Database Security Guide

23 Administering the Audit Trail

As a user who has been granted the `AUDIT_ADMIN` role, you can manage the audit trail, archive the audit trail, and purge audit trail records.

Topics:

- Managing the Unified Audit Trail (audit_admin.htm#GUID-2A5CFDFC-320B-41D0-86AB-B524992E02A8)
- Archiving the Audit Trail (audit_admin.htm#GUID-58CFB1F4-7E76-46A4-8DDD-2546DBD67FBE)
- Purging Audit Trail Records (audit_admin.htm#GUID-9B891A44-3DF4-4B52-98D4-A931DBAEC1D)
- Audit Trail Management Data Dictionary Views (audit_admin.htm#GUID-2AC6F69F-5BEA-4AC7-9A22-2966121A43ED)

Managing the Unified Audit Trail

Auditing is enabled by default, but you can control when audit records are written to disk.

Topics:

- When Are Audit Records Created? (audit_admin.htm#GUID-4E26E99D-C2CC-4A35-8720-F9E1CA3D95CC)
- Activities That Are Mandatorily Audited (audit_admin.htm#GUID-AA781864-5756-464E-AFB6-675625AF0EF5)
- How Do Cursors Affect Auditing? (audit_admin.htm#GUID-59DE03A1-8446-4F38-B35E-306D61616F21)
- Writing the Unified Audit Trail Records to the AUDSYS Schema (audit_admin.htm#GUID-1DD625ED-AC75-47E7-ADF6-1C7C93656F22)
- Moving Operating System Audit Records into the Unified Audit Trail (audit_admin.htm#GUID-91B14CCD-E408-4669-82FA-EE833EDBBBA9)
- Disabling Unified Auditing (audit_admin.htm#GUID-80D9305C-29F6-4F3B-BDDB-371F619B08D8)

> **See Also:**
>
> Purging Audit Trail Records (audit_admin.htm#GUID-9B891A44-3DF4-4B52-98D4-A931DBAEC1D)

When Are Audit Records Created?

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

Oracle Database individually audits SQL statements inside PL/SQL program units, as necessary, when the program unit is run.

You can control the frequency when audit trail records are written to disk. If the write mode is set to queued, then the audit records are written periodically to the SGA and not to disk immediately. If you want to write the audit records immediately to disk, then you can set it to immediate-write mode. Otherwise, there is a minimum flush threshold configured by default which checks if the queues were flushed more than 3 seconds before and accordingly flushes the audit records in the SGA queue. However, owing to the database activity, the flush may not happen every 3 seconds and could take longer.

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 the session is already active, setting or changing statement or privilege unified audit options does not take effect in that session. The modified statement or privilege audit options take effect only when the current session ends and a new session is created.

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

By default, audit trail records are written to the `SYSAUX` tablespace. You can designate a different tablespace, including one that is encrypted, by using the `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION` procedure.

> **See Also:**
>
> - Writing the Unified Audit Trail Records to the AUDSYS Schema (audit_admin.htm#GUID-1DD625ED-AC75-47E7-ADF6-1C7C93656F22)for more information about immediate-write mode and queued-write mode
> - *Oracle Database Concepts* (../CNCPT/sqllangu.htm#CNCPT015)for information about the different phases of SQL statement processing and shared SQL
> - *Oracle Database PL/SQL Packages and Types Reference* (../ARPLS/d_audit_mgmt.htm#ARPLS65427)for more information about the `DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION` procedure

Activities That Are Mandatorily Audited

The `UNIFIED_AUDIT_TRAIL` data dictionary view captures activities from administrative users such as `SYSDBA`, `SYSBACKUP`, and `SYSKM`.

You do not need to audit the unified audit trail. The unified audit trail resides in a read-only table in the `AUDSYS` schema. Hence, DMLs are not permitted on the unified audit trail views. Even DML and DDL operations on the underlying dictionary tables from `AUDSYS` schema are not permitted.

