Installing Oracle 10g on a Mandrake 10

Prepared For:
Brody Technology Consulting
Linux Customers'

Prepared By:

Daniel Brody
Brody Technology Consulting
20 Lawnhurst Blvd, Toronto, ON
M6B 3C7 Canada

MAY 2007

Oracle 10g on Mandrake 10

I recently decided to install Oracle on Mandrake 10.

You can get it from http://otn.oracle.com. extract using

1. Run "gunzip " on all the files.

Eg.: gunzip ship.db.cpio.gz

NOTE OTN for a while has carried a ship.db.cpio.gz that is not gzipped. If gzip reports this is not a zip file simply run step 2 directly.

2. Extract the cpio archives with the command "cpio -idcmv < " Eg.: cpio -idcmv < ship.db.cpio

Follow the quick install guide http://download-west.oracle.com/docs/html/B10813_01/toc.htm to the letter (it matters1.1) except

Page 26 - Put the bash/c scripts in an exectable file in /etc/profile.d (see the others that and there and copy the naming).

Wherever you extracted Oracle to go to Disk1/stage/prereq and edit db_server_prereq.xml and change all instances of CertifiedVersions prereq to warning.

```
<PREREQUISITEREF NAME="CertifiedVersions" SEVERITY="Warning"/>
```

and in refhosts.xml add this

```
<OPERATING_SYSTEM>
        <VERSION VALUE="10"/>
        <ARCHITECTURE VALUE="x86"/>
        <NAME VALUE="Linux"/>
        <VENDOR VALUE="Mandrake"/>
        </OPERATING_SYSTEM>
```

In install/oraparam.ini change certified versions to

```
[Certified Versions]
        Linux=redhat-2.1,UnitedLinux-1.0,redhat-3,mandrake-10
```

Next run the installer as in the quickinstall instructions. It should be fooled into thinking Mandrake is a supported platform

Running it as a service

This is loosely based on an old Oracle supplied scipt for this at http://download-uk.oracle.com/docs/html/B10812_01/chapter2.htm#sthref210 but it needs rsh.

Copy $ORACLE_HOME/install/oratab to /etc Edit /etc/oracle and put a Y on the end of instances you want to autostart

put the script below in /etc/profile.d called oracle.sh run chmod a+x oracle.sh

```
export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
 export PATH="$PATH:$ORACLE_HOME/bin"
 export ORACLE_SID=YOUR_SID
```

Put the script below in /etc/init.d and called in oracle chmod u+x on it to make it runabble

```
#1.1  /bin/sh  -x
 #
 # chkconfig: 2345 90 20
 # description: starts and stops Oracle
 #
 # Change the value of ORACLE_HOME to specify the correct Oracle home
 # directory for you installation
 ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1

 #

 # change the value of ORACLE to the login name of the

 # oracle owner at your site

 #

 ORACLE=oracle

 PATH=${PATH}:$ORACLE_HOME/bin

 HOST=`hostname`

 PLATFORM=`uname`

 export ORACLE_HOME PATH

 #

 LOG=$ORACLE_HOME/startup.log

 touch $LOG

 chmod a+r $LOG

 #

 case $1 in

 'start')

        echo "$0: starting up" >> $LOG

        date >> $LOG

        # Start Oracle Net
```

```
        if [ -f $ORACLE_HOME/bin/tnslsnr ] ;
        then
             echo "starting Oracle Net Listener"
             sudo -u $ORACLE $ORACLE_HOME/bin/lsnrctl start >> $LOG 2>&1 &
        fi
        echo "Starting Oracle databases"
        sudo -u $ORACLE $ORACLE_HOME/bin/dbstart >> $LOG 2>&1 &
#UNCOMMENT THIS if you Oracle Enterprise Manager
        #echo "Starting Oracle Enterprise Manager"
        #sudo -u $ORACLE $ORACLE_HOME/bin/emctl start dbconsole >> $LOG
2>&1 &


        ;;
 'stop')
        echo "$0: shutting down" >> $LOG
        date >> $LOG
        # Stop Oracle Net
        if [ -f $ORACLE_HOME/bin/tnslsnr ] ;
        then
             echo "stopping Oracle Net Listener"
             sudo -u $ORACLE $ORACLE_HOME/bin/lsnrctl stop >> $LOG 2>&1 &
        fi
        echo "stopping Oracle databases"
        sudo -u $ORACLE $ORACLE_HOME/bin/dbshut >> $LOG 2>&1 &
#UNCOMMENT THIS if you Oracle Enterprise Manager
        # echo "Stopping Oracle Enterprise Manager"
        # sudo -u $ORACLE $ORACLE_HOME/bin/emctl stop dbconsole >> $LOG
2>&1 &
        ;;
 *)
        echo "usage: $0 {start|stop}"
        exit
```

```
        ;;
esac
#
exit
```

