# 32
# Administering the Audit Trail

Properly managing the audit trail on your databases ensures efficient performance and optimum use of the disk space. Users granted the `AUDIT_ADMIN` role can manage, archive, and purge audit trail.

- Managing the Unified Audit Trail
  Unified auditing is enabled by default, and audit trail management ensures audit configuration is efficient for your needs.
- Archiving the Audit Trail
  To maintain the integrity and reliability of audit data, keep only minimal required audit data locally in the database.
- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.
- Audit Trail Management Data Dictionary Views
  Oracle Database provides data dictionary views that list information about audit trail management settings.

## 32.1 Managing the Unified Audit Trail

Unified auditing is enabled by default, and audit trail management ensures audit configuration is efficient for your needs.

- How and Where Unified Audit Records Are Created
  Auditing is always enabled. Oracle Database generates audit records during or after the execution phase of the audited SQL statements.
- Sizing Recommendations for Unified Auditing
  Unified audit trail records require at least 50 percent more disk space than traditional audit records.
- How Audit Trail Records Are Written to the AUDSYS Schema
  Oracle Database automatically writes audit records to an internal relational table in the `AUDSYS` schema.
- Writing the Unified Audit Trail Records to SYSLOG or the Windows Event Viewer
  You can write the unified audit trail records to SYSLOG or the Windows Event Viewer by setting an initialization parameter.
- How Unified Audit Records are Written to the Operating System
  When the database cannot write audit trail records in the database itself, Oracle Database writes these records to operating system spillover audit files (`.bin` format).
- Moving Operating System Audit Records into the Unified Audit Trail
  Audit records that have been written to the spillover audit files can be moved to the unified audit trail database table.
- Improving the Performance of Queries and Purge Operations
  If the partition on which the `AUDSYS.AUD$UNIFIED` table is located is too large, then queries to and purges of the `UNIFIED_AUDIT_TRAIL` data dictionary view may take a long time to complete.

- Using Oracle Data Pump to Export and Import Unified Audit Trail Records
  You can include the unified audit trail in Oracle Database Pump export and import dump files.
- How Do Cursors Affect Auditing?
  For each execution of an auditable operation within a cursor, Oracle Database inserts one audit record into the audit trail.

**Related Topics**

- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.

## 32.1.1 How and Where Unified Audit Records Are Created

Auditing is always enabled. Oracle Database generates audit records during or after the execution phase of the audited SQL statements.

The unified audit records are written immediately to disk to an internal relational table in the `AUDSYS` schema. In the previous release, the unified audit records were written to SecureFile LOBs.The partitioned version of this table is based on the `EVENT_TIMESTAMP` timestamp as a partition key with a default partition interval of once a day. If the database version does not support partitioning, then the internal table is a regular, non-partitioned table.

> **Note:**
>
> If you had migrated to unified auditing in Oracle Database 12*c* release 1 (12.1), then you can manually transfer the unified audit records from the SecureFile LOBS to this internal table. If the version of the database that you are using supports partitioned tables, then this internal table is a partitioned table. In this case, you can modify the partition interval of the table by using the `DBMS_AUDIT_MGMT.ALTER_PARTITION_INTERVAL` procedure.

The generation and insertion of an audit trail record is independent of the user transaction being committed. That is, even if a user transaction is rolled back, the audit trail record remains committed.

Statement and privilege audit options from unified audit policies that are in effect at the time a database user connects to the database remain in effect for the duration of the session. When an unified audit policy is created and enabled, it will take effect immediately in the on-going session of the user on whom that policy is enabled without requiring that user to restart the database session. This holds true even when the unified audit policy gets disabled as well. However, any modifications (with respect to the statement audit option, privilege audit option, and audit conditions) to the existing unified audit policy definition using `ALTER AUDIT POLICY` statement will take effect in the subsequent sessions of the users on whom that policy is enabled.

In contrast, changes to schema object audit options become immediately effective for current sessions.

By default, audit trail records are written to the `AUDSYS` schema in the `SYSAUX` tablespace. Oracle recommends that you designate a different tablespace, including the one that is encrypted, by using the `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION` procedure.

**Example 32-1 Designate a different tablespace by using `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION`**

1. Create a dedicated auto segment space managed (ASSM) tablespace for unifited auditing:

```
CREATE TABLESPACE auto_seg_audit_tablespace DATAFILE 'DiskGroup_name' SIZE
1M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;
```

2. Designate the tablespace for unified auditing:

```
BEGIN
DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION(
    audit_trail_type            => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    audit_trail_location_value  =>  'auto_seg_audit_tablespace');
END;
```

**Related Topics**

- How Audit Trail Records Are Written to the AUDSYS Schema
  Oracle Database automatically writes audit records to an internal relational table in the `AUDSYS` schema.
- Activities That Are Mandatorily Audited
  Certain security sensitive database activities are always audited and such audit configurations cannot be disabled.
- *Oracle Database Upgrade Guide*
- `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION` in the *Oracle Database PL/SQL Packages and Types Reference*

## 32.1.2 Sizing Recommendations for Unified Auditing

Unified audit trail records require at least 50 percent more disk space than traditional audit records.

As a best practice, Oracle recommends that you archive and purge unified audit trail records on a regular basis.

**Related Topics**

- Archiving the Audit Trail
  To maintain the integrity and reliability of audit data, keep only minimal required audit data locally in the database.
- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.

## 32.1.3 How Audit Trail Records Are Written to the AUDSYS Schema

Oracle Database automatically writes audit records to an internal relational table in the `AUDSYS` schema.

Writing audit records to a relational table in the `AUDSYS` schema prevents the risk of audit records being lost in the event of an instance crash or during a `SHUTDOWN ABORT` operation. By

default, the AUDSYS schema is dictionary protected, which means that other users cannot use system privileges (including ANY privileges) to modify or tamper with its data.

> **Note:**
>
> In Oracle Database 12c release 1 (12.1), you had the option of queuing the audit records in memory (queued-write mode) and be written periodically to the AUDSYS schema audit table. However, starting with Oracle Database 12c release 2 (12.2), immediate-write mode and queued-write mode are deprecated. The parameters that controlled them (DBMS_AUDIT_MGMT.AUDIT_TRAIL_IMMEDIATE_WRITE and DBMS_AUDIT_MGMT.AUDIT_TRAIL_QUEUED_WRITE), while still viewable, no longer have any functionality.
> If you have upgraded from Oracle Database 12c release 1 (12.1) and migrated to unified auditing in that release, then Oracle recommends that you use the DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS procedure to transfer the audit records as generated in the previous release to the AUDSYS audit internal table.
> *Oracle Database Upgrade Guide* provides information about transferring unified audit records after an upgrade.