The `SYSTEM_PRIVILEGE_USED` column shows the type of administrative privilege that was used for the activity.

The following audit-related activities, such as modifications to audit policies, are mandatorily audited:

- `CREATE AUDIT POLICY`
- `ALTER AUDIT POLICY`

- `DROP AUDIT POLICY`
- `AUDIT`
- `NOAUDIT`
- `EXECUTE` of the `DBMS_FGA` PL/SQL package
- `EXECUTE` of the `DBMS_AUDIT_MGMT` PL/SQL package
- `ALTER TABLE` attempts on the `AUDSYS` audit trail table (remember that this table cannot be altered)
- Top level statements by the administrative users `SYS`, `SYSDBA`, `SYSOPER`, `SYSASM`, `SYSBACKUP`, `SYSDG`, and `SYSKM`, until the database opens. When the database opens, Oracle Database audits these users using the audit configurations in the system—not just the ones that were applied using the `BY` clause in the `AUDIT` statement, for example, but those that were applied for all users when `AUDIT` statement does not have a `BY` clause or when the `EXCEPT` clause was used and these users were not excluded.
- All user-issued DML statements on the `SYS.AUD$` and `SYS.FGA_LOG$` dictionary tables
- Any attempts to modify the data or metadata of the unified audit internal table. `SELECT` statements on this table are not audited by default or mandatorily.
- All configuration changes that are made to Oracle Database Vault

> **See Also:**
>
> Auditing Administrative Users (audit_config.htm#GUID-D352B575-2EA9-451D-9170-FD9298993DEF)

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 executed 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 her or his own audit trail records on first execution of the cursor.

Writing the Unified Audit Trail Records to the AUDSYS Schema

Oracle Database writes audit records to the `AUDSYS` schema.

Topics:

- About Writing Unified Audit Trail Records to AUDSYS (audit_admin.htm#GUID-84EE2115-DE09-4028-ACE9-CC331389A9CA)
- Setting the Write Mode for Unified Audit Trail Records (audit_admin.htm#GUID-6C2F428C-E209-4626-8A5F-616CF0D31476)
- Manually Flushing Audit Records to the Audit Trail in Queued-Write Mode (audit_admin.htm#GUID-11FEBEC6-608F-48D9-8582-47BB555FC700)

About Writing Unified Audit Trail Records to AUDSYS

In a new or just-migrated Oracle Database installation, the `AUDSYS` schema is empty until unified auditing is initiated and records are generated.

> **Note:**
>
> Starting with Oracle Database 12*c* release 12.2, the ability to flush audit records deprecated but is retained for backward compatibility.

This design greatly improves the performance of the audit trail processes and the database as a whole. In the previous release, in the event of an instance crash or during `SHUTDOWN ABORT` operations, there was a chance that some audit records would be lost. Oracle recommends that you configure the audit trail to immediately write audit records to the `AUDSYS` schema audit table by using the `DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS` procedure to transfer the audit records to an internal relational table in the `AUDSYS` schema.

From the previous release, the following modes, deprecated but retained for backward compatibility, are available. However, Oracle recommends that you use the `DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS` procedure instead, as described in *Oracle Database Upgrade Guide* (http://www.oracle.com/pls/topic/lookup?ctx=E5052901&id=UPGRD-GUID-4BC5F146-BF0D-4BCF-8A0B-1B67B767EEF1).

- **Immediate-write mode.** This setting writes all audit records to the audit trail immediately. However, be aware that database performance may be affected.
- **Queued-write mode.** This setting, which is the default write mode, queues the audit records in memory to be written periodically to the `AUDSYS` schema audit table. To set the size of the SGA, set the `UNIFIED_AUDIT_SGA_QUEUE_SIZE` initialization parameter. The default size is 1 MB, and you can enter a range of `1` through `30`.

> **Note:**
>
> The `UNIFIED_AUDIT_SGA_QUEUE_SIZE` initialization parameter has been deprecated, but is currently retained for backward compatibility.