Run chkconfig —add oracle
Oracle will now start on boot

How to install Oracle 10g on Mandrake 10.1 Official

1- Creating Oracle User Accounts

To create the oracle account and groups, execute the following commands:

```
su - root

groupadd dba          # group of users to be granted SYSDBA system
privilege

groupadd oinstall     # group owner of Oracle files

useradd -c "Oracle software owner" -g oinstall -G dba oracle
```

2- Unzip the database by running this command:

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

3- Uncompress the database by running this command:

```
cpio -idmv < ship.db.lnx32.cpio
```

In order to install oracle the system must have at least 512MB of RAM and 1GB of swap space or twice the size of RAM. And for systems with more than 2 GB of RAM, the swap space can be between one and two times the size of RAM.

4- To check the size of physical memory, execute:

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

5- To check the size of swap space, execute:

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

(OPTIONAL)

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

```
su - root

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

chmod 600 tmpswap

mkswap tmpswap

swapon tmpswap
```

To disable the temporary swap space execute the following commands:

```
su - root

swapoff tmpswap

rm tmpswap
```

6- Check how much space is in the /tmp :

According to Oracle's documentation, the Oracle Universal Installer (OUI) requires up to 400 MB of free space in the /tmp directory. But OUI checks if /tmp is only greater than 80 MB.

To check the space in /tmp, run:

```
df /tmp
```

(OPTIONAL)

If you do not have enough space in the /tmp filesystem, you can temporarily create a tmp directory in another filesystem. Here is how you can do this:

```
su - root

mkdir //tmp

chown root.root //tmp

chmod 1777 //tmp

export TEMP=/           # used by Oracle

export TMPDIR=/         # used by Linux programs like the linker "ld"
```

When you are done with the Oracle installation, shutdown Oracle and remove the temporary /tmp directory:

```
su - root

rmdir //tmp

unset TEMP

unset TMPDIR
```

7- Check for required RPMs

```
rpm -q gcc make binutils setarch openmotif mandrakelinux-release
```

This command queries the RPM database if these RPMs are installed or not (gcc, make, binutils, setarch, openmotif, mandrakelinux-release) If not so execute:

```
urpmi gcc

urpmi make

urpmi binutils

urpmi setarch

urpmi openmotif

urpmi mandrakelinux-release
```

8- Make the OUI believe that its installing Oracle 10g on Redhat , because Mandrake is not supported by Oracle.

You have to edit these 2 files (/etc/redhat-release , /etc/mandrakelinux-release) file on Mandrake to make Oracle10g believe it is running on RHELAS3. To change the /etc/redhat-release file, you can simply copy/paste the following commands:

```
su - root

cp /etc/redhat-release /etc/redhat-release.backup

cat > /etc/redhat-release << EOF

Red Hat Enterprise Linux AS release 3 (Taroon)

EOF

su - root

cp /etc/mandrakelinux-release /etc/mandrakelinux-release.backup

cat > /etc/mandrakelinux-release << EOF

Red Hat Enterprise Linux AS release 3 (Taroon)

EOF
```

After you are done with the Oracle10g installation undo the changes you made to /etc/redhat-release:

```
su - root

cp /etc/redhat-release.backup /etc/redhat-release

su - root

cp /etc/mandrakelinux-release.backup /etc/mandrakelinux-release
```

