10

Managing Archived Redo Log Files

You manage the archived redo log files by completing tasks such as choosing between `NOARCHIVELOG` or `ARCHIVELOG` mode and specifying archive destinations.

- What Is the Archived Redo Log?
  Oracle Database lets you save filled groups of redo log files to one or more offline destinations, known collectively as the **archived redo log**.
- Choosing Between NOARCHIVELOG and ARCHIVELOG Mode
  You must choose between running your database in `NOARCHIVELOG` or `ARCHIVELOG` mode.
- Controlling Archiving
  You can set the archiving mode for your database and adjust the number of archiver processes.
- Specifying Archive Destinations
  Before you can archive redo logs, you must determine the destination to which you will archive, and familiarize yourself with the various destination states.
- About Log Transmission Modes
  The two modes of transmitting archived logs to their destination are **normal archiving transmission** and **standby transmission** mode. Normal transmission involves transmitting files to a local disk. Standby transmission involves transmitting files through a network to either a local or remote standby database.
- Managing Archive Destination Failure
  Sometimes archive destinations can fail, causing problems when you operate in automatic archiving mode. Oracle Database provides procedures to help you minimize the problems associated with destination failure.
- Controlling Trace Output Generated by the Archivelog Process
  Background processes always write to a trace file when appropriate. In the case of the archivelog process, you can control the output that is generated to the trace file.
- Viewing Information About the Archived Redo Log
  You can display information about the archived redo log using dynamic performance views or the `ARCHIVE LOG LIST` command.

See Also:

- Using Oracle Managed Files for information about creating an archived redo log that is both created and managed by the Oracle Database server
- *Oracle Real Application Clusters Administration and Deployment Guide* for information specific to archiving in the Oracle Real Application Clusters environment

10.1 What Is the Archived Redo Log?

Oracle Database lets you save filled groups of redo log files to one or more offline destinations, known collectively as the **archived redo log**.

The process of turning redo log files into archived redo log files is called **archiving**. This process is only possible if the database is running in **`ARCHIVELOG` mode**. You can choose automatic or manual archiving.

An archived redo log file is a copy of one of the filled members of a redo log group. It includes the redo entries and the unique log sequence number of the identical member of the redo log group. For example, if you are multiplexing your redo log, and if group 1 contains identical member files `a_log1` and `b_log1`, then the archiver process (ARC*n*) will archive one of these member files. Should `a_log1` become corrupted, then ARC*n* can still archive the identical `b_log1`. The archived redo log contains a copy of every group created since you enabled archiving.

When the database is running in `ARCHIVELOG` mode, the log writer process (LGWR) cannot reuse and hence overwrite a redo log group until it has been archived. The background process ARC*n* automates archiving operations when automatic archiving is enabled. The database starts multiple archiver processes as needed to ensure that the archiving of filled redo logs does not fall behind.

You can use archived redo log files to:

- Recover a database
- Update a standby database
- Get information about the history of a database using the LogMiner utility

> **See Also:**
>
> The following sources document the uses for archived redo log files:
>
> – *Oracle Database Backup and Recovery User's Guide*
>
> – *Oracle Data Guard Concepts and Administration* discusses setting up and maintaining a standby database
>
> – *Oracle Database Utilities* contains instructions for using the LogMiner PL/SQL package

10.2 Choosing Between NOARCHIVELOG and ARCHIVELOG Mode

You must choose between running your database in `NOARCHIVELOG` or `ARCHIVELOG` mode.

The choice of whether to enable the archiving of filled groups of redo log files depends on the availability and reliability requirements of the application running on the database. If you cannot afford to lose any data in your database in the event of a disk failure, use `ARCHIVELOG` mode. The archiving of filled redo log files can require you to perform extra administrative operations.

10.2.1 Running a Database in NOARCHIVELOG Mode

When you run your database in NOARCHIVELOG mode, you disable the archiving of the redo log.

The database control file indicates that filled groups are not required to be archived. Therefore, when a filled group becomes inactive after a log switch, the group is available for reuse by LGWR.

NOARCHIVELOG mode protects a database from instance failure but not from media failure. Only the most recent changes made to the database, which are stored in the online redo log groups, are available for instance recovery. If a media failure occurs while the database is in NOARCHIVELOG mode, you can only restore the database to the point of the most recent full database backup. You cannot recover transactions subsequent to that backup.

In NOARCHIVELOG mode you cannot perform online tablespace backups, nor can you use online tablespace backups taken earlier while the database was in ARCHIVELOG mode. To restore a database operating in NOARCHIVELOG mode, you can use only whole database backups taken while the database is closed. Therefore, if you decide to operate a database in NOARCHIVELOG mode, take whole database backups at regular, frequent intervals.

10.2.2 Running a Database in ARCHIVELOG Mode

When you run a database in ARCHIVELOG mode, you enable the archiving of the redo log.

The database control file indicates that a group of filled redo log files cannot be reused by LGWR until the group is archived. A filled group becomes available for archiving immediately after a redo log switch occurs.

