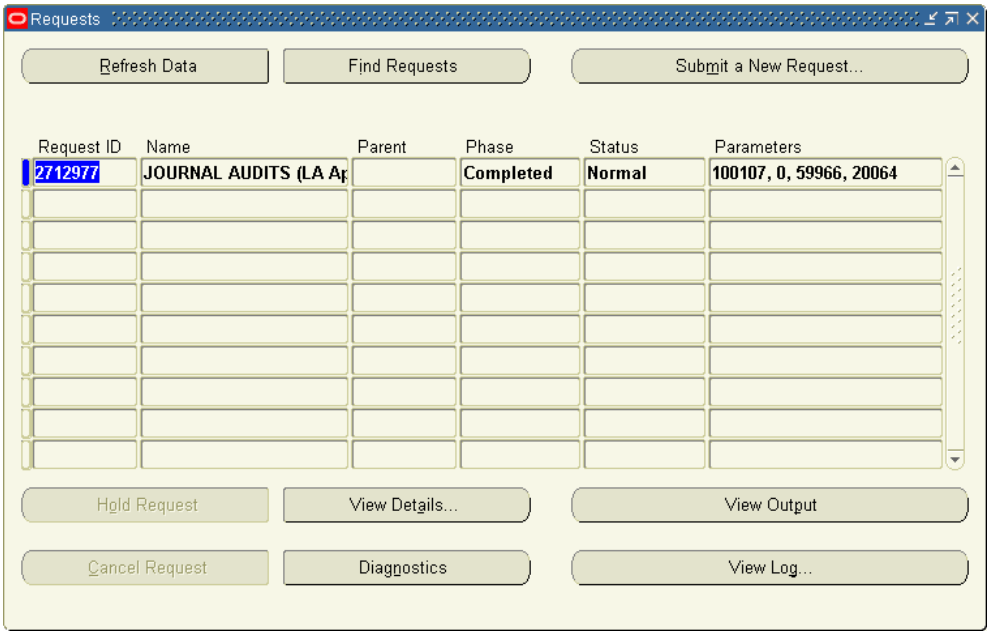
To view the report:

1. Click on View in the Oracle Applications menu bar, then on Requests on the View menu. The Find Requests form appears:



2. To see your report and its progress, click the My Requests in Progress radio button and then the Find button. Or click the Specific Reports button, type the request number for your report in the Request ID field, and click the Find button.

3. A Requests form shows the status of your request. When it informs you that the report is completed (you may need to click on the Refresh Data button), click on the View Output button. The report appears.



The form that the report takes depends on the selection you made in the Report Style list box when you entered header information in the report definition. If you chose Master Detail, the report looks like this:

```
Date : 08-OCT-04 07:21:43                          JOURNAL AUDITS                          Page : 1   Of 4


                Group Name :  GL JOURNALS

                Table Name :

                Start Date :

                  End Date :

                  Username :
                             1
           Number of  Days :
□
  Date  08-OCT-04 07:21:43                         JOURNAL AUDITS                           Page  2 Of 4

 Table Name:       GL_JE_BATCHES

 Selection Criteria :
 ------------------------------------------------------------------------------------------------------------------------
 Column Name                       Old Value From          Old Value To            New Value From          New Value To
 --------------------------------  ----------------------  ----------------------  ----------------------  ----------------
 Audit Data :
 ------------------------------------------------------------------------------------------------------------------------
 Primary Key                     Value

 ------------------------------- ------------------------------
 Je Batch Id                     31499

                Transaction
  User Name     Type        Time Stamp               Column Name              Old Value                 New Value
  ------------  ----------  -----------------------  -----------------------  ------------------------  -------------------
  SMCLAUGHLIN   Insert      07-OCT-2004 16:57:15     Name                     JAN-ADJ 07-OCT-2004       JAN-ADJ 07-OCT-2004
                                                                               16:56:43                  16:56:43
_
```

If you chose Delimited File, the report looks like this:

```
Group_Name~Table_Name~Start_Date~End_Date~Primary_Key~Value~DB_Column~DB_Value~Username~Transaction_Type~Timestamp~Colum
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Insert~13-OCT-2004 16:43:18~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Insert~13-OCT-2004 16:43:18~Name~nu.
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Insert~13-OCT-2004 16:43:18~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~13-OCT-2004 16:43:23~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~13-OCT-2004 16:43:23~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~13-OCT-2004 16:43:26~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~13-OCT-2004 16:43:26~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:06~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:06~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:10~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:10~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:13~Descripi
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-CONTROLLER~JE_HEADER_ID~33372~SMCLAUGHLIN~Update~14-OCT-2004 09:10:13~Set Of I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Insert~12-OCT-2004 09:09:42~Descriptio
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Insert~12-OCT-2004 09:09:42~Name~null~I
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Insert~12-OCT-2004 09:09:42~Set Of Bool
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:04~Descriptio
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:04~Set Of Bool
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:08~Descriptio
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:08~Set Of Bool
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:11~Descriptio
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ-GERMANY~JE_HEADER_ID~33363~SMCLAUGHLIN~Update~12-OCT-2004 09:10:11~Set Of Bool
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ~JE_HEADER_ID~33362~SM1~Update~12-OCT-2004 06:18:45~Description~ADJUSTMENTS FEI
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ~JE_HEADER_ID~33362~SM1~Update~12-OCT-2004 06:18:45~Set Of Books Id~Vision Com
GL JOURNAL ENTRIES~~~~Je Header Id~FEB-ADJ~JE_HEADER_ID~33362~SM1~Update~12-OCT-2004 06:18:58~Description~ADJUSTMENTS FEI
```

Changes to Audit Setup

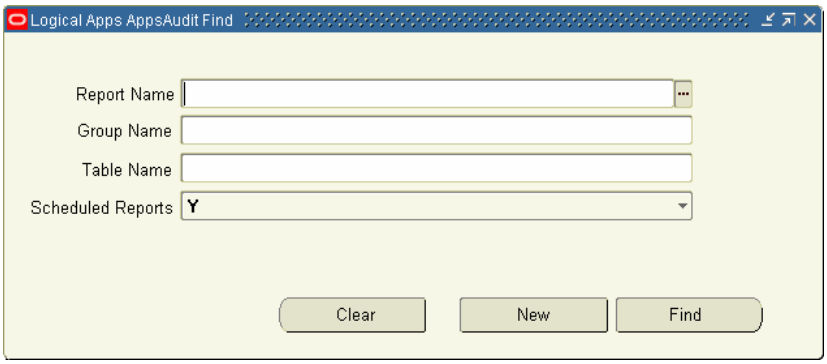
Reports based on Audit Groups that have been changed (tables have been added) need to be refreshed. Use the Refresh button on the Report Submission form to import automatically any new table columns that were not previously defined on this report.

