INTEGRIGY

WHITE PAPER

Guide to Auditing in Oracle Applications

JANUARY 2006

GUIDE TO AUDITING IN ORACLE APPLICATIONS

March 2003
February 2004 – Updated
September 2005 – Updated
January 2006 – Updated

Authors: Stephen Kost

If you have any questions, comments, or suggestions regarding this document, please send them via e-mail to alerts@integrigy.com.

The Information contained in this document includes information derived from various third parties. While the Information contained in this document has been presented with due care, Integrigy Corporation does not warrant or represent that the Information is free from errors or omission. The Information is made available on the understanding that Integrigy Corporation and its employees and agents shall have no liability (including liability by reason of negligence) to the users for any loss, damage, cost or expense incurred or arising by reason of any person using or relying on the information and whether caused by reason of any error, negligent act, omission or misrepresentation in the Information or otherwise. Furthermore, while the Information is considered to be true and correct at the date of publication, changes in circumstances after the time of publication may impact on the accuracy of the Information. The Information may change without notice.

Table of Contents

1. Introduction

1.1 Introduction

This guide describes the auditing capabilities available in Oracle Applications and the Oracle database. It is basically a “how to” for configuring and using the auditing features within Oracle Applications. This guide will attempt to provide a clear explanation of the features available, simple configuration steps, and best practices for auditing within Oracle Applications.

The auditing capabilities within Oracle Applications are sophisticated and able to satisfy most organizations' requirements. However, the auditing setup and usage of audit data can be complex and error-prone.

Most implementations do not fully take advantage of the auditing features due to the complexity and perceived performance issues. Properly configuring auditing and only auditing appropriate tables will not have any measurable performance impact.

1.2 Scope

All the information in this guide is related to Oracle Applications 11i (11.5.7 – 11.5.10 CU2). It should also pertain and be relevant for versions 11.0 and R12, although this has not been validated or tested.

This guide covers the technical aspects of configuring and using auditing within Oracle Applications. The intended audience is Oracle Applications DBAs, application administrations, and internal audit staff. It does not provide any information on functional aspects of auditing. Suggestions are provided for auditing of common application objects like users and responsibilities, but no information is provided for functional modules such as General Ledger or Account Payable.

For more detailed information on auditing within Oracle Applications, see the Oracle Applications System Administration Guide Chapter 3. Other references are available in Chapter 7 of this document.

2. Auditing in Oracle Applications

2.1 Overview

Oracle Applications is delivered with several different auditing features. These auditing features provide many capabilities, but also can be disjointed and overlapping. Some are automatically enabled while others require configuration and maintenance.

There are four basic methods of auditing within Oracle Applications:

1. Database auditing
2. Database row
3. End user access
4. Database row changes

Database Auditing

The Oracle database has sophisticated auditing capabilities and can audit virtually any action within the database. Oracle9i introduces Fine Grained Auditing which allows auditing down to the column level in queries. The Oracle database auditing features and Oracle Applications auditing features are completely separate mechanisms.

By default, only "Instance Startup", "Instance Shutdown", and connections to the database with administrator privileges are audited.

Usually only session and database user auditing are enabled. Few other database auditing features are used by most Oracle Applications implementations, unless there are special requirements. Most auditing functions are performed within Oracle Applications.

Database Row (Creation and Last Update)

Almost every row of data within Oracle Applications is automatically tagged with the following information:

- Creation Date
- Created By
- Last Update Login
- Last Update Date
- Last Updated By

Only the creation and information about the last update is stored. Audit trail information is not stored, thus any updates between the creation and last update are lost. Also, no details regarding which columns were updated are stored.

This information can be easily accessed through the application or SQL. Within the applications, select ***Help > Record History*** from the main menu to view the information for the current record.

End User Access

Oracle Applications stores information regarding user logon, unsuccessful password attempts, responsibility selection, form usage, and concurrent process execution. By default, Oracle Applications only stores basic user logon information, unsuccessful password attempts, and concurrent process execution. More detailed logon information, responsibility selection, and form usage must be enabled through a system profile option. This information can be accessed through standard reports or SQL.

It is important to note that the user logon information may not be complete (missing stop times) if the user session terminates unexpectedly (turn off PC, applet crash, etc.).

Database Row Changes (AuditTrail)

Any Oracle Applications table can be selectively chosen to have all row changes audited. Auditing can be done at the row or column level. By default, no tables are audited. Auditing frequently accessed tables (especially transactional tables) can cause severe database performance issues, thus auditing should be setup very carefully.

