



**Northwestern
Polytechnic
University**

In the world of technology, mastering the art of true craftsmanship from relative ease of use to intricate robust hardware and robust software applications are increased to 80% core and primary.

This is the source of much in the necessary technical services we provide have been here.



In the world of technology, mastering the art of true craftsmanship from relative ease of use to intricate robust hardware and robust software applications are increased to 80% core and primary.

This is the source of much in the necessary technical services we provide have been here.

In the world of technology, mastering the art of true craftsmanship from relative ease of use to intricate robust hardware and robust software applications are increased to 80% core and primary.

This is the source of much in the necessary technical services we provide have been here.

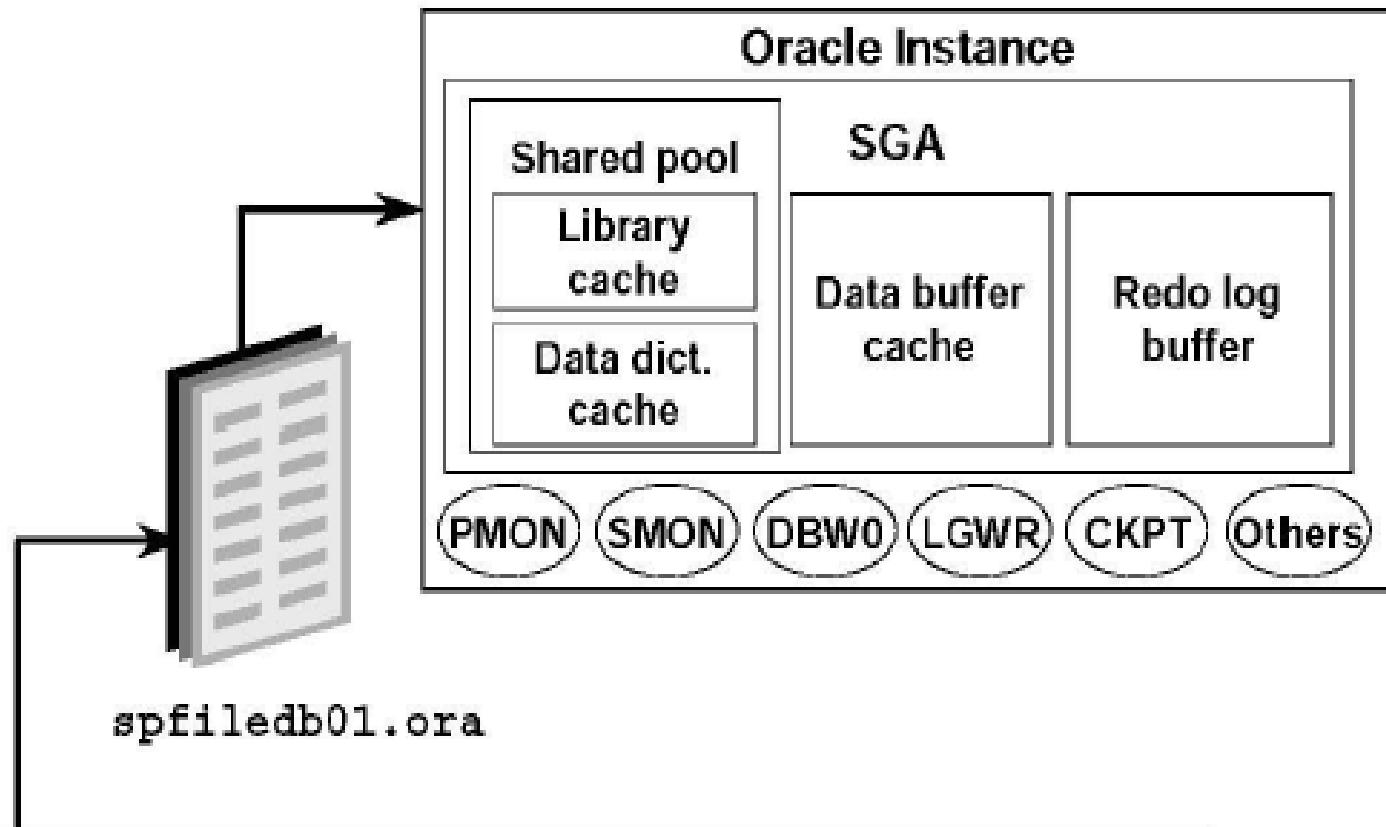
In the world of technology, mastering the art of true craftsmanship from relative ease of use to intricate robust hardware and robust software applications are increased to 80% core and primary.