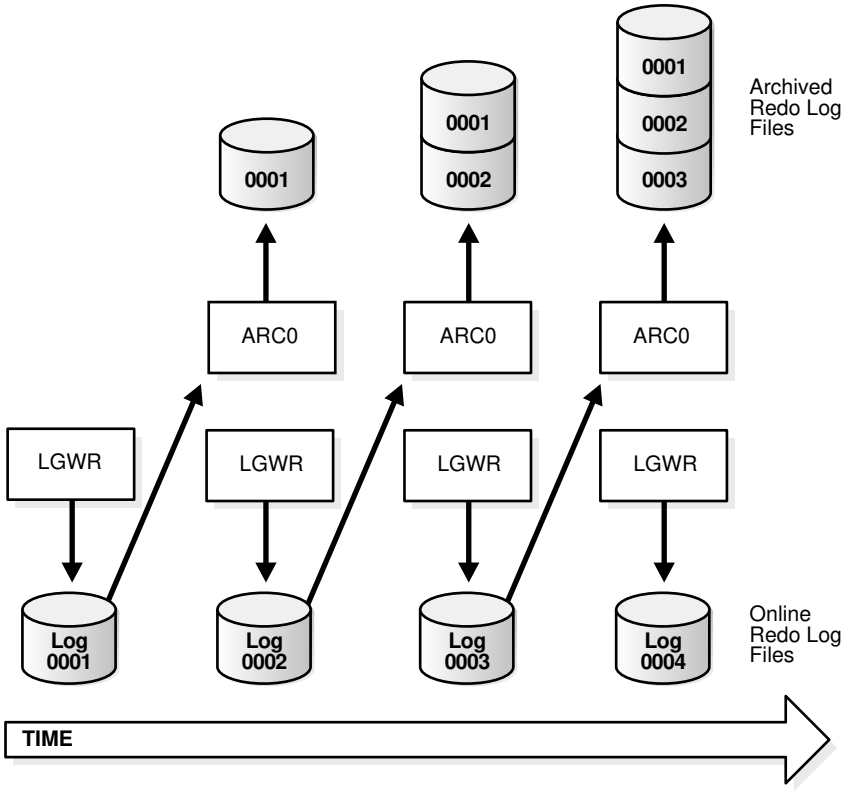
The archiving of filled groups has these advantages:

- A database backup, together with online and archived redo log files, guarantees that you can recover all committed transactions in the event of an operating system or disk failure.
- If you keep archived logs available, you can use a backup taken while the database is open and in normal system use.
- You can keep a standby database current with its original database by continuously applying the original archived redo log files to the standby.

You can configure an instance to archive filled redo log files automatically, or you can archive manually. For convenience and efficiency, automatic archiving is usually best. Figure 10-1 illustrates how the archiver process (ARC0 in this illustration) writes filled redo log files to the database archived redo log.

If all databases in a distributed database operate in ARCHIVELOG mode, you can perform coordinated distributed database recovery. However, if any database in a distributed database is in NOARCHIVELOG mode, recovery of a global distributed database (to make all databases consistent) is limited by the last full backup of any database operating in NOARCHIVELOG mode.

Figure 10-1 Redo Log File Use in ARCHIVELOG Mode



> **Tip:**
>
> It is good practice to move archived redo log files and corresponding database backups from the local disk to permanent offline storage media such as tape. A primary value of archived logs is database recovery, so you want to ensure that these logs are safe should disaster strike your primary database.

10.3 Controlling Archiving

You can set the archiving mode for your database and adjust the number of archiver processes.

- Setting the Initial Database Archiving Mode
  You set the initial archiving mode as part of database creation in the `CREATE DATABASE` statement.
- Changing the Database Archiving Mode
  To change the archiving mode of the database, use the `ALTER DATABASE` statement with the `ARCHIVELOG` or `NOARCHIVELOG` clause.
- Performing Manual Archiving
  For convenience and efficiency, automatic archiving is usually best. However, you can configure your database for manual archiving only.

- Adjusting the Number of Archiver Processes
  The `LOG_ARCHIVE_MAX_PROCESSES` initialization parameter specifies the number of ARC*n* processes that the database initially starts. The default is four processes.

> **See Also:**
>
> Your Oracle operating system specific documentation for additional information on controlling archiving modes

10.3.1 Setting the Initial Database Archiving Mode

You set the initial archiving mode as part of database creation in the `CREATE DATABASE` statement.

Usually, you can use the default of `NOARCHIVELOG` mode at database creation because there is no need to archive the redo information generated by that process. After creating the database, decide whether to change the initial archiving mode.

If you specify `ARCHIVELOG` mode, you must have initialization parameters set that specify the destinations for the archived redo log files (see "Setting Initialization Parameters for Archive Destinations").

10.3.2 Changing the Database Archiving Mode

To change the archiving mode of the database, use the `ALTER DATABASE` statement with the `ARCHIVELOG` or `NOARCHIVELOG` clause.

To change the archiving mode, you must be connected to the database with administrator privileges (`AS SYSDBA`).

The following steps switch the database archiving mode from `NOARCHIVELOG` to `ARCHIVELOG`:

1. Shut down the database instance.

   ```
   SHUTDOWN IMMEDIATE
   ```

   An open database must first be closed and any associated instances shut down before you can switch the database archiving mode. You cannot change the mode from `ARCHIVELOG` to `NOARCHIVELOG` if any data files need media recovery.

2. Back up the database.

   Before making any major change to a database, always back up the database to protect against any problems. This will be your final backup of the database in `NOARCHIVELOG` mode and can be used if something goes wrong during the change to `ARCHIVELOG` mode. See *Oracle Database Backup and Recovery User's Guide* for information about taking database backups.

3. Edit the initialization parameter file to include the initialization parameters that specify the destinations for the archived redo log files (see "Setting Initialization Parameters for Archive Destinations").

4. Start a new instance and mount, but do not open, the database.

   ```
   STARTUP MOUNT
   ```

   To enable or disable archiving, the database must be mounted but not open.