If a report is based on an Audit Group that has been changed, and the report definition has not been refreshed, the report will fail. Simply click the Refresh button and submit the report again.

Querying a Report

To find an existing report:

1. Ensure that the Audit Report form is open (page 17).
2. Click on View in the menu bar, then on Find in the View menu. The following form appears:



3. Define search parameters:
   - Report Name — Type the name of a report or select one from list of available reports.
   - Group Name — Type the name of a group or select one from list of available groups defined on saved reports.
   - Table Name — Type the name of a table or select one from list of available tables defined on saved reports.
   - Scheduled Reports — Select *Y* for reports scheduled for regular resubmission or *N* for unscheduled reports.
4. Click on the Find button. AppsAudit populates the Audit Report from with definition data for the report you specified.

Alternatively, click on the New button to clear the Audit Report from for the creation of a new report definition. Or click on the Clear button to clear selections you've made in the Find form.

Chapter 5

AppsAudit Online Form

An AppsAudit Online form can display audit data about a record currently selected in an Oracle Applications form. To use it, complete two tasks:

- Create an AppsForm rule that establishes a navigation link from the form you are auditing to the online audit form.
- Add an AppsAudit Online function to the menu structure of the responsibility that is to have access to the online audit form.

To complete the process, you need the following information:

- The name of the form you are auditing and the application in which it runs.

  This information is available in the Form Functions form of the System Administrator responsibility: the form name you want is in the Function field (no matter what tab you've selected) and the application name is in the Application field on the Form tab. For example, if you query on the Enter Vendor user form name (in the Form field, Form tab), you obtain the values APXVDMVD in the Function field and Payables in the Application field.

- The name of the responsibility that both has access to the application and is assigned to a user who needs to view the online audit form.

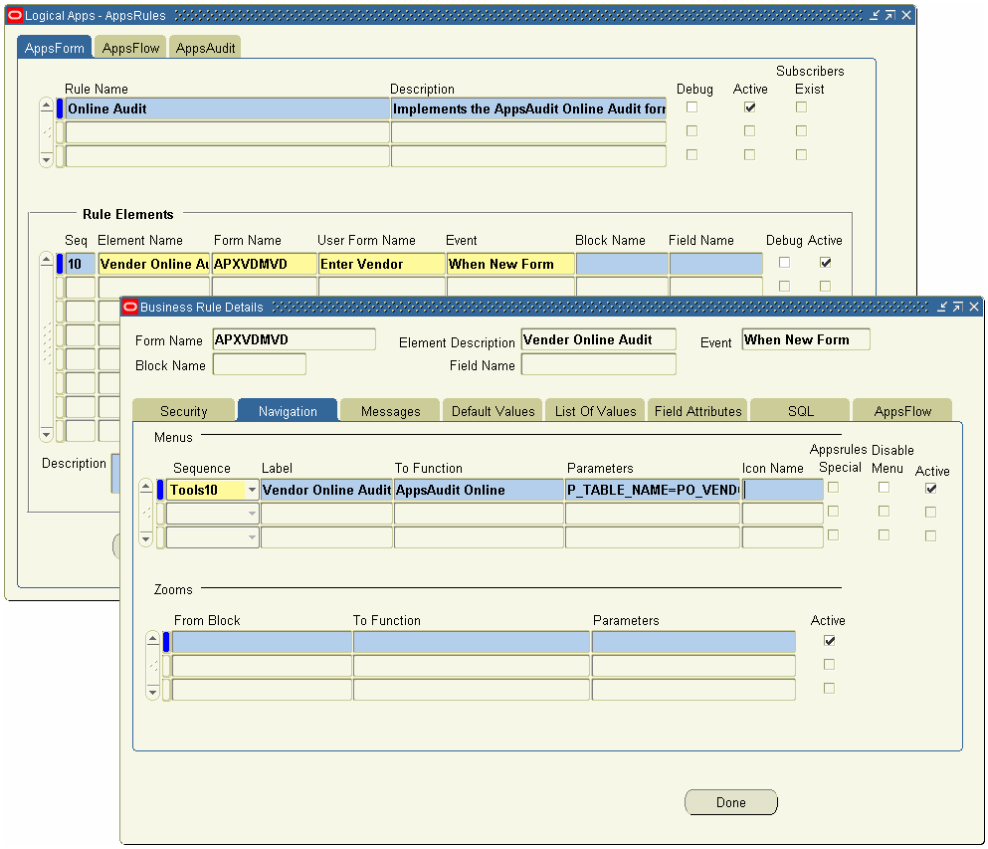
  This information is available in the Responsibilities form of the System Administrator responsibility. You can use the application value (for example, Payables) to query on the Application field and obtain all the responsibilities with access to that application. (One of those responsibilities, for instance, is Payables Manager.)

