Construction of an Oracle 10g Linux Server

V0.5 29SEP2005

Synopsis

This is a step-by-step Construction Guide to build a Linux Server from scratch, and then to install the Oracle 10g RDBMS in an industry-standard way on it. It also shows you how to set up the server's network and how to configure a Linux-based and Windows-based client environment.

You need the following: an X86-based server (a good PC will do), Linux and Oracle installation disks and a network to connect to. A PC that can connect to the network, with either Linux or Windows installed is also useful.

The Linux distributions used here are Red Hat Enterprise and Red Hat Fedora Core. Nearly any other Linux distribution will also work, although it is useful to check in the respective forums for some finer points.

1. Introduction

For clarity, the path of lowest complexity is chosen. Although there may be better ways to achieve the desired result, this would cause this document to become too complex.

1.1. Document conventions

Example source code on the console command line is preceded by the '$' prompt for a non-root user, and by the '#' prompt for the root-user. The password needs to be entered wherever the 'Password' prompt appears. This is illustrated below:

```
$ whoami
oracle
$ su –
Password:
# whoami
root
```

Assume that all the example code in this document is case sensitive. Linux (and all other UNIX's) is a case sensitive operating system. Note however that the SQL language used by Oracle is not case sensitive (although Oracle data is by default).

2. Preparation

2.1. Prerequisites

The minimum requirements for setting up an Oracle 10g database on a Linux server are:

- A server with at least 256Mbyte memory and 10Gbyte disk space

- Basic graphics card, monitor, keyboard and mouse

- Windows 2000 or higher client devices that will connect to the database

- A network infrastructure that will support the clients and the servers

- Red Hat Linux installation disks - Red Hat Enterprise 3 (AS) or later is the preferred distribution and is also supported by Oracle. It is possible to use Red Hat's Fedora community releases – verified to work with Fedora Core 3, although there is a small additional step.

- Oracle 10g RDBMS for Linux server installation disks

- Oracle 10g Client installation disks for Windows

2.2. Server preparation

If the server's disks are RAID'ed, ensure that your Red Hat distribution supports the particular RAID controller. Setting up RAID controllers is not in the scope of this document, and the reader is referred to reference [1]. Fortunately, most common RAID controllers are supported by the Red Hat Linux distribution without much configuration.

When the Red Hat Enterprise 3 (AS) distribution is installed on the target server, it will remove all the existing data from the server. Back any important data up before proceeding with the installation.

The installation is a graphical-based installation, which means that a monitor, keyboard and mouse are connected. A Network Interface Card (NIC) should also be installed. Red Hat supports most NIC's.

2.3. Network environment preparation

Ensure that the basic network has been set up and works by pinging the devices that will participate in this environment. If a firewall is located between the servers and the clients, open the following ports:

- 22 for SSH

- 1521for Oracle's SQL*Net protocol (default port).

3. Guide to installing Red Hat Linux

The following instructions apply to the Enterprise Edition and the Fedora Core distributions:

3.1. Linux Operating System Installation

3.1.1. Booting up

Boot the server up with the Red Hat Enterprise Disk 1 loaded in the server's CD tray

3.1.2. Basic configuration

Select the basic parameters, such keyboard layout, time zone, etc.

3.1.3. Disk partitioning

When the installer prompts about the disk partitioning, select the automatic option and delete all existing partitions. This will create a boot partition of approximately 100Mbyte and a swap partition of approximately 500Mbyte. The remaining (third partition) will be used to host the operating system, RDBMS and for the purposes of this guide, the data as well.

3.1.4. Basic Network configuration

Select either DHCP or fixed IP address and hostname in the required network settings window. If you select DHCP, ensure that you also manually enter the host name. For argument's sake a host name of ORACLE1 is selected here, as it will appear on many of the example scripts.

Generally, it is preferable to use fixed rather than dynamically allocated IP address for servers in network environments.

3.1.5. Root access

Enter the password for the super user; 'root'. Other users will be later created.

3.1.6. Package selection

Select the Custom installation option, scroll down to the bottom of the list of packages to install and select to install all packages. This will save lots of time and agonising about which packages might be required later on.

3.1.7. Firewall

Select the firewall option, and open ports 22, 80, 1521 and 10000 to operate.

3.1.8. Installation

Proceed with the installation and insert CD's when required.

3.1.9. Rebooting

At the end of the installation the server will reboot. If all goes well, you will be prompted with a graphical log-on screen. Select the KDE-desktop and log in as user 'root' using the previously supplied password.

Check that it comes up with no errors. Log in to the server can view the start up message log:

```
[root@ oracle1]$ less /var/log/messages
```

Check that networking works by pinging another device on the network, e.g.:

```
[root@ oracle1]$ ping 10.44.4.44
64 bytes from oracle1 (192.168.0.2): icmp_seq=1 ttl=64 time=0.338 ms
64 bytes from oracle1 (192.168.0.2): icmp_seq=1 ttl=64 time=0.338 ms
64 bytes from oracle1 (192.168.0.2): icmp_seq=1 ttl=64 time=0.338 ms
etc..
```

Hit Ctrl-C to stop the pinging loop.

Linux has now been installed on the server. A few services still need to be configured to make this server useful to other devices.

4. Configuring Linux services

A number of services need to be configured so that the server can be administered and monitored from a remote client PC. Most of them rely on the network service being operational and correctly configured.

4.1. Basic Network configuration

During Linux installation, the network cards will be recognized and corresponding drivers will have been installed. Should Linux not recognize the network card, try a different network card, or experiment with different network card drivers.

Once the server with its network card is up and running, log in as root, and run the following application from the X-terminal:
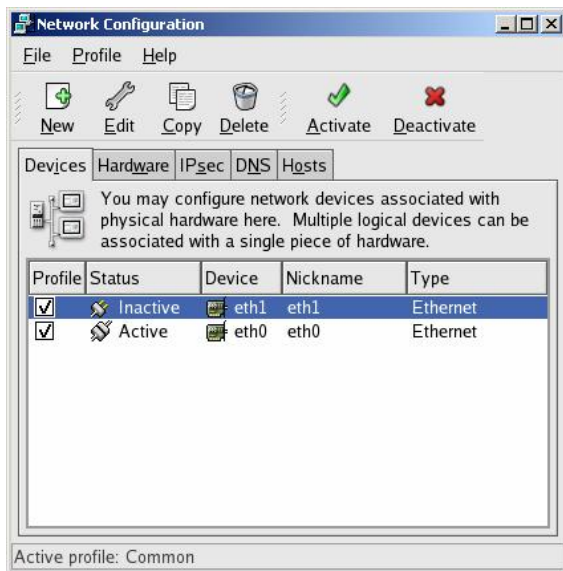
```
# neat &
```

Figure 1

The application shown in Figure 1 will allow you to set the IP address and other network settings for each network card. Note that for a typical server there will be redundant NIC's, which each will require unique fixed IP addresses if not using DHCP. When all the adjustment have been made, save all the changes and restart the network as user 'root' as follows:

```
# /etc/init.d/network restart
Shutting down interface eth0:                          [  OK  ]
Shutting down loopback interface:                      [  OK  ]
Setting network parameters:                            [  OK  ]
Bringing up loopback interface:                        [  OK  ]
Bringing up interface eth0:                            [  OK  ]
Bringing up interface eth1:
Determining IP information for eth1... failed; no link present.  Check cable?
                                                       [FAILED]
```

In the above example, one of the NIC's was not configured or is not connected to a router.

The network configuration can be verified with the following command (again as user 'root'). The Ipv4 and the Ipv6 addresses are highlighted.

```
# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:B0:D0:E1:14:BE
          inet addr:10.44.195.84  Bcast:10.44.195.255  Mask:255.255.255.0
          inet6 addr: fe80::2b0:d0ff:fee1:14be/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3329 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1338 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:683374 (667.3 Kb)  TX bytes:421610 (411.7 Kb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5187 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5187 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3282451 (3.1 Mb)  TX bytes:3282451 (3.1 Mb)
```

4.2. SSH configuration

Secure Shell is a terminal and file transmission protocol that is very secure, as it encrypts all the transmitted data using SSL public/private key encryption. It is similar in use to Telnet and FTP, and is by far the most-preferred protocol.

Refer to the Troubleshooting Guide in the Appendix to resolve problems with SSH.

4.2.1. Starting the SSH service

To use SSH on the server, the SSH service should be running. This can be checked:

```
# /etc/init.d/sshd status
SSH service is not running:                          [  OK  ]
```

Start it if this is the case:

```
# /etc/init.d/sshd start
Starting SSH service:                                [  OK  ]
done.
```

Check that the SSH service will start on boot-up:

```
# chkconfig --list sshd
smb             0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

This shows that SSH will not automatically be started in any of the 6 run-levels. To remedy this and get it to automatically start for run-levels 3, 4 and 5, type the following and verify that the desired effect was achieved:

```
# chkconfig --add sshd
# chkconfig --level 345 sshd on
# chkconfig --list sshd
sshd            0:off   1:off   2:off   3:on    4:on    5:on    6:of
```

4.2.2. Windows Client configuration

An open-source Windows SSH-client, putty, can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty.

Run putty.exe, set the IP address or host name from the Linux server, and make sure that the SSH option is selected. Click on 'Open'
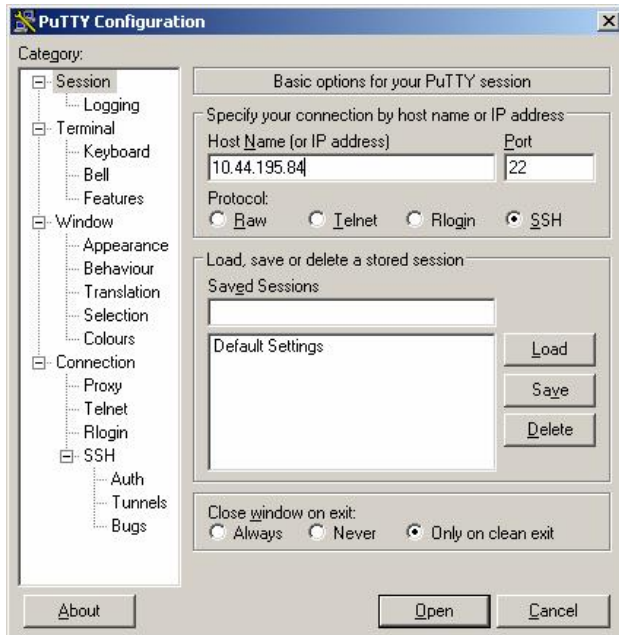
Figure 2

In a DOS-like console, you will be prompted for a log-in name and a password. Use 'root' for the log in name, as this may be the only user currently configured on the server.

4.2.3. Linux client configuration

You can also access the Linux server from another Linux machine as follows:

```
# ssh root@10.44.4.44
Password for user root:
[oracle1]#
```

This will create a command shell on the remote machine.

4.3. Samba configuration

4.3.1. Setup

Samba is the open-source implementation of the SMB network protocol. The SMB protocol was recently renamed by Microsoft to CIFS for "Common Internet File System" (which it is not). With this protocol, it is possible to access any Windows-based machine with the correct access credentials, or indeed, to participate on a Windows-based network in the role of a Windows domain server, a WINS server, an Active Directory Server, etc.

To get Linux to publish shared directories or to behave like a domain controller, the configuration file `/etc/samba/smb.conf` is edited to suit. The example shows a minimum configuration that published the `/tmp` directory to all users on the network:

```
 [global]
# Server Naming Options:
# workgroup = NT-Domain-Name or Workgroup-Name
workgroup = WORKGROUP
server string = Oracle Server on Linux
# Logging Options:
log file = /var/log/samba/log.%m
```

```
# Put a capping on the size of the log files (in Kb).
max log size = 50
# Set the log (verbosity) level (0 <= log level <= 10)
log level = 3
# Security and Domain Membership Options:
hosts allow = 10.44.195. 127.
security = user
encrypt passwords = yes
smb passwd file = /etc/samba/private/smbpasswd
# Browser Control and Networking Options:
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192 IPTOS_LOWDELAY
interfaces = eth0 eth1
# Name Resolution Options:
name resolve order = lmhosts hosts bcast
bind interfaces only = true
debug level = 1
create mask = 0644
directory mask = 0755
level2 oplocks = True
read raw = no
write cache size = 262144
[tmp]
comment = Temporary file space
path = /tmp
read only = no
public = yes
```

Save this file after having made suitable changes.

4.3.2. Starting the Samba Service

Restart the Samba service as follows:

```
# /etc/init.d/smb restart
Restarting SMB services: Shutting down SMB services:        [  OK  ]
Starting SMB services:                                      [  OK  ]
done.
```

4.3.3. Automatic starting of the service

You can check if the Samba service has been set to start when the server is booted up with the following command:

```
# chkconfig --list smb
smb             0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

This shows that Samba will not automatically be started in any of the 6 run-levels. To remedy this and get it to automatically start for run-levels 3, 4 and 5, type the following and verify that the desired effect was achieved:

```
# chkconfig --add smb
# chkconfig --level 345 smb on
# chkconfig --list smb
smb             0:off   1:off   2:off   3:on    4:on    5:on    6:of
```

4.3.4. Verifying operation from a Windows client

You should be able to see the \\<server_name>\tmp directory share from any Win32 client on the network. Under the example configuration above, a password is not required for read and -write access. This is neither a useful nor a secure configuration for Samba. The reader is referred to reference [3] for a detailed understanding of the configuration of Samba. Several GUI-based and Web-based tools exist that facilitate the configuration of Samba, e.g. SWAT, WEBMIN.

