

Satisfying SOX Compliance Requirements with Database Auditing

(A Case Study at Silicon Image)

White Paper

March 1st, 2005

[Revised on February 13, 2008]

Written by

Sunita Sarathy of Absolute Technologies

Edited by

Cameron Lerner of Absolute Technologies &

Kenny Gilbert of Silicon Image



Absolute Technologies[®]

Leverage Your Investment in Oracle Applications™!

888.270.3012

www.absolute-tech.com

Table of Contents:

Audit and Compliance - Introduction 3

Auditing Options..... 4

 Oracle Database Auditing..... 4

 E-Business Suite Auditing 7

 Row Who Columns: 7

 End User Signon Access 9

 System Administrator Audit Trail..... 11

 Oracle Alerts 13

 Absolute Technologies – Applications Auditor..... 15

Company Overview – Silicon Image 21

 Procedural Guidelines..... 21

 Oracle Database Auditing..... 21

 E-Business Suite Auditing..... 21

 Absolute Technologies Application Auditor..... 22

Case Studies..... 25

Conclusion: 26

References:..... 27

Audit and Compliance - Introduction

Sarbanes-Oxley legislation is placing new and significant compliance and audit trail demands on Finance and IT departments across the nation. Auditing application transactions in the database is one of the many requirements saddled on these departments.

To comply, IT departments must design and build mechanisms in the database to track and report changes to key data elements in the database. While this task is not difficult for most staffers, it does require significant time and effort that goes above and beyond their normal duties, possibly impacting their ability to provide adequate support to their users. Additionally, such software customizations are often difficult or tedious to document and maintain, and increase the risk of issues relating to the application processes which are the source of the auditing.

Purpose

The purpose of this white paper is to evaluate available mechanisms that can identify *high risk or unauthorized* transactions within the company's Oracle database and associated applications.

These mechanisms have the objective of ensuring that Segregation of Duties (SOD) and Risk Control procedures and enforcement have been properly established with respect to database activity across both end users and business analysts.

Segregation of Duties in Oracle Applications dictates that no one individual has control of the various aspects of a business process, from authorization to record keeping. Responsibilities are assigned to individuals in such a way as to encourage checks and balances within the system, and minimize the opportunity for unauthorized access or fraud. For example, users should not be able to change data that is not in their job description, and IT analysts should not be able to manipulate Application objects without leaving a trail. These procedures, once established, can be monitored through various forms of database auditing.

This paper documents a case study of the auditing process at Silicon Image (SI). The various options of auditing and their application have been discussed in some depth.

Scope

The Oracle database can be accessed and changed in the following ways:

➤ Oracle E-Business Suite:

Information can be accessed within the Oracle Applications via secure login and by navigating to forms using granted responsibilities. An analyst with System Administrator responsibility could easily give himself access to forms that allow him to manipulate sensitive financial information. Without password control, a user or analyst can also access and change column values through Oracle's Help->Tools->Examine menu path. Oracle Alert in the hands of an experienced user also provides access to alter data in tables within the APPS schema that would be

otherwise restricted if the user only had access to standard module responsibilities like Payables User or GL Super User.

➤ **Direct SQL access:**

A knowledgeable user or analyst can access and manipulate information through SQL using SQL*Plus, or other tools like TOAD, SQL*Navigator, etc. A user with SQL access to APPS cannot only issue database manipulation language (DML) commands like select, update or delete, but database definition language (DDL) commands as well, like create, drop or alter a schema object.

Once the procedures to prevent unauthorized access are in place, they need to be monitored, or audited on a regular basis. These procedures could include limiting access to APPS user, restricting the assignment of System Administrator or Alerts responsibilities, following strict change control processes for code changes, etc.

Auditing at Silicon Image

Silicon Image involved their auditors in discussions about information and procedures that needed to be audited within the database. Comments on audit reports were monitored for further reporting. At SI, various methods of data auditing as outlined in this paper were implemented, and the feedback is presented in this case study.

Auditing Options

The various options for auditing as discussed in this paper are:

- Oracle Database Auditing
- E-Business Suite R11i
 - Row Who Columns
 - End User Access
 - System Administrator Audit Trail
 - Oracle Alerts
- Absolute Technologies – Application Auditor

Oracle Database Auditing

Overview

Database auditing allows the tracking of virtually any type of database transaction, from a session login, to the creation or alteration of any schema object, to the execution of SQL statements. All privileges that can be granted to a user or role within a database can be audited.

Oracle 8 introduced system triggers, or triggers that fire when system events occur. System events include user logins and creating, altering or dropping schema objects. However, with regard to read-only access, or SELECT statements, traditional auditing

options can only track the “who”, or the identity of the user that accessed the table. Oracle 9i takes auditing a step further by introducing **Fine Grained Auditing (FGA)**, which allows the tracking of the “what”. In other words, auditors can now track exactly what data was selected from a particular table. As of Oracle 9i, FGA could only track “SELECT” statements. With the introduction of Oracle 10g, FGA can also handle DML statements – INSERT, UPDATE and DELETE, making it a more comprehensive auditing feature.

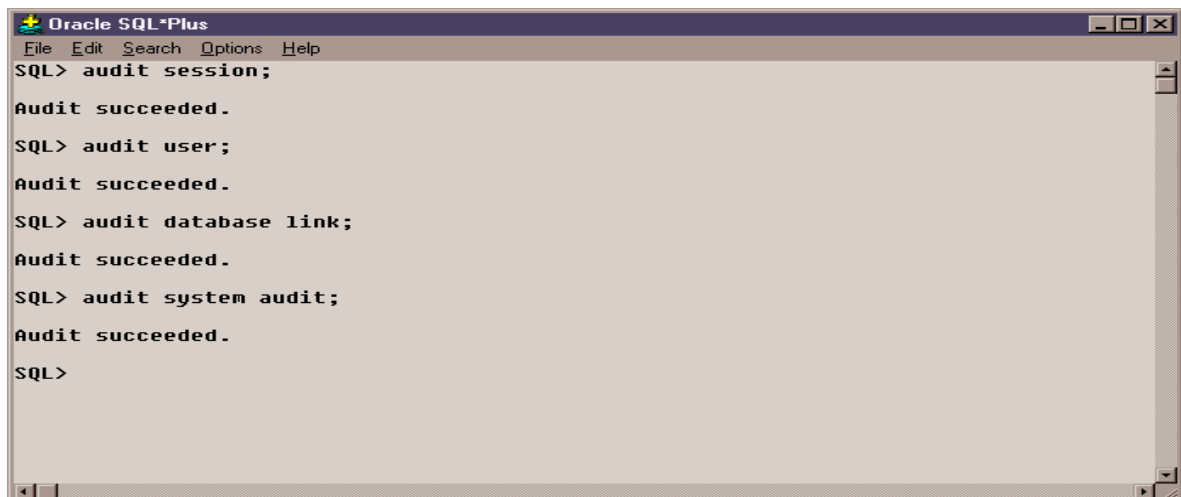
Configuration

There are two steps to configure auditing at the database level:

1. Set audit_trail parameter: Open the init.ora file for the Oracle instance. Set the audit_trail parameter to ‘True’ or ‘DB’. This will enable auditing to the SYS.AUD\$ table, which resides in the data dictionary and is otherwise known as the *Database Audit Trail*. Restart the database.

