

Installing and Configuring Oracle on the Linux Platform

Roger Schrag
Database Specialists, Inc.
<http://www.dbspecialists.com>

Introduction

This document will walk you through the steps of installing Oracle in a Linux environment. Linux offers the robustness and familiarity of Unix, but is free (or very cheap) and can run on very inexpensive Intel hardware. The power and affordability of Linux make it an ideal platform for your Oracle database server.

This concludes the marketing hype portion of this document. Everything that follows is hands on, roll-up-your-sleeves-and-get-busy material for Oracle users who want to get an Oracle database up and running quickly, but want the database to be scalable and to perform well. As of this writing, Oracle 8i is not yet commercially available on Linux, so we will cover the installation of Oracle 8.0.5.0.0 Standard Edition.

There are four phases to getting Oracle up and running on your Linux server:

1. Prepare the server
2. Install the Oracle software
3. Create an Oracle database
4. Complete the server configuration

We will walk through these phases one at a time, detailing all the steps involved. The end result will be a very usable database that can be scaled up quite large. Of course, every implementation is unique, and you will need to evaluate each step carefully against your particular requirements. But this document will get you off to a very solid start.

One last note before we get started: Linux is extremely similar to flavors of Unix such as Solaris. About 95% of the material here is directly applicable to other Unix platforms. The main areas of divergence are the setting of the operating system kernel parameters and the locations of the oratab file and local bin directory.

Prepare the Server

These steps configure your machine so that it will be ready to accept the Oracle software and database. In this section, we will make sure the operating system meets Oracle's minimum requirements, create a special user and group to "own" the software, and create some directories that will be used by the Oracle software and database. All of the steps in this section are run as the root user.

1. Make sure that your version of Linux is sufficient. Oracle works with many different distributions of Linux. The key requirements are that the kernel be 2.0.33 or higher, and GNU C libraries be 2.0.6 or higher. Many users have reported that the GNU C libraries must in fact be 2.0.6 or 2.0.7—that the "or higher" is not true. You will greatly simplify your life if you work with the Red Hat 5.2 distribution. This package is reasonably priced, readily available, and known to work very well with Oracle 8.0.5.0.0 Standard Edition. Note that the Red Hat 6.0 distribution includes GNU C libraries 2.1, and many users have reported difficulties installing Oracle on Linux systems that use GNU C libraries 2.1.
2. Make sure that your hardware is sufficient. You'll need at least 32 Mb RAM, a CD ROM drive, and about 500 Mb of disk space. This will let you create a small database for prototyping; a real implementation will likely require more RAM and more disk space.
3. Make sure that the Linux kernel has parameters set sufficiently high for Oracle. The Oracle architecture makes heavy use of shared memory segments for sharing data between multiple processes and semaphores for handling locking.

Many operating systems, such as Solaris, do not by default offer sufficient shared memory or semaphores for maintaining an Oracle database. Happily, Red Hat Linux 5.2, by default, builds a kernel that will support most Oracle implementations.

Kernel Parameter	Red Hat 5.2 Default	Purpose
SHMMAX	0x2000000 (32 Mb)	Maximum size of a single shared memory segment
SHMMIN	1	Minimum size of a single shared memory segment
SHMMNI	128	Maximum number of shared memory segments in entire system
SHMSEG	128	Maximum number of shared memory segments one process can attach
SEMMNS	4096	Maximum number of semaphores in entire system
SEMMNI	128	Maximum number of semaphore sets in entire system
SEMMSL	32	Maximum number of semaphores per set

The first four kernel parameters configure shared memory segments. The Red Hat 5.2 defaults allow your database SGAs to get as big as you could possibly want, when you consider that Linux is limited to 2 Gb of physical memory. If a database has an SGA bigger than 32 Mb, it will be spread across multiple shared memory segments. This is not a concern.

The last three kernel parameters configure semaphores. Each Oracle instance requires one semaphore for each process. Oracle does not seem to care how many semaphores are in each set, but it appears that one instance cannot handle more than 30 sets. The Red Hat 5.2 defaults allow you to have four instances on one server, with each instance having about 960 processes. This should be sufficient for most implementations.

