



Deploying Oracle9i Database on Red Hat® Linux® 3

Jennifer E. Lamb
March 2004

Abstract

This document examines how enterprise-class databases may be deployed using Red Hat Enterprise Linux 3 and Oracle9i Database. An overview of the status of Red Hat Enterprise Linux AS as a platform for Oracle® database technology. A tutorial outlining the steps required to install Red Hat Enterprise Linux AS 3 and Oracle9i Database is provided.

Table of Contents

Introduction.....	3
Oracle Product Availability on Red Hat Enterprise Linux.....	3
Deploying an Oracle Database on Red Hat Enterprise Linux AS 3 - A Tutorial.....	4
The Database.....	4
Resources.....	18

Introduction

Red Hat Enterprise Linux has become a core component of most enterprise IT environments and an infrastructure standard in the Oracle database space. Firms such as AOL, Verisign and Amazon have found that Red Hat Enterprise Linux systems prove themselves time and again with regard to performance, reliability, and security, and with impressive Total Cost of Ownership (TCO) advantages.

Red Hat Enterprise Linux version 3, released in October, 2003, is being embraced in the technical arena as an integral part of a complete platform that combines the power of Oracle database technology with the advantages of a UNIX-like environment running on x86-compatible hardware.

Red Hat has been able to elevate Linux to an enterprise-ready workhorse by working closely with partners like Oracle to perfect the underlying technologies that make up the enterprise operating system. The Red Hat/Oracle partnership has strengthened as the Red Hat Enterprise Linux product line has evolved. Red Hat and Oracle testing teams have collaborated to create communication channels for efficient engineering exchanges that have resulted in joint kernel development for specific areas of Red Hat Enterprise Linux 3. For example, Red Hat engineering included approximately 50 enhancements in Red Hat Enterprise Linux 3 that are directly related to improving performance scalability and stability with Oracle's entire product stack. Red Hat's ongoing relationship with Oracle includes a joint support offering which means Red Hat Enterprise Linux customers will have both Oracle and Red Hat available for assistance should the need arise.

Oracle Product Availability on Red Hat Enterprise Linux

Oracle releases two versions of its database simultaneously with releases of Red Hat Enterprise Linux. In addition, Oracle has released a select set of additional application components, as illustrated below:

Product	Release
Oracle9i and Oracle9i RAC Database	Oracle9i Database Release 2 Enterprise Edition (9.2.0.1.0)
Oracle8i Database	Oracle8i Enterprise Edition Release 3 (8.1.7.0.1)
Oracle9i AS Application Server	Oracle9i AS Release 2 v9.0.2
OracleE-Business Suite	Oracle E-Business Suite, rel 11.5.7
Oracle Collaboration Suite	

Table 1. Oracle Products Supporting Red Hat Enterprise Linux

Deploying an Oracle Database on Red Hat Enterprise Linux AS 3 - A Tutorial

The Database

The database used in this tutorial is Oracle9i Database Release 2 (9.2.0.1.0). To simplify the installation process outlined in this tutorial, install a single instance database without the RAC or Red Hat Cluster Manager clustering capabilities. Standard sample Oracle tables are used for all data sources rather than custom tables and records.

The Database Version

Oracle9i Database Release 2 (9.2.0.1.0)

Oracle9i is available for free download under a special developer license from: <http://otn.oracle.com>

The Operating System Version

Red Hat Enterprise Linux AS 3

The Hardware

Red Hat Enterprise Linux is most frequently deployed on x86-compatible systems that provide increased performance at significantly lower cost. The hardware used for this configuration is represented in the table below:

	x86 Configuration
System Name	DL360
Manufacturer	Compaq
CPU type	1000Mhz Pentium III
CPU count	2
RAM	2 GB
Local Disk	2 @ 36GB
System Size	1U

Table 3. Hardware Configurations

The Installation Plan

Installation and configuration will occur in this order:

1. Install Red Hat Enterprise Linux AS 3
2. Configure and update the operating system
3. Install the Oracle9i Database Server
4. Configure and test the Oracle9i Database Server

Install Red Hat Enterprise Linux AS 3

If necessary, please refer to the installation manual for the target system architecture online at:

<http://www.redhat.com/docs/manuals/enterprise/>

Most system administrators find the graphical installation program for Red Hat Enterprise Linux AS straight forward and intuitive. During this tutorial, This tutorial uses the default installation values. Instances where this is not the case are noted.