Audit trail information is stored in separate database tables detailing the user information and types of updates. The Oracle Applications AuditTrail functionality only tracks inserts, updates, or deletes, whereas the database's auditing capability can also track selects on tables.

3. Database Auditing

3.1 Overview

Database auditing is useful for capturing information not directly logged or audited within Oracle Applications. There are several risk areas that should be monitored using database-level auditing – (1) connections to the database and (2) changes to Oracle database accounts. In most environments, other auditing does not provide substantial information or benefits.

Database Connections and Sessions

Monitoring and auditing database sessions provides valuable information on database activity and is the only way to identify certain types of attacks (e.g., brute force password attempts against database accounts). The standard Oracle database user account and password controls (i.e., lockout controls) should not be used with Oracle Applications, thus many types of attacks can go undetected without session auditing. Also, the APPLSYSPUB account is vulnerable to attack and gives an attacker significant information about the configuration of the database and application. By auditing database sessions, any inappropriate connections using the APPLSYSPUB account can quickly be identified.

Changes to Database Accounts

Any changes to the standard Oracle Applications database accounts or creation of new accounts should be audited. Such changes are rare and can indicate either inappropriate activity.

Other Auditing

A few other audits should be enabled, such as ***create database link*** and ***alter system*** as these activities are not routinely executed and may indicate inappropriate activity.

Auditing other database activities usually generates significant amount of audit entries of little value. Oracle Applications dynamically creates, alters, and drops objects (tables, index, packages, etc.) on a regular basis, thus auditing these types of actions does not provide meaning information.

Fine-Grained Auditing in Oracle9i

Oracle9i introduced a new feature called “fine-grained auditing”. Fine-grained auditing allows detailed conditions to trigger auditing. For example, all access to a salary table when the salary is greater than $100,000 can be audited. Fine-grained auditing may be useful for specific auditing situations, but is not suggested for daily auditing activities.

3.2 Configuring

There are basically two steps required to enable database-level auditing – (1) set the ***audit_trail*** parameter in init.ora and (2) execute audit statements.

Step 1 – Set “audit_trail” parameter

In the init.ora file for the instance, set the ***audit_trail*** parameter to ***true***. This will enable auditing to the SYS.AUD$ table. Optionally, audit information can be written to the operating system by setting ***audit_trail*** parameter to ***OS***.   The database must be restarted for this parameter to take effect.

Step 2 – Execute audit statements

Execute the following SQL statements as the SYSTEM user –

Audit Statement	Description
`audit session;`	Session auditing – connects to the database
`audit user;`	Create, alter, and drop user
`audit database link;`	Create or drop database links
`audit public database link;`	Create or drop public database links
`audit system audit;`	Audit and noaudit statements

To review the existing system-level audits –

```
select * from DBA_STMT_AUDIT_OPTS;
select * from DBA_PRIV_AUDIT_OPTS;
```

3.3 Accessing

Administrative Users and Actions

Connections to the database and actions (instance startup/shutdown) as SYSDBA or SYSOPER are always logged to the directory $ORACLE_HOME/rdbms/audit. This file contains the operating system user name and terminal ID.

Database Audit Trail

Database audit trail information is logged in the table SYS.AUD$. It is important to note that by default the SYS.AUD$ table can be queried by any user.

Session audit information can be accessed via the database view DBA_AUDIT_SESSION.

There are no standard reports or queries to extract audit information.

3.4 Purging

The ***SYS.AUD$*** table needs to be purged on a periodic basis, at least every 90 days. The database connection entries can be significant. All rows should be backed up prior to being purged.

4. Apps Auditing – Database Row

4.1 Overview

Database rows are automatically tagged with creation and last update information for almost all Oracle Applications tables.

The following columns are in almost every Oracle Applications table –

Column	Description
CREATION_DATE	Date and Time row was created
CREATED_BY	Oracle Applications user ID from FND_USER
LAST_UPDATE_LOGIN	Login ID from FND_LOGINS
LAST_UPDATE_DATE	Date and Time row as last updated
LAST_UPDATED_BY	Oracle Applications user ID from FND_USERS

It is important to note that any changes to the row between the creation and last_update are not saved – only the creation information and last time the record was updated will be saved. To save all update history for a row, Oracle Applications Audit Trail must be enabled for the table (see Apps Auditing – Database Row Changes).

4.2 Configuring

There is no configuration required.

4.3 Accessing

There are two methods to access the creation and last update information – from within Oracle Applications or through SQL.