4.3.5. Accessing a Windows share from the Linux server

Conversely, it possible for a Linux server to access a Windows-share. For this to work, it is not necessary to run the Samba service, although the Samba client needs to be installed on the server. To check if Samba is installed on Red Hat Linux distribution, type the following:

```
# rpm –qa | grep samba
samba-client-3.0.3-5
system-config-samba-1.2.9-2
samba-3.0.3-5
samba-swat-3.0.3-5
samba-common-3.0.3-5
```

The highlighted line above indicates that the Samba client is installed, and as an example, a file browser on the Linux desktop can be pointed to a Windows share through the SMB protocol as shown below in Figure 3:
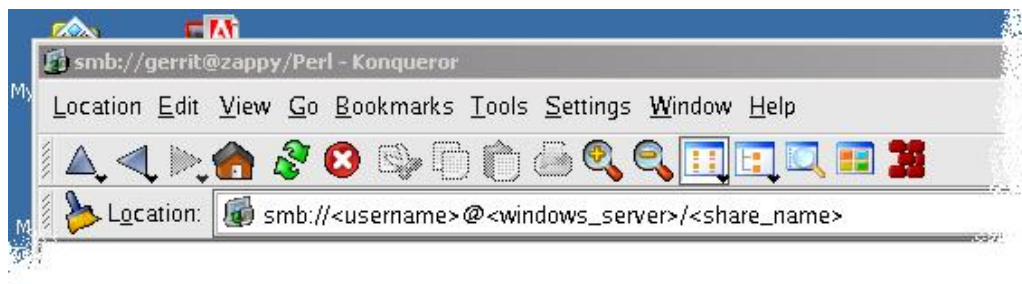

Figure 3: Using KDE's 'Konqueror' browser to access a Windows share

A password dialog will prompt for a password to the Windows share.

4.4. Telnet service

The Telnet service provides the most basic terminal protocol although it is highly insecure. SSH is nowadays used in preference to Telnet.

4.4.1. Setup

No configuration is necessary to run the Telnet service, other than ensuring that Red Hat's RPM package `telnet-server` has been installed. To check that is it installed:

```
# rpm –qa | grep telnet
# telnet-server-0.17.20
# telnet-0.17.20
```

Ensure that the Telnet service will automatically start when the Linux server is restarted, by running the following command:

```
# chkconfig --add telnet
# chkconfig --level 345 telnet on
# chkconfig --list telnet
telnet          on
```

4.5. FTP service

FTP is the most basic file-transfer protocol although it is reasonably insecure, as the network packets are not encrypted. SSH is nowadays used in preference to FTP for this reason. We will configure the VSFTP server here, which is recognised for its general robustness.

4.5.1. Setup

The FTP service requires some configuration and needs to be set to automatically start after the Linux server has been rebooted. First check that VSFTP has been installed:

```
# rpm -qa  | grep ftp
tftp-server-0.33-3
vsftpd-1.2.1-5
ncftp-3.1.7-2
gftp-2.0.17-2
lftp-2.6.12-1
ftpcopy-0.6.2-6
ftp-0.17-21
tftp-0.33-3
```

Add the FTP service for the service controller:

```
# chkconfig --add vsftpd
# chkconfig --level 345 vsftpd on
# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

By default, the public FTP directory on Red Hat Linux is `/var/ftp/pub`. This can be changed by editing the file `/etc/vsftpd/vsftpd.conf`. For information on configuring the FTP server, run this command:

```
# man vsftp
```

4.6. NFS

The Network File System is the most commonly used network-based file system between UNIX computers. It has robust user access control, but does not encrypt transmitted data by default, nor does not always recover elegantly from severed networked connections. Not unlike the SMB protocol.

Network file shares are published from any UNIX machine that runs the NFS service and can be used by any other machine that has the NFS client installed.

This includes Windows-machines: Windows Services for Unix (WSU) can be installed on Windows from http://www.microsoft.com/downloads.

4.6.1. Setup

Configure the service to run and to automatically start after boot-up:

```
# chkconfig --add nfs
# chkconfig --level 345 nfs on
# chkconfig --list nfs
nfs     0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

Start the service:

```
./nfs start
Starting NFS services:                                  [  OK  ]
Starting NFS quotas:                                    [  OK  ]
Starting NFS daemon:                                    [  OK  ]
Starting NFS mountd:                                    [  OK  ]
```

As user root, add the following NFS share to the server:

```
# echo /u02/oradata  (ro,insecure,all_squash) >> /etc/export
```

The NFS service needs to be restarted after a new share is published:

```
# /etc/init.d/nfs restart
```

This NFS share can be accessed from any Unix-based or Windows machine running WSU, as long as it can supply the access credentials. To manually add this share to another Linux machine, do the following:

```
# mkdir /remote_data
# echo "<server_name>:/u02/oradata /remote_data  nfs  defaults  0 0" >> /etc/fstab
# mount –a
# ls /remote_data
..
```

4.7. Installing Java

4.7.1. Oracle JVM

It is not necessary to perform a separate install of Java on Oracle version 10g and the following section can be skipped. This was not the case for older version of Oracle. Oracle 10g now ships with its own Java Virtual Machine (VM), which is compatible with Sun's Java 1.4.1.

Once Oracle has been installed, refer to the readme.txt file in `$ORACLE_HOME/javavm/doc` for information on using the JVM.

4.7.2. Sun JDK

If you have to install a Java SDK, the SDK of choice is Sun's JDK. Download the JDK installation with a `.bin` extension (`j2sdk-1_4_2_05-linux-i586.bin`) from the website http://java.sun.com/j2se/download.html. When the download has completed, do to the download directory (mostly the currently logged-on user's home directory), set the file to executable mode, start the installation, accept the license agreement and choose to install the JDK to the `/usr/java/j2sdk1.4.2` – directory

```
$ whoami
oracle
$ cd ~
$ chmod +x j2sdk-1_4_2_05-linux-i586.bin
$ su
Password:
# ./ j2sdk-1_4_2_05-linux-i586.bin
# . . .
done
# ln –s /usr/java/j2sdk1.4.2 /usr/local/java
```

A symbolic link was established from the JDK to the `/usr/local` directory. By referring to installations through a symbolic link, it is possible to upgrade the JDK to a more recent version without affecting the server's environment:

```
# rm –fr /usr/local/java
# ln –s /usr/java/j2sdk1.5.0 /usr/local/java
```

The setting of the environment is a one-operation, and is mostly set in each user's profile file. Add the following directory to the PATH statement in each user's `.bash_profile` file:

```
$ whoami
oracle
$ nano ~/.bash_profile
. . .
export PATH=$PATH:/usr/local/java/bin:...
```

Save and exit the editor, and make the changes effective:

```
$ source ~/.bash_profile
```

In the case of user 'oracle', this will override the Java environment supplied by Oracle.

4.8. Starting and Stopping the Server

Although this Linux distribution uses a journalled file system, it is advisable to cleanly power the server down. This can be done from the KDE Desktop GUI, or by issuing the following command from a terminal:

```
$ su
Password:
# /sbin/shutdown -h now
```

All registered services will be cleanly stopped.

To reboot the server for whatever reason, issue the following commands:

```
$ su
Password:
# /sbin/shutdown -r now
```

5. Guide to installing and X-Windows client on a Windows PC

5.1. Installation

A frequently-used X-Windows service that can be run on a Windows PC to use it an X-Windows client is X-Win32. A 30-day evaluation copy can be downloaded from www.starnet.com. Register, download and install it.

5.2. Setup an X-Window client

Run the X-Win32 Configuration utility from the Windows Start menu. You will be greeted with the following screen:
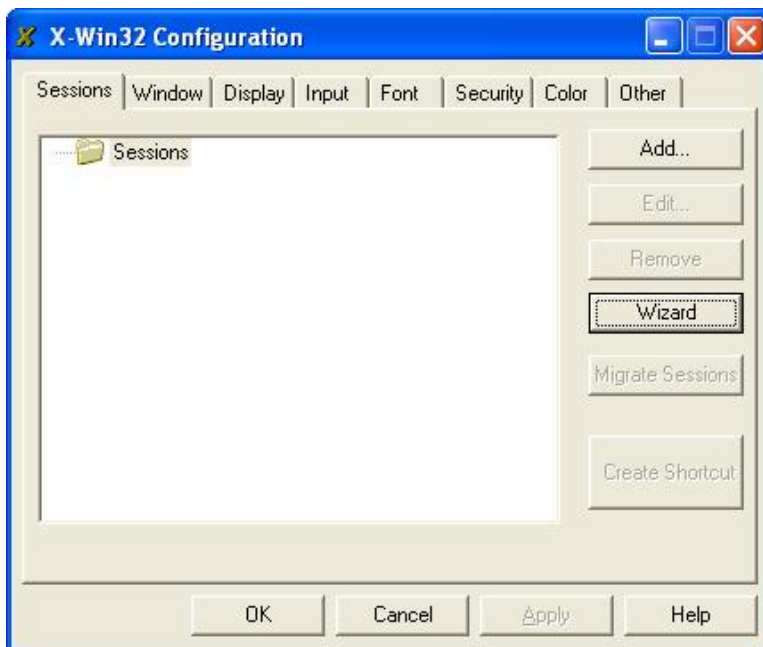
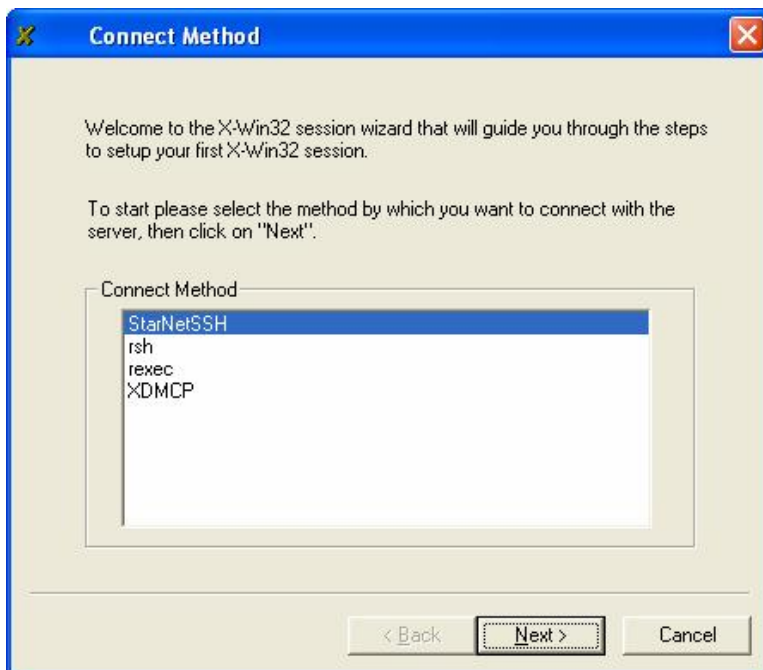**Figure 4**

Click on the 'Wizard' button:

Figure 5

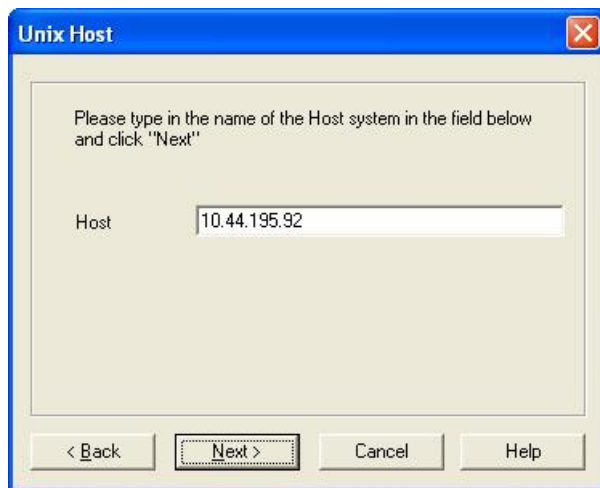Select the 'StarNetSSH' option and click 'Next'.