Recompiling a Linux kernel is somewhat complicated and definitely requires that you know what you are doing. We won't cover it here, but you probably won't need to recompile your kernel to install Oracle, anyway.

4. Create a Linux group that will be used by the Oracle software owner and database administrators. You can call it what you like, but the standard is "dba". If you will be installing Oracle on multiple Linux servers on your network, you might want to keep the groupid the same on all servers. I created my dba group with the command:

```
groupadd -g 300 dba
```

5. Create a Linux user that will be the Oracle software owner. You can call it what you like, but the standard is "oracle". If you will be installing Oracle on multiple Linux servers on your network, you might want to keep the userid the same on all servers. Note that this user's home directory will not be the ORACLE_HOME or where the actual Oracle software is installed; this user's home directory should be in the same place as other users' home directories. This user's initial group should be the dba group created in the previous step. I created my oracle user with the commands:

```
useradd -c 'Oracle software owner' -d /home/oracle -g dba -m -n \  
-s /bin/bash -u 300 oracle  
passwd oracle
```

6. Create mount points for the Oracle software and the Oracle database. Each mount point should correspond to a separate physical disk device. You'll need at least one mount point. Typically you use one mount point for the Oracle software and as many as you can afford for each database. (More physical devices allow better performance.) A nice convention is to call the mount points /u01, /u02, and so on. Because mount points are typically owned by root and the Oracle installer will run as the oracle user and not root, you should create some subdirectories now to avoid permissions problems later. Create an app subdirectory below the software mount point, and oradata subdirectories below the mount points to be used for databases. (You can put software and a database on the same mount point if you wish.) Make these subdirectories owned by the oracle user and dba group, and give them 755 permissions.
7. Choose a directory that the Oracle software will refer to as the "local bin" directory. A common choice is /usr/local/bin, and your installation will go more smoothly if you stick with this choice. Make sure this directory is on users' path by default.

8. If you downloaded a trial version of Oracle off of the internet, then untar the distribution. If you have the software on CD ROM, then mount the CD ROM now. I use automount to mount my CDs, but alternatively you can use a command like:

```
mount -t iso9660 /dev/cdrom /cdrom
```

9. Create the /etc/oratab file. After installation, this will be a plain text file that briefly describes the Oracle software installations and databases on the server. The file must exist before installation can begin. The easiest way to create the file is simply to create an empty file called /etc/oratab and change the ownership to oracle, the group to dba, and the permissions to 664. Alternatively, you can set the ORACLE_OWNER environment variable and execute a shell script on the CD ROM:

```
ORACLE_OWNER=oracle
export ORACLE_OWNER
cd /cdrom/orainst
./oratab.sh
```

Install the Oracle Software

These steps install the Oracle database software onto your server so that you will then be able to create databases and use server-side tools like SQL*Plus and Server Manager. In this section, we will prepare the oracle user's environment, run the Oracle Installer, and tidy up a few minor messes that the installer leaves behind. We will create a database later by invoking the Oracle Installer a second time. This allows us to choose our own database block size instead of being forced to use the default. All of the steps in this section, except where noted, are run as the oracle user.

1. Edit the oracle user's login file so that the environment will be configured automatically on login. If you are using Bash, then edit the .bashrc file. If you are using Bourne or Korn shell, then edit .profile. For now, we will hardcode certain things. But after the software and database are installed, we will come back and eliminate all hardcodings. Here is what I added to my .bashrc:

```
umask 022
# Substitute your Oracle software mount point in the line below.
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/8.0.5
# Substitute the name of your Oracle database below.
export ORACLE_SID=MYDB
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
# Substitute terminal type in line below. vt100 or vt220 is
# recommended.
export ORACLE_TERM=vt100
# Following two lines are not required if you'll be using the
# default character set, 7-bit ASCII. To use another character
# set, see Appendix C of the Installation Guide and change
# NLS_LANG accordingly.
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export NLS_LANG=american_america.US7ASCII
# Fill in the following line as you wish, but make sure that
# $ORACLE_HOME/bin, /bin, /usr/bin, and "local bin" are all
# in the PATH.
export PATH=...
# If your /var/tmp doesn't have at least 20 Mb free for the
# Oracle Installer to use (or is not writable by oracle), then
# complete the following line.
export TMPDIR=...
# Ensure that SRCHOME and TWO_TASK are not set.
# If you will be using Java, include the following two lines:
export CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/jdbc/lib
```