- The name of the menu associated with the responsibility. When the responsibility is loaded in the Responsibilities form, the menu name is displayed in the Menu field on that form — for example, AP_NAVIGATE_GUI12 for Payables Manager.
- The name of the database table that supports the form, and its primary keys. One way to obtain the table name is to open the form in question, query a record, select Record History from the Help tool, and view the table name in the About This Record dialog. If the dialog provides the name of a database view (if the name ends in *_V*), you'll need to obtain the database name elsewhere. For the APXVDMVD form, the database table is PO_VENDORS and the primary key is VENDOR_ID.
- For each primary key, the name of the corresponding block and field in the form being audited. One way to obtain this information is to open the form in question and select the Diagnositcs/Examine option of the Help menu. For example the block and field for the VENDOR_ID primary key column is VNDR.VENDOR_ID.
- The application ID for the application that runs the audited form. (For example, the ID for the Payables application is 201.) To find the value, run this SQL query:

```
SELECT APPLICATION _ID FROM FND_TABLES WHERE TABLE_NAME = 'table name'
```

Creating the AppsForm Rule

Open AppsForm to create a link from the audited form to the online audit form. (For detailed information on using AppsForm, see the *AppsForm Training Guide*.)



After entering a name and description for the rule, enter the following in the Rule Elements block:

- Sequence: Enter a number that reflects the order in which you want this element to be performed with respect to any other elements.
- Element Name: Type a name for the rule element.
- Form Name: From the list of values, select the name of the form you are auditing — for example, APXVDMVD if you are auditing the Enter Vendor form.
- User Form Name: Accept the value supplied by AppsForm.
- Event: Select the value *When New Form*.

Once these values are selected, click on the Details button and, in the Business Rule Details form, click on the Navigation tab. Then enter these values:

- Sequence: Select a sequence number with a *Tools* prefix to place the link on the Tools menu. (Select a high value to avoid the possibility of overwriting existing Tools-menu options.)
- Label: Type a name for the link. This name will appear as an option on the Tools menu.
- To Function: Select the value *AppsAudit Online* from the list of values.
- Parameters: Supply the name of the database table that supports the audited form; the names of its primary key columns and, for each, the corresponding block and field in the audited form; and the application ID for the application that runs the audited form. Optionally, specify a number of days of audited data to display, or select columns for display. See the following discussion for more on these values.

Parameter Values

When you select AppsAudit Online as the To Function, AppsForm automatically presents the following template in the Parameters field:

```
P_TABLE_NAME=<AUDIT TABLENAME>
P_PRIMARY_KEYS=!*!<KEY_COLUMN1>=#<BLOCK.FIELD1>#!~!<KEY_COLUMN2>=
#<BLOCK.FIELD2>#!~!!*! P_APPLICATION_ID=<APPLICATION_ID>
```

If the table has one primary key, delete this block from the template:

```
<KEY_COLUMN2>=#<BLOCK.FIELD2>#!~!
```

If the table has two primary keys, leave the template as it is; if the table has three or more primary keys, add in a new block for each key. In any case, replace the placeholder names — the text enclosed by angle brackets (and the angle brackets themselves) — with approriate values.

For example, the parameter statement for the Enter Vendors form (whose appropriate values are discussed on pages 25–26) would be:

```
P_TABLE_NAME=PO_VENDORS P_PRIMARY_KEYS=!*!VENDOR_ID=
#VNDR.VENDOR_ID#!~!!*! P_APPLICATION_ID=201
```

If the primary key value is of the varchar data type or includes spaces, enclose the entire primary-key value in double quotation marks and the block and field name value in an additional pair of single quotation marks, as shown:

```
P_TABLE_NAME=<Audit_Tablename> P_PRIMARY_KEYS="!*!<Primary Key
Name>='#<Block.Field Name>#'!~!!*!" P_APPLICATION_ID=<Application_ID>
```

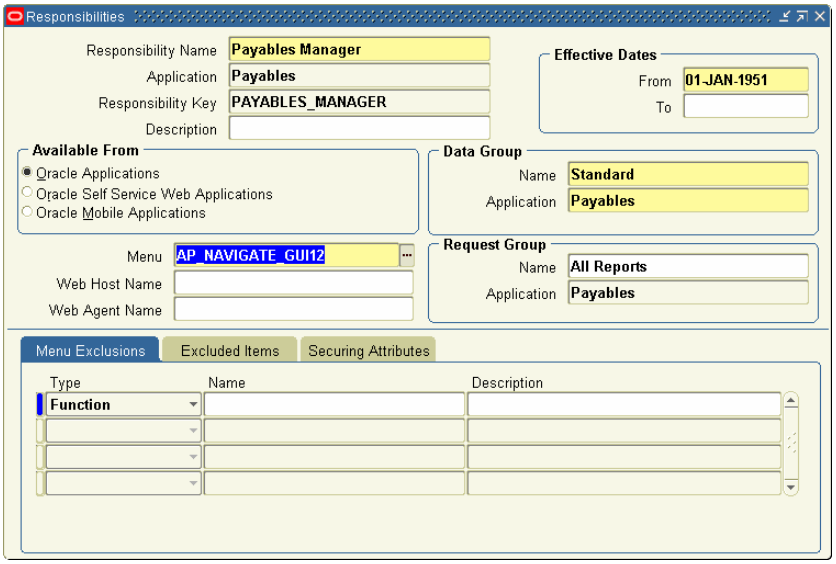
You can add parameters that specify the number of days worth of audit data to present and select columns about which to display data. These optional parameters take the following form, and follow the application ID parameter:

```
P_AUDIT_COLUMNS=!*!<Audit_Column>!*!<Audit_Column> P_NUM_OF_DAYS=
<Number_of_days>
```