This is the source of much in the necessary technical services we provide have been here.

**Preparing Business and High-Tech
Professionals and the Leaders of Tomorrow**

Database Administration I

Initialization Parameter Files



```
SQL> CONNECT / AS SYSDBA  
SQL> STARTUP
```

Initialization Parameter Files

- Entries are specific to the instance being accessed
- There are two kinds of parameters:
 - Explicit : Having an entry in the file
 - Implicit : No entry within the file, but assuming the Oracle default values
- Multiple files can be used for a single database to optimize performance in different situations.
- Changes to entries in the file take effect based on the type of initialization parameter file used;
 - Static parameter file, PFILE
 - Persistent parameter file, SPFILE

PFILE initSID.ora

- The PFILE is a text file that can be modified with an operating system editor.
- Modifications to the file are made manually
- Changes to the file take effect on the next startup
- Its default location is \$ORACLE_HOME/dbs.

PFILE Example

```
# Change '<ORACLE_BASE>' to point to the oracle base (the one you specify at
# install time)

db_name='ORCL'
memory_target=1G
processes = 150
audit_file_dest='<ORACLE_BASE>/admin/orcl/adump'
audit_trail ='db'
db_block_size=8192
db_domain=''
db_recovery_file_dest='<ORACLE_BASE>/fast_recovery_area'
db_recovery_file_dest_size=2G
diagnostic_dest='<ORACLE_BASE>'
dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'
open_cursors=300
remote_login_passwordfile='EXCLUSIVE'
undo_tablespace='UNDOTBS1'
# You may want to ensure that control files are created on separate physical
# devices
control_files = (ora_control1, ora_control2)
compatible ='11.2.0'
```

SPFILE - spfileSID.ora

- Binary file with the ability to make changes persistent across shutdown and startup
- Maintained by the Oracle server
- Records parameter value changes made with the ALTER SYSTEM command
- Can specify whether the change being made is temporary or persistent
- Values can be deleted or reset to allow an instance to revert to the default value

`ALTER SYSTEM SET undo_tablespace = 'UNDO2';`

Creating an SPFILE

- SPFILE can be created from an initSID.ora file using the CREATE SPFILE command, which can be executed before after instance startup:

CREATE SPFILE FROM PFILE;

SPFILE Example

```
*.background_dump_dest='$ORACLE_HOME/admin/db01/bdump'
*.compatible='9.0.0'
*.control_files='/u03/oradata/db01/ctrl01db01.ctl','/u03/orad
ata/db01/ctrl02db01.ctl'
*.core_dump_dest='$ORACLE_HOME/admin/db01/cdump'
*.db_block_buffers=500
*.db_block_size=4096
*.db_files=40
*.db_name='db01'
*.instance_name='db01'
*.remote_login_passwordfile='exclusive'
*.shared_pool_size=31457280 # 30M Shared Pool
*.undo_management='AUTO'
db01.undo_tablespace='UNDOTBS01'
db02.undo_tablespace='UNDOTBS02'
* * *
```