9- Check the kernel parameters

To see all kernel parameters, execute:

```
su - root

sysctl -a
```

For Oracle10g, the following kernel parameters have to be set to values greater than or equal to the recommended values which can be changed in the proc filesystem:

```
shmmax  = 2147483648     (To verify, execute: cat /proc/sys/kernel/shmmax)

shmmni  = 4096           (To verify, execute: cat /proc/sys/kernel/shmmni)

shmall  = 2097152        (To verify, execute: cat /proc/sys/kernel/shmall)

shmmin  = 1              (To verify, execute: ipcs -lm |grep "min seg
size")

shmseg  = 10             (It's hardcoded in the kernel - the default is
much higher)

semmsl  = 250            (To verify, execute: cat /proc/sys/kernel/sem |
awk '{print $1}')

semmns  = 32000          (To verify, execute: cat /proc/sys/kernel/sem |
awk '{print $2}')

semopm  = 100            (To verify, execute: cat /proc/sys/kernel/sem |
awk '{print $3}')

semmni  = 128            (To verify, execute: cat /proc/sys/kernel/sem |
awk '{print $4}')

file-max = 65536         (To verify, execute: cat /proc/sys/fs/file-max)

ip_local_port_range = 1024 65000
                         (To verify, execute: cat
/proc/sys/net/ipv4/ip_local_port_range)
```

I added the following lines to the /etc/sysctl.conf file which is used during the boot process:

```
kernel.shmmax=2147483648

kernel.sem=250 32000 100 128

fs.file-max=65536

net.ipv4.ip_local_port_range=1024 65000
```

Adding these lines to the /etc/sysctl.conf file will cause the system to change these kernel parameters after each boot using the /etc/rc.d/rc.sysinit script which is invoked by /etc/inittab. But in order that these new added lines or settings in /etc/sysctl.conf become effective immediately, execute the following command:

```
su - root
```

```
sysctl -p
```

Explanation for Semaphores:

Semaphores can best be described as counters which are used to provide synchronization between processes or between threads within a process for shared resources like shared memories. System V semaphores support semaphore sets where each one is a counting semaphore. So when an application requests semaphores, the kernel releases them in "sets". The number of semaphores per set can be defined through the kernel parameter SEMMSL.

To see all semaphore settings, run:

```
ipcs -ls
```

The SEMMSL Parameter

This parameter defines the maximum number of semaphores per semaphore set.

Oracle recommends to set SEMMSL to the largest PROCESSES init.ora parameter of any database on the Linux system plus 10. Oracle also recommends to set SEMMSL to a minimum value of 100.

The init.ora parameter PROCESSES specifies the maximum number of operating system processes that can be started by the Oracle instance. In a non MTS environment, Oracle spawns a system user process for each connection. This means that in such an environment the PROCESSES parameter defines the maximum number of simultaneous Oracle connections minus sum of all Oracle background processes. It can also be said that the PROCESSES value should never be greater than SEMMSL.

The SEMMNI Parameter

This parameter defines the maximum number of semaphore sets in the entire Linux system.

Oracle recommends to set SEMMNI to a minimum value of 100.

The SEMMNS Parameter

This parameter defines the total number of semaphores (not semaphore set) in the entire Linux system. A semaphore set can have more than one semaphore, and according to the semget(2) man page, values greater than SEMMSL * SEMMNI makes it irrelevant.

Setting it to a minimum value of 256 is for initial Oracle installation only. Oracle recommends to set SEMMNS to the sum of the PROCESSES parameter for each database on the system, adding the largest PROCESSES twice, and then adding 10 for each DB.

The maximum number of semaphores that can be allocated on a Linux system will be the lesser of: SEMMNS or (SEMMSL * SEMMNI)

Setting SEMMSL and SEMMNI to 100 makes sure that SEMMNS semaphores can be allocated as determined by the above calculation.

The SEMOPM Parameter

This parameter defines the maximum number of semaphore operations that can be performed per semop(2) system call.

The semop(2) function provides the ability to do operations for multiple semaphores with one semop(2) system call. Since a semaphore set can have the maximum number of SEMMSL semaphores per semaphore set, it is often recommended to set SEMOPM equal to SEMMSL.

Oracle recommends to set SEMOPM to a minimum value of 100.

10- Setting semaphores

First of all to avoid wasting your time if you want to undo the following settings take a backup for this file "sysctl.conf" by using this command:

```
cp /etc/sysctl.conf /etc/sysctl.conf.backup
```

Setting the Semaphore Kernel Parameters

```
echo "kernel.sem=250 32000 100 128" >> /etc/sysctl.conf
```

To make the change permanent, add or change the following line in the file /etc/sysctl.conf. This file is used during the boot process.

