

# **Oracle Database 10g (10.1.0.2) Installation on Fedora Core 1 (FC1)**

This article will take you step by step for Oracle Database 10g (10.1.0.2) installation on Fedora Core 1. Please note that Oracle Database 10g is not certified to run on Fedora Linux, therefore you should not use this combination on a production server. Check the Oracle certification matrix at Metalink for certified platforms.

## **Pre-Installation Steps:**

I am assuming that you have already installed Fedora Core 1. If you need Fedora then visit <http://fedora.redhat.com/>

### **1. Downloading the software and unpacking installation files**

You can [download](#) Oracle Database 10g software from Oracle Technology Network. The following files is required for database installation.

ship.db.lnx32.cpio.gz (649,170,233 bytes) (cksum - 1778052285)

Uncompress the file using gunzip.

```
$ gunzip ship.db.lnx32.cpio.gz
```

Unpack ship.db.lnx32.cpio.

```
$ gunzip ship.db.lnx32.cpio.gz
```

### **2. Checking Memory and Swap Space**

Oracle recommends at least 512MB of RAM and 1GB of swap space or twice the size of RAM on the system. For testing purpose, you can work even with 256MB as well.

To check the size of physical memory, run the following command.

```
$ grep MemTotal /proc/meminfo
```

To check the size of swap space, run the following command.

```
$ grep SwapTotal /proc/meminfo
```

What if you don't have enough swap space?

You can add temporary swap space to your system by creating a temporary swap file instead of using a raw device. Below is the procedure.

Login as root (su - root)

```
$ dd if=/dev/zero of=tmpswap bs=1k count=900000
```

```
$ chmod 600 tmpswap
```

```
$ mkswap tmpswap
```

```
$ swapon tmpswap
```

You can also disable the temporary swap space using the following commands.

Login as root (su - root)

```
$ swapoff tmpswap
```

```
$ rm tmpswap
```

### 3. Checking temporary (/tmp) Space

Oracle recommends that you should have up to 400 MB space.

To check the space in /tmp, run the following command.

```
$ df /tmp
```

What if I don't have enough temporary space?

If you do not have enough space in the /tmp filesystem, you can temporarily create a tmp directory in another filesystem. Below is the procedure to create it.

Login as root (su - root)

```
$ mkdir /<AnotherFilesystem>/tmp # e.g. mkdir /u02/tmp
```

```
$ chown root.root /<AnotherFilesystem>/tmp
```

```
$ chmod 1777 /<AnotherFilesystem>/tmp
```

```
$ export TEMP=/<AnotherFilesystem> # This is used by Oracle
```

```
$ export TMPDIR=/<AnotherFilesystem> # This is used by Linux programs like the linker "ld"
```

After Oracle installation is done, you can remove the temporary /tmp directory. I would recommend to shutdown Oracle first.

Login as root (su - root)

```
$ rmdir /<AnotherFilesystem>/tmp
```

```
$ unset TEMP
```

```
$ unset TMPDIR
```

### 4. Set Kernel Parameters

Perform the following changes in the kernel parameters.

i. Add the following lines to the /etc/sysctl.conf file.

Login as root (su - root)

```
kernel.shmall = 2097152
```

```
kernel.shmmax = 2147483648
```

```
kernel.shmmni = 4096
```

```
# semaphores: semmsl, semmns, semopm, semmni
```

```
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
```

ii. Run the following command to change the current kernel parameters.

```
$ /sbin/sysctl -p
```

iii. Add the following lines to the `/etc/security/limits.conf` file.

```
* soft nproc 2047
* hard nproc 16384
* soft nofile 1024
* hard nofile 65536
```

iv. Add the following line to the `/etc/pam.d/login` file, if it does not already exist.

```
session required /lib/security/pam_limits.so
```

## 5. Fedora Core Packages required for Installation

Install the following packages from Fedora Core 1 Disks.

```
# From Fedora Core 1 Disk 1
```

```
cd /mnt/cdrom/Fedora/RPMS
rpm -Uvh setarch-1.0-1.i386.rpm
```

```
# From Fedora Core 1 Disk 2
```

```
cd /mnt/cdrom/Fedora/RPMS
rpm -Uvh openmotif-2.2.2-16.1.i386.rpm
```

```
# From Fedora Core 1 Disk 3
```

```
cd /mnt/cdrom/Fedora/RPMS
rpm -Uvh compat-libstdc++-7.3-2.96.118.i386.rpm
rpm -Uvh compat-libstdc++-devel-7.3-2.96.118.i386.rpm
rpm -Uvh compat-db-4.0.14-2.i386.rpm
rpm -Uvh compat-gcc-7.3-2.96.118.i386.rpm
rpm -Uvh compat-gcc-c++-7.3-2.96.118.i386.rpm
```