Oracle Managed Files

Oracle Managed Files (OMF) simplify file administration

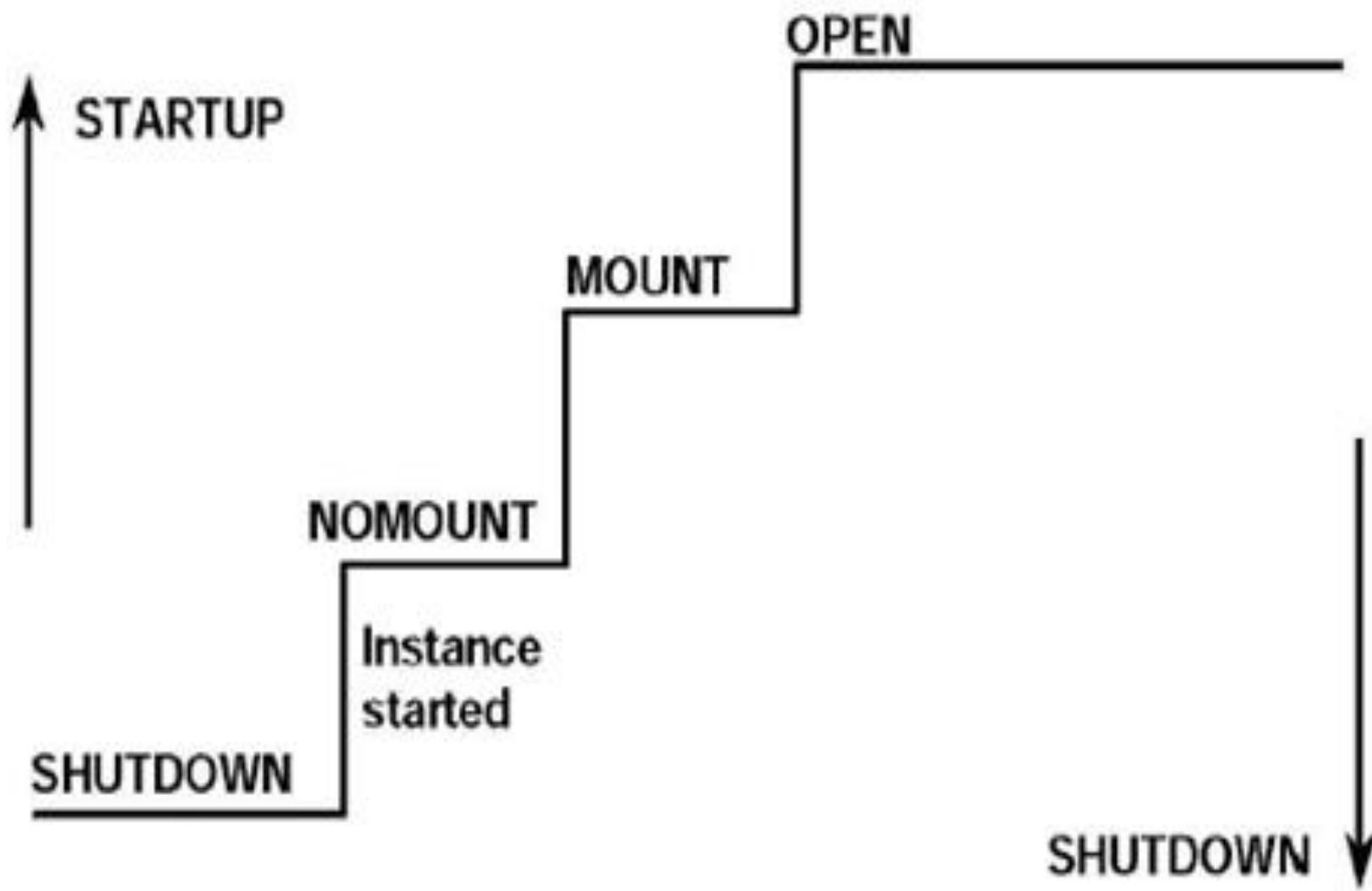
- OMF are created and deleted by the Oracle server as directed by SQL commands
- OMF are established by setting two parameters:
 - **DB_CREATE_FILE_DEST**: Set to give the default location for data files
 - **DB_CREATE_ONLINE_LOG_DEST_N**: Set to give the default locations for online redo logs and control files, up to a maximum of 5 locations