The table SYS.AUD\$ can be queried by any user. This table needs to be purged on a regular basis to prevent crowding of the SYSTEM tablespace, and is the only SYS owned table from which Oracle allows rows to be deleted.

2. Log on as the SYSTEM user in SQL*Plus and execute SQL AUDIT commands as required. Some examples of auditing commands are:



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> audit session;
Audit succeeded.
SQL> audit user;
Audit succeeded.
SQL> audit database link;
Audit succeeded.
SQL> audit system audit;
Audit succeeded.
SQL>
```

- Audit session:
The “audit session” command audits the users who log in to the database, their location, and the time that they log in. It can be useful in pinpointing attacks on a user’s login through multiple password attempts, or logins monitoring database access outside of normal business hours.
- Audit user:
Audits create, alter and drop user commands.

Satisfying SOX Compliance Requirements with Database Auditing

- Audit database link:
Audits create or drop database link user commands.
- Audit public database link:
Audits create or drop public database link user commands.
- Audit system audit:
Audits audit and noaudit user commands.

Benefits

Oracle allows auditing options to be set at three levels:

- Statement auditing:
Audits specific SQL statements or groups of statements that affect a particular type of database object. For example, the "AUDIT TABLE" command audits "SELECT", "INSERT", "UPDATE" or "DELETE" statements on a table.
- Privilege Auditing:
Audits SQL statements that are authorized by system privileges, such as "create table" or "drop table". The "AUDIT CREATE TABLE" command audits statements using the "CREATE TABLE" system privilege.
- Object Auditing:
Audits statements on specific objects, such as triggers, tables or packages, such as the "EMP" table.

Limitations:

- Database auditing does not provide before and after values for column features. There is no visibility to the original data.
- Audits are at the SQL statement level and do not reflect the resulting impact to individual records and columns within a table.
- There is no standard reporting or access to data from a form.
- Data in the SYS.AUD\$ table is not user-friendly.
- Event based alerts or user notifications cannot be created, as the table is owned by SYS. (No triggers can be defined on SYS tables)