Figure 6

Enter the host name or the IP address of the Oracle server and click 'Next'.

Figure 7

Enter the login details for user 'root', as you would log into the actual server, and click 'Next':

Figure 8

Select Linux as the Operating System and use the suggested command that will launch an X-terminal screen on your Windows desktop. Click 'Next'.
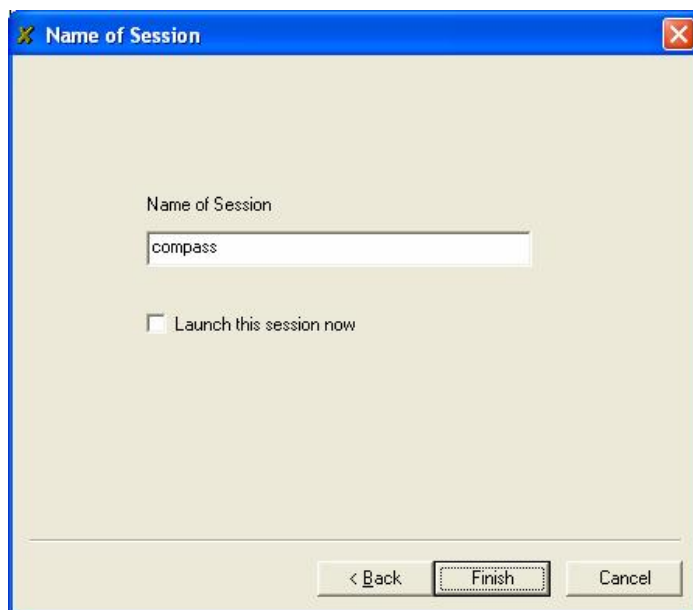

Figure 9

Enter the name by which you want to this session to be known as (usually the same as the server host name) and click on 'Finish'. This should return you to the configuration screen. Here, we called the session 'compass', which is shown in Figure 10.
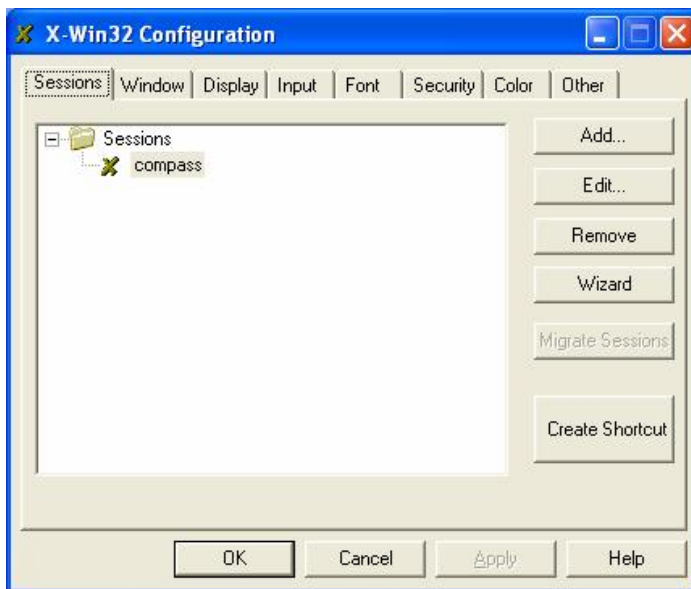
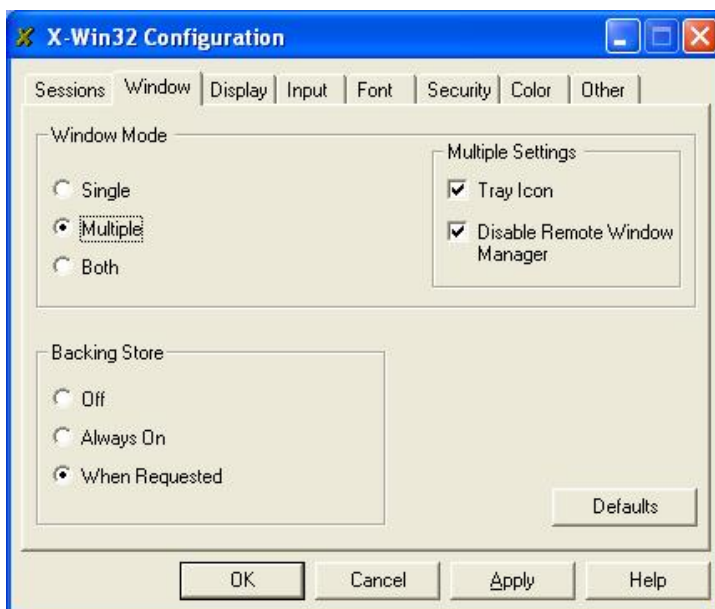**Figure 10**

Click on the 'Window' tab:

Figure 11

Ensure the 'Multiple' Window mode is selected.
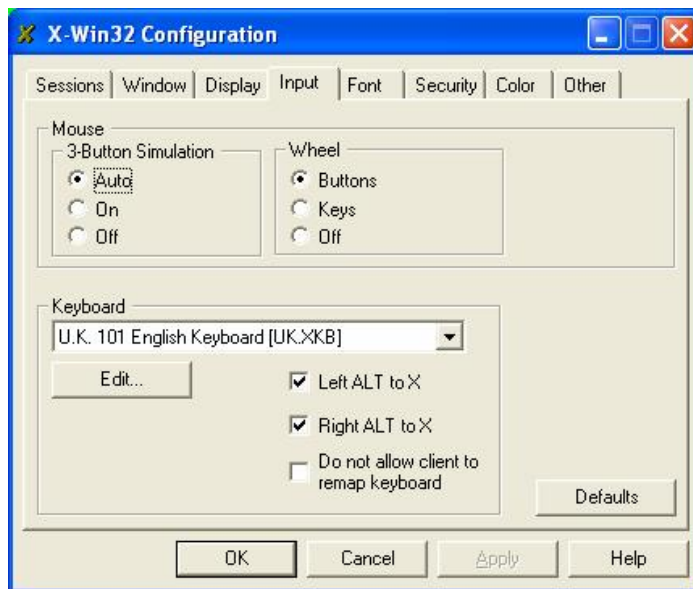
Select the 'Input' tab:

Figure 12

Ensure that the UK Keyboard is selected.

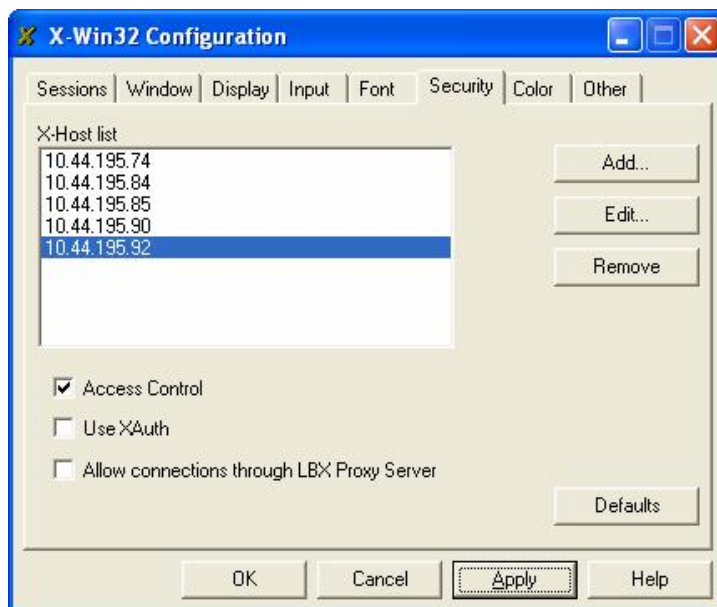Click on the 'Security' tab:

Figure 13

Add the IP address or the host name of the server to which you want to connect to, and check the "Access Control" box. Click the 'Apply' button.

Click the 'OK' button to close the configuration wizard.

5.3. Running an X-Window client

5.3.1. First time run

Start the X-Win32 service from the Windows Start button.
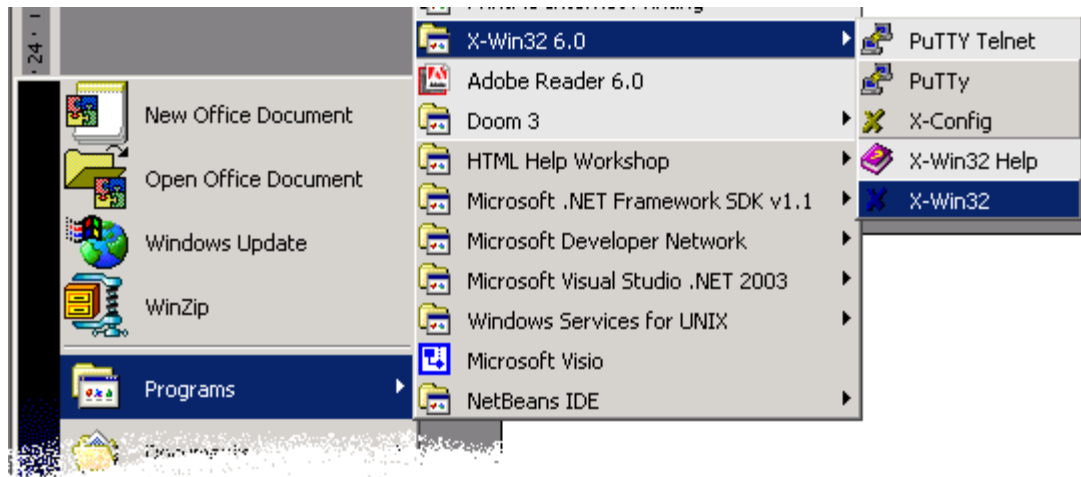
Figure 14

This will run the X-server in an icon on the Windows task bar. From the task bar, you can select the server that you want to start an X-session, as shown in Figure 15 below:
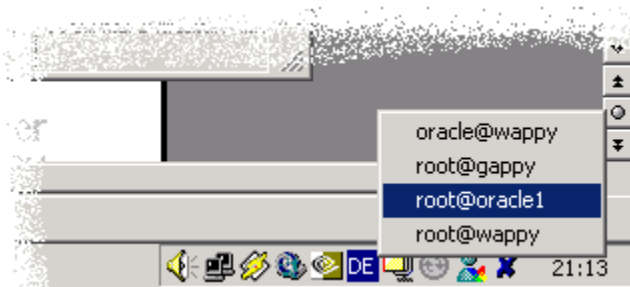

Figure 15: Launching the X-Server from the Windows Taskbar

If this is the first time that an X-session is created to a device of this IP address, a public-private key-pair will be generated, and you will be asked to store the server's public key:

Figure 16

Click 'Yes' and continue.

One of the following scenarios may unfold before you:

1.1.1.1.   Cannot connect to the X-Server

If the following message appears:
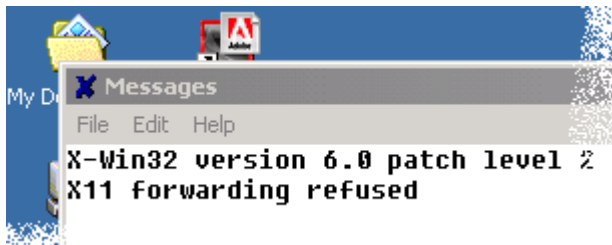
Figure 17

Edit the file `/etc/ssh/sshd_config` (as user 'root')

```
# nano /etc/ssh/sshd_config

.. and set the parameter:
X11Forwarding yes
.. and then save
```

Restart SSH on the server:

```
# /etc/init.d/sshd restart
 * Stopping sshd...                                          [ ok ]
 * Starting sshd...                                          [ ok ]
```

Reattempt to establish an X-server connection as shown in Figure 15 above.