Oracle Managed File Example

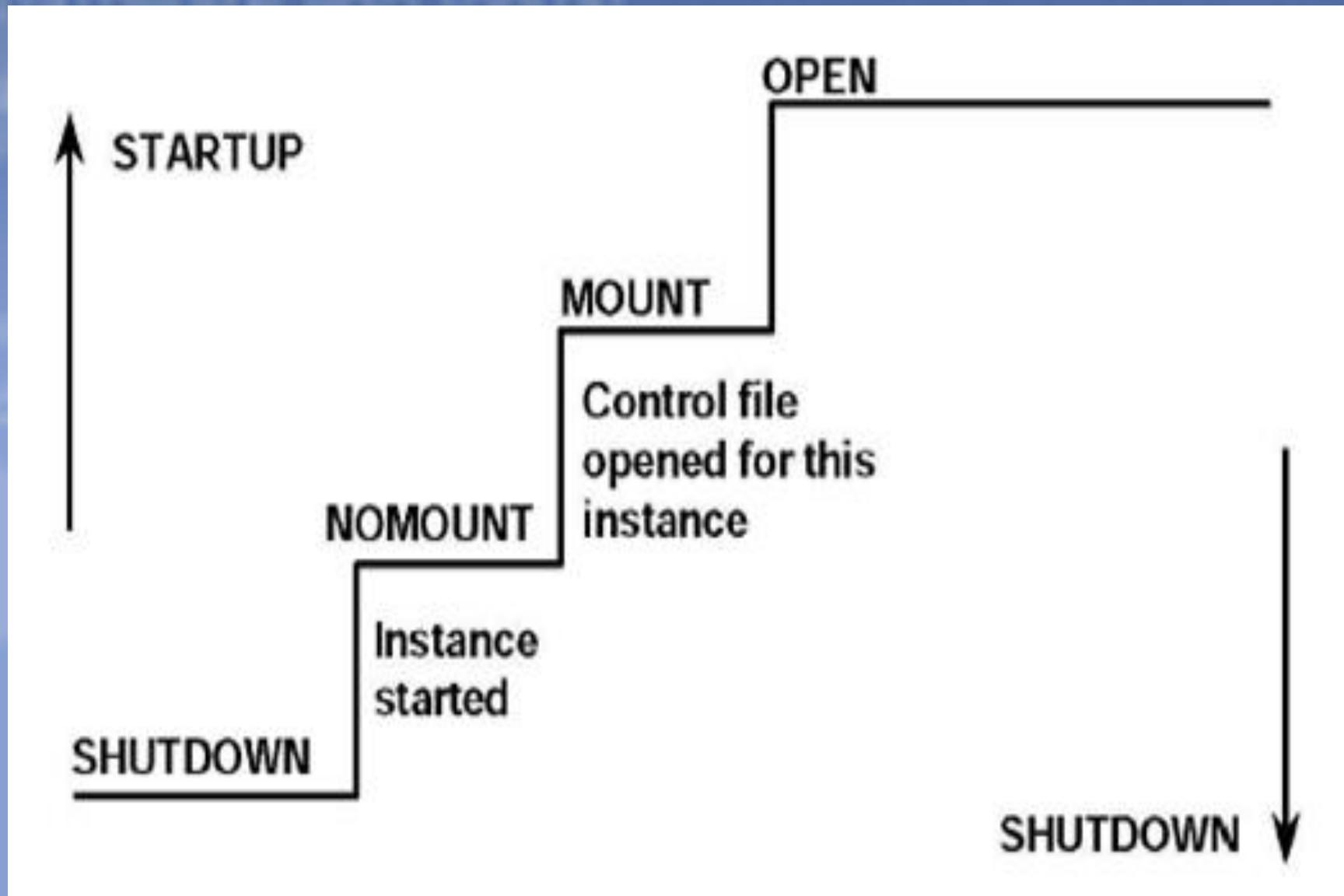
To create a database where data files, control files, and online redo log files are created in separate directories:

- Set the initialization parameters:
 - DB_CREATE_FILE_DEST = '/u01/oradata/'
 - DB_CREATE_ONLINE_LOG_DEST_1= '/u02/oradata/'
 - DB_CREATE_ONLINE_LOG_DEST_2= '/u03/oradata/'
- Issue the CREATE DATABASE SQL statement