2. Log out and log back in as the oracle user so that the environment is set correctly.
3. Use ftp to transfer a small file to a remote host to prove to yourself that TCP/IP networking is installed and working properly on your server.
4. Ensure that the mount point you designated for the Oracle software has sufficient free space. Allow 400 Mb as a bare minimum.
5. Double check that you are logged in as oracle and not root. Then start the Oracle Installer. I recommend that with this release of Oracle, you simplify your life and use the character mode version of the installer. To do this, make sure your DISPLAY environment variable is not set, and that ORACLE_TERM is set accordingly. Start the installer with these commands:

```
cd /cdrom/orainst
./orainst
```

We'll walk through the installer prompts one at a time:

- a. Choose Custom Install.
 - b. Read two not very interesting readme files.
 - c. Choose Install, Upgrade, or De-Install Software.
 - d. Choose Install New Product – Do Not Create DB Objects.
 - e. Confirm settings for ORACLE_BASE and ORACLE_HOME. These should be correct, because you set your environment variables beforehand.
 - f. Confirm log file locations. The Oracle Installer will write four log files in the `$ORACLE_HOME/orainst` directory.
 - g. Choose Install from CD-ROM.
 - h. Choose the language you want your Oracle installation to support. This is the language that banners, prompts, and error messages will be displayed in. The languages available for your data stored in the database will be dictated by the character set you choose when creating the database, and not by how you answer this installer prompt. I chose American English.
 - i. Confirm the root.sh location. This file will contain post-installation steps to be performed by the root user. If you run the installer multiple times, you'll be prompted to move the previous root.sh file to another location.
 - j. On the Software Asset Manager screen, select the products you wish to install and choose the Install button. Note that selecting products that have sub-products (like "Net8 Protocol Adapters") doesn't do anything—you need to select the sub-products themselves. Here's what I installed:
 - Client Software
 - Net8
 - TCP/IP Protocol Adapter
 - Unix Installer
 - JDBC Drivers
 - Server Documentation
 - Standard RDBMS
 - PL/SQL
 - Pro*C/C++
 - SQL*Plus
- I recommend installing the Unix Installer. This will make it easier to perform maintenance operations (such as relinking) in the future without having to dig out the CD ROM. Also, note that the Server Documentation selection will only install Linux platform-specific documentation; the generic Oracle product documentation is contained on a separate CD ROM.
- k. Choose the Linux group you created in the previous section as the DBA group. Any Linux user belonging to this group will have full database administration privileges over all databases on this server.
 - l. Choose the Linux group you created in the previous section as the osoper group. Any Linux user belonging to this group will have "operator" privileges over all databases on this server. You could use a separate Linux group for this purpose, but most installations do fine with using the same Linux group for both DBA and osoper.
 - m. Choose the JDBC drivers you wish to install (if you selected JDBC drivers on the Software Asset Manager screen). There is support for JDK 1.0 and 1.1, and there are drivers that require Net8 or drivers that run on a thin client.
 - n. Confirm the directory for your online documentation, if you chose to install it. The default is `$ORACLE_HOME/doc`, and this seems like a good choice.
 - o. Choose whether to install documentation in HTML format, PDF format, or both. Remember, this only applies to the Linux platform-specific documentation.
 - p. At this point, the Oracle Installer installs software onto your server.
 - q. I encountered one glitch during the software installation. You will not encounter this problem unless you chose to install Server Documentation. The installer reported a write error, which may have occurred because the installer tried to copy a file without creating the directory first. I fixed this problem easily by going to another window on the server and creating the directory specified in the error message. Then I chose the Retry option in the Oracle Installer, and things continued normally.
 - r. When the installation is complete you'll receive a message that the requested actions have been performed. Choose OK, and you will be returned to the Software Asset Manager screen.
 - s. Exit the installer.

