

**Author – A.Kishore/Sachin**  
<http://appsdba.info>

## **AuditTrail in Oracle Applications**

AuditTrail is a way of keeping track of changes made to important data in Oracle Application tables. AuditTrail keeps a history of the following:

1. What changed?
2. Who changed it?
3. When did the change take place?

This can be done on a row-by-row basis or on individual columns of a table.

### **© Steps to Setup AuditTrail**

If we want to setup auditTrail, then we need to follow the below steps.

**Step 1.** Logon to the Applications as the sysadmin user and select the System Administrator responsibility.

**Step 2.** Make sure APPLSYS has select privileges on SYS.DBA\_TABLES.

For example, if you are auditing the PA\_PROJECTS\_ALL table, use the following SQL statement to verify this:

```
select TABLE_NAME from SYS.DBA_TABLES  
where TABLE_NAME like 'PA_PROJECTS_ALL';
```

**Step 3.** Now we need to define auditgroups: It is required to group the tables that we need to audit since auditing is enabled on an audit groups basis.

**Navigation: Security -> AuditTrail -> Groups**

**Step 4.** Now define audit installations: In order to be able to audit across application installations, you must specify the Oracle ID's whose tables you wish to audit. At this point, you will need to specify the list of columns from the table you want audited.

**Navigation: Security -> AuditTrail -> Install**

**Step 5.** Run the "AuditTrail Update Tables" report from the submit request form to enable auditing.

**Navigation: Requests -> Run -> select a single request**

**Author – A.Kishore/Sachin**  
<http://appsdba.info>

© **What does actually the "AuditTrail Update Tables" report do internally?**

- Creates shadow tables with a name of the first 26 characters of the audit table and a "\_A" suffix. The columns will have the same data types as those of the audited table. The program also creates views \_AC# and \_AV# on the shadow table to facilitate the access of data. The "#" in the view name represents a number, and can be any number depending on how many views are needed to access the data. The \_AC# view contains the current state of the data at the time of query from the join of the shadow table and the table audited. The \_AV# view contains the same data as the shadow table plus the current value of the rows. Basically, the \_AC# view contains the current status of the audited table with the unchanged columns filled-in with the unchanged values. The shadow table has special columns in addition to the columns we specified to be audited:

AUDIT\_TIMESTAMP(DATE): Keeps track of date(HH:MI:SS) and time when auditing was done.

AUDIT\_TRANSACTION\_TYPE (VARCHAR2(1)): This column has the information of what type of transaction was conducted on the table, I for "INSERT", U for "UPDATE", D for "DELETE" or C for "CURRENT".

AUDIT\_USER\_NAME (VARCHAR2(100)): The Oracle Application UserID if changes are made from the forms or Oracle ID if changes are made from SQL\*PLUS.

AUDIT\_TRUE\_NULLS (VARCHAR2(250)): Delimited list of column names that were changed from NULL actually.

AUDIT\_SESSION\_ID(NUMBER): ID for the session.

AUDIT\_COMMIT\_ID (NUMBER): ID for the COMMIT.

PRIMARY KEY: The combined columns primary key value for the table audited. This is not a special column but the actual value of the primary key for the table.

**Author – A.Kishore/Sachin**  
<http://appsdba.info>

- Creates the after event, transaction level database triggers on the tables in your audit group. Database triggers can be created based on timing events, such as "BEFORE" or "AFTER" a DATA MANUPULATION LANGUAGE (DML) activity on the table. In other words, it could be an "INSERT", "UPDATE" or "DELETE" activity on the table that could be the triggering event that causes the creation of a corresponding trigger. Triggers are created either at a "ROW" or "TRANSACTION" level, raising the total number of triggers that can be created on a table to 12 (3\*2\*2). It is this after event transaction type triggers that get created by the "AuditTrail Update Table" report during this process. The names of these triggers start with the first 26 characters of the name of the table being audited, plus an \_AI, \_AU or \_AD, which stand for after insert, after update or after delete triggers respectively. These triggers call the procedures \_AIP, \_AUP, or \_AID which are also created by this program. These procedures save the old data rows from the audited table into the shadow table.