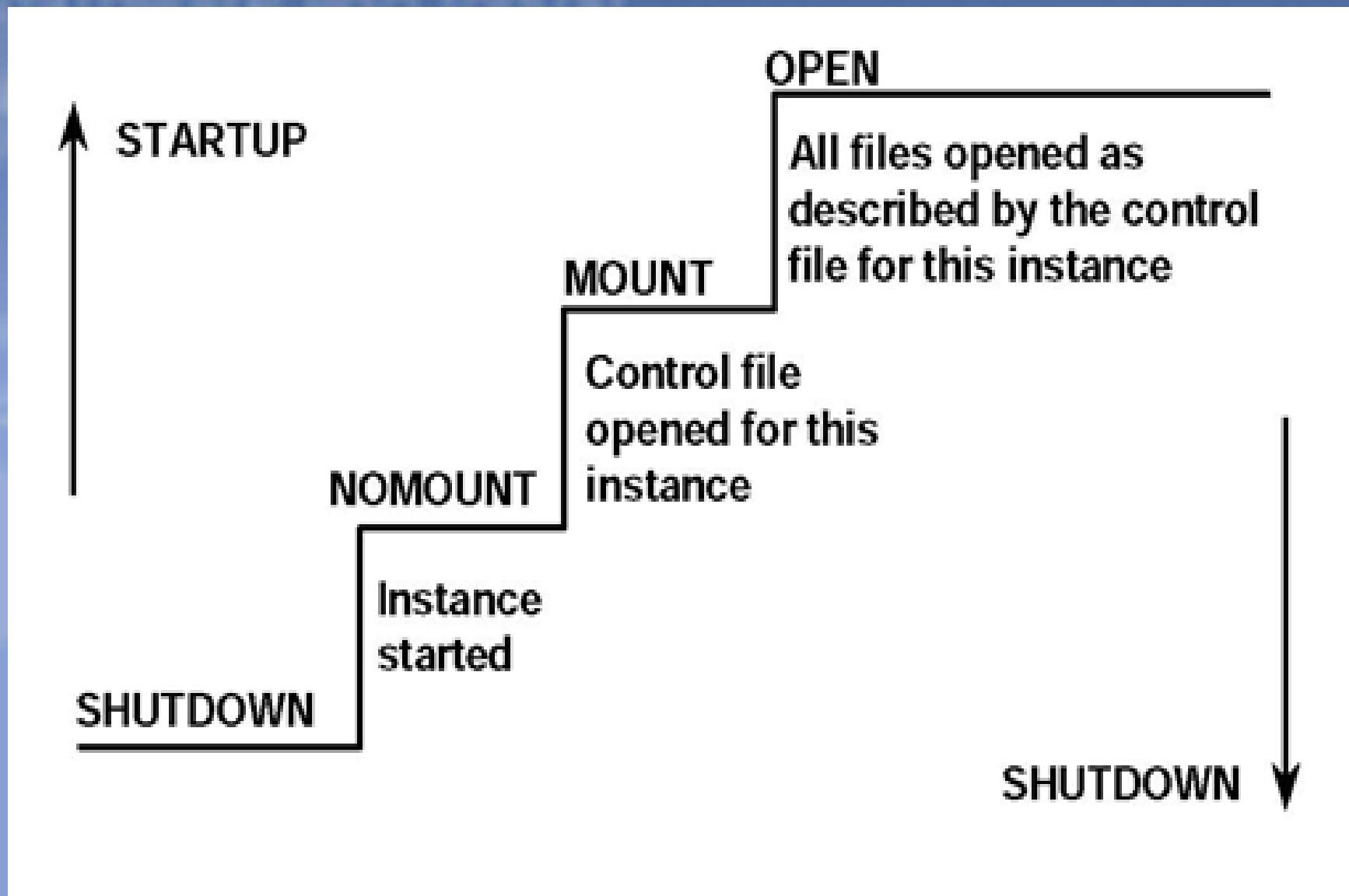
Starting Up a Database NOMOUNT



Starting Up a Database MOUNT



Starting Up a Database OPEN



STARTUP Command

Start up the instance and open the database:

STARTUP

STARTUP PFILE=\$ORACLE_HOME/dbs/initdb01.ora

The ALTER DATABASE Command

- Change the state of the database from NOMOUNT to MOUNT:

Alter database db01 mount;

- Open the database as a read-only database:

ALTER DATABASE db01 OPEN READ ONLY;

Opening Database in Restricted Mode

- Use the **STARTUP** command to restrict access to a database:

STARTUP RESTRICT

- Use the **ALTER SYSTEM** command to place an instance in restricted mode:

ALTER SYSTEM ENABLE RESTRICTED SESSION;

Opening a Database in Read-Only Mode

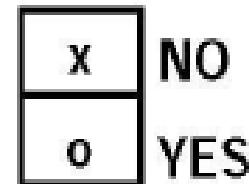
- A database can be opened as read-only database.
- A read-only database can be used to:
 - Execute queries
 - Execute disk sorts using locally managed tablespaces
 - Take data files offline and online, not tablespaces
 - Perform recovery of offline data files and tablespaces

Shutting Down the Database

Shutdown Mode	A	I	T	N
Allow new connections	X	X	X	X
Wait until current sessions end	X	X	X	O
Wait until current transactions end	X	X	O	O
Force a checkpoint and close files	X	O	O	O

Shutdown Mode:

- * NORMAL
- * TRANSACTIONAL
- * IMMEDIATE
- * ABORT



Shutdown Options

On the way down:

- Database buffer cache written to the data files
- Uncommitted changes rolled back
- Resources released.

During a
Shutdown Normal,
Shutdown Transactional
or
Shutdown Immediate

On the way up:

- No instance recovery

Consistent Database
(clean database)

Shutdown Options

On the way down:

- Modified buffers are not written to the data files
- Uncommitted changes are not rolled back

During a
Shutdown Abort
or
Instance Failure
or
Startup Force

On the way up:

- Redo logs used to reapply changes
- Undo segments used to roll back uncommitted changes
- Resources released

Inconsistent Database
(dirty database)

Managing an Instance by Monitoring Diagnostic Files

Diagnostic files contain information about significant events encountered while the instance is operational.

