Redo Transport Services

An Oracle Data Guard configuration requires that Oracle redo transport services be configured and monitored.

Introduction to Redo Transport Services

Redo transport services performs the automated transfer of redo data between members of an Oracle Data Guard configuration and other databases.

The following redo transport destinations are supported:

- Oracle Data Guard standby databases

  This guide describes how to create and manage physical, logical, and snapshot standby databases.

- Oracle Data Guard primary databases

  This destination is supported only with Oracle Data Guard broker. A primary database is used as a redo transport destination to provide data protection for one or more pluggable databases (PDBs) within a database.

- Archive log repository

  This destination type is used for temporary offsite storage of archived redo log files. An archive log repository consists of an Oracle database instance and a physical standby control file. An archive log repository does not contain data files, so it cannot support role transitions.

  The procedure used to create an archive log repository is identical to the procedure used to create a physical standby database, except for the copying of data files.

- Oracle Streams downstream capture databases
- Far sync instances
- Zero Data Loss Recovery Appliance (Recovery Appliance)

Each redo transport destination is individually configured to receive redo data via one of two redo transport modes:

The SYNC attribute specifies that the synchronous redo transport mode be used to send redo data to a redo transport destination.

The ASYNC attribute specifies that the asynchronous redo transport mode be used to send redo data to a redo transport destination. The asynchronous redo transport mode is used if neither the SYNC nor the ASYNC attribute is specified.
The NET_TIMEOUT attribute specifies how long the LGWR process waits for an acknowledgement that redo data has been successfully received by a destination that

uses the synchronous redo transport mode. If an acknowledgement is not received within NET_TIMEOUT seconds, the redo transport connection is terminated and an error is logged.

- Synchronous

  The synchronous redo transport mode transmits redo data synchronously with respect to transaction commitment. A transaction cannot commit until all redo generated by that transaction has been successfully sent to every enabled redo transport destination that uses the synchronous redo transport mode.

  Although there is no limit on the distance between a primary database and a SYNC redo transport destination, transaction commit latency increases as network latency increases between a primary database and a SYNC redo transport destination.
  This transport mode is used by the Maximum Protection and Maximum Availability data protection modes described in Oracle Data Guard Protection Modes.

  Note:

  Synchronous redo transport is not supported for Zero Data Loss Recovery Appliance.

- Asynchronous

  The asynchronous redo transport mode transmits redo data asynchronously with respect to transaction commitment. A transaction can commit without waiting for the redo generated by that transaction to be successfully sent to any redo transport destination that uses the asynchronous redo transport mode.

  This transport mode is used by the Maximum Performance data protection mode described in Oracle Data Guard Protection Modes.

Configure Redo Transport Authentication

Oracle Data Guard uses Oracle Net sessions to transport redo data and control messages between the members of an Oracle Data Guard configuration.

These redo transport sessions are authenticated using either the Secure Sockets Layer (SSL) protocol or a remote login password file.

SSL is used to authenticate redo transport sessions between two databases if:

- The databases are members of the same Oracle Internet Directory (OID) enterprise domain and it allows the use of current user database links
- The LOG_ARCHIVE_DEST_n, and FAL_SERVER database initialization parameters that correspond to the databases use Oracle Net connect descriptors configured for SSL

Redo Transport Authentication Using SSL

Secure Sockets Layer (SSL) is an industry standard protocol for securing network connections.

SSL uses RSA public key cryptography and symmetric key cryptography to provide authentication, encryption, and data integrity. SSL is automatically used for redo transport authentication between two Oracle databases if:

- The databases are members of the same Oracle Internet Directory (OID) enterprise domain and that domain allows the use of current user database links.
- The LOG_ARCHIVE_DEST_*n*, and FAL_SERVER database initialization parameters that correspond to the databases use Oracle Net connect descriptors configured for SSL.
- Each database has an Oracle wallet or a supported hardware security module that contains a user certificate with a distinguished name (DN) that matches the DN in the OID entry for the database.

Redo Transport Authentication Using a Password File

If the SSL authentication requirements are not met, then each database must use a remote login password file.

In an Oracle Data Guard configuration, all physical and snapshot standby databases must use a copy of the password file from the primary database. If database compatibility is set to 12.2 or higher on both the source and target database, then the copy of the password file is automatically refreshed whenever an administrative privilege (SYSDG, SYSOPER, SYSDBA, and so on) is granted or revoked, and after the password of any user with administrative privileges is changed. The only exception to this is far sync instances. Updated password files must still be manually copied to far sync instances because far sync instances receive redo, but do not apply it. Once the password file is up-to-date at the far sync instance the redo containing the password update at the primary is automatically propagated to any standby databases that are set up to receive redo from that far sync instance. The password file is updated on the standby when the redo is applied.

When a password file is used for redo transport authentication, the password of the user account used for redo transport authentication is compared between the database initiating a redo transport session and the target database. The password must be the same at both databases to create a redo transport session.

By default, the password of the SYS user is used to authenticate redo transport sessions when a password file is used. The REDO_TRANSPORT_USER database initialization parameter can be used to select a different user password for redo transport authentication by setting this parameter to the name of any user who has been granted the SYSOPER privilege. For administrative ease, Oracle recommends that the REDO_TRANSPORT_USER parameter be set to the same value on the redo source database and at each redo transport destination.

Configuring an Oracle Database to Send Redo Data

To specify a redo transport destination, use the LOG_ARCHIVE_DEST_*n* database initialization parameter (where *n* is an integer from 1 to 31).

Note:

Configuring redo transport services to provide protection for one or more pluggable databases (PDBs) is supported through Oracle Data Guard broker only.
There is a LOG_ARCHIVE_DEST_STATE_*n* database initialization parameter (where *n* is an integer from 1 to 31) that corresponds to each LOG_ARCHIVE_DEST_*n* parameter. This parameter is used to enable or disable the corresponding redo destination.

LOG_ARCHIVE_DEST_STATE_n Initialization Parameter Values

Value	Description
ENABLE	Redo transport services can transmit redo data to this destination. This is the default.
DEFER	Redo transport services do not transmit redo data to this destination.
ALTERNATE	This destination becomes enabled if communication to its associated destination fails.

A redo transport destination is configured by setting the LOG_ARCHIVE_DEST_*n* parameter to a character string that includes one or more attributes. This section briefly describes the most commonly used attributes.
The SERVICE attribute, which is a mandatory attribute for a redo transport destination, must be the first attribute specified in the attribute list. The SERVICE attribute is used to specify the Oracle Net service name used to connect to the redo transport destination. The service name must be resolvable through an Oracle Net naming method to an Oracle Net connect descriptor that matches the Oracle Net listener(s) at the redo transport destination. The connect descriptor must specify that a dedicated server connection be used, unless that is the default connection type for the redo transport destination.

Configure the Primary Database to Receive Redo Data

It is a best practice to configure the primary database to receive redo if this is the first time a standby database is added to the configuration.

The primary database can then quickly transition to the standby role and begin receiving redo data, if necessary.

To create a standby redo log, use the

```
SQL ALTER DATABASE ADD STANDBY LOGFILE statement. For example:

SQL> ALTER DATABASE ADD STANDBY LOGFILE ('/oracle/dbs/slog1.rdo')
SIZE 500M;

SQL> ALTER DATABASE ADD STANDBY LOGFILE ('/oracle/dbs/slog2.rdo')
SIZE 500M;
```

Apply Services

These concepts describe how redo data is applied to a standby database.

Introduction to Apply Services

Apply services automatically apply *redo* to standby databases to maintain synchronization with the primary database and allow transactionally consistent access to the data.
By default, apply services waits for a standby redo log file to be archived before applying the redo that it contains. However, you can enable real-time apply, which allows apply services to apply the redo in the current standby redo log file as it is being filled.

Apply services use the following methods to maintain physical and logical standby databases:

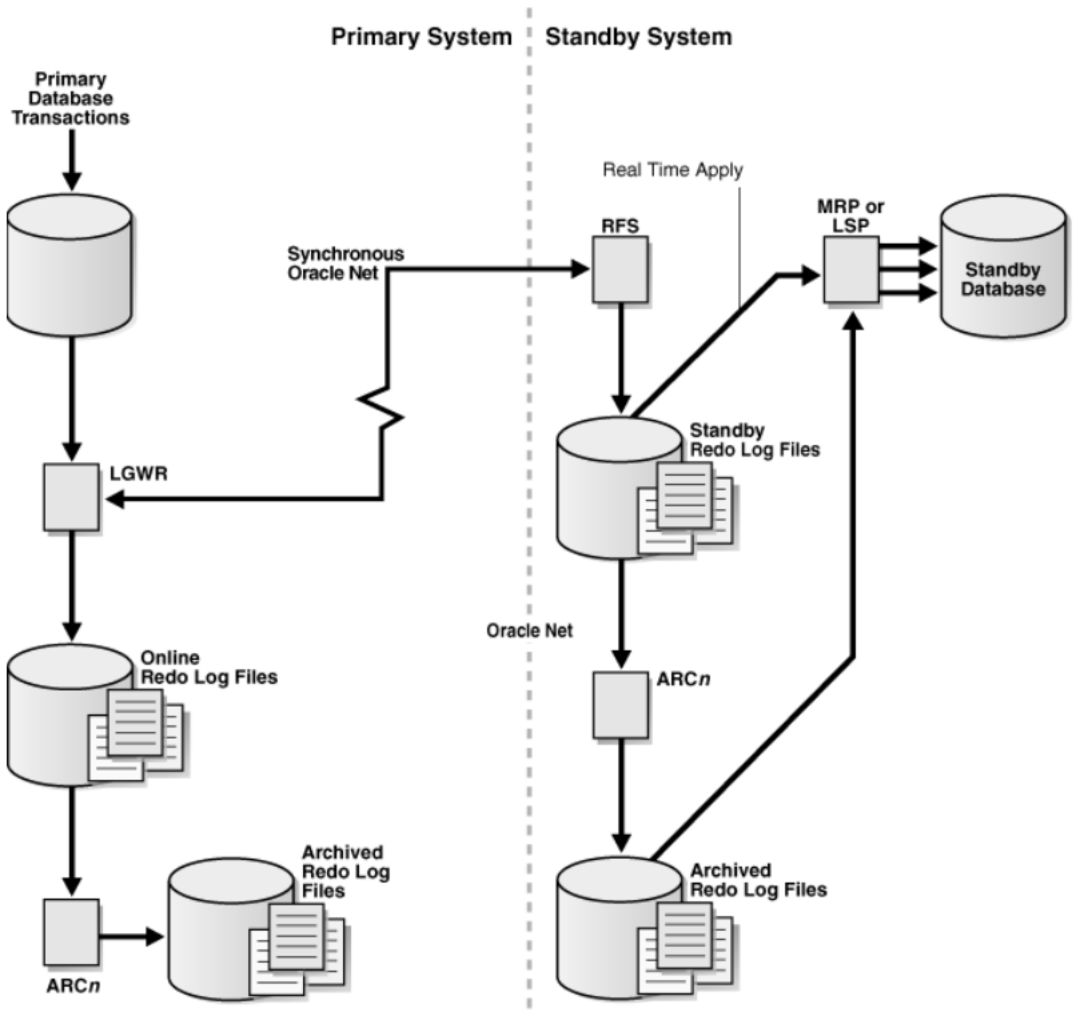
- Redo Apply (physical standby databases only)

  Uses media recovery to keep the primary and physical standby databases synchronized.

- SQL Apply (logical standby databases only)

  Reconstitutes SQL statements from the redo received from the primary database and executes the SQL statements against the logical standby database.

Figure 8-1 Applying Redo Data to a Standby Destination Using Real-Time Apply



Applying Redo Data to Physical Standby Databases

When performing Redo Apply, a physical standby database can use the real-time apply feature to apply redo directly from the standby redo log files as they are being written by the remote file server (RFS) process.

This section contains the following topics:

- Starting Redo Apply
- Stopping Redo Apply
- Monitoring Redo Apply on Physical Standby Databases

Starting Redo Apply

To start apply services on a physical standby database, ensure the physical standby database is started and mounted and then start Redo Apply.

Start apply services on a physical standby database as follows:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;
```

This also automatically enables real-time apply provided the standby database is configured with a standby redo log and is in ARCHIVELOG mode.

Redo Apply can be run either as a foreground session or as a background process. To start Redo Apply in the foreground, issue the following SQL statement:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;
```

If you start a foreground session, control is not returned to the command prompt until recovery is canceled by another session.

To start Redo Apply in the background, include the DISCONNECT keyword on the SQL statement. For example:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING ARCHIVED
LOGFILE DISCONNECT;
```

Stopping Redo Apply

Use an ALTER DATABASE SQL statement to stop Redo Apply.

For example, issue the following SQL statement:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Viewing Attributes With V$ARCHIVE_DEST

The V$ARCHIVE_DEST view can be queried to see the current settings and status for each redo transport destination.

Managing Standby Redo Logs

The synchronous and asynchronous redo transport modes require that a redo transport destination have a standby redo log. A standby redo log is used to store redo received from another Oracle database. Standby redo logs are structurally identical to redo logs, and are created and managed using the same SQL statements used to create and manage redo logs.

Redo received from another Oracle database via redo transport is written to the current standby redo log group by a remote file server (RFS) foreground process. When a log switch occurs on the redo source database, incoming redo is then written to the next standby redo log group, and the previously used standby redo log group is archived by an ARC*n* background process.

The process of sequentially filling and then archiving redo log file groups at a redo source database is mirrored at each redo transport destination by the sequential filling and archiving of standby redo log groups.

Each standby redo log file must be at least as large as the largest redo log file in the redo log of the redo source database. For administrative ease, Oracle recommends that all redo log files in the redo log at the redo source database and the standby redo log at a redo transport destination be of the same size.

The standby redo log must have at least one more redo log group than the redo log at the redo source database, for each redo thread at the redo source database. At the redo source database, query the V$LOG view to determine how many redo log groups are in the redo log at the redo source database and query the V$THREAD view to determine how many redo threads exist at the redo source database.

Perform the following query on a redo source database to determine the size of each log file and the number of log groups in the redo log:

```
SQL> SELECT GROUP#, BYTES FROM V$LOG;
```

Perform the following query on a redo destination database to determine the size of each log file and the number of log groups in the standby redo log:

```
SQL> SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

If the redo source database is an Oracle Real Applications Cluster (Oracle RAC) or Oracle Real Application Clusters One Node (Oracle RAC One Node) database, query the V$LOG view at the redo source database to determine how many redo threads exist and specify the corresponding thread numbers when adding redo log groups to the standby redo log.

The following sample SQL statements create a standby redo log at a database that is to receive redo from a redo source database that has two redo threads:

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;
```

Note:

Whenever a redo log group is added to a primary database, a log group must also be added to the standby redo log of each standby database in the configuration. Otherwise, the standby database may become unsynchronized after a primary log switch, which could temporarily prevent a zero data loss failover or cause a primary database operating in maximum protection mode to shut down.

Cases Where Redo Is Written Directly To an Archived Redo Log File

Redo received by a standby database is written directly to an archived redo log file if a standby redo log group is not available or if the redo was sent to resolve a redo gap. When this occurs, redo is written to the location specified by the LOCATION attribute of one LOG_ARCHIVE_DEST_*n* parameter that is valid for archiving redo received from another database. The LOG_ARCHIVE_DEST_*n* parameter that is used for this purpose is determined when the standby database is mounted, and this choice is reevaluated each time a LOG_ARCHIVE_DEST_*n* parameter is modified.

Location of Archived Redo Log Files

Certain criteria define where the remote file server (RFS) creates an archived redo log file.

- If no local log archive destinations are configured:
  - When Oracle Managed Files (OMF) is used, the archived redo log file is stored in the fast recovery area.
  - When OMF is not used, the archived redo log file is stored in the /dbs/arch directory.
- If one or more local log archive destinations are defined, the archived redo log file is stored in the first (lowest) valid, enabled, and active log archive destination that is valid for SRLs and the current role.
- If one or more local log archive destinations are defined, but none of them are valid, enabled, and active, the archived redo log file is stored in the /dbs/arch directory.

Monitoring Redo Transport Services

You can monitor redo transport status, as well as redo transport response time.

Monitoring Redo Transport Status

You can query views to monitor redo transport status on a redo source database.

Take the following steps to monitor redo transport status on a redo source database.

1. Perform the following query on the redo source database to determine the most recently archived sequence number for each thread:

   ```
   SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG WHERE
   RESETLOGS_CHANGE# = (SELECT MAX(RESETLOGS_CHANGE#) FROM
   V$ARCHIVED_LOG)GROUP BY THREAD#;
   ```

2. Perform the following query on the redo source database to determine the most recently archived redo log file at each redo transport destination:

   ```
   SQL> SELECT DESTINATION, STATUS, ARCHIVED_THREAD#,
   ARCHIVED_SEQ# FROM V$ARCHIVE_DEST_STATUS WHERE STATUS <>
   'DEFERRED' AND STATUS <> 'INACTIVE';

