

## Installation of Oracle 10g Release 2 (10.2.0.1.0) on RedHat EL 3, 4, 5 and (Oracle) Enterprise Linux 4, 5.

---

This paper (HOWTO) describes step-by-step installation of Oracle 10g R2 database software on RedHat Enterprise Server 3, 4, 5 and (Oracle) Enterprise Linux 4, 5. This article is useful for Centos Linux release 3, 4 and 5 and for White Box Enterprise Linux release 3 and 4. Note that Centos and White Box distributions are not certified by Oracle Corporation.

This article does not cover database creation process, and ASM Instance creation process.

This paper covers following steps:

- [Pre-Installation Tasks](#)
- [Download & Install](#)
- [Post-Installation Tasks](#)
- [Common Installation Errors](#)

### Pre-Installation Tasks

#### 1. Create oracle User Account

Login as root and create te user oracle which belongs to dba group.

```
su -  
# groupadd dba  
# useradd -g dba oracle
```

#### 2. Setting System parameters

Edit the /etc/sysctl.conf and add following lines:

```
kernel.shmall = 2097152  
kernel.shmmax = 2147483648  
kernel.shmmni = 4096  
kernel.sem = 250 32000 100 128  
fs.file-max = 65536  
net.ipv4.ip_local_port_range = 1024 65000  
net.core.rmem_default = 262144  
net.core.rmem_max = 262144  
net.core.wmem_default = 262144
```

```
net.core.wmem_max = 262144
```

Note: You need reboot system or execute "sysctl -p" command to apply above settings.

For RedHat (OEL, Centos, WBL) 3 and 4 versions: Edit the /etc/pam.d/login file and add following line:

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

For RedHat (OEL, Centos) 5 version: Edit the /etc/pam.d/login file and add following line:

```
session required pam_limits.so
```

Edit the /etc/security/limits.conf file and add following lines:

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

### 3. Creating oracle directories

```
# mkdir /opt/oracle
# mkdir /opt/oracle/102
# chown -R oracle:dba /opt/oracle
```

### 4. Setting Oracle Enviroment

Edit the /home/oracle/.bash\_profile file and add following lines:

Use this settings for 32bit (x86) architecture.

```
ORACLE_BASE=/opt/oracle
ORACLE_HOME=$ORACLE_BASE/102
ORACLE_SID=ORCL
LD_LIBRARY_PATH=$ORACLE_HOME/lib
PATH=$PATH:$ORACLE_HOME/bin

export ORACLE_BASE ORACLE_HOME ORACLE_SID LD_LIBRARY_PATH PATH
```

Use this settings for 64bit (x86\_64) architecture.

```
ORACLE_BASE=/opt/oracle
ORACLE_HOME=$ORACLE_BASE/102
```

```
ORACLE_SID=ORCL
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32
PATH=$PATH:$ORACLE_HOME/bin

export ORACLE_BASE ORACLE_HOME ORACLE_SID LD_LIBRARY_PATH PATH
```

Save the .bash\_profile and execute following commands for load new enviroment:

```
cd /home/oracle
. .bash_profile
```

## Download & Install

### 1. Download and install required .rpm packages

Some additional packages are required for succesful instalation of Oracle software. To check wheter required packages are installed on your operating system use following command: **Note: Since RHEL 5 (OEL 5, Centos 5) pdksh package was renamed to ksh**

For **32 bit (x86)** Linux version:

```
rpm -q binutils gcc glibc glibc-headers glibc-kernheaders glibc-devel compat-libstdc++ cpp compat-gcc make compat-db
compat-gcc-c++ compat-libstdc++ compat-libstdc++-devel setarch sysstat pdksh libaio libaio-devel --qf '%{name}.%
{arch}\n'|sort
```

For **64 bit (x86\_64)** Linux version:

```
rpm -q binutils compat-db compat-libstdc++-33 glibc glibc-devel glibc-headers gcc gcc-c++ libstdc++ cpp make libaio ksh
elfutils-libelf sysstat libaio libaio-devel setarch --qf '%{name}.%{arch}\n'|sort
```