**Related Topics**

- *Oracle Database Upgrade Guide*

## 32.1.4 Writing the Unified Audit Trail Records to SYSLOG or the Windows Event Viewer

You can write the unified audit trail records to SYSLOG or the Windows Event Viewer by setting an initialization parameter.

- About Writing the Unified Audit Trail Records to SYSLOG or the Windows Event Viewer
  With this feature, you can copy some of the key unified audit fields to SYSLOG or the Windows Event Viewer.
- Enabling SYSLOG and Windows Event Viewer Captures for the Unified Audit Trail
  You can write a subset of unified audit trail records to the UNIX SYSLOG or to the Windows Event Viewer.

### 32.1.4.1 About Writing the Unified Audit Trail Records to SYSLOG or the Windows Event Viewer

With this feature, you can copy some of the key unified audit fields to SYSLOG or the Windows Event Viewer.

Only key fields of unified audit records in the UNIFIED_AUDIT_TRAIL data dictionary view are copied to SYSLOG. SYSLOG records in a unified audit environment provide proof of operational integrity.

You can configure this feature on both UNIX and Microsoft Windows systems. On Windows systems, you either enable it or disable it. If enabled, it writes the records to the Windows Event Viewer.

On UNIX systems, you can fine-tune the capture of unified audit trail records for SYSLOG to specify the facility where the SYSLOG records are sent and the severity level of the records (for example, DEBUG if it is capturing debugging-related messages).

Table 32-1 maps the names given to the unified audit records fields that are written to SYSLOG and the Windows Event Viewer to the corresponding column names in the UNIFIED_AUDIT_TRAIL view.

**Table 32-1 Audit Record Field Names for SYSLOG and the Windows Event Viewer**

| Field Name | Column Name in UNIFIED_AUDIT_TRAIL | Column Type | Column Description |
|---|---|---|---|
| TYPE | AUDIT_TYPE | NUMBER | Type of the audit record |
| DBID | DBID | NUMBER | Database identifier |
| SESID | SESSION_ID | NUMBER | Session identifier |
| CLIENTID | CLIENT_IDENTIFIER | VARCHAR2 | Client identifier in the session |
| STMTID | STATEMENT_ID | NUMBER | Identifier for each statement run in the system |
| DBUSER | DB_USERNAME | VARCHAR2 | Session user |
| CURUSER | CURRENT_USER | VARCHAR2 | Effective user for the audited event |
| ACTION | ACTION | NUMBER | Action code of the audited event |
| RETCODE | RETURN_CODE | NUMBER | Return code for the audited event |
| SCHEMA | OBJECT_SCHEMA | VARCHAR2 | Schema name of the object |
| OBJNAME | OBJECT_NAME | VARCHAR2 | Name of the object |
| PDB_GUID | NULL (there are no columns in UNIFIED_AUDIT_TRAIL for this field) | VARCHAR2 | GUID of the container in which the unified audit record is generated |

## 32.1.4.2 Enabling SYSLOG and Windows Event Viewer Captures for the Unified Audit Trail

You can write a subset of unified audit trail records to the UNIX SYSLOG or to the Windows Event Viewer.

1. Locate the init.ora initialization file, which by default is in the $ORACLE_HOME/dbs directory.
2. Edit the init.ora file to include the UNIFIED_AUDIT_SYSTEMLOG parameter.

   You can set UNIFIED_AUDIT_SYSTEMLOG in either the CDB root or in a PDB.

   In an Oracle Database Real Application Clusters (Oracle RAC) environment, set UNIFIED_AUDIT_SYSTEMLOG to the same value on each Oracle RAC instance.

- On Windows, set UNIFIED_AUDIT_SYSTEMLOG to either TRUE or FALSE. TRUE writes the SYSLOG values to the Windows Event Viewer; FALSE disables the parameter. On Windows, the default is FALSE. For example:

  ```
  UNIFIED_AUDIT_SYSTEMLOG = TRUE
  ```

- On UNIX systems, use the following syntax:

  ```
  UNIFIED_AUDIT_SYSTEMLOG = 'facility_clause.priority_clause'
  ```

  There is no default setting for UNIFIED_AUDIT_SYSTEMLOG on UNIX systems.

  In this specification:

  - *facility_clause* refers to the facility to which you will write the audit trail records. Valid choices are USER and LOCAL. If you enter LOCAL, then optionally append 0–7 to designate a local custom facility for the SYSLOG records.
  - *priority_clause* refers to the type of warning in which to categorize the record. Valid choices are NOTICE, INFO, DEBUG, WARNING, ERR, CRIT, ALERT, and EMERG.

For example:

```
UNIFIED_AUDIT_SYSTEMLOG = 'LOCAL7.EMERG'
```

3. On UNIX platforms, to write unified audit records to SYSLOG set the UNIFIED_AUDIT_COMMON_SYSTEMLOG parameter to either TRUE or FALSE in the init.ora file in the root.

   Setting UNIFIED_AUDIT_COMMON_SYSTEMLOG to TRUE writes predefined columns of unified audit records from common unified audit policies to SYSLOG. FALSE disables these columns from being written to SYSLOG.

   You cannot set this parameter in a pluggable database (PDB). There is no Windows equivalent of the UNIFIED_AUDIT_COMMON_SYSTEMLOG parameter.

4. Add the audit file destination to the SYSLOG configuration file /etc/syslog.conf.

   For example, assuming you had set the UNIFIED_AUDIT_SYSTEMLOG to LOCAL7.EMERG, enter the following:

   ```
   local7.emerg /var/log/audit.log
   ```

   This setting logs all emergency messages to the /var/log/audit.log file.

5. Restart the SYSLOG logger.

   ```
   $/etc/rc.d/init.d/syslog restart
   ```

   Now, all audit records will be captured in the file /var/log/audit.log through the syslog daemon.

6. Log back in to the database instance.
7. Restart the database.

For example:

```
SHUTDOWN IMMEDIATE
STARTUP
```

If you set `UNIFIED_AUDIT_SYSTEMLOG` in a PDB, then close and reopen the PDB:

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

**Related Topics**

- About Writing the Unified Audit Trail Records to SYSLOG or the Windows Event Viewer
  With this feature, you can copy some of the key unified audit fields to SYSLOG or the Windows Event Viewer.
- *Oracle Database Reference*

## 32.1.5 How Unified Audit Records are Written to the Operating System

When the database cannot write audit trail records in the database itself, Oracle Database writes these records to operating system spillover audit files (`.bin` format).

This can happen in situations such as the following:

- The audit tablespace is offline.
- The tablespace is read only.
- The tablespace is full.
- The database is read only.

