

Oracle Data Guard

New Features for 18c



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain Private Temporary Tables on Active Data Guard Instances
- Describe the Database Nologging Enhancements
- List the New Data Guard Broker Commands
- List the new Data Guard Configurable Properties



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 2

Buffer Cache Preservation During Role Change

- The database buffer cache state is now maintained on an Active Data Guard (ADG) standby during a role change.
- Previously when an open ADG Standby transitioned to Primary, the buffer cache that was in use was recycled and has to be populated again from disk as blocks were read.
- With 18.1 the buffer cache remains intact during the role change so that performance is not affected by physical blocks reads from disk to populate the buffer cache.
- Because of this, application performance is improved on the new primary after a role transition.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 3

The database buffer cache state is now maintained on an Oracle Active Data Guard standby during a role change. As of Oracle Database 18c, the database buffer cache state is maintained on an Active Data Guard standby during a role transition so that application performance is not affected by physical blocks read from disk to populate the buffer cache. This results in improved application performance on the new primary after a role transition.

Global Temporary Tables on Active Data Guard Instances

- Global temporary tables can now be dynamically created on an Oracle Active Data Guard standby database.
- Read-mostly reporting applications that use global temporary tables for storing temporary data can be offloaded to an Oracle Active Data Guard instance.
 - When temporary undo is enabled on the primary, undo for changes to a global temporary table are not logged in the redo so the primary generates less redo.
 - The amount of redo that Oracle Data Guard must ship to the standby is also reduced, thereby reducing network bandwidth consumption and storage consumption.
- To enable temporary undo on the primary database, use the `TEMP_UNDO_ENABLED` initialization parameter.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 4

Global temporary tables can now be dynamically created on an Oracle Active Data Guard standby database. When a global temporary table is changed by a DML operation, the change itself does not generate redo because it is only a temporary table. But the undo generated for the change does in turn generate redo. Redo generation on a read-only database (such as an Active Data Guard standby) is not allowed. However, DML operations on global temporary tables are allowed on Oracle Active Data Guard standbys because the temporary undo feature allows the undo for changes to a global temporary table to be stored in the temporary tablespace as opposed to the undo tablespace. And undo stored in the temporary tablespace does not generate redo.

This feature benefits Oracle Data Guard in the following ways:

- Read-mostly reporting applications that use global temporary tables for storing temporary data can be offloaded to an Oracle Active Data Guard instance.
- When temporary undo is enabled on the primary database, undo for changes to a global temporary table are not logged in the redo and thus, the primary database generates less redo. Therefore, the amount of redo that Oracle Data Guard must ship to the standby is also reduced, thereby reducing network bandwidth consumption and storage consumption.

To enable temporary undo on the primary database, use the `TEMP_UNDO_ENABLED` initialization parameter. On an Oracle Active Data Guard standby, temporary undo is always enabled by default so the `TEMP_UNDO_ENABLED` parameter has no effect.

Private Temporary Tables on Active Data Guard Instances

- You can create private temporary tables on Oracle Active Data Guard instances even though they are read-only.
- Private temporary tables can be created in read-only databases because the metadata is stored in memory, rather than on disk.
- The lifetime of a private temporary table is only during the session which created it.
- This allows reporting applications to run on Active Data Guard standby databases.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 5

The metadata and content of a private temporary table is visible only within the session that created the it. Private temporary tables are useful in the following situations:

- When an application stores temporary data in transient tables that are populated once, read few times, and then dropped at the end of a transaction or session
- When a session is maintained indefinitely and must create different temporary tables for different transactions
- When the creation of a temporary table must not start a new transaction or commit an existing transaction
- When different sessions of the same user must use the same name for a temporary table
- When a temporary table is required for a read-only database

For example, assume a reporting application uses only one schema, but the application uses multiple connections with the schema to run different reports. The sessions use private temporary tables for calculations during individual transactions, and each session creates a private temporary table with the same name. When each transaction commits, its temporary data is no longer needed.

