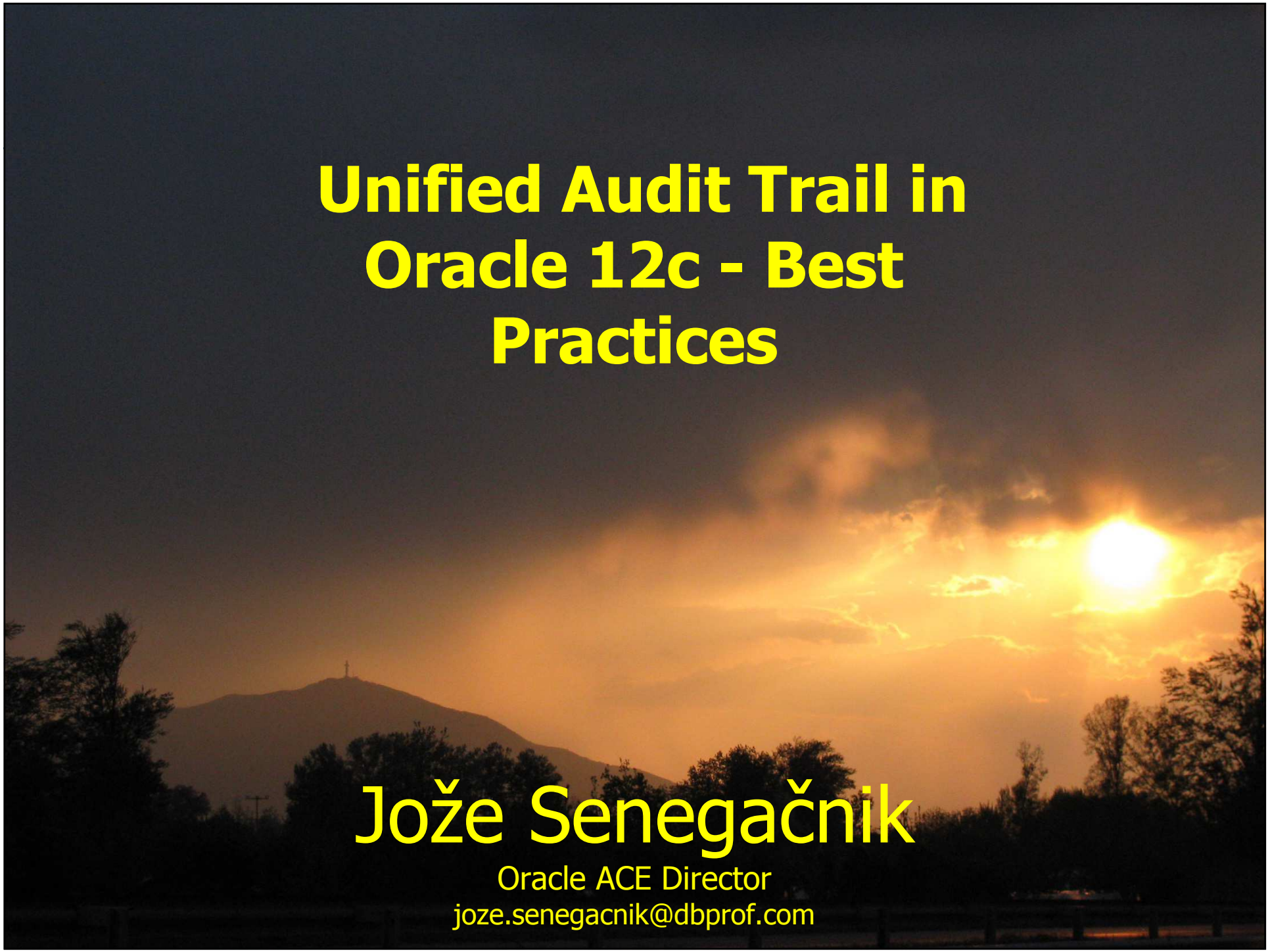

Unified Audit Trail in Oracle 12c - Best Practices
Jože Senegačnik
Oracle ACE Director
joze.senegacnik@dbprof.com

About the Speaker

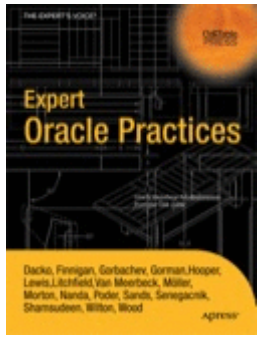
Jože Senegačnik

- First experience with Oracle Version 4 in 1988
- 27 years of experience with Oracle RDBMS.
- Proud member of the OakTable Network www.oaktable.net
- Oracle ACE Director
- Co-author of the OakTable book "Expert Oracle Practices" by Apress (Jan 2010)
- VP of Slovenian OUG (SIOUG) board
- CISA – Certified IS auditor
- Blog about Oracle: http://joze-senegacnik.blogspot.com

OakTable.net

ORACLE ACE Director

Oracle ACE



- PPL(A) – private pilot license PPL(A) / instrument rated IR/SE
- Blog about flying: http://jsenegacnik.blogspot.com
- Blog about Building Ovens, Baking and Cooking: http://senegacnik.blogspot.com

New Auditing Features in Oracle 12c

- Before Oracle 12c
  - Separated database audit trails in different locations/tables
- In 12c we have:
  - By default 12c is in MIXED MODE (DB and OS)
  - Some audit policies are already created by Oracle
    - ORA_SECURECONFIG (enabled by default when database is created)
    - ORA_ACCOUNT_MGMT
    - ORA_DATABASE_PARAMETER
    - …
- UNIFIED AUDIT TRAIL
  - Reduce the performance overhead associated with database auditing and enable more effective analysis of audit logs
  - Conditional Auditing
    - Support selective logging policies for specific events
    - Variables used for selective logging like SQL statements, actions, IP address, programs, time period, connection types, …
- Only one place to perform analysis of audit records
- Integration with Oracle Audit Vault and Database Firewall

Mixed Audit Trail Mode

- For newly created 12c databases, **mixed mode** auditing is enabled by default through the predefined Oracle policy ORA_SECURECONFIG.
- To start using unified auditing, you must enable at least one unified audit policy, and to stop using it one should disable all unified audit policies. More about that later on.
- Pure Unified Audit trail requires deliberate action – relinking oracle executable or copying DLL on Windows

Unified Audit Trail

- Captures audit information from:
  - Audit records (including SYS audit records) from unified audit policies and AUDIT settings
  - Fine-grained audit records from the DBMS_FGA PL/SQL package
  - Oracle Database Real Application Security audit records
  - Oracle Recovery Manager audit records
  - Oracle Database Vault audit records
  - Oracle Label Security audit records
  - Oracle Data Mining records
  - Oracle Data Pump
  - Oracle SQL*Loader Direct Load

Characteristics of Unified Audit Trail

- Stored in read-only table in the AUDSYS schema in the SYSAUX tablespace.
- In a Multitenant environment, each PDB, including the root, has its own unified audit trail.
- Who can access UAT:
  - SYS,
  - users with granted AUDIT_ADMIN and AUDIT_VIEWER roles
- Audit trail is written to database when it is opened read/write, otherwise (for instance during mount state) audit records are written to new format operating system files in the $ORACLE_BASE/audit/$ORACLE_SID directory.

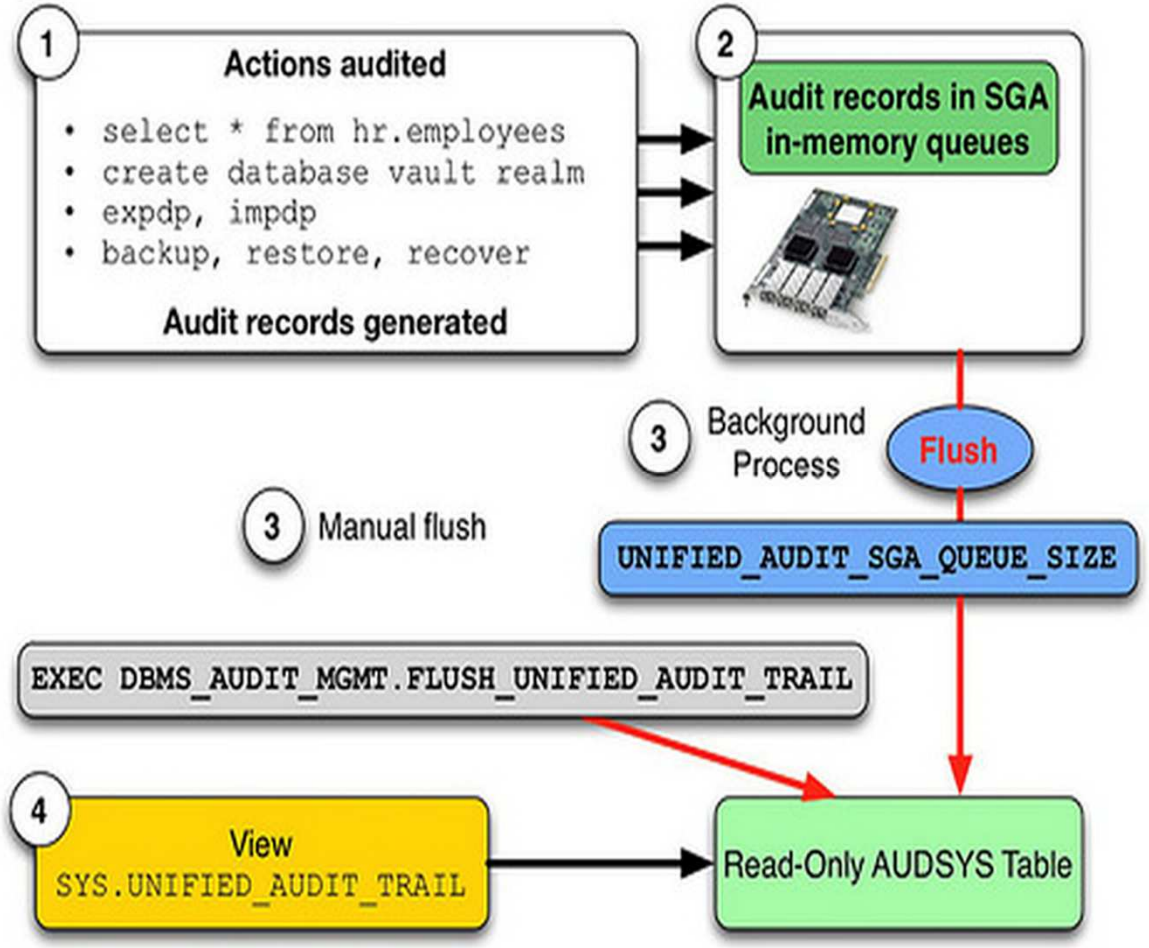
Characteristics of Unified Audit Trail

- Default mode for unified audit is „Queued Write mode".
- One can define conditions and exclusions into policies.
- SYS audit records appear with AUDIT_TYPE set to „Standard Audit".
- In mixed mode, you can use the new unified audit facility alongside the traditional auditing facility. In pure unified auditing, you can only use the unified audit facility.

Oracle Real Application Security (RAS)

- Is next generation Virtual Private Database (VPD)
- Provides a declarative model that enables security policies that encompass not only the business objects being protected but also the principals (users and roles) that have permissions to operate on those business objects.
- Benefits are:
  - End-user session propagation to the database
  - Data security based upon application users, role, privileges, and various relationships
  - Audit of end-user activity
  - Simplified administration with declarative security
- Auditing performed through application context's attributes, which are configured to be captured in the audit trail.
  - Application sessions encapsulate end user's security context.
  - Enable applications to use database authorization mechanisms for access control based on the end user identity.
  - Application sessions have these performance advantages over traditional database sessions

How Unified Audit Trail Works



Faster Performance

- Oracle 12c provides faster audit performance.
- DBA has control how the audit records are written to the audit trail:
  - Immediately („traditional")
    - For massively audited actions can see heavy contention
  - Queued to memory
    - Much faster than traditional one with much less performance impact

Activation of UAT

- Not enabled by default!!!
- Check by running:

```
select * from v$option where PARAMETER = 'Unified Auditing';
```

- On Unix/Linux one has to rebuild (relink) oracle executable
  - Stop all Oracle processes (database), listener and EM
  - Then relink Oracle executable:

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_on ioracle
```

  - Restart database, listener, EM
  - Set AUDIT_TRAIL=DB – should be set to DB

- On Windows

```
cd %ORACLE_HOME%\bin
copy orauniaud12.dll.dbl orauniaud12.dll
```

Auditing Policies

- „Old fashion" auditing required to define direct AUDIT/NOAUDIT statements
- In Oracle 12c one can create AUDITING POLICY and the use AUDIT/NOAUDIT statemnt to enable/disable auditing of that policy.
- Complex rules can be defined.

Creating Audit Policy

```
CREATE AUDIT POLICY policy_name
    { {privilege_audit_clause [action_audit_clause ] [role_audit_clause ]}
        | { action_audit_clause  [role_audit_clause ] }
        | { role_audit_clause }
     }
    [WHEN audit_condition EVALUATE PER {STATEMENT|SESSION|INSTANCE}]
    [CONTAINER = {CURRENT | ALL}];
```

- **privilege_audit_clause** : list of system privileges to be audited.
- **action_audit_clause** : list of actions to be audited. These can be either standard_actions, like UPDATE, or object-specific, like UPDATE ON schema.table. Here we can speciafy also component_actions for audting specific features like RMAN, data pump or SQL*Loader.
- **role_audit_clause** : list of roles to be audited. All system privileges granted via those roles are audited.
- **WHEN ... EVALUATE PER** : optional condition when the auditing should take place. This condition can be evaluated at STATEMENT, SESSION or INSTANCE level .
- **CONTAINER** : defines if policy is created for a PDB (CURRENT) or is common to all PDBs (ALL).

AUDIT_UNIFIED_POLICIES View

- AUDIT_UNIFIED_POLICIES displays the policies that are created in the database and their definition.
- It is populated only when unified auditing is enabled.

```
select POLICY_NAME, AUDIT_OPTION
from   AUDIT_UNIFIED_POLICIES
where  policy_name =  'ORA_SECURECONFIG'
order by 2 ;
```

```
POLICY_NAME          AUDIT_OPTION
-------------------- ----------------------------------------
ORA_SECURECONFIG     ADMINISTER KEY MANAGEMENT
ORA_SECURECONFIG     ALTER ANY PROCEDURE
ORA_SECURECONFIG     ALTER ANY SQL TRANSLATION PROFILE
ORA_SECURECONFIG     ALTER ANY TABLE
ORA_SECURECONFIG     ALTER DATABASE
ORA_SECURECONFIG     ALTER DATABASE LINK
ORA_SECURECONFIG     ALTER PROFILE
ORA_SECURECONFIG     ALTER ROLE
ORA_SECURECONFIG     ALTER SYSTEM
ORA_SECURECONFIG     ALTER USER
ORA_SECURECONFIG     AUDIT SYSTEM
ORA_SECURECONFIG     CREATE ANY JOB
ORA_SECURECONFIG     CREATE ANY LIBRARY
ORA_SECURECONFIG     CREATE ANY PROCEDURE
ORA_SECURECONFIG     CREATE ANY SQL TRANSLATION PROFILE
```

AUDIT_UNIFIED_ENABLED_POLICIES View

- **AUDIT_UNIFIED_ENABLED_POLICIES** view shows all audit policies that are enabled in the database.
- Column ENABLED_OPT: BY, EXCEPT, DISABLED

```
select USERNAME,POLICY_NAME, ENABLED_OPT,SUCCESS,FAILURE
from   AUDIT_UNIFIED_ENABLED_POLICIES
```

```
USER_NAME POLICY_NAME                ENABLED_ SUC FAI
--------- -------------------------- -------- --- ---
ALL USERS ORA_SECURECONFIG           BY       YES YES
```

Oracle Predefined Auditing Policies

ORA_LOGON_FAILURES Policy

```
CREATE AUDIT POLICY ORA_LOGON_FAILURES ACTIONS LOGON;
```

```
AUDIT POLICY ORA_LOGON_FAILURES WHENEVER NOT SUCCESSFUL;
```

- Enabled for failures only, a new audit policy can be created to audit all database logons.

ORA_SECURECONFIG Policy

```
CREATE AUDIT POLICY ORA_SECURECONFIG
 PRIVILEGES ALTER ANY TABLE, CREATE ANY TABLE, DROP ANY TABLE,
            CREATE ANY PROCEDURE, DROP ANY PROCEDURE, ALTER ANY PROCEDURE,
            GRANT ANY PRIVILEGE, GRANT ANY OBJECT PRIVILEGE, GRANT ANY ROLE,
            AUDIT SYSTEM, CREATE EXTERNAL JOB, CREATE ANY JOB,
            CREATE ANY LIBRARY,
            EXEMPT ACCESS POLICY,
            CREATE USER, DROP USER,
            ALTER DATABASE, ALTER SYSTEM,
            CREATE PUBLIC SYNONYM, DROP PUBLIC SYNONYM,
            CREATE SQL TRANSLATION PROFILE, CREATE ANY SQL TRANSLATION PROFILE,
            DROP ANY SQL TRANSLATION PROFILE, ALTER ANY SQL TRANSLATION PROFILE,
            TRANSLATE ANY SQL,
            EXEMPT REDACTION POLICY,
            PURGE DBA_RECYCLEBIN, LOGMINING,
            ADMINISTER KEY MANAGEMENT
 ACTIONS    ALTER USER, CREATE ROLE, ALTER ROLE, DROP ROLE,
            SET ROLE, CREATE PROFILE, ALTER PROFILE,
            DROP PROFILE, CREATE DATABASE LINK,
            ALTER DATABASE LINK, DROP DATABASE LINK,
            CREATE DIRECTORY, DROP DIRECTORY,
            CREATE PLUGGABLE DATABASE,
            DROP PLUGGABLE DATABASE,
            ALTER PLUGGABLE DATABASE,
            EXECUTE ON DBMS_RLS;
```

ORA_DATABASE_PARAMETER Policy

```
CREATE AUDIT POLICY ORA_DATABASE_PARAMETER
 ACTIONS ALTER DATABASE, ALTER SYSTEM, CREATE SPFILE;
```

ORA_ACCOUNT_MGMT Policy

```
CREATE AUDIT POLICY ORA_ACCOUNT_MGMT
 ACTIONS CREATE USER, ALTER USER, DROP USER, CREATE ROLE, DROP ROLE,
  ALTER ROLE, SET ROLE, GRANT, REVOKE;
```

Audit Administration

- Separation of duties introduced:
  - 2 new roles in 12c:
    - AUDIT_ADMIN for audit trail configuration and administration
    - AUDIT_VIEWER for viewing and analyzing audit data
- Under unified auditing, users are no longer able to create auditing policies against their own objects.

Related Database Packages

- DBMS_AUDIT_MGMT
  - Requires the user to have EXECUTE privilege over the DBMS_AUDIT_MGMT package.
    - Roles SYSDBA and AUDIT_ADMIN roles have EXECUTE privileges on the package by default.
  - Many constants are defined which should be used when one calls certain procedures within the package – see manual for details.
  - Only subset of procedures/functions can be used for unified audit trail because of UAT specifics.

Maintaing Audit Trail

- Audit trail is stored in the database (SYSAUX tablespace by default).
- The only possible way to maintain it is to use DBMS_AUDIT_MGMT package.
- Neither SYS can do any direct operations on audit trail table which is stored under separate AUDSYS schema.
- The size of audit trail is dependent on the amount of auditing defined and of course number of actions performed by users.
- The audit trail can grow quite fast and unfortunately some queries run by EM can take a substantial time.
- Bare in mind that in multitenant databases the audit trail is generated on root (CDB$ROOT) and PDB (pluggable database) level

Selecting from UAT in PDB

- CDB_UNIFIED_AUDIT_TRAIL
  - Bug when selecting
  - One should select from audit trail at CDB$ROOT and subsequently from all container databases (PDB)
- Regularly Archive audit trail:
  - Advance the archiving time stamp to last time stamp archived
  - Keep the adit trail relatively small in order to boost the preformance of queries

Regular Purging

- Keep in mind that like on OS level where you can run out of disk space with unified audit trail one can get the below error when there is no space in SYSAUX tablespace

```
ORA-02002: error while writing to audit trail
ORA-55917: Table flush I/O failed for log ID: 2 bucket ID: 0
ORA-01692: unable to extend lob segment
AUDSYS.SYS_LOB0000091859C00014$$ partition SYS_LOB_P2135 by 128 in
tablespace SYSAUX
ORA-02002: error while writing to audit trail
ORA-55917: Table flush I/O failed for log ID: 2 bucket ID: 0
ORA-01692: unable to extend lob segment
AUDSYS.SYS_LOB0000091859C00014$$ partition SYS_LOB_P2135 by 128 in
tablespace SYSAUX
```

Purging the Unified Audit Trail

Several option hoe to purge:
- Without limitation
- To last archived time stamp
- The last archive timestamp represents the timestamp of the most recent audit record that was securely archived.
- The CLEAN_AUDIT_TRAIL procedure is usually called after the SET_LAST_ARCHIVE_TIMESTAMP procedure has been used to set the last archived timestamp for the audit records.

```
select audit_trail, last_archive_ts FROM dba_audit_mgmt_last_arch_ts;

BEGIN
  DBMS_AUDIT_MGMT.clean_audit_trail(
   audit_trail_type        => DBMS_AUDIT_MGMT.audit_trail_unified,
   use_last_arch_timestamp => TRUE,
   container => DBMS_AUDIT_MGMT.CONTAINER_CURRENT
);
END;
/
```

Creating UAT Purging Job

- Use DBMS_AUDIT_MGMT.CREATE_PURGE_JOB to create a purge job for periodically deleting the audit trail records.