The default locations for unified audit spillover `.bin` files are as follows:

- **For pluggable databases (PDBs):** `$ORACLE_BASE/audit/$ORACLE_SID/`*`PDB_GUID`*
- **For the CDB root:** `$ORACLE_BASE/audit/$ORACLE_SID/`

The unified audit records will continue to be written to OS spillover files until the OS disk space becomes full. At this point, when there is no room in the OS for the audit records, user auditable transactions will fail with `ORA-02002 error while writing to audit trail` errors. To prevent this problem, Oracle recommends that you purge the audit trail on a regular basis.

**Related Topics**

- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.

## 32.1.6 Moving Operating System Audit Records into the Unified Audit Trail

Audit records that have been written to the spillover audit files can be moved to the unified audit trail database table.

When the database is not writable (such as during database mounts), if the database is closed, or if it is read-only, then Oracle Database writes the audit records to these external files. The default location for these external files is the `$ORACLE_BASE/audit/$ORACLE_SID` directory.

You can load the files into the database by running the DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES procedure. If you are loading a large number of operating system audit records in the external files, then consider the impact on the performance.

Follow these steps to load the audit records from operating system files to the AUDSYS schema audit table when the database is writable:

1. Log into the database as a user who has been granted the AUDIT_ADMIN role.

   Before you can upgrade to the current release or Oracle Database, you must run the DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES procedure from the CDB root to avoid losing operating system spillover files during the upgrade process.

2. Ensure that the database is open and writable.

   To find if the database is open and writable, query the V$DATABASE view.

   ```
   SELECT NAME, OPEN_MODE FROM V$DATABASE;

   NAME            OPEN_MODE
   --------------- ----------
   HRPDB           READ WRITE
   ```

   You can run the show pdbs command to find information about PDBs associated with the current instance.

3. Run the DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES procedure.

   For example:

   ```
   EXEC DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES;
   ```

   If you want to load a specific batch size of spillover operating system audit files, include the load_batch_size parameter. For example, to load 10 spillover files for the current container:

   ```
   BEGIN
    DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES(
     container        => 1,
     load_batch_size  => 10);
   END;
   /
   ```

   If you omit the load_batch_size parameter, then the default value of load_batch_size is 3. In that case, EXEC DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES; only loads 3 files at a time.

4. If you want to load individual PDB audit records, then log in to each PDB and run the DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES procedure again.

The audit records are loaded into the AUDSYS schema audit table immediately, and then deleted from the $ORACLE_BASE/audit/$ORACLE_SID directory.

If the session ID linked to the spillover audit files is owned by the PMON process, then the files can't be loaded until the database is restarted.

## 32.1.7 Improving the Performance of Queries and Purge Operations

If the partition on which the AUDSYS.AUD$UNIFIED table is located is too large, then queries to and purges of the UNIFIED_AUDIT_TRAIL data dictionary view may take a long time to complete.

- To improve performance, break the partition into smaller portions by using the `ALTER TABLE SPLIT PARTITION` statement.

  For example:

```
ALTER TABLE "AUDSYS"."AUD$UNIFIED" SPLIT PARTITION "SYS_P1602"
 INTO
   (PARTITION SYS_P1602_1 VALUES LESS THAN (DATE '2020-08-15'),
    PARTITION SYS_P1602
    );
```

**Related Topics**

- *Oracle Database VLDB and Partitioning Guide*

## 32.1.8 Using Oracle Data Pump to Export and Import Unified Audit Trail Records

You can include the unified audit trail in Oracle Database Pump export and import dump files.

The unified audit trail is automatically included in either full database or partial database export and import operations using Oracle Data Pump. As part of the schema level export or import operation, Oracle Database does not include the audit policy's metadata in the `SYS` schema during the export or import operation. Instead, use full export (`expdp`) or import (`impdp`) for the export and import of the metadata in unified audit policies.

For example, for a partial database export operation that does not use schema level export or import, if you wanted to export only the unified audit trail tables, then you could enter the following commands:

1. In SQL*Plus, move any operating system audit records that have been written to the spillover audit files to the unified audit trail table. Doing so ensures that all records will be exported.
2. From the operating system prompt, run the following command:

```
expdp system
full=y
directory=aud_dp_dir
logfile=audexp_log.log
dumpfile=audexp_dump.dmp
version=18.02.00.02.00
INCLUDE=AUDIT_TRAILS

Password: password
```

Next, you can import all the exported content by reading the export dump file. This operation imports only the unified audit trail tables.

```
impdp system
full=y
directory=aud_dp_dir
dumpfile=audexp_dump.dmp
logfile=audimp_log.log

Password: password
```

You do not need to perform any special configuration to achieve this operation. However, you must have the `EXP_FULL_DATABASE` role if you are performing the export operation and the `IMP_FULL_DATABASE` role if you are performing the import operation.

**Related Topics**

- Moving Operating System Audit Records into the Unified Audit Trail
  Audit records that have been written to the spillover audit files can be moved to the unified audit trail database table.

### 32.1.9 How Do Cursors Affect Auditing?

For each execution of an auditable operation within a cursor, Oracle Database inserts one audit record into the audit trail.

Events that cause cursors to be reused include the following:

- An application, such as Oracle Forms, holding a cursor open for reuse
- Subsequent execution of a cursor using new bind variables
- Statements run within PL/SQL loops where the PL/SQL engine optimizes the statements to reuse a single cursor

Auditing is *not* affected by whether or not a cursor is shared. Each user creates their own audit trail records on first execution of the cursor.

## 32.2 Archiving the Audit Trail

To maintain the integrity and reliability of audit data, keep only minimal required audit data locally in the database.

Move audit data to a dedicated repository outside of the source database (such as Oracle Audit Vault and Database Firewall (AVDF) or Oracle Data Safe) for long-term audit data retention and detailed analysis.

- Archiving the Traditional Operating System Audit Trail
  You can create an archive of the traditional operating system audit files after you have upgraded Oracle Database.
- Archiving the Unified and Traditional Database Audit Trails
  You should periodically archive and then purge the audit trail to prevent it from growing too large.

### 32.2.1 Archiving the Traditional Operating System Audit Trail

You can create an archive of the traditional operating system audit files after you have upgraded Oracle Database.

To archive the traditional operating system audit trail from an upgraded database, use your platform-specific operating system tools to create an archive of the traditional operating system audit files.

> **Note:**
>
> Traditional auditing is desupported in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

- Use the following methods to archive the traditional operating system audit files:

- **Use Oracle Audit Vault and Database Firewall.** You install Oracle Audit Vault and Database Firewall separately from Oracle Database.
- **Create tape or disk backups.** You can create a compressed file of the audit files, and then store it on tapes or disks. Consult your operating system documentation for more information.

Afterwards, you should purge (delete) the traditional operating system audit records to facilitate audit trail management.