6. In `$ORACLE_HOME/bin` you will find a shell script called `oraenv`. This script can be called from `.bashrc` or `.profile` to set up a user's environment. Unfortunately, there are several variables that the script does not set—some handy, some very important. Make a backup copy of the script and then edit it, adding the following lines to the very end:

```
# Begin customizations
ORACLE_BASE=`dirname $ORACLE_HOME`
ORACLE_BASE=`dirname $ORACLE_BASE`
DBA=$ORACLE_BASE/admin
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
# Substitute character set you plan to use in following line.
NLS_LANG=american_america.US7ASCII
export ORACLE_BASE DBA ORA_NLS33 NLS_LANG
case "$LD_LIBRARY_PATH" in
  *$OLDHOME/lib*) LD_LIBRARY_PATH=`echo $LD_LIBRARY_PATH | \
                  sed "s;$OLDHOME/lib;$ORACLE_HOME/lib;g"` ;;
  *$ORACLE_HOME/lib*) ;;
  *) LD_LIBRARY_PATH=${LD_LIBRARY_PATH}$ORACLE_HOME/lib: ;;
  "") LD_LIBRARY_PATH=$ORACLE_HOME/lib ;;
  *) LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib ;;
esac
# Following case statement only required if using JDBC.
case "$LD_LIBRARY_PATH" in
  *$OLDHOME/jdbc/lib*) LD_LIBRARY_PATH=`echo $LD_LIBRARY_PATH | \
                      sed "s;$OLDHOME/jdbc/lib;$ORACLE_HOME/jdbc/lib;g"` ;;
  *$ORACLE_HOME/jdbc/lib*) ;;
  *) LD_LIBRARY_PATH=${LD_LIBRARY_PATH}$ORACLE_HOME/jdbc/lib: ;;
  "") LD_LIBRARY_PATH=$ORACLE_HOME/jdbc/lib ;;
  *) LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/jdbc/lib ;;
esac
export LD_LIBRARY_PATH
# End customizations
```

7. I made a further edit to `$ORACLE_HOME/bin/oraenv` that you may or may not need. After editing my `.bashrc` to call `oraenv` each time a new shell is created (we'll do this in a future step) I found that sometimes my server would run out of processes. I have not seen this behavior on other Linux servers, so I might have something peculiar in my configuration. If your server starts giving you `vfork` errors further down in the installation process, then come back and perform this step. Otherwise, you may wish to try and get by without it. The edit is as follows: Find the line in `$ORACLE_HOME/bin/oraenv` that contains a call to `dbhome`. Replace it with the following:

```
# Begin customizations
#ORAHOME=`dbhome "$ORACLE_SID" `
# Following two lines taken from dbhome:
ORATAB=/etc/oratab
ORAHOME=`awk -F: "/^${ORACLE_SID}:/ {print \$2; exit}" \
          $ORATAB 2>/dev/null`
# End customizations
```

8. In `$ORACLE_HOME/bin` you'll find a script called `dbshut`. This is a utility that you can run to shut down databases on the server. Unfortunately, it shuts down databases with normal priority. This means that if any users are logged into a database, the shutdown will hang until they log out. You might want to change this script to shut down databases with immediate priority. To do this, find the two lines that contain just the word "shutdown". Change these to read "shutdown immediate".

9. Look at the script `$ORACLE_HOME/orainst/root.sh`. This is a script that the installer created during software installation. It's a collection of commands that need to be run as the root user. You should review the script carefully and decide whether to run it as is, or hand-pick the commands you wish to run as root. Running the script as is creates more problems than it fixes. Really, all you need to do as root is copy `oraenv`, `dbhome`, and `coraenv` from `$ORACLE_HOME/bin` to the local bin directory. If you do choose to run `root.sh`, please be advised that there is a documented serious security problem. `root.sh` will set the `setuid` bit on several files in `$ORACLE_HOME/bin`, and change some of them to be owned by root. Nothing in `$ORACLE_HOME/bin` should be owned by root, and only the `oracle` and `dbsnmp` executables should have a `setuid` bit turned on. Oracle Corporation has released an emergency patch called `setuid_patch.sh` to address this problem. The problem is also present on all Unix releases of Oracle 8.0 and 8i.
10. Whether or not you run `root.sh`, double-check a few file permissions. The permissions on `$ORACLE_HOME/bin/oracle` should be `6751`. My installation came out as `4755`. Also, `oraenv` in `$ORACLE_HOME/bin` and the local bin directory should have permissions `755`. My installation came out as `777`.