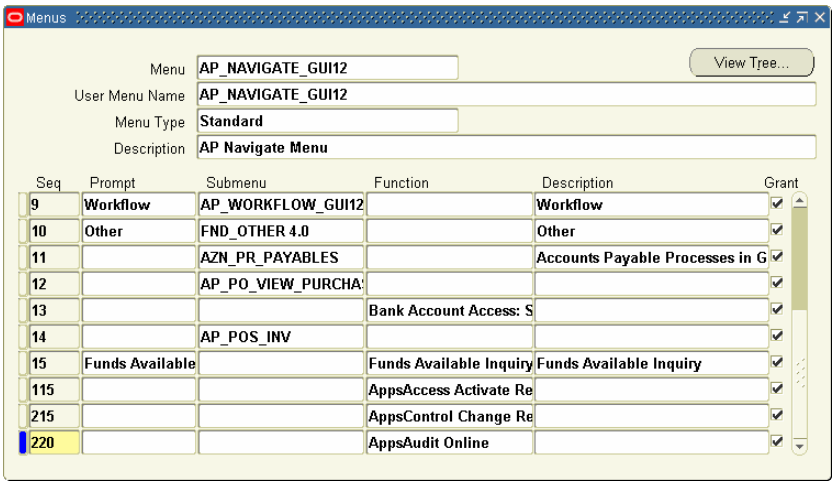
Adding Online Audit to a Responsibility

To add the AppsAudit Online function to the menu structure of the responsibility that is to have access to the online audit form, complete these steps:

1. Determine the responsibility whose menu structure is to be altered. Assuming you know the name of the application that runs the form that is to be audited:
   - Open the Responsibilities form in the System Administrator responsibility.
   - Query on the Application field — press F11, type the application name in the Application field, and press Ctrl+F11.
   - The form loads all the responsibilities from which the application can be run. Press the up- or down-arrow button to scroll through them.
   - Stop at the appropriate responsibility — one assigned to a user (or users) who need to view the online audit form.
2. In the Responsibilities form, note the name of the menu associated with the responsibility you've selected — for example, AP_NAVIGATE_GUI12 for the Payables Manager responsibility.



3 Close the Responsibilities form and open the Menus form. Query for the menu whose name you've just identified:

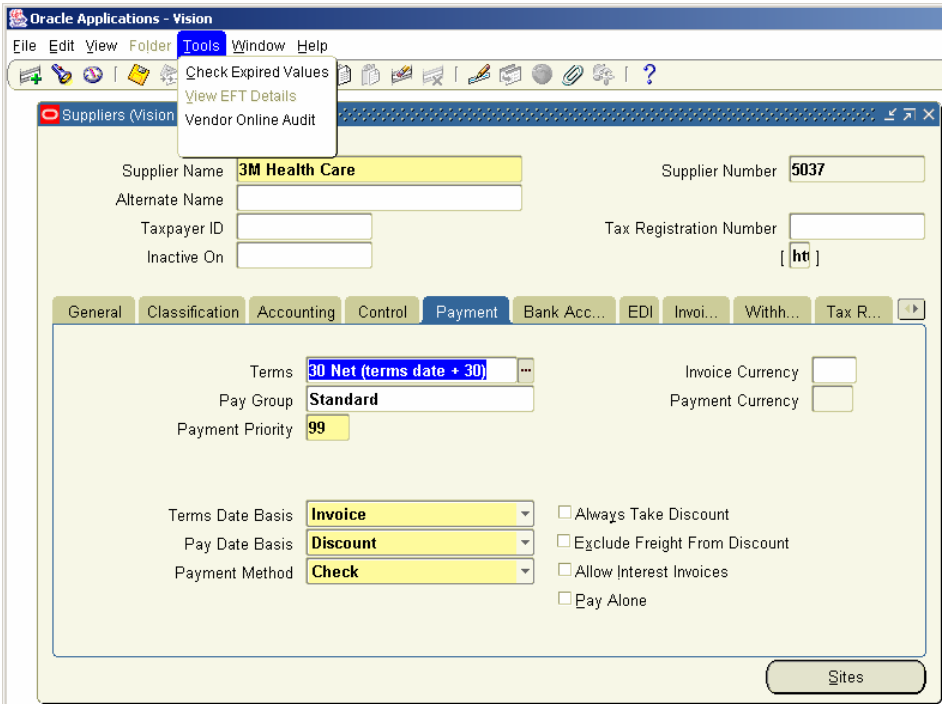


4 Add a row to the grid (click on File in the menu bar, then on New in the File menu). Add a sequence number in the Seq field, and select AppsAudit Online in the Function list of values. (Do not enter a value for Prompt or Submenu.)

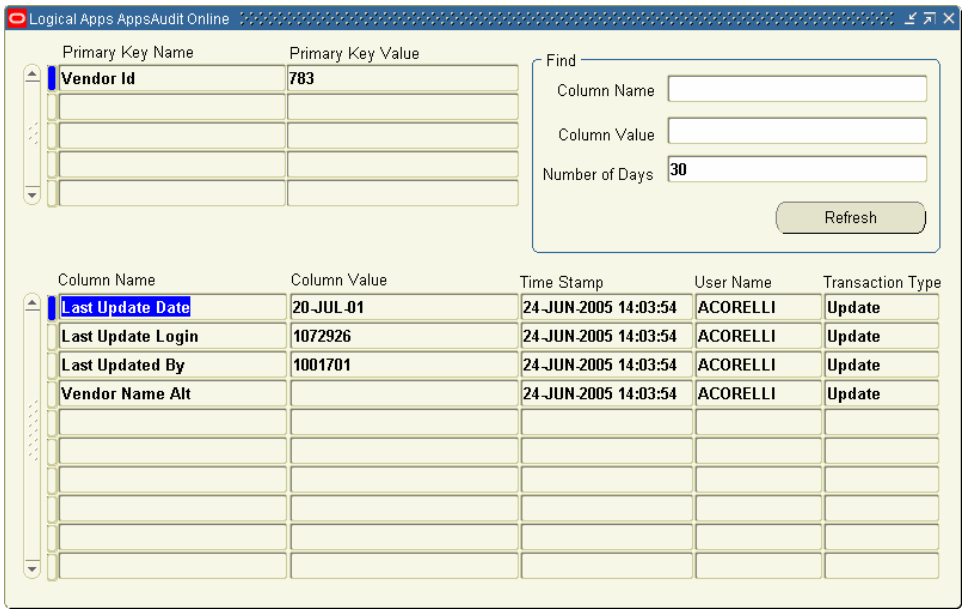
5 Click on File in the menu bar, then on Save in the File menu.