Start by placing CD #1 in the system's CD-ROM drive, and rebooting. Select the default graphical installation program by pressing [Enter] at the `boot:` prompt.

The graphical installation program guides the user through the installation. Follow the instructions below clicking the **NEXT** button after the fields on each screen until they are properly filled in. In most circumstances it is safe to accept the defaults because the installer will try to select the best values. However, in specific environments the default values may not be appropriate and should be appropriately adjusted.

Language Selection - Accept the defaults.

Keyboard Configuration - Accept the defaults.

Mouse Configuration - Accept the defaults.

Installation Type - Select **Advanced Server**.

Note: Experienced users can select the **Custom** installation type to apply specific package sets.

Disk Partitioning Setup - Select **Manually partition with Disk Druid**. Create the following partitions:

Partition	Type	Size
/boot	ext3	150 MB
swap	swap	2x physical RAM, minimum of 1024 MB
/	ext3	1024 MB
/tmp	ext3	1024 MB

/usr	ext3	3072 MB
/var	ext3	512 MB
/home	ext3	3072 MB (this should be large enough to hold the three Oracle9i installation Cds)
/opt	ext3	6144 MB (fill to maximum allowable size)

Table 4. Recommended Partitions

Boot Loader Configuration - Accept the defaults.

Boot Loader Password Configuration - Accept the defaults.

Network Configuration - Accept the defaults.

Firewall Configuration - Select **No firewall**.

Additional Language Support - Accept the defaults.

Time Zone Selection - Choose the time zone that is appropriate for the location of the system. If the system's clock is set to UTC time, select **System clock uses UTC**.

Account Configuration - Enter a root password. The users and groups required for the installation of Oracle9i are created later in the tutorial.

Authentication Configuration - Accept the defaults.

Package Group Selection -The **GNOME** group should already be checked. Also add the **Development Tools** group and **Legacy Software Development** group, as these are necessary for the successful installation of Oracle9i.

Graphical Interface (X) Configuration - Accept the defaults.

Installing Packages - At this point, the installation program installs the packages that have been selected. A screen will display the status of each package as it is installed, along with a tally of total packages, installed packages, and packages to be installed.

Boot Disk Creation -Insert a blank, formatted 1.44 MB floppy

Monitor Configuration - Accept the defaults.

Customize Graphics Configuration - Change the login type to **Text**.

The installation of Red Hat Enterprise Linux AS 3 is now complete. Boot the

latest enterprise kernel ([kernel--2.4.21-4ELsmp](#)) which is used on SMP systems with up to 16 GB of RAM. Systems with more than 16GB of RAM should use the hugemem kernel discussed earlier in this document.

Configure and update the operating system

It is critical that the network configuration for the operating system is correct because Oracle9i database is a networked service and the system needs to be networked to get Red Hat Network packages. RHN is used to update the system with the latest errata released from Red Hat. The errata releases contain important bug fixes and security updates that should be applied to all production systems. The actions performed here must all be performed while logged into the operating system as the root user.

Change the hostname in `/etc/sysconfig/network`. For the purposes of this document the hostname we will be using is `oracle1.example.com`.

```
NETWORKING=yes
HOSTNAME=oracle1.example.com
```

Edit `/etc/hosts` and add the system's hostname to the `127.0.0.1` line before the `localhost.localdomain` and `localhost` entries:

```
127.0.0.1 oracle1.example.com oracle1 localhost.localdomain localhost
```

Inform the system of the change by using the following command (from the root shell):

```
hostname oracle1.example.com
```

Oracle recommends adding these entries to `/etc/resolv.conf` to allow more time for name resolution to succeed (otherwise on busy systems, Oracle SQL*Net connections may fail):

```
options attempts:5
options timeout:15
```

After configuring the network, register the system with RHN. Begin by running (as root) `up2date` to register the system:

```
up2date
```

Note: The Red Hat Network Management entitlement provides the ability to use Activation keys so that all of this can be done non-interactively (see <http://rhn.redhat.com> for more details).

After registering the system it should automatically be entitled to receive RHN updates. Verify that the system has been entitled at the RHN web interface <http://rhn.redhat.com> (or at the address of the RHN Satellite Server, if this option applies).

