RedHat 6.0

Step-by-step Install notes for Oracle 8.0.5

IMPORTANT! NOTES ABOUT REDHAT 6.0

RedHat 6.0 is using a new version glibc, **and this new version doesn't agree with Oracle.**
Some of the symptoms of installing on RedHat 6.0 are:

- Installation fails if you try to create database objects during install.
- Oracle binaries, such as SVRMGRL and SQLPLUS, core dump when you run them.

Oracle has posted the glibcpatch.tgz file, which will patch and relink your binaries so that they work on RedHat 6.0. This "fix" also involves using four compatibility RPMs from RedHat (they are provided on the RH 6.0 CD).

This document explains how to install Oracle in the RedHat 6.0 environment.

The machine I used to install Oracle

I installed it on a more powerful system for our business, but the lowest machine I installed was a Pentium 166, 32 megs RAM, 1.6 gig HD. The install on the 166 takes a while, more than three hours using this method.

Sections in this doc

There are six:

RedHat initial install - Covers what is needed at minimum to install with RedHat 6.0, plus memory parameters recommended by Oracle and why you shouldn't use them.
Oracle pre-install - Setting up your users, groups, mount points and environment variables.
Oracle install - Where you actually install the software.
Oracle Documentation install - Needs to be a separate process due to a bug.
Patching the binaries - Needed to keep them from core-dumping.
Oracle post-install - Clean-up tasks and testing.

I make no guarantees that the installation will work if you decide to just use part of these instructions vs. the whole thing.

Editorial comment about this document

There's always more than one way to do this sort of thing. This is what worked for me. I'm not necessarily a Linux guru, so some of you may see something and question what I was thinking. But hey, it works, so grab what you can from this document, and have a good laugh when you think I'm nuts. When you're done laughing, drop a note to me, tbissett@fortwayne.com, and (kindly) let me know where I'm nuts.

Why use Redhat?

Oracle used RedHat when they ported the server to Linux and from all I have heard, will continue to use RedHat when developing. It will be the least hassle to install.

Redhat Initial Install

1) Install any way you would like, keeping the following steps in mind as you're doing it.

2) Make swap partitions (max. 128 mb ea.) equivalent to **three times RAM installed.**

3) **Partition the drive** to your preferences. I ignored the OFA-compliance and just made one mount point (/u01). You can also fake OFA by making four directories, but you won't gain any performance from this. You'll gain some performance by creating four distinct partitions (/u01, /u02, /u03, /u04) and you'll gain the most performance by having each of these mount points on four distinct drives, as Oracle recommends.

4) **AT MINIMUM,** the **C Development** package must be installed. You can add others to suit your needs.

5) After the install is completed, make a /cdrom directory (or can use the one defined in etc/fstab, which is /mnt/cdrom). This is where we'll mount the CD.

- **mkdir /cdrom**
- **chmod 777 /cdrom**

6) If you have the RedHat 6.0 CD, Mount it and change to the /RedHat/RPMS directory. Then, go to step 8. If you downloaded the RPMs go to step 7:

- **mount -t iso9660 /dev/cdrom /cdrom**
- **cd /cdrom/RedHat/RPMS**

7) Change into the directory where you placed your compatibility RPMs:

8) The first thing is to install a couple of RPMs:

- **rpm -ivh kernel-source-2.2.5-15.i386.rpm.** This is the kernel source, so that we can recompile the kernel for new memory parameters. More on that later.
- **rpm -ivh tcl-8.0.3-20.i386.rpm** This is **ONLY** if you want to install the Intelligent Agent, which offers minimal extra capabilities. The agent itself is very buggy right now, and really doesn't add a lot to Oracle, so installing it may be more trouble than it's worth.