## 6. Create the new groups and users for installation

```
$ groupadd oinstall
```

```
$ groupadd dba
```

```
$ useradd -g oinstall -G dba oracle
```

```
$ passwd oracle
```

## 7. Create the directories in which the Oracle software will be installed

```
$ mkdir -p /u01/app/oracle/product/10.1.0/db_1
```

```
$ chown -R oracle.oinstall /u01
```

## 8. Disable the access control so that clients can connect from any host.

Login as root on the host where you want to install oracle software and issue the following command.

```
$ xhost +<host-name>
```

## 9. Setup the profile for Oracle user.

Login as the oracle [user](#) and add the following lines at the end of the .bash\_profile file.

```
# Oracle User Settings

TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR

ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/10.1.0/db_1; export ORACLE_HOME
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib; export CLASSPATH
#LD_ASSUME_KERNEL=2.4.1; export LD_ASSUME_KERNEL

if [ $USER = "oracle" ]; then
if [ $SHELL = "/bin/ksh" ]; then
ulimit -p 16384
ulimit -n 65536
else
ulimit -u 16384 -n 65536
fi
fi
```

## 10. Change the redhat release information

Edit the /etc/redhat-release file replacing the current release information (Fedora Core release 1 (Yarrow)) with the following. You can copy the original file somewhere else as it will be required after installation.

```
redhat-3
```

## Installation Steps:

The following steps are necessary for Oracle [software installation](#).

### 1. Using SSH

If you are using ssh for installing Oracle software on the remote host then use -X option set set x-forwarding.

```
$ ssh -X oracle@remote_host
```

Now, test it by running xclock command.

```
$ xclock
```

If you do not see clock then check the ssh-config and sshd-config files on the remote host and make sure x-forwarding is enabled.

## 2. Using other X emulation

If you are using other X emulation then set the DISPLAY environmental variable.

```
$ DISPLAY=<machine-name>:0.0; export DISPLAY
```

```
$ echo $DISPLAY
```

```
$ xclock
```

You should see xclock.

## 3. Start the Oracle Universal Installer (OUI)

As Oracle user, go to the Disk1 directory and start the Oracle Universal Installer (OUI) by issuing the following command.

```
$ ./runInstaller
```

Follow the instruction as they appear on the screen. Make sure you use appropriate ORACLE\_HOME.

If you have Oracle 10g Database CD, login as root and mount it.

```
# mount /mnt/cdrom
```

## 4. Create a database using dbca

In the article, I have assumed that you are creating "ORCL" oracle database instance. You can select database creation during the software installation or can do it later.

To invoke database assistant later, use the following command as Oracle user.

```
$ dbca
```

**Note:** This will also required x-forwarding. If you are doing it in a different session then use step 1 or 2 again.

## Post Installation Steps:

Follow the steps given below to complete the installation.

### 1. Change the redhat release information back to the original

Edit the /etc/redhat-release file restoring the original release information. If you have save the file in pre-installation steps 10, then move it back.

### 2. Enable restart flag in the oratab file

Edit the /etc/oratab file setting the restart flag for each instance to 'Y'.

```
ORCL:/u01/app/oracle/product/10.1.0/db_1:Y
```

### 3. Setup auto restart of the databases

i. Create a file called dbora in /etc/init.d directory with the following contents. You can copy and paste the contents.

```
#!/bin/bash
# Oracle auto start-stop script.
# Set ORA_HOME to be equivalent to the $ORACLE_HOME from which you wish to execute
# dbstart and dbshut;
#
# Set ORA_OWNER to the user id of the owner of the
# Oracle database in ORA_HOME.

ORA_HOME=/u01/app/oracle/product/10.1.0/db_1
ORA_OWNER=oracle
if [ ! -f $ORA_HOME/bin/dbstart ]
then
echo "Oracle startup: cannot start"
exit
fi
case "$1" in
'start')
# Start the Oracle databases:
# The following command assumes that the oracle login
# will not prompt the user for any values
su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start"
su - $ORA_OWNER -c $ORA_HOME/bin/dbstart
;;
'stop')
# Stop the Oracle databases:
# The following command assumes that the oracle login
# will not prompt the user for any values
su - $ORA_OWNER -c $ORA_HOME/bin/dbshut
su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop"
;;
esac
```

ii. Set the privileges to 750 for this file.

```
$ chmod 750 /etc/init.d/dbora
```

iii. Create symbolic link to the appropriate run level script directories. I like to have kill script in the rc0 and start script in rc3 run levels.

```
$ ln -s /etc/init.d/dbora /etc/rc0.d/K10dbora
```

```
$ ln -s /etc/init.d/dbora /etc/rc3.d/S99dbora
```