- Used to resolve problems or to better manage the database on a day-to-day basis.
- Several types of diagnostic files exist:
 - alertSID.log file
 - Background trace files
 - User trace files

Alert Log File

The alertSID.log file records the commands and Results of major events while the database is operational.

- It is used for day-to-day operational information or diagnosing database errors.
- Each entry has a time stamp associated with it.
- The DBA manages the alertSID.log file.
- Its location is defined by

BACKGROUND_DUMP_DEST.

Background Trace Files

Background trace files support information errors detected by any background process.

- They are used to diagnose and troubleshoot errors.
- They are created when a background process encounters an errors.
- Their location is defined by

BACKGROUND_DUMP_DEST

User Trace File

A user trace file is produced by the user process connected to the Oracle server through the server process.

- A user trace file contains statistics for traced SQL statements or user error messages.
- It is created when a user encounters user session errors.
- It can also be generated by a you or a server process.
- Its location is defined by `USER_DUMP_DEST`.
- Its size is defined by `MAX_DUMP_FILE_SIZE` and defaults to 10M.

Enabling or Disabling User Tracing

User tracing is enable or disable at the session or instance level by using the following commands and parameter:

- Session level using the ALTER SESSION command:
`ALTER SESSION SET SQL_TRACE = TRUE`
- Session level by executing DBMS procedure:
`dbms_system.SET_SQL_TRACE_IN_SESSION`
- Instance level by setting the initialization parameter:
`SQL_TRACE = TRUE`

Managing and Organizing a Database

- Creating a database is the first step in managing a database system.
- A database may have been created automatically as part of Oracle9i Server installation, or you can create a new one later.
- Oracle Data Migration Assistant is used to migrate from an earlier version of the database.

Creation Prerequisites

To create a new database, you must have the following:

- A privileged account authenticated in one of the following ways:
 - By the operating system
 - Using a password file
- Sufficient memory to start the instance
- Sufficient disk space for the planned database

Planning Database File Locations

- Keep at least two active copies of a database control file on at least two different devices.
- Multiplex the redo log files and put group members on different disks
- Separate data files whose data:
 - Will participate in disk resource contention across different physical disk resources
 - Have different life spans
 - Have different administrative characteristics

Creating a Database

An Oracle database can be created using:

- Oracle Database Configuration Assistant
- The CREATE DATABASE command

Operating System Environment

On Unix, set the following environment variables:

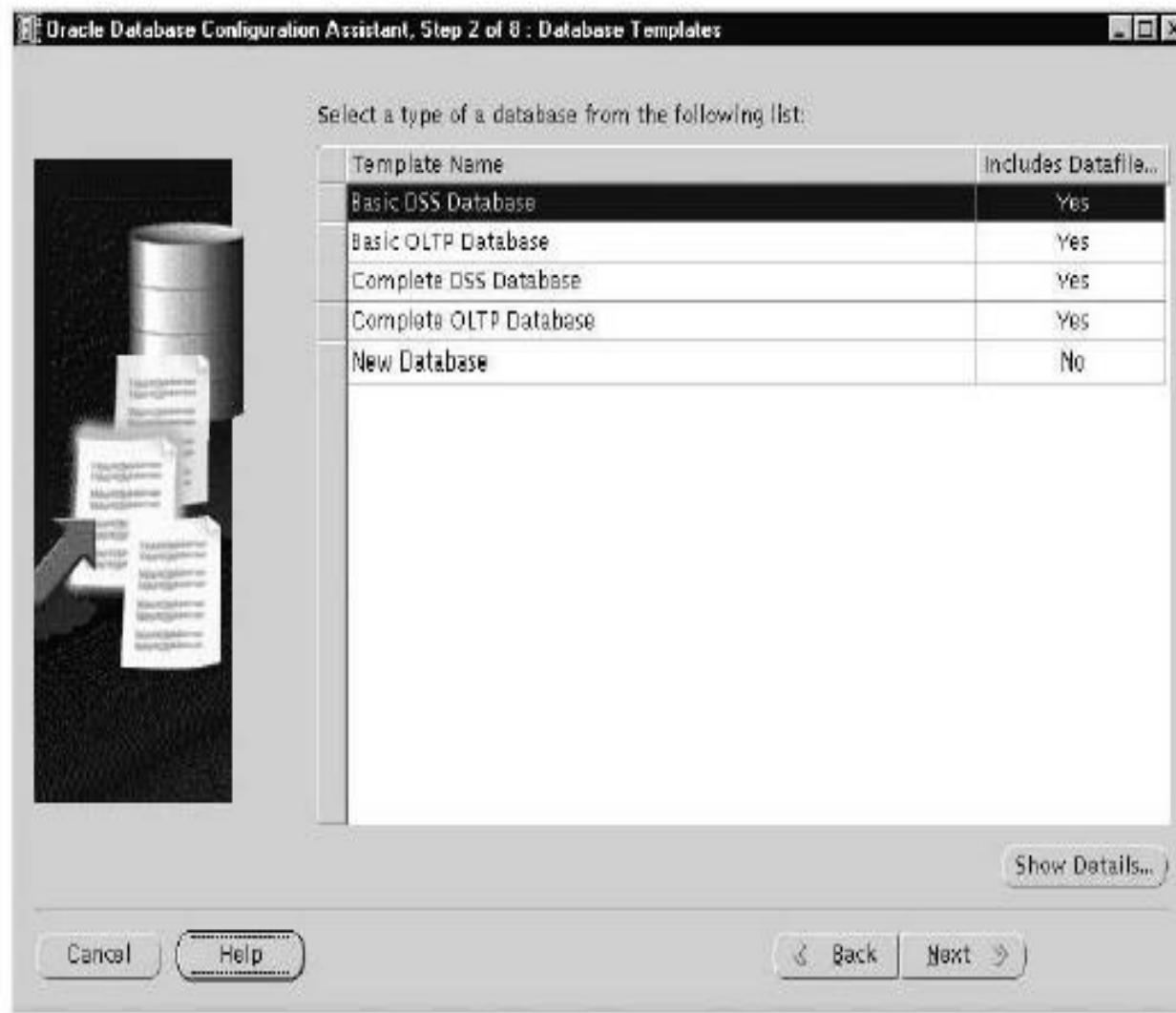
- ORACLE_BASE
- ORACLE_HOME
- ORACLE_SID
- ORA_NLS33
- PATH
- LD_LIBRARY_PATH

Using the Database Configuration Assistant

The Database Configuration Assistant allows you to:

- Create a database
- Configure database options
- Delete a database
- Manage templates
 - Create new template using pre-define template settings
 - Create new template from an existing database
 - Delete database template

