

Installing Oracle Database 10g on Red Hat Enterprise Linux AS 4, 3, 2.1, Red Hat Fedora Core 3 and 1, and on RH 9

The following procedure is a step-by-step guide (Cookbook) with tips and information for installing Oracle Database 10g on Red Hat Linux. Oracle10g is the easiest Oracle on Linux installation so far.

This guide shows how I installed:

- Oracle 10g (10.1.0.3) on Red Hat Enterprise Linux Advanced Server 4 (RHELAS4)
- Oracle 10g (10.1.0.2) on Red Hat Enterprise Linux Advanced Server 3 (RHELAS3)
- Oracle 10g (10.1.0.2) on Red Hat Enterprise Linux Advanced Server 2.1 (RHELAS2.1)
 - Oracle 10g (10.1.0.3) on Red Hat Fedora Core 3 (FC3)
 - Oracle 10g (10.1.0.2) on Red Hat Fedora Core 3 (FC3)
 - Oracle 10g (10.1.0.2) on Red Hat Fedora Core 1 (FC1)
 - Oracle 10g (10.1.0.2) on Red Hat 9 (RH9)

These instructions will also work for Red Hat Enterprise Linux ES 2.1 and 3.

Oracle Database 10g on other Linux Distributions

WhiteBox Linux 3

Works exactly like on RHELAS3.

Mandrake 10.0 Community Edition

Some files/packages were not included with Mandrake 10 which can be retrieved from:

ftp://ftp.rediris.es/sites/carroll.cac.psu.edu/mandrake-devel/cooker/i586/Mandrake/RPMS/setarch-1.3-1mdk.i586.rpm
ftp://194.199.20.114/linux/contrib/openmotif/linux-rpm/openmotif-2.1.30-1_I586.i386.rpm

Validation/Certification

For Validations/Certifications, go to [Oracle's Certification Matrices](#).

This article covers the following subjects and steps:

- * **[Downloading Oracle10g Software and Burning Oracle10g CDs](#)**
- * **[Checking Memory and Swap Space](#)**
- * **[Checking /tmp Space](#)**
- * **[Checking Software Packages \(RPMs\)](#)**
 - [Checking the kernel RPM](#)
 - [Checking glibc RPMs](#)
 - [Checking gcc, make, and binutils RPMs](#)
 - [Checking the openmotif RPM](#)
 - [Checking the setarch RPM](#)
 - [Checking the redhat-release RPM](#)
- * **[Checking Kernel Parameters](#)**
- * **[Sizing Disk Space for Oracle10g](#)**
- * **[Creating Oracle User Accounts](#)**
- * **[Setting Shell Limits for the Oracle User](#)**
- * **[Creating Oracle Directories](#)**
 - [Optimal Flexible Architecture \(OFA\) for 10g \(10.1.0.2\)](#)
- * **[Setting Oracle Environments](#)**
- * **[Installing Oracle10g](#)**
 - [Installing Oracle10g on a Remote Linux Server](#)
 - [Starting Oracle Universal Installer](#)
 - [Using Oracle Universal Installer](#)
 - [Updates after Running Oracle Universal Installer](#)
- * **[Oracle Post-installation Tasks](#)**
 - [Startup and Shutdown of the Oracle10g Database](#)
 - [Shutdown of other Oracle 10g Background Processes](#)
- * **[Tips and Hints for Oracle10g on Linux](#)**
- * **[Oracle10g/Linux Errors and Problems](#)**
- * **[References](#)**

[Downloading Oracle10g Software and Burning Oracle10g CDs](#)

Download Oracle10g for Linux from OTN:

<http://otn.oracle.com/software/products/database/oracle10g/index.html>

NOTE: To install a Oracle Database 10g (without RAC) you only need to download the file `ship.db.lnx32.cpio.gz`.

Compute a cyclic redundancy check (CRC) checksum for the downloaded files and compare the checksum numbers against the numbers posted on OTN's website. For example:

```
cksum ship.db.lnx32.cpio.gz
```

Uncompress the downloaded file(s):

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

Unpack ship.db.lnx32.cpio:

```
$ cpio -idmv < ship.db.lnx32.cpio
Disk1/stage/Components/oracle.server/10.1.0.3.0/1
Disk1/stage/Components/oracle.server/10.1.0.3.0
Disk1/stage/Components/oracle.server
Disk1/stage/Components/oracle.tg/10.1.0.3.0/1/DataFiles
Disk1/stage/Components/oracle.tg/10.1.0.3.0/1
Disk1/stage/Components/oracle.tg/10.1.0.3.0
Disk1/stage/Components/oracle.tg
Disk1/stage/Components/oracle.assistants.dbca/10.1.0.3.0/1/DataFiles/doc.
3.1.jar
Disk1/stage/Components/oracle.assistants.dbca/10.1.0.3.0/1/DataFiles/clas
s.jar
...
```

I executed the following command to burn the Disk1 directory on a CD:

```
# mkisofs -r Disk1 | cdrecord -v dev=0,0,0 speed=20 -
```

(Drives' speed varies; you can get the dev numbers when you execute cdrecord -scanbus).