5. Change the database archiving mode. Then open the database for normal operations.

```
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
```

6. Shut down the database.

```
SHUTDOWN IMMEDIATE
```

7. Back up the database.

   Changing the database archiving mode updates the control file. After changing the database archiving mode, you must back up all of your database files and control file. Any previous backup is no longer usable because it was taken in `NOARCHIVELOG` mode.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about switching the archiving mode when using Real Application Clusters

10.3.3 Performing Manual Archiving

For convenience and efficiency, automatic archiving is usually best. However, you can configure your database for manual archiving only.

When you operate your database in manual `ARCHIVELOG` mode, you must archive inactive groups of filled redo log files or your database operation can be temporarily suspended.

To operate your database in manual archiving mode:

1. Follow the procedure described in "Changing the Database Archiving Mode ", but replace the `ALTER DATABASE` statement with the following statement:

   ```
   ALTER DATABASE ARCHIVELOG MANUAL;
   ```

2. Connect to the database as a user with administrator privileges.
3. Ensure that the database is either mounted or open.
4. Use the `ALTER SYSTEM` statement with the `ARCHIVE LOG` clause to manually archive filled redo log files. For example, the following statement archives all unarchived redo log files:

   ```
   ALTER SYSTEM ARCHIVE LOG ALL;
   ```

When you use manual archiving mode, you cannot specify any standby databases in the archiving destinations.

Even when automatic archiving is enabled, you can use manual archiving for such actions as rearchiving an inactive group of filled redo log members to another location. In this case, it is possible for the instance to reuse the redo log group before you have finished manually archiving, and thereby overwrite the files. If this happens, the database writes an error message to the alert log.

Related Topics

- Running a Database in ARCHIVELOG Mode
  When you run a database in `ARCHIVELOG` mode, you enable the archiving of the redo log.

10.3.4 Adjusting the Number of Archiver Processes

The `LOG_ARCHIVE_MAX_PROCESSES` initialization parameter specifies the number of ARC*n* processes that the database initially starts. The default is four processes.

To avoid any run-time overhead of starting additional ARC*n* processes:

- Set the `LOG_ARCHIVE_MAX_PROCESSES` initialization parameter to specify that up to 30 ARC*n* processes be started at instance startup.

The `LOG_ARCHIVE_MAX_PROCESSES` parameter is dynamic, so you can change it using the `ALTER SYSTEM` statement.

The following statement configures the database to start six ARC*n* processes upon startup:

```
ALTER SYSTEM SET LOG_ARCHIVE_MAX_PROCESSES=6;
```

The statement also has an immediate effect on the currently running instance. It increases or decreases the current number of running ARC*n* processes to six.

10.4 Specifying Archive Destinations

Before you can archive redo logs, you must determine the destination to which you will archive, and familiarize yourself with the various destination states.

The dynamic performance (V$) views, listed in "Viewing Information About the Archived Redo Log", provide all needed archive information.

- Setting Initialization Parameters for Archive Destinations
  You can choose to archive redo logs to a single destination or to multiple destinations.
- Expanding Alternate Destinations with Log Archive Destination Groups
  You can expand the number of alternate archive destinations by using log archive destination groups.
- Understanding Archive Destination Status
  Several variables determine an archive destination's status.
- Specifying Alternate Destinations
  To specify that a location be an archive destination only in the event of a failure of another destination, you can make it an alternate destination. Both local and remote destinations can be alternates.

10.4.1 Setting Initialization Parameters for Archive Destinations

You can choose to archive redo logs to a single destination or to multiple destinations.

Destinations can be local—within the local file system or an Oracle Automatic Storage Management (Oracle ASM) disk group—or remote (on a standby database). When you archive to multiple destinations, a copy of each filled redo log file is written to each destination. These redundant copies help ensure that archived logs are always available in the event of a failure at one of the destinations.

To archive to only a single destination:

- Specify that destination using the `LOG_ARCHIVE_DEST` initialization parameter.

To archive to multiple destinations:

- Choose to archive to two or more locations using the LOG_ARCHIVE_DEST_*n* initialization parameters, or to archive only to a primary and secondary destination using the LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST initialization parameters.

For local destinations, in addition to the local file system or an Oracle ASM disk group, you can archive to the Fast Recovery Area. The database uses the Fast Recovery Area to store and automatically manage disk space for a variety of files related to backup and recovery. See *Oracle Database Backup and Recovery User's Guide* for details about the Fast Recovery Area.

Typically, you determine archive log destinations during database planning, and you set the initialization parameters for archive destinations during database installation. However, you can use the ALTER SYSTEM command to dynamically add or change archive destinations after your database is running. Any destination changes that you make take effect at the next log switch (automatic or manual).

The following table summarizes the archive destination alternatives, which are further described in the sections that follow.

Method	Initialization Parameter	Host	Example
1	LOG_ARCHIVE_DEST_*n* where: *n* is an integer from 1 to 31. Archive destinations 1 to 10 are available for local or remote locations. Archive destinations 11 to 31 are available for remote locations only.	Local or remote	LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk1/arc' LOG_ARCHIVE_DEST_2 = 'LOCATION=/disk2/arc' LOG_ARCHIVE_DEST_3 = 'SERVICE=standby1'
2	LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST	Local only	LOG_ARCHIVE_DEST = '/disk1/arc' LOG_ARCHIVE_DUPLEX_DEST = '/disk2/arc'

- Method 1: Using the LOG_ARCHIVE_DEST_n Parameter
  You can use the LOG_ARCHIVE_DEST_*n* initialization parameter to specify different destinations for archived logs.
- Method 2: Using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST
  To specify a maximum of two locations, use the LOG_ARCHIVE_DEST parameter to specify a primary archive destination and the LOG_ARCHIVE_DUPLEX_DEST to specify an optional secondary archive destination.