> **See Also:**
>
> *Oracle Database Reference* (../REFRN/GUID-060707DF-8431-4866-8B9F-4F450472D95E.htm#REFRN10343) for more information about the `UNIFIED_AUDIT_SGA_QUEUE_SIZE` initialization parameter

Setting the Write Mode for Unified Audit Trail Records

In a multitenant environment, the DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY procedure applies to the current pluggable database (PDB) only.

If the database is read-only, then DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY sets the value directly in the SGA.

1. Log in to SQL*Plus as a user who has been granted the AUDIT_ADMIN role.
   For example:

   ```
   sqlplus audit_admin Enter password: password
   ```

2. In a multitenant environment, connect to the appropriate PDB.
   For example:

   ```
   CONNECT audit_admin@hrpdb Enter password: password
   ```

   To find the available PDBs, query the DBA_PDBS data dictionary view. To check the current PDB, run the `show con_name` command.

3. Set the AUDIT_TRAIL_MODE property of the DBMS_AUDIT_MGMT package, as follows:

   - To use immediate-write mode, run the following procedure:

     ```
     BEGIN DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
     DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED, DBMS_AUDIT_MGMT.AUDIT_TRAIL_WRITE_MODE,
     DBMS_AUDIT_MGMT.AUDIT_TRAIL_IMMEDIATE_WRITE); END; /
     ```

   - To use queued-write mode, run the following procedure:

     ```
     BEGIN DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
     DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED, DBMS_AUDIT_MGMT.AUDIT_TRAIL_WRITE_MODE,
     DBMS_AUDIT_MGMT.AUDIT_TRAIL_QUEUED_WRITE); END; /
     ```

The settings take effect on subsequent database sessions.

Manually Flushing Audit Records to the Audit Trail in Queued-Write Mode

If you choose to use queued-write mode, then you can manually writing records to the unified audit trail.

Be aware that there is a minimum flush threshold that is configured by default. It checks if the queues were flushed more than 3 seconds before and accordingly flushes the audit records in the SGA queue. However, owing to the database activity, the flush may not happen every 3 seconds and could take longer.

- Writing Records to Disk for the Current Database Instance (audit_admin.htm#GUID-9A17031A-5350-4D69-AE88-F46CC88562A6)
- Writing Records to Disk Across an Oracle RAC Environment (audit_admin.htm#GUID-509A994F-9354-4A25-B169-7E6A0C5A62C4)
- Writing Records to Disk in a Multitenant Environment (audit_admin.htm#GUID-F5F69055-025F-4FCF-8A50-3A23ED74AD7D)

Writing Records to Disk for the Current Database Instance

You can manually write records to disk for the current database instance or the current Oracle Real Application Clusters (Oracle RAC) instance.

- To manually write records to disk, run either of the following procedures:

  ```
  EXEC DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL; EXEC
  DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL(DBMS_AUDIT_MGMT.FLUSH_CURRENT_INSTANCE);
  ```

Writing Records to Disk Across an Oracle RAC Environment

You can flush audit records across all Oracle Real Application Cluster (Oracle RAC) instances.

- Run the following procedure to flush audit records on Oracle RAC instances:

  ```
  EXEC DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL(DBMS_AUDIT_MGMT.FLUSH_ALL_INSTANCES);
  ```

Writing Records to Disk in a Multitenant Environment

In a multitenant environment, you can write the audit trail records to disk for the current PDB or across all PDBs in a multitenent environment.

- To write audit trail records to disk, use one of the following procedures:

  - For the current PDB:

    ```
    BEGIN DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL( CONTAINER =>
    DBMS_AUDIT_MGMT.CONTAINER_CURRENT); END; /
    ```

  - For all PDBs in the multitenant environment:

    ```
    BEGIN DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL( CONTAINER =>
    DBMS_AUDIT_MGMT.CONTAINER_ALL); END; /
    ```

Moving Operating System Audit Records into the Unified Audit Trail

Audit records can be written to external files in the `$ORACLE_BASE/audit/$ORACLE_SID` directory.

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.

You can load the files into the database by running the `DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES` procedure. Be aware that if you are moving a large number of operating system audit records to the unified audit trail, performance may be affected.

To move the audit records in these files to the `AUDSYS` schema audit table when the database is writable:

1. Log into the database instance as a user who has been granted the `AUDIT_ADMIN` role.
   For example:

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

In a multitenant environment, log into the PDB in which you want to move the audit trail records to the unified audit trail.

For example:

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

To find the available PDBs, query the `DBA_PDBS` data dictionary view. To check the current PDB, run the `show con_name` command.

2. Ensure that the database is open and writable.
   For a non-CDB architecture, to find the databases that are open and writable, query the `V$DATABASE` view.

   For example:

   ```
   SELECT NAME, OPEN_MODE FROM V$DATABASE; NAME OPEN_MODE --------------- ----------
   HRPDB READ WRITE
   ```

   In a multitenant environment, you can query the `V$PDBS` view to find information about PDBs associated with the current instance.

   For example:

   ```
   SELECT NAME, OPEN_MODE FROM V$PDBS; NAME OPEN_MODE --------------- ----------
   HRPDB READ WRITE
   ```

3. Run the `DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES` procedure.

   ```
   EXEC DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES;
   ```

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

Disabling Unified Auditing

You can disable unified auditing.

1. Disable any unified audit policies that are currently enabled.

   This step prevents the database from going into mixed mode auditing after you complete this procedure. See About Mixed Mode Auditing (auditing.htm#GUID-E9DB31BB-E90C-4B22-B29D-F9A44B350F9B) for more information about mixed mode auditing.

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

   b. Query the `POLICY_NAME` and `ENABLED_OPT` columns of the `AUDIT_UNIFIED_ENABLED_POLICIES` data dictionary view to find unified audit policies that are enabled.

c. Run the `NOAUDIT POLICY` statement to disable each enabled policy.

See Disabling Unified Audit Policies (audit_config.htm#GUID-1577CCEB-2AFD-417C-9238-9DFF13149766for more information.

2. Connect as user `SYS` with the `SYSOPER` privilege.

```
CONNECT sys as sysoper Enter password: password
```

In a multitenant environment, this command connects you to the root.

3. Shut down the database.

For example:

```
SHUTDOWN IMMEDIATE
```

In a multitenant environment, this command shuts down all PDBs in the CDB.

4. Depending on your platform, do the following:

- **UNIX systems:** Run the following commands:
```
cd $ORACLE_HOME/rdbms/lib make -f ins_rdbms.mk uniaud_off ioracle
```
- **Windows systems:** Rename the `%ORACLE_HOME%/bin/orauniaud12.dll` file to `%ORACLE_HOME%/bin/orauniaud12.dll.dbl`.

In a multitenant environment, these actions disable unified auditing in all PDBs in the CDB.

5. In SQL*Plus, restart the database.

```
STARTUP
```

In a multitenant environment, this command restarts all PDBs in the CDB.

Archiving the Audit Trail

You can archive the traditional operating system, unified database, and traditional database audit trails.

Topics:

- Archiving the Traditional Operating System Audit Trail (audit_admin.htm#GUID-04612F83-EC57-462C-A4F4-7562C4CAD8B5)
- Archiving the Unified and Traditional Database Audit Trails (audit_admin.htm#GUID-DACD856C-F3B2-40A1-9D48-882BBF438FCB)

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.

- 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. For more information, see *Oracle Audit Vault and Database Firewall Administrator's Guide* (http://www.oracle.com/pls/topic/lookup?ctx=E5052901&id=SIGAD40485).

  - **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 both to free audit trail space and to facilitate audit trail management.

> **See Also:**
>
> - Moving Operating System Audit Records into the Unified Audit Trail (audit_admin.htm#GUID-91B14CCD-E408-4669-82FA-EE833EDBBBA9)
> - Purging Audit Trail Records (audit_admin.htm#GUID-9B891A44-3DF4-4B52-98D4-A931DBAEC1D)

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 both frees audit trail space and facilitates 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. You install Oracle Audit Vault and Database Firewall separately from Oracle Database. For more information, see *Oracle Audit Vault and Database Firewall Administrator's Guide* (http://www.oracle.com/pls/topic/lookup?ctx=E5052901&id=SIGAD40485).

After you complete the archive, you can purge the database audit trail contents. See Purging Audit Trail Records (audit_admin.htm#GUID-9B891A44-3DF4-4B52-98D4-A931DBAEC1D) for more information.

- To archive the unified, traditional standard, and traditional fine-grained audit records, copy the relevant records to a normal database table.

  For example:

```
INSERT INTO table SELECT ... FROM UNIFIED_AUDIT_TRAIL ...; INSERT INTO table SELECT
... FROM SYS.AUD$ ...; INSERT INTO table SELECT ... FROM SYS.FGA_LOG$ ...;
```

Purging Audit Trail Records

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

Topics:

- About Purging Audit Trail Records (audit_admin.htm#GUID-60118D0C-75D7-4FF6-9965-2A23A2584E2A)
- Selecting an Audit Trail Purge Method (audit_admin.htm#GUID-5761FBDA-91F9-4E0F-82E5-B52F3022324F)
- Scheduling an Automatic Purge Job for the Audit Trail (audit_admin.htm#GUID-C1EB1E76-B15B-4450-A4C3-ED525E17CBCA)
- Manually Purging the Audit Trail (audit_admin.htm#GUID-B9D2B078-2567-4002-A906-E293DE94549E)
- Other Audit Trail Purge Operations (audit_admin.htm#GUID-91B17A5D-A702-4C9D-92C4-CF62A4BC4A5A)
- Example: Directly Calling a Unified Audit Trail Purge Operation (audit_admin.htm#GUID-532781DA-B568-4F37-84B7-D051D564D355)

> **See Also:**
>
> Managing the Unified Audit Trail (audit_admin.htm#GUID-2A5CFDFC-320B-41D0-86AB-B524992E02A8)

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, in most cases, 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 Audit Vault and Database Firewall installed, the audit trail purge process differs from the procedures described in this manual. For example, Oracle Audit Vault archives the audit trail for you. See *Oracle Audit Vault and Database Firewall Administrator's Guide* (http://www.oracle.com/pls/topic/lookup?ctx=E5052901&id=SIGAD40485).

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

> **See Also:**
>
> - *Oracle Database PL/SQL Packages and Types Reference* (../ARPLS/d_audit_mgmt.htm#ARPLS241)for more information about the `DBMS_AUDIT_MGMT` PL/SQL package
> - *Oracle Database Reference* (../REFRN/toc.htm) for detailed information about the `DBA_AUDIT_MGMT`-related views

Selecting an Audit Trail Purge Method

How you select an audit trail purge method depends on whether you want perform the purge on a regularly scheduled basis or run it at a specified times.

Topics:

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. If necessary, tune online and archive redo log sizes 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.
   See Scheduling an Automatic Purge Job for the Audit Trail (audit_admin.htm#GUID-C1EB1E76-B15B-4450-A4C3-ED525E17CBCA) for more information.

Manually Purging the Audit Trail at a Specific Time

You can manually purge the audit records right away in a one-time operation, rather than creating a purge schedule.

1. If necessary, tune online and archive redo log sizes 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. Run the purge operation.
   See Manually Purging the Audit Trail (audit_admin.htm#GUID-B9D2B078-2567-4002-A906-E293DE94549E) for more information.

Scheduling an Automatic Purge Job for the Audit Trail

When you schedule an automatic purge job, you need to tune the online and archive redo log sizes, plan a timestamp and archive strategy, optionally set an archive timestamp, and then create and schedule the purge job.

Topics:

- Step 3: Optionally, Set an Archive Timestamp for Audit Records (audit_admin.htm#GUID-1C9053BB-30B8-48DB-9062-44DD42A75B4E)
- Step 4: Create and Schedule the Purge Job (audit_admin.htm#GUID-07ECB9A9-369A-4668-A0BC-D29CDB367ABF)

About Scheduling an Automatic Purge Job

You can purge the entire audit trail, or only a portion of the audit trail that was created before a timestamp.

The individual audit records created before the timestamp can be purged.

Be aware that purging the audit trail, particularly a large one, can take a while to complete. Consider scheduling the purge job so that it runs during a time when the database is not busy.

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. In addition, be aware that the jobs created by the `DBMS_SCHEDULER` PL/SQL package do not execute 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.

Step 1: If Necessary, Tune Online and Archive Redo Log Sizes

The purge process may generate additional redo logs.

- If necessary, tune online and archive redo log sizes to 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.

> **See Also:**
>
> *Oracle Database Administrator's Guide* (../ADMIN/onlineredo.htm#ADMIN007)for more information about tuning log files

Step 2: Plan a Timestamp and Archive Strategy

You must record the timestamp of the audit records before you can archive them.

- To find the timestamp date, query the `DBA_AUDIT_MGMT_LAST_ARCH_TS` data dictionary view.

Later on, when the purge takes place, Oracle Database purges only the audit trail records that were created before the date of this timestamp. See Step 3: Optionally, Set an Archive Timestamp for Audit Records (audit_admin.htm#GUID-1C9053BB-30B8-48DB-9062-44DD42A75B4E)

After you have timestamped the records, you are ready to archive them. See Archiving the Audit Trail (audit_admin.htm#GUID-58CFB1F4-7E76-46A4-8DDD-2546DBD67FBE)for more information.

Step 3: 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 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. If you want to clear the archive timestamp setting, see Clearing the Archive Timestamp Setting (audit_admin.htm#GUID-4AE3F98A-BB9E-437B-BCEB-4DA626F2302E)

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.

To set the timestamp for the purge job:

1. Log into the database instance as a user who has been granted the `AUDIT_ADMIN` role.
   In a multitenant environment, log into 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. 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 traditional audit data from previous releases:

     - `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.

- 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.

- RAC_INSTANCE_NUMBER specifies the instance number for an Oracle RAC installation. 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 0, which is used for single-instance database installations. You can find the instance number by issuing the SHOW PARAMETER INSTANCE_NUMBER command in SQL*Plus.

- CONTAINER applies the timestamp to a multitenant environment. 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, and DBMS_MGMT.CONTAINER_CURRENT only from a PDB.

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.

Step 4: 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:

  - 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 audit trail records, that is, both database audit trail and operating system audit trail types. (This setting does not apply to read-only databases.)

- `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 have been deleted.

- `CONTAINER` is used for a multitenant environment to define where to create the purge job. If you set `CONTAINER` to `DBMS_AUDIT_MGMT.CONTAINER_CURRENT`, then it is available, visible, and managed only from the current PDB. The `DBMS_AUDIT_MGMT.CONTAINER_ALL` setting creates the job in the root. This defines the job as 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 root, then it is visible only in the root. Hence, you can enable, disable, and drop it from the root only.

Manually Purging the Audit Trail

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

Topics:

- About Manually Purging the Audit Trail (audit_admin.htm#GUID-2663202E-F69F-4635-811A-5217F6C66BF0)
- Using DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL to Manually Purge the Audit Trail (audit_admin.htm#GUID-7D2E55CE-2820-47E3-A989-5F7AAB19CD76)

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` 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 acknowledgement 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` initialization parameter).