You should also refer to the Troubleshooting guide in the Appendix and rectify the problem before continuing.

1.1.1.2. Successfully connect to the X-Server

You may get the following dialog, in which case the X-server (the program that is running on the Windows client PC) works as it should:

Figure 18

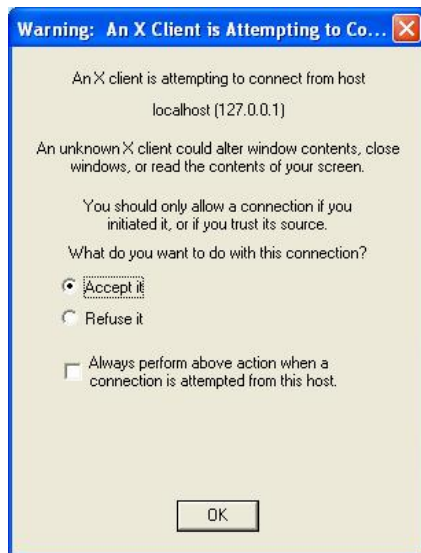Click "Accept it" and as this is a local server, it is safe to check the "Always perform…" box low. Click 'OK'.

An X-terminal will now appear, from which X-windows applications from the Linux box can be launched.

Remember that to backspace you must hit Ctrl-H. The Backspace key does the same as the Delete key. This is default terminal behaviour on UNIX terminals.
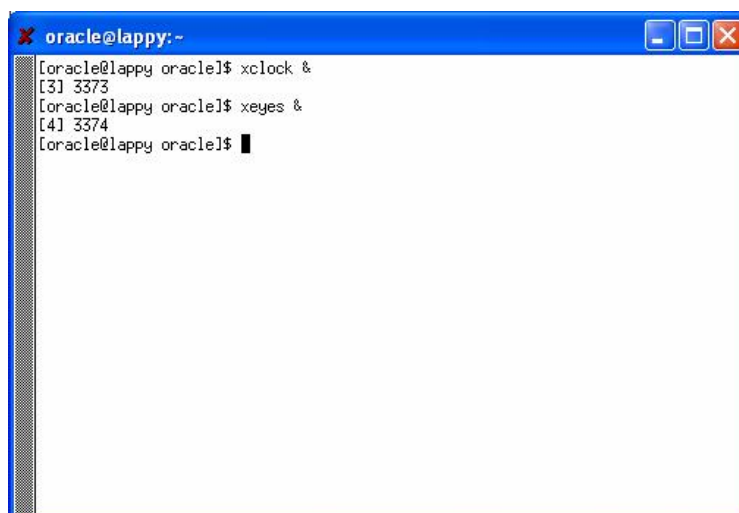
Figure 19

5.3.2. Testing X

To test this, type `xclock &` and `xeyes &` to get the following results:
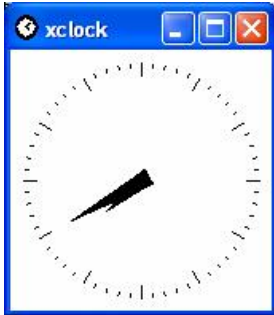
Figure 20

- And

Figure 21

5.3.3. Fixing the X-configuration

If you managed to start an X-terminal but get the following out when attempting to run an X-based program, then the X terminal still needs to be configured.

```
# xclock &
[1] 5293
Xlib: connection to "localhost:10.0" refused by server
Xlib: Authentication failed at PuTTY X11 proxy
Error: Can't open display: localhost:10.0
```

Type the following and try again:

```
# export DISPLAY=localhost:10.0
```

For further problem solving, please refer to the Troubleshooting guide in the Appendix.

5.3.4. Monitoring server behaviour

A more useful tool to monitor the servers' performance is '`ksysguard`', which is invoked in a similar way:

```
# ksysguard &
```
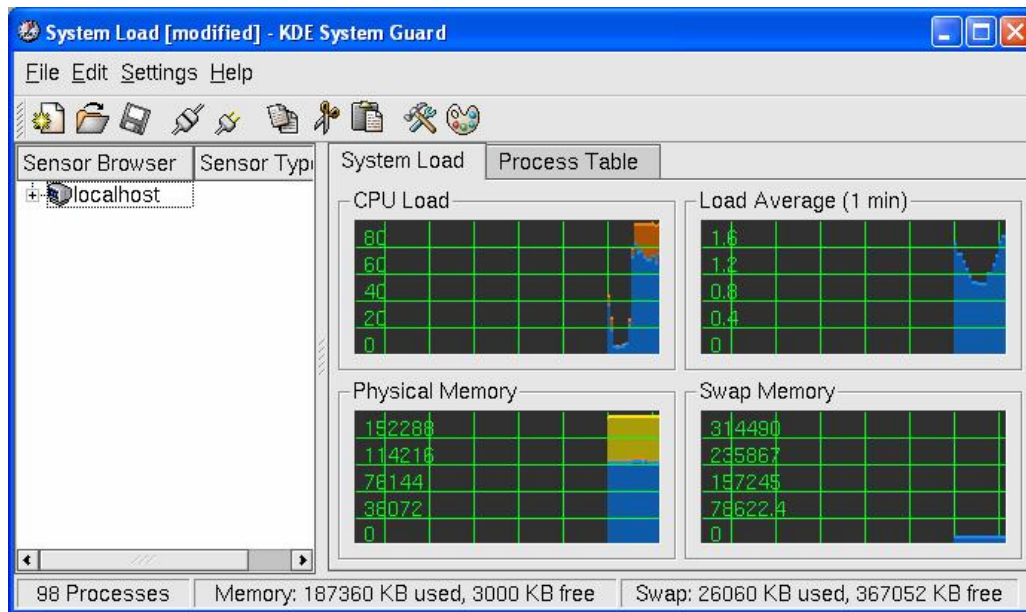
Figure 22

5.4. Hosting a Linux desktop on your Windows PC

Generally, X-Windows applications on the Linux server can be run through the Windows PC desktop with few limitations.

You can host the entire Linux KDE desktop environment on your Windows PC by running the following command:

```
# startkde
```

Note that numerous warning messages appear on the X-terminal. These can be safely ignored.
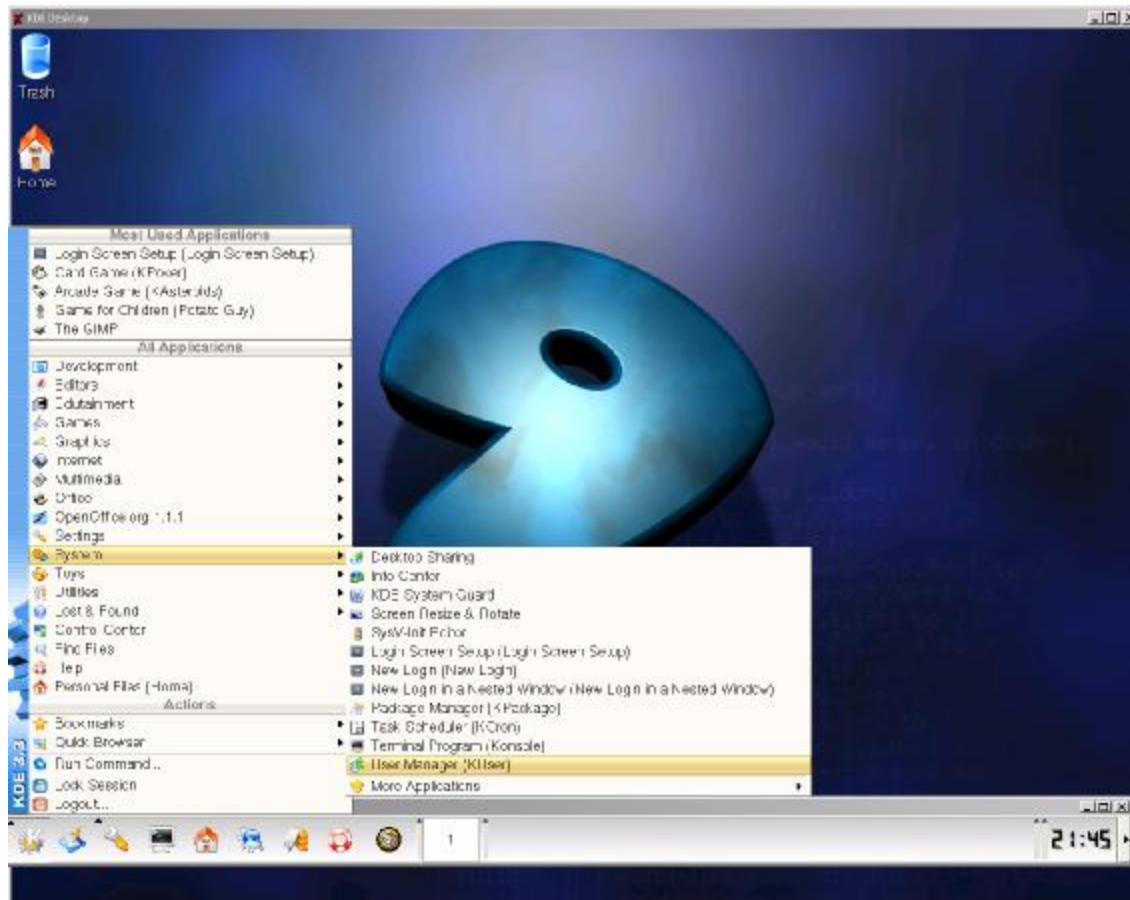
Figure 23: KDE Desktop on a Windows PC over the X-protocol

To only access the Linux desktop task-bar through X11, run kicker. This is a similar to the Windows task bar:

```
# kicker &
```


Figure 24

6. Guide to installing Oracle 10g on Red Hat Linux

This specifically describes the installation of Oracle 10g on Red Hat Enterprises Edition 3. Other versions of Oracle require different installation procedures, which are not in the scope of this document.

6.1. Configuring the Kernel

Various kernel parameters need to be set for an optimal Oracle installation and the best performance. Without these settings, Oracle 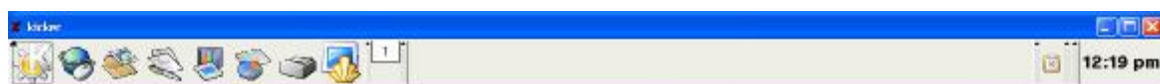10g is still likely to install without any problems and will run adequately well. On a production system, these settings should be implemented.

Edit the file `/etc/sysctl.conf` as user 'root':

```
$ su
Password:
# nano /etc/sysctl.conf
```

Add the following lines if they are not already there:

```
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
```

Save the file and update the file to the kernel for an immediate effect (no need to reboot here):

```
# sysctl –e –p /etc/sysctl.conf
```

Add the following to the `/etc/security/limits.conf` file if it does not already exist:

```
#   nano /etc/security/limits.conf

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

Add the following to `/etc/pad.d/login` if it does not already exist:

```
# nano /etc/pam.d/login
session     required     /lib/security/pam_limits.so
```

6.2. Special case: Dealing with the Fedora Core distribution

If the operating system is not the Red Hat Enterprise Application Server but a Fedora Core release, then the following will make Oracle believe that it is installing to the Enterprise edition:

```
# cp /etc/redhat-release /etc/redhat-release.bak
# echo "Red Hat Enterprise Linux AS release 3 (Taroon)" > /etc/redhat-release
```

At the end of the Oracle installation, remember to set the release name back to the original:

```
# cp /etc/redhat-release.bak /etc/redhat-release
```

6.3.

6.4. Create user 'oracle'

The Oracle RDBMS on UNIX runs under the auspices of a dedicated UNIX user that performs the role of the Oracle Administrator. Such a user needs to be created, and the user is typically called 'oracle'. This is done as follows:

```
$ su
Password:
# groupadd oinstall
# groupadd dba
# useradd oracle -g  oinstall -G dba oracle -d /home/oracle -s /bin/bash
# passwd oracle
Changing password for user oracle.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

6.5. Oracle Directory tree on the Server

The way in which disk storage is arranged depends on the application and the storage devices. For simplicity, the approach is based on the Oracle File Architecture (OFA) specification.

Another approach to data storage would be to use the Automatic Storage Management (ASM) facility that was introduced with Oracle 10g, but this is not in the scope of this implementation.