10.4.1.1 Method 1: Using the LOG_ARCHIVE_DEST_*n* Parameter

You can use the LOG_ARCHIVE_DEST_*n* initialization parameter to specify different destinations for archived logs.

Set the LOG_ARCHIVE_DEST_*n* initialization parameter (where *n* is an integer from 1 to 31) to specify from one to 31. Each numerically suffixed parameter uniquely identifies an individual destination.

You specify the location for LOG_ARCHIVE_DEST_*n* using the keywords explained in the following table:

Keyword	Indicates	Example
LOCATION	A local file system location or Oracle ASM disk group	LOG_ARCHIVE_DEST_*n* = 'LOCATION=/disk1/arc' LOG_ARCHIVE_DEST_*n* = 'LOCATION=+DGROUP1/orcl/arc_1'
LOCATION	The Fast Recovery Area	LOG_ARCHIVE_DEST_*n* = 'LOCATION=USE_DB_RECOVERY_FILE_DEST'

Keyword	Indicates	Example
SERVICE	Remote archival through Oracle Net service name.	LOG_ARCHIVE_DEST_*n* = 'SERVICE=standby1'

If you use the LOCATION keyword, specify one of the following:

- A valid path name in your operating system's local file system
- An Oracle ASM disk group
- The keyword USE_DB_RECOVERY_FILE_DEST to indicate the Fast Recovery Area

If you specify SERVICE, supply a net service name that Oracle Net can resolve to a connect descriptor for a standby database. The connect descriptor contains the information necessary for connecting to the remote database.

Perform the following steps to set the destination for archived redo log files using the LOG_ARCHIVE_DEST_*n* initialization parameter:

1. Set the LOG_ARCHIVE_DEST_*n* initialization parameter to specify from one to 31 archiving locations. For example, enter:

   ```
   LOG_ARCHIVE_DEST_1 = 'LOCATION = /disk1/archive'
   LOG_ARCHIVE_DEST_2 = 'LOCATION = /disk2/archive'
   LOG_ARCHIVE_DEST_3 = 'LOCATION = +RECOVERY/orcl/arc_3'
   ```

   If you are archiving to a standby database, then use the SERVICE keyword to specify a valid net service name. For example, enter:

   ```
   LOG_ARCHIVE_DEST_4 = 'SERVICE = standby1'
   ```

2. (Optional) Set the LOG_ARCHIVE_FORMAT initialization parameter, using %t to include the thread number as part of the file name, %s to include the log sequence number, and %r to include the resetlogs ID (a timestamp value represented in ub4). Use capital letters (%T, %S, and %R) to pad the file name to the left with zeroes.

   > **Note:**
   >
   > The database requires the specification of resetlogs ID (%r) when you include the LOG_ARCHIVE_FORMAT parameter. The default for this parameter is operating system dependent.
   >
   > The incarnation of a database changes when you open it with the RESETLOGS option. Specifying %r causes the database to capture the resetlogs ID in the archived redo log file name. See *Oracle Database Backup and Recovery User's Guide* for more information about this method of recovery.

   The following example shows a setting of LOG_ARCHIVE_FORMAT:

   ```
   LOG_ARCHIVE_FORMAT = arch_%t_%s_%r.arc
   ```

   This setting will generate archived logs as follows for thread 1; log sequence numbers 100, 101, and 102; resetlogs ID 509210197. The identical resetlogs ID indicates that the files are all from the same database incarnation:

   ```
   /disk1/archive/arch_1_100_509210197.arc,
   /disk1/archive/arch_1_101_509210197.arc,
   ```

```
/disk1/archive/arch_1_102_509210197.arc

/disk2/archive/arch_1_100_509210197.arc,
/disk2/archive/arch_1_101_509210197.arc,
/disk2/archive/arch_1_102_509210197.arc

/disk3/archive/arch_1_100_509210197.arc,
/disk3/archive/arch_1_101_509210197.arc,
/disk3/archive/arch_1_102_509210197.arc
```

The `LOG_ARCHIVE_FORMAT` initialization parameter is ignored in some cases. See *Oracle Database Reference* for more information about this parameter.

10.4.1.2 Method 2: Using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST

To specify a maximum of two locations, use the `LOG_ARCHIVE_DEST` parameter to specify a primary archive destination and the `LOG_ARCHIVE_DUPLEX_DEST` to specify an optional secondary archive destination.

All locations must be local. Whenever the database archives a redo log, it archives it to every destination specified by either set of parameters.

Perform the following steps the use method 2:

1. Specify destinations for the `LOG_ARCHIVE_DEST` and `LOG_ARCHIVE_DUPLEX_DEST` parameter (you can also specify `LOG_ARCHIVE_DUPLEX_DEST` dynamically using the `ALTER SYSTEM` statement). For example, enter:

   ```
   LOG_ARCHIVE_DEST = '/disk1/archive'
   LOG_ARCHIVE_DUPLEX_DEST = '/disk2/archive'
   ```

2. Set the `LOG_ARCHIVE_FORMAT` initialization parameter as described in step 2 for method 1.

> **Note:**
>
> If you configure a Fast Recovery Area (by setting the `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE` parameters) and do not specify any local archive destinations, the database automatically selects the Fast Recovery Area as a local archive destination and sets `LOG_ARCHIVE_DEST_1` to `USE_DB_RECOVERY_FILE_DEST`.

> **WARNING:**
>
> You must ensure that there is sufficient disk space at all times for archive log destinations. If the database encounters a disk full error as it attempts to archive a log file, an irrecoverable error occurs and the database stops responding. You can check the alert log for a disk full message.

> **See Also:**
>
> - *Oracle Database Reference* for additional information about the initialization parameters used to control the archiving of redo logs
> - *Oracle Data Guard Concepts and Administration* for information about using the `LOG_ARCHIVE_DEST_n` initialization parameter for specifying a standby destination. There are additional keywords that can be specified with this initialization parameter that are not discussed in this book.
> - *Oracle Database Net Services Administrator's Guide* for a discussion of net service names and connect descriptors.
> - *Oracle Database Backup and Recovery User's Guide* for information about the Fast Recovery Area

10.4.2 Expanding Alternate Destinations with Log Archive Destination Groups

You can expand the number of alternate archive destinations by using log archive destination groups.

- About Log Archive Destination Groups
  A log archive destination group specifies multiple archive destinations, and the destinations in the group can be prioritized. You can specify multiple groups to expand the number of possible archive destinations for your database.
- Specifying Log Archive Destination Groups
  Use the `GROUP` attribute of the `LOG_ARCHIVE_DEST_n` initialization parameter to specify log archive destination groups.

10.4.2.1 About Log Archive Destination Groups

A log archive destination group specifies multiple archive destinations, and the destinations in the group can be prioritized. You can specify multiple groups to expand the number of possible archive destinations for your database.

To specify a log archive destination group, use the `GROUP` attribute of the `LOG_ARCHIVE_DEST_n` initialization parameter. There can be up to 30 log archive destinations included in a group. One member of each group is active, and the others are available for use in the event of a failure of the active destination. If the active destination becomes inactive, then Oracle Database switches to an available destination as long as one or more are available in the group. You can indicate which destinations to use first by prioritizing the destinations with the `PRIORITY` attribute.

A log archive destination group is referenced by a group number, which is specified when the group is created. There can be up to eight groups. To specify where to archive the redo data within a group, all of the log archive destinations must specify the `SERVICE` attribute.

To prioritize the destinations in a group, set the `PRIORITY` attribute for a destination to an integer in the range of 1 through 8. The lower number indicates the higher priority. The priority determines which destination within a group to make active when the database is mounted or when the active destination fails. For example, a `PRIORITY` value of `2` is higher priority than a `PRIORITY` value of `7`. Therefore, if the currently active destination with the `PRIORITY` value of `1`

in the group becomes inactive, then the destination with the PRIORITY value of 2 is used before the destination with the PRIORITY value of 7. If the PRIORITY attribute is not set for a destination, then the default value is 1.

The priority is also considered when a previously failed destination becomes available. If an active destination fails, and Oracle Database switches to a destination with a lower priority, then Oracle Database switches back to the destination with higher priority when it becomes available again. For example, if an active destination with priority 1 becomes inactive, and Oracle Database switches to a destination with priority 2, then Oracle Database switches back to the destination with priority 1 when it becomes available again, even if the priority 2 destination did not fail.

However, more than one destination assigned to the same group can have the same priority. For example, there can be three destinations with priority 1. In such a group, a failure of the active destination results in a switch to another member with the same priority. In this case, there is no switch back to the original destination when it becomes available again because both destinations have the same priority. If the second destination fails after the first destination has become available again, then the database will switch to the first destination or to another destination in the group with the same priority.

See Also:

Oracle Data Guard Concepts and Administration

10.4.2.2 Specifying Log Archive Destination Groups

Use the GROUP attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter to specify log archive destination groups.

You can create up to eight log archive destination groups, and each group can have up to 30 destinations specified.

To specify log archive destination groups, the database must be running in ARCHIVELOG mode.

- Set the LOG_ARCHIVE_DEST_*n* initialization parameter, and include the GROUP attribute to specify log archive destination groups.

  Optionally, include the PRIORITY attribute to specify which log archive destination within a group to make active when the system is started or when a destination fails.

Example 10-1 Specifying Two Log Archive Destination Groups

This example specifies two log archive destination groups (1 and 2). Each group has three log archive destinations specified.

```
LOG_ARCHIVE_DEST_1 = 'SERVICE=SITEa VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=1'
LOG_ARCHIVE_DEST_2 = 'SERVICE=SITEb VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=1'
LOG_ARCHIVE_DEST_3 = 'SERVICE=SITEc VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=1'

LOG_ARCHIVE_DEST_4 = 'SERVICE=SITE1 VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=2'
LOG_ARCHIVE_DEST_5 = 'SERVICE=SITE2 VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=2'
```

```
LOG_ARCHIVE_DEST_6 = 'SERVICE=SITE3 VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
GROUP=2'
```

Example 10-2 Specifying Priority Within a Log Archive Destination Group

This example specifies different priority levels for destinations within a single log archive destination group. Specifically, destination 1 and 2 are both at priority level 1, destination 3 is at priority level 2, and destination 4 is at priority level 3.

```
LOG_ARCHIVE_DEST_1 = 'SERVICE=SITE1 SYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) GROUP=1 PRIORITY=1'
LOG_ARCHIVE_DEST_2 = 'SERVICE=SITE2 SYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) GROUP=1 PRIORITY=1'
LOG_ARCHIVE_DEST_3 = 'SERVICE=SITE3 ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) GROUP=1 PRIORITY=2'
LOG_ARCHIVE_DEST_4 = 'SERVICE=SITE4 ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) GROUP=1 PRIORITY=3'
```

In this example, sites 1, 2 and 3 could be Oracle Data Guard far sync instances that only forward the redo, and site 4 is the actual remote standby database. Alternatively, sites 1, 2, 3, and 4 could all be standby databases which are configured to cascade the redo to the other sites when they are the active destination.

The following priority rules are followed:

- The default active destination can be destination 1 or destination 2 because both are at priority level 1.
- If destination 1 is active but then becomes unavailable, then Oracle Database switches to destination 2. Similarly, if destination 2 is active but then becomes unavailable, then Oracle Database switches to destination 1. When either destination 1 or 2 is available, one of them is used.
- If both destination 1 and destination 2 become unavailable, then destination 3 is used.
- If, when destination 3 is active, destination 1 or destination 2 becomes available, Oracle Database switches to the available priority 1 destination.
- If destination 1, 2, and 3 all become unavailable, then destination 4 is used.
- If, when destination 4 is active, destination 1, 2, or 3 becomes available, Oracle Database switches to the available priority 1 destination first and then to the available priority 2 destination.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration*

10.4.3 Understanding Archive Destination Status

Several variables determine an archive destination's status.

Each archive destination has the following variable characteristics that determine its status:

- **Valid/Invalid**: indicates whether the disk location or service name information is specified and valid

- **Enabled/Disabled**: indicates the availability state of the location and whether the database can use the destination
- **Active/Inactive**: indicates whether there was a problem accessing the destination

Several combinations of these characteristics are possible. To obtain the current status and other information about each destination for an instance, query the `V$ARCHIVE_DEST` view.

The `LOG_ARCHIVE_DEST_STATE_n` (where *n* is an integer from 1 to 31) initialization parameter lets you control the availability state of the specified destination (*n*).

- `ENABLE` indicates that the database can use the destination.
- `DEFER` indicates that the location is temporarily disabled.
- `ALTERNATE` indicates that the destination is an alternate. The availability state of an alternate destination is `DEFER`. If its parent destination fails, the availability state of the alternate becomes `ENABLE`. `ALTERNATE` cannot be specified for destinations `LOG_ARCHIVE_DEST_11` to `LOG_ARCHIVE_DEST_31`.

10.4.4 Specifying Alternate Destinations

To specify that a location be an archive destination only in the event of a failure of another destination, you can make it an alternate destination. Both local and remote destinations can be alternates.

The following example makes `LOG_ARCHIVE_DEST_4` an alternate for `LOG_ARCHIVE_DEST_3`:

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_4 = 'LOCATION=/disk4/arch';
ALTER SYSTEM SET LOG_ARCHIVE_DEST_3 = 'LOCATION=/disk3/arch MAX_FAILURE=1
   ALTERNATE=LOG_ARCHIVE_DEST_4';
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_4=ALTERNATE;

SQL> SELECT dest_name, status, destination FROM v$archive_dest;