**Related Topics**

- Moving Operating System Audit Records into the Unified Audit Trail
  Audit records that have been written to the spillover audit files can be moved to the unified audit trail database table.
- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.
- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

## 32.2.2 Archiving the Unified and Traditional Database Audit Trails

You should periodically archive and then purge the audit trail to prevent it from growing too large.

Archiving and purging facilitate the purging of the database audit trail.

You can create an archive of the unified and traditional database audit trail by using Oracle Audit Vault and Database Firewall or Oracle Data Safe. You install both of these products separately from Oracle Database.

> **Note:**
>
> Traditional auditing is desupported in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

After you complete the archive, you can purge the database audit trail contents.

**Related Topics**

- Purging Audit Trail Records
  The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.
- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

# 32.3 Purging Audit Trail Records

The `DBMS_AUDIT_MGMT` PL/SQL package can schedule automatic purge jobs, manually purge audit records, and perform other audit trail operations.

- About Purging Audit Trail Records
  You can use a variety of ways to purge audit trail records.
- Selecting an Audit Trail Purge Method
  You can perform the purge on a regularly scheduled basis or at a specified times.
- Scheduling an Automatic Purge Job for the Audit Trail
  Scheduling an automatic purge job requires planning beforehand, such as tuning the online and archive redo log sizes.
- Manually Purging the Audit Trail
  You can use the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` procedure to manually purge the audit trail.
- Other Audit Trail Purge Operations
  Other kinds of audit trail purge include enabling or disabling the audit trail purge job or setting the default audit trail purge job interval.
- Example: Directly Calling a Unified Audit Trail Purge Operation
  You can create a customized archive procedure to directly call a unified audit trail purge operation.
- Purge CLI Records in Databases Upgraded from Oracle Database 12.1 or Earlier
  In Oracle Database 12c release 12.1, the unified audit records used to reside in the common logging infrastructure (CLI) SGA back-end tables.

**Related Topics**

- Managing the Unified Audit Trail
  Unified auditing is enabled by default, and audit trail management ensures audit configuration is efficient for your needs.

## 32.3.1 About Purging Audit Trail Records

You can use a variety of ways to purge audit trail records.

You should periodically archive and then delete (purge) audit trail records. You can purge a subset of audit trail records or create a purge job that performs at a specified time interval. Oracle Database either purges the audit trail records that were created before the archive timestamp, or it purges all audit trail records. You can purge audit trail records in both read-write and read-only databases.

The purge process takes into account not just the unified audit trail, but audit trails from earlier releases of Oracle Database. For example, if you have migrated an upgraded database that still has operating system or XML audit records, then you can use the procedures in this section to archive and purge them.

To perform the audit trail purge tasks, you use the `DBMS_AUDIT_MGMT` PL/SQL package. You must have the `AUDIT_ADMIN` role before you can use the `DBMS_AUDIT_MGMT` package. Oracle Database mandatorily audits all executions of the `DBMS_AUDIT_MGMT` PL/SQL package procedures.

If you have Oracle Database activity monitoring solutions such as Oracle Audit Vault and Database Firewall (AVDF) or Oracle Data Safe to collect audit data, refer to the documentation of these solutions to check the specific recommendations for purge process.

> **Note:**
>
> Oracle Database audits all deletions from the audit trail, without exception.

**Related Topics**

- *Oracle Database PL/SQL Packages and Types Reference*

## 32.3.2 Selecting an Audit Trail Purge Method

You can perform the purge on a regularly scheduled basis or at a specified times.

- Purging the Audit Trail on a Regularly Scheduled Basis
  You can purge all audit records, or audit records that were created before a specified timestamp, on a regularly scheduled basis.
- Purging the Audit Trail on Demand
  You can manually purge the audit records on demand rather than scheduling the purge.

### 32.3.2.1 Purging the Audit Trail on a Regularly Scheduled Basis

You can purge all audit records, or audit records that were created before a specified timestamp, on a regularly scheduled basis.

For example, you can schedule the purge for every Saturday at 2 a.m.

1. Ensure that online and archive redo log sizes are tuned to accommodate the additional records generated during the audit table purge process.
2. Plan a timestamp and archive strategy.
3. Optionally, set an archive timestamp for the audit records.
4. Create and schedule the purge job.

**Related Topics**

- Scheduling an Automatic Purge Job for the Audit Trail
  Scheduling an automatic purge job requires planning beforehand, such as tuning the online and archive redo log sizes.

### 32.3.2.2 Purging the Audit Trail on Demand

You can manually purge the audit records on demand rather than scheduling the purge.

1. Ensure that online and archive redo log sizes are tuned to accommodate the additional records that were generated during the audit table purge process.
2. Plan a timestamp and archive strategy.
3. Optionally, set an archive timestamp for the audit records.
4. Run the purge operation.

**Related Topics**

- Manually Purging the Audit Trail
  You can use the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` procedure to manually purge the audit trail.

## 32.3.3 Scheduling an Automatic Purge Job for the Audit Trail

Scheduling an automatic purge job requires planning beforehand, such as tuning the online and archive redo log sizes.

- About Scheduling an Automatic Purge Job
  You can purge the entire audit trail, or purge older audit records in an audit trail that was created before a specific time period.
- Step 1: Ensure Online and Archive redo Log Sizes Are Tuned Appropriately
  The purge process may generate additional redo logs.
- Step 2: Optionally, Set an Archive Timestamp for Audit Records
  If you want to delete all of the audit trail, then you can bypass this step.
- Step 3: Create and Schedule the Purge Job
  You can use the `DBMS_AUDIT_MGMT` PL/SQL package to create and schedule the purge job.

### 32.3.3.1 About Scheduling an Automatic Purge Job

You can purge the entire audit trail, or purge older audit records in an audit trail that was created before a specific time period.

Be aware that purging the audit trail, particularly a large one, can take a while to complete. Oracle recommends that you schedule the purge job at a time when the database is not busy. If the audit trail is considerably large, then the purge process can take a while to complete.

You can create multiple purge jobs for different audit trail types, so long as they do not conflict. For example, you can create a purge job for the standard audit trail table and then the fine-grained audit trail table. However, you cannot then create a purge job for both or all types, that is, by using the `DBMS_AUDIT_MGMT.AUDIT_TRAIL_DB_STD` or `DBMS_AUDIT_MGMT.AUDIT_TRAIL_ALL` property.

> **Note:**
>
> In addition, be aware that the jobs created by the `DBMS_SCHEDULER` PL/SQL package do not run on a read-only database. An automatic purge job created with `DBMS_AUDIT_MGMT` uses the `DBMS_SCHEDULER` package to schedule the tasks. Therefore, these jobs cannot run on a database or PDB that is open in read-only mode.

### 32.3.3.2 Step 1: Ensure Online and Archive redo Log Sizes Are Tuned Appropriately

The purge process may generate additional redo logs.

You may consider skipping the step if you have turned **off** traditional auditing in the upgraded instance.

