**Auditing in Oracle**

Auditing is a default feature of the Oracle server. The initialization parameters that influence its behaviour can be displayed using the SHOW PARAMETER SQL\*Plus command.

SQL> SHOW PARAMETER AUDIT

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

audit\_file\_dest string C:\ORACLE\PRODUCT\10.2.0\ADMIN

\DB10G\ADUMP

audit\_sys\_operations boolean FALSE

audit\_trail string NONE

SQL>

Auditing is disabled by default, but can enabled by setting the AUDIT\_TRAIL static parameter, which has the following allowed values.

AUDIT\_TRAIL = { none | os | db | db,extended | xml | xml,extended }

The following list provides a description of each setting:

* none or false - Auditing is disabled.
* db or true - Auditing is enabled, with all audit records stored in the database audit trial (SYS.AUD$).
* db,extended - As db, but the SQL\_BIND and SQL\_TEXT columns are also populated.
* xml- Auditing is enabled, with all audit records stored as XML format OS files.
* xml,extended - As xml, but the SQL\_BIND and SQL\_TEXT columns are also populated.
* os- Auditing is enabled, with all audit records directed to the operating system's audit trail.

Note. In Oracle 10g Release 1, db\_extended was used in place of db,extended. The XML options are new to Oracle 10g Release 2.

The AUDIT\_SYS\_OPERATIONS static parameter enables or disables the auditing of operations issued by users connecting with SYSDBA or SYSOPER privileges, including the SYS user. All audit records are written to the OS audit trail.

The AUDIT\_FILE\_DEST parameter specifies the OS directory used for the audit trail when the os, xml and xml,extended options are used. It is also the location for all mandatory auditing specified by theAUDIT\_SYS\_OPERATIONS parameter.

To enable auditing and direct audit records to the database audit trail, we would do the following.

SQL> ALTER SYSTEM SET audit\_trail=db SCOPE=SPFILE;

System altered.

SQL> SHUTDOWN

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> STARTUP

ORACLE instance started.

Total System Global Area 289406976 bytes

Fixed Size 1248600 bytes

Variable Size 71303848 bytes

Database Buffers 213909504 bytes

Redo Buffers 2945024 bytes

Database mounted.

Database opened.

SQL>

**Audit Options**

One look at the [AUDIT](http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4007.htm) command syntax should give you an idea of how flexible Oracle auditing is. There is no point repeating all this information, so instead we will look at a simple example.

First we create a new user called AUDIT\_TEST.

CONNECT sys/password AS SYSDBA

CREATE USER audit\_test IDENTIFIED BY password

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA UNLIMITED ON users;

GRANT connect TO audit\_test;

GRANT create table, create procedure TO audit\_test;

Next we audit all operations by the AUDIT\_TEST user.

CONNECT sys/password AS SYSDBA

AUDIT ALL BY audit\_test BY ACCESS;

AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY audit\_test BY ACCESS;

AUDIT EXECUTE PROCEDURE BY audit\_test BY ACCESS;

These options audit all DDL and DML, along with some system events.

* DDL (CREATE, ALTER & DROP of objects)
* DML (INSERT UPDATE, DELETE, SELECT, EXECUTE).
* SYSTEM EVENTS (LOGON, LOGOFF etc.)

Next, we perform some operations that will be audited.

CONN audit\_test/password

CREATE TABLE test\_tab (

id NUMBER

);

INSERT INTO test\_tab (id) VALUES (1);

UPDATE test\_tab SET id = id;

SELECT \* FROM test\_tab;

DELETE FROM test\_tab;

DROP TABLE test\_tab;

In the next section we will look at how we view the contents of the audit trail.

**View Audit Trail**

The audit trail is stored in the SYS.AUD$ table. Its contents can be viewed directly or via the following views.

SELECT view\_name

FROM dba\_views

WHERE view\_name LIKE 'DBA%AUDIT%'

ORDER BY view\_name;

VIEW\_NAME

------------------------------

DBA\_AUDIT\_EXISTS

DBA\_AUDIT\_OBJECT

DBA\_AUDIT\_POLICIES

DBA\_AUDIT\_POLICY\_COLUMNS

DBA\_AUDIT\_SESSION

DBA\_AUDIT\_STATEMENT

DBA\_AUDIT\_TRAIL

DBA\_COMMON\_AUDIT\_TRAIL

DBA\_FGA\_AUDIT\_TRAIL

DBA\_OBJ\_AUDIT\_OPTS

DBA\_PRIV\_AUDIT\_OPTS

DBA\_REPAUDIT\_ATTRIBUTE

DBA\_REPAUDIT\_COLUMN

DBA\_STMT\_AUDIT\_OPTS

14 rows selected.

SQL>

The three main views are shown below.