Required packages for **32bit (x86)** architecture:

```
binutils.i386
compat-gcc-7.3-2.96.128.i386
compat-gcc-c++-7.3-2.96.128.i386
compat-libstdc++-7.3-2.96.128.i386
compat-libstdc++-devel-7.3-2.96.128.i386
cpp.i386
gcc.i386
gcc-c++.i386
```

```
glibc.i386
glibc-common.i386
glibc-devel.i386
glibc-headers.i386
glibc-kernheaders.i386
libstdc++.i386
libstdc++-devel.i386
libaio
libai-devel.i386
pdksh.i386
setarch.i386
sysstat.i386
```

Required packages for 64bit (x86\_64) architecture:

```
binutils.x86_64
compat-db.x86_64
compat-libstdc++-33.i386
compat-libstdc++-33.x86_64
cpp.x86_64
elfutils-libelf.i386
elfutils-libelf.x86_64
gcc-c++.x86_64
gcc.x86_64
glibc-devel.i386
glibc-devel.x86_64
glibc-headers.x86_64
glibc.i686
glibc.x86_64
ksh.x86_64
libaio-devel.i386
libaio-devel.x86_64
libaio.i386
libaio.i386
libaio.x86_64
libaio.x86_64
libstdc++.i386
libstdc++.x86_64
make.x86_64
setarch.x86_64
sysstat.x86_64
```

If some package is not installed then install it from installation media or download it from following locations:

[RedHat Enterprise Linux 3](#) - source packages only

[RedHat Enterprise Linux 4](#) - source packages only

[White Box Linux 3](#)

[White Box Linux 4](#)

[Centos Linux 3](#)

[Centos Linux 4](#)

[Centos Linux 5](#)

This is example how to build RPM package from source package (libaio-0.3.96-3.src.rpm). Note gcc, make and rpm-build (and dependent) packages must be already installed on your system.

```
# rpm -ivh libaio-0.3.96-3.src.rpm
# cd /usr/src/redhat/SPECS/
# rpmbuild -bb --target i386 libaio.spec
# cd ../RPMS/i386/
```

Install the required packages using the rpm command:

```
# rpm -ivh <package_name>.rpm
```

2. Download the Oracle 10g release 2 (10.2.0.1.0) software from [Oracle website](#).

Extract the files using following command: For 32bit installation archive

```
unzip 10201_database_linux32.zip
```

For 64bit installation archive

```
gunzip 10201_database_linux_x86_64.cpio.gz
cpio -idmv <10201_database_linux_x86_64.cpio
```

For RHEL 5, Centos 5: Modify database/install/oraparam.ini file and add "redhat-5" to "Certified Versions" section.

Example:

```
[Certified Versions]
Linux=redhat-3,SuSE-9,redhat-4,UnitedLinux-1.0,asianux-1,asianux-2,redhat-5
```

3. Start the Oracle software installation process.

Now the system is prepared for Oracle software installation. To start the installation process execute the following commands:

```
cd db/Disk1/  
./runInstaller
```

Note: You may get "Warning" status during some pre-requisites checks. This will happen on RH EL 3 where Update 3 or 4 were not installed. You can continue in installation when you simply change the status as "User verified".

## Post-Instalation Tasks

### 1. (Optional) Auto Startup and Shutdown of Database and Listener

Login as root and modify /etc/oratab file and change last character to Y for appropriate database.

```
ORCL:/opt/oracle/102:Y
```