- Ensure that the online and archive redo log sizes accommodate the additional records generated during the audit table purge process.

  In a unified auditing environment, the purge process does not generate as many redo logs as in a mixed mode auditing environment, so if you have migrated to unified auditing, then you may want to bypass this step.

**Related Topics**

- *Oracle Database Administrator's Guide*

## 32.3.3.3 Step 2: Optionally, Set an Archive Timestamp for Audit Records

If you want to delete all of the audit trail, then you can bypass this step.

You must record the timestamp of the audit records before you can archive them. You can set a timestamp for when the last audit record was archived. Setting an archive timestamp provides the point of cleanup to the purge infrastructure. If you are setting a timestamp for a read-only database, then you can use the `DBMS_AUDIT.MGMT.GET_LAST_ARCHIVE_TIMESTAMP` function to find the last archive timestamp that was configured for the instance on which it was run. For a read-write database, you can query the `DBA_AUDIT_MGMT_LAST_ARCH_TS` data dictionary view.

To find the last archive timestamps for the unified audit trail, you can query the `DBA_AUDIT_MGMT_LAST_ARCH_TS` data dictionary view. After you set the timestamp, all audit records in the audit trail that indicate a time earlier than that timestamp are purged when you run the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` PL/SQL procedure. Optionally, you can clear the archive timestamp setting.

If you are using Oracle Database Real Application Clusters, then use Network Time Protocol (NTP) to synchronize the time on each computer where you have installed an Oracle Database instance. For example, suppose you set the time for one Oracle RAC instance node at 11:00:00 a.m. and then set the next Oracle RAC instance node at 11:00:05. As a result, the two nodes have inconsistent times. You can use Network Time Protocol (NTP) to synchronize the times for these Oracle RAC instance nodes.

1. As a user who has been granted the `AUDIT_ADMIN` role, log into the either the root or the PDB in which you want to schedule the purge job.

   In most cases, you may want to schedule the purge job on individual PDBs. For example, to log into a PDB called `hrpdb`:

   ```
   CONNECT aud_admin@hrpdb
   Enter password: password
   Connected.
   ```

2. Fnd the timestamp date, by querying the `DBA_AUDIT_MGMT_LAST_ARCH_TS` data dictionary view.

   The last archived timestamp is set automatically if you are using Oracle Audit Vault and Database Firewall or Oracle Data Safe after the audit record is collected. Later on, when the purge takes place, Oracle Database purges only the audit trail records that were created before the date of this archive timestamp. After you have timestamped the records, you are ready to archive them.

3. Run the `DBMS_AUDIT_MGMT.SET_LAST_ARCHIVE_TIMESTAMP` PL/SQL procedure to set the timestamp.

   For example:

   ```
   BEGIN
     DBMS_AUDIT_MGMT.SET_LAST_ARCHIVE_TIMESTAMP(
      AUDIT_TRAIL_TYPE     =>  DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
      LAST_ARCHIVE_TIME    =>  '12-OCT-2013 06:30:00.00',
      RAC_INSTANCE_NUMBER  =>  1,
      CONTAINER            =>  DBMS_AUDIT_MGMT.CONTAINER_CURRENT);
   END;
   /
   ```

In this example:

- AUDIT_TRAIL_TYPE specifies the audit trail type. DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED sets it for the unified audit trail.

  For upgraded databases that still have audit data from previous releases, use the following settings. Note that traditional auditing is desupported in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_AUD_STD is used for the traditional standard audit trail table, AUD$. (This setting does not apply to read-only databases.)
  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_FGA_STD is used for the traditional fine-grained audit trail table, FGA_LOG$. (This setting does not apply to read-only databases.)
  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS is used for the traditional operating system audit trail files with the .aud extension. (This setting does not apply to Windows Event Log entries.)
  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_XML is used for the XML traditional operating system audit trail files.

  To archive records from the AUDSYS.AUD$UNIFIED table or from the operating system spillover files:

  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_TABLE archives records from the AUDSYS.AUD$UNIFIED table.
  - DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_FILES archives records from the operating system spillover files in each database (primary or standby).

- LAST_ARCHIVE_TIME specifies the timestamp in YYYY-MM-DD HH:MI:SS.FF UTC (Coordinated Universal Time) format for AUDIT_TRAIL_UNIFIED, AUDIT_TRAIL_AUD_STD, and AUDIT_TRAIL_FGA_STD, and in the Local Time Zone for AUDIT_TRAIL_OS and AUDIT_TRAIL_XML. Do not enter a future system date or timestamp (for example, SYSDATE + 1, or a date in the future) for this value.
- RAC_INSTANCE_NUMBER specifies the instance number for an Oracle RAC installation. This setting is not relevant for single instance databases. If you specified the DBMS_AUDIT_MGMT.AUDIT_TRAIL_AUD_STD or DBMS_AUDIT_MGMT.AUDIT_TRAIL_FGA_STD audit trail types, then you can omit the RAC_INSTANCE_NUMBER argument. This is because there is only one AUD$ or FGA_LOG$ table, even for an Oracle RAC installation. The default is NULL. You can find the instance number for the current instance by issuing the SHOW PARAMETER INSTANCE_NUMBER command in SQL*Plus.
- CONTAINER applies the timestamp to either the current PDB or to all PDBs. DBMS_AUDIT_MGMT.CONTAINER_CURRENT specifies the current PDB; DBMS_AUDIT_MGMT.CONTAINER_ALL applies to all PDBs in the multitenant environment.

  Note that you can set CONTAINER to DBMS_MGMT.CONTAINER_ALL only from the root.

Typically, after you set the timestamp, you can use the DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL PL/SQL procedure to remove the audit records that were created before the timestamp date.

**Related Topics**

- Clearing the Archive Timestamp Setting
  The DBMS_AUDIT_MGMT.CLEAR_LAST_ARCHIVE_TIMESTAMP procedure can clear the archive timestamp setting.

## 32.3.3.4 Step 3: Create and Schedule the Purge Job

You can use the `DBMS_AUDIT_MGMT` PL/SQL package to create and schedule the purge job.

- Create and schedule the purge job by running the `DBMS_AUDIT_MGMT.CREATE_PURGE_JOB` PL/SQL procedure.

  For example:

```
CONNECT aud_admin@hrpdb
Enter password: password
Connected.

BEGIN
  DBMS_AUDIT_MGMT.CREATE_PURGE_JOB (
   AUDIT_TRAIL_TYPE            => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
   AUDIT_TRAIL_PURGE_INTERVAL  => 12,
   AUDIT_TRAIL_PURGE_NAME      => 'Audit_Trail_PJ',
   USE_LAST_ARCH_TIMESTAMP     => TRUE,
   CONTAINER                   => DBMS_AUDIT_MGMT.CONTAINER_CURRENT);
END;
/
```

  In this example:

  – `AUDIT_TRAIL_TYPE`: Specifies the audit trail type. `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED` sets it for the unified audit trail.