Viewing the Online Audit

Ensure that the table supporting the form for which you've created the link is included in an audit group. Then navigate to the form and click on the Tools menu. It should include the link you've created:



Click on the link to open the AppsAudit Online form:



Primary key values include the following:

- Primary Key Name — Primary key of table being audited.
- Primary Key Value — Value of primary key being audited, or translated value if defined.

Detailed audited data values include the following:

- Column Name — Column being audited.
- Column Value — Value of column being audited.
- Time Stamp — Time data was audited.
- User Name — User that changed the data.
- Transaction Type — Whether a new value updates a previously entered value or fills a field that had previously been empty.

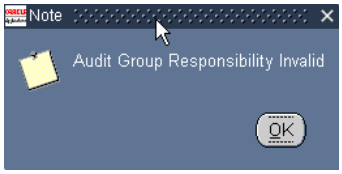
Find criteria include the following:

- Column Name —Search for audited data by column name. Wild cards (%) may be used. For example, to show only the Description column, enter Desc% in the Column Name and click Refresh.
- Column Value —Search for audited data by column value. Wild cards (%) may be used. For example, to show audited data where value is like INV, enter INV% in the Column Name and click Refresh.
- Number of Days — The value 1 indicates audited data with the current date for an audit timestamp; the value 2 indicates audited data from the current day and the previous day; and so on. The default number is 30.

Note
Translated values are displayed if they have been defined in the Audit Setup

Security

The online form opens only if the user is currently in the responsibility to which the table has been restricted. The following message appears if the current responsibility does not have access to view audited data for the table:



Note
If a table exists in more than one group, all restricted responsibilities are considered. The online audit form shows data only if the user is currently in one of the restricted responsibilities.

Chapter 6

Audit Migration

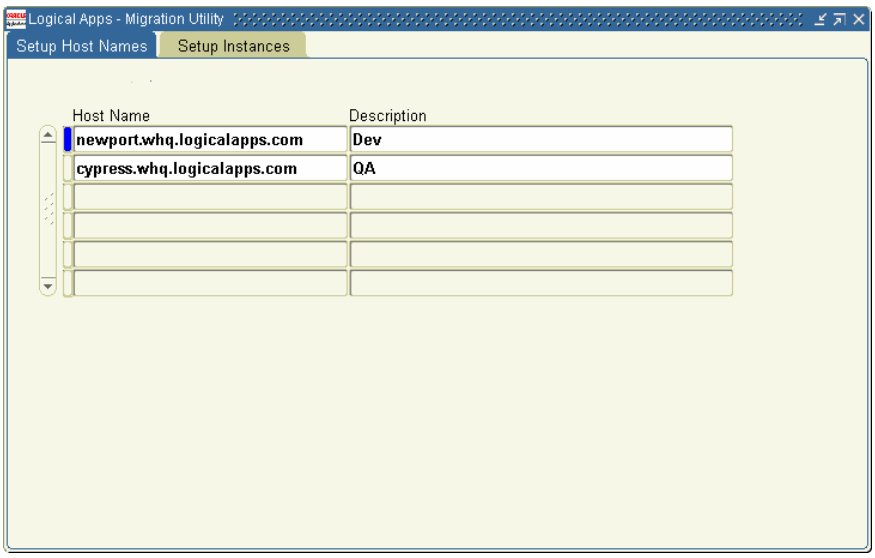
Once you have created audit groups for an instance of Oracle Applications, you can "migrate" them — copy an audit group, the auditing instructions for an individual table within a group, or a report directly to another Oracle Applications instance. You can also export groups and tables to, or import them from, XML files.

Preparing for Migration

Before you can migrate audit groups and tables, you need to specify connection information in all the environments to and from which you plan to transmit data. You need to know the host name, instance SID, and database instance port for each environment. This information is found in the TNSNAMES.ora file, which is located in *ORACLE_HOME*/network/admin.

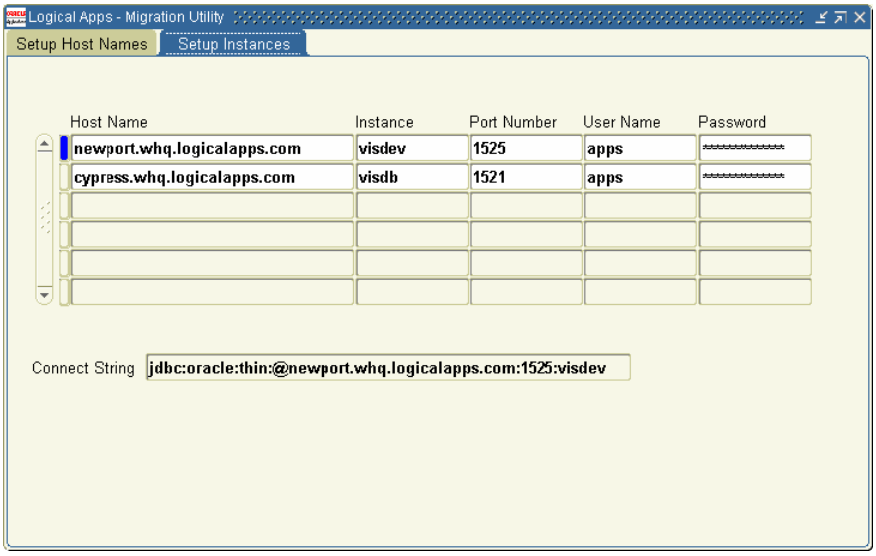
Once you've gathered this information, use the Logical Apps Migration Utility to perform the connectivity configuration:

1. With AppsAudit open (see page 2), click on LogicalApps Utilities in the menu bar, then on Migration Setup in the Utilities menu. A Migration Utility form appears (as shown at the top of the next page).
2. Ensure that the Setup Host Names tab is selected.
3. In the Host Name column enter the host name (machine name) for each of the machines hosting the database and involved in audit migration.



