

**In a Real Application Cluster (RAC) environment, Oracle uses redo log files for each instance to maintain instance recovery and crash recovery. Loss of redo log files, particularly active or current redo log files, can lead to severe issues as they contain data needed for crash recovery. A redo log file loss scenario requires careful handling, as it could compromise database integrity.**

## **1. Redo Log File Loss Scenario**

Let's assume you're working in a two-node Oracle RAC environment, and the redo log file for instance 1 becomes corrupted or is lost.

- **Node 1 (Instance 1):** Lost a member of its current or active redo log group.
- **Node 2 (Instance 2):** The redo log files are intact.

In Oracle RAC, each instance has its own redo log files, but they all share the same control file. The loss of redo logs on one instance could potentially affect the recovery of that instance.

## **2. Real-Time Steps to Recover from Redo Log Loss in RAC**

### **Step 1: Check the Status of the Redo Log Files**

First, determine the status of the lost redo log files. Use the following query to check the status of the redo logs for the instance:

```
sql
```

```
SELECT GROUP#, STATUS, MEMBER FROM V$LOGFILE WHERE STATUS <> 'CURRENT';
```

Check if the affected log file group is in `CURRENT`, `ACTIVE`, or `INACTIVE` state:

- **CURRENT:** The log file group is actively being written.
- **ACTIVE:** The log file group contains data that hasn't been archived yet but is no longer being written to.
- **INACTIVE:** The log file group is no longer needed for recovery.

### **Step 2: Attempt to Clear the Redo Log Group (if INACTIVE or UNUSED)**

If the lost redo log file group is not in use (i.e., `INACTIVE`), you can try to clear the log file group. This can be done if you have archived copies of the logs.

```
sql
```

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP <group_number>;
```

This command clears the redo log group, and Oracle will recreate the lost redo log file. However, this action will fail if the log file is **CURRENT** or **ACTIVE**, and unarchived, because Oracle still needs the contents of the redo log for recovery.

### Step 3: Take the Instance Offline and Shutdown the Node

If the redo log group is **CURRENT** or **ACTIVE** and the redo log cannot be cleared, the affected instance must be shut down immediately. This helps prevent further corruption.

```
sql  
  
SHUTDOWN ABORT;
```

On the remaining node(s), the RAC instances will continue to run. You must prevent the node with the lost redo log from coming back online until recovery is complete.

### Step 4: Recreate the Missing Redo Logs

After the affected node is down, the lost redo log files need to be recreated. This step will require downtime for that node, but the other nodes in the RAC cluster can continue functioning.

1. Remove the corrupted redo log file members:

```
sql  
  
ALTER DATABASE DROP LOGFILE MEMBER '<redo_log_file_path>;
```

2. Recreate the redo log files:

```
sql  
  
ALTER DATABASE ADD LOGFILE MEMBER '<new_redo_log_file_path>' TO  
GROUP <group_number>;
```

Repeat the process for any additional redo log files that are lost or corrupted.

### Step 5: Startup the Instance and Verify

Once the redo logs have been recreated, you can restart the instance:

```
sql  
  
STARTUP;
```

Verify that the instance starts up without error and that the redo logs are functioning correctly:

```
sql
```

```
SELECT GROUP#, STATUS, MEMBER FROM V$LOGFILE;
```

### 3. Workaround Solutions

#### a) Multiplexing Redo Log Files

To avoid a redo log file loss in the future, ensure that redo log files are **multiplexed**. Oracle recommends keeping at least two members in each redo log group, stored on separate disks. This ensures that if one member is lost, the other can still be used for recovery.

```
sql
```

```
ALTER DATABASE ADD LOGFILE MEMBER '<new_redo_log_path>' TO GROUP  
<group_number>;
```

#### b) Enable ARCHIVELOG Mode

Ensure that your database is running in ARCHIVELOG mode. This ensures that every redo log file is archived after being written to, which is crucial for recovery in case of redo log file loss.