Checking Memory and Swap Space

Oracle says that 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. You might also want to check out [Sizing Swap Space](#).

For test sake I tried to install an Oracle Database 10g (Type: General Purpose Database) on a little PC with 256MB of RAM and 1 GB of swap space. I was able to get a 10g database up and running on this little PC without a problem.

To check the size of physical memory, execute:

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

To check the size of swap space, execute:

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

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
```

Checking /tmp Space

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
```

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 /<AnotherFilesystem>/tmp
chown root.root /<AnotherFilesystem>/tmp
chmod 1777 /<AnotherFilesystem>/tmp
export TEMP=/<AnotherFilesystem>           # used by Oracle
export TMPDIR=/<AnotherFilesystem>         # 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 /<AnotherFilesystem>/tmp
unset TEMP
unset TMPDIR
```

Checking Software Packages (RPMs)

Before you install a Oracle Database 10g, you need to check the system for required RPMs. For example, when I installed Red Hat Enterprise Linux AS 3 I selected the default installation environment with the default package list. Because of that some required RPMs like gcc, redhat-release, and others were missing.

The installation of Oracle10g on my RHELAS3 and RHELAS4 systems worked fine without installing any compatibility RPMs like compat-gcc, compat-libstdc++, etc.

Checking the kernel RPM

It is important that the kernel version for RHELAS2.1 is 2.4.9-e.25 or higher. To check the kernel version run:

```
uname -r
```

For more information on upgrading the kernel, see [Upgrading the Linux Kernel](#).

Checking glibc RPMs

The glibc version that comes with RHELAS4, RHELAS3, RH9, FC1, and FC3 CDs passed Oracle's "Product-specific Prerequisite" checks.

For RHELAS2.1 I had to upgrade glibc in order to pass Oracle's "Product-specific Prerequisite" checks. Oracle's recommended glibc version is 2.2.4.31.7 or higher. For RHELAS2.1 I downloaded the latest glibc RPMs and I executed the following command:

```
su - root
rpm -Uvh glibc-2.2.4-32.11.i686.rpm glibc-common-2.2.4-32.11.i386.rpm
```

Checking gcc, make, and binutils RPMs

Verify that the following RPMs are installed:

- For RHELAS4 the 10g 10.1.0.3 check will fail for gcc and binutils. Ignore this failed check and proceed.
- For RHELAS3 and RHELAS2.1: make-3.79 or newer
- For RHELAS3 and RHELAS2.1: binutils-2.11.90.0.8-12 or newer
- For RHELAS3: gcc-3.2.3-2 or newer
- For RHELAS2.1: gcc-2.96-108.1 or newer

```
rpm -q gcc make binutils
```

On my RHELAS4 system, the make and binutils RPMs were already installed. However, I had to install a few other RPMs to satisfy dependencies of gcc which was not installed on my system. You can find these RPMs on RHELAS4 disk 3 (you can also use newer package versions).

Note that the 10g 10.1.0.3 OUI Product-specific Prerequisite check will fail for the gcc and binutils versions on RHELAS4. You can ignore this failed check and proceed.

```
su - root
rpm -Uvh gcc-3.4.3-9.EL4.i386.rpm \
      glibc-devel-2.3.4-2.i386.rpm \
      glibc-headers-2.3.4-2.i386.rpm \
      glibc-kernheaders-2.4-9.1.87.i386.rpm
```

On my RHELAS3 system, the make and binutils RPMs were already installed.

However, I had to install a few other RPMs to satisfy dependencies of gcc which was not installed on my system. You can find these RPMs on RHELAS3 disk 3 (you can also use newer package versions).

```
su - root
rpm -Uvh gcc-3.2.3-20.i386.rpm \
      glibc-devel-2.3.2-95.3.i386.rpm \
      glibc-headers-2.3.2-95.3.i386.rpm \
      glibc-kernheaders-2.4-8.34.i386.rpm
```

On RHELAS2.1 system, the binutils and gcc RPMs will be missing if you did not select the "Software Development" package group during the "RedHat 2.1 Advanced Server" installation. To satisfy dependencies, execute the following command:

```
su - root
rpm -ivh gcc-2.96-108.1.i386.rpm \
      binutils-2.11.90.0.8-12.i386.rpm \
      cpp-2.96-108.1.i386.rpm \
      glibc-devel-2.2.4-32.11.i386.rpm \
      kernel-headers-2.4.9-e.3.i386.rpm
```

On my RH9 system, the make RPM was already installed. However, I had to install a few other RPMs to satisfy dependencies of gcc which was not installed on my system (you can also use a newer gcc version). *Note that RH9 is not supported by Oracle, and you will have a hard time to find the right RPM versions for Oracle10g. I tried to use the RPMs that came with the RH9 CDs which are easily available to everyone. I did not have any problems installing an Oracle Database 10g (General Purpose Database) on RH9 with these RPMs:*