4 In the Description column, you may enter a description for each host name. (This step is optional.)

5 Click on the Setup Instances tab. The following form appears:



6 In the Host Name column, select the host name for each of the machines from the list of values. (The entries are those defined in the Setup Host Names tab).

7 In the Instance and Port Name columns, type the instance name and port number that corresponds to each host name.

8 Under User Name, type the value *apps* for each entry. Under Password, enter the password for the apps user.

9 Click on File in the menu bar and Save in the File menu. When the configuration is saved, the system automatically generates and displays a connection string.

10 Close the Migration Utility: Click on the × symbol in the upper right corner of the form.

Dependencies

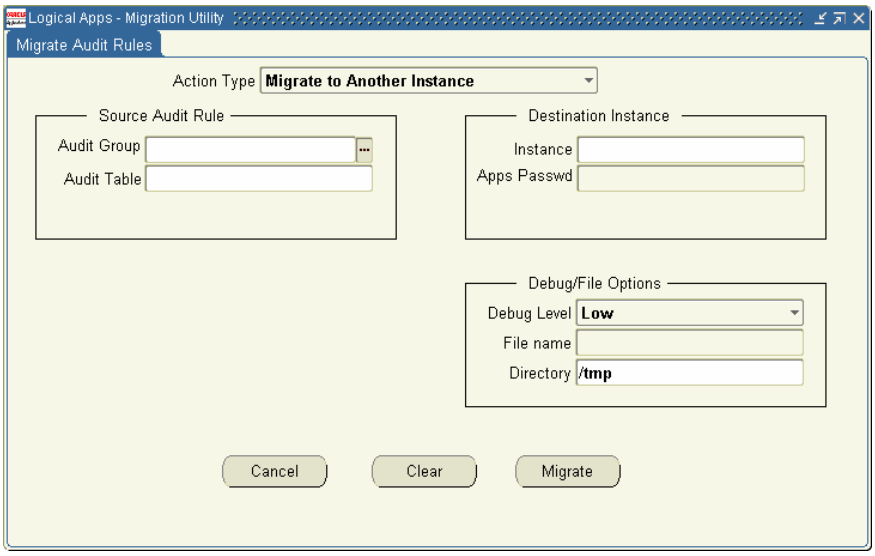
The following conditions apply to migration, export, and import operations:

- For a table or a report to be migrated, its audit group must already exist on the destination instance.
- For an instance-to-instance online migration, the ID of the person who created an audit group, table, or report in the source instance must exist in the destination instance. (However, the user's status on the destination instance may be active or inactive. Audit migration does not validate whether the user is active.)
- For an XML file import, the user ID of the person who created an audit group, table, or report need *not* exist in the destination instance. The CREATED_BY and LAST_UPDATED_BY fields are updated with the ID of the person who performs the file import.
- A log file gathers information about a migration, export, or import operation. If an operation fails and you are unable to determine why, rerun the operation with the debug level changed from low to high and evaluate the log data.
- Commonly, problems with migration result from missing translations. In such cases, the audit log shows errors as INVALID. For instance, if a table or a responsibility does not exist in the destination, a migration error occurs.

Migrating or Exporting a Group or Table

To migrate an audit group, or a table from the group, to another instance, or to export a group or table to an XML file, complete the following steps:

1 With AppsAudit open (page 2), click on LogicalApps Utilities in the menu bar, then on Migrate Rules in the Utilities menu. The Migrate Audit Rules form appears:



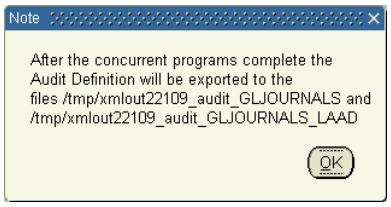
2. In the Action Type list box, select the operation you want to perform:
   - Migrate to Another Instance (the default) if you want to migrate a group or table.
   - Export to File if you want to export a group or table to an XML file.
3. In the Audit Group list of values, select the group you want to migrate or export.

   If a group has been opened in the AppsAudit Audit Groups form, that group appears here by default when you open the Migrate form.
4. If you want to migrate or export the group you selected in step 3, leave the Audit Table list of values blank. If you want to migrate or export a table from that group, select the table in the Audit Table list of values.

   If a group has been opened in the AppsAudit Audit Groups form and the cursor is focused on a table listed in its Table Name column, that table appears here by default when you open the Migrate form.
5. If you are performing a migration, make entries in the Destination Instance block:
   - In the Instance list of values, select a destination instance for the migration.
   - In the Apps Passwd text box, type the apps password for the destination instance if you are prompted to do so. (This prompt appears if the XXLAAPPS: Enable for Migration Security profile option is set to Yes on the source instance. If the option is set to No, the prompt does not appear and a password need not be entered.)

   If you are performing a file export, fields in the Destination Instance block do not apply and do not accept input.
6. In the Debug Level list box, select a level of detail for error reporting to a log. Ordinarily, select Low; select High instead if you need to uncover the cause of a failed migration or export.
7. In the Directory text box, type the path that designates a temporary staging file location for XML files to be generated and, in the case of migration, copied to the destination instance.
8. Click on a button that launches the process. Its label varies depending on the selection you made in step 2: Migrate if you chose Migrate to Another Instance or Export if you chose Export to File.
9. Review several messages:
   - Depending on the parameters you have specified, the system may launch one or two concurrent programs to implement the migration or export. For each concurrent program, a message provides an ID number. Click on the OK button in each to clear each message.
   - If you have performed a file export, the system may create one or two files (again depending on the parameters you have specified). AppsAudit generates the name for these files, and a message similar to the following one displays

the names of the files you have generated. In each file name, the term *xmlout* designates XML output, a number (22109 in this example) serves as a unique identifier for an export operation, the term *audit* identifies the Logical Apps component involved in the export operation, and a last phrase (*GLJOURNALS* in this example) identifies the audit group (or group and table, if appropriate) that is being exported.



