89 DBMS_FLASHBACK_ARCHIVE

The `DBMS_FLASHBACK_ARCHIVE` package contains procedures for performing various flashback archive tasks.

These include:

- Disassociation and reassociation of a Flashback Archive enabled table from/with its underlying Flashback Time Travel
- Tamper-proofing the tables of an application
- Importing of user history

  Caution:

  Importing user-generated history can lead to inaccurate, or unreliable results. This procedure should only be used after consulting with Oracle Support.

- Enabling and disabling of session-level support for valid-time

See Also:

Oracle Database Development Guide for more information about Using Flashback Time Travel

This chapter contains the following topics:

- Overview
- Security Model
- Constants
- Summary of DBMS_FLASHBACK_ARCHIVE Subprograms

DBMS_FLASHBACK_ARCHIVE Overview

Flashback Time Travel provides strict protection on the internal history tables that it creates and maintains for users.

Caution:

The `DBMS_FLASHBACK_ARCHIVE` package does not support migration between databases. It allows you to move Flashback Time Travel history within the same database only, and not outside it. This package does not allow you to move Flashback Time Travel table data. It allows you to import customer history only and not Flashback Time Travel history.

The read-only semantics prohibit users, including a DBA, from doing updates, deletes, and inserts on the Flashback Time Travel internal history tables. Users are also prevented from issuing any DDL statements on these tables. This strict security enforcement helps meet the requirements of applications in regulatory / compliance environments. Flashback Time Travel supports most common DDL statements, including those that alter the table definition or incur data movement. However, some DDL statements are not supported on Flashback Archive enabled tables. Since most application schemas are modified during application software upgrades, the ability to perform DDL operations on tracked tables is critical.

To support schema evolution during application upgrades and other table maintenance tasks that require use of DDL statements not supported by Flashback Time Travel, the `DBMS_FLASHBACK_ARCHIVE` package provides a set of simple-to-use PL/SQL procedures:

- To disassociate a Flashback Archive enabled base table from the underlying Flashback Time Travel
- To reassociate a temporarily disassociated base table with its underlying Flashback Time Travel

After a user has disassociated the base table from its Flashback Archive, it's possible to issue any DDL statements on the base table or the history tables in the Flashback Archive. Having finished with the schema changes, the user can then reassociate the base table with its Flashback Archive so that Flashback Time Travel protection is in operation and automatic tracking and archiving is resumed.

DBMS_FLASHBACK_ARCHIVE Security Model

Users need the `FLASHBACK_ARCHIVE_ADMINISTER` privilege to import user-generated history, to set context level, and to tamper-proof tables. After a table is disassociated, users can perform DDL and DML statements on the table if they have the necessary privileges. Enabling and disabling session-level Valid Time Temporal flashback needs no additional privileges.

DBMS_FLASHBACK_ARCHIVE Constants

The `DBMS_FLASHBACK_ARCHIVE` package uses the constants shown in the following table.

Table 89-1 DBMS_FLASHBACK_ARCHIVE Constants

Constant	Type	Value	Description
NODROP	BINARY_INTEGER	1	Do not drop temporary history table
NOCOMMIT	BINARY_INTEGER	2	Do not commit transaction
NODELETE	BINARY_INTEGER	4	Do not delete data in history table

Summary of DBMS_FLASHBACK_ARCHIVE Subprograms

This table lists the `DBMS_FLASHBACK_ARCHIVE` subprograms and briefly describes them.

Table 89-2 DBMS_FLASHBACK_ARCHIVE Package Subprograms

Subprogram	Description
ADD_TABLE_TO_APPLICATION Procedure	Takes an application name and adds a table to the application as a security table
CREATE_TEMP_HISTORY_TABLE Procedure	Creates a table called `TEMP_HISTORY` with the correct definition in schema
DISABLE_APPLICATION Procedure	Takes an application name and marks a table in it as a security table
DISABLE_ASOF_VALID_TIME Procedure	Disables session level valid-time flashback
DISASSOCIATE_FBA Procedure	Disassociates the given table from the Flashback Time Travel
DROP_APPLICATION Procedure	Takes an application name and removes it from the list of applications
ENABLE_APPLICATION Procedure	Takes an application name and enables Flashback Time Travel on all the security tables for this application
ENABLE_AT_VALID_TIME Procedure	Enables session level valid time flashback
EXTEND_MAPPINGS Procedure	Extends time mappings to times in the past
GET_CURRENT_LIFESPAN_DIGEST Function	Generates the current lifespan digest for the specified row in a user table
GET_SYS_CONTEXT Function	Gets the context previously selected by the SET_CONTEXT_LEVEL Procedure
IMPORT_HISTORY Procedure	Imports history from a table called `TEMP_HISTORY` in the given schema.
LOCK_DOWN_APPLICATION Procedure	Takes an application name and makes all the security tables read-only. The group called `SYSTEM` cannot be locked
PURGE_CONTEXT Procedure	Purges the context to be saved selected by the SET_CONTEXT_LEVEL Procedure
REASSOCIATE_FBA Procedure	Reassociates the given table with the Flashback Time Travel
REGISTER_APPLICATION Procedure	Takes an application name and optionally a Time Travel, and registers an application for database hardening
REMOVE_TABLE_FROM_APPLICATION Procedure	Takes an application name and marks a table in it as no longer being a security table

Table 89-2 (Cont.) DBMS_FLASHBACK_ARCHIVE Package Subprograms

Subprogram	Description
SET_CONTEXT_LEVEL Procedure	Defines how much of the user context is to be saved
VERIFY_BLOCKCHAIN_LIFESPAN Procedure	Verifies the content of the current, historical, or all lifespans of user table rows

ADD_TABLE_TO_APPLICATION Procedure

This procedure takes an application name and adds a table to the application as a security table. If the application is enabled for Flashback Time Travel, then this table will also be enabled for Flashback Time Travel.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.ADD_TABLE_TO_APPLICATION (
   application_name           IN    VARCHAR2,
   table_name                 IN    VARCHAR2,
   schema_name                IN    VARCHAR2 := NULL);
```

Parameters

Table 89-3 ADD_TABLE_TO_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application for which a table has been added as a security table
`table_name`	Name of the table to add as a security table for the given application
`schema_name`	Name of the schema containing the desired table. If no schema name is specified, the current schema is used.

CREATE_TEMP_HISTORY_TABLE Procedure

This procedure creates a table called `TEMP_HISTORY` with the correct definition in schema.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.CREATE_TEMP_HISTORY_TABLE (
   owner_name1       IN    VARCHAR2,
   table_name1       IN    VARCHAR2);
```

Parameters

Table 89-4 CREATE_TEMP_HISTORY_TABLE Procedure Parameters

Parameter	Description
`owner_name1`	Schema of the Flashback Time Travel-enabled table
`table_name1`	Name of the Flashback Time Travel-enabled table

DISABLE_APPLICATION Procedure

This procedure takes an application name and disables Flashback Time Travel on all of its security tables.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.DISABLE_APPLICATION (
   application_name           IN   VARCHAR2);
```

Parameters

Table 89-5 DISABLE_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application whose security tables will be disabled for Flashback Time Travel

DISABLE_ASOF_VALID_TIME Procedure

This procedure disables session level valid-time flashback.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.DISABLE_ASOF_VALID_TIME;
```

DISASSOCIATE_FBA Procedure

This procedure disassociates the given table from the Flashback Time Travel.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.DISASSOCIATE_FBA (
   owner_name     IN    VARCHAR2,
   table_name     IN    VARCHAR2);
```

Parameters

Table 89-6 DISASSOCIATE_FBA Procedure Parameters

Parameter	Description
`owner_name`	Schema of the Flashback Time Travel enabled base table
`table_name`	Name of the Flashback Time Travel enabled base table

Exceptions

Table 89-7 DISASSOCIATE_FBA Procedure Exceptions

Exception	Description
`ORA-55602`	User table is not enabled for Flashback Time Travel
`ORA-55634`	Cannot acquire the lock on the table for disassociation

DROP_APPLICATION Procedure

This procedure takes an application name and removes it from the list of applications. As part of this procedure, Flashback Time Travel will be disabled on all security-enabled tables and all history data will be lost. The group called `SYSTEM` cannot be dropped.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.DROP_APPLICATION (
   application_name           IN   VARCHAR2);
```

Parameters

Table 89-8   DROP_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application for which a table has been added as a security table

ENABLE_APPLICATION Procedure

This procedure takes an application name and enables Flashback Time Travel on all the security tables for this application. Once an application is enabled, every change to an Flashback Time Travel enabled table will be tracked.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.ENABLE_APPLICATION (
   application_name           IN   VARCHAR2);
```

Parameters

Table 89-9   ENABLE_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application for which to enable Flashback Time Travel on all its security tables

ENABLE_AT_VALID_TIME Procedure

This procedure enables session level valid time flashback.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME (
   level          IN    VARCHAR2,
   query_time     IN    TIMESTAMP DEFAULT SYSTIMESTAMP);
```

Parameters

Table 89-10 ENABLE_AT_VALID_TIME Procedure Parameters

Parameter	Description
`level`	Options: • `All` - Sets the visibility of temporal data to the full table, which is the default temporal table visibility • `CURRENT` - Sets the visibility of temporal data to currently valid data within the valid time period at the session level • `ASOF` - Sets the visibility of temporal data to data valid as of the given time as defined by the timestamp
`query_time`	Used only if level is `ASOF`. Data which is valid at this `query_time` will only be shown.

EXTEND_MAPPINGS Procedure

This procedure extends time mappings to times in the past.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.EXTEND_MAPPINGS;
```

GET_CURRENT_LIFESPAN_DIGEST Function

This function generates the current lifespan digest for the specified row in a user table.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.GET_CURRENT_LIFESPAN_DIGEST (
            rid          IN   VARCHAR2,
            start_scn    IN   RAW,
            xid          IN   RAW,
            op           IN   VARCHAR2,
            objno        IN   NUMBER)
RETURN RAW;
```

Parameters

Table 89-11 GET_CURRENT_LIFESPAN_DIGEST Function Parameters

Parameter	Description
`rid`	The ROWID of the user table row.
`start_scn`	The latest start SCN of the row.
`xid`	The latest transaction ID of the row.
`op`	The latest operation that is performed on the row.
`objno`	The table's object number.

The cryptographic hash is generated over all column values in the ascending order of the column IDs. Each column value is prefixed by column metadata. For more information, refer to the Appendix B.1 Column Content in Blockchain Tables section of the *Database Administrator's Guide*.

- Each row of the user table transitions through a series of "lifespans" created by transactions that commit changes to the row. The most recent lifespan is referred to as the "current lifespan," and all other previous lifespans are referred to as "historical lifespans."
- The historical lifespans are recorded into a Flashback Archive (FBA) internal table: `SYS_FBA_HIST_<objno>`.
- The current lifespans are always present in the user table. However, the relevant metadata of current lifespans are recorded in another FBA internal table: `SYS_FBA_TCRV_<objno>`.
- This API can be used to compute the current lifespan digest of the specific row of a user table using its associated metadata from the FBA internal table: `SYS_FBA_TCRV_<objno>`. A user can record this digest for detecting corruptions, if any, in the row data.

GET_SYS_CONTEXT Function

This function gets the context previously selected by the SET_CONTEXT_LEVEL Procedure.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT (
   xid             IN    RAW,
   namespace       IN    VARCHAR2,
   parameter       IN    VARCHAR2)
 RETURN VARCHAR2;
```

Parameters

Table 89-12 GET_SYS_CONTEXT Function Parameters

Parameter	Description
`xid`	Transaction identifier is an opaque handle to a transaction obtained from the versions query
`namespace`	Namespace
`parameter`	If undefined, the subprogram returns `NULL`

Related Topics

- SET_CONTEXT_LEVEL Procedure
  This procedure defines how much of the user context is to be saved.

IMPORT_HISTORY Procedure

This procedure is called after invoking the CREATE_TEMP_HISTORY_TABLE procedure, and after the `TEMP_HISTORY` table is populated with user-generated history data

Caution:

Importing user-generated history can lead to inaccurate, or unreliable results. This procedure should only be used after consulting with Oracle Support.

The `DBMS_FLASHBACK_ARCHIVE` package does not support migration between databases. It allows you to move Flashback Time Travel history within the same database only, and not outside it. This package does not allow you to move Flashback Time Travel table data. It allows you to import customer history only and not Flashback Time Travel history.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.IMPORT_HISTORY (
   owner_name1          IN   VARCHAR2,
   table_name1          IN   VARCHAR2
   temp_history_name    IN   VARCHAR2 DEFAULT 'TEMP_HISTORY',
   options              IN   BINARY_INTEGER DEFAULT 0);
```

Parameters

Table 89-13 IMPORT_HISTORY Procedure Parameters

Parameter	Description
`owner_name1`	Schema of the Flashback Time Travel-enabled table
`table_name1`	Name of the Flashback Time Travel-enabled table
`temp_history_name`	Optional temporary history table from which we import history data
`options`	One (or a combination) of constants (`NODROP`, `NOCOMMIT`, and `NODELETE`) to specify if we want to drop, commit changes of, or truncate the temporary history table

Usage Notes

The database function `TIMESTAMP_TO_SCN` can be used to convert times to SCN when populating the temporary history table.

Related Topics

- CREATE_TEMP_HISTORY_TABLE Procedure
  This procedure creates a table called `TEMP_HISTORY` with the correct definition in schema.

LOCK_DOWN_APPLICATION Procedure

This procedure takes an application name and makes all the security tables read-only. The group called `SYSTEM` cannot be locked.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.LOCK_DOWN_APPLICATION (
   application_name           IN    VARCHAR2);
```

Parameters

Table 89-14 LOCK_DOWN_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application for which a table has been added as a security table

PURGE_CONTEXT Procedure

This procedure purges the context to be saved selected by the SET_CONTEXT_LEVEL Procedure.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.PURGE_CONTEXT;
```

Related Topics

- SET_CONTEXT_LEVEL Procedure
  This procedure defines how much of the user context is to be saved.