Create an Oracle Database

These steps create an Oracle database on your server. In this section we will create a default Oracle database, tweak the default configuration so that the database is actually usable, create application users and tablespaces in the database, and configure Net8. All of the steps in this section are run as the oracle user.

1. Choose a block size for your Oracle database. The default is 2 Kb, and this is appropriate for very small demo or "sand box" databases. If you anticipate your database growing beyond perhaps 100 Mb in size, you should use a block size of 8 Kb or 16 Kb.
2. The Oracle Installer will not ask you what block size you want, and you cannot change a database's block size after creation. If you want your database to have a block size other than 2 Kb, then you will need to edit the file `$ORACLE_HOME/rdbms/install/rdbms/cnfg.orc` to set the `db_block_size` to 8192 or 16384. Make sure the line is not commented out.
3. Double check that you are logged in as oracle and not root. Then start the Oracle Installer. Again, I recommend that with this release of Oracle you simplify your life and use the character mode version of the installer. To do this, make sure your `DISPLAY` environment variable is not set, and that `ORACLE_TERM` is set accordingly. This time we will run the Oracle Installer installed on your system; we no longer have a need for the software CD ROM. Start the installer with these commands:

```
cd $ORACLE_HOME/orainst
./orainst
```

We'll walk through the installer prompts one at a time:

- a. Choose Custom Install.
- b. Read two not very interesting readme files.
- c. Choose Create/Upgrade Database Objects.
- d. Choose Create Database Objects.
- e. Confirm settings for `ORACLE_BASE` and `ORACLE_HOME`. These should be correct, because you set your environment variables beforehand. Note that the installer will want the parent directory above `ORACLE_BASE` to be writable by the oracle user. This does not make a lot of sense, but the easiest thing to do is humor the installer and open up permissions on that directory (for example, `/u01/app`) temporarily. After the database is created, you can lock down the permissions again.
- f. Confirm log file locations. The Oracle Installer will write four log files in the `$ORACLE_HOME/orainst` directory.
- g. Confirm the setting for `ORACLE_SID`. This should be correct, because you set your environment variables beforehand.

- h. On the Software Asset Manager screen, you'll see a list of all products that have been installed. Select every product except the Online Text Viewer and choose Install. This will cause Oracle to do whatever database setup is required for each product. Many products (like the TCP/IP protocol adapter) don't have any database objects associated with them, but some (like SQL*Plus and PL/SQL) do. It's safest to select all products, but don't select the Online Text Viewer because the installer will give you an error message if you do.
 - i. Choose Create Product DB Objects for Server Manager.
 - j. Choose Filesystem-Based Database (unless you want to deal with raw devices, in which case you are on your own).
 - k. Choose No to Distribute control files over three mount points. The default database that the installer builds for you will need tweaking whether you distribute files across three mount points or not. The mess is a little easier to clean up if it's all in one directory instead of spread across three.
 - l. Enter the mount point where the installer should create the database—for example, /u01. You should have already created an oradata subdirectory under the mount point and made it owned by the oracle user, or else you will run into file permission problems.
 - m. Choose a character set for the database. US7ASCII is the default on Linux and Unix. (It is not the default on Windows NT.) If you want to use a different character set, then you should have set NLS_LANG and ORA_NLS33 before invoking the installer. You cannot change a database's character set after creation. If you want to store non-English text in your database, such as umlauts and diacritical marks, then you should definitely not use US7ASCII as your character set. See Appendix C of the Installation Guide for a list of recognized character sets.
 - n. Choose a national character set for the database. This is the character set to use for storing data of type NCHAR, NVARCHAR2, and NCLOB. Again, see Appendix C of the Installation Guide for a list of recognized character sets.
 - o. Enter passwords for the SYSTEM and SYS users. Remember that Oracle passwords are not case-sensitive, must start with a letter, and certain special characters are not allowed. I've heard that putting a period in the SYSTEM or SYS password will cause the installer to crash, but I've never tried it.
 - p. If you want to be able to start up and shut down this database without actually logging on to the server, then choose Yes when asked, "Do you want to set the passwords for the internal users (dba and operator)?" This will cause the installer to create a password file external to the database. This is necessary if you will be controlling your database from Enterprise Manager running on a Windows machine.
 - q. Enter a password for the Net8 listener.
 - r. Choose whether or not you want the installer to configure the multi-threaded server (MTS) on your database. MTS is typically not appropriate unless there will be large numbers of OLTP users.
 - s. Choose Yes and OK repeatedly to accept the names and sizes that the installer is proposing for your database files. Actually these defaults are pretty awful, but it's much easier to fix later.
 - t. You might be asked which JDK drivers you wish to install. This is not relevant to database creation, so just choose them all and humor the installer.
 - u. You will be asked if you would like to load the SQL*Plus help facility. You should probably choose No, as the HTML and PDF documentation is infinitely more useful. But if you do like the SQL*Plus help facility, feel free to install it.
 - v. You will be asked if you would like to load the SQL*Plus demo tables. These include the venerable emp and dept tables in the scott/tiger schema. If you are new to Oracle or would like an easy schema to start testing with, then choose Yes. In most cases, however, you'll chose No.
 - w. The installer now creates the database and leaves it running.
 - x. When the database creation is complete you'll receive a message that the requested actions have been performed. Choose OK, and you will be returned to the Software Asset Manager screen.
 - y. Exit the installer.
4. The Oracle Installer probably told you to run root.sh again, but you definitely should not.