As root user create new file "oracle" (init script for startup and shutdown the database) in /etc/init.d/ directory with following content:

```
#!/bin/bash  
#  
# oracle Init file for starting and stopping  
# Oracle Database. Script is valid for 10g and 11g versions.  
#  
# chkconfig: 35 80 30  
# description: Oracle Database startup script  
  
# Source function library.  
  
. /etc/rc.d/init.d/functions  
  
ORACLE_OWNER="oracle"  
ORACLE_HOME="/opt/oracle/102"  
  
case "$1" in  
start)  
echo -n $"Starting Oracle DB:"  
su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/dbstart $ORACLE_HOME"  
echo "OK"
```

```
;;
stop)
echo -n $"Stopping Oracle DB:"
su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/dbshut $ORACLE_HOME"
echo "OK"
;;
*)
echo $"Usage: $0 {start|stop}"
esac
```

Execute (as root) following commands (First script change the permissions, second script is configuring execution for specific runlevels):

```
chmod 750 /etc/init.d/oracle
chkconfig --add oracle --level 0356
```

## 2. (Optional) Auto Startup and Shutdown of Enterprise Manager Database Control

As root user create new file "oraemctl" (init script for startup and shutdown EM DB Console) in /etc/init.d/ directory with following content:

```
#!/bin/bash
#
# oraemctl Starting and stopping Oracle Enterprise Manager Database Control.
# Script is valid for 10g and 11g versions.
#
# chkconfig: 35 80 30
# description: Enterprise Manager DB Control startup script

# Source function library.

. /etc/rc.d/init.d/functions

ORACLE_OWNER="oracle"
ORACLE_HOME="/opt/oracle/102"

case "$1" in
start)
echo -n $"Starting Oracle EM DB Console:"
su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/emctl start dbconsole"
echo "OK"
;;
stop)
echo -n $"Stopping Oracle EM DB Console:"
su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/emctl stop dbconsole"
```

```
echo "OK"
;;
*)
echo $"Usage: $0 {start|stop}"
esac
```

Execute (as root) following commands (First script change the permissions, second script is configuring execution for specific runlevels):

```
chmod 750 /etc/init.d/oraemctl
chkconfig --add oraemctl --level 0356
```

3. (Optional) You may consider to use rlwrap for comfortable work with sqlplus and rman utility.

RPM package for RedHat compatible (x86) distribution you can download [here](#).

RPM package for RedHat compatible (x86\_64) distribution you can download [here](#).

```
su -
# rpm -ivh rlwrap*.rpm
# exit
echo "alias sqlplus='rlwrap sqlplus'" >> /home/oracle/.bash_profile
echo "alias adrci='rlwrap rman'" >> /home/oracle/.bash_profile
. /home/oracle/.bash_profile
```

## Common Installation Errors

**DISPLAY not set. Please set the DISPLAY and try again.**

Solution: Execute "export DISPLAY=:0.0" when you perform installation on local machine or "export DISPLAY=:0.0 when you perform installation on remote machine connected over SSH". Don't forget to execute "xhost +" command on client machine.

**Exception java.lang.UnsatisfiedLinkError: /tmp/OraInstall2005-07-07\_09-40-45AM/jre/1.4.2/lib/i386/libawt.so: libXp.so.6: cannot open shared object file: No such file or directory occurred.. java.lang.UnsatisfiedLinkError: /tmp/OraInstall2005-07-07\_09-40-45AM/jre/1.4.2/lib/i386/libawt.so: libXp.so.6: cannot open shared object file: No such file or directory**

Solution: RH 3, WB 3, Centos 3 - Install the XFree86-libs-4.3.0-81.EL.i386.rpm and dependent packages.

RH 4, WB 4, Centos 4 - Install the xorg-x11-deprecated-libs-6.8.2-1.EL.13.6.i386.rpm package.

RH 5, OEL 5, Centos 5 - Install the libXp-1.0.0-8.1.el5.i386.rpm package.

**error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory**

Solution: Install libaio and libaio-devel packages. If packages already installed and error still occurs try execute "ldconfig" as root.



**Check complete. The overall result of this check is: Failed <<<<**

Solution: Install missing package or set check system parameters (See reason of failure).

Comments, suggestions, questions, errors (also grammatical :) )? Feel free to [contact me](#).