DEST_NAME               STATUS    DESTINATION
----------------------- --------- -------------------------------------------
LOG_ARCHIVE_DEST_1      VALID     /disk1/arch
LOG_ARCHIVE_DEST_2      VALID     /disk2/arch
LOG_ARCHIVE_DEST_3      VALID     /disk3/arch
LOG_ARCHIVE_DEST_4      ALTERNATE /disk4/arch
```

10.5 About Log Transmission Modes

The two modes of transmitting archived logs to their destination are **normal archiving transmission** and **standby transmission** mode. Normal transmission involves transmitting files to a local disk. Standby transmission involves transmitting files through a network to either a local or remote standby database.

- Normal Transmission Mode
  In normal transmission mode, the archiving destination is another disk drive of the database server.
- Standby Transmission Mode
  In standby transmission mode, the archiving destination is either a local or remote standby database.

10.5.1 Normal Transmission Mode

In normal transmission mode, the archiving destination is another disk drive of the database server.

In this configuration archiving does not contend with other files required by the instance and can complete more quickly. Specify the destination with either the `LOG_ARCHIVE_DEST_n` or `LOG_ARCHIVE_DEST` parameters.

10.5.2 Standby Transmission Mode

In standby transmission mode, the archiving destination is either a local or remote standby database.

> **Note:**
>
> You can maintain a standby database on a local disk, but Oracle strongly encourages you to maximize disaster protection by maintaining your standby database at a remote site.

> **See Also:**
>
> - *Oracle Data Guard Concepts and Administration*
> - *Oracle Database Net Services Administrator's Guide* for information about connecting to a remote database using a service name

10.6 Managing Archive Destination Failure

Sometimes archive destinations can fail, causing problems when you operate in automatic archiving mode. Oracle Database provides procedures to help you minimize the problems associated with destination failure.

- Specifying the Minimum Number of Successful Destinations
  The optional initialization parameter `LOG_ARCHIVE_MIN_SUCCEED_DEST=n` determines the minimum number of destinations to which the database must successfully archive a redo log group before it can reuse online log files. The default value is 1. Valid values for *n* are 1 to 2 if you are using duplexing, or 1 to 31 if you are multiplexing.
- Rearchiving to a Failed Destination
  Use the `REOPEN` attribute of the `LOG_ARCHIVE_DEST_n` parameter to specify whether and when ARC*n* should attempt to rearchive to a failed destination following an error. `REOPEN` applies to all errors, not just `OPEN` errors.

10.6.1 Specifying the Minimum Number of Successful Destinations

The optional initialization parameter LOG_ARCHIVE_MIN_SUCCEED_DEST=*n* determines the minimum number of destinations to which the database must successfully archive a redo log group before it can reuse online log files. The default value is 1. Valid values for *n* are 1 to 2 if you are using duplexing, or 1 to 31 if you are multiplexing.

- Specifying Mandatory and Optional Destinations
  The LOG_ARCHIVE_DEST_*n* initialization parameter lets you specify whether a destination is OPTIONAL (the default) or MANDATORY.
- Specifying the Number of Successful Destinations: Scenarios
  You can see the relationship between the LOG_ARCHIVE_DEST_*n* and LOG_ARCHIVE_MIN_SUCCEED_DEST initialization parameters most easily through sample scenarios.

10.6.1.1 Specifying Mandatory and Optional Destinations

The LOG_ARCHIVE_DEST_*n* initialization parameter lets you specify whether a destination is OPTIONAL (the default) or MANDATORY.

- Set the destination as OPTIONAL (the default) or MANDATORY in the LOG_ARCHIVE_DEST_*n* initialization parameter.

The LOG_ARCHIVE_MIN_SUCCEED_DEST=*n* parameter uses all MANDATORY destinations plus some number of non-standby OPTIONAL destinations to determine whether LGWR can overwrite the online log. The following rules apply:

- Omitting the MANDATORY attribute for a destination is the same as specifying OPTIONAL.
- You must have at least one local destination, which you can declare OPTIONAL or MANDATORY.
- The MANDATORY attribute can only be specified for destinations LOG_ARCHIVE_DEST_1 through LOG_ARCHIVE_DEST_10.
- When you specify a value for LOG_ARCHIVE_MIN_SUCCEED_DEST=*n*, Oracle Database will treat at least one local destination as MANDATORY, because the minimum value for LOG_ARCHIVE_MIN_SUCCEED_DEST is 1.
- The LOG_ARCHIVE_MIN_SUCCEED_DEST value cannot be greater than the number of destinations, nor can it be greater than the number of MANDATORY destinations plus the number of OPTIONAL local destinations.
- If you DEFER a MANDATORY destination, and the database overwrites the online log without transferring the archived log to the standby site, then you must transfer the log to the standby manually.

If you are duplexing the archived logs, you can establish which destinations are mandatory or optional by using the LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST parameters. The following rules apply:

- Any destination declared by LOG_ARCHIVE_DEST is mandatory.
- Any destination declared by LOG_ARCHIVE_DUPLEX_DEST is optional if LOG_ARCHIVE_MIN_SUCCEED_DEST = 1 and mandatory if LOG_ARCHIVE_MIN_SUCCEED_DEST = 2.

10.6.1.2 Specifying the Number of Successful Destinations: Scenarios

You can see the relationship between the LOG_ARCHIVE_DEST_*n* and LOG_ARCHIVE_MIN_SUCCEED_DEST initialization parameters most easily through sample scenarios.

- Scenario for Archiving to Optional Local Destinations
  In this scenario, you archive to three local destinations, each of which you declare as OPTIONAL.
- Scenario for Archiving to Both Mandatory and Optional Destinations
  In this scenario, you archive to MANDATORY and OPTIONAL local destinations.

10.6.1.2.1 Scenario for Archiving to Optional Local Destinations

In this scenario, you archive to three local destinations, each of which you declare as OPTIONAL.

Table 10-1 illustrates the possible values for LOG_ARCHIVE_MIN_SUCCEED_DEST=*n* in this case.

Table 10-1 LOG_ARCHIVE_MIN_SUCCEED_DEST Values for Scenario 1

Value	Meaning
1	The database can reuse log files only if at least one of the OPTIONAL destinations succeeds.
2	The database can reuse log files only if at least two of the OPTIONAL destinations succeed.
3	The database can reuse log files only if all of the OPTIONAL destinations succeed.
4 or greater	ERROR: The value is greater than the number of destinations.

This scenario shows that even though you do not explicitly set any of your destinations to MANDATORY using the LOG_ARCHIVE_DEST_*n* parameter, the database must successfully archive to one or more of these locations when LOG_ARCHIVE_MIN_SUCCEED_DEST is set to 1, 2, or 3.

10.6.1.2.2 Scenario for Archiving to Both Mandatory and Optional Destinations

In this scenario, you archive to MANDATORY and OPTIONAL local destinations.

Consider a case in which:

- You specify two MANDATORY destinations.
- You specify two OPTIONAL destinations.
- No destination is a standby database.

Table 10-2 shows the possible values for LOG_ARCHIVE_MIN_SUCCEED_DEST=*n*.

Table 10-2 LOG_ARCHIVE_MIN_SUCCEED_DEST Values for Scenario 2

Value	Meaning
1	The database ignores the value and uses the number of MANDATORY destinations (in this example, 2).

Table 10-2 (Cont.) LOG_ARCHIVE_MIN_SUCCEED_DEST Values for Scenario 2

Value	Meaning
2	The database can reuse log files even if no OPTIONAL destination succeeds.
3	The database can reuse logs only if at least one OPTIONAL destination succeeds.
4	The database can reuse logs only if both OPTIONAL destinations succeed.
5 or greater	ERROR: The value is greater than the number of destinations.

This case shows that the database must archive to the destinations you specify as MANDATORY, regardless of whether you set LOG_ARCHIVE_MIN_SUCCEED_DEST to archive to a smaller number of destinations.

10.6.2 Rearchiving to a Failed Destination

Use the REOPEN attribute of the LOG_ARCHIVE_DEST_*n* parameter to specify whether and when ARC*n* should attempt to rearchive to a failed destination following an error. REOPEN applies to all errors, not just OPEN errors.

REOPEN=*n* sets the minimum number of seconds before ARC*n* should try to reopen a failed destination. The default value for *n* is 300 seconds. A value of 0 is the same as turning off the REOPEN attribute; ARC*n* will not attempt to archive after a failure. If you do not specify the REOPEN keyword, ARC*n* will never reopen a destination following an error.

You cannot use REOPEN to specify the number of attempts ARC*n* should make to reconnect and transfer archived logs. The REOPEN attempt either succeeds or fails.

When you specify REOPEN for an OPTIONAL destination, the database can overwrite online logs if there is an error. If you specify REOPEN for a MANDATORY destination, the database stalls the production database when it cannot successfully archive. In this situation, consider the following options:

- Archive manually to the failed destination.
- Change the destination by deferring the destination, specifying the destination as optional, or changing the service.
- Drop the destination.

When using the REOPEN keyword, note the following:

- ARC*n* reopens a destination only when *starting* an archive operation from the beginning of the log file, never *during* a current operation. ARC*n* always retries the log copy from the beginning.
- If you specified REOPEN, either with a specified time the default, ARC*n* checks to see whether the time of the recorded error plus the REOPEN interval is less than the current time. If it is, ARC*n* retries the log copy.
- The REOPEN clause successfully affects the ACTIVE=TRUE destination state. The VALID and ENABLED states are not changed.

  Something wrong here. A destination can be inactive, or valid, or disabled. There is no ACTIVE status. So I think maybe it should say, "The REOPEN clause sets the destination status to VALID" ...? DL

10.7 Controlling Trace Output Generated by the Archivelog Process

Background processes always write to a trace file when appropriate. In the case of the archivelog process, you can control the output that is generated to the trace file.

To control the output that is generated to the trace file for the archivelog process:

- Set the `LOG_ARCHIVE_TRACE` initialization parameter to specify a **trace level**, such as 0, 1, 2, 4, 8, and so on.

You can combine tracing levels by specifying a value equal to the sum of the individual levels that you would like to trace. For example, setting `LOG_ARCHIVE_TRACE=12` will generate trace level 8 and 4 output. You can set different values for the primary and any standby database.

The default value for the `LOG_ARCHIVE_TRACE` parameter is 0. At this level, the archivelog process generates appropriate alert and trace entries for error conditions.

You can change the value of this parameter dynamically using the `ALTER SYSTEM` statement. For example:

```
ALTER SYSTEM SET LOG_ARCHIVE_TRACE=12;
```

Changes initiated in this manner will take effect at the start of the next archiving operation.

> **See Also:**
>
> - "Monitoring Errors with Trace Files and the Alert Log"
> - *Oracle Database Reference* for more information about the `LOG_ARCHIVE_TRACE` initialization parameter, including descriptions of the valid values for this parameter
> - *Oracle Data Guard Concepts and Administration* for information about using this parameter with a standby database

10.8 Viewing Information About the Archived Redo Log

You can display information about the archived redo log using dynamic performance views or the `ARCHIVE LOG LIST` command.

- Archived Redo Log Files Views
  You can query a set of dynamic performance views for information about archived redo log files.
- Using the ARCHIVE LOG LIST Command
  The SQL*Plus command `ARCHIVE LOG LIST` displays archiving information for the connected instance.

10.8.1 Archived Redo Log Files Views

You can query a set of dynamic performance views for information about archived redo log files.

Several dynamic performance views contain useful information about archived redo log files, as summarized in the following table.

Dynamic Performance View	Description
V$DATABASE	Shows if the database is in ARCHIVELOG or NOARCHIVELOG mode and if MANUAL (archiving mode) has been specified.
V$ARCHIVED_LOG	Displays historical archived log information from the control file. If you use a recovery catalog, the RC_ARCHIVED_LOG view contains similar information.
V$ARCHIVE_DEST	Describes the current instance, all archive destinations, and the current value, mode, and status of these destinations.
V$ARCHIVE_PROCESSES	Displays information about the state of the various archive processes for an instance.
V$BACKUP_REDOLOG	Contains information about any backups of archived logs. If you use a recovery catalog, the RC_BACKUP_REDOLOG contains similar information.
V$LOG	Displays all redo log groups for the database and indicates which need to be archived.
V$LOG_HISTORY	Contains log history information such as which logs have been archived and the SCN range for each archived log.

For example, the following query displays which redo log group requires archiving:

```
SELECT GROUP#, ARCHIVED
   FROM SYS.V$LOG;

GROUP#     ARC
--------   ---
       1   YES
       2   NO
```

To see the current archiving mode, query the V$DATABASE view:

```
SELECT LOG_MODE FROM SYS.V$DATABASE;

LOG_MODE
------------
NOARCHIVELOG
```

10.8.2 Using the ARCHIVE LOG LIST Command

The SQL*Plus command ARCHIVE LOG LIST displays archiving information for the connected instance.

For example:

```
SQL> ARCHIVE LOG LIST

Database log mode              Archive Mode
Automatic archival             Enabled
```

```
Archive destination            D:\oracle\oradata\IDDB2\archive
Oldest online log sequence     11160
Next log sequence to archive   11163
Current log sequence           11163
```

This display tells you all the necessary information regarding the archived redo log settings for the current instance:

- The database is currently operating in `ARCHIVELOG` mode.
- Automatic archiving is enabled.
- The archived redo log destination is D:\oracle\oradata\IDDB2\archive.
- The oldest filled redo log group has a sequence number of 11160.
- The next filled redo log group to archive has a sequence number of 11163.
- The current redo log file has a sequence number of 11163.

See Also:

*SQL*Plus User's Guide and Reference* for more information on the `ARCHIVE LOG LIST` command