```
su - root
rpm -Uvh gcc-3.2.2-5.i386.rpm \
      glibc-devel-2.3.2-5.i386.rpm \
      cpp-3.2.2-5.i386.rpm \
      glibc-kernheaders-2.4-8.10.i386.rpm \
      binutils-2.13.90.0.18-9.i386.rpm
```

On my FC1 system, the make RPM was already installed. However, I had to install a few other RPMs to satisfy dependencies of gcc which was not installed on my system (you can also use a newer gcc version). *Note that FC1 is not supported by Oracle, and you will have a hard time to find the right RPM versions for Oracle10g. I tried to use the RPMs that came with the FC1 CDs which are easily available to everyone. I did not have any problems installing an Oracle Database 10g (General Purpose Database) on FC1 with these RPMs:*

```
su - root
rpm -Uvh gcc-3.3.2-1.i386.rpm \
      glibc-devel-2.3.2-101.i386.rpm \
      cpp-3.3.2-1.i386.rpm \
      glibc-headers-2.3.2-101.i386.rpm \
      glibc-kernheaders-2.4-8.36.i386.rpm \
      binutils-2.14.90.0.6-3.i386.rpm
```

On my FC3 system, the make RPM was already installed. However, I had to install a few

other RPMs to satisfy dependencies of `gcc` which was not installed on my system (you can also use a newer `gcc` version). *Note that FC3 is not supported by Oracle, and you will have a hard time to find the right RPM versions for Oracle10g. I tried to use the RPMs that came with the FC3 CDs which are easily available to everyone. I did not have any problems installing an Oracle Database 10g (General Purpose Database) on FC3 with these RPMs:*

```
su - root
rpm -Uvh gcc-3.4.2-6.fc3.i386.rpm \
        cpp-3.4.2-6.fc3.i386.rpm \
        glibc-devel-2.3.3-74.i386.rpm \
        glibc-headers-2.3.3-74.i386.rpm \
        glibc-kernheaders-2.4-9.1.87.i386.rpm
```

NOTE: *The `gcc` version that comes with FC3 will not pass Oracle's "Product-specific Prerequisite" checks. I ignored this warning on FC3.*

Checking the `openmotif` RPM

For RHELAS4 the 10g 10.1.0.3 check will fail for `openmotif`. Ignore this failed check and proceed.

For RHELAS3 verify that `openmotif-2.2.2-16` or a newer RPM version is installed.

For RHELAS2.1 verify that `openmotif-2.1.30-11` or a newer RPM version is installed.

```
rpm -q openmotif
```

For RHELAS4, execute the following command if `openmotif` is not installed (you can also use a newer package version).

Note that the 10g 10.1.0.3 OUI Product-specific Prerequisite check will fail for the `openmotif` version on RHELAS4. You can ignore this failed check and proceed.

```
su - root
rpm -ivh openmotif-2.2.3-6.RHEL4.2.i386.rpm xorg-x11-deprecated-libs-
6.8.1-23.EL.i386.rpm
```

For RHELAS3, execute the following command if `openmotif` is not installed (you can also use a newer package version):

```
su - root
rpm -ivh openmotif-2.2.2-16.i386.rpm
```

For RHELAS2.1, execute the following command if `openmotif` is not installed (you can also use a newer package version):

```
su - root
rpm -ivh openmotif-2.1.30-11.i386.rpm
```

On RH9, I executed the following command to install `openmotif` (you can also use a newer package version). *Note that RH9 is not supported by Oracle, and you will have a hard time to find the right RPM version for Oracle10g. I tried to use the RPM that came*

with the RH9 CDs which are easily available to everyone. I did not had any problems installing an Oracle Database 10g (General Purpose Database) on RH9 with this RPM.

```
su - root
rpm -Uvh openmotif-2.2.2-14.i386.rpm
```

On FC1, I executed the following command to install `openmotif` (you can also use a newer package version). *Note that FC1 is not supported by Oracle, and you will have a hard time to find the right RPM version for Oracle10g. I tried to use the RPM that came with the FC1 CDs which are easily available to everyone. I did not had any problems installing an Oracle Database 10g (General Purpose Database) on FC1 with this RPM.*

```
su - root
rpm -Uvh openmotif-2.2.2-16.1.i386.rpm
```

On FC3, I executed the following command to install `openmotif` (you can also use a newer package version). *Note that FC3 is not supported by Oracle, and you will have a hard time to find the right RPM version for Oracle10g. I tried to use the RPM that came with the FC3 CDs which are easily available to everyone. I did not had any problems installing an Oracle Database 10g (General Purpose Database) on FC3 with this RPM.*

```
su - root
rpm -Uvh openmotif-2.2.3-6.i386.rpm xorg-x11-deprecated-libs-6.8.1-12.i386.rpm
```

Checking the `setarch` RPM

For RHELAS3, verify that the `setarch-1.3-1` RPM or a newer RPM version is installed on your system:

```
rpm -q setarch
```

The `setarch` utility is new in RHEL4, RHEL3 and Fedora Core. It is used to tell the kernel to report a different architecture than the current one. It is also used to emulate a 3GB virtual address space for applications that don't run properly with a larger virtual address space.

On RHELAS4, the `setarch` RPM should already be installed by default.

For RHELAS3, execute the following command if `setarch` is not installed (you can also use a newer package version):

```
su - root
rpm -Uvh setarch-1.3-1.i386.rpm
```

For RHELAS2.1 and RH9 there is no `setarch` RPM.

For FC1, you can execute the following command if `setarch` is not installed (you can also use a newer package version). But you won't really need it.