- Configure and run the Net8 listener. Edit \$ORACLE_HOME/network/admin/listener.ora to suit your needs. You'll need to fill in the ORACLE_SID. You might need to change the hostname or IP. (In my case my server is multi-homed, but I only want the database to accept connections from the internal network.) You will probably want to leave the extproc settings as they are; extproc is part of the mechanism that allows PL/SQL to call out to procedures outside the database. You may set global_name to the hostname. This will allow clients to connect to the database by specifying the hostname as the Net8 connect string—in this way clients don't need to have a tnsnames.ora file. My listener.ora file looks like this:

```
#
# Filename: Listener.ora
#
LISTENER =
  (ADDRESS_LIST =
    (ADDRESS= (PROTOCOL= IPC) (KEY= MYDB))
    (ADDRESS= (PROTOCOL= IPC) (KEY= PNPKEY))
    (ADDRESS= (PROTOCOL= TCP) (Host= 192.168.1.1) (Port= 1521))
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME= myhost.dbspecialists.com.)
      (ORACLE_HOME= /u01/app/oracle/product/8.0.5)
      (SID_NAME = MYDB)
    )
    (SID_DESC =
      (SID_NAME = extproc)
      (ORACLE_HOME = /u01/app/oracle/product/8.0.5)
      (PROGRAM = extproc)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
```

- Prepare a tnsnames.ora file in \$ORACLE_HOME/network/admin on the server and distribute it to all clients. Edit the default file to suit your needs. Fill in the ORACLE_SID. Change the hostname or IP if needed. My tnsnames.ora file looks like this:

```
#
# Filename: Tnsnames.ora
#
extproc_connection_data =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC) (KEY = MYDB))
    (CONNECT_DATA = (SID = extproc))
  )
MYDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL= TCP) (Host= 192.168.1.1) (Port= 1521))
    (CONNECT_DATA = (SID = MYDB))
  )
MYDB_BEQ =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = BEQ) (PROGRAM = /u01/app/oracle/product/8.0.5)
      (argv0 = oracleMYDB)
      (args = '(DESCRIPTION = (LOCAL=YES) (ADDRESS=(PROTOCOL=BEQ)))')
      (envs = 'ORACLE_HOME=/u01/app/oracle/product/8.0.5,ORACLE_SID=MYDB')
    )
  )
```