* DBA\_AUDIT\_TRAIL - Standard auditing only (from AUD$).
* DBA\_FGA\_AUDIT\_TRAIL - Fine-grained auditing only (from FGA\_LOG$).
* DBA\_COMMON\_AUDIT\_TRAIL - Both standard and fine-grained auditing.

The most basic view of the database audit trail is provided by the DBA\_AUDIT\_TRAIL view, which contains a wide variety of information. The following query displays the some of the information from the database audit trail.

COLUMN username FORMAT A10

COLUMN owner FORMAT A10

COLUMN obj\_name FORMAT A10

COLUMN extended\_timestamp FORMAT A35

SELECT username,

extended\_timestamp,

owner,

obj\_name,

action\_name

FROM dba\_audit\_trail

WHERE owner = 'AUDIT\_TEST'

ORDER BY timestamp;

USERNAME EXTENDED\_TIMESTAMP OWNER OBJ\_NAME ACTION\_NAME

---------- ----------------------------------- ---------- ---------- ----------------------------

AUDIT\_TEST 16-FEB-2006 14:16:55.435000 +00:00 AUDIT\_TEST TEST\_TAB CREATE TABLE

AUDIT\_TEST 16-FEB-2006 14:16:55.514000 +00:00 AUDIT\_TEST TEST\_TAB INSERT

AUDIT\_TEST 16-FEB-2006 14:16:55.545000 +00:00 AUDIT\_TEST TEST\_TAB UPDATE

AUDIT\_TEST 16-FEB-2006 14:16:55.592000 +00:00 AUDIT\_TEST TEST\_TAB SELECT

AUDIT\_TEST 16-FEB-2006 14:16:55.670000 +00:00 AUDIT\_TEST TEST\_TAB DELETE

AUDIT\_TEST 16-FEB-2006 14:17:00.045000 +00:00 AUDIT\_TEST TEST\_TAB DROP TABLE

6 rows selected.

SQL>

When the audit trail is directed to an XML format OS file, it can be read using a text editor or via the V$XML\_AUDIT\_TRAIL view, which contains similar information to the DBA\_AUDIT\_TRAIL view.

COLUMN db\_user FORMAT A10

COLUMN object\_schema FORMAT A10

COLUMN object\_name FORMAT A10

COLUMN extended\_timestamp FORMAT A35

SELECT db\_user,

extended\_timestamp,

object\_schema,

object\_name,

action

FROM v$xml\_audit\_trail

WHERE object\_schema = 'AUDIT\_TEST'

ORDER BY extended\_timestamp;

DB\_USER EXTENDED\_TIMESTAMP OBJECT\_SCH OBJECT\_NAM ACTION

---------- ----------------------------------- ---------- ---------- ----------

AUDIT\_TEST 16-FEB-2006 14:14:33.417000 +00:00 AUDIT\_TEST TEST\_TAB 1

AUDIT\_TEST 16-FEB-2006 14:14:33.464000 +00:00 AUDIT\_TEST TEST\_TAB 2

AUDIT\_TEST 16-FEB-2006 14:14:33.511000 +00:00 AUDIT\_TEST TEST\_TAB 6

AUDIT\_TEST 16-FEB-2006 14:14:33.542000 +00:00 AUDIT\_TEST TEST\_TAB 3

AUDIT\_TEST 16-FEB-2006 14:14:33.605000 +00:00 AUDIT\_TEST TEST\_TAB 7

AUDIT\_TEST 16-FEB-2006 14:14:34.917000 +00:00 AUDIT\_TEST TEST\_TAB 12

6 rows selected.

SQL>

Several fields were added to both the standard and fine-grained audit trails in Oracle 10g, including the following.

* EXTENDED\_TIMESTAMP - A more precise value than the exising TIMESTAMP column.
* PROXY\_SESSIONID - Proxy session serial number when an enterprise user is logging in via the proxy method.
* GLOBAL\_UID - Global Universal Identifier for an enterprise user.
* INSTANCE\_NUMBER - The INSTANCE\_NUMBER value from the actioning instance.
* OS\_PROCESS - Operating system process id for the oracle process.
* TRANSACTIONID - Transaction identifier for the audited transaction. This column can be used to join to the XID column on the FLASHBACK\_TRANSACTION\_QUERY view.
* SCN - System change number of the query. This column can be used in flashback queries.
* SQL\_BIND - The values of any bind variables if any.
* SQL\_TEXT - The SQL statement that initiated the audit action.

The SQL\_BIND and SQL\_TEXT columns are only populated when the AUDIT\_TRAIL parameter is set to db,extended or xml,extended.

**Maintenance and Security**

Auditing should be planned carefully to control the quantity of audit information. Only audit specific operations or objects of interest. Over time you can refine the level of auditing to match your requirements.