Create a Database



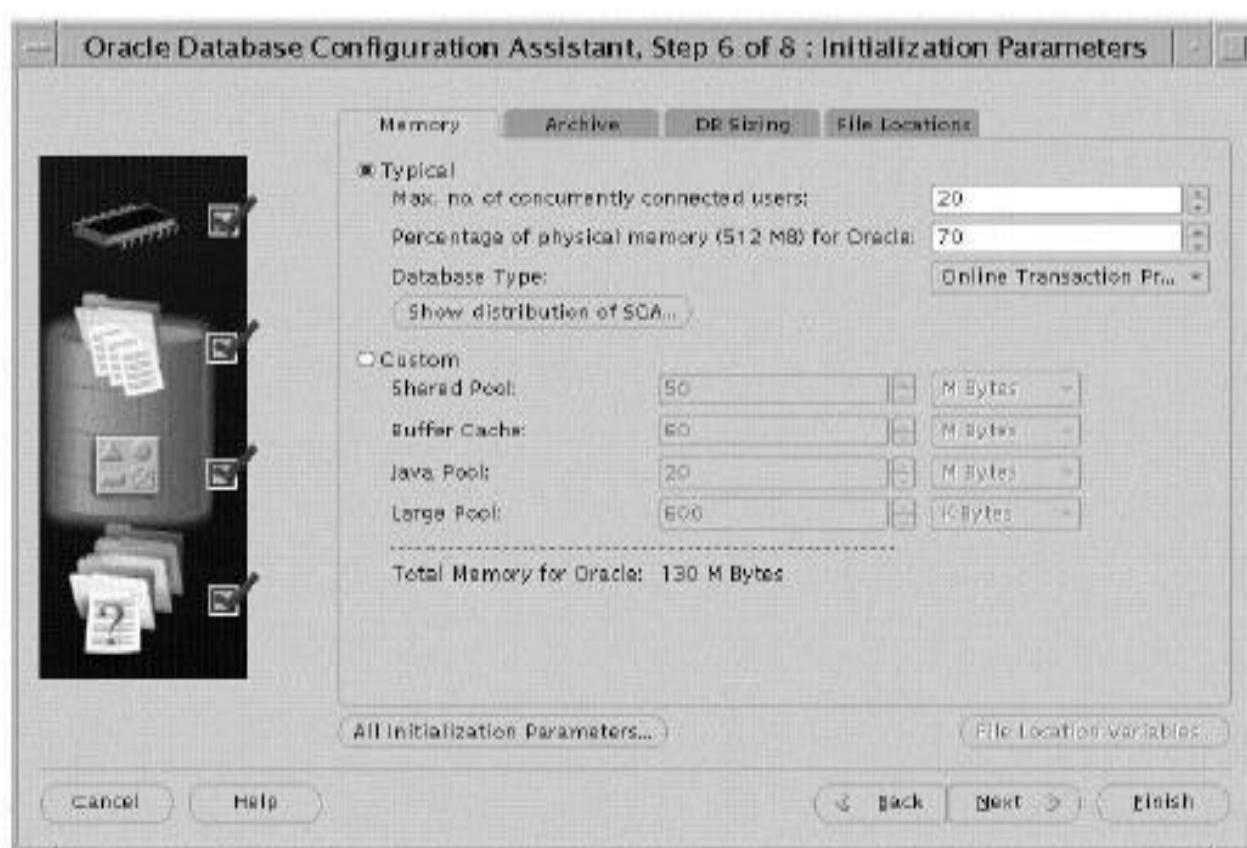
Database Information

Specify:

- **Global database name and SID**
- **The features you want to use for your database,**
- **such as:**
 - Oracle Spatial
 - Oracle OLAP Services
 - Example Schemas
- **Mode in which you want the database to operate**
 - Dedicated server mode
 - Shared server mode

Typical or Custom Install

- Choose between typical or custom install



Other Parameters

- Archive Parameters
 - Use for database recovery
 - May also be used for a standby database
- Data Block Sizing
 - Sets the default database block size
 - Helps to determine the SORT_AREA_SIZE
- File Locations
 - Specify paths for trace files
 - Specify paths for parameter files
- Database storage
 - Specify storage parameters

Complete Database Creation

Complete database creation using the following options:

- Create database
- Save as a database template
- Generate database creation scripts

Creating a Database Manually

- Decide on a unique instance and database name
- Choose a database character set
- Set the operating system variables
- Edit/Create the initialization parameter file
- Start the instance (NOMOUNT)
- Execute the CREATE DATABASE command
- Run scripts to generate the data dictionary and accomplish post creation steps

Preparing the Parameter File

- Create the new initSID.ora

```
$ cp init.ora $ORACLE_HOME/dbs/initdb01.ora
```

- Modify the initSID.ora by editing the parameters

Creating SPFILE

- Create the SPFILE from initSID.ora

CREATE SPFILE FROM PFILE;

Starting the Instance

- Connect as SYSDBA
- Start the instance in NOMOUNT stage

STARTUP NOMOUNT

Creating a Database Using OMF

```
@crdbdb01.sql
SQL> create database db01
  2  logfile
  3    GROUP 1 ('/u01/oradata/db01/log_01_db01.rdo') SIZE 15M,
  4    GROUP 2 ('/u01/oradata/db01/log_02_db01.rdo') SIZE 15M,
  5    GROUP 3 ('/u01/oradata/db01/log_03_db01.rdo') SIZE 15M
  6  datafile '/u01/oradata/db01/system_01_db01.dbf' SIZE 100M
  7  undo tablespace UNDO
  8    datafile '/u01/oradata/db01/undo_01_db01.dbf' SIZE 40M
  9  default temporary tablespace TEMP
 10    tempfile '/u01/oradata/db01/temp_01_db01.dbf' SIZE 20M
 11    extent management local uniform size 128k
 12  character set AL32UTF8
 13  national character set AL16UTF16
 14  set time_zone = 'America/New_York'
 15 ;
```

Creating a Database Using OMF

- Define the OMF initialization parameters in the parameter file

- DB_CREATE_FILE_DEST

- DB_CREATE_ONLINE_DEST_n

STARTUP NOMOUNT

CREATE DATABASE

DEFAULT TEMPORARY TABLESPACE TEMP;

Troubleshooting

- Creation of the database fails if:
 - There are syntax errors in the SQL script
 - Files that should be created already exist
 - Operating system errors such as file or directory permission or insufficient space errors occur

After Database Creation

The database contains:

- Datafiles, control files, and redo log files
- User SYS with the password change_on_install
- User SYSTEM with the password manager
- Internal tables (but no data dictionary views)