9) Change out of the cdrom directory and unmount the RedHat CD:

- **cd /**
- **umount /cdrom**

Kernel memory params

The front of the Oracle installation manual lists some memory parameters which, according to the Oracle install manual, need to be changed before installing the software. **On RedHat 6.0, Oracle will install**

and run without reconfiguring the kernel. These new values are just recommendations. **Should you decided to change these parameters, the only parameter you really need to worry about in RH 6.0 is the SHMMAX.** However, if you set the SHMMAX to what Oracle says, you are telling the system to allow a shared memory segment to grow as large as 4 Gigs. **Many users have reported problems setting the value this high.** Furthermore, if you only have 32 megs RAM, what's the point? The Redhat 6.0 default setting for SHMMAX is 32 megs, by the way, so if you have more RAM, you may want to change these settings. Here's how I did it:

WARNING!

Changing the kernel parameters requires recompiling the Linux kernel. If you have never done this, I highly recommend you get a good book that walks you through the process and all the 100+ different options when you run the configure script. If you aren't sure of some of them, choosing the wrong one may prevent you from booting your system!

Changing memory params

1) Change to the directory listed below and edit the shmparam.h file.

- **cd /usr/src/linux/include/asm**
- **vi shmparam.h**

2) A little further down is a line that says "#define SHMMAX 0x2000000." The 0x2000000 is hex for 32 megs. Change it to a value that suits you. I've bumped it up to 0x8000000, which is 134217728, or slightly above the 128 megs RAM I have installed.
Other values:
0x4000000 = 67108854
0x6000000 = 100663296

3) Save the file.

4) You're now ready to proceed with the process of recompiling the kernel. Recompile it, then continue below.

Oracle Pre-Install

Booting RedHat Linux in text mode

By default, RedHat 6.0 installs Run Level 5 as the default. This runlevel starts X Windows on boot. To install Oracle, we're going to (temporarily) switch the boot runlevel to 3, which is text mode. If you already boot into text mode, skip these steps. If not, here's how to change it:

1) Login as **root.**

2) Open a terminal window.

3) Change to the /etc directory

- **cd /etc**

4) Edit the **inittab** file

5) Change the line that reads:
id:5:initdefault:
to read:
id:3:initdefault:

6) **Reboot** your system.

7) **You should now have a plain-old boring text command line to login.** Don't worry. We'll change it back when we're finished. **Of course, if you're running a true database server, you probably wouldn't want to put X Windows on in the first place. After all, why waste processing power just for a GUI?**

Obtaining and installing patches

Before you can install Oracle on a RedHat 6.0 system, you must add five compatibility packages to make Oracle work. You will also need to download a glibc patch from Oracle's TechNet.

1) If you have the Redhat 6.0 CD the packages listed below are in your RedHat/RPMS directory. If you do not have the RPMs on your installation, you'll need to download them from ftp.redhat.com, or one of it's mirrors.

Mount your CD as root or download the appropriate packages. Install them by using the following commands in the order listed below:

- **mount -t iso9660 /dev/cdrom /cdrom** (ignore this if you downloaded them)
- **cd /cdrom/RedHat/RPMS** (substitute with your directory if not RH6.0)
- **rpm -ivh compat-binutils-5.2-2.9.1.0.23.1.i386.rpm**
- **rpm -ivh compat-glibc-5.2-2.0.7.1.i386.rpm**
- **rpm -ivh compat-egcs-5.2-1.0.3a.1.i386.rpm**
- **rpm -ivh compat-egcs-c++-5.2-1.0.3a.1.i386.rpm**
- **rpm -ivh compat-libs-5.2-1.i386.rpm**
- **cd /**
- **umount /cdrom**

2) Create your groups. Afterward, take a look at /etc/group to see what IDs are assigned to these groups.
    - **groupadd dba (On RedHat 6, the group ID will probably be 500).**
    - **groupadd oper (this is an optional group. It will likely by ID 501).**

3) Create the oracle user
    - **useradd oracle -g 500 (sub number with dba group number from above)**
    - **passwd oracle (to change password)**

4) If you didn't make mount points with disk partitioning, make a directory for oracle, such as /u01.
    - **mkdir /u01.** (You can also fake the OFA-compliancy by mkdir /u01, /u02, /u03 and /u04.

5) Change owner and group permissions on these mount points.

- **chown -R oracle.dba /u01** (run same command for /u02, /u03 and /u04 if you have them)

6) Mount the Oracle Server CD:

- **mount -t iso9660 /dev/cdrom /cdrom** (sub w/ your CD device locator).