The database audit trail must be deleted, or archived, on a regular basis to prevent the SYS.AUD$ table growing to an unnacceptable size.Only DBAs should have maintenance access to the audit trail. Auditing modifications of the data in the audit trail itself can be achieved using the following statement.

AUDIT INSERT, UPDATE, DELETE ON sys.aud$ BY ACCESS;

The OS and XML audit trails are managed through the OS. These files should be secured at the OS level by assigning the correct file permissions.

**Fine Grained Auditing (FGA)**

Fine grained auditing extends Oracle standard auditing capabilities by allowing the user to audit actions based on user-defined predicates. It is independant of the AUDIT\_TRAIL parameter setting and all audit records are stored in the FGA\_LOG$ table, rather than the AUD$ table. The following example illustrates how fine grained auditing is used.

First, create a test table.

CONN audit\_test/password

CREATE TABLE emp (

empno NUMBER(4) NOT NULL,

ename VARCHAR2(10),

job VARCHAR2(9),

mgr NUMBER(4),

hiredate DATE,

sal NUMBER(7,2),

comm NUMBER(7,2),

deptno NUMBER(2)

);

INSERT INTO emp (empno, ename, sal) VALUES (9999, 'Tim', 1);

INSERT INTO emp (empno, ename, sal) VALUES (9999, 'Larry', 50001);

COMMIT;

The following policy audits any queries of salaries greater than £50,000.

CONN sys/password AS sysdba

BEGIN

DBMS\_FGA.add\_policy(

object\_schema => 'AUDIT\_TEST',

object\_name => 'EMP',

policy\_name => 'SALARY\_CHK\_AUDIT',

audit\_condition => 'SAL > 50000',

audit\_column => 'SAL');

END;

/

Querying both employees proves the auditing policy works as expected.

CONN audit\_test/password

SELECT sal FROM emp WHERE ename = 'Tim';

SELECT sal FROM emp WHERE ename = 'Larry';

CONN sys/password AS SYSDBA

SELECT sql\_text

FROM dba\_fga\_audit\_trail;

SQL\_TEXT

------------------------------------------

SELECT sal FROM emp WHERE ename = 'Larry'

1 row selected.

SQL>

Extra processing can be associated with an FGA event by defining a database procedure and associating this to the audit event. The following example assumes the FIRE\_CLERK procedure has been defined.

BEGIN

DBMS\_FGA.add\_policy(

object\_schema => 'AUDIT\_TEST',

object\_name => 'EMP',

policy\_name => 'SALARY\_CHK\_AUDIT',

audit\_condition => 'SAL > 50000',

audit\_column => 'SAL',

handler\_schema => 'AUDIT\_TEST',

handler\_module => 'FIRE\_CLERK',

enable => TRUE);

END;

/

The DBMS\_FGA package contains the following procedures.

* ADD\_POLICY
* DROP\_POLICY
* ENABLE\_POLICY
* DISABLE\_POLICY

In Oracle9i fine grained auditing was limited queries, but in Oracle 10g it has been extended to include DML statements, as shown by the following example.

-- Clear down the audit trail.

CONN sys/password AS SYSDBA

TRUNCATE TABLE fga\_log$;

SELECT sql\_text FROM dba\_fga\_audit\_trail;

no rows selected.

-- Apply the policy to the SAL column of the EMP table.

BEGIN

DBMS\_FGA.add\_policy(

object\_schema => 'AUDIT\_TEST',

object\_name => 'EMP',

policy\_name => 'SAL\_AUDIT',

audit\_condition => NULL, -- Equivalent to TRUE

audit\_column => 'SAL',

statement\_types => 'SELECT,INSERT,UPDATE,DELETE');

END;

/

-- Test the auditing.

CONN audit\_test/password

SELECT \* FROM emp WHERE empno = 9998;

INSERT INTO emp (empno, ename, sal) VALUES (9998, 'Bill', 1);

UPDATE emp SET sal = 10 WHERE empno = 9998;

DELETE emp WHERE empno = 9998;

ROLLBACK;

-- Check the audit trail.

CONN sys/password AS SYSDBA

SELECT sql\_text FROM dba\_fga\_audit\_trail;

SQL\_TEXT

--------------------------------------

SELECT \* FROM emp WHERE empno = 9998

INSERT INTO emp (empno, ename, sal) VALUES (9998, 'Bill', 1)

UPDATE emp SET sal = 10 WHERE empno = 9998

DELETE emp WHERE empno = 9998

4 rows selected.

-- Drop the policy.

CONN sys/password AS SYSDBA

BEGIN

DBMS\_FGA.drop\_policy(

object\_schema => 'AUDIT\_TEST',

object\_name => 'EMP',

policy\_name => 'SAL\_AUDIT');

END;

/