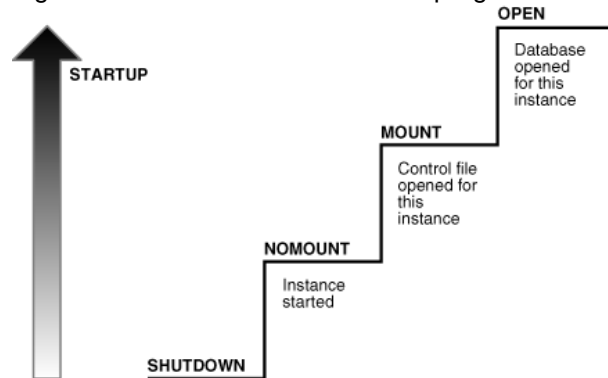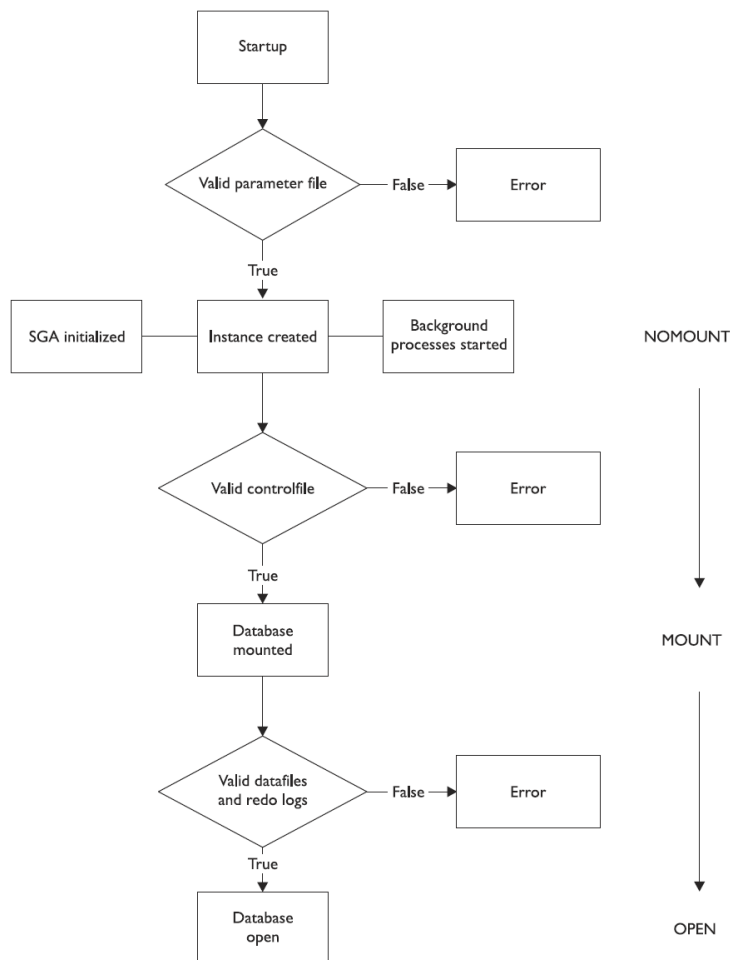# Oracle DB instance startup

In a typical use case, you manually start an instance and then mount and open the database, making it available for users. You can use the SQL*Plus STARTUP command, Oracle Enterprise Manager (Enterprise Manager), or the SRVCTL utility to perform these steps.

Figure 13-3 shows how a database progresses from a shutdown state to an open state.

A database goes through the following phases when it proceeds from a shutdown state to an open database state:

Stage One: Instance start

```
[oracle@Linux1 ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Apr 24 16:23:21 2014

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup nomount
ORACLE instance started.

Total System Global Area  734892032 bytes
Fixed Size                  2216984 bytes
Variable Size             574622696 bytes
Database Buffers          155189248 bytes
Redo Buffers                2863104 bytes
```

When Instance Starts, first thing is searching for a server parameter file in a platform-specific default location and, if not found, for a text initialization parameter file (specifying STARTUP with the SPFILE or PFILE parameters overrides the default behavior), For example:

STARTUP PFILE = /u01/oracle/dbs/spf_init.ora

SPFILE, A server parameter file is a repository for initialization parameters that is managed by Oracle Database. A server parameter file has the following key characteristics:
a. Only one server parameter file exists for a database. This file must reside on the database host.
b. The server parameter file is written to and read by only by Oracle Database, not by client applications.
c. The server parameter file is binary and cannot be modified by a text editor.
d. Initialization parameters stored in the server parameter file are persistent. Any changes made to the parameters while a database instance is running can persist across instance shutdown and startup.
e. A server parameter file eliminates the need to maintain multiple text initialization parameter files for client applications. A server parameter file is initially built from a text initialization parameter file using the CREATE SPFILE statement. It can also be created directly by the Database Configuration Assistant.