You can create private temporary tables on Oracle Active Data Guard instances even though they are read-only.

The reason private temporary tables can be created in read-only databases is that their metadata is stored in memory, rather than on disk. The lifetime of a private temporary table is only during the session which created it and it gets dropped automatically when the session ends. This functionality allows reporting applications to run on Active Data Guard standby databases.

Creating a Private Temporary Table

- Use the `CREATE PRIVATE TEMPORARY TABLE` statement to create a private temporary table.
- The `ON COMMIT` clause indicates if the data in the table is transaction-specific or session-specific.
 - This statement creates a private temporary table that is transaction specific:

```
SQL> CREATE PRIVATE TEMPORARY TABLE ORA$PTT_sales_ptt_transaction  
(time_id DATE, amount_sold NUMBER(10,2))  
ON COMMIT DROP DEFINITION;
```

- This statement creates a private temporary table that is session specific:

```
SQL> CREATE PRIVATE TEMPORARY TABLE ORA$PTT_sales_ptt_session  
(time_id DATE, amount_sold NUMBER(10,2))  
ON COMMIT PRESERVE DEFINITION;
```

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 6

Use the `CREATE PRIVATE TEMPORARY TABLE` statement to create a private temporary table. The `ON COMMIT` clause indicates if the data in the table is transaction-specific (the default) or session-specific the implications of which are as follows:

DROP DEFINITION: This creates a private temporary table that is transaction specific. All data in the table is lost, and the table is dropped at the end of transaction.

PRESERVE DEFINITION: This creates a private temporary table that is session specific. All data in the table is lost, and the table is dropped at the end of the session that created the table.

The examples above illustrate creating a private temporary table. The first statement creates a private temporary table that is transaction specific. The second statement creates a private temporary table that is session specific.

By default, rows in a private temporary table are stored in the default temporary tablespace of the user who creates it. However, you can assign a private temporary table to another temporary tablespace during the creation of the temporary table by using the `TABLESPACE` clause of `CREATE PRIVATE TEMPORARY TABLE` statement.

New V\$DATAGUARD_PROCESS View

- To verify database modifications are being successfully transmitted from the primary database to the standby database:

```
SQL> SELECT ROLE, THREAD#, SEQUENCE#, ACTION FROM V$DATAGUARD_PROCESS;
```

ROLE	THREAD#	SEQUENCE#	ACTION
RFS ping	1	9	IDLE
recovery apply slave	0	0	IDLE
recovery apply slave	0	0	IDLE
managed recovery	0	0	IDLE
recovery logmerger	1	9	APPLYING_LOG
RFS archive	0	0	IDLE
RFS async	1	9	IDLE

- The recovery logmerger role shows that redo is being applied at the standby.

Note: The V\$DATAGUARD_PROCESS view now shows broker processes.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 7

After you create the physical standby database and set up redo transport services, you may want to verify database modifications are being successfully transmitted from the primary database to the standby database. Oracle recommends that you use the new V\$DATAGUARD_PROCESS view instead of the V\$MANAGED_STANDBY view.

On the standby database, query the V\$DATAGUARD_PROCESS view to verify that redo is being transmitted from the primary database and applied to the standby database. In addition, the V\$DATAGUARD_PROCESS view now shows broker processes.

ADG_ACCOUNT_INFO_TRACKING initialization parameter,

ADG_ACCOUNT_INFO_TRACKING controls login attempts of users on Active Data Guard Standby databases.

Property	Description
Parameter type	String
Syntax	ADG_ACCOUNT_INFO_TRACKING = { LOCAL GLOBAL }
Default value	LOCAL
Modifiable`	No
Modifiable in a PDB	Yes
Basic	No
Oracle RAC	The same value must be used on all instances.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 8

A new initialization parameter, `ADG_ACCOUNT_INFO_TRACKING`, extends control of user account security against login attacks across a production database and all Oracle Active Data Guard standby databases. This parameter controls login attempts of users on Active Data Guard Standby databases. It extends the control of user account security information.