The OFA directory tree template is created as follows as user 'root' and ownership to the last node in these directories is granted to user 'oracle':

```
$ su
Password:
# mkdir -p /u01/app/oracle
# chown oracle:oinstall /u01/app/oracle
# mkdir -p /u02/oradata
# chown oracle:oinstall /u02/oradata
# exit
exit
$
```

The mount point `/u01` holds the application software, and the other mount point `/u02` holds the data.

6.6. Extending the data storage

This data storage can be extended to additional mount points /u03, /u04 etc, which can be added disk drives or off-host storage like Direct Access Data Storage (DASD) or a Storage Area Network (SAN).

As an example, the Network File Sharing (NFS) protocol could be used to a SAN.

The configuration of the SAN is not in the scope of this document, although the creation of NFS shares is discussed somewhere else. We will assume here that the SAN exports an NFS share called '`oradata`' and that the SAN has a DNS name of 'san'.

The SAN storage could permanently be mounted on the Oracle server by adding the following line to the `/etc/fstab` file:

```
$ su
Password:
```

```
# mkdir -p /u03/oradata
# chown oracle:oinstall /u03/oradata
# echo "san:oradata /u03/oradata nfs defaults 0 0" >> /etc/fstab
```

Note that the empty mount point /u03/oradata should exist on the server and that user 'oracle' should have full access to it before the NFS share is mounted, as shown above. The share is mounted with the following command:

```
# mount -a
```

The mounts specified in the /etc/fstab file will be made effective every time to server boots. To see which mounts are currently active, type the following command:

```
# mount
/dev/hda1 on /boot type ext3 (rw)
/dev/hda2 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
san:/oradata on /u03/oradata type nfs (rw,addr=san,addr=san)
```

To remove the mount, simply remove the line that was added above using an editor and then un-mount the share:

```
$ su
Password:
# nano /etc/fstab
..remove mount line and save

# umount san:/oradata
# mount
```

6.7. C-compiler

A number of gcc C-compiler versions are included in the Red Hat O/S, the default the version is 3.3. The Oracle installation requires version 2.96. This can be published to the Oracle installer without making system changes by creating an overriding symbolic link to the older gcc-compiler and putting it in user 'oracle's path:

```
$ su - oracle
Password:
$ mkdir ~/bin
$ ln -s /usr/bin/gcc296 ~/bin/gcc
```

The final step is to amend user 'oracle's path. This is described below as part of setting up user 'oracle's profile.

6.8. Updating the "oracle" profile

As user 'oracle', add or amend the following lines in the ~/.bash_profile file, using an editor such as nano: and add or amend the following lines:

```
$ su - oracle
Password:
[oracle@compass]$ nano ~/.bash_profile
Add or modify the following lines:
export TMP=/tmp
export TMPDIR=/tmp
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/<oracle_version>/db_1
export TNS_ADMIN=$ORACLE_HOME/network/admin
export ORACLE_SID=<instance_name>
```

```
export ORACLE_TERM=xterm
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export EDITOR=nano
export PATH=$HOME/bin:$PATH:/sbin:$ORACLE_HOME/bin:$ORACLE_BASE/bin:$ORACLE_HOME/bin
```

Replace the string `<oracle_version>` in `ORACLE_HOME` with the version of Oracle that will be installed, e.g. `10.1.0`. The value of `ORACLE_HOME` may need to be amended if the Oracle Server software was installed in a different directory.

Replace the string `<instance_name>` with the name of the oracle database Oracle that will be installed, e.g. `export ORACLE_SID=my_db`. If you don't know the name of the Oracle instance yet, remember to add this in later, as it is crucial for the maintenance of the database.

Note the order of items in the `PATH` statement so that the correct version of the C-compiler (version 2.96) is used during installation, as explained in the previous step.

Save the changes and make them effective with the following command:

```
$ source ~/.bash_profile
```

Remember to execute command very time changes to user `oracle`'s profile need to be activated.

The server is now ready to have Oracle 10g installed on it.

7. Installing Oracle 10g

At the end of this chapter, you should have a fully functioning Oracle 10g RDBMS system running on your Linux machine.

7.1. Obtaining Oracle Media from the Internet

The Oracle RDBMS is can be obtained as either a CPIO or a GZ file from the Internet and will be assumed to be located somewhere on the server that the user 'oracle' has access to. Go to http://otn.oracle.com to register yourself if you have not done so yet, log in and then download the 'Oracle 10g RDBMS Server for Linux'. The downloaded file is normally called `ship.db.cpio.gz`, although sometimes the `.gz` extension is omitted by ill-behaved web-browsers.

We assume here that the downloaded media has been downloaded into the `/tmp` directory and has the `.gz` extension. The file is GNU-unzipped first:

```
$ cd /tmp
$ ls ship.db.cpio.gz
ship.db.cpio.gz
$ gunzip ship.db.cpio.gz
$ ls ship.db.cpio
ship.db.cpio
```

You should now have a file called `ship.db.cpio`. Unpack the CPIO file as follows:

```
$cpio –idmv < ship.db.cpio
```

A directory `disk1` is created, which can be verified with the following:

```
$ ls –al | grep ^d | grep -i disk1
drwx------   3 oracleoinstall    504 Aug 01 13:57 disk1
```

You can write the contents of the disk1 directory to a CDROM. It fits on a single 650MB CD.

The next section describes how to install Oracle 10g from a CDROM that has been created using the above process, or one that was obtained from an Oracle vendor.

Skip this next section and go to section 7.3 to proceed with the installation.

7.2. Mounting the Oracle CDROM

This section is only relevant if you are installing Oracle from a CD. If you are installing Oracle from downloaded files, skip this section and go to section 7.3.

The Oracle RDBMS can also be obtained as an uncompressed distribution on a CD from the vendor. Either copy the content of the CD from another PC to the server to say, the `/tmp` directory using `ftp` or `scp`, or load the RDBMS Server CD into the server's CD drive and type the following commands:

```
$su
Password:
# mount /mnt/cdrom
# su - oracle
$ cp -r /mnt/cdrom/* /tmp/.
$ exit
# eject
# exit
```

7.3. Starting the installation

The installation can now be started from the `/tmp` directory as user 'oracle'

```
$ whoami
oracle
$ cd /tmp/disk1
$ ./runInstaller &
```
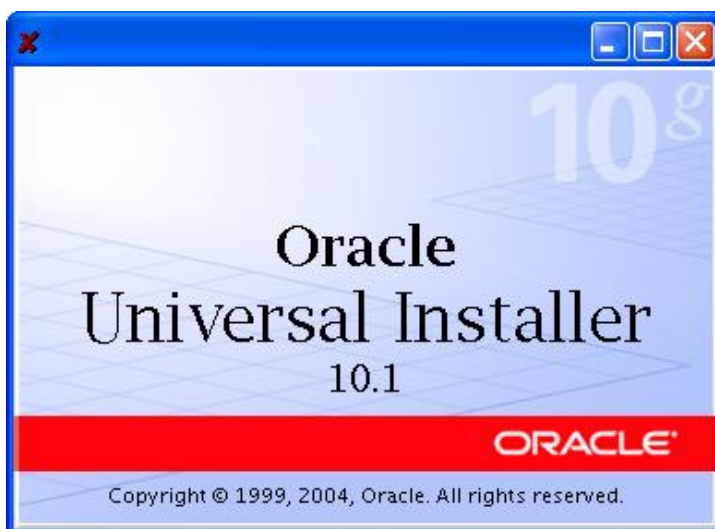
After a short while, this appears:

Figure 25

7.4. Running the installation
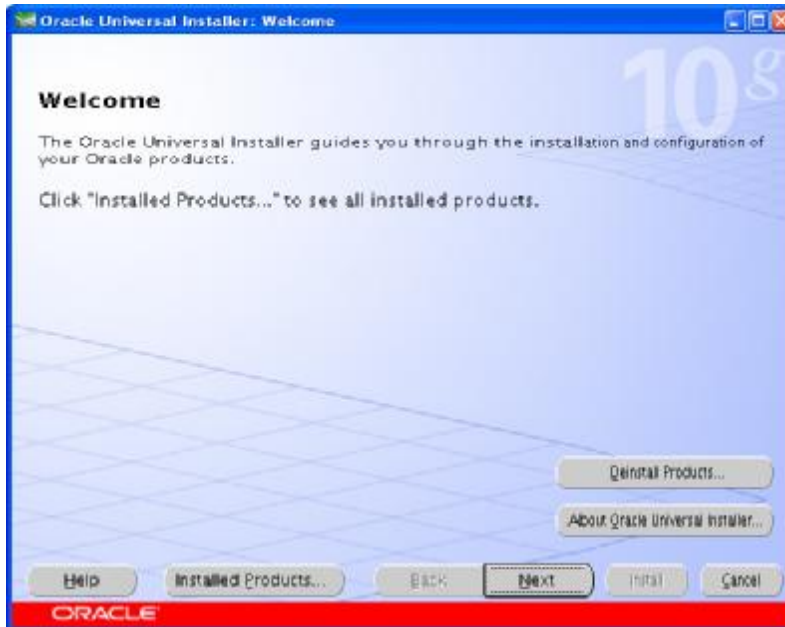
Wait until the following screen appears:

Figure 26

Click on 'Next'

Figure 27

Change the inventory directory from `/home/oracle/oraInventory` to `/u01/app/oracle/oraInventory` and click 'Next'.

Figure 28

Go back to the X-terminal from which you launched this installation, hit 'Return' once to get your prompt back and type the following:

```
$ su
Password:
# /u01/app/oracle/oraInventory/orainstRoot.sh
Creating the Oracle inventory pointer file (/etc/oraInst.loc)
Changing groupname of /u01/app/oracle/oraInventory to oinstall.
# exit
```
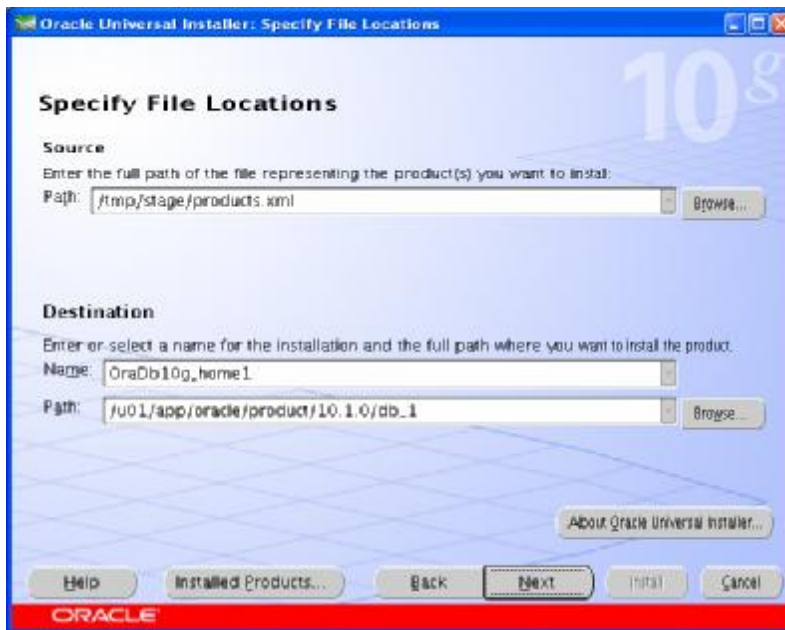
Click on 'Continue'

Figure 29

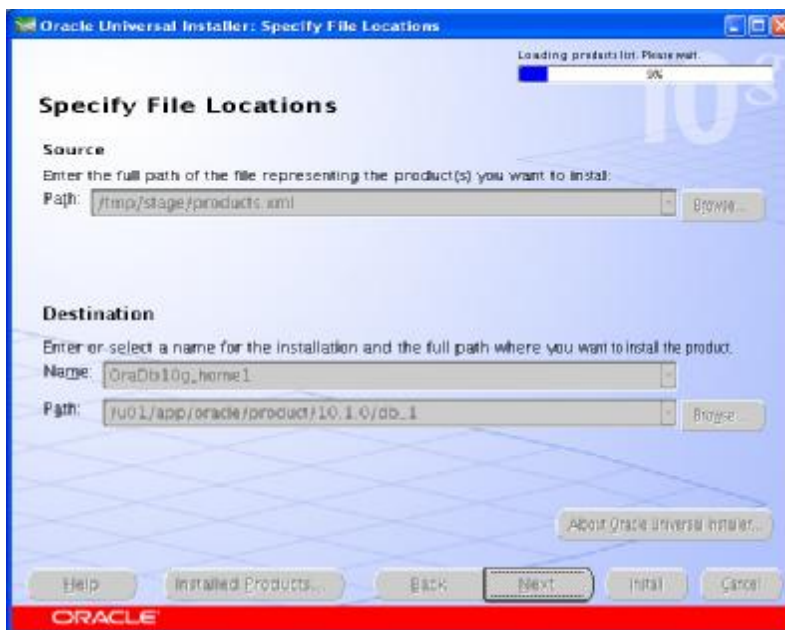Change /home/oracle/product/10.1.0/db_1 to /u01/app/oracle/product/10.1.0/db_1 and click 'Next'.