A text initialization parameter file is a text file that contains a list of initialization parameters. This type of parameter file, which is a legacy implementation of the parameter file, has the following key characteristics:
f. When starting up or shutting down a database, the text initialization parameter file must reside on the same host as the client application that connects to the database.
g. A text initialization parameter file is text-based, not binary.
h. Oracle Database can read but not write to the text initialization parameter file. To change the parameter values you must manually alter the file with a text editor.
i. Changes to initialization parameter values by ALTER SYSTEM are only in effect for the current instance. You must manually update the text initialization parameter file and restart the instance for the changes to be known.

The text initialization parameter file contains a series of key=value pairs, one per line. For example, a portion of an initialization parameter file could look as follows:
      db_name=sample
      control_files=/disk1/oradata/sample_cf.dbf
      db_block_size=8192
      open_cursors=52
      undo_management=auto
      shared_pool_size=280M
      pga_aggregate_target=29M

After reading the parameter file to determine the values of initialization parameters, instance allocates the SGA based on the initialization parameter settings, starts the Oracle background processes, opens the alert log and traces files and writes all explicit parameter settings to the alert log in valid parameter syntax

Stage Two: Database mount

```
SQL> alter database mount
  2  ;

Database altered.
```

After Instance startup completed, Oracle uses this instance starts to mount database

To mount the database, the instance obtains the names of the database control files specified in the **CONTROL_FILES** initialization parameter and opens the files. Oracle Database reads the control files to find the names of the data files and the online redo log files that it will attempt to access when opening the database.

The control file is the root file that Oracle Database uses to find database files and to manage the state of the database generally. A control file contains information such as the following:
- The database name and database unique identifier (DBID)
- The time stamp of database creation
- Information about data files, online redo log files, and archived redo log files
- Tablespace information
- RMAN backups

The control file serves the following purposes:
It contains information about data files, online redo log files, and so on that is required to open the database.

The control file tracks structural changes to the database. For example, when an administrator adds, renames, or drops a data file or online redo log file, the database updates the control file to reflect this change.

It contains metadata that must be accessible when the database is not open. For example, the control file contains information required to recover the database, including checkpoints. A checkpoint indicates the SCN in the redo stream where instance recovery would be required to begin (see "Overview of Instance Recovery"). Every committed change before a checkpoint SCN is guaranteed to be saved on disk in the data files. At least every three seconds the checkpoint process records information in the control file about the checkpoint position in the online redo log.

Oracle Database reads and writes to the control file continuously during database use and must be available for writing whenever the database is open. For example, recovering a database involves reading from the control file the names of all the data files contained in the database. Other operations, such as adding a data file, update the information stored in the control file.

Stage Three: Database Open

```
SQL> alter database open;

Database altered.
```

After Instance accesses and read database control file successfully, database mounted. In a mounted database, the database is closed and accessible only to database administrators. Administrators can

keep the database closed while completing specific maintenance operations. However, the database is not available for normal operations.

If Oracle Database allows multiple instances to mount the same database concurrently, then the CLUSTER_DATABASE initialization parameter setting can make the database available to multiple instances. Database behavior depends on the setting:

- If CLUSTER_DATABASE is false (default) for the first instance that mounts a database, then only this instance can mount the database.
- If CLUSTER_DATABASE is true for the first instance, then other instances can mount the database if their CLUSTER_DATABASE parameter settings are set to true. The number of instances that can mount the database is subject to a predetermined maximum specified when creating the database.

Opening a mounted database makes it available for normal database operations. Any valid user can connect to an open database and access its information. Usually, a database administrator opens the database to make it available for general use.

When you open the database, Oracle Database performs the following actions:
- Opens the online data files in tablespaces other than undo tablespaces
  If a tablespace was offline when the database was previously shut down (see "Online and Offline Tablespaces"), then the tablespace and its corresponding data files will be offline when the database reopens.
- Acquires an undo tablespace
  If multiple undo tablespaces exist, then the UNDO_TABLESPACE initialization parameter designates the undo tablespace to use. If this parameter is not set, then the first available undo tablespace is chosen.
- Opens the online redo log files

Startup/Shutdown specific credentials: Username: oracle; Password: oracle

Note: For crashed instance startup, instance recovery is nothing more than using the contents of the online log files to rebuild the database buffer cache to the state it was in before the crash. This rebuilding process replays all changes extracted from the redo logs that refer to blocks that had not been written to disk at the time of the crash. Once this has been done, the database can be opened. You can start and shut down individual instances from each instance's home page. However, it is easier to perform instance startup and shutdown operations directly from the Startup/Shutdown: Select Operation page.