```
su - root
```

```
rpm -Uvh setarch-1.0-1.i386.rpm
```

For FC3, you can execute the following command if `setarch` is not installed (you can also use a newer package version). But you won't really need it.

```
su - root
rpm -Uvh setarch-1.6-1.i386.rpm
```

Checking the `redhat-release` RPM

Verify that the `redhat-release` RPM is installed on your Red Hat system:

```
rpm -q redhat-release
```

This RPM is important for RHEL since RHELAS4, RHELAS3, and RHELAS2.1 are Linux releases supported by Oracle. Without this RPM, Oracle 10g OUI won't be able to recognize it as a supported OS. However, the 10g 10.1.0.3 installer does not recognize RHELAS4 as a supported release yet. This means that you will have to edit the `/etc/redhat-release` file, see below.

On RHELAS4, the `redhat-release` RPM should already be installed by default. But note that the 10g 10.1.0.3 installer does not recognize RHELAS4 as a supported release yet. This means you will have to edit the `/etc/redhat-release` file, see below.

For RHELAS3, execute the following command if `redhat-release` was not installed on your system (`redhat-release` can be found on RHELAS3 disk 1):

```
su - root
rpm -ivh redhat-release-3AS-1.i386.rpm
```

For RHELAS2.1, execute the following command if `redhat-release` is not installed on your system (`redhat-release` can be found on RHELAS3 disk 1):

```
su - root
rpm -ivh redhat-release-as-2.1AS-4.noarch.rpm
```

NOTE: On RHELAS4, RH9, and FC you have to change the `/etc/redhat-release` file to make Oracle 10g believe it's running on a supported release.

Regarding RHELAS4, the 10g 10.1.0.3 installer does not recognize RHELAS4 as a supported release yet.

To change the `/etc/redhat-release` file, you can simply copy/paste the following commands:

```
su - root
cp /etc/redhat-release /etc/redhat-release.orig
cat > /etc/redhat-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.orig /etc/redhat-release
```

NOTE: On FC3 you don't have to edit /etc/redhat-release since the OUI is not very clever. If OUI finds e.g. digit number "3" in the /etc/redhat-release file, you pass the "operating system version" check.

I do not recommend to execute "runInstaller -ignoreSysPrereqs" since this will disable other checks you probably don't want to.

Checking 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}')
semnmi = 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)
```

NOTE: Do not change the value of any kernel parameter on a system where it is already

higher than listed as minimum requirement.

On [RHELAS4](#), [RHELAS3](#), [RHELAS2.1](#), [RH9](#), [FC1](#), and [FC3](#) I had to increase the kernel parameters `shmmax`, `semopm`, and `filemax` to meet the minimum requirement.

Oracle also recommends to set the local port range `ip_local_port_range` for outgoing messages to "1024 65000" which is needed for high-usage systems. This kernel parameter defines the local port range for TCP and UDP traffic to choose from.

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
```

For more information on `shmmax`, `shmmni`, `shmmin`, `shmseg`, and `shmall`, see [Setting Shared Memory](#).

For more information on `semmsl`, `semmni`, `semmns`, and `semopm`, see [Setting Semaphores](#).

For more information on `filemax`, see [Setting File Handles](#).

[Sizing Disk Space for Oracle10g](#)

Oracle says that about 2.5 GB of disk space should be reserved for the [Oracle software](#) on Linux.

When I did an Oracle 10g Release 1 (10.1.0.3.0) "General Purpose Database" installation (not including any software from the Oracle Database 10g Companion CD), the [Oracle software](#) used about 1.3 GB of disk space, and the preconfigured "General Purpose Database" (datafiles, etc.) used about 710 MB of disk space.

```
$ du -m -s /u01
1963    /u01
$ du -m -s /u01/app/oracle/oradata
720     /u01/app/oracle/oradata
```

If you also install additional software from the Oracle Database 10g Companion CD, then add at least 1 GB of free disk space.

So if you install Oracle10g Enterprise Edition and additional software from the Oracle Database 10g Companion CD, then you need about 2.5 GB of disk for the Oracle software. And if you also want to add a preconfigured database on the same filesystem, make sure to add another 1 GB of disk space.

NOTE: If you don't put Oracle10g on a separate filesystems, then make sure the root filesystem "/" has enough disk space. You can check the free space of the root filesystem with the following command:

```
df -h /
```

[Using Automatic Storage Management \(ASM\)](#)

For more information on installing and configuring ASM, see [Installing and Configuring Automatic Storage Management \(ASM\) and Disks](#). And for information on how to make use of ASM disk groups when running OUI, see [Installing Oracle Database 10g with Real Application Cluster \(RAC\)](#).

[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
passwd oracle
```

For more information on the "oinstall" group account, see [When to use "OINSTALL" group during install of oracle](#).