```

Oracle SQL*Plus
File Edit Search Options Help
SQL> desc SYS.AUD$
Name                                                    Null?    Type
-----
SESSIONID                                               NOT NULL NUMBER
ENTRYID                                                 NOT NULL NUMBER
STATEMENT                                               NOT NULL NUMBER
TIMESTAMP#                                              NOT NULL DATE
USERID                                                  VARCHAR2(30)
USERHOST                                               VARCHAR2(128)
TERMINAL                                               VARCHAR2(255)
ACTION#                                                 NOT NULL NUMBER
RETURNCODE                                             NOT NULL NUMBER
OBJ$CREATOR                                             VARCHAR2(30)
OBJ$NAME                                               VARCHAR2(128)
AUTH$PRIVILEGES                                       VARCHAR2(16)
AUTH$GRANTEE                                          VARCHAR2(30)
NEW$OWNER                                             VARCHAR2(30)
NEW$NAME                                              VARCHAR2(128)
SES$ACTIONS                                           VARCHAR2(19)
SES$TID                                               NUMBER
LOGOFF$LREAD                                           NUMBER
LOGOFF$PREAD                                           NUMBER
LOGOFF$LWRITE                                          NUMBER
LOGOFF$DEAD                                           NUMBER
LOGOFF$TIME                                           DATE
COMMENT$TEXT                                          VARCHAR2(4000)
SPARE1                                                VARCHAR2(255)
SPARE2                                                NUMBER
OBJ$LABEL                                             RAW(255)
SES$LABEL                                             RAW(255)
PRIV$USED                                             NUMBER
SQL>

```

E-Business Suite Auditing

E-Business Suite R11i) provides the following options for auditing:

Row Who Columns:

Overview:

The “Row Who” Column information can be easily accessed through the application. Click on Help>Record History to view history for the current record. Ensure that the cursor is in the correct block. This information can also be accessed through SQL by joining to the appropriate tables.

The following information is provided.

- 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 the row was last updated

Satisfying SOX Compliance Requirements with Database Auditing

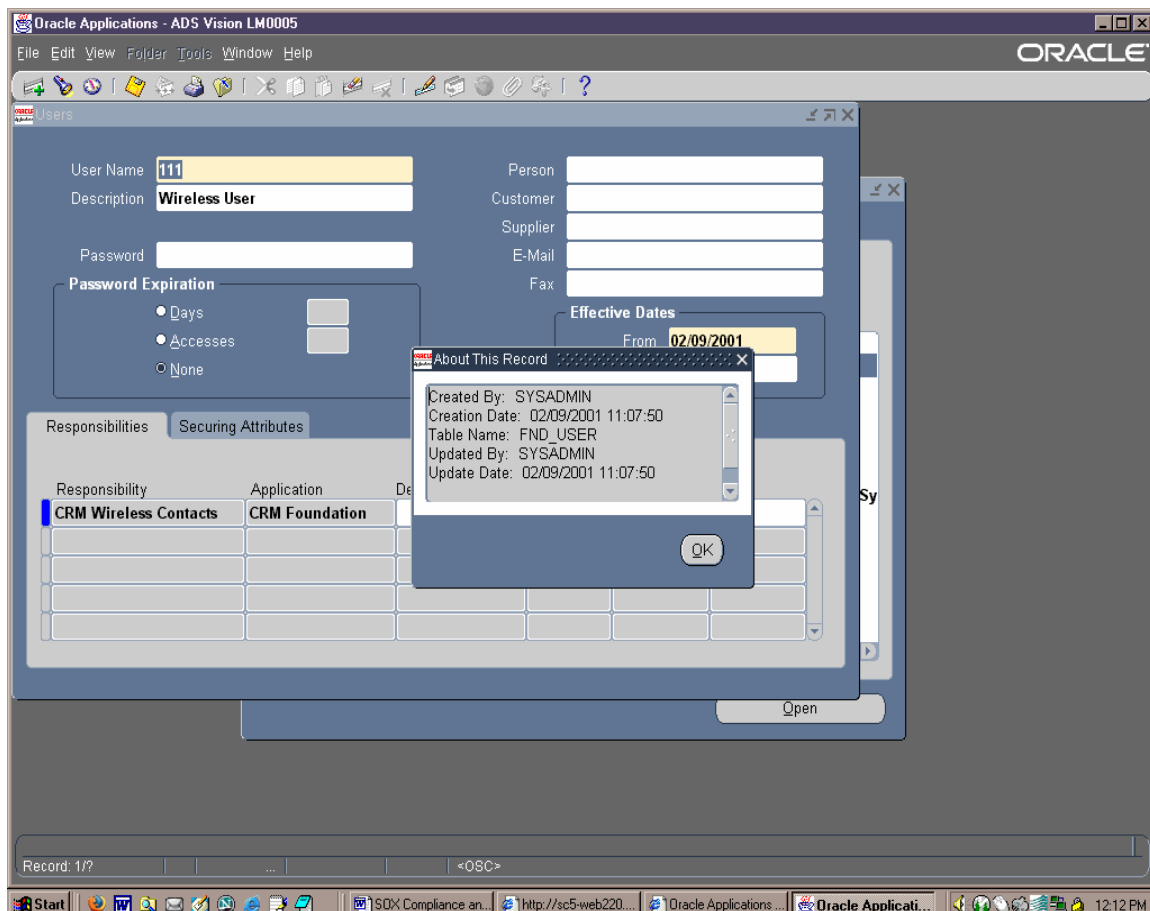
- Last Updated By – Oracle Applications User ID from FND_USERS

Benefits:

This is probably the quickest and easiest audit feature in Oracle Applications. As a first line of defense, it reveals a partial history of the record in question, the identity of the user that created it, and that of the last user to modify it. This information is sometimes sufficient to investigate the history behind a change.

Limitations:

- Only the initial creation and last updated date and user are stored.
- A complete audit trail of changed values is not stored, thus the date or user of any updates occurring between the creation and last update of the record are lost.
- No details regarding the value of columns updated are stored. There is no visibility to before and after changes to column values.
- This information is not updated if the database record is inserted, updated or deleted by a user or process that is external to the security of Oracle Applications. A user with SQL access to tables can change records without impacting the FND_USER and FND_LOGIN tables.



End User Signon Access

Overview:

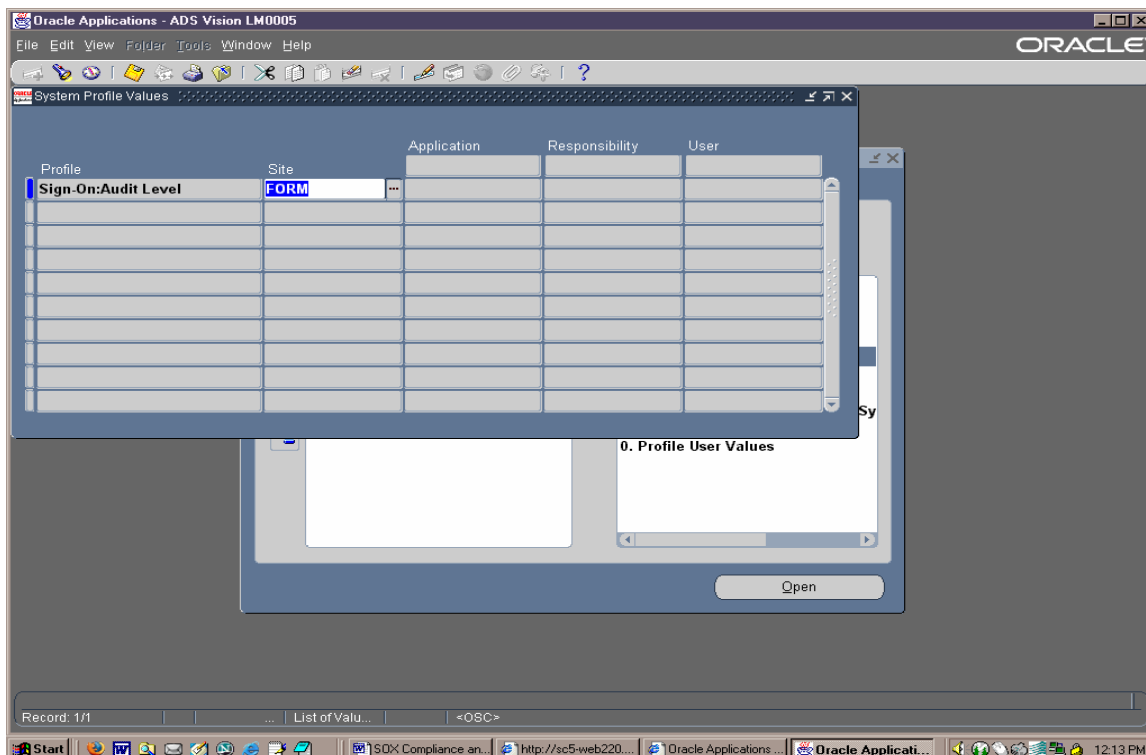
Oracle Applications stores information regarding all end user access information, like user logins, unsuccessful password attempts, responsibility selection, Form usage and concurrent process execution.

The following are standard reports for end-user auditing:

- SignOn Audit User
- SignOn Audit Responsibilities
- SignOn Audit Forms
- SignOn Audit Concurrent Requests
- SignOn Audit Unsuccessful Logins

Configuration:

- The level of End User Access auditing is controlled by the system profile “option” Sign-On: Audit Level”. The valid settings are None, User, Responsibility and Form. This profile option should always be set to ‘Form’ to enable the most auditing.