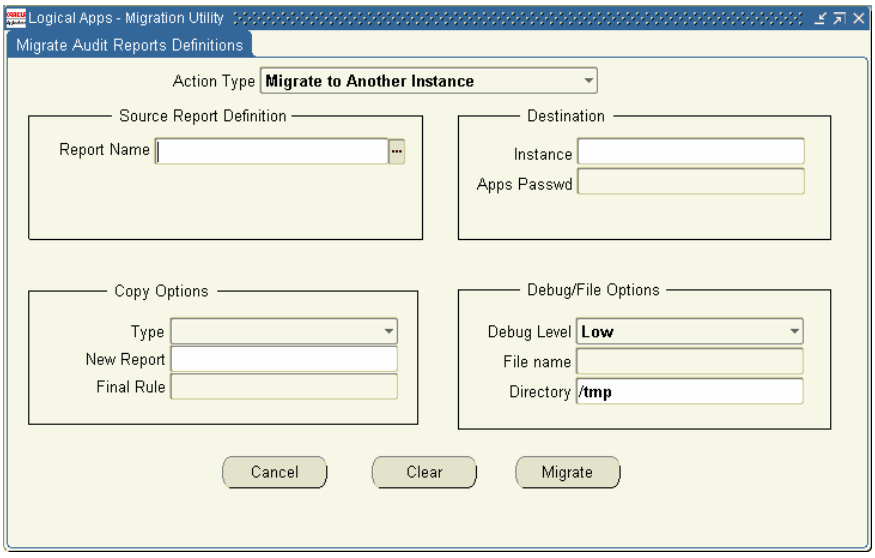
Make a note of file names and locations, and click the OK button to clear the message.

- Finally, a dialog prompts you to perform another migration. Click Yes to do so or No to close the Migration form.

Migrating or Exporting an Audit Report

Follow a similar process to migrate a report or export it to an XML file:

1. With AppsAudit open (page 2), open the Audit Reports form (page 17). Click on Actions in the menu bar, then on Migrate Report Definitions in the Actions menu. The Migrate Audit Reports Definitions form appears:



2. In the Action Type list box, select the operation you want to perform:
   - Migrate to Another Instance (the default) if you want to migrate a report.
   - Export to File if you want to export a report to an XML file.
   - Copy within the Same Instance if you want to copy the report under a new or modified name on the source instance.

3 In the Report Name list of values, select the report you want to migrate or export.

If a report has been opened in the Audit Reports form, that report appears here by default when you open the Migrate Reports form.

4 If you are performing a migration, make entries in the Destination block:

- In the Instance list of values, select a destination instance for the migration.
- In the Apps Passwd text box, type the apps password for the destination instance if you are prompted to do so. (This prompt appears if the XXLAAPPS: Enable for Migration Security profile option is set to Yes on the source instance. If the option is set to No, the prompt does not appear and a password need not be entered.)

If you are performing a file export or copying a report to the source instance, fields in the Destination block do not apply and do not accept input.

5 If you are copying the report to the source instance, make entries in the Copy Options block:

- In the Type list box, select Copy as a New Report if you want to assign a completely new name to the copy, or select Prefix or Suffix if you want to assign a name to the copy that consists of the original report name with text added at the beginning or end.
- In the second box (labeled New Report Name, Prefix, or Suffix, depending on the selection you made in the Type list box), type the text you want to use as a new report name or as a prefix or suffix to the existing name.
- The Final Rule box does not accept input.

If you are performing a migration or a file export, fields in the Copy Options block do not apply and do not accept input.

6 In the Debug Level list box, select a level of detail for error reporting to a log. Ordinarily, select Low; select High instead if you need to uncover the cause of a failed migration or export.

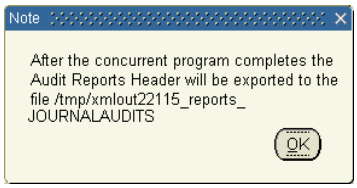
7 In the Directory text box, type the path that designates a temporary staging file location for XML files to be generated and, in the case of migration, copied to the destination instance.

8 Click on a button that launches the process. Its label varies depending on the selection you made in step 2: Migrate if you chose Migrate to Another Instance, Export if you chose Export to File, or Copy if you chose Copy within the same Instance.

9 Review several messages:

- The system launches a concurrent program to implement the migration, export, or copy. The first message provides an ID number. Click on the OK button to clear the message.
- If you have performed a file export, a message similar to the following one displays the name of the export file you have generated. In the file name, the

term *xmlout* designates XML output, a number (22115 in this example) serves as a unique identifier for an export operation, the term *reports* identifies the Logical Apps component involved in the export operation, and a last phrase (*JOURNALAUDITS* in this example) gives the report name.



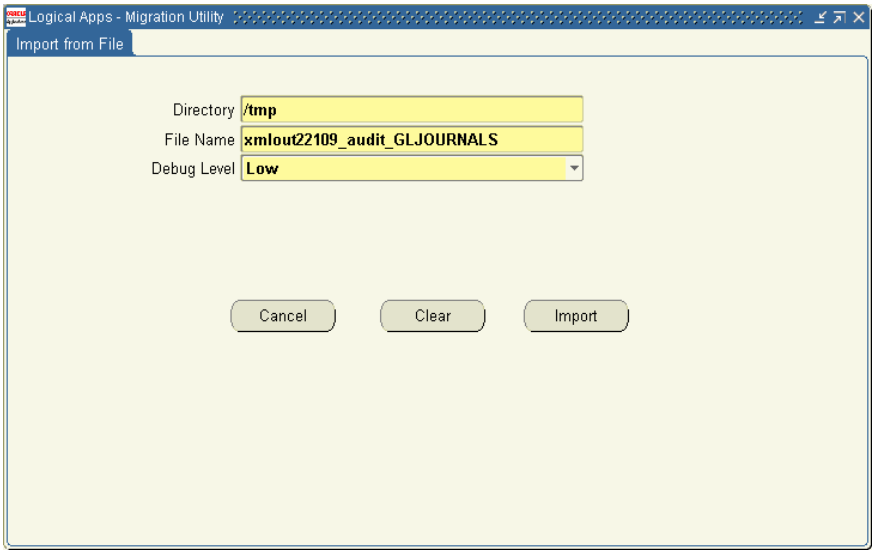
Make a note of file name and location, and click the OK button to clear the message.