Setting `ADG_ACCOUNT_INFO_TRACKING` to `LOCAL` (the default value) continues to enforce the default behavior, by maintaining a local copy of users account information in the standby's in-memory view. Login failures are only tracked locally on a per database basis, and login is denied when the failure maximum is reached.

Setting the parameter to `GLOBAL` triggers a more secure behavior, by maintaining a single global copy of users account information across all Data Guard databases (primary and standby). Login failures across all databases in the Data Guard configuration count towards the maximum count and logins anywhere will be denied when the count is reached. This setting improves security against login attacks across a production database and all Active Data Guard standby databases.

Database Nologging Enhancements

- The `STANDBY NOLOGGING FOR DATA AVAILABILITY` mode causes the load operation to send the loaded data to each standby through its own standby connection.
- To enable `STANDBY NOLOGGING FOR DATA AVAILABILITY` mode :

```
SQL> ALTER DATABASE SET STANDBY NOLOGGING FOR DATA AVAILABILITY;
```

- When `STANDBY NOLOGGING FOR LOAD PERFORMANCE` mode is enabled, the loading process can stop sending the data to the standbys if the network cannot keep up with the speed at which data is being loaded to the primary.
- To enable `STANDBY NOLOGGING FOR LOAD PERFORMANCE` mode:

```
SQL> ALTER DATABASE SET STANDBY NOLOGGING FOR LOAD PERFORMANCE;
```

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 9

Database nologging has been extended with two new modes:

- *Standby Nologging for Data Availability*
`STANDBY NOLOGGING FOR DATA AVAILABILITY` mode causes the load operation to send the loaded data to each standby through its own connection to the standby. The commit is delayed until all the standbys have applied the data as part of running managed recovery in an Active Data Guard environment.
- *Standby Nologging for Load Performance*
`STANDBY NOLOGGING FOR LOAD PERFORMANCE` is similar to the previous mode except that the loading process can stop sending the data to the standbys if the network cannot keep up with the speed at which data is being loaded to the primary. In this mode it is possible that the standbys may have missing data, but each standby automatically fetches the data from the primary as a normal part of running managed recovery in an Active Data Guard environment.

When you issue any of these statements, the primary database must at least be mounted (and it can also be open). The statement can take a considerable amount of time to complete, because it waits for all unlogged direct write I/O to finish.

These modes provide better support for use in an Oracle Active Data Guard environment without significantly increasing the amount of redo generated.

Rolling Forward a Standby With One Command

- A standby database can now be refreshed over the network using one RMAN command, `RECOVER STANDBY DATABASE`.

```
RECOVER STANDBY DATABASE FROM SERVICE primary_db;
```

- The `RECOVER STANDBY DATABASE` command:
 - Restarts the standby instance
 - Refreshes the control file from the primary database
 - Automatically renames data files, temp files, and online logs
 - Restores new data files that were added to the primary database
 - Recovers the standby database up to the current time.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 10

As of Oracle Database 18c, you can refresh a standby database over the network using one RMAN command, `RECOVER STANDBY DATABASE`. The `RECOVER STANDBY DATABASE` command restarts the standby instance, refreshes the control file from the primary database, and automatically renames data files, temp files, and online logs. It restores new data files that were added to the primary database and recovers the standby database up to the current time.

When you use the `RECOVER STANDBY DATABASE` command to refresh a standby database, you specify either a `FROM SERVICE` clause or a `NOREDO` clause. The `FROM SERVICE` clause specifies the name of a primary service. The `NOREDO` clause specifies that backups should be used for the refresh, which allows a standby to be rolled forward to a specific time or SCN.

The MRP must be manually stopped on the standby before any attempt is made to sync with primary database.