Alternatively, you can use sysctl(8) to change it:

```
sysctl -w kernel.sem="250 32000 100 128"
```

To see the new updated semaphore settings, run:

```
ipcs -ls
```

11- Setting Shell Limits for the Oracle User (steps 12 & 13)

If you just install a small test database, then you might be ok with the current settings (note that the limits very often vary). But for (larger) production databases, you should increase the following shell limits to the following values recommended by Oracle:

```
nofile = 63536     (To verify, execute: ulimit -n)

nproc  = 16384     (To verify, execute: ulimit -u)
```

The nofile option denotes the maximum number of open file descriptors, and nproc denotes the maximum number of processes available to a single user.

To see all shell limits, execute:

```
ulimit -a
```

12- Setting the file handles

The maximum number of file handles can be changed in the proc file system without reboot:

```
su - root
```

```
echo "63536" > /proc/sys/fs/file-max
```

Alternatively, you can use sysctl(8) to change it:

```
sysctl -w fs.file-max=63536
```

To make the change permanent, add or change the following line in the file /etc/sysctl.conf. This file is used during the boot process.

```
echo "fs.file-max=63536" >> /etc/sysctl.conf
```

13- Setting the shell limits for the Oracle user

Now login to the oracle account again since the changes will become effective for new login sessions only.

```
$ su - oracle

$ ulimit -n

1024

$
```

To change this, you have to edit the file /etc/security/limits.conf as root and add the following lines, respectively:

```
oracle           soft    nofile          4096

oracle           hard    nofile          63536
```

Now login to the oracle account again since the changes will become effective for new login sessions only.

```
$ su - oracle

$ ulimit -n

4096

$
```

The default limit for oracle is now 4096 and the oracle user can increase the number of file handles up to 63536:

```
$ su - oracle

$ ulimit -n

4096

$ ulimit -n 63536
```

```
$ ulimit -n

63536

$
```

To make this change permanent, add "ulimit -n 63536" (for Bash) to the ~oracle/.bash_profile file which is the user startup file for the Bash shell on Mandrake Linux (to verify your shell run: echo $SHELL). To do this you could simply copy/paste the following commands for the oracle's Bash shell:

```
su - oracle

cat >> ~oracle/.bash_profile << EOF

ulimit -n 63536

EOF
```

PAM modules are required for performing login

This is the PAM module that will read the /etc/security/limits.conf file. You need to add the following entry to these files: (1) /etc/pam.d/system-auth (2) /etc/pam.d/sshd (3) /etc/pam.d/su (4) /etc/pam.d/login

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

14- Setting Limits for the Maximum Number of Processes for the Oracle User

To see the current limit of the maximum number of processes for the oracle user, run:

```
su - oracle

ulimit -u
```

Note that the ulimit options are different for other shells.

To change the "soft" and "hard" limits for the maximum number of processes for the oracle user (as ROOT), add the following lines to the /etc/security/limits.conf file:

```
oracle           soft    nproc          2047

oracle           hard    nproc          16384
```

To make this change permanent, add "ulimit -u 16384" (for Bash) to the ~oracle/.bash_profile file which is the user start-up file for the Bash shell on Red Hat Linux (to verify your shell run: echo $SHELL). To do this you could simply copy/paste the following commands for the oracle's Bash shell:

```
su - oracle

cat >> ~oracle/.bash_profile << EOF
```

```
ulimit -u 16384

EOF
```

15- Creating Oracle Directories

For Oracle10g you only need to create the directory for $ORACLE_BASE:

```
su - root

mkdir -p /u01/app/oracle

chown -R oracle.oinstall /u01
```

16- Setting Oracle environments

```
su - oracle

export ORACLE_BASE=/u01/app/oracle

export ORACLE_SID=orcl
```

To have these environment variables set automatically each time you login as oracle, you can add these environment variables to the ~oracle/.bash_profile file which is the user startup file for the Bash shell on Red Hat Linux. To do this you could simply copy/paste the following commands to make these settings permanent for your oracle's Bash shell:

```
su - oracle

cat >> ~oracle/.bash_profile << EOF

export ORACLE_BASE=/u01/app/oracle

export ORACLE_SID=orcl

EOF
```

SID = "System Identifier" whicjh is the global databse name

17- Before you execute runInstaller, make sure the Oracle environment variables are set. You can verify the settings by running the set command:

```
su - oracle

oracle$ set
```