The kernel errata updates are disabled by default to give system administrators complete control over the kernel update process for production systems. Configure the RHN entitled database server to receive kernel updates by editing `/etc/sysconfig/rhn/up2date`. Find the line that reads:

```
pkgSkipList=kernel*;
```

and change it to:

```
pkgSkipList=;
```

Next, find the line that reads:

```
removeSkipList=kernel*;
```

and change it to:

```
removeSkipList=;
```

At this point, use RHN to update the system with the latest errata available from Red Hat. To do this, use either the RHN web interface accessible via the toolbar icon, or issue the following command on the system to be updated:

```
up2date -u
```

After the system has been successfully updated, verify the default kernel is the latest kernel update in `/boot/grub/grub.conf`. Reboot to use the new kernel.

Make sure several key compatibility libraries are installed on the Red Hat Enterprise Linux 3 system prior to installing the Oracle9i Database Server. To verify that these libraries are present type:

```
rpm -q compat-gcc compat-libstdc++ compat-gcc-c++ compat-db
```

If these libraries are not present on the system use the up2date client to download and install them.

Next, make sure the system uses gcc 2.96 rather than gcc 3.2 while installing the Oracle Database. To make gcc 2.96 the default, type the following:

```
mv /usr/bin/gcc /usr/bin/gcc323
ln -s /usr/bin/gcc296 /usr/bin/gcc
mv /usr/bin/g++ /usr/bin/g++323
ln -s /usr/bin/g++296 /usr/bin/g++
```

To restore the system to gcc3.2 after the Oracle installation type:

```
rm /usr/bin/gcc
mv /usr/bin/gcc32 /usr/bin/gcc
```

For more information on using Red Hat Network, visit <http://rhn.redhat.com>.

Install the Oracle9i Database

The Oracle9i database server will be installed under `/opt/oracle`, using the General Purpose starter database provided with the Oracle9i installation media. The Oracle database System ID (SID) is `orcl`. Some database administrators may choose to install Oracle9i using a different configuration, however, the following steps still apply.

The installation of Oracle9i Database under UNIX or Linux involves multiple configuration steps, with some commands issued as `root`, and some issued as the `oracle` user. It is critical that the kernel and system parameters are configured properly; otherwise, the creation of the database may fail. Begin as the `root` user to configure the system and then move to the installation of Oracle9i itself.

As `root`, amend the file `/etc/sysctl.cf` to include the following:

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.default.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
# Parameters for Oracle9i Release 2 (9.2.0)
kernel.sem = 250 32000 100 128
kernel.shmmax = 2147483647
kernel.shmmni = 4096
kernel.shmall = 2097152
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
```

Note: The SHMMAX parameter in `/etc/sysctl.cf` should be able to accommodate the entire SGA. Note that Oracle9i Database on Red Hat Enterprise Linux supports the creation of an SGA above these sizes for systems with more than 4 GB of RAM; for more information, please see the Oracle whitepaper *Tips and Techniques: Install and Configure Oracle9i on Red Hat Enterprise Linux Advanced Server, Aug 2002* in the final section of this paper.

After editing the `/etc/sysctl.cf` file, as `root` run the command

```
sysctl -p
```

to make the changes take effect. Alternatively, reboot the system.

Next add the appropriate groups and users required for the Oracle9i installation. Issue the following commands (while logged in as `root`):

```
groupadd dba
groupadd oper
groupadd oinstall
useradd -g oinstall -G dba,oper oracle
passwd oracle
```

This creates a new user account (named `oracle`) whose primary group is `oinstall` and whose secondary groups are `dba` and `oper`.

Increase the number of files the `oracle` user can open and make sure PAM can read the `/etc/security/limits.conf` file by ensuring that the following lines are in `/etc/pam.d/system-auth`:

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

Add the following lines to `/etc/security/limits.conf`:

```
oracle soft nfile 4096
oracle hard nfile 8192
```

The above changes go into effect the next time the `oracle` user logs into the system.

After the `oracle` user is created and the file handles have been increased, create and change the ownership of the Oracle9i installation directory for that user. This is necessary because `/opt/oracle` is used as our `$ORACLE_BASE` directory. Issue the following commands while logged in as `root`:

```
mkdir /opt/oracle
chown -R oracle /opt/oracle
```

Just as it is important to configure the system to support Oracle9i, it is also important to configure the environment for each user that will access the Oracle database. Log out of the `root` account, and log in as the `oracle` user to perform the following actions (note that Oracle9i must be installed as the `oracle` user that was just created - not as the `root` user).

Create a file called `libcwait.c` in the `oracle` user's home directory to avoid getting an error when running the Oracle installer. The `libcwait.c` file should contain the following text:

```
/*
    gcc -O2 -shared -o $HOME/libcwait.so -fpic $HOME/libcwait.c
*/
#include
#include
#include
```

```
#include
pid_t
__libc_wait (int *status)
{
    int res;
    asm volatile ("pushl %%ebx\n\t"
"movl %2, %%ebx\n\t"
"movl %1, %%eax\n\t"
"int $0x80\n\t"
"popl %%ebx"
: "=a" (res)
: "i" (__NR_wait4), "0" (WAIT_ANY), "c" (status), "d" (0), "S" (0));
return res;
}
```

Next, compile this file in the `oracle` user's home directory by executing the following command:

```
gcc -O2 -shared -o $HOME/libcwait.so -fpic $HOME/libwait.c
```

Set the `LD_ASSUME_KERNEL` and `LD_PRELOAD` environment variables:

```
export LD_ASSUME_KERNEL=2.4.19
export LD_PRELOAD=$HOME/libcwait.so
```

Alternatively, the above symbol issue can be remedied by downloading and installing the `p3006854_9204_LINUX.zip` from <http://metalink.oracle.com>

Download or copy the Oracle9i installation files to the `oracle` user's home directory. The following files should exist files under `/home/oracle/MEDIA1`:

```
lnx_920_disk1.cpio
lnx_920_disk2.cpio
lnx_920_disk3.cpio
```

Run the following commands (as `oracle`) to extract the files:

```
cd ~/MEDIA
cpio -idmv < lnx_920_disk1.cpio
cpio -idmv < lnx_920_disk2.cpio
cpio -idmv < lnx_920_disk3.cpio
```

Start the X Window System.

```
startx
```

Once X has started, open two terminal windows. In one of the windows, issue the following command to obtain a shell with root privileges:

```
su -
```

¹ A special developer's license of Oracle9i can be downloaded from <http://otn.oracle.com>.

Use this root shell to execute some commands that must be done by `root` during the course of the installation.

As the `oracle` user, run the `runInstaller` script. It is located in the `Disk1` directory that was created when the Oracle9i files were extracted from the `.cpio` files. This will start the Oracle Universal Installer.

```
cd MEDIA/Disk1
./runInstaller
```

NOTE: Two linking errors could occur during the Oracle 9i installation. The first occurs because the install target for `$ORACLE_HOME/lib/stubs` was not used. There could be an error from the install target to `$ORACLE_HOME/network/lib/ins_oemagent.mk` reference. To continue installing through this error, comment out the following line in `$ORACLE_HOME/lib/ins_oemagent.mk`:

```
# LDFLAGS = $(STDMODE) -L$(LIBHOME) -L$(ORACLE_HOME)/rdbms/lib -L
$(ORACLE_HOME)/network/lib
```

Click **RETRY** to resume the installation.

The second linking error references a hidden symbol:

```
$ORACLE_HOME/ctx/lib/ins_ctx.mk because /usr/bin/ld: ctxhx: hidden symbol
`stat' in /usr/lib/libc_nonshared.a(stat.oS)
```

and can safely be ignored.

Oracle Universal Installer - The Oracle Universal Installer will begin. Click **Next** to continue.

Inventory Location - Accept the default Inventory Location and click **OK**

UNIX Group Name - Enter `oinstall` for the UNIX Group Name and click **Next**.

At this point a script called `/tmp/orainstRoot.sh` should be run as `root`. In the root shell previously opened execute:

```
/tmp/orainstRoot.sh
Creating Oracle Inventory pointer file (/etc/orainst.loc)
Changing groupname of /opt/oracle/orainventory to oinstall.
```

After executing the script, click **Continue**.

File Locations - Accept the defaults for File Locations and click **Next**

Available Products - Select the default for Available Products (Oracle9i Database 9.2.0.1.0) and click **Next**

Installation Types - Select the default Installation Type (Enterprise Edition) and click **Next**

Database Configuration - Select the default Database Configuration (General Purpose) and click **Next**.

Database Identification - The default SID (`orcl`) should already be selected. For the Global Database Name, enter something like `SID.domain` (for example, `orcl.example.com`), then click **Next**.

Database File Location - Accept the default Database File Location (`/opt/oracle/oradata`) and click **Next**.

Database Character Set - Accept the default Database Character Set and click **Next**.

Summary - The Summary screen will appear. Click **Install** to begin the installation of the database software.

Install - During the installation, run the `root.sh` script as the `root` user after being prompted by the installer. To do this, execute the script:

```
/opt/oracle/product/9.2.0.1.0/root.sh
Running Oracle9 root.sh script...
\nThe following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME= /opt/oracle/product/9.2.0.1.0

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

After the script has finished executing, click **OK** to continue.

Configuration Tools - Once the database server installation is complete, the Configuration Tools screen appears. This will run through the through the following steps:

- Oracle Net Configuration Assistant
- Oracle Database Configuration Assistant (DBCA)
- Agent Configuration Assistant
- Starting HTTP Server

NOTE: Errors could occur setting up the configuration tools. These errors are caused by a need for a

JRE on the Red Hat Enterprise Linux 3 system. The next step is to install a working JRE and make the location of this JRE be JAVA_HOME 2. Rename \$ORACLE_HOME/oracle.swd.jre to \$ORACLE_HOME/oracle.swd.jre.bak:

```
mv $ORACLE_HOME/oracle.swd.jre $ORACLE_HOME/oracle.swd.jre.bak
```

Create a link from jre to java in the \$JAVA_HOME/bin directory by typing the following:

```
ln -s $JAVA_HOME/bin/java $JAVA_HOME/bin/jre
```

Next , link \$ORACLE_HOME/oracle.swd.jre to \$JAVA_HOME:

```
ln -s $JAVA_HOME $ORACLE_HOME/oracle.swd.jre
```

Once the JRE for Oracle Database is working, run the Oracle Database configuration tools as part of the install process or run them later. (To run the ODCA execute \$ORACLE_HOME/bin/dbca as the `oracle` user)

The Database Configuration Assistant (DBCA) takes a while to start, so do not be concerned by the blank window that initially appears. If the following error occurs during the execution of the DBCA, there is a good possibility the kernel parameters were not properly configured.

```
ORA-03113: end-of-file on communication channel
```

If this occurs, exit the installation, set the kernel parameters as detailed earlier in this document, and restart the installation. When the DBCA is complete, the installer prompts for the Oracle database's SYS and SYSTEM user passwords. Do this, then click **OK**.

End of Installation- Oracle9i Database has successfully been installed. Click **Exit** to exit the Oracle Universal Installer.

The **Oracle Enterprise Manager Console, Standalone** should now appear. Exit this tool for now.

Configure and test the Oracle9i Database Server

Assuming the installation went well, the database server and the TNS Listener should both be up and running. To verify this, as the `oracle` user issue the command:

```
tnsping orcl
```

There should be an OK at the end of the output. This indicates that the TNS listener (the service that listens for incoming SQL*Net connections across the network) is working properly.

Next log in to SQL-Plus² and issue a sample SQL query:

```
sqlplus scott/tiger
```

```
SQL> select * from emp;
SQL> exit
```

After determining that the database server is up and running, create an init script to start and stop the database services when the system boots and is shut down. To do this, log in as root and create a script named `/etc/init.d/dbora`. While many DBAs choose to customize their init scripts, the example below will suffice for the purposes of this tutorial:

```
#!/bin/bash
#
# Startup script for Oracle9i R2
#
# chkconfig: 345 99 10
# description: Oracle9i R2 RDBMS

ORA_HOME=/opt/oracle/product/9.2.0.1.0
ORA_OWNER=oracle

if [ ! -f $ORA_HOME/bin/dbstart ]
then
    echo "Oracle startup: cannot start"
    exit
fi

case "$1" in
    'start')
        su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start" > /dev/null
        2>&1
        su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart" > /dev/null 2>&1
        ;;
    'stop')
```

```
su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop" > /dev/null 2>&1
su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut" > /dev/null 2>&1
;;

*)
echo "Usage: dbora {start|stop}"
exit 1
```

```
esac
```

Next, make the script executable:

```
chmod 755 /etc/init.d/dbora
```

Use `chkconfig` to turn the service on such that it will start in runlevels 3, 4, and 5.

```
/sbin/chkconfig dbora on
```