The following is an example of using the `RECOVER STANDBY DATABASE` command. It shows optional usage of the `PFILE` clause to specify a parameter file for the standby database (used when the spfile is not available).

```
RECOVER STANDBY DATABASE FROM SERVICE service_name PFILE=pfile_location;
```

New Data Guard Broker Commands:

VALIDATE DATABASE SPFILE

- `VALIDATE DATABASE SPFILE` performs a comparison of SPFILE entries between the primary database and a specified standby database.

```
DGMGRL> VALIDATE DATABASE chicago SPFILE;
```

- Parameter file validation detects parameter value discrepancies between the primary and the specified standby database so that they can be rectified before a role change.
- Using this command frees you from having to restart a database to correct improperly set parameters.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 11

The `VALIDATE DATABASE SPFILE` command performs a comparison of server parameter file (SPFILE) entries between the primary database and a specified standby database. The validation of the server parameter file detects parameter value discrepancies between the primary and the specified standby database so that they can be rectified before a role change, thus ensuring that after a role change the databases perform at the same level they did prior to the role change. Additionally, using this command frees you from having to restart a database to correct improperly set parameters.

The `VALIDATE DATABASE SPFILE` command reports `No parameter differences found` if there are no differences and a list of the parameters with their differing values on the primary and the specified standby databases.

When the `VALIDATE DATABASE SPFILE` command is issued, it makes a connection to the primary database and the specified standby database based on the respective values of the `DGConnectIdentifier` property. The command fails if a connection attempt cannot complete successfully.

New Data Guard Broker Commands:

VALIDATE NETWORK CONFIGURATION

- `VALIDATE NETWORK CONFIGURATION` performs network connectivity checks between members of a broker configuration.

```
DGMGRL> VALIDATE NETWORK CONFIGURATION FOR East_Sales;
```

- The connect identifier for each connectivity check is generated based on the `DGConnectIdentifier` property of the associated database.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 12

The `VALIDATE NETWORK CONFIGURATION` command performs network connectivity checks between members of a configuration. This command also performs a check for the static connect identifier. The connect identifier for each connectivity check is generated based on the `DGConnectIdentifier` property of the associated database.

New Data Guard Broker Commands:

VALIDATE STATIC CONNECT IDENTIFIER

- `VALIDATE STATIC CONNECT IDENTIFIER` validates the static connect identifier of a database.

```
DGMGRL> VALIDATE STATIC CONNECT IDENTIFIER FOR North_Sales;
```

- To perform this validation, the broker makes a new connection to the database using a static connect identifier based on the `StaticConnectIdentifier` property.
- A new attribute, `STATIC_SERVICE=TRUE` is added to the connect identifier to ensure that a connection to the database is established using only a static service.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 13

The `VALIDATE STATIC CONNECT IDENTIFIER` command validates the static connect identifier of a database. To perform this validation, the broker makes a new connection to the database using a static connect identifier based on the `StaticConnectIdentifier` property of the database. A new attribute, `STATIC_SERVICE=TRUE` is added to the connect identifier to ensure that a connection to the database is established using only a static service, not a dynamic service.

```
DGMGRL> VALIDATE STATIC CONNECT IDENTIFIER FOR North_Sales;
```

Oracle Clusterware on database "North_Sales" is available for database restart.

Miscellaneous New Data Guard Broker Commands:

SET ECHO and SHOW ALL

- SET ECHO controls whether or not to echo commands that are issued either at the command-line prompt or from a DGMGRL script.

```
SET ECHO [ON | OFF];
```

- SHOW ALL shows the values of DGMGRL CLI properties.

```
DGMGRL> SHOW ALL;  
debug ON  
echo OFF  
time OFF  
observerconfigfile = observer.ora
```

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 14

The SET ECHO command controls whether or not to echo commands that are issued either at the command-line prompt or from a DGMGRL script.

```
DGMGRL> SET ECHO ON;  
DGMGRL> SHOW CONFIGURATION;  
SHOW CONFIGURATION;  
Configuration - DRSolution  
Protection Mode: MaxPerformance  
Members:  
North_Sales - Primary database  
South_Sales - Physical standby database  
Fast-Start Failover: DISABLED  
Configuration Status:  
SUCCESS
```

The SHOW ALL command shows the values of DGMGRL CLI properties.