Figure 30

Wait a while for the above process to complete and the following screen to appear. Note that it can unbeknownst appear behind other desktop windows if other windows have since been brought into the foreground.

Figure 31

Select 'Enterprise Edition' and click 'Next'.
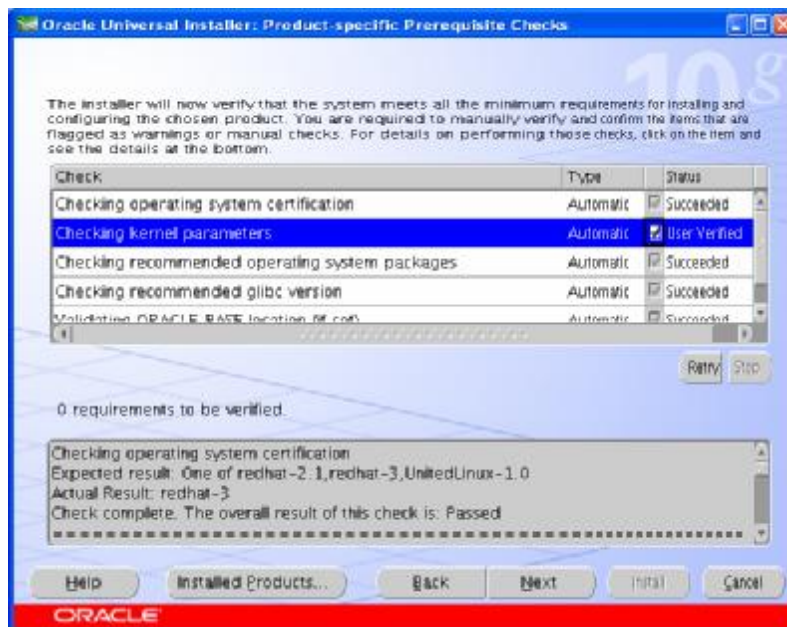
Figure 32

Allow the above checks to complete. You can override failures by checking unchecked boxes, although there should no be any failures if all the system parameters have correctly been set up. Click 'Next' and await the next screen.

Figure 33

The creation of a database is discussed later on, so select the 'Do not create a started database' option and click 'Next'.

Figure 34

This is a summary of the products that will be installed. Click Install to proceed.

Figure 35

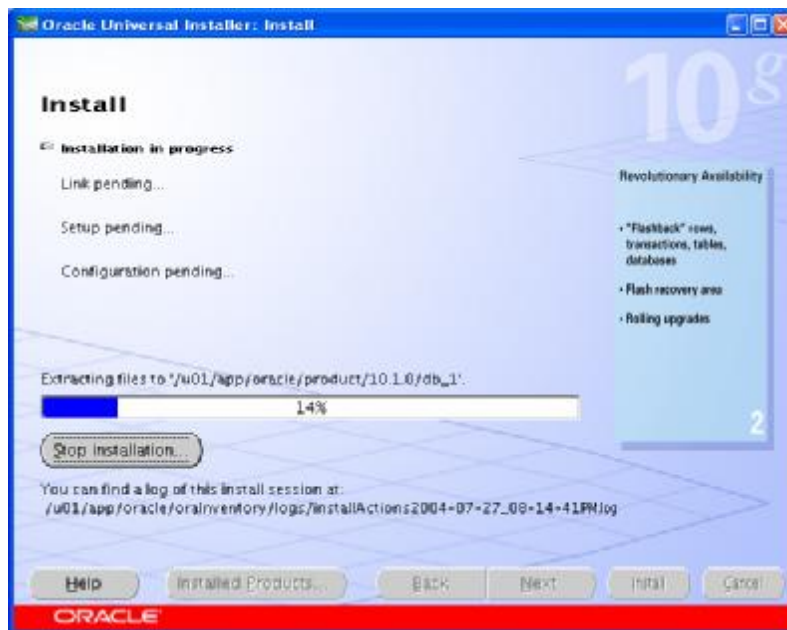If the kernel –parameters have not been correctly configured, a series of the following dialogs appear which will need to be ignored for now by clicking the 'Ignore' button:
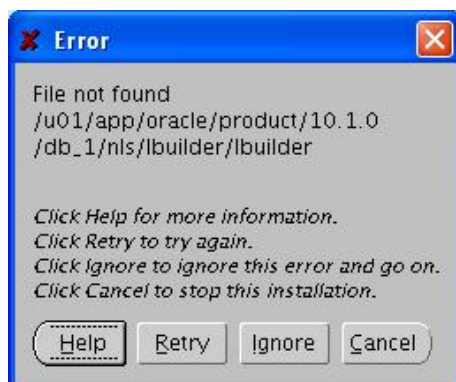

Figure 36

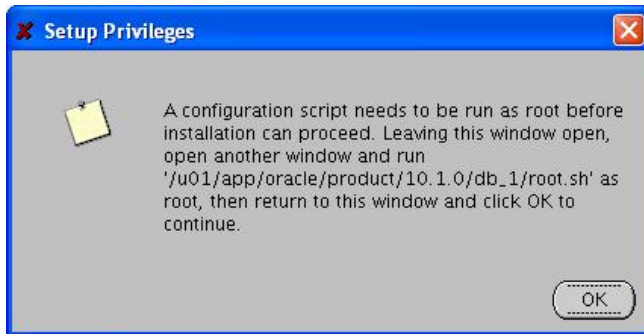Eventually, the following dialog appears:

Figure 37

Return to the X-terminal and type the commands below. Hit 'Return' to the question of where the `/usr/local/bin` directory is, and wait for the script to complete:

```
$ su
Password:
# /u01/app/oracle/product/10.1.0/db_1/root.sh
Running Oracle10 root.sh script...
The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/10.1.0/db_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...

\nCreating /etc/oratab file...
Adding entry to /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
/etc/oracle does not exist. Creating it now.
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Oracle Cluster Registry for cluster has been initialized

Adding to inittab
Checking the status of Oracle init process...
Expecting the CRS daemons to be up within 600 seconds.
CSS is active on these nodes.

CSS is active on all nodes.
Oracle CSS service is installed and running under init(1M)
# exit
```

Click 'OK' on the dialog when the script has completed running.

Figure 38

Click 'Exit'
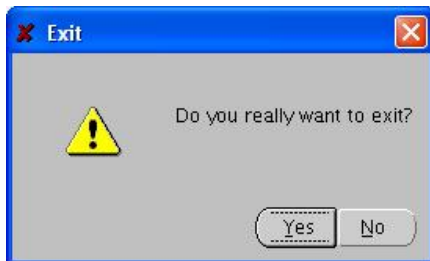
Figure 39

Click 'Yes'.

The Oracle RDMS has now been installed. It does not yet have a database for you to test the installation against. This will be performed in the next section.

8. Creating a Development Database

A development database is created using Oracle's Database Creation Assistant and creates a database with additional testing schemas and libraries. The development database differs from a production database in that many additional support tools are loaded during creation.

A production database only contains the essential tools. Production database can either be created using a template for the Database Creation Assistant, or by running a custom script.

8.1. Preparation for Red Hat Fedora Core

If you are using DBCA to create a database on Fedora Core distribution, you need to perform the following first as user 'oracle':

```
$ cd $ORACLE_HOME/bin
$ mv oracle oracle.bin
$ cat >oracle <<"EOF"
> #!/bin/bash
> export DISABLE_HUGETLBFS=1
> exec $ORACLE_HOME/bin/oracle.bin $@
> EOF
$ chmod +x oracle
```

8.2. Starting the Database Creation Assistant

From the X-terminal, run:

```
$ /u01/app/oracle/product/10.1.0/db_1/bin/dbca &
```

The following dialog appears:

Figure 40
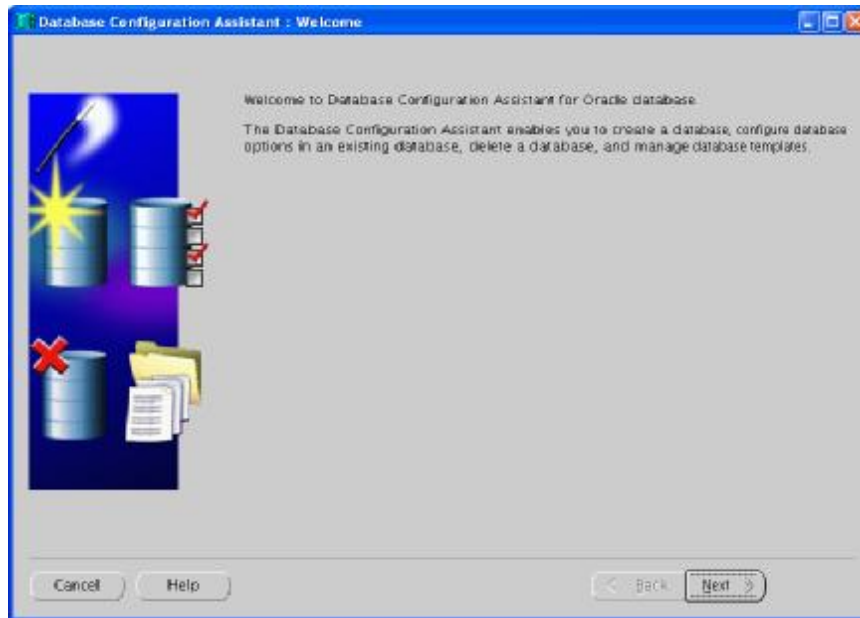
Figure 41

Click 'Next'.
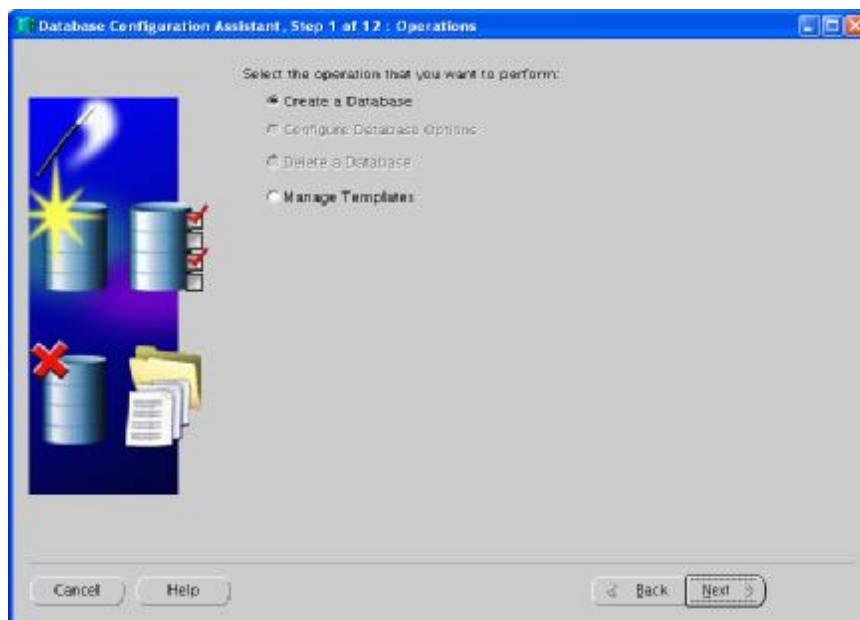
Figure 42

Select 'Create a Database' and click 'Next'.
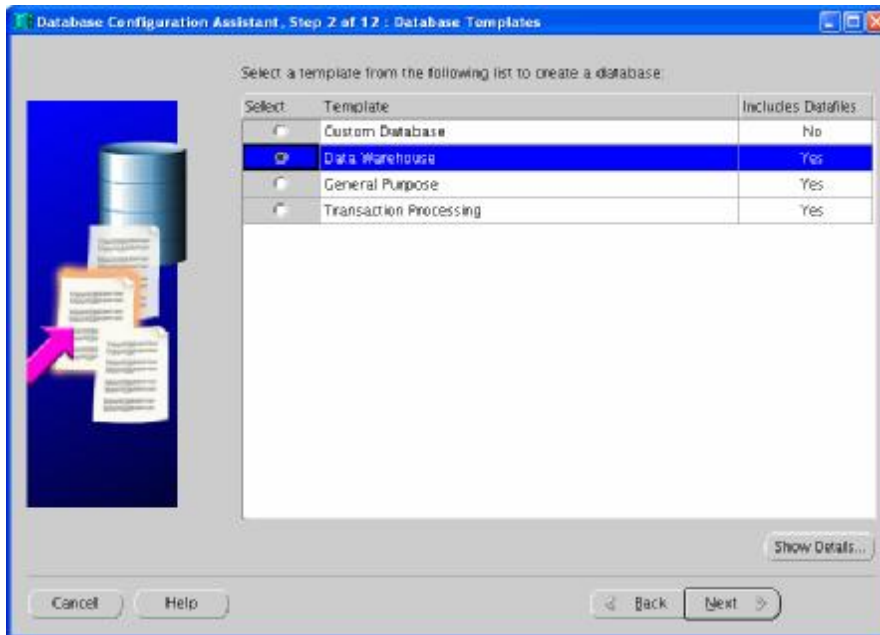
Select Data Warehouse and click 'Next'.
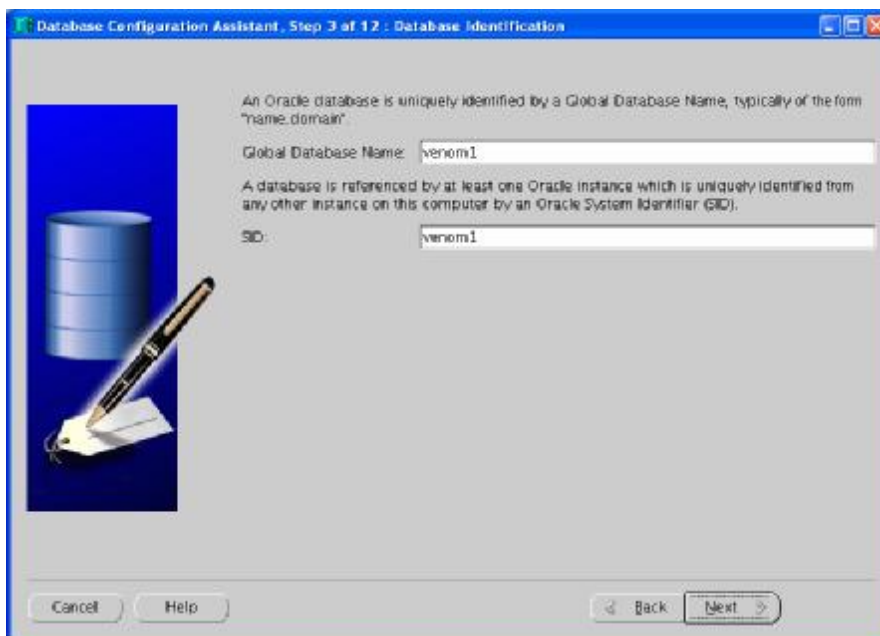
Type the name of the database in the "Global Database Name" field. The name will also appear in the "SID" field. Click Next.
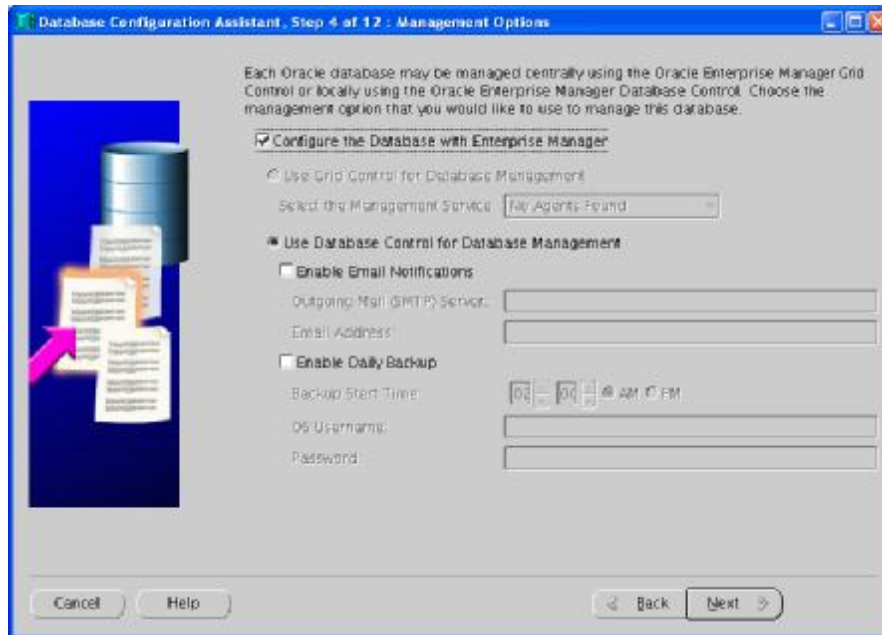
Keep the default options and click 'Next'.
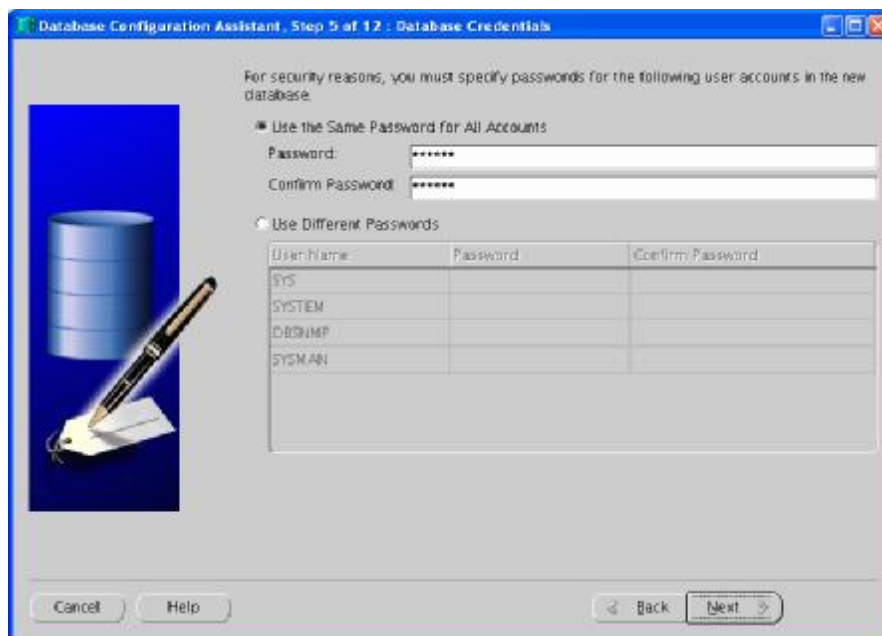
As it is a development database, select the option "Use the Same Password for All Account" and enter the password. Click 'Next'.
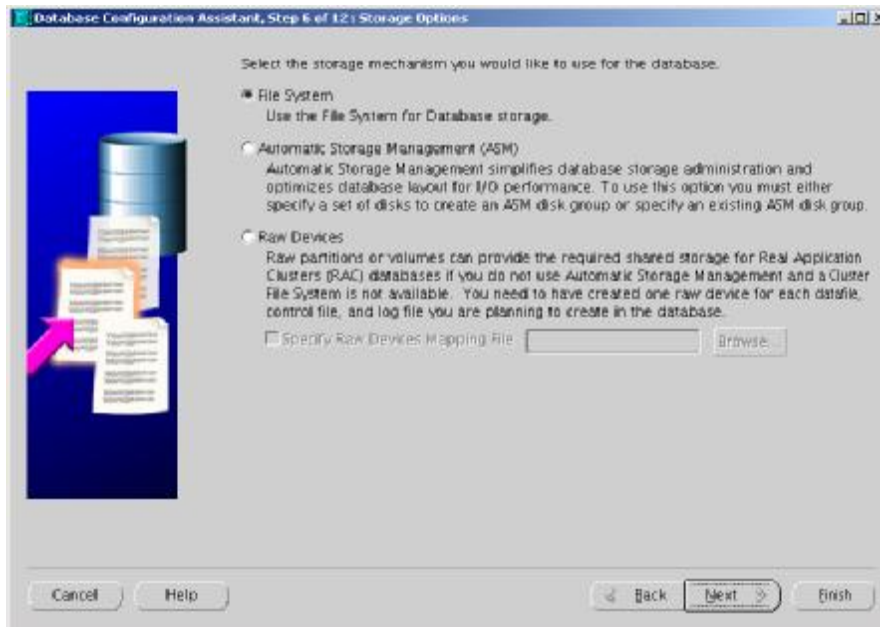
Figure 43

Select 'File System' and click on 'Next'.
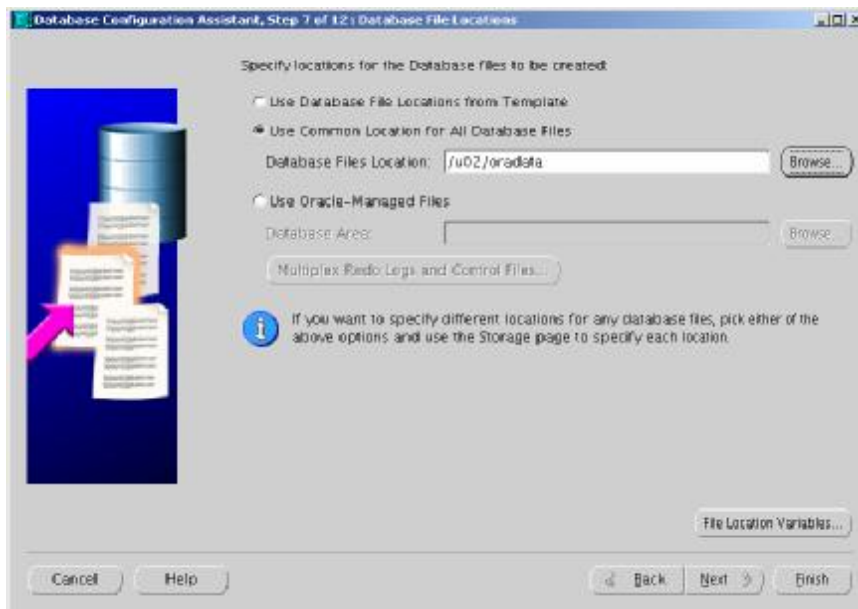
Figure 44

Change the location where the data files will be stored. Earlier on, we configured it be in the `/u01/oradata` directory. Enter this and click on 'Next'.
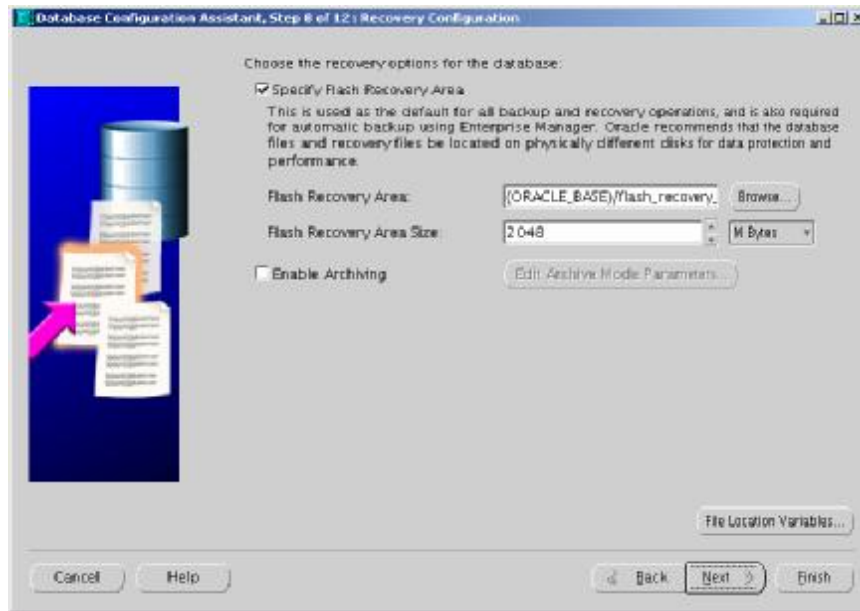
You can specify a flash recovery area. This is a feature that was introduced for Oracle 9i, and allows one to determine what the exact state of data was at specific point in the past. Click on 'Next'