7. Create bdump, cdump, and udump directories in \$ORACLE_BASE/admin/\$ORACLE_SID. The installer on Linux violates the OFA (Optimal Flexible Architecture) standard in the placement of these three directories, and we will move them to where they belong.
8. Shut down the database.
9. The default database was created with three control files all in one directory. Move two of the control files to other directories that are on separate physical devices.
10. The default parameter file that the installer created for the database is deficient in several ways. You should move the three parameter files in \$ORACLE_BASE/admin/\$ORACLE_SID/pfile to a backup directory and create a new parameter file from scratch. Some of the things you should correct or improve upon are:
 - a. You might want to merge the config.ora file contents into the init.ora file. Unless you are using Oracle Parallel Server, there is not a lot of value in spreading the parameters between two files.
 - b. Update the control_files parameter to reflect the new control file locations.
 - c. Update the background_dump_dest, user_dump_dest, and core_dump_dest parameters to comply with the OFA standard.
 - d. Update the shared_pool_size, db_block_buffers, and sort_area_size parameters to reasonable values based on how much physical memory your server has available.
 - e. Update the sessions and processes parameters based on how many concurrent sessions you anticipate.

Here's the parameter file I ended up with.

```
#
# initMYDB.ora
# =====
#
# Parameter file for MYDB database.
#
# config parameters
control_files              = (/u02/oradata/MYDB/control01.ctl,
                             /u03/oradata/MYDB/control02.ctl,
                             /u04/oradata/MYDB/control03.ctl)
background_dump_dest      = /u01/app/oracle/admin/MYDB/bdump
core_dump_dest             = /u01/app/oracle/admin/MYDB/cdump
user_dump_dest            = /u01/app/oracle/admin/MYDB/udump
db_block_size             = 8192
db_files                  = 80
db_name                   = MYDB
compatible                = 8.0.5
rollback_segments         = (r01,r02,r03,r04)
# tuning parameters
shared_pool_size          = 16000000
sort_area_size            = 4096000
db_block_buffers          = 2000
db_file_multiblock_read_count = 64
processes                 = 100
sessions                  = 100
log_checkpoint_interval   = 10000
log_buffer                 = 163840
sequence_cache_entries    = 100
sequence_cache_hash_buckets = 89
job_queue_processes       = 2
max_dump_file_size        = 10240 # limit trace file size
                             # to 5 Meg each
```

11. Restart the database so that the new parameter settings take effect.
12. Tweak the default install so that the database will be usable for more than trivial tasks and will be capable of offering decent performance. Here are some of the things you'll probably want to do:
 - a. Relocate data files to spread them over multiple physical devices.
 - b. Enlarge the online redo logs—the default log size of 500 Kb is often too small. Since you can't resize online redo logs, you'll need to drop and recreate them. At the same time, you can give the logs OFA compliant names, and you can relocate them as appropriate to distribute I/O.
 - c. Resize the temporary tablespace—the default size of 1 Mb won't be big enough unless your database will be extremely small.
 - d. Alter the temporary tablespace to give it a content type of TEMPORARY and appropriate default storage parameters. Also, make sure all users including SYS and SYSTEM have the TEMP tablespace designated as their temporary tablespace.
 - e. Resize the rollback segment tablespace—the default size of 15 Mb won't be big enough unless your database is somewhat small.
 - f. Depending on the anticipated size of your database and the expected number and type of concurrent users, you may want to adjust the number of rollback segments and their storage parameters. At the very least, you'll probably want to set the OPTIMAL storage parameter for each rollback segment except SYSTEM.
 - g. Adjust the sizing and default storage of the TOOLS and USERS tablespaces as needed if you will be using these tablespaces.
13. Create new tablespaces for holding application segments. Create separate tablespaces with data files on separate physical devices for tables and indexes. You may want to split your application segments into several tablespaces, based on object size, permanence, volatility, I/O volume, or any of a number of other criteria.
14. Choose default storage parameters carefully for each application tablespace. One strategy I highly recommend is as follows:
 - a. Set INITIAL to either 128k, 4m, or 128m, depending on the planned sizes of the objects to be placed in the tablespace.
 - b. Set NEXT the same as INITIAL.
 - c. Set MINEXTENTS to 1 and MAXEXTENTS to 1024.
 - d. Set PCTINCREASE to 0.
15. Create application roles if desired. Alternatively, you can use the default roles CONNECT, RESOURCE, and DBA.
16. Create your application users that will own the application schemas. Set the default tablespace to one of your application tablespaces designated to hold tables, and set the temporary tablespace to TEMP. Assign quotas on all of the application tablespaces where the user will need to be able to create schema objects. (You can use the keyword UNLIMITED.) You should not set any quota on the temporary tablespace. Do not plan to create any application objects in the SYS or SYSTEM schemas, or store any application objects in the SYSTEM or TEMP tablespaces.
17. Grant roles and/or system privileges to the application users. Note that if you grant the RESOURCE role to a user, that user will also receive the UNLIMITED TABLESPACE system privilege. This will let the user create objects in any tablespace they wish, regardless of quotas. I recommend you revoke UNLIMITED TABLESPACE from all users other than SYS.