- Finally, a dialog prompts you to perform another migration. Click Yes to do so or No to close the Migration form.

Importing a Group, Table, or Report File

To import an XML file containing any of the audit components — a group, a table, or a report — complete these steps. Remember that in some cases an export operation prepares more than one file, and in such cases you would need to import all of the files prepared in an export.

1. Transmit exported files via FTP to the destination OS for import.
2. With AppsAudit open (page 2), click on LogicalApps Utilities in the menu bar, then on Import from File in the Utilities menu. An Import From File form appears:



3. In the Directory box, type the path to the folder that contains the import file or files.

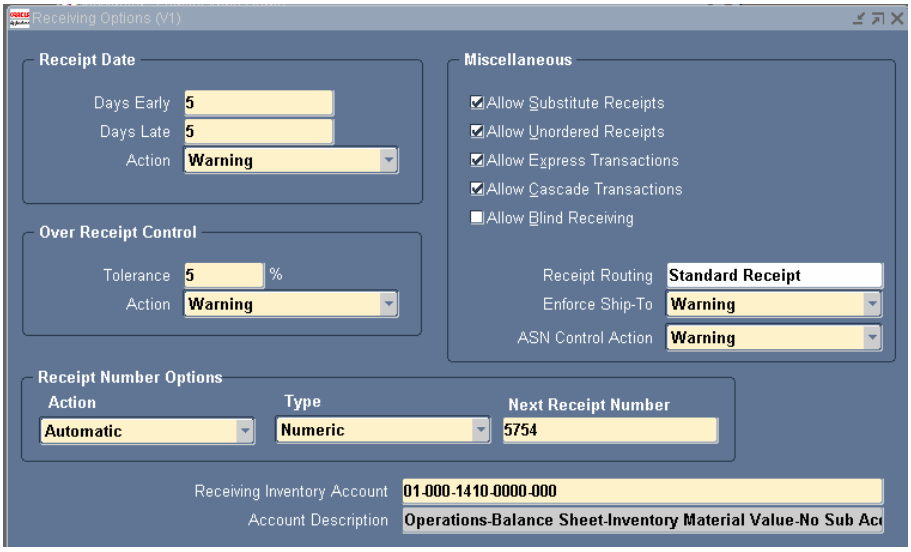
4. In the File Name box, type the name of a file you want to import. This would be a name displayed by a message at the culmination of a file export (pages 37 and 39).
5. Select a value for Debug Level. Ordinarily, select Low; select High instead if you need to uncover the cause of a failed import.
6. Click on the Import button. A concurrent request message displays the ID number of the concurrent program that implements the import. Click on the OK button to clear the message.
7. If you need to import more than one file, repeat this process as necessary.

Appendix A

Determining Tables to Audit

The first step in auditing data is to determine the tables where your data resides. To do this:

1. Navigate to the form that contains data you want to audit. For example, assume you want to audit the Days Early, Days Late, and Receipt Routing fields in the Receiving Options form:



2. Select Help from the menu bar, then Record History from the Help menu. The About This Record window appears.

3 Note the name of the table or view that supports the form. Usually a view has _V at the end, as in this example.

Because a view cannot be audited, you will need to perform additional research if the form fields you are trying to audit are based on a view. At this point, you may need to work with someone in your organization who is familiar with describing views.

In this example, additional research led to RCV_PARAMETERS, the name of the table to audit.

Appendix B

Auditing Prerequisites

A table being audited must have a unique primary key. In some cases, the primary key Oracle has registered for the table is not unique and can be changed. Changing primary key values after data has been audited will cause data integrity issues.

Verify Primary Key Is Unique

The easiest way to verify whether a primary key is unique is to check the first unique index that has been setup for that table. In most cases, this will be the same as the primary key. For instance, the following query shows a unique index used for a table.

```
select fc.column_name
  from fnd_columns fc,
            fnd_index_columns fic,
            fnd_indexes f
 where fic.column_id = fc.column_id
       and fic.index_id = f.index_id
   and f.index_name like '%U1%'
   and f.table_id = (select table_id
                     from fnd_tables
                     where table_name = 'AR_APPROVAL_USER_LIMITS')
```

This query returns five columns that make up a unique key:

```
COLUMN_NAME
CURRENCY_CODE
USER_ID
DOCUMENT_TYPE
REASON_CODE
```

The primary key for AR_APPROVAL_USER_LIMITS is only defined as USER_ID.

Problem

If the other columns are truly part of the primary key, then they would not be changeable and in the form. You can change these values.

Suggestion

Create a rule in AppsForm that does not allow the five columns that make up the new primary key to be changed.

Use Oracle APIs to register the additional fields as primary keys.

Support

Logical Apps offers many services to assist you with the AppsAudit implementation. From on-site support to remote phone and web support, our team of experienced professionals provides the help and information you need to ensure quick and effective implementation. The Logical Apps team includes a Technical Support Representative, an Account Manager, and a Logical Apps staff consisting of consultants and support specialists.

Feedback

Thank you for using Logical Apps AppsAudit. We value your comments and feedback. Mail your comments to the following address, or call us directly at (949) 453-9101.

Logical Apps
Attn: Documentation Management
15420 Laguna Canyon Road
Suite 150
Irvine, CA 92618
U.S.A.

Or send email to support@logicalapps.com.