[Setting Shell Limits for the Oracle User](#)

Most shells like Bash provide control over various resources like the maximum allowable number of open file descriptors or the maximum number of processes available to a user. For more information on `ulimit` for the Bash shell, see `man bash` and search for `ulimit`.

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 = 65536      (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
```

Oracle's suggestions for setting these shell parameters grants those increased parameters to everyone which is not recommended in secure environments. The following procedure will show how you can increase these parameters for the `oracle` account only.

For more information on `nofile` and how to increase the limit, see [Setting Limits for the Maximum Number of Open File Descriptors for the Oracle User](#). Even though this procedure was written for Oracle9i on RHAS 2.1, it also applies to Oracle10g on RHELAS 2.1, RHELAS 3, and other versions.

For more information on `nproc` and how to increase the limit, see [Setting Limits for the Maximum Number of Processes for the Oracle User](#). Even though this procedure was written for Oracle9i on RHAS 2.1, it also applies to Oracle10g on RHELAS 2.1, RHELAS 3, and other versions.

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
```

But if you want to comply with Oracle's Optimal Flexible Architecture (OFA), then you don't want to place the database files in the `/u01` directory but in another directory/filesystem/disk like `/u02`. This is not a requirement but if you want to comply with OFA, then you might want to create the following directories as well:

```
su - root
mkdir -p /u02/oradata/orcl
chown -R oracle.oinstall /u02
```

In this example, "orcl" stands for the name of the database which will also be the name of the instance. This is typically the case for single instance databases.

Optimal Flexible Architecture (OFA) for 10g (10.1.0.2)

The OFA standard is a guideline created by Oracle to ensure reliable Oracle installations. For Oracle 10g Database, the OFA recommended Oracle home path has changed.

The home path for the first 10g (10.1.0) database installation on a system would be:

```
/u01/app/oracle/product/10.1.0.2db_1
```

If you would install a second Oracle 10g Database 10g (10.1.0) on the same system, the Oracle home directory would be as follows:

```
/u01/app/oracle/product/10.1.0/Db_2
```

If the Oracle10g software is not owned by the user `oracle` but by the user "oraowner", then the path of the Oracle home directory would be:

```
/u01/app/oraowner/product/10.1.0/Db_1  
/u01/app/oraowner/product/10.1.0/Db_2
```

The standard directory name for Oracle10g is "app":

```
/u01/app/oracle/product/10.1.0/Db_1
```

Oracle recommends to use mount points such as `/u01`, `/u02`, etc. which complies with the OFA guidelines. But others can be used, for example:

```
/disk_1/app/oracle/product/10.1.0/Db_1
```

The subtree for database files not stored in ASM disk groups should be named as follows:

```
/u02/oradata/<db_name_1>  
/u02/oradata/<db_name_2>  
/u03/oradata/<db_name_1>  
/u03/oradata/<db_name_2>
```

The mount point `/u01` should be used for the Oracle software only. `/u02`, `/u03`, `/u04` etc. should be used for the database files. The `db_name` stands for the `DB_NAME` initialization parameter which is typically the same as the SID name for single instance databases.

Setting Oracle Environments

Since the Oracle Universal Installer (OUI) "runInstaller" is run from the `oracle` account, some environment variables must be configured for this account before OUI is started.

Execute the following commands for the Bash shell which is the default shell on Red Hat Linux (to verify your shell run: `echo $SHELL`):

```
su - oracle  
export ORACLE_BASE=/u01/app/oracle  
export ORACLE_SID=orcl
```

NOTE: If `ORACLE_BASE` is used, then Oracle recommends that you don't set the `ORACLE_HOME` environment variable but that you choose the default path suggested by the OUI. You can set and use `ORACLE_HOME` after you finished running OUI.

Also, the environment variables `ORACLE_HOME` and `TNS_ADMIN` should not be set. If you've already set these environment variables, you can unset them by running the following commands:

```
unset ORACLE_HOME
unset TNS_ADMIN
```

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
```

Installing Oracle10g

Installing Oracle10g on a Remote Linux Server

If you don't install Oracle on your local system but on a remote server, then you need to relink X to your local desktop. The easiest way to do this is to use the "X11 forwarding" feature of `ssh`. This means that you don't have to run `xhost` and set the `DISPLAY` environment variable.

Here is an example how to make use of the "X11 forward" feature of `ssh`. Simply run the following command from your local desktop:

```
$ ssh -X oracle@oracle_remote_server_name
```

Now when you try to run any GUI tool on the remote server, it will automatically be relinked to your local desktop. If this is not working, verify that the `ForwardX11` setting is not set to "no" in `/etc/ssh/ssh_config` on the remote server:

```
su - root
# grep ForwardX11 /etc/ssh/ssh_config | grep -v "^#"
ForwardX11 yes
#
```

If you are using `telnet`, however, you will have to set `DISPLAY` manually, see my other article [Starting runInstaller](#) for more information.

Starting Oracle Universal Installer

Insert the Oracle CD that contains the image of the downloaded file `ship.db.lnx32.cpio`, or change to the directory that contains the image directory `Disk1`.

If you install Oracle10g from a CD, mount the CD by running the following commands in another terminal:

```
su - root
mount /mnt/cdrom
```

For RHELAS4, RHELAS3, and FC3 you have to execute:

```
su - root
mount /media/cdrom
```

Before you execute `runInstaller`, make sure the Oracle environment variables are set, see [Setting Oracle Environments](#). You can verify the settings by running the `set` command:

```
su - oracle
oracle$ set
```

To execute `runInstaller` from the mounted CD, run the following command as the `oracle` user:

```
oracle$ /mnt/cdrom/runInstaller
```

On RHELAS4, RHELAS3, and FC3 run the following command:

```
oracle$ /media/cdrom/runInstaller
```

Using Oracle Universal Installer (OUI)

The following example shows how to install Oracle10g Database Software and a "General Purpose" database:

```
- Welcome Screen:
- Basic Installation:      Checked it which is the
default
- Oracle Home Location:   Use default: /
u01/app/oracle/product/10.1.0/Db_1
- Installation Type:      I used the default:
Enterprise Edition
- UNIX DBA Group:         Use default: dba
- Create Starter Databases: I checked it for this
example which is the default
- Global Database Name:    orcl
- Database password:      Type in the password
for SYS, SYSTEM, SYSMAN, and DBSNMP accounts
- Advanced Installation:  For this example I did
not check it
Click Next
```

- Specify Inventory directory and credentials:
 - Full path of the inventory directory: Use default: /u01/app/oracle/oraInventory
 - Specify Operating System group name: Use default: oinstall
 Click Next
- A window pops up to run the oraInstRoot.sh script:
 - Run the script in another terminal:
 - su - root
 - # /u01/app/oracle/oraInventory/oraInstRoot.sh
 - Creating the Oracle inventory pointer file (/etc/oraInst.loc)
 - Changing groupname of /u01/app/oracle/oraInventory to oinstall.
 - #
 Click Continue
- Product-specific Prerequisite Checks:
 - Verify that all checks have been passed.
 - Make sure that the status of each Check is set to "Succeeded".**
 - On RHELAS4 ignore the warnings for binutils, gcc, and openmotif and proceed.**
 - If a check failed, see [Oracle10g Installation Errors on Linux](#).
 - Note that the "Retry" button doesn't work after you fixed one of the failed checks.
 Click Next
- Select Database Configuration:
 - I selected "General Purpose".
 Click Next
- Specify Database Configuration Options:
 - Global Database Name: I used "orcl".
 - SID: I used "orcl".
 Click Next
- Select Database Management Option:
 - I selected "Use Database Control for Database Management".
 Click Next
- Specify Database File Storage Option:
 - I selected "File System".
 - File System
 - Specify Database file location: /u01/app/oracle/oradata/
 - If you want to comply with OFA, you might want to select another mount point than '/u01', e.g. /u02/oradata.
 Click Next
- Specify Backup and Recovery Options:
 - For my test installation I selected "Do not enable

Automated Backups".

Click Next

- Specify Database Schema Passwords:

Make sure that the password(s) don't start with a digit number! Otherwise you will later get error message(s) like "ORA-00988 missing or invalid password".

Click Next

- Summary:

Click Install

If Enterprise manager configuration fails due to port allocation problems, check out [Oracle10g/Linux Errors and Problems](#).

When a window pops up to run the root.sh script, execute the script in another terminal as root:

```
su - root
# /u01/app/oracle/product/10.1.0/Db_1/root.sh
Running Oracle10 root.sh script...
\nThe following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /
u01/app/oracle/product/10.1.0/Db_1
```

Enter the full pathname of the local bin directory: [/usr/local/bin]:

```
Copying dbhome to /usr/local/bin ...
Copying oraenv to /usr/local/bin ...
Copying coraenv to /usr/local/bin ...
```

\nCreating /etc/oratab file...
Adding entry to /etc/oratab file...
Entries will be added to the /etc/oratab file as

needed by

Database Configuration Assistant when a database

is created

Finished running generic part of root.sh script.
Now product-specific root actions will be

performed.

```
/var/opt/oracle does not exist. Creating it now.
/etc/oracle does not exist. Creating it now.
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp
```

'root'..

```
Operation successful.
Oracle Cluster Registry for cluster has been
```

initialized

```
Adding to inittab
Checking the status of Oracle init process...
Expecting the CRS daemons to be up within 600
```

seconds.