7) Change to oracle CD and orainst directory, define a quick variable for ORACLE_OWNER, which should be oracle, then run the oratab script.

- **cd /cdrom/orainst**
- **ORACLE_OWNER=oracle; export ORACLE_OWNER**
- **sh oratab.sh (accept defaults for prompts).**

8) Log out.

9) Log in as oracle.

10) **At this point, you will need to download glibcpatch.tgz,** which you can get via FTP. Make a directory in your oracle user home directory for this patch file, the go and download it:

- **mkdir ~/orapatch**
- **cd ~/orapatch**
- **ftp ftp.oracle.com (log in as anonymous)**
- **cd /pub/www/otn/linux**
- **get glibcpatch.tgz**
- **quit**

Setting up the oracle user's environment

1) Verify umask is 022 by typing umask [enter]. If it is, great. If not, you'll need to put it in .bash_profile, which we edit in the next step.

2) Use you favorite editor to edit the **.bash_profile** file in your oracle account's home directory. Put the following lines in the file:

- **ORACLE_HOME=/u01/app/oracle/product/8.0.5; export ORACLE_HOME**
- **LD_LIBRARY_PATH=/u01/app/oracle/product/8.0.5/lib; export LD_LIBRARY_PATH**
- **ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE**
- **ORACLE_SID=ORCL; export ORACLE_SID**
- **ORACLE_TERM=386; export ORACLE_TERM (this is for Intel PCs, use appropriate value if on different platform)**
- **PATH=$PATH:/u01/app/oracle/product/8.0.5/bin; export PATH**
- **(optional) umask 022 (only if umask does not default to 022)**

3) Log out, then back in as oracle. Type env to see if all the above variables now appear in your environment.

4) Also make sure all the following paths are in your path statement, in addition to the Oracle path you

specified above: /bin, /usr/bin, /usr/local/bin.

Oracle Install

1) Change to the Oracle install directory

- **cd /cdrom/orainst**

2) Start the installer

- **./orainst /c**

3) Choose Custom Install (this let's you create all your passwords ahead of time, which is convenient).

4) Choose OK to get through the README's.

5) Choose **Install, Upgrade or De-install software.**

6) Choose **Install new product- DO NOT Create DB objects. It is important that you don't create the database now!**

7) Verify your **ORACLE_HOME** and **ORACLE_BASE** are correct.

8) Accept the defaults for log files.

9) Choose Install from CD-ROM.

10) Select your appropriate language.

11) The Installer then tells you where the root.sh script will end up. Hit OK.

12) When you get to the Install options menu, select what you want EXCEPT:

- **Do not pick Intelligent agent** unless you installed the TCL rpm package above.
- **Do not pick JDBC drivers** unless you defined a path to your classesxxx.zip file.
- **Do not pick Oracle 8.0.5 documentation here. There is a bug in it which will crash your install process.** Luckily, there is a fix, which we will be apply after the initial install. Just trust me on this one.

13) Hit the install button.

14) Hit Ok at the **ULIMIT** message. You don't need to worry about this.

15) Select **dba** as your dba group.

16) If you specified a different group for operators (oper), enter it at the OSOPER prompt. Otherwise, accept the default.

17) If you did enter a different group, the installer will tell you programs will need to be relinked. Just hit OK.

18) **At this point, the installer actually begins copying software!**
The install process will take a long time on slower machines. Taking multiple hours is not out of the question. Just be patient.

19) If you get a message that says, "**The requested action has been performed for selected products,**" You're in luck. Hit OK. You should return to the main install screen.

20) Select **Exit** and then yes at the confirmation.

21) **If you do not get the above message, even though the installer acts like everything was installed correctly, I'll bet it wasn't. Just a note from experience.**

22) You should see a line after exiting that says **Result: Success.** Congratulations. This means your software installed successfully. Before you can use it, though, **you must do a few more steps,** and you might want to optionally install the Oracle Documentation.