```
BEGIN
   DBMS_AUDIT_MGMT.CREATE_PURGE_JOB(
     audit_trail_type             =>  DBMS_AUDIT_MGMT.AUDIT_TRAIL_ALL,
     audit_trail_purge_interval   =>  24 /* hours */,
     audit_trail_purge_name       =>  'PURGE_UAT',
     use_last_arch_timestamp      =>  TRUE);
END;
```

Relocating Unified Audit Trail to Non-Default Location

- CREATE TABLESPACE **audit_ts** DATAFILE SIZE 1G AUTOEXTEND ON NEXT 100M MAXSIZE 10G;

```
BEGIN
  dbms_audit_mgmt.set_audit_trail_location(
     audit_trail_type => dbms_audit_mgmt.audit_trail_unified,
     audit_trail_location_value => 'AUDIT_TS');
END;
/
```

- For AUDIT_TRAIL_UNIFIED this procedure sets the tablespace for newer audit records in the unified audit trail but does not move the older audit records.

Unified Audit Trail Write Mode

```
-- Queued write mode (Default)
BEGIN
  DBMS_AUDIT_MGMT.set_audit_trail_property(
    audit_trail_type           => DBMS_AUDIT_MGMT.audit_trail_unified,
    audit_trail_property       => DBMS_AUDIT_MGMT.audit_trail_write_mode,
    audit_trail_property_value => DBMS_AUDIT_MGMT.audit_trail_queued_write
  );
END;
/

-- Immediate write mode
BEGIN
    DBMS_AUDIT_MGMT.set_audit_trail_property(
        audit_trail_type => DBMS_AUDIT_MGMT.audit_trail_unified,
        audit_trail_property => DBMS_AUDIT_MGMT.audit_trail_write_mode,
        audit_trail_property_value => DBMS_AUDIT_MGMT.audit_trail_immediate_write
);
END;
/
```

Queued write mode

- UNIFIED_AUDIT_SGA_QUEUE_SIZE parameter defines the size of the queue in the SGA.
  - Sizes can be different in RAC instances.
  - Size can be set from 1 to 32 MB, default 1MB.
- FLUSH_UNIFIED_AUDIT_TRAIL procedure can be used to flush audit trail when in queued write mode. Accepts two parameters:
  - FLUSH_TYPE - current instance or all instances in RAC
  - CONTAINER – current container all all containers

```
BEGIN
  DBMS_AUDIT_MGMT.flush_unified_audit_trail(
    flush_type => DBMS_AUDIT_MGMT.flush_current_instance,
    container  => DBMS_AUDIT_MGMT.container_all);
END;
/
```

Audit Trail Management Data Dictionary Views

- DBA_AUDIT_MGMT_CLEANUP_JOBS - currently configured audit trail purge jobs
- DBA_AUDIT_MGMT_CONFIG_PARAMS - currently configured audit trail properties that are used by the DBMS_AUDIT_MGMT PL/SQL package.
- DBA_AUDIT_MGMT_LAST_ARCH_TS - last archive timestamps set for audit trail purges used in DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL

Some Audit Trail Goes Still to Disk

- When database is not opened in a write mode.
- Procedure LOAD_UNIFIED_AUDIT_FILES - Loads the data from the spillover OS audit files in a unified audit trail into the designated unified audit trail tablespace (audit trail generated when the database is not in write mode)
- BUGS related output – auditing for background processes (needs applying patch)

Switching Off UAT

- Disable default audit policies (if enabled):
  - ORA_SECURECONFIG
  - ORA_LOGON_FAILURES
  - ...

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_off ioracle ORACLE_HOME=$ORACLE_HOME
```

- Windows systems:

```
Rename the %ORACLE_HOME%/bin/orauniaud12.dll file to
%ORACLE_HOME%/bin/orauniaud12.dll.dbl.
```

Auditing Policies – Best Practices

- Create one large audit policy with all necessary auditing for a session.
- More policies put bigger overhead on logon, greater consumption of memory in UGA.
- Use Queued Write mode to get optimal performance.
- Regularely use `DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES` to move audit trail from OS to UAT.
- Archive regularely audit trail records. Everything what has been archived should be purged from UAT as soon as possible. This will keep UAT performant for queries perfomed by EM.
- Create different database users in multitenant environment (for root and PDBs) and not one global user that have AUDIT_ADMIN or AUDIT_VIEW role (due to some bugs in current version when selecting from CDB_* views).

Thank you for your interest!

Q&A