New OnlineArchiveoLocation Configurable Property

- The new `OnlineArchiveoLocation` property specifies the online redo log archive location for primary, logical, and snapshot standby databases.

Category	Description
Datatype	String
Valid values	<ul style="list-style-type: none">An empty string when you do not want broker to manage the online archive location.A nonempty file specification of the redo log archive location for the instance.Specify <code>USE_DB_RECOVERY_FILE_DEST</code> if a database recovery area is desired.A valid <code>LOG_ARCHIVE_DEST_n</code> string with the <code>LOCATION</code> attribute specified,.
Default	Empty string
Parameter class	Dynamic
Role`	Primary and standby
Scope	Instance
Standby type	Physical, logical, or snapshot standby

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 15

The `OnlineArchiveLocation` configurable property specifies the online redo log archive location for primary, logical, and snapshot standby databases. Oracle recommends that you always explicitly set a value for this property.

New OnlineAlternateLocation Configurable Property

- The `OnlineAlternateLocation` property specifies an alternate online redo log archive location for primary, logical, and snapshot standby databases.

Category	Description
Datatype	String
Valid values	<ul style="list-style-type: none">• An empty string when no alternate location is desired.• A directory specification that is accessible by the instance.• A valid <code>LOG_ARCHIVE_DEST_n</code> parameter string with the <code>LOCATION</code> attribute specified, but no <code>VALID_FOR</code>, <code>ALTERNATE</code>, or <code>SERVICE</code> attributes..
Default	Empty string
Parameter class	Dynamic
Role`	Physical, logical, or snapshot standby
Scope	Instance
Standby type	Physical, logical, or snapshot standby

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 16

The `OnlineAlternateLocation` configurable property specifies an alternate online redo log archive location for primary, logical, and snapshot standby databases when the location specified by the `OnlineArchiveLocation` configurable property fails. This property has instance-specific scope, and the location it specifies must be accessible by the instance.

StandbyAlternateLocation Configurable Property

- The AlternateLocation configurable property is renamed to StandbyAlternateLocation and has new valid values.

Category	Description
Datatype	String
Valid values	<ul style="list-style-type: none">• An empty string when no alternate location is desired.• A directory specification that is accessible by the instance.• A valid LOG_ARCHIVE_DEST_n parameter string with the LOCATION attribute specified, but no VALID_FOR, ALTERNATE, or SERVICE attributes..
Default	Empty string
Parameter class	Dynamic
Role`	Primary, standby, and far sync instance
Scope	Instance
Standby type	Physical, logical, or snapshot standby, or a far sync instance

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 17

The StandbyAlternateLocation configurable property specifies an alternate standby redo log archive location to use when the location specified by the StandbyArchiveLocation configurable property fails. The property has instance-specific scope, and the location it specifies has to be accessible by the instance.

Quiz 1



A standby database can now be refreshed over the network using a single RMAN command, `RECOVER STANDBY DATABASE`.

- a. True
- b. False

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 18

Answer: a

Quiz 2



Which of the following statements regarding Private Temporary Tables are true?

- a. You can create private temporary tables on Oracle Active Data Guard instances even though they are read-only.
- b. Private temporary tables can be created in read-only databases because the metadata is stored in memory, rather than on disk.
- c. The lifetime of a private temporary table is only during the session which created it.
- d. This behavior allows reporting applications to run on Active Data Guard standby databases.

ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 19

Answer: a, b, c, d

Summary

In this lesson, you should have learned how to:

- Explain Private Temporary Tables on Active Data Guard Instances
- Describe the Database Nologging Enhancements
- List the New Data Guard Broker Commands
- List the new Data Guard Configurable Properties



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1 - 20