**Manual Oracle 11g r2 Database creation on Linux**

* **Overview of Database creation**
* **Set environment variable by editing ‘.DEV.env' file.**
* **Create the Initialization Parameter File**
* **Create necessary directory structure.**
* **Create database creation script**
* **Start the Instance**
* **Run the Database creation script at SQL prompt**
* **Run Scripts to Build Data Dictionary Views**
* **Check the Database name and status**
* **Overview of Database creation**

Using the CREATE DATBASE SQL statement is a more manual approach to creating a database. If you use the CREATE DATABASE statement, you must complete additional actions before you have an operational database. These actions include building views on the data dictionary tables and installing standard PL/SQL packages. You perform these actions by running prepared scripts.

**Specify an Instance Identifier (SID)**

Decide on a unique Oracle system identifier (SID) for your instance, open a command window, and set the ORACLE\_SID environment variable. Use this command windows for the subsequent steps.

The maximum number of characters for ORACLE\_SID is 12, and only letters and numeric digits are permitted. On some platforms, the SID is case-sensitive.

The following example for UNIX and Linux operating systems

Bourne, Bash, or Korn shell:

ORACLE\_SID=DEV

export ORACLE\_SID

**The Required Environment Variables**

Before you can start SQL\*Plus, you may have to set environment variables

For example, ORACLE\_SID and ORACLE\_HOME must be set. In addition, it is advisable to set the PATH variable to include the ORACLE\_HOME/bin directory. On the UNIX and Linux platforms, you must set these environment variables manually.

* **Set environment variable by editing ‘.DEV.env' file.**

[oracle@linux ~]$ pwd

/home/oracle

[oracle@linux ~]$ vi DEV.env

ORACLE\_BASE=/u01/app/DEV; export ORACLE\_BASE

ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0.4; export ORACLE\_HOME

ORACLE\_SID=DEV; export ORACLE\_SID

PATH=$ORACLE\_HOME/bin:$PATH; export PATH

[oracle@linux ~]$

[oracle@linux ~]$. DEV.env

**Create the Initialization Parameter File**

When an Oracle instance starts, it reads an initialization parameter file. This file can be a text file, which can be created and modified with a text editor, or a binary file, which is created and dynamically modified by the database. The binary file, which is preferred, is called a server parameter file. In this step, you create a text initialization parameter file. In a later step, you create a server parameter file from the text file.

go to $ORACLE\_HOME/dbs

[oracle@linux dbs]$ pwd

/u01/app/DEV/product/11.2.0.4/dbs

[oracle@linux dbs]$ vi initDEV.ora

DEV.\_\_db\_cache\_size=247463936

DEV.\_\_java\_pool\_size=4194304

DEV.\_\_large\_pool\_size=8388608

DEV.\_\_oracle\_base='/u01/app/DEV'#ORACLE\_BASE set from environment

DEV.\_\_pga\_aggregate\_target=251658240

DEV.\_\_sga\_target=373293056

DEV.\_\_shared\_io\_pool\_size=0

DEV.\_\_shared\_pool\_size=104857600

DEV.\_\_streams\_pool\_size=0

\*.audit\_file\_dest='/u01/app/DEV/admin/DEV/adump'

\*.audit\_trail='db'

\*.compatible='11.2.0.4.0'

\*.control\_files='/u01/app/DEV/oradata/DEV/control01.ctl','/u01/app/DEV/fast\_recovery\_area/DEV/control02.ctl'

\*.db\_block\_size=8192

\*.db\_domain=''

\*.db\_name='DEV'

\*.db\_recovery\_file\_dest='/u01/app/DEV/fast\_recovery\_area'

\*.db\_recovery\_file\_dest\_size=4385144832

\*.diagnostic\_dest='/u01/app/DEV'

\*.dispatchers='(PROTOCOL=TCP) (SERVICE=DEVXDB)'

#\*.local\_listener='LISTENER\_DEV'

\*.memory\_target=624951296

\*.open\_cursors=300

\*.processes=150

\*.remote\_login\_passwordfile='EXCLUSIVE'

\*.undo\_tablespace='UNDOTBS1'

[oracle@linux dbs]$

- **Create directory structure**

$mkdir -p /u01/app/DEV/oradata/

$mkdir -p /u01/app/DEV/admin/DEV/adump

$mkdir -p /u01/app/DEV/oradata/DEV/

$mkdir -p /u01/app/DEV/fast\_recovery\_area/DEV/

* **Create database creation script**

Before you create the database make sure you have done the planning about the size of the database, number of tablespaces and redo log files you want in the database.

The following script creates database DEV. This database name must agree with the DB\_NAME parameter in the initialization parameter file. This example assumes the following:

The initialization parameter file specifies the number and location of control files with the CONTROL\_FILES parameter.

[oracle@linux ~]$ pwd

/home/oracle

[oracle@linux ~]$ vi DEV.sql

CREATE DATABASE "DEV"

MAXLOGFILES 5

MAXLOGMEMBERS 5

MAXDATAFILES 100

MAXINSTANCES 1

MAXLOGHISTORY 292

LOGFILE

GROUP 1 '/u01/app/DEV/oradata/redo1.log' SIZE 100M,

GROUP 2 '/u01/app/DEV/oradata/redo2.log' SIZE 100M,

GROUP 3 '/u01/app/DEV/oradata/redo3.log' SIZE 100M

DATAFILE

'/u01/app/DEV/oradata/system.dbf' size 100m,

'/u01/app/DEV/oradata/usr04.dbf' size 200m

sysaux datafile '/u01/app/DEV/oradata/sysaux.dbf' size 100m

undo tablespace UNDOTBS1

datafile '/u01/app/DEV/oradata/undo.dbf' size 50m

[oracle@linux ~]$

**Start the Instance**

[oracle@linux ~]$ pwd

/home/oracle

[oracle@linux ~]$. DEV.env

[oracle@linux ~]$cd $ORACLE\_HOME/dbs

[oracle@linux dbs]$sqlplus '/as sysdba'

SQL> startup nomount pfile='initDEV.ora';

ORACLE instance started.

Total System Global Area 622149632 bytes

Fixed Size 2255792 bytes

Variable Size 423625808 bytes

Database Buffers 192937984 bytes

Redo Buffers 3330048 bytes

Create Server parameter file (SPFILE) using this parameter file and STARTUP the instance in NOMOUNT mode

- **Run the Database creation script at SQL prompt**

sql>@/home/oracle/DEV.sql

Database created.

SQL>

**Run Scripts to Build Data Dictionary Views**

Run the scripts necessary to build data dictionary views, synonyms, and PL/SQL packages, and to support proper functioning of SQL\*Plus:

$sqlplus '/as sysdba'

@?/rdbms/admin/catalog.sql

@?/rdbms/admin/catproc.sql

@?/sqlplus/admin/pupbld.sql

CATALOG.SQL Creates the views of the data dictionary tables, the dynamic performance views, and public synonyms for many of the views. Grants PUBLIC access to the synonyms.

CATPROC.SQL Runs all scripts required for or used with PL/SQL.

PUPBLD.SQL Required for SQL\*Plus. Enables SQL\*Plus to disable commands by user.

**Check the Database name and status**

Open new terminal as oracle user

Run env file

$. DEV.env

$sqlplus ‘/as sysdba’

SQL> select name from v$database;

NAME

---------

DEV

SQL> select status from v$instance;

STATUS

------------

OPEN

SQL>