```

        CSS is active on these nodes.
            mars
        CSS is active on all nodes.
        Oracle CSS service is installed and running under
init(1M)
        #
        Click OK

- End of Installation:
        Click Exit

```

Updates after Running Oracle Universal Installer

After Oracle10g has been installed, make sure that ORACLE_HOME, PATH, and LD_LIBRARY_PATH are set for the oracle account.

*Note that the path for ORACLE_HOME might be different on your system!
Also note that LD_LIBRARY_PATH is needed for some Oracle binaries such as sysresv!*

For 10g (10.1.0.3) on my system:

```

su - oracle
$ cat >> ~oracle/.bash_profile << EOF
export ORACLE_HOME=\$ORACLE_BASE/product/10.1.0/Db_1
export PATH=$PATH:\$ORACLE_HOME/bin export
LD_LIBRARY_PATH=\$ORACLE_HOME/lib
EOF
. ~oracle/.bash_profile

```

This command will add the environment variables to the ~oracle/.profile and source in the file for the current shell by executing ". ~oracle/.bash_profile".

NOTE: **Do not add a trailing "/" on the ORACLE_HOME environment variable.** Otherwise you will get the error "ORACLE not available" when you try to connect to sys, see [Oracle10g/Linux Errors and Problems](#) for more information.

Oracle Post-installation Tasks

Before you continue, make sure you followed the steps at [Updates after Running Oracle Universal Installer](#).

Startup and Shutdown of the Oracle10g Database

To startup the database:

```

oracle$ sqlplus /nolog

```

```
SQL> connect / as sysdba
SQL> startup
```

To shutdown the database:

```
oracle$ sqlplus /nolog
SQL> connect / as sysdba
SQL> shutdown
```

The slash connects you to the schema owned by SYS. In the above example you will be connected to the schema owned by SYS with the privilege SYSDBA. SYSDBA gives you the following privileges:

- sysoper privileges WITH ADMIN OPTION
- create database
- recover database until

Shutdown of other Oracle 10g Background Processes

If you installed a preconfigured database using OUI, then several Oracle background processes are now running on your server. Execute the following command to see the background processes:

```
ps -ef
```

To shutdown the Oracle background processes after an Oracle Database 10g installation, you can execute the following commands:

- iSQL*Plus

To stop iSQL*Plus, run:

```
su - oracle
isqlplusctl stop
```

- Database Management Processes

During the installation of Oracle 10g, OUI offered two Database Management Options:

If you selected "Database Control for Database Management", then the Oracle Enterprise Manager Database Control (Database Control) can be shutdown with the following command which stops both the agent and the Oracle Containers for Java (OC4J) management service:

```
su - oracle
emctl stop dbconsole
```

If you selected "Grid Control for Database Management" which is used for full "Grid Control" installations, then the Oracle Management Agent (standalone agent) for the Oracle Enterprise Manager Grid Control (Grid Control) can be stopped with the following command:

```
su - oracle
emctl stop agent
```

- Oracle Net Listener

To stop the listener, run:

```
su - oracle
lsnrctl stop
```

- Cluster Synchronization Services (CSS)

To shutdown Oracle CSS daemon, run:

```
su - root
/etc/rc.d/init.d/init.cssd stop
```

Tips and Hints for Oracle10g on Linux

- To reinstall Oracle10g after a failed installation attempt, you might want to execute the following commands.

Make sure you first used the De-installation option in OUI.

```
su - root
```

```
export ORACLE_HOME=/u01/app/oracle/product/10.1.0/Db_1
. $ORACLE_HOME/bin/localconfig delete      # stops the Oracle CSS
daemon and deletes configuration
```

```
rm -rf /u01/app/oracle/*
```

```
rm -f /etc/oraInst.loc /etc/oratab /etc/oracle
rm -f /etc/inittab.cssd
rm -f /usr/local/bin/coraenv /usr/local/bin/dbhome /
usr/local/bin/oraenv
```

Make also sure to `unset` and uncomment `ORACLE_HOME` from `~oracle/.bash_profile`.

Oracle10g/Linux Errors and Problems

Here is a list of common Oracle10g installation problems and other issues.

Note that most of the issues are due to not following correctly the installation procedure. And some errors are due to not using an Oracle supported Linux OS.

The Installation log file can be found in `$ORACLE_BASE/oraInventory/logs`.
The Database Creation log file can be found in

```
$ORACLE_BASE/admin/$ORACLE_SID/create.
```

```
Starting Oracle Universal Installer...
```

```
Checking installer requirements...
```

```
Checking operating system version: must be redhat-2.1, UnitedLinux-1.0  
or redhat-3
```

```
Failed <<<<
```

```
Exiting Oracle Universal Installer, log for this session can be found  
at ...
```

At the time of this writing, Oracle10g (10.1.0) is supported on Red Hat Enterprise Linux 2.1 and 3, and on UnitedLinux 1.0.

If you get this error on Red Hat Enterprise Linux 3, then the `redhat-release` RPM is not installed. To rectify this problem, execute the following command:

```
su - root  
rpm -ivh redhat-release-3AS-1.i386.rpm openmotif-2.2.2-16.i386.rpm
```

For all other Linux systems that are not supported by Oracle, you have to edit the `/etc/redhat-release` file to make Oracle10g believe it is running on a supported Linux operating system. For RH9 and FC1 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
```

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
```

I do not recommend to execute "`runInstaller -ignoreSysPrereqs`" since this will disable other checks you probably don't want to.