    For upgraded databases that still have audit data from previous releases, use the following settings. Note that traditional auditing is desupported in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_AUD_STD` is used for the standard audit trail table, `AUD$`. (This setting does not apply to read-only databases.)
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_FGA_STD` is used for the fine-grained audit trail table, `FGA_LOG$`. (This setting does not apply to read-only databases.)
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_DB_STD` is used for both standard and fine-grained audit trail tables. (This setting does not apply to read-only databases.)
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS` is used for the operating system audit trail files with the `.aud` extension. (This setting does not apply to Windows Event Log entries.)
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_XML` is used for the XML operating system audit trail files.
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_FILES` is used for both operating system and XML audit trail files.
    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_ALL` is used for all traditional audit trail records, that is, both database audit trail and operating system audit trail types. (This setting does not apply to read-only databases.)

    To purge records from the `AUDSYS.AUD$UNIFIED` table or from the operating system spillover files:

    * `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_TABLE` purges records from the `AUDSYS.AUD$UNIFIED` table.

    - DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_FILES purges records from the operating system spillover files in each database (primary or standby).

  - AUDIT_TRAIL_PURGE_INTERVAL specifies the hourly interval for this purge job to run. The timing begins when you run the DBMS_AUDIT_MGMT.CREATE_PURGE_JOB procedure, in this case, 12 hours after you run this procedure. Later on, if you want to update this value, run the DBMS_AUDIT_MGMT.SET_PURGE_JOB_INTERVAL procedure.

  - USE_LAST_ARCH_TIMESTAMP accepts either of the following settings:

    - TRUE deletes audit records created before the last archive timestamp. To check the last recorded timestamp, query the LAST_ARCHIVE_TS column of the DBA_AUDIT_MGMT_LAST_ARCH_TS data dictionary view for read-write databases and the DBMS_AUDIT_MGMT.GET_LAST_ARCHIVE_TIMESTAMP function for read-only databases. The default value is TRUE. Oracle recommends that you set USE_LAST_ARCH_TIMESTAMP to TRUE.

    - FALSE deletes all audit records without considering last archive timestamp. Be careful about using this setting, in case you inadvertently delete audit records that should not have been deleted.

    To purge records from the AUDSYS.AUD$UNIFIED table or from the operating system spillover files:

    - DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_TABLE purges records from the AUDSYS.AUD$UNIFIED table.

    - DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_FILES purges records from the operating system spillover files in each database (primary or standby).

  - CONTAINER defines where to create the purge job in the multienant environment. You can set it as follows:

    - DBMS_AUDIT_MGMT.CONTAINER_CURRENT can be set in either the CDB root or the current PDB, enabling the purge job to be available, visible, and managed from these locations. If set in the CDB root, then the purge job applies only to the CDB root; if set in the current PDB, then it applies only to that PDB.

    - DBMS_AUDIT_MGMT.CONTAINER_ALL is set in the CDB root, enabling the purge job to be a global job, which runs according to the defined job schedule. When the job is invoked, it cleans up audit trails in all the PDBs in the multitenant environment. If you create the job in the CDB root, then it is visible only in the CDB root. Hence, you can enable, disable, and drop it from the CDB root only.

## 32.3.4 Manually Purging the Audit Trail

You can use the DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL procedure to manually purge the audit trail.

- About Manually Purging the Audit Trail
  You can manually purge the audit trail right away, without scheduling a purge job.
- Using DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL to Manually Purge the Audit Trail
  After you complete preparatory steps, you can use the DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL procedure to manually purge the audit trail.

### 32.3.4.1 About Manually Purging the Audit Trail

You can manually purge the audit trail right away, without scheduling a purge job.

Similar to a purge job, you can purge audit trail records that were created before an archive timestamp date or all the records in the audit trail. Only the current audit directory is cleaned up when you run this procedure.

For upgraded databases that may still have audit trails from earlier releases, note the following about the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` PL/SQL procedure:

- On Microsoft Windows, because the `DBMS_AUDIT_MGMT` package does not support cleanup of Windows Event Viewer, setting the `AUDIT_TRAIL_TYPE` property to `DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS` has no effect. This is because operating system audit records on Windows are written to Windows Event Viewer. The `DBMS_AUDIT_MGMT` package does not support this type of cleanup operation.
- On UNIX platforms, if you had set the `AUDIT_SYSLOG_LEVEL` (deprecated) initialization parameter, then Oracle Database writes the operating system log files to syslog files. (Be aware that when you configure the use of syslog files, the messages are sent to the syslog daemon process. The syslog daemon process does not return an acknowledgment to Oracle Database indicating a committed write to the syslog files.) If you set the `AUDIT_TRAIL_TYPE` property to `DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS`, then the procedure only removes `.aud` files under audit directory (This directory is specified by the `AUDIT_FILE_DEST` (deprecated) initialization parameter).

## 32.3.4.2 Using DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL to Manually Purge the Audit Trail

After you complete preparatory steps, you can use the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` procedure to manually purge the audit trail.

1. If you have set the `AUDIT_SYSLOG_LEVEL` (deprecated) initialization parameter so that the audit trail will be written to operating system log files (`syslog`), then check for the following:
   - Ensure that no one is currently writing to the audit trail files.
   - Ensure that the session ID that is associated with the audit trail files is not owned by the PMON process.

   If either of these conditions is true, then the audit trail cannot be purged.
2. Perform the following scheduling tasks:
   - If necessary, tune the online and archive redo log sizes.
   - Plan a timestamp and archive strategy.
   - Optionally, set an archive timestamp for the audit records.
3. Connect to the root or to the PDB in which you created the purge job.

   If you created the purge job in the root, then you must log into the root. If you created the purge job in a specific PDB, then log into that PDB.
4. Purge the audit trail records by running the `DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL` PL/SQL procedure.

   For example:

```
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;
/
```

In this example:

- `AUDIT_TRAIL_TYPE`: Specifies the audit trail type. `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED` sets it for the unified audit trail.

  For upgraded databases that still have audit data from previous releases, use the following settings. Note that traditional auditing is desupported in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_AUD_STD`: Standard audit trail table, `AUD$`. (This setting does not apply to read-only databases.)
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_FGA_STD`: Fine-grained audit trail table, `FGA_LOG$`. (This setting does not apply to read-only databases.)
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_DB_STD`: Both standard and fine-grained audit trail tables. (This setting does not apply to read-only databases)
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS`: Operating system audit trail files with the `.aud` extension. (This setting does not apply to Windows Event Log entries.)
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_XML`: XML Operating system audit trail files.
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_FILES`: Both operating system and XML audit trail files.
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_ALL`: All audit trail records, that is, both database audit trail and operating system audit trail types. (This setting does not apply to read-only databases.)