```
sql
```

```
ARCHIVE LOG LIST;
```

If your database is not in ARCHIVELOG mode, enable it:

```
sql
```

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;  
ALTER DATABASE ARCHIVELOG;  
ALTER DATABASE OPEN;
```

#### c) Use Fast Recovery Area (FRA)

Set up an **Oracle Fast Recovery Area (FRA)** to automatically manage and back up your critical files, including redo logs and archived redo logs. This helps ensure that there are always backup copies available.

```
sql
```

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/path_to_FRA';  
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=20G;
```

## 4. Recommendations for Best Practices

- **Multiplex all redo log files across different disks** to avoid a single point of failure.
- **Backup the control files regularly**, as they keep track of redo log files.
- **Monitor redo log file usage** and configure Oracle's alert log to detect any issues with redo logs.
- **Use Oracle Data Guard** to set up a standby database for disaster recovery, providing another level of protection.

### Redo Log File Status in Oracle 19c

Status	Explanation
<b>CURRENT</b>	The redo log group is the one currently being written to by the database. No other group can be CURRENT.
<b>ACTIVE</b>	The redo log group is required for instance recovery and has not yet been archived (if in ARCHIVELOG mode). The contents are needed to protect against data loss, and this group cannot be reused yet.
<b>CLEARING</b>	The redo log group is being cleared after a <code>CLEAR LOGFILE</code> command was issued. The log is being initialized and will soon be available for reuse.
<b>CLEARING_CURRENT</b>	The current redo log group is being cleared after a failure. The log is still being written to, but is in the process of being cleared. It cannot be reused until this process is finished.
<b>INACTIVE</b>	The redo log group is no longer required for instance recovery. It has been archived (if in ARCHIVELOG mode) and can be reused in the next log switch.
<b>UNUSED</b>	The redo log group has never been written to since it was created. It will become ACTIVE or CURRENT when it's used for the first time.
<b>EXPIRED</b>	The redo log group has been archived and its contents are no longer required. It's ready to be reused.
<b>STALE</b>	One or more members of the redo log group are unavailable or corrupted. Action is required to resolve the issue (e.g., re-adding or recovering the log files).

## Explanation of Statuses

- **CURRENT:** Indicates the redo log group that is being used for the current transactions. Only one redo log group can be in the `CURRENT` state at any time. This group cannot be archived or reused until a log switch occurs.
- **ACTIVE:** Refers to redo log groups that are still required for crash or instance recovery. If your database is running in `ARCHIVELOG` mode, these logs must be archived before they can be reused.
- **CLEARING:** Occurs when a redo log group is being cleared. This happens when redo logs have been manually cleared or when a redo log group is corrupted and is in the process of being initialized for reuse.
- **CLEARING\_CURRENT:** Similar to `CLEARING`, but it applies to the currently active redo log group. Oracle clears the current log in the event of certain errors.
- **INACTIVE:** Refers to a redo log group that is no longer needed for recovery. If the database is in `ARCHIVELOG` mode, this group has been archived and is ready for reuse.
- **UNUSED:** Redo log groups that have been created but never written to since their creation.
- **EXPIRED:** This status typically appears for archived redo log files that have been processed and no longer hold any significant recovery information. These logs are available for reuse.
- **STALE:** This status indicates that some members of the redo log group have become unusable due to file corruption or hardware issues. A redo log in this state may require recovery or manual intervention.

## Queries to Check Redo Log Status

You can use the following queries to check the status of your redo log files:

### 1. View the Redo Log File Status in V\$LOG

```
sql
```

```
SELECT GROUP#, STATUS, ARCHIVED, FIRST_CHANGE#  
FROM V$LOG;
```

### 2. View Redo Log File Members in V\$LOGFILE

```
sql
```

```
SELECT GROUP#, STATUS, MEMBER  
FROM V$LOGFILE;
```

- `V$LOG` shows the status of each redo log group.
- `V$LOGFILE` shows the individual members (redo log files) for each group, including their status.

These statuses help monitor the usage of redo log files and troubleshoot any issues that might arise with redo logs in your Oracle 19c RAC or standalone environment.

## **Conclusion**

Handling redo log file loss in a RAC environment requires careful management, especially if the log is **CURRENT** or **ACTIVE**. The key steps involve clearing the logs, shutting down the affected instance, and recreating the lost redo logs, while ensuring the remaining nodes continue functioning. Future-proofing strategies like multiplexing redo logs and using FRA are critical for avoiding such issues.