Oracle Documentation install

Why we have to do this separately: The Oracle install process tries to install one of the doc files in a **non-existent directory** (must be a typo in the install script). This error hangs the above install process if you try to install the docs along with everything else. Thanks to the kind help of Ben Drasin, here is a way to get the docs installed in their proper place:

1) First we make the Oracle doc directory ahead of time.

- **mkdir /u01/app/oracle/doc**

2) Then we make a link between two directories. This link will help to correct the pathname for that one bad file.

- **ln -s /u01/app/oracle/doc /u01/app/oracle/product/8.0.5**

3) Now, we're going to run the installer from the CD again.

- **./orainst /c**

4) Select **Custom Install.**

5) Hit **OK** at the next **two** prompts.

6) Select **Install, Upgrade, or De-Install software.**

7) Select **Add/Upgrade software.**

8) Accept the default for **ORACLE_HOME.**

9) Accept the log defaults.

10) Select **Install from CD-ROM.**

11) Select your language.

12) The installer will now tell you that root.sh already exists. Since we have not run it yet, we want to **append to our existing root.sh,** so select **Append.**

13) Hit **Ok** at the "Post-Installation..." message.

14) You are now at the familiar Install screen. Select **Oracle Server Release 8.0.5 Documentation.**

15) Hit **Install.**

16) Accept the pathname for **ORACLE_DOC.**

17) Select which format you want for the documentation.

18) You should then get a window saying the selected action has been performed. Hit **OK.**

19) Hit Exit and confirm, and you'll (hopefully) get the Result: success line.

Now we can proceed with all the Post-install stuff.

Patching your binaries

Unless you apply this patch, your binaries will core-dump when you run them. Do NOT apply this patch until you have installed the software above.

1) Change to the directory where you downloaded your patch.

- **cd ~/orapatch**

2) Extract the patch.

- **tar -xvzf glibcpatch.tgz**

3) Run the script.

- **sh glibcpatch.sh**

4) As with everything else, the wait could be an hour or more on slower machines. When the script finishes, you will get a message saying "Applied glibc patch for Oracle 8.0.5.x successfully."

Create the DB objects

This is where we create the original database.

1) Change to the orainst directory on the CD-ROM.

- **cd /cdrom/orainst**

2) Run the installer again.

- **./orainst /c**

3) Select **Custom Install.**

4) Hit **OK** to get through the two read me's.

5) Select **Create/Upgrade Database objects.**

6) At the next menu, select **Create Database Objects.**

7) Verify **ORACLE_HOME** and **ORACLE_BASE** are correct, then hit OK.

8) Accept the defaults for the logfiles by hitting OK.

9) Verify your **ORACLE_SID** is correct.

10) You are now at the familiar Oracle install screen. **You must select "Oracle 8 Standard RDBMS 8.0.5.0.0" for the database objects to be created.**

11) Hit the **Install** button.

12) Select **Create Product DB Objects.**

13) Choose **Filesystem-based Database** regardless of whether you're installing in a single mount point or multiple mount points.

14) If you have a single mount point, when the installer asks you to distribute control files over multiple mount points, hit no, and specify your only mount point in the next window (exam: /u01). If you have multiple mount points, hit yes and specify them in the next window.

15) Select the appropriate **character set.**

16) Select the appropriate **national character set.**

17) Enter in you password for the **SYSTEM** account. Confirm it.

18) Enter in your password for the **SYS** account. Confirm it.

19) If you want an internal password for **dba** and **operator,** you can choose yes at this prompt. Otherwise, hit no.

20) Enter in a password for the **TNS listener.**

21) Hit **No** if it asks to configure the MTS Listener.

22) Accept the defaults for control files.

23) Hit OK through the next two defaults screens.

24) Finally, select Yes to accept those defaults.

25) At this point the database creation process begins. On slower machines, this process can also take more than an hour.

26) If you get a message that says, "**The requested action has been performed for selected products,**" You're in luck. Hit OK. You should return to the main install screen.

27) Select **Exit** and then yes at the confirmation.

28) Proceed to the Post-Install tasks.

Oracle post-install

1) Log out, then back in as root.