The `dbstart` and `dbshut` commands called from the init script read the `/etc/oratab` file to determine which databases are to started or stopped. By default, this file is written so that the `dbstart` and `dbshut` commands do not affect the database created during the installation of Oracle9i. To change this, edit `/etc/oratab` and change the entry for the SID (in this case `orcl`) to have Y at the end of the line. The file should look like this :

```
# This file is used by ORACLE utilities. It is created by root.sh
# and updated by the Database Configuration Assistant when creating
# a database.

# A colon, ':', is used as the field terminator. A new line terminates
# the entry. Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively. The third field indicates
# to the dbstart utility that the database should , "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
*/opt/oracle/product/9.2.0.1.0:N
orcl:/opt/app/oracle/product/9.2.0.1.0:Y
```

Shut down the database server using `dbshut` as the `oracle` user:

```
Dbshut
```

Create the initialization file for the `orcl` database as the `oracle` user. Name the file `initSID.ora` (in this case our SID is `orcl`, so the file should be called `initorcl.ora`). An error is generated when the database is started if this file is not created:

```
Can't find init file for Database "orcl".
Database "orcl" NOT started.
```

The easiest way to create this file is to issue the following commands as the `oracle` user.

```
cd $ORACLE_HOME/dbs
cp init.ora initorcl.ora
```

This creates a basic initialization file template. In a production environment, it is quite likely that the DBA will want to modify this file, as it contains a number of configuration parameters for the database. For now, however, it is fine to use the default values contained in this file.

The system is now configured to automatically start and stop the database when the system starts and stops. The init script will start and stop both the listener and the database server itself.

To shut down the database and listener issue the following command:

```
/etc/init.d/dbora stop
```

To start the database and the listener issue the following command:

```
/etc/init.d/dbora start
```

The last configuration step we will take is to enable asynchronous I/O, a feature that can improve the performance of the Oracle database. Oracle did not enable asynchronous I/O by default to accommodate other Linux distributions without this feature. To enable asynchronous I/O, log in as the `oracle` user and begin by shutting down the database server.

```
Dbshut
```

Once the database has shut down, relink Oracle9i to use asynchronous I/O. This is done by issuing these commands:

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk async_on
make -f ins_rdbms.mk ioracle
```

Next, edit the `initSID.ora` file (`$ORACLE_HOME/dbs/initorcl.ora` in this example) and add the following to the end of the file:

```
# asynch i/o configuration
disk_asynch_io = true
filesystemio_options=asynch
```

Finally, restart the database.

```
Dbstart
```

To ensure that the database is now configured to start and stop properly, repeat the `tnsping` and `sqlplus` commands described in Section 5.4.4. At this point, the database is ready for any testing and configuration that may be required.

Resources

- *Proven Performer* (Oracle9i RAC TPC-C results):
<http://www.redhat.com/mktg/oracle9i>
- *Oracle Delivers World Record TPC-C Benchmark on Linux; Outperforms Microsoft SQL Server by 14% on Identical Processors:*
<http://www.oracle.com/corporate/press/index.html?1456818.html>
- *Tips and Techniques: Install and Configure Oracle9i on Red Hat Enterprise Linux Advanced Server, Aug 2002:*
http://otn.oracle.com/tech/linux/pdf/installtips_nal.pdf
- *Oracle9i Database Release 2 on Linux: Performance, Reliability and Manageability Enhancements on Red Hat Enterprise Linux Advanced Server 3:*
<http://otn.oracle.com/tech/linux/pdf/9iR2-on-Linux-Tech-WP-Final.PDF>
- *Red Hat, Inc.'s Network Console and Crash Dump Facility:*
<http://www.redhat.com/support/wpapers/redhat/netdump/index.html>
- *Database: Linux Powered and Enterprise Ready* (Unbreakable Linux overview):
http://www.oracle.com/features/9i/index.html?t1db_unbrlinux.html
- Oracle at Red Hat, Inc.:
<http://www.redhat.com/oracle>
- *Tuning and Optimizing Red Hat Linux Advanced Server for Oracle9i Database:*
<http://www.puschitz.com/TuningLinuxForOracle.shtml>

Upgrading from Red Hat Enterprise Linux 2.1 AS
To Red Hat Enterprise Linux 3 Author: Wim Coekaerts, Director of Linux
Engineering
Date: January 2004