Complete the Server Configuration

These steps complete the configuration of your server for smooth Oracle operation. These steps could have been performed earlier, but are more straightforward if performed after a database has been created. In this section we will configure the server to start the database and Net8 listener automatically whenever the server is rebooted, change the oracle user's login script to eliminate hardcoding, and create individual operating system accounts for each database user.

1. Edit the `/etc/oratab` file to verify that the entry for your database is correct. All of the lines in the file should be comments (starting with a pound symbol) except for one. This one line should contain the name of your Oracle instance, its `ORACLE_HOME`, and a Y or N. A Y indicates that the database should be started automatically on server reboot, and an N indicates that it should not. The three fields should be separated by colons. A sample `/etc/oratab` file looks like this:

```
#
# /etc/oratab
# =====
#
MYDB:/u01/app/oracle/product/8.0.5:Y
```

2. Edit the login file for the oracle user to eliminate hardcodings and call the `oraenv` script to set the environment instead. The following should work with Bash, Bourne shell, or Korn shell:

```
# Settings for Oracle environment
ORACLE_SID=MYDB
ORAENV_ASK=NO
export ORACLE_SID ORAENV_ASK
. oraenv
```

3. Create separate Linux accounts for DBAs and database users who will log onto the server. You should only log in as oracle when installing or patching software. The Linux accounts for DBAs should be members of the `dba` group, and other users should not be members of the `dba` group. Give each of these accounts a login file like oracle's so that their environment initializes correctly when they log in.
4. To make the database and Net8 listener start up automatically when the server reboots, you'll need to create a `dbora` file in `/etc/rc.d/init.d` and link it to `/etc/rc.d/rc3.d`. You'll need to do this as the root user. First create a file called `dbora` in `/etc/rc.d/init.d` as follows:

```
#!/bin/sh
ORA_HOME=/u01/app/oracle/product/8.0.5
ORA_OWNER=oracle
if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
    echo "Oracle startup: cannot start"
    exit
fi
case "$1" in
    'start') # Start the Oracle databases and Net8 listener
        su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart" &
        su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start" &
        ;;
    'stop') # Stop the Oracle databases and Net8 listener
        su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop" &
        su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut" &
        ;;
esac
```

Note that the spacing around the brackets shown here is different from what appears in the Oracle documentation.

After creating the dbora file, you need to link it to /etc/rc.d/rc3.d:

```
ln -s /etc/rc.d/init.d/dbora /etc/rc.d/rc0.d/K10dbora
ln -s /etc/rc.d/init.d/dbora /etc/rc.d/rc3.d/K10dbora
ln -s /etc/rc.d/init.d/dbora /etc/rc.d/rc3.d/S99dbora
```

Conclusion

This document walks you through all of the intricate details of getting Oracle up and running on the Linux platform. It may look complicated, but that's only because this document goes down to a nitty gritty level of detail.

Please keep in mind, though, that the requirements are different for every Oracle implementation. I am extremely confident that if you follow these steps to install Oracle 8.0.5.0.0 Standard Edition on a server running Red Hat 5.2 Linux, then the process will go very smoothly for you. However, no single document can address every specific hardware configuration and every set of business needs. Please use this document as a starting point to get Oracle on Linux up and running in your shop. To get the best performance and scalability, each system needs to be considered individually.

About the Author

Roger Schrag has been an Oracle DBA and application architect for over ten years, starting out at Oracle Corporation on the Oracle Financials development team. He is the founder of Database Specialists, Inc., a consulting group specializing in business solutions based on Oracle technology. You can visit Database Specialists on the web at <http://www.dbspecialists.com>, and you can reach Roger by calling +1.415.344.0500 or via email at rschrag@dbspecialists.com.