REASSOCIATE_FBA Procedure

This procedure reassociates the given table with the Flashback Time Travel.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.REASSOCIATE_FBA (
   owner_name      VARCHAR2,
   table_name      VARCHAR2);
```

Parameters

Table 89-15 REASSOCIATE_FBA Procedure Parameters

Parameter	Description
`owner_name`	Schema of the Flashback Time Travel enabled base table
`table_name`	Name of the Flashback Time Travel enabled base table

Exceptions

Table 89-16 REASSOCIATE_FBA Procedure Exceptions

Parameter	Description
`ORA-55602`	User table is not enabled for Flashback Time Travel
`ORA-55636`	Table definition validation failed

Usage Notes

- The procedure will signal an error if the base table and the history table do not have identical data definitions. For example, when columns are added or table is split, the resulting base table and history table need to have the same schema.
- The Flashback Time Travel internal history table schema has some row versions metadata columns. The procedure will signal an error if any metadata column is dropped by users.

REGISTER_APPLICATION Procedure

This procedure takes an application name and optionally a Flashback Archive, and registers an application for database hardening.

When database hardening is enabled, then all the security tables for that application are enabled for Flashback Archive using the given Flashback Archive. If no Flashback Archive is specified, the default Flashback Archive is used.

See Also:

Using Flashback Time Travel in *Oracle Database Development Guide* regarding database hardening

Syntax

```
DBMS_FLASHBACK_ARCHIVE.REGISTER_APPLICATION (
   application_name           IN   VARCHAR2,
   flashback_archive_name     IN    VARCHAR2 := NULL);
```

Parameters

Table 89-17 REGISTER_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application which is being registered. The application `SYSTEM` is already registered when the package is created and is populated with list of tables needed for database hardening.
`flashback_archive_name`	Name of the Flashback Archive in which the historical data for the security tables for given application is stored. If no Flashback Archive is specified, the default Flashback Archive is used.

REMOVE_TABLE_FROM_APPLICATION Procedure

This procedure takes an application name and marks a table in it as no longer being a security table.

If the application is already enabled for Flashback Archive, Flashback Archive will be disabled for this table.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.REMOVE_TABLE_TO_APPLICATION (
   application_name           IN   VARCHAR2,
   table_name                 IN   VARCHAR2,
   schema_name                IN   VARCHAR2 := NULL);
```

Parameters

Table 89-18 REMOVE_TABLE_FROM_APPLICATION Procedure Parameters

Parameter	Description
`application_name`	Name of the application for which a table is being removed from the list of security tables
`table_name`	Name of the table to mark as being no longer a security table for the given application
`schema_name`	Name of the schema containing the desired table. If no schema name is specified, the current schema is used.

SET_CONTEXT_LEVEL Procedure

This procedure defines how much of the user context is to be saved.

Syntax

```
DBMS_FLASHBACK_ARCHIVE.SET_CONTEXT_LEVEL (
   level           VARCHAR2);
```

Parameters

Table 89-19 SET_CONTEXT_LEVEL Procedure Parameters

Parameter	Description
`level`	Depending on how much of the user context needs to be saved: • `ALL` - the entire `SYS_CONTEXT` • `TYPICAL` - the user ID, global user ID and the hostname • `NONE` - nothing

VERIFY_BLOCKCHAIN_LIFESPAN Procedure

This procedure verifies the contents of the current lifespans, historical lifespans, or all lifespans of the user table rows that are protected using blockchain Flashback Archive (FBA). The verification is done by comparing the row's cryptographic hash with the ORAFBA_CURRENT_LIFESPAN_DIGEST$ column value recorded in its previous historical lifespan.

Syntax

```
PROCEDURE verify_blockchain_lifespan(
          schema_name                     IN  VARCHAR2,
          table_name                      IN  VARCHAR2,
          number_of_lifespan_verified    OUT NUMBER,
          type_of_lifespan                IN  NUMBER DEFAULT LIFESPAN_ALL);
```

Parameters

Table 89-20 VERIFY_BLOCKCHAIN_LIFESPAN Procedure Parameters

Parameter	Description
schema_name	The user schema name.
table_name	The user table name.
number_of_lifespan_verified	The number of lifespan rows verified.
type_of_lifespan	The type of lifespan to be verified. The following are the constants for the type_of_lifespan parameter. • LIFESPAN_CURRENT CONSTANT NUMBER := 1; • LIFESPAN_HISTORICAL CONSTANT NUMBER := 2; • LIFESPAN_ALL CONSTANT NUMBER := 3;