Using Oracle Applications

The database row auditing is the simplest to access. From Oracle Applications, query the row you would like to check the creation or last update information. Be sure the cursor is in the correct block. If the form has header and detail information, information can be obtained for either. From the menu, select ***Help > Record History***. The system login and terminal values should usually be oracle and ?, since all users (except ADI) will be using the applications servers to access Oracle Applications. Records entered through ADI, interfaces, or other external systems may contain different values.

Using SQL

Creation and update information for any row can be found by joining the row with the FND_USERS and FND_LOGINS tables.

4.4 PURGING

No purging of database row history is required.

5. Apps Auditing – End User Access

5.1 Overview

Since end user authentication is managed at the application level, all end user access information is stored in Oracle Applications' tables. By default, only minimal audit data is saved. Standard reports and purge programs exist allowing easy access to the audit data.

5.2 Configuring

User, Responsibility, and Form Access

The system profile option "Sign-On: Audit Level" controls the level of end user access auditing. This profile option only controls the auditing information from logins into the Professional Interface (Forms).

The valid settings are None, User, Responsibility, and Form. This profile option should always be set to "Form" to enable the most auditing.

All user signons, responsibility selections, and form accesses will be logged to APPLSYS.FND_LOGINS, APPLSYS.FND_LOGIN_RESPONSIBILITIES, and APPLSYS.FND_LOGIN_RESP_FORMS, respectively.

Unsuccessful Logins

Unsuccessful password attempts are automatically recorded in the APPLSYS.FND_UNSUCCESSFUL_LOGINS and ICX.ICX_FAILURES tables. There is no way to disable this functionality.

Concurrent Requests

All concurrent requests are recorded in the FND_CONCURRENT_REQUESTS table. There is no configuration required.

5.3 Accessing

There are two methods to access the standard end user access audit information – from within Oracle Applications or through SQL.

Using Oracle Applications

There are standard reports to access signon, unsuccessful signon, responsibility usage, form usage, and concurrent request usage. These reports are accessed through the system administrator responsibility.

The following are the standard reports for end-user auditing –

- Signon Audit Users
- Signon Audit Responsibilities
- Signon Audit Forms
- Signon Audit Concurrent Requests
- Signon Audit Unsuccessful Logins

Using SQL

The end-user access data is stored in the following tables –

APPLSYS.FND_LOGINS
APPLSYS.FND_LOGIN_RESPONSIBILITIES
APPLSYS.FND_LOGIN_RESP_FORMS
FND_CONCURRENT_REQUESTS
APPLSYS.FND_UNSUCCESSFUL_LOGINS
ICX.ICX_FAILURES

Unsuccessful logins via the Personal Home Page (Self Service/Web Interface) are stored in both the FND_UNSUCCESSFUL_LOGINS and ICX_FAILURES tables. The ICX_FAILURES table contains more information than the FND_UNSUCCESSFUL_LOGINS. Failed logins to the Professional Interface (Forms) are only logged to the FND_UNSUCCESSFUL_LOGINS tables.

5.4 Purging

The end-user access data can be purged using the ***Purge Signon Audit Data*** concurrent program. The only parameter is a date which all audit data older this date is purged. This program should be scheduled to be run weekly or monthly and saving at least 30-90 days of data.

The following tables are purged by this program –

FND_LOGIN_RESP_FORMS
FND_LOGIN_RESPONSIBILITIES
FND_LOGINS
FND_UNSUCCESSFUL_LOGINS

Concurrent request data is purged using the ***Purge Concurrent Request and/or Manager Data*** concurrent program. This program should be scheduled to be run at least weekly and saving at least 14-90 days of data.

6. Apps Auditing – Database Row Changes

6.1 Overview

Oracle Applications uses its own auditing mechanisms rather than Oracle database auditing features. The auditing capabilities are sophisticated and can satisfy most auditing requirements. However, the auditing configuration is complex and error-prone. There are no standard reports available to access the audit data.

Oracle Applications "AuditTrails" maintain a full history of changes made at a table and column level. The AuditTrails are enabled by a shadow table (table name appended with _A) of the audited table and triggers on the audited columns. A concurrent program is used to create the shadow table and triggers.

Auditing database row changes is a very performance intensive and can cause significant database performance problems. Careful planning and reviews with a DBA should be performed before enabling any auditing. Only a minimal amount of auditing should be done and limited only to non-transactional data. Auditing on transactional data may cause significant performance degradation of the entire application. Tables with more than a few changes an hour should not be considered for row level auditing.