2) Edit root's .bash_profile. Add the following lines to it:

- **ORACLE_HOME=/u01/app/oracle/product/8.0.5; export ORACLE_HOME**
- **LD_LIBRARY_PATH=/u01/app/oracle/product/8.0.5/lib; export LD_LIBRARY_PATH**
- **ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE**
- **ORACLE_SID=ORCL; export ORACLE_SID**
- **PATH=$PATH:/u01/app/oracle/product/8.0.5/bin; export PATH**

3) Log out, then back in as root.

4) Run the root.sh script

- **cd $ORACLE_HOME/orainst**
- **sh root.sh**

5) Verify **ORACLE_OWNER, ORACLE_HOME, and ORACLE_SID** are correct, then **hit Y.**

6) When it asks for the full path name to your local bin directory, enter /usr/local/bin.

7) The script then tells you that ORACLE_HOME does not match the home directory for oracle. You need not worry about this. Type a Y and continue. The script will complete.

8) Log out, then back in as oracle

9) Stop the server:

- **svrmgrl**
- **connect internal**
- **shutdown**
- **quit**

Modifying the init(ORACLE_SID)file

After you shut down the Oracle server, it copies a file called init(ORACLE_SID)_0.ora to init(ORACLE_SID).ora. Example: If you set ORACLE_SID to ORCL, the file names would be initORCL_0.ora and initORCL.ora. **There is a link to the the init file in $ORACLE_HOME/dbs.**

The file that has the "**_0.ora"** was a temporary file created by oracle for the installation. The other file is the one you want to look at. It is what Oracle uses to define its System Global Area. The file itself is pretty self-explanatory. If you have a lot of RAM, consider commenting out the entries marked "small" and uncommenting the lines marked "medium" or "large." The default is "small."

Changing the TNS Listener permissions

Installation of the TNS listener has two problems. First, the TNS listener is installed with improper permissions. Second, the listener.ora file needs to be checked to see if the ORACLE_SID was properly inserted (most likely, it wasn't). Run the following commands to become root and get tings straightened out:

- **su root**
- **[enter password]**
- **cd $ORACLE_HOME**

- **chown oracle.dba bin/tnslsnr**
- **chmod 750 bin/tnslsnr**
- **chown oracle.dba network/log**
- **chmod 775 network/log**
- **touch network/log/listener.log**
- **chown root.dba network/log/listener.log**
- **chmod 664 network/log/listener.log**
- **exit (to leave root)**
- **cd $ORACLE_HOME/network/admin**
- **edit listener.ora**
- **You need to find all instances where it says <ORACLE_SID> and replace these with your actual ORACLE_SID (without the <> symbols). If you don't, the listener will not start correctly.**
- **Save the file.**

Starting the Oracle server and testing

Important: If for some reason the passwords you typed in earlier for SYSTEM and SYS do not work during any of these steps, try the defaults that Oracle supplies. There seems to be a bug in some installs.
Default password for SYSTEM: manager
Default password for SYS: change_on_install

Now comes the fun part.
1) If you aren't logged in as oracle, go ahead and log in.

2) If you want to access the Oracle server with the GUI management tools from another PC, you will need to start the TNS listener. The default config works for most TCP/IP installations, but you may need to change it. It worked for us.

- **lsnrctl start**

3) Start Oracle server back up by connecting as internal.

- **svrmgrl**
- **connect internal**
- **startup**
- **quit**

4) Now, we run a script which creates user profiles using sqlplus:

- **cd $ORACLE_HOME/sqlplus/admin**
- **svrmgrl**
- **connect system/<password> (This is the password you created for system account (default is "manager")).**
- **@pupbld.sql**

- **quit**

5) If you want to set up the sample tables for user scott/tiger, you can do it this way

- **cd $ORACLE_HOME/rdbms/admin**
- **sqlplus (Connect as sys)**
- **@utlsampl.sql**

6) You should now be able to create users tables etc. Your RDBMS should be up and running!

Change your system back to run level 5

If you want your system to automatically come back up into X Windows again, simply edit your /etc/inittab file and change the line that says id:3:initdefault: back to id:5:initdefault: