169
DBMS_ROLLING

The `DBMS_ROLLING` PL/SQL package is used to implement the Rolling Upgrade Using Active Data Guard feature.

The Active Data Guard feature of Oracle Data Guard streamlines the process of upgrading Oracle Database software in a Data Guard configuration in a rolling fashion. The Rolling Upgrade Using Active Data Guard feature requires a license for the Oracle Active Data Guard option. You can use this feautre for database release upgrades starting with the first patchset of Oracle Database 12c.

Starting with Oracle Database 23ai, Transaction Guard works during `DBMS_ROLLING` operations to ensure continuous application functions during switchover, issued by `DBMS_ROLLING` to Transient Logical Standby. Transaction Guard returns the commit outcome of the current in-flight transaction when an error or outage occurs. Applications embed the Transaction Guard APIs in their error handling procedures to ensure that work continues without any in-flight work lost or duplicate submissions after an outage.

Additionally, you can use this feature immediately for other database maintenance tasks. The database where maintenance is performed must be operating at a minimum of Oracle Database 12c Release 1 (12.1). Such maintenance tasks include:

- Adding partitioning to non-partitioned tables
- Changing BasicFiles LOBs to SecureFiles LOBs
- Changing `XMLType` stored as `CLOB` to `XMLtype` stored as binary XML
- Altering tables to be OLTP-compressed

This chapter contains the following topics:

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for information about using `DBMS_ROLLING` to perform a rolling upgrade

DBMS_ROLLING Overview

The `DBMS_ROLLING` PL/SQL package provides procedures that you can use to perform any change throughout a Data Guard configuration in a rolling fashion.
These procedures include a rolling upgrade of the Oracle Database software. Although the focus of this document is rolling upgrade operations, the content is applicable to the deployment of any rolling changes.

Starting with Oracle Database 23ai, Transaction Guard works during DBMS_ROLLING operations to ensure continuous application functions during switchover, issued by DBMS_ROLLING to Transient Logical Standby.

All the procedures are executed at the current primary database, which eliminates the potential confusion of moving between remote databases to perform various operations related to the rolling upgrade. If necessary, all the procedures can be called again to resume the rolling upgrade after an error or interruption. (The upgrade script must still be run at the standby.)

The package also provides a procedure that allows you to return a Data Guard configuration back to its original, pre-upgrade state in the event users wish to abandon the rolling upgrade.

The actual execution of a rolling upgrade has been reduced to three steps (excluding the upgrade of the Oracle Database software itself and the on-disk setup of the new Oracle Database software). The number of steps remains the same regardless of the size of the Data Guard configuration.

Conceptually, for the purposes of the DBMS_ROLLING package, you divide your Data Guard configuration into two groups: the leading group and the trailing group. The databases in the leading group undergo the upgrade operation (or any other change that you are deploying) first. The databases in the trailing group undergo the upgrade of the Oracle Database software (or any other change that you are deploying) only after the switchover operation. This insulates them from the upgrade and gives you time to evaluate the effect of the change in the leading group databases.

Each group has a master database: the future primary database as specified in the DBMS_ROLLING.INIT_PLAN procedure is the master of the leading group, called Leading Group Master (LGM), while the original primary database is the master of the trailing group called Trailing Group Master (TGM). You can configure databases to protect the LGM and the TGM. Standbys designated to protect the LGM are referred to as Leading Group Standbys (LGS). Standbys designated to protect the TGM are referred to as Trailing Group Standbys (TGS). These terms are used throughout this documentation.

If you perform a rolling upgrade using a transient logical standby database (including using DBMS_ROLLING), then you must manually load audit records from the operating system spillover location before the switchover. Before you run DBMS_ROLLING.SWITCHOVER, run DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES on the trailing group master (current primary database).

DBMS_ROLLING Security Model

The DBMS_ROLLING package is available to users who have been granted the DBA role.

Summary of DBMS_ROLLING Subprograms

This table lists and briefly describes the DBMS_ROLLING package subprograms.

Table 169-1 DBMS_ROLLING Package Subprograms

Subprogram	Description
INIT_PLAN Procedure	Initializes a rolling operation plan with system-generated default values.
DESTROY_PLAN Procedure	Destroys any existing rolling operation plan, its parameters, and all resources associated with the rolling operation.

Table 169-1 (Cont.) DBMS_ROLLING Package Subprograms

Subprogram	Description
BUILD_PLAN Procedure	Validates plan parameters and creates or modifies a rolling operation plan.
SET_PARAMETER Procedure	Modifies a rolling operation parameter.
START_PLAN Procedure	Starts the rolling operation.
SWITCHOVER Procedure	Performs a switchover between the current primary database and the transient logical standby database.
FINISH_PLAN Procedure	Finalizes the rolling operation.
ROLLBACK_PLAN Procedure	Completely rolls back the rolling operation.

INIT_PLAN Procedure

This procedure initializes a rolling operation plan with system-generated default values.

Syntax

```
DBMS_ROLLING.INIT_PLAN (
   future_primary   IN VARCHAR2);
```

Parameters

Table 169-2 *INIT_PLAN Procedure Parameters*

Parameter	Description
future_primary	DB_UNIQUE_NAME of the future primary (also known as the Leading Group Master (LGM))

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45401: upgrade plan is already active
- ORA-45402: LOG_ARCHIVE_CONFIG must contain the DG_CONFIG attribute
- ORA-45403: database %s must be specified in DG_CONFIG
- ORA-45411: operation requires additional arguments
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- This procedure communicates with all databases defined in the log_archive_config init.ora parameter and validates that each database is a valid participant in the rolling upgrade. Valid participants other than the primary database must have a database role of either physical standby, DG, or logical standby.
- The designated future primary must be a physical standby or ADG.

DESTROY_PLAN Procedure

This procedure destroys any existing upgrade plan, its parameters, and all resources associated with a rolling operation.

Syntax

```
DBMS_ROLLING.DESTROY_PLAN ();
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-45422: operation requires existing plan
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- When a rolling operation is complete, this procedure can be called to completely purge all states related to a rolling operation.
- This procedure must also be called after a ROLLBACK_PLAN to purge the metadata.

BUILD_PLAN Procedure

This procedure validates plan parameters and creates or modifies a rolling operation plan.

A successfully constructed plan is required in order to perform a rolling operation. This procedure must return successfully before the START_PLAN procedure can be called to start the rolling operation. Parameter changes made after a plan has been created may require calling the BUILD_PLAN procedure to modify the existing plan. The DBA_ROLLING_EVENTS view will indicate if any invocation of the SET_PARAMETER procedure requires a plan rebuild. Failure to rebuild the plan will result in an ORA-45416 error when attempting to resume the rolling operation.

Syntax

```
DBMS_ROLLING.BUILD_PLAN ();
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45403: database %s must be specified in the DG_CONFIG
- ORA-45414: could not connect to a remote database
- ORA-45419: DB_UNIQUE_NAME parameter must be specified
- ORA-45433: failover was detected on an unsupported database
- ORA-45434: multiple failovers of the same type detected

- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- This procedure connects to databases specified as plan parameters. These instances must be mounted or open, and must be reachable via the network.

SET_PARAMETER Procedure

This procedure modifies a rolling operation parameter.

Starting with Oracle Database 20c, a new parameter called BLOCK_UNSUPPORTED is added to the DBMS_ROLLING.SET_PARAMETER procedure. By default, BLOCK_UNSUPPORTED is set to 1, indicating that operations performed on tables that are unsupported by Transient Logical Standby will be blocked on the primary database. If set to 0, then the DBMS_ROLLING package does not block operations on unsupported tables. Those tables will not be maintained by Transient Logical Standby, and will diverge from the primary database.

Syntax

```
DBMS_ROLLING.SET_PARAMETER (
   scope             IN VARCHAR2 DEFAULT NULL,
   name              IN NUMBER,
   value             IN VARCHAR2);
```

Parameters

Table 169-3 *SET_PARAMETER Procedure Parameters*

Parameter	Description
scope	Parameter scope. It can either be NULL for global parameters, or the DB_UNIQUE_NAME of a specific database for local parameters.
name	The DBMS_ROLLING constant for a given parameter.
value	New value for the parameter or NULL to revert to a default value.

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45408: parameter name is unknown
- ORA-45409: parameter value is invalid or out of bounds
- ORA-45410: parameter may not be modified
- ORA-45411: operation requires additional arguments
- ORA-45412: parameter scope argument is unknown
- ORA-45413: parameter has no default value
- ORA-45414: could not connect to a remote database
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- Changes to a parameter value may require a call to the DBMS_ROLLING.BUILD_PLAN procedure to modify the existing plan. Users should check the DBA_ROLLING_EVENTS view after setting a parameter to determine if a rebuild is necessary.
- Table 169-4 lists all the available parameters and their descriptions. The parameter names and values described in the table are all of type VARCHAR2.
- The MINVAL and MAXVAL columns in the DBA_ROLLING_PARAMETERS view identify the valid range of values for a parameter. The view does not contain any parameters until the DBMS_ROLLING.INIT_PLAN procedure has been successfully invoked.

Table 169-4 Valid Values for DBMS_ROLLING.SET_PARAMETER Procedure

Parameter Name	Global?	Description	Default
ACTIVE_SESSIONS_TIMEOUT	Yes	The maximum amount of time in seconds to enforce ACTIVE_SESSIONS_WAIT before halting the rolling upgrade. This parameter is only valid if ACTIVE_SESSIONS_WAIT is set to 1.	3600
ACTIVE_SESSIONS_WAIT	Yes	Whether the switchover operation will wait for active sessions to finish. If set to 1, the SWITCHOVER procedure waits for active sessions to compete. If set to 0, the SWITCHOVER procedure kills active sessions to expedite the switchover.	0
BACKUP_CONTROLFILE	Yes	File name of the backup control file that is created during a rolling upgrade.	rolling_change_backup.f
BLOCK_UNSUPPORTED	Yes	This parameter can be set for Rolling Upgrade to Block Unsupported Operations on the source database. Possible values are 0 and 1. • 1: Turns blocking on. This is the default value. • 0: Turns blocking off. Scope is not considered for this parameter.	1
DGBROKER	Yes	Use Data Guard broker for managing apply, recovery, and log archive destinations.	1 if broker is enabled, 0 otherwise.
DICTIONARY_LOAD_TIMEOUT	Yes	The maximum amount of time in seconds to enforce DICTIONARY_LOAD_WAIT before halting the rolling upgrade. This parameter is only valid if DICTIONARY_LOAD_WAIT is set to 1.	3600
DICTIONARY_LOAD_WAIT	Yes	Whether the instantiation of the transient logical standby will include a wait for the complete loading of the data dictionary snapshot in redo. If set to 1, then the START_PLAN procedure will not return until the dictionary has been completely loaded. If set to 0, then the START_PLAN procedure will only verify that the loading of the dictionary has started.	0
DICTIONARY_PLS_WAIT_INIT	Yes	The time in seconds to wait in between attempts to quiesce PL/SQL activity in order to write the data dictionary to redo.	300
DICTIONARY_PLS_WAIT_TIMEOUT	Yes	The maximum amount of time in seconds to attempt to quiesce PL/SQL activity in order to write the data dictionary to redo.	3600
EVENT_RECORDS	Yes	The maximum number of records to permit in DBA_ROLLING_EVENTS	10000

Table 169-4 (Cont.) Valid Values for DBMS_ROLLING.SET_PARAMETER Procedure

Parameter Name	Global?	Description	Default
FAILOVER	Yes	Automatically attempt to adjust the upgrade plan as a result of a failover event. This parameter resets its value to 0 upon completion of a subsequent call to BUILD_PLAN.	0
GRP_PREFIX	Yes	Execution of procedures in DBMS_ROLLING results in a number of Guaranteed Restore Points (GRP) taken in various databases participating in the Data Guard configuration. All such GRPs have the same prefix in their names. You can use this parameter to override the default prefix.	DBMSRU
IGNORE_BUILD_WARNINGS	Yes	Ignore warnings which would otherwise raise exceptions during execution of the BUILD_PLAN procedure.	1
IGNORE_LAST_ERROR	Yes	Ignore last encountered error upon startup of next rolling operation. This parameter resets its value to 0 upon invocation of a procedure call which resumes the rolling upgrade.	0
LAD_ENABLED_TIMEOUT	Yes	The maximum time in seconds to wait for a recently enabled log archive destination to reach a VALID state.	600
LOG_LEVEL	Yes	Logging level for the DBS_ROLLING PL/SQL package. A value of INFO results in the logging of errors and relevant non-fatal warnings. A value of FULL results in the logging of all events.	INFO
MEMBER	No	The upgrade group in which the specified database is a member. A value of LEADING indicates that the standby is a member of the leading upgrade group. As such, it is a standby of the Leading Group Master (LGM). The LGM is the database which is converted into the transient logical standby, and which becomes the new primary after the switchover. A value of TRAILING indicates that the standby is a member of the trailing upgrade group. As such, it is a standby of the Trailing Group Master (TGM). The TGM is the original primary database.	LEADING
READY_LGM_LAG_TIME	Yes	The apply lag time in seconds associated with the READY_LGM_LAG_WAIT parameter.	600
READY_LGM_LAG_TIMEOUT	Yes	The maximum amount of time in seconds to enforce READY_LGM_LAG_WAIT before halting the rolling upgrade. This parameter is only valid if READY_LGM_LAG_WAIT is set to 1.	60
READY_LGM_LAG_WAIT	Yes	Whether the START_PLAN procedure will wait for the apply lag on the leading group master to fall below READY_LGM_LAG_TIME seconds before returning control back to the user. If set to 1, the wait is performed. If set to 0, the wait is not performed.	0
SWITCH_LGM_LAG_TIME	Yes	The apply lag time in seconds associated with the SWITCH_LGM_LAG_WAIT parameter.	600
SWITCH_LGM_LAG_TIMEOUT	Yes	The maximum amount of time in seconds to enforce SWITCH_LGM_LAG_WAIT before halting the rolling upgrade. This parameter is only valid if SWITCH_LGM_LAG_WAIT is set to 1.	60
SWITCH_LGM_LAG_WAIT	Yes	Whether the SWITCHOVER procedure will wait for the apply lag on the leading group master to fall below SWITCH_LGM_LAG_TIME seconds before initiating the switchover. If set to 1, the wait is performed. If set to 0, the wait is not performed.	1

Table 169-4 (Cont.) Valid Values for DBMS_ROLLING.SET_PARAMETER Procedure

Parameter Name	Global?	Description	Default
SWITCH_LGS_LAG_TIME	Yes	The apply lag time in seconds associated with the SWITCH_LGS_LAG_WAIT parameter.	60
SWITCH_LGS_LAG_TIMEOUT	Yes	The maximum amount of time in seconds to enforce SWITCH_LGS_LAG_WAIT before halting the rolling upgrade. This parameter is only valid if SWITCH_LGS_LAG_WAIT is set to 1.	60
SWITCH_LGS_LAG_WAIT	Yes	Whether the SWITCHOVER procedure will wait for the apply lag on the leading group standbys to fall below SWITCH_LGS_LAG_TIME seconds before initiating the switchover. If set to 1, the wait is performed. If set to 0, the wait is not performed.	0
UPDATED_LGS_TIMEOUT	Yes	The maximum amount of time in seconds to enforce UPDATED_LGS_WAIT before halting the rolling upgrade. This parameter is only valid if UPDATED_LGS_WAIT is set to 1.	10800
UPDATED_LGS_WAIT	Yes	Whether the SWITCHOVER procedure will wait for the leading group standbys to complete recovery of all upgrade redo before initiating the switchover. If set to 1, the wait is performed. If set to 0, the wait is not performed	1
UPDATED_TGS_TIMEOUT	Yes	The maximum amount of time in seconds to enforce UPDATED_TGS_WAIT before halting the rolling upgrade. This parameter is only valid if UPDATED_TGS_WAIT is set to 1.	10800
UPDATED_TGS_WAIT	Yes	Whether the FINISH_PLAN procedure will wait for the trailing group standbys to complete recovery of all upgrade redo before returning control to the user. If set to 1, the wait is performed. If set to 0, the wait is not performed.	1

START_PLAN Procedure

This procedure starts the rolling operation. This procedure must be executed on the primary database to formally start the rolling operation.

When the START_PLAN procedure is complete, the future_primary parameter in the INIT_PLAN procedure will be converted into a fully configured transient logical standby database.

Syntax

```
DBMS_ROLLING.START_PLAN ();
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45414: could not connect to a remote database
- ORA-45415: instruction execution failure
- ORA-45416: operation cannot start until plan rebuild
- ORA-45417: operation not permitted since current phase was not %s

- ORA-45422: operation requires existing plan
- ORA-45426: managed recovery process was not running
- ORA-45427: logical standby Redo Apply process was not running
- ORA-45428: database was not in expected database role
- ORA-45435: managed recovery process was running
- ORA-45436: logical standby Redo Apply process was running
- ORA-45438: database is not in mounted mode
- ORA-45439: database is not in open read/write mode
- ORA-45486: database update progress is inconsistent
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- A rolling operation plan must have previously been generated through the BUILD_PLAN procedure.

SWITCHOVER Procedure

This procedure performs a switchover between the current primary database and the transient logical standby database.

At the successful completion of the procedure, the LGM assumes the primary role for the Data Guard configuration.

Syntax

```
DBMS_ROLLING.SWITCHOVER ();
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-16224: database guard is enabled
- ORA-45400: operation not permitted on current database
- ORA-45414: could not connect to a remote database
- ORA-45415: instruction execution failure
- ORA-45416: operation cannot start until plan rebuild
- ORA-45417: operation not permitted since current phase was not %s
- ORA-45422: operation requires existing plan
- ORA-45426: managed recovery process was not running
- ORA-45427: logical standby Redo Apply process was not running
- ORA-45428: database was not in expected database role
- ORA-45435: managed recovery process was running
- ORA-45436: logical standby Redo Apply process was running

- ORA-45438: database is not in mounted mode
- ORA-45439: database is not in open read/write mode
- ORA-45486: database update progress is inconsistent
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- This procedure can only be called after you have manually upgraded the transient logical standby and opened it on the higher Oracle Database version.
- Once the future primary has been upgraded, logical apply should be restarted and allowed to catch up with the primary since the primary will have been open for business and generating redo which the standby must now apply. Failing to do so will cause the switchover to take a long time since it will have to apply all of that redo.

FINISH_PLAN Procedure

This procedure finalizes the rolling operation.

It configures the former primary as a physical standby of the new primary by flashing it back to an earlier taken GRP (guaranteed restore point), converting its role to physical standby, and then starting managed recovery which will recover all redo generated by the new primary including upgrade redo and any other redo generated before or after the upgrade.

Syntax

```
DBMS_ROLLING.FINISH_PLAN ();
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45414: could not connect to a remote database
- ORA-45415: instruction execution failure
- ORA-45416: operation cannot start until plan rebuild
- ORA-45417: operation not permitted since current phase was not %s
- ORA-45422: operation requires existing plan
- ORA-45426: managed recovery process was not running
- ORA-45427: logical standby Redo Apply process was not running
- ORA-45428: database was not in expected database role
- ORA-45435: managed recovery process was running
- ORA-45436: logical standby Redo Apply process was running
- ORA-45438: database is not in mounted mode
- ORA-45439: database is not in open read/write mode
- ORA-45486: database update progress is inconsistent

- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- This procedure can only be called after you have remounted the former primary and remaining physical standbys on the higher Oracle Database version.

ROLLBACK_PLAN Procedure

This procedure rolls back the configuration-wide rolling operation.

Once completed, all of the databases in the leading group become physical standbys of the original primary database. This procedure can only be called if the configuration has not yet gone through a switchover operation since the START_PLAN procedure was invoked.

Syntax

```
DBMS_ROLLING.ROLLBACK_PLAN;
```

Parameters

This procedure has no parameters.

Exceptions

- ORA-45400: operation not permitted on current database
- ORA-45414: could not connect to a remote database
- ORA-45415: instruction execution failure
- ORA-45441: no databases eligible for rollback
- ORA-45442: rollback is not permitted after a role change
- ORA-65040: operation not allowed from within a pluggable database

Usage Notes

- You must manually restart media recovery on the lower Oracle Database version if the upgrade of the transient logical standby has already been performed.