The following AOL tables should be considered for auditing –

Table Name	Security Priority
FND_AUDIT_GROUPS FND_AUDIT_SCHEMAS FND_AUDIT_TABLES FND_AUDIT_COLUMNS	Minimal
ALR_ALERTS	Standard
FND_ORACLE_USERID	Standard
FND_USER FND_USER_RESP_GROUPS	Above Average
FND_FORM_FUNCTIONS FND_MENUS FND_MENU_ENTRIES FND_REQUEST_GROUPS FND_REQUEST_GROUP_UNITS FND_ENABLED_PLSQL FND_RESP_FUNCTIONS FND_CONCURRENT_PROGRAMS FND_EXECUTABLES FND_DATA_GROUPS FND_DATA_GROUP_UNITS	High

6.2 Configuring

Step 1 – Set AuditTrail Profile Option

The System Profile Option ***AuditTrail:Activate*** must be set to ***Yes.***

Be sure to log out of the applications to activate the profile option in your session.

Step 2 – Select the Audit Installations

- As System Administrator, select ***Security -> AuditTrail -> Install***.
- Check all the schemas for which auditing should be enabled. For example, if you want to audit FND_USERS, you would check APPLSYS since the FND_USERS table is in the APPLSYS schema.
- Save your selections.

Step 3 – Create a New Audit Group

- As System Administrator, select ***Security -> AuditTrail -> Groups***.
- Create a new audit group by setting the ***Application Name*** to the application that owns the table (e.g., Application Object Library for APPLSYS), the ***Audit Group*** to a new name (e.g., My Audits), and ***Group State*** should be set to ***Enable Requested***.
- Add the tables to be audited. Columns will be defined in the next step.
- Save the new audit group.

Step 4 – Define Table Columns to be Audited

For each table defined in the above step, define the columns to be audited using these steps –

- As System Administrator, select ***Security -> AuditTrail -> Tables***.
- Query the table name.
- The primary key columns will always be saved. Add the columns that need to be audited. Do not ever add the following columns as user information is automatically added –

  Creation Date
  Created By
  Last Update Login
  Last Update Date
  Last Updated By

- Save the columns.

Step 5 – Run AuditTrail Update Program

Run the ***AuditTrail Update Tables*** program to activate the auditing. This program will create a shadow table for each audited table and create triggers on each audited column in the original table. The shadow table will have the same name as the audited table appended with "_A". Two views will be created for each column with the names "_AC#" and "_AV#" where # is a sequential number.

Troubleshooting

See the Oracle Applications System Administration Manual Chapter 3 for more information on accessing the audit trail information. Metalink Note 105624.1 contains information on troubleshooting AuditTrail issues.

6.3 Accessing

Database row audit trail information can only be accessed through SQL. There are no standard Oracle Applications reports to access AuditTrail data.

The auditing information is stored in "shadow" tables for each audited table. The shadow tables are named tablename_A (e.g., FND_RESPONSIBILITY_A). There are also views for easier access to the audit data.

6.4 Purging

The audit trail information should be purged on a periodic basis. There is no standard purge program and the AuditTrail must be manually disabled to permit purging.

Use the following procedure to purge audit date –

1. As System Administrator, select ***Security -> Audit Trail -> Groups***
   a. Select the "Security Audit" group and set the group state to "Disable – Purge Table"
2. Run the "Audit Trail Update Tables" Report
3. Purge the data from the shadow table
4. Select ***Security -> Audit Trail -> Groups***
   a. Select the "Security Audit" group and set the group state to "Enable"
5. Run the "Audit Trail Update Tables" Report

7. REFERENCES

- *Oracle8i Administrator's Guide – Chapter 24 Auditing Database Use*
- *Oracle Applications System Administrator's Guide – Chapter 3 User and Data Auditing*
- Metalink Note 105624.1 – Troubleshooting (Audit Trail)
- Metalink Note 60828.1 – Overview of Oracle Applications AuditTrails
- Metalink Note 69660.1 – Understanding Data Auditing in Oracle Application Tables

About Integrigy

Integrigy Corporation (www.integrigy.com)

Integrigy Corporation is a leader in application security for large enterprise, mission critical applications. Our application vulnerability assessment tool, AppSentry, assists companies in securing their largest and most important applications. Integrigy Consulting offers security assessment services for leading ERP and CRM applications.

Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60681 USA
888/542-4802
www.integrigy.com