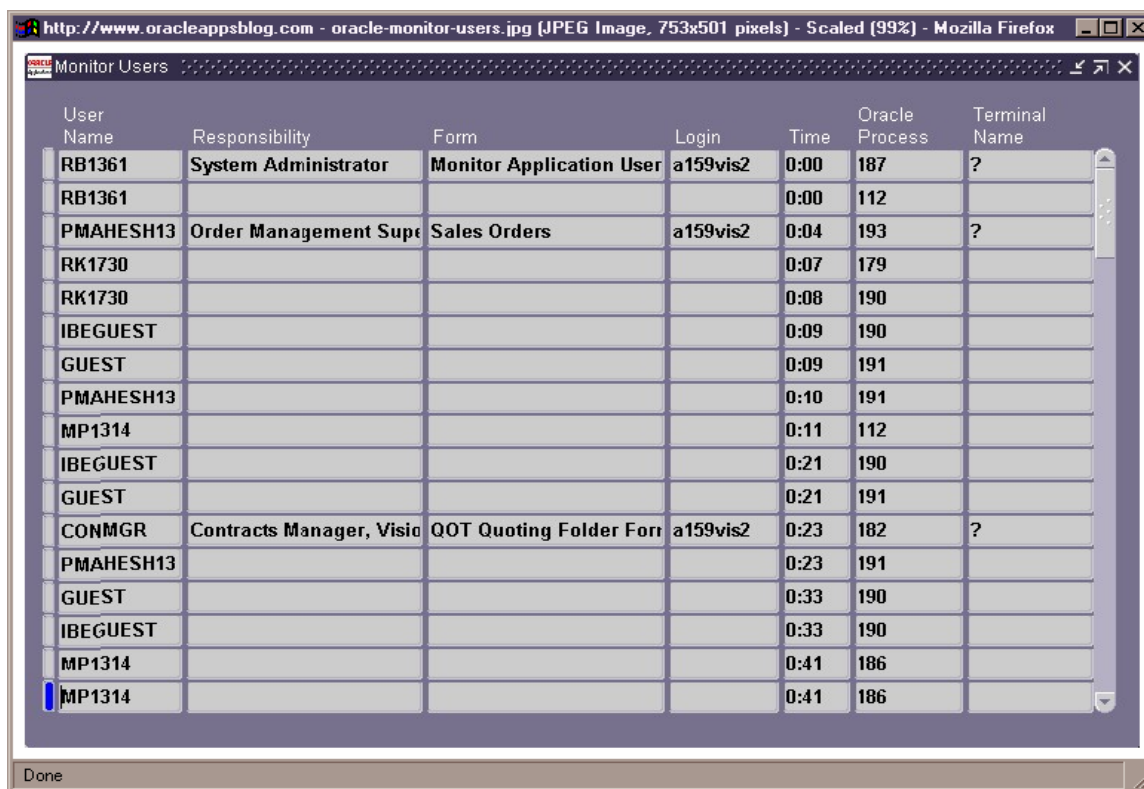
Satisfying SOX Compliance Requirements with Database Auditing

- All user sign-ons, responsibility selections, and form accesses will be logged to the tables APPLSYS.FND_LOGINS, APPLSYS.FND_LOGIN_RESPONSIBILITIES, and APPLSYS.FND_LOGIN_RESP_FORMS, respectively.
- Unsuccessful logins are automatically recorded in the table APPLSYS.FND_UNSUCCESSFUL_LOGINS.
- Concurrent requests are recorded in the FND_CONCURRENT_REQUESTS table.

Benefits:

In addition to the reports listed above, user activity can also be monitored online. This is done through the “Monitor Users Form” in the System Administrator responsibility. The screen is user-friendly and comprehensive, and conveys login information at a glance.

The Navigation path for this form is System Administrator->Security->User->Monitor



The screenshot shows the Oracle Monitor Users form in a Mozilla Firefox browser window. The form displays a table with the following columns: User Name, Responsibility, Form, Login, Time, Oracle Process, and Terminal Name. The table contains 18 rows of data, including users like RB1361, PMAHESH13, RK1730, IBEGUEST, GUEST, MP1314, and CONMGR.

User Name	Responsibility	Form	Login	Time	Oracle Process	Terminal Name
RB1361	System Administrator	Monitor Application User	a159vis2	0:00	187	?
RB1361				0:00	112	
PMAHESH13	Order Management Super	Sales Orders	a159vis2	0:04	193	?
RK1730				0:07	179	
RK1730				0:08	190	
IBEGUEST				0:09	190	
GUEST				0:09	191	
PMAHESH13				0:10	191	
MP1314				0:11	112	
IBEGUEST				0:21	190	
GUEST				0:21	191	
CONMGR	Contracts Manager, Visio	QOT Quoting Folder Form	a159vis2	0:23	182	?
PMAHESH13				0:23	191	
GUEST				0:33	190	
IBEGUEST				0:33	190	
MP1314				0:41	186	
MP1314				0:41	186	

Limitations:

- This method of audit only audits end user usage of specified forms.
- It does not audit changes at the database transaction level.
- It does not audit any specific form activity or database transactions that may be of interest to ensure compliance, just identifies usage or signon.

- User logon information may not be complete if the user session terminates unexpectedly (PC crash, etc).
- Failure to purge the end user access data may lead to performance issues. The concurrent program *Purge Signon Audit Data* is used to purge sign on information.

System Administrator Audit Trail

Overview:

E-Business Suite features its own auditing mechanism, which can satisfy normal auditing requirements. This is done through the System Administrator *Audit Trail feature*, which can be configured to maintain a full history of changes made at the table and column level. The Audit Trail concurrent program creates database triggers on the audited table and writes change records to a shadow table based on the selected columns to audit. It also creates several views based on the shadow table for reporting purposes. Note, however, that Oracle indicates that this method of auditing should not be used for transactional data as it can lead to significant database slowdown and performance issues. Low volume, setup tables, like the following AOL tables are good candidates for using audit trail: FND_DATA_GROUPS, FND_MENU, FND_REQUEST_GROUPS, FND_CONCURRENT_PROGRAMS, etc.

Configuration:

- Set system profile option "Audit Trail: Activate to Yes".
- As System administrator, select Security->Audit Trail->Install
- Check schemas for which auditing should be enabled.
- Select Security->Audit Trail->Groups.
- Define applications, groups, tables and columns to audit.
- Run the program *Audit Trail Update Tables* to activate auditing.

The following will be created for each table:

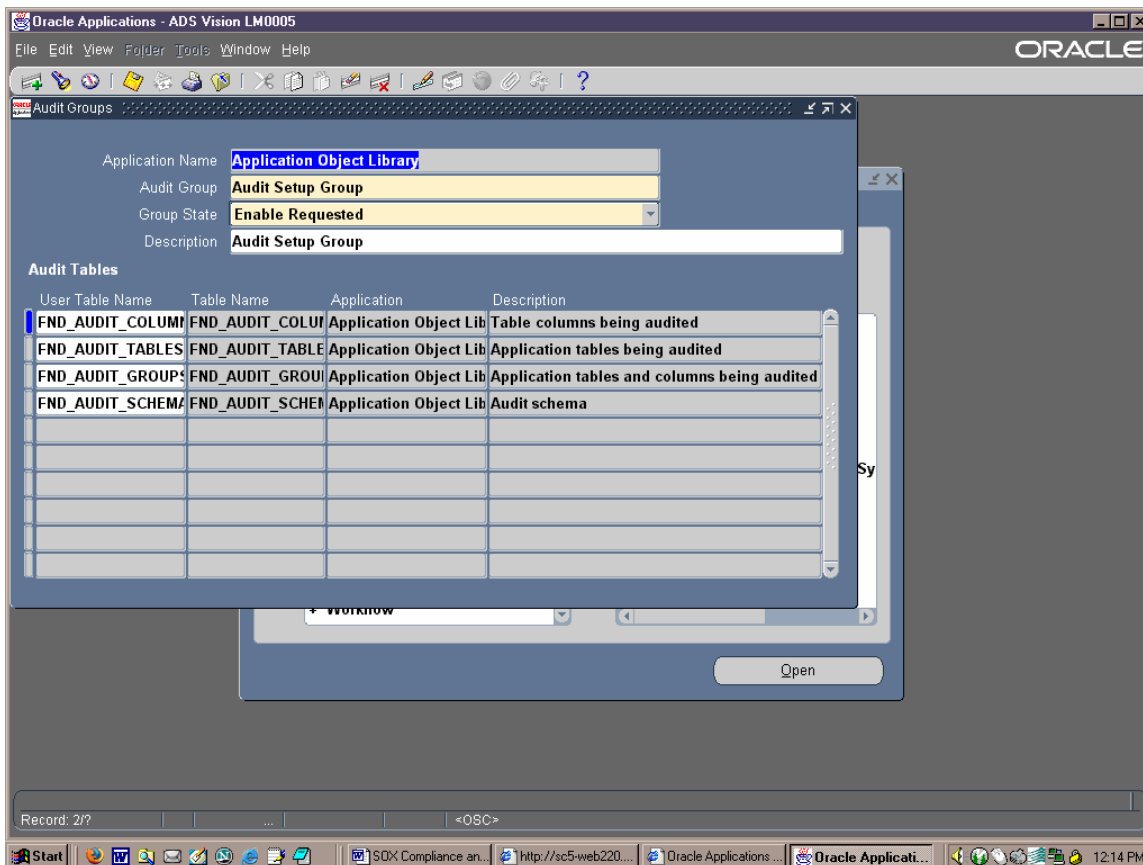
- Shadow table (Original table name appended with _A). One row is created in the Shadow Table for each audited transaction against the original table.
- Trigger on each audited column in the original table
- Views: One view containing all audited columns and one view for each column audited.

Benefits:

- Audit Groups allow you to enable auditing for groups rather than individual tables. This allows you to organize audits belonging to the same business process, e.g., Purchase Order tables.
- Set up and activation is accomplished through the familiar, secure and standard Oracle Applications user interface.

Limitations:

- There is no *single* audit table for ease of reporting.
- Conditions cannot be applied to the triggers.
- The program *Audit Trail Update Tables* has no parameters – the group’s “state” must be modified, then the program can be run.
- Auditing cannot be toggled on/off for a single table, but is performed at the group level.
- Data outside the scope of the audit table cannot be captured, like foreign table column values for ease of reporting.
- There is no revision control mechanism.
- There is no mechanism to support migration across database instances.
- There is no standard reporting.
- The resulting shadow tables and views are not user friendly.
- No single record holds the before and after details of changed column values.
- There is no error handling visibility to the end user.
- Auditing database row changes can be performance intensive.



Oracle Alerts

Overview:

Alerts are Oracle's exception-reporting tool, they can send email notification of pre-defined high-risk transactions. This reduces reaction time to unauthorized or suspicious activity, and helps the analyst get information on database exceptions as they happen. Alerts also allow application data to be shared outside the application.

Alerts can be set up to fire periodically, or based upon an Event (creates a database trigger). The exception conditions can be specified along with the frequency, and actions can be specified according to a recipient's response.

Configuration:

- Navigate to the Alert Manager Responsibility->Alert->Define.
- Enter an application, Name and Description for the Alert.
- Select the type (Periodic, Event).
- Enter the number of days required to keep history.
- Enter the SQL statement to specify the exception condition.
- Write and verify a SQL statement.
- Define Actions that the Alert should run.

Benefits:

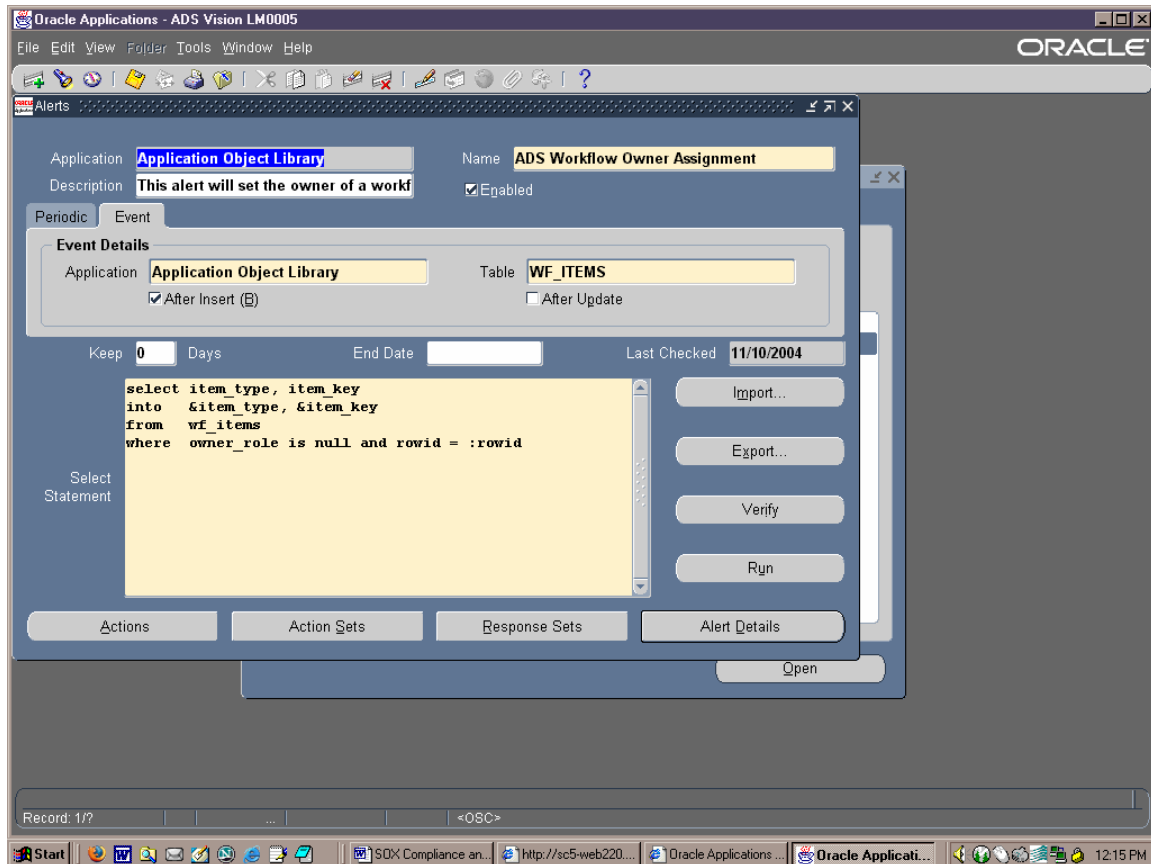
- The flexibility of ALERTS allows a database administrator the ability to monitor activities from tablespace sizing to activities associated with particular applications.
- When the exception condition is triggered, Alerts can email a message, solicit a response, and perform actions based on the response received.
- Alerts can work with custom applications as well, as long as they are properly registered within Oracle Applications.

Limitations:

- Periodic alerts only capture a snapshot of data.
- Alerts cannot provide Before and After values of changed columns.
- Too many Alerts can clog up the concurrent manager.
- Alerts do not provide ad-hoc reporting, only notification.
- On Event Alerts fire upon any change to a record within a defined table, capturing unwanted transactions and impacting system performance. They do not allow individual column changes to be tracked.
- Alerts do not provide "On Delete" capabilities, only "On Insert" and Update

Satisfying SOX Compliance Requirements with Database Auditing

- Alerts do not utilize stored procedures in the PL/SQL body of the triggers they create, thus their SQL is not maintained into the shared pool of parsed statements and must be parsed for each execution, a possible performance issue.



Absolute Technologies – Applications Auditor

Overview:

The **Application Auditor** enables end users to report on changes to selected columns of tables in the database and writes them to one or many protected audit table(s). Tracking changes of selected columns will ensure that proper change management is being followed, and will provide before and after column values, along with key data points from other tables as well as session details like E-Business Suite Username and Responsibility, resulting from any tracked column to provide maximum visibility to the auditor

Application Auditor (AA) is installed in its own schema for security within the same database instance as E-Business Suite. The AA Administrator module may be configured to prevent and audit any tampering of AA database objects by unauthorized users, even DBAs, to secure AA's audit mechanisms and trail. AA can be flexibly configured to detect data changes upon record insert, update or delete and can define additional rules and filters to manage the scope or content of the resulting audit trail, and minimize the correlating load it may place on the system. [Although our customers attest that the system performance impact of auditing even high volume table is insignificant.]