```
Checking for gcc-2.96; found Not found. Failed <<<<
```

See [Checking gcc, make, and binutils RPMs](#).

Note that "Retry" in the "Product-specific Prerequisite Checks" window does not work. So you either set it manually to Passed or you restart OUI.

```
Checking for openmotif-2.1.30-11; found Not found. Failed <<
```

See [Checking the openmotif RPM](#).

Note that "Retry" in the "Product-specific Prerequisite Checks" window does not work. So you either set it manually to Passed or you restart OUI.

```
Checking for shmmax=2147483648; found shmmax=33554432. Failed <<<<
```

Increase the `shmmax` kernel parameter.

For more information on `shmmax`, see [Checking Kernel Parameters](#).

Note that "Retry" in the "Product-specific Prerequisite Checks" window does not work. So you either set it manually to Passed or you restart OUI.

```
Checking for semopm=100; found semopm=32. Failed <<<<
```

Increase the `semopm` kernel parameter.

For more information on `semopm`, see [Checking Kernel Parameters](#).

Note that "Retry" in the "Product-specific Prerequisite Checks" window does not work. So you either set it manually to Passed or you restart OUI.

```
Checking for filemax=65536; found filemax=26163. Failed <<<<
```

Increase the `file-max` kernel parameter:

For more information on `file-max`, see [Checking Kernel Parameters](#).

Note that "Retry" in the "Product-specific Prerequisite Checks" window does not work. So you either set it manually to Passed or you restart OUI.

```
ORA-01034: ORACLE not available
ORA-27101: shared memory realm does not exist
Linux Error: 2: No such file or directory or
ORA-01034: ORACLE not available
```

First check if `ORACLE_SID` is set correctly.

If `ORACLE_SID` is set correctly, then you probably have a trailing slash "/" on the `ORACLE_HOME` environment variable. Remove it and try again to connect to sys.

```
ORA-00988 missing or invalid password(s).
```

During the Oracle10g installation you probably provided a password for the Oracle database accounts that started with a digit number. Ignore this error message and change the password when you are done with the Oracle10g installation.

```
$ sysresv -i
```

```
sysresv: error while loading shared libraries: libcintsh.so.10.1:
cannot open shared object file: No such file or directory
```

Make sure LD_LIBRARY_PATH is set to \$ORACLE_HOME/lib:

```
oracle$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

```
X11 connection rejected because of wrong authentication.
X connection to localhost:10.0 broken (explicit kill or server
shutdown).
```

To rectify this problem, try to login to the remote Oracle server again by using the "X11 forward" feature of ssh. Execute the following command from your local desktop:

```
$ ssh -X oracle@oracle_remote_server_name
```

Now when you try to run any GUI tool on the remote server, it will automatically be relinked to your local desktop. If this is not working, verify that the ForwardX11 setting is not set to "no" in /etc/ssh/ssh_config on your remote server:

```
su - root
# grep ForwardX11 /etc/ssh/ssh_config | grep -v "^#"
ForwardX11 yes
#
```

If you are using telnet, however, you will have to set DISPLAY manually, see my other article [Starting runInstaller](#) for more information.

- **Recovery Manager rman hangs**

You are probably running the wrong rman binary which belongs to the XFree86-devel RPM:

```
$ which rman
/usr/X11R6/bin/rman
```

```
ORA-00988 missing or invalid password(s).
```

During the Oracle10g installation you probably provided a password for the Oracle database accounts that started with a digit number. Ignore this error message and change the password when you are done with the Oracle10g installation.

```
$ ./runInstaller
...
```

```
Exception in thread "main" java.lang.UnsatisfiedLinkError: /
tmp/OraInstall2004-12-13_07-01-19PM/jre/1.4.2/lib/i386/libawt.so:
libXp.so.6: cannot open shared object file: No such file or
directory.
```

If you are missing a shared library, simply run the following command to find out

which RPM you are missing:

```
# rpm -q --whatprovides libXp.so.6
xorg-x11-deprecated-libs-6.8.1-12.FC3.21
```

In this example install the xorg-x11-deprecated-libs RPM.

```
$ lsnrctl start
...
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC)))
TNS-12547: TNS:lost contact
TNS-12560: TNS:protocol adapter error
TNS-00517: Lost contact
Linux Error: 104: Connection reset by peer
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=centauri)
(PORT=1521)))
TNS-12547: TNS:lost contact
TNS-12560: TNS:protocol adapter error
TNS-00517: Lost contact
Linux Error: 104: Connection reset by peer
```

Make sure the loopback entry in `/etc/hosts` is not missing when you start the listener:

```
127.0.0.1          localhost.localdomain  localhost
```

Now try to run `lsnrctl start` as oracle again.