  To purge records from the `AUDSYS.AUD$UNIFIED` table or from the operating system spillover files:

  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_TABLE` purges records from the `AUDSYS.AUD$UNIFIED` table.
  - `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_FILES` purges records from the operating system spillover files in each database (primary or standby).

- `USE_LAST_ARCH_TIMESTAMP`: Enter either of the following settings:
  - `TRUE`: Deletes audit records created before the last archive timestamp. The default (and recommended) value is `TRUE`. Oracle recommends that you set `USE_LAST_ARCH_TIMESTAMP` to `TRUE`.
  - `FALSE`: Deletes all audit records without considering last archive timestamp. Be careful about using this setting, in case you inadvertently delete audit records that should not have been deleted.
- `CONTAINER`: Applies the cleansing to either the current PDB or to all PDBs. `DBMS_AUDIT_MGMT.CONTAINER_CURRENT` specifies the current PDB; `DBMS_AUDIT_MGMT.CONTAINER_ALL` applies to all PDBs.

**Related Topics**

- Step 1: Ensure Online and Archive redo Log Sizes Are Tuned Appropriately
  The purge process may generate additional redo logs.
- Step 2: Optionally, Set an Archive Timestamp for Audit Records
  If you want to delete all of the audit trail, then you can bypass this step.

## 32.3.5 Other Audit Trail Purge Operations

Other kinds of audit trail purge include enabling or disabling the audit trail purge job or setting the default audit trail purge job interval.

- Enabling or Disabling an Audit Trail Purge Job
  The `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure enables or disables an audit trail purge job.
- Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job
  You can set a default purge operation interval, in hours, that must pass before the next purge job operation takes place.
- Deleting an Audit Trail Purge Job
  You can delete existing audit trail purge jobs.
- Clearing the Archive Timestamp Setting
  The `DBMS_AUDIT_MGMT.CLEAR_LAST_ARCHIVE_TIMESTAMP` procedure can clear the archive timestamp setting.

### 32.3.5.1 Enabling or Disabling an Audit Trail Purge Job

The `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure enables or disables an audit trail purge job.

Where you run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure in the multitenant environment depends on the location of the purge job, which is determined by the `CONTAINER` parameter of the `DBMS_MGMT.CREATE_PURGE_JOB` procedure. If you had set `CONTAINER` to `CONTAINER_ALL` (to create the purge job in the root), then you must run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure from the root. If you had set `CONTAINER` to `CONTAINER_CURRENT`, then you must run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure from the PDB in which it was created.

- To enable or disable an audit trail purge job, use the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` PL/SQL procedure.

  For example, assuming that you had created the purge job in a the `hrpdb` PDB:

  ```
  CONNECT aud_admin@hrpdb
  Enter password: password
  Connected.

  BEGIN
   DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS(
    AUDIT_TRAIL_PURGE_NAME      => 'Audit_Trail_PJ',
    AUDIT_TRAIL_STATUS_VALUE    => DBMS_AUDIT_MGMT.PURGE_JOB_ENABLE);
  END;
  /
  ```

  In this example:

  – `AUDIT_TRAIL_PURGE_NAME` specifies a purge job called `Audit_Trail_PJ`. To find existing purge jobs, query the `JOB_NAME` and `JOB_STATUS` columns of the `DBA_AUDIT_MGMT_CLEANUP_JOBS` data dictionary view.

  – `AUDIT_TRAIL_STATUS_VALUE` accepts either of the following properties:

    * `DBMS_AUDIT_MGMT.PURGE_JOB_ENABLE` enables the specified purge job.
    * `DBMS_AUDIT_MGMT.PURGE_JOB_DISABLE` disables the specified purge job.

### 32.3.5.2 Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job

You can set a default purge operation interval, in hours, that must pass before the next purge job operation takes place.

The interval setting that is used in the `DBMS_AUDIT_MGMT.CREATE_PURGE_JOB` procedure takes precedence over this setting.

- To set the default audit trail purge job interval for a specific purge job, run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_INTERVAL` procedure.

  For example, assuming that you had created the purge job in the `hrpdb` PDB:

```
CONNECT aud_admin@hrpdb
Enter password: password
Connected.

BEGIN
 DBMS_AUDIT_MGMT.SET_PURGE_JOB_INTERVAL(
  AUDIT_TRAIL_PURGE_NAME       => 'Audit_Trail_PJ',
  AUDIT_TRAIL_INTERVAL_VALUE   => 24);
END;
/
```

  In this example:

  - `AUDIT_TRAIL_PURGE_NAME` specifies the name of the audit trail purge job. To find a list of existing purge jobs, query the `JOB_NAME` and `JOB_STATUS` columns of the `DBA_AUDIT_MGMT_CLEANUP_JOBS` data dictionary view.
  - `AUDIT_TRAIL_INTERVAL_VALUE` updates the default hourly interval set by the `DBMS_AUDIT_MGMT.CREATE_PURGE_JOB` procedure. Enter a value between `1` and `999`. The timing begins when you run the purge job.

  Where you run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_INTERVAL` procedure depends on the location of the purge job, which is determined by the `CONTAINER` parameter of the `DBMS_MGMT.CREATE_PURGE_JOB` procedure. If you had set `CONTAINER` to `CONTAINER_ALL`, then the purge job exists in the root, so you must run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure from the root. If you had set `CONTAINER` to `CONTAINER_CURRENT`, then you must run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_INTERVAL` procedure from the PDB in which it was created.

### 32.3.5.3 Deleting an Audit Trail Purge Job

You can delete existing audit trail purge jobs.

To find existing purge jobs, query the `JOB_NAME` and `JOB_STATUS` columns of the `DBA_AUDIT_MGMT_CLEANUP_JOBS` data dictionary view.

- To delete an audit trail purge job, use the `DBMS_AUDIT_MGMT.DROP_PURGE_JOB` PL/SQL procedure.

  For example, assuming that you had created the purge job in the `hrpdb` PDB:

```
CONNECT aud_admin@hrpdb
Enter password: password
Connected.

BEGIN
```

```
 DBMS_AUDIT_MGMT.DROP_PURGE_JOB(
  AUDIT_TRAIL_PURGE_NAME  => 'Audit_Trail_PJ');
END;
/
```

Where you run the `DBMS_AUDIT_MGMT.DROP_PURGE_JOB` procedure in the multitenant environment depends on the location of the purge job, which is determined by the `CONTAINER` parameter of the `DBMS_MGMT.CREATE_PURGE_JOB` procedure. If you had set `CONTAINER` to `CONTAINER_ALL`, then the purge job exists in the root, so you must run the `DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS` procedure from the root. If you had set `CONTAINER` to `CONTAINER_CURRENT`, then you must run the `DBMS_AUDIT_MGMT.DROP_PURGE_JOB_INTERVAL` procedure from the PDB in which it was created.

## 32.3.5.4 Clearing the Archive Timestamp Setting

The `DBMS_AUDIT_MGMT.CLEAR_LAST_ARCHIVE_TIMESTAMP` procedure can clear the archive timestamp setting.

To find a history of audit trail log cleanup, you can query the `UNIFIED_AUDIT_TRAIL` data dictionary view, using the following criteria: `OBJECT_NAME` is `DBMS_AUDIT_MGMT`, `OBJECT_SCHEMA` is `SYS`, and `SQL_TEXT` is set to `LIKE %DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL%`.

- To clear the archive timestamp setting, use the `DBMS_AUDIT_MGMT.CLEAR_LAST_ARCHIVE_TIMESTAMP` PL/SQL procedure to specify the audit trail type.

  For example, assuming that you had created the purge job in the `hrpdb` PDB:

  ```
  CONNECT aud_admin@hrpdb
  Enter password: password
  Connected.

  BEGIN
    DBMS_AUDIT_MGMT.CLEAR_LAST_ARCHIVE_TIMESTAMP(
     AUDIT_TRAIL_TYPE     =>  DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
     CONTAINER            =>  DBMS_AUDIT_MGMT.CONTAINER_CURRENT);
  END;
  /
  ```

  In this example:

  - `AUDIT_TRAIL_TYPE` is set for the unified audit trail. If the `AUDIT_TRAIL_TYPE` property is set to `DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS` or `DBMS_AUDIT_MGMT.AUDIT_TRAIL_XML`, then you cannot set `RAC_INSTANCE_NUMBER` to `0`. You can omit the `RAC_INSTANCE_NUMBER` setting if you set `AUDIT_TRAIL_TYPE` to `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED`.

    You can clear the archive timestamps from the `AUDSYS.AUD$UNIFIED` table by setting `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_TABLE`. To clear the archive timestamps from the operating system spillover files in each database (primary or standby), set `DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED_FILES`.

  - `CONTAINER` specifies where to perform the purge. `DBMS_AUDIT_MGMT.CONTAINER_CURRENT` specifies the local PDB; `DBMS_AUDIT_MGMT.CONTAINER_ALL` applies to all containers in the CDB environment.

## 32.3.6 Example: Directly Calling a Unified Audit Trail Purge Operation

You can create a customized archive procedure to directly call a unified audit trail purge operation.

The pseudo code in Example 32-2 creates a database audit trail purge operation that the user calls by invoking the `DBMS_ADUIT.CLEAN_AUDIT_TRAIL` procedure for the unified audit trail.

The purge operation deletes records that were created before the last archived timestamp by using a loop. The loop archives the audit records, calculates which audit records were archived and uses the `SetCleanUpAuditTrail` call to set the last archive timestamp, and then calls the `CLEAN_AUDIT_TRAIL` procedure. In this example, major steps are in **bold** typeface.

**Example 32-2 Directly Calling a Database Audit Trail Purge Operation**

```
-- 1. Set the last archive timestamp:
PROCEDURE SetCleanUpAuditTrail()
 BEGIN
  CALL FindLastArchivedTimestamp(AUD$);
  DBMS_AUDIT_MGMT.SET_LAST_ARCHIVE_TIMESTAMP(
   AUDIT_TRAIL_TYPE          => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
   LAST_ARCHIVE_TIME         => '23-AUG-2013 12:00:00',
   CONTAINER                 => DBMS_AUDIT_MGMT.CONTAINER_CURRENT);
 END;
/
-- 2. Run a customized archive procedure to purge the audit trail records:
BEGIN
  CALL MakeAuditSettings();
  LOOP (/* How long to loop*/)
    -- Invoke function for audit record archival
    CALL DoUnifiedAuditRecordArchival();

    CALL SetCleanUpAuditTrail();
    IF(/* Clean up is needed immediately */)
      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 IF
  END LOOP /*LOOP*/
END; /* PROCEDURE */
/
```

## 32.3.7 Purge CLI Records in Databases Upgraded from Oracle Database 12.1 or Earlier

In Oracle Database 12c release 12.1, the unified audit records used to reside in the common logging infrastructure (CLI) SGA back-end tables.

There is one CLI back-end table per GUID of the container and the correct GUID needs to be passed to purge audit records present in CLI table. When a pluggable database gets cloned, the unified audit tables get newly created in the new pluggable database with new GUID.

If the `container_guid` parameter is not passed during execution of the `CLEAN_AUDIT_TRAIL` procedure then the current GUID of the container will be used for purging and when the current GUID of the container is different from the old GUID, audit records do not get deleted from the CLI table.

To purge CLI records successfully in databases upgraded from Oracle Database release 12.1 or earlier:

1. Get GUIDs of CLI table by running the following command:

```
SQL> SELECT DISTINCT guid FROM sys.cli_tab$;
```

   If this command doesn't return any rows then you can skip the next step as there are no CLI tables in database.

2. Execute the CLEAN_AUDIT_TRAIL Procedure by passing each of these GUIDs one by one along with other parameters to ensure that you purge the unified audit records from these CLI back-end tables.

# 32.4 Audit Trail Management Data Dictionary Views

Oracle Database provides data dictionary views that list information about audit trail management settings.

Table 32-2 lists these views.

**Table 32-2 Views That Display Information about Audit Trail Management Settings**

| View | Description |
|---|---|
| DBA_AUDIT_MGMT_CLEAN_EVENTS | Displays the history of purge events of the traditional (that is, non-unified) audit trails. Periodically, as a user who has been granted the AUDIT_ADMIN role, you should delete the contents of this view so that it does not grow too large. For example:<br>`DELETE FROM DBA_AUDIT_MGMT_CLEAN_EVENTS;`<br>This view applies to read-write databases only. For read-only databases, a history of purge events is in the alert log.<br>For unified auditing, you can find a history of purged events by querying the UNIFIED_AUDIT_TRAIL data dictionary view, using the following criteria: OBJECT_NAME is DBMS_AUDIT_MGMT, OBJECT_SCHEMA is SYS, and SQL_TEXT is set to LIKE %DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL%. |
| DBA_AUDIT_MGMT_CLEANUP_JOBS | Displays the currently configured audit trail purge jobs |
| DBA_AUDIT_MGMT_CONFIG_PARAMS | Displays the currently configured audit trail properties that are used by the DBMS_AUDIT_MGMT PL/SQL package |
| DBA_AUDIT_MGMT_LAST_ARCH_TS | Displays the last archive timestamps that have set for audit trail purges |

**Related Topics**

- *Oracle Database Reference*