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. Follow these steps under Scheduling an Automatic Purge Job for the Audit Trail (audit_admin.htm#GUID-C1EB1E76-B15B-4450-A4C3-ED525E17CBCA):
   - Step 1: If Necessary, Tune Online and Archive Redo Log Sizes (audit_admin.htm#GUID-2F1AAC07-30A4-4B3A-B9E2-ADEC02E70533)
   - Step 2: Plan a Timestamp and Archive Strategy (audit_admin.htm#GUID-F5B4C780-584A-4555-9608-1F5F9D62C7CA)
   - Step 3: Optionally, Set an Archive Timestamp for Audit Records (audit_admin.htm#GUID-1C9053BB-30B8-48DB-9062-44DD42A75B4E)
2. If you are using a multitenant environment, then connect to the database 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.

   For example:

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

3. 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:

  - 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.)

- USE_LAST_ARCH_TIMESTAMP: Enter either of the following settings:

  - TRUE: Deletes audit records created before the last archive timestamp. To set the archive timestamp, see Step 3: Optionally, Set an Archive Timestamp for Audit Records (audit_admin.htm#GUID-1C9053BB-30B8-48DB-9062-44DD42A75B4E) 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 have been deleted.

- CONTAINER: Applies the cleansing to a multitenant environment. DBMS_AUDIT_MGMT.CONTAINER_CURRENT specifies the local PDB; DBMS_AUDIT_MGMT.CONTAINER_ALL applies to all databases.

Other Audit Trail Purge Operations

Other kinds of audit trail purge include enabling or disabling the audit trail purge job, setting the default audit trail purge job interval, deleting an audit trail purge job, and clearing an archive timestamp setting.

Topics:

- Enabling or Disabling an Audit Trail Purge Job (audit_admin.htm#GUID-34AE7B90-11F2-42FD-9D80-8E411AA4406A)
- Setting the Default Audit Trail Purge Job Interval for a Specified Purge Job (audit_admin.htm#GUID-2D0E56B0-A17E-441A-8F40-321C0E594D51)
- Deleting an Audit Trail Purge Job (audit_admin.htm#GUID-B5DD2078-5EFD-4979-BE22-643D4EC028EB)

- Clearing the Archive Timestamp Setting (audit_admin.htm#GUID-4AE3F98A-BB9E-437B-BCEB-4DA626F2302E)

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.

In a multitenant environment, where you run the DBMS_AUDIT_MGMT.SET_PURGE_JOB_STATUS 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 (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.

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.

In a multitenant environment, 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.

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; /
```

In a multitenant environment, where you run the DBMS_AUDIT_MGMT.DROP_PURGE_JOB 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.DROP_PURGE_JOB_INTERVAL procedure from the PDB in which it was created.

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 and for a multitenant environment, the container 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`.

- `CONTAINER` applies the timestamp to a multitenant environment. `DBMS_AUDIT_MGMT.CONTAINER_CURRENT` specifies the local PDB; `DBMS_AUDIT_MGMT.CONTAINER_ALL` applies to all databases.

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 23-1 (audit_admin.htm#GUID-532781DA-B568-4F37-84B7-D051D564D355__BCGJGCHA) 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 23-1 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 */ /
```

If you want to modify this example for other audit trail types, be aware that additional steps may be required. For more information, see the Oracle Database 11*g* Release 2 (11.2) version of *Oracle Database Security Guide*, which is available from the following documentation library:

`http://www.oracle.com/pls/db112/homepage` (http://www.oracle.com/pls/db112/homepage)

Audit Trail Management Data Dictionary Views

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

Table 23-1 (audit_admin.htm#GUID-2AC6F69F-5BEA-4AC7-9A22-2966121A43ED__CDCJJEFD) lists these views.

Table 23-1 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: `DELETE FROM DBA_AUDIT_MGMT_CLEAN_EVENTS;` This view applies to read-write databases only. For read-only databases, a history of purge events is in the alert log. 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