Silicon Image implemented Absolute Technologies' Application Auditor within their E-Business Suite production database. Database auditing information is captured in AA's secured audit history table. Any attempt to turn off or drop Application Auditor objects triggers the creation of an audit trail record and an instant email notification to the Director of IS.

Benefits:

- Eliminates custom, heterogeneous column tracking mechanisms by providing a single, flexible audit mechanism for any table/column in the database.
- Change tracking to generate an audit trail within Application Auditor can be configured and activated within minutes.
- Configuration Reports provide a record of all Audit setup details.
- A single audit table stores change details across all tables for flexible and efficient reporting:
 - Before and After values of column
 - Table and Column name
 - Trigger Action (Insert, Update or Delete)
 - Session Details – Application User Name, Network Address, OS User, Terminal, etc.
 - Primary Key of Table
 - When and Who changed the column value
 - Reference additional column values within the same table at time of change
 - Embedded SQL can select additional values from other tables upon change
- Shares the same database instance as Oracle Applications, thus eliminating the need to maintain another server or database instance.

Satisfying SOX Compliance Requirements with Database Auditing

- Is not E-Business Suite specific: can audit tables in any Oracle database in the customer's environment.
- Uses standard Oracle Development tools and programming language.
- Includes revision control architecture that enables the modification of configurations while retaining prior setups.
- Migration utility enables the migration of configuration objects between development, test and production systems.
- Features an Administrator module that secures the audit mechanisms and audit trail and *"Audits the Auditor"*. Changes made to objects within the AA schema may be prevented and/or captured and stored in a secure schema to prevent the AA user from disabling or modifying audits without the knowledge of others.
- Web Browser Based Screens provide visibility to:
 - Audit Configurations
 - Compiled Audit Objects
 - Compilation and Object Errors
 - Audit Transactions
 - Migration Utility
 - Revision Control
- Database Objects (DDL) auditing by schema and operation.
- Database Connection Auditing
- E-Business Suite Extensions
 - Over 90 Pre-seeded table audit configurations for E-Business Suite.
 - Application User Watch Lists.
 - Segregation of Duties Conflict Manager.
 - Over 1200 seeded Function Conflict Pairs available
 - Baseline Violation Report
 - Continuous Violation Detection, Prevention and Notification
 - Application User Exemptions
 - False Positive Free Logic

Satisfying SOX Compliance Requirements with Database Auditing

Info Manage Audits View Revisions Extensions Lookups Action Query Block Record Edit Field Window Help

Aa - Define Audit Configurations. (DB=VISDEMO) (User=AI225) (Version=2.26.)

Register Objects Source Columns Default Elements Conditions Map Elements

Source and Destination Header

Compile Status: VALID Frozen:

Source

Object Name: AP_CHECKS_ALL

Alias: ACA

DB Link & Host:

Destination

AI_CE_CHANGE_TRX

ACCT

Select Source Columns to Capture

Source Column	On Change Action
ADDRESS_LINE1	Execute
ADDRESS_LINE2	Execute
ADDRESS_LINE3	Execute
ADDRESS_LINE4	Execute
AMOUNT	Execute
BANK_ACCOUNT_ID	Execute
CITY	Execute
COUNTRY	Execute
CURRENCY_CODE	Execute
STATE	Execute
VENDOR_NAME	Execute
VENDOR_SITE_ID	Execute
BASE_AMOUNT	Reference
CHECK_ID	Reference
CHECK_NUMBER	Reference
CREATED_BY	Reference
CREATION_DATE	Reference
LAST_UPDATED_BY	Reference
LAST_UPDATE_DATE	Reference
VENDOR_ID	Reference

Current Revision

Install

ACTION

A Few Details:

- In the AA Audit Configuration screen, table columns can be audited in two ways, as Reference Items or as Execute Items:
 - A Reference column is one whose value is captured at the time an AA trigger fires on its table. It is designed to provide a reference value on the audit record. This reference can be either the old value or the new value or both.
 - An Execute column will cause the AA trigger to fire if an insert, update or delete is made to the value of this column. By default it tracks the old and new values of the column in the audit table.