   DESTINATION         STATUS  ARCHIVED_THREAD#  ARCHIVED_SEQ#
   ------------------  ------  ----------------  ---------
   /private1/prmy/lad   VALID                 1            947
   standby1             VALID                 1            947
   ```

   The most recently archived redo log file should be the same for each destination. If it is not, a status other than VALID may identify an error encountered during the archival operation to that destination.

   Perform a query at a redo source database to find out if an archived redo log file has been received at a particular redo transport destination. Each destination has an ID number associated with it. You can query the DEST_ID column of the V$ARCHIVE_DEST view on a database to identify each destination's ID number.

   Assume that destination 1 points to the local archived redo log and that destination 2 points to a redo transport destination. Perform the following query at the redo source database to find out if any log files are missing at the redo transport destination:

   ```
   SQL> SELECT LOCAL.THREAD#, LOCAL.SEQUENCE# FROM(SELECT
   THREAD#, SEQUENCE# FROM V$ARCHIVED_LOG WHERE DEST_ID=1)LOCAL
   WHERE LOCAL.SEQUENCE# NOT IN(SELECT SEQUENCE# FROM
   V$ARCHIVED_LOG WHERE DEST_ID=2 AND THREAD# = LOCAL.THREAD#);

   THREAD#    SEQUENCE#
   ---------  ---------
      1        12
      1        13
      1        14
   ```

Redo Gap Detection and Resolution

A redo gap occurs whenever redo transmission is interrupted.

When redo transmission resumes, redo transport services automatically detects the redo gap and resolves it by sending the missing redo to the destination.

The time needed to resolve a redo gap is directly proportional to the size of the gap and inversely proportional to the effective throughput of the network link between the redo source database and the redo transport destination. Use redo transport compression to reduce the redo gap resolution time. The COMPRESSION attribute of the LOG_ARCHIVE_DEST_*n* parameter is used to specify that redo data be compressed before transmission to the destination.

Manual Gap Resolution

In some situations, gap resolution cannot be performed automatically and it must be performed manually.

For example, redo gap resolution must be performed manually on a logical standby database if the primary database is unavailable.

Perform the following query at the physical standby database to determine if there is redo gap on a physical standby database:

```
SQL> SELECT * FROM V$ARCHIVE_GAP;

    THREAD# LOW_SEQUENCE# HIGH_SEQUENCE#
-----------  -------------  --------------

          1             7              10
```

The output from the previous example indicates that the physical standby database is currently missing log files from sequence 7 to sequence 10 for thread 1.

Perform the following query on the primary database to locate the archived redo log files on the primary database (assuming the local archive destination on the primary database is LOG_ARCHIVE_DEST_1):

```
SQL> SELECT NAME FROM V$ARCHIVED_LOG WHERE THREAD#=1 AND DEST_ID=1
AND SEQUENCE# BETWEEN 7 AND 10;

NAME
---------------------------------------------
/primary/thread1_dest/arcr_1_7.arc
/primary/thread1_dest/arcr_1_8.arc /primary/thread1_dest/arcr_1_9.arc
```

Redo Transport Services Wait Events

You can use Oracle wait events to track redo transport wait time on a redo source database.

Table lists several of these Oracle wait events, which are found in the V$SYSTEM_EVENT dynamic performance view.

Wait Event	Description
LNS wait on ATTACH	Total time spent waiting for redo transport sessions to be established to all ASYNC and SYNC redo transport destinations
LNS wait on SENDREQ	Total time spent waiting for redo data to be written to all ASYNC and SYNC redo transport destinations
LNS wait on DETACH	Total time spent waiting for redo transport connections to be terminated to all ASYNC and SYNC redo transport destinations

Tuning Redo Transport

You can optimize redo transport for best performance.

Data Guard automatically tunes redo transport to optimize performance. However, if you want more control over the tuning, you can tune the following areas that are used in redo transport:

- network configuration
- storage configuration
- fast recovery area
- redo transport configuration