Satisfying SOX Compliance Requirements with Database Auditing

- In the Default Elements tab of the Configuration screen, you can define SQL statements that select a single column value from a foreign key table reference to be mapped to the audit table, thus capturing reference data beyond the scope of the triggered table to create a more meaningful and reportable audit record.

Info Manage Audits View Revisions Extensions Lookups Action Query Block Record Edit Field Window Help

Aa - Define Audit Configurations. (DB=VISDEMO) (User=AI225) (Version=2.26.)

Register Objects | Source Columns | **Default Elements** | Conditions | Map Elements

Source and Destination Header

Compile Status: Frozen:

Source: Object Name: Alias: DB Link & Host:

Destination: DB Link & Host:

Current Revision: ACTION

Define Source Default Elements to Generate upon Execute Column Change

Name	Default Type	Default Value	SQL Name	SQL Meaning
D_HOST	Session Details	Host		
D_IP_ADDRESS	Session Details	IP Address		
D_IS_DBA	Session Details	Is DBA?		
D_MODULE	Session Details	Module		
D_NETWORK_PROTOCOL	Session Details	Network Protocol		
D_OS_USER	Session Details	OS User		
D_PREVENTED_FLAG	Trigger Info	Prevented Flag		
D_PRIMARY_COLUMN_NAME	Constant	CHECK ID		
D_PROGRAM	Session Details	Program		
D_PROXY_USER	Session Details	Proxy User		
D_REVISION_NAME	Trigger Info	Revision Name		
D_SESSION_USER	Session Details	Session User		
D_TABLE_ALIAS	Trigger Info	Source Table Alias		
D_TABLE_NAME	Trigger Info	Table Name		
D_TABLE_OWNER	Trigger Info	Table Owner Name		
D_TERMINAL	Trigger Info	OS Terminal		
D_TRANSACTED_DATE	Sysdate	Sysdate		
D_TRIGGER_ACTION	Trigger Info	Action		
D_TRX_ID	Sequence	AI CE CHANGE TRX S		
D_VENDOR_NAME	SQL	select vendor name	VENDOR_NAME	Vendor Name for AP and

Satisfying SOX Compliance Requirements with Database Auditing

- Audits may be viewed, compiled and monitored using the Compile Audits and View Error Screen.

Compile Audit Configurations

DB Link Host Name

Revision Source Alias

View/Disable/Enable Compiled Audit Objects

Enabled								Associated Audit Configuration	
Select	Name	Type	Owner	Last Compiled	Status	PLSQL	Errors	Changed Since Compiled	Enabled
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AP_SPA_P0	PROCEDURE	AI225	22-OCT-2007 15:42:19	INVALID	View	View	22-OCT-2007 15:42:09	<input checked="" type="checkbox"/>
<input type="checkbox"/>	AI_CE_CHANGE_AP_SPA_T0	TRIGGER	AI225	22-OCT-2007 15:42:22	INVALID	View	View	22-OCT-2007 15:42:09	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AA2_P0	PROCEDURE	AI225	18-OCT-2007 10:52:31	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AA2_T0	TRIGGER	AI225	18-OCT-2007 10:52:32	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AAA_P0	PROCEDURE	AI225	18-OCT-2007 10:50:54	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AAA_T0	TRIGGER	AI225	18-OCT-2007 10:50:57	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AAP_P0	PROCEDURE	AI225	18-OCT-2007 10:52:27	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AAP_T0	TRIGGER	AI225	18-OCT-2007 10:52:28	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AA_P0	PROCEDURE	AI225	18-OCT-2007 10:50:51	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_AA_T0	TRIGGER	AI225	18-OCT-2007 10:50:53	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_ABAA_P0	PROCEDURE	AI225	18-OCT-2007 10:51:55	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_ABAA_T0	TRIGGER	AI225	18-OCT-2007 10:51:57	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_ABAUA_P0	PROCEDURE	AI225	18-OCT-2007 10:51:26	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_ABAUA_T0	TRIGGER	AI225	18-OCT-2007 10:51:27	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	AI_CE_CHANGE_ACA_P0	PROCEDURE	AI225	18-OCT-2007 10:51:00	VALID	View	View	18-OCT-2007 09:33:20	<input checked="" type="checkbox"/>

Select All

Compiled Objects 154 Enabled Configs 90 Frozen Configs 77

Satisfying SOX Compliance Requirements with Database Auditing

- All audit trail records may be logged to one or more audit tables that are under the control of the secure AA schema and cannot be updated by anyone else. Here's a sample audit trail record:

The screenshot displays a window titled "View All Column Values for Record" with a menu bar (Info, Manage Audits, View, Revisions, Extensions, Action, Query, Block, Record, Edit, Field, Window, Help). The main content area shows the following audit trail record:

```
TRX_ID: 823
TABLE_NAME: FINANCIALS_SYSTEM_PARAMS_ALL
TABLE_ALIAS: FSPA
COLUMN_NAME: MATCH_OPTION
OLD_COLUMN_VALUE: R
NEW_COLUMN_VALUE: P
PRIMARY_COLUMN_NAME: SET_OF_BOOKS_ID
PRIMARY_COLUMN_KEY: 1
TRIGGER_ACTION: UPDATE
TERMINAL:
IP_ADDRESS: 71.146.207.25
OS_USER: appldemo
AUTHENTICATION_TYPE: DATABASE
SESSION_USER: APPS
PROXY_USER:
FND_USER_NAME: OPERATIONS
FND_RESP_NAME: Payables, Vision Operations (USA)
COMMIT_ID: 7.10.611389
REVISION_NAME: Install
EMAIL:
CURRENT_USER: YYAI
IS_DBA: FALSE
DB_DOMAIN:
DB_NAME: VISDEMO
HOST: demo.absoluteauditor.com
NETWORK_PROTOCOL: tcp
DB_USER_NAME: APPS
DB_USER_ID: 47
TRANSACTION_DATE: 07-JUN-2006 16:06:06
TABLE_ATTRIBUTE1: Vision Operations (USA)
```

Annotations on the screenshot include:

- Blue circle:** Surrounds the column name and old/new values. Text: "Before and after values for the column changed."
- Red circle:** Surrounds the terminal and OS user fields. Text: "No Terminal, along with FND User Name and Resp Name indicate the change was made by a user with a secure EBS login."
- Orange circle:** Surrounds the IS_DBA field. Text: "Is DBA indicates whether the user logged on as SYS. Useful for auditing unauthorized backend changes. OS USER also handy for identifying a backend user. Will provide the Windows network user if logged on remotely via SQL*Plus or Toad."
- Green circle:** Surrounds the transaction date and table attribute fields. Text: "User definable contextual attributes help identify the audit record in user friendly terms."

At the bottom of the window are buttons for "OK", "Cancel", and "Search". The status bar at the very bottom shows "Record: 1/?" and "<OSC>".

Company Overview – Silicon Image

Silicon Image (SI) develops and markets semiconductor solutions for the secure transmission and storage of digital media. The company was founded in January 1995, and went public in October 1999. The company's annual revenue for the year 2004 was \$173 million. The company has about 300 employees.

SI first went live on Oracle 11.5.8 in May 2003. The company upgraded to 11.5.9 in July 2004. Their auditors are the firm of Price Waterhouse Coopers (PWC) with SOX audit guidance from the firm of Horn, Murdoch & Cole.

Procedural Guidelines

Access to APPS schema is highly restricted and not available to analysts. APPS SQL access is withheld from developers to prevent back-end SQL manipulation of Oracle tables. The approach to auditing detailed below allows SI management immediate visibility to violations of these guidelines.

Oracle Database Auditing

SI uses standard database auditing (SQL AUDIT commands) to monitor database activities. These include changes to database accounts, which are relatively rare but may indicate suspicious activity or unauthorized access. DB auditing is also used to investigate creation of or changes to database links. At SI, links to the production database are very tightly controlled and restricted.

Changes to system audit commands/parameters are tracked closely. If someone turns off audit triggers or changes them, an Oracle Alert email is sent to the Director of FP&A and the Director of IS.

Also, a Periodic Alert has been defined on SYS.AUD\$ to notify the audit manager if any issues are raised.

E-Business Suite Auditing

E-Business Suite – Row Who Columns

SI uses "Row Who" Record History tracking for casual, low risk level auditing, in order to find the created by or last updated by identity of a user for a particular record. They do not rely on Row Who details to ensure SOD compliance.

E-Business Suite – End User Access

The standard Sign On Reports provided by Oracle Applications are used for end-user sign on and usage auditing. The chief use is to identify and track failed login attempts, as these can indicate either typing errors during login, or a more serious database hacking attempt. Users and the responsibilities granted to them are tracked using the base FND signon tables via Discoverer Reports.

The profile Option “Sign-On: Audit Level” is set to “Form”, to enable the most auditing.

E-Business Suite – Audit Trail

The System Administrator Audit Trail feature is not used at SI. They found it to be cumbersome to configure and maintain, and encountered too many issues during configuration and testing. There is no single audit table to facilitate ease of reporting, and the data in the audit table and views is not easy to follow or extract meaningful data. Other showstoppers were the inability to capture foreign key table/column details in the audit tables to facilitate reporting, and no way to conditionally restrict the creation of an audit record.

Due to these and other reasons, SI decided to abandon this method after testing its usefulness to the company.

E-Business Suite – Oracle Alerts

Oracle Alerts were used, among other things, to audit changes to high-risk tables. Event alerts would fire when specified conditions on the above tables were met, generating an audit notification..

However, this approach was scrapped in favor of Absolute’s Application Auditor because Alerts provided no easily reportable audit trail, just an email notification. Furthermore, the Alert triggers were found to be less than optimal, Alert was granted to too many analysts to satisfy security concerns, and it required too much effort to maintain.

Absolute Technologies Application Auditor

Application Auditor (AA) was installed in SI as a response to their SOX SOD audit requirements. The tool has greatly simplified the audit process at SI, due to all the audit records being housed in a single and secure audit table. This also makes reporting flexible and efficient.

Complete implementation and setup of all the tables was achieved in under a week.

At SI, AA audits several key setup tables in order to track user logins, menus and request groups assigned to users, and other significant activities. Here are a few examples:

Satisfying SOX Compliance Requirements with Database Auditing

Table Name	Table Description	Comments
FND_USER	Application User Accounts	Tracking Oracle Applications users' start and end date and password lifecycle.
FND_RESPONSIBILITY	Main Responsibility definition	Tracking menu and request group assigned
FND_EXECUTABLES	What executes behind a Reports/ request	Track new reports or processes setup in Oracle
FND_ENABLED_PLSQL	PLSQL procedures	Tracking PL*SQL installed or changed
FND_PRODUCT_INSTALLATIONS	Modules installed	Tracking new Modules Installed
AD_APPLIED_PATCHES	Patches Applied	
FND_MENU_ENTRIES	Menus Assignments	Tracking functions, grants and sub menus assigned to menu's
FND_PROFILE_OPTION_VALUES	Profile Options	Unauthorized changes to profile options
FND_USER_RESP_GROUPS	Responsibilities assigned to users	Tracking all adds and changes to responsibilities assigned to Application users

- Tables that have a financial impact if updated by an unauthorized user or database account are also audited, with additional logic configured within the tool to track the following two conditions:
 - Track changes that have been made to a table without going through a registered Oracle Applications form. These would be changes executed from the backend using SQL. In all situations, direct SQL access to change data requires a change control form to be completed before the change is applied. This audit identifies possible cases where prior authorization was not granted.
 - Track all changes made to these tables by Business Analysts, DBA's and System Administrators who actually login and use the standard application forms to make an unauthorized change.

In both these cases the details of the change are written to the audit table, and an email is sent to the Directors of FP&A and IS instantly. This will then be reviewed to ensure that a change control ticket was logged.

At SI, AA audits many tables where unauthorized activity can have a financial impact. Here are a few examples:

Satisfying SOX Compliance Requirements with Database Auditing

Table Name	Comments
AP_INVOICES_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: AMT_DUE_EMPLOYEE, APPROVAL_STATUS, VENDOR_ID, APPROVED_AMOUNT
AP_CHECKS_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: ADDRESS_LINE1, ADDRESS_LINE2, AMOUNT, BASE_AMOUNT, CHECK_DATE, CHECK_NUMBER, CLEARED_AMOUNT, CLEARED_DATE, VENDOR_ID, BANK_ACCOUNT_ID, VOID_DATE, CHECK_ID
PO_HEADERS_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: VENDOR_ID
PO_VENDORS	Tracking all if updated from DB or id done by a IS employee Columns tracked are: VENDOR_NAME
RA_CUSTOMER_TRX_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: APPROVAL_CODE, BILL_TO_CUSTOMER_ID, PAYING_CUSTOMER_ID, SHIP_TO_CUSTOMER_ID
HZ_PARTIES	Tracking all if updated from DB or id done by a IS employee Columns tracked are: PARTY_NAME
FA_DISTRIBUTION_HISTORY	Tracking all if updated from DB or id done by a IS employee Columns tracked are: ASSIGNED_TO, UNITS_ASSIGNED
CE_STATEMENT_HEADERS_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: CONTROL_TOTAL_CR, CONTROL_TOTAL_DR, GL_DATE, STATEMENT_DATE
GL_JE_BATCHES	Tracking all if updated from DB or id done by a IS employee Columns tracked are: APPROVAL_STATUS_CODE
GL_JE_LINES	Tracking all if updated from DB or id done by a IS employee Columns tracked are: STATUS_CODE
WF_ROUTING_RULES	Tracking all if updated from DB or id done by a IS employee Columns tracked are: ROLE, END_DATE
AP_WEB_SIGNING_LIMITS_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: SIGNING_LIMIT
AP_EXPENSE_REPORT_HEADERS_ALL	Tracking all if updated from DB or id done by a IS employee Columns tracked are: AMT_DUE_EMPLOYEE, CORE_WF_STATUS_FLAG, EMPLOYEE_ID, SOURCE, TOTAL, WORKFLOW_APPROVED_FLAG

The Application Auditor software is stored in a custom schema and access to this schema is tightly controlled. There is no E-Business Suite access to the audit transactions the tool generates. Also, the custom software follows the standard application change control process for any updates or granting of access.

There are several cases at SI to date where AA auditing has identified Segregation of Duties or controls risk issues, and aided in the review and resolution thereof.

Case Studies

Case I: Identify unauthorized changes to signing authority

A user with Sysadmin responsibility assigned the CFO's signing authority to another employee, causing an audit record to be generated and inserted into the audit table. An email was also sent to the Director of IS and the Change control Help Desk.

The change was researched by the Director of IS and found to be genuine. An explanation was entered for the Help Desk and approval awaited from the Director of Finance.

Once approval was received, the audit record was updated through SQL*Plus to reflect the status of the issue.

This case highlights how any misappropriation of signing authority will automatically raise alarms, preventing misuse. Due to the audit measures in place, the change has to be approved by a higher authority before it can be put in place, in this case, the Director of Finance.

Case II: Identify unauthorized update to customer

An analyst changed customer records to update the territory column, and in one case mistakenly updated the name of a customer and saved the record. This triggered an audit record, causing an alarm to be sent to the Director of IS and Change control Help Desk.

Once the issue was researched and found to be a genuine mistake, the Director IS updated the Help Desk with an explanation. The issue then awaited the approval of the Director of Finance. As the audit table stored the old and new values of the changed record, it was a simple matter to restore the original customer name.

The situation demonstrates how unauthorized changes, whether unintended or malicious, need to raise an alarm immediately. Even though the change was found to be inadvertent, there is still a procedure to be followed before it can be corrected, and the issue is logged for auditing purposes. The auditing process needs to be rigorous enough to ensure that there is no loophole for such transactions can slip through.

Case III: Issue with Promise Date in GOP

A user reported that the Promise Date for a Sales Order was incorrect. In the Global Order Promising (GOP) module, sales orders are automatically scheduled. SI also has a program that auto updates Promise Dates.

After the user's complaint, the analyst reviewed the audit history of the sales order, which displayed the entire history of changes to the Order.

It was found that GOP had originally provided the correct Promise Date, but it had subsequently been modified several times by the user herself. The analyst was able to quickly and confidently review the changes with the user, mitigating the user's concern of the "system" incorrectly updating the dates.

Conclusion:

Auditing in Oracle is more critical now than ever with the advent of Sarbanes-Oxley legislation. Segregation of Duties and Enforcement of Controls policies designed to minimize risk of fraud must have systematic and continuous evidence of their application and effectiveness. Unauthorized changes to application data which directly or indirectly impact financial reporting, access controls or cause fraudulent transactions have to be audited by secure and reliable mechanisms, and audit trail records must be maintained as required by legislative or auditor guidance.

The measures implemented in the case study at Silicon Image are used to comprehensively satisfy IT audit requirements for the company, and enable them to successfully avoid material deficiencies by their external auditors.

References:

- Oracle 8i SQL Reference Manual
- Oracle 9i SQL Reference Manual
- Oracle Applications Release 11.5.9 Reference Manuals
- Oracle 9i Concepts Manual