Figure 45

For now, we choose not to add sample schemas to the database. Click on 'Finish'
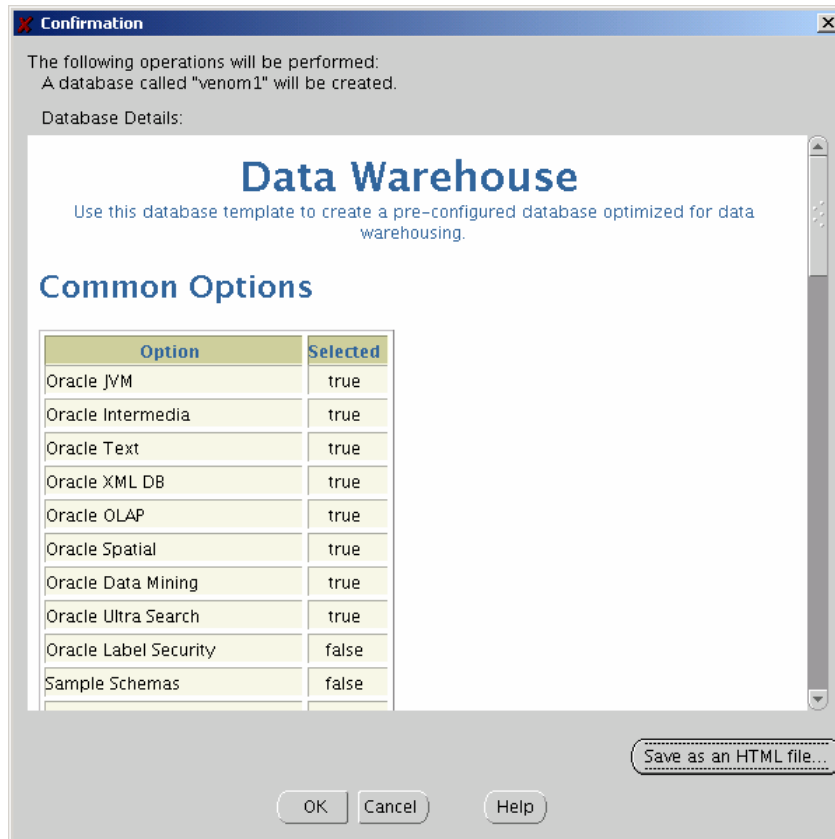
Figure 46

This shows the database configuration prior to creation. Click on 'OK'
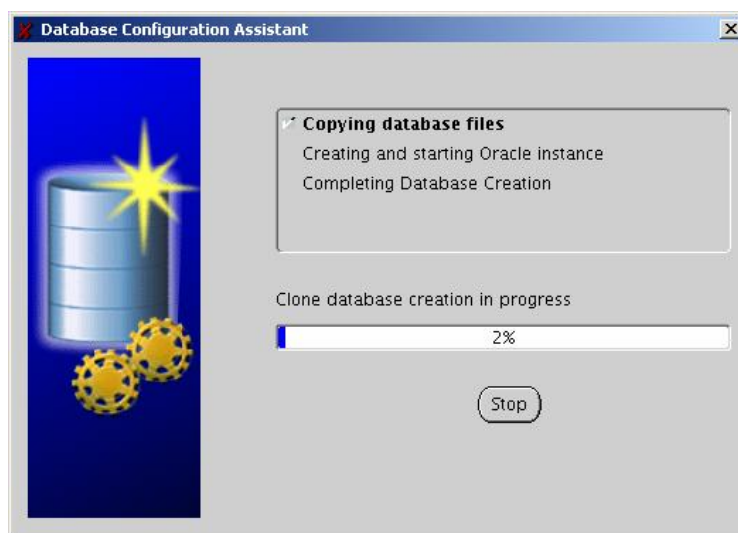
Figure 47

The database is being created.

8.3. Possible errors

If you get either of the following error dialogs then one of the following are the most likely causes:

- The environment variable ORACLE_HOME has not been set - this is probably because it was not set in user 'oracle's `.bash_profile` file.

- You are attempting to install Oracle on the Fedora release without having the step described in paragraph 8.1above.
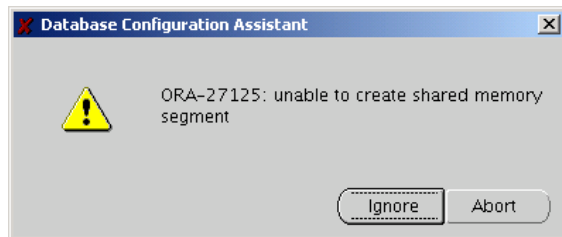
Or
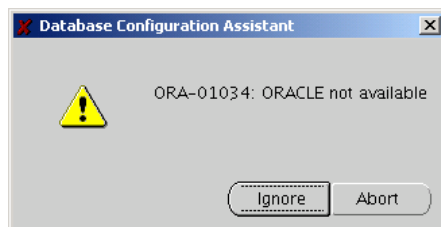
8.4. Completion
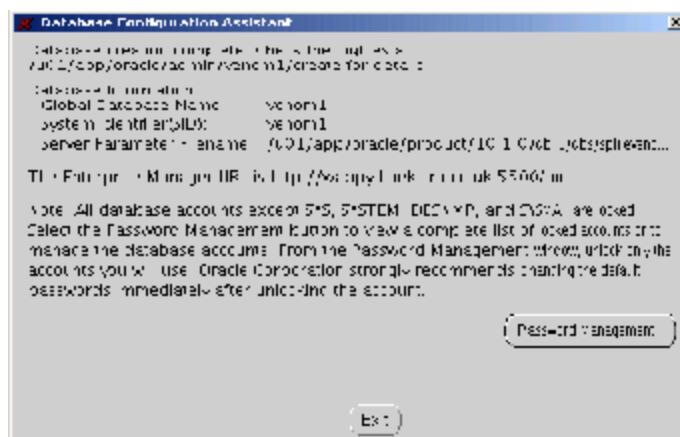
Figure 48

Click on Password Management and unlock the account SCOTT (shown below) if you intend to run any examples and exercises that Oracle provides.
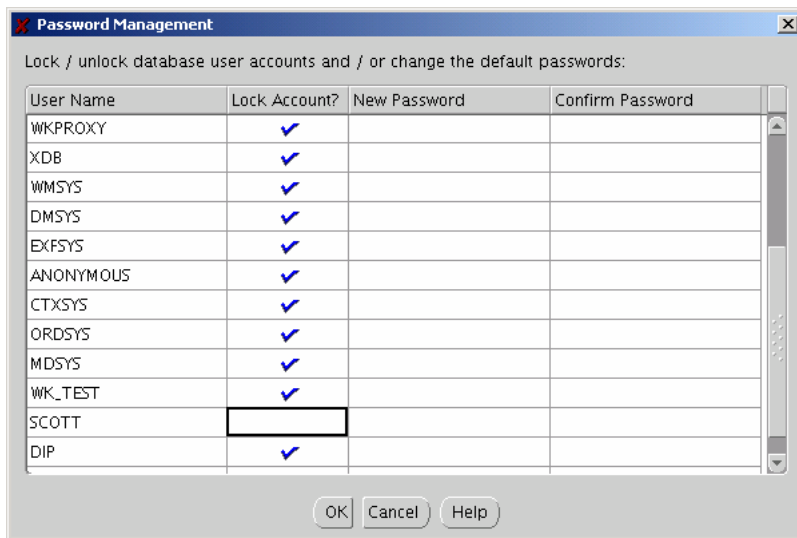
Figure 49
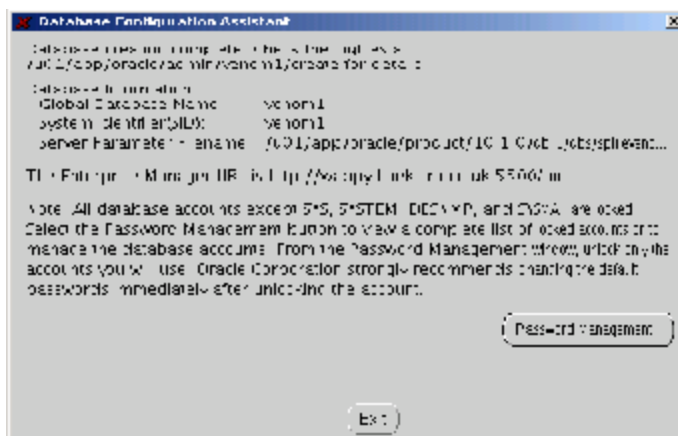
Figure 50

Click on 'Exit'

The database has now been created.

8.5. Verifying operation of the database

8.5.1. Local log in

Log in to the created Oracle database from the command line as follows:

```
$ sqlplus / as sysdba

SQL*Plus: Release 10.1.0.2.0 - Production on Tue Aug 31 00:17:19 2004

Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
```

```
SQL>
```

Shut the database down and start it up again:

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area  180355072 bytes
Fixed Size                   777996 bytes
Variable Size             162537716 bytes
Database Buffers           16777216 bytes
Redo Buffers                 262144 bytes
Database mounted.
Database opened.
SQL>
```

8.6.   Oracle Listener configuration

To log in from a remote client on the network, the Oracle Listener service needs to be started in the server, and the Oracle client software needs to be installed on the remote client.

The Listener is started as follows:

```
$ lsnrctl start

LSNRCTL for Linux: Version 10.1.0.2.0 - Production on 31-AUG-2004 00:23:24

Copyright (c) 1991, 2004, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/10.1.0/db_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 10.1.0.2.0 - Production
Log messages written to /u01/app/oracle/product/10.1.0/db_1/network/log/listener.log
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=wappy.hoekstra.co.uk)(PORT=1521)))

Connecting to (ADDRESS=(PROTOCOL=tcp)(PORT=1521))
STATUS of the LISTENER
-----------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 10.1.0.2.0 - Production
Start Date                31-AUG-2004 00:23:24
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Log File         /u01/app/oracle/product/10.1.0/db_1/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=wappy.hoekstra.co.uk)(PORT=1521)))
The listener supports no services
The command completed successfully
```

8.7.   Remote log in from a Windows Client

The installation of the Oracle client software on Windows is straightforward. Choose to install it in the `C:\Oracle` directory. Once this is installed, add the following lines to the `tnsnames.ora` file, which should be in the `C:\Oracle\network\admin` directory:

```
my_db =
  (DESCRIPTION =
```

```
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.44.4.44)(PORT = 1521))
    (CONNECT_DATA = (SID = my_db)))
```

Save the `tnsnames.ora` file and check if the following command from the DOS command line works:

```
C:\>tnsping my_db

TNS Ping Utility for 32-bit Windows: Version X.X.X.X.X

(c) Copyright XXXX Oracle Corporation.  All rights reserved.

Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=10.44.4.44)(PORT=1521))
OK (170 msec)
```

This should show a reasonable response time. If the Oracle instance is not found, then either the Listener on the Oracle server is not running or the `tnsnames.ora` configuration file has errors in it.

If the `tnsping` utility is not found, set the directory C:\Oracle\bin on the Windows system path and reattempt this in a new DOS console.

For any further troubleshooting, refer to the Troubleshooting Guide in the Appendix.

8.8. Automatic start-up of the Listener and the Oracle database

The following start-up script, `/etc/init.d/oracle`, controls the Oracle database and the Listener service, and is used to start these services automatically on boot-up. This script will only work if the environment variable, ORACLE_SID, has been set up in user 'oracle's profile, e.g. the following line should be in `~/.bash_profile`:

```
export ORACLE_SID=my_db
```

The content of `/etc/init.d/oracle` is listed below, and is owned by user 'root' (i.e. log in as 'root' before creating this file):

```
#!/bin/bash
#
# oracle   Control Oracle server and associated services
#
# chkconfig: 345 90 10
# description: Oracle database instance
# probe: true

# Source function library
. /etc/init.d/functions

# Source network configuration
if [ ! -f /etc/sysconfig/network ] ; then
  exit 0
fi
. /etc/sysconfig/network

# Starts the database and listeners
function start(){
su -l oracle -c "lsnrctl start listener > /dev/null" > /dev/null
su -l oracle -c "sqlplus /nolog > /dev/null <<!
connect / as sysdba
startup
quit
!
" > /dev/null
}

# Shows status of database and listeners
```

```
function status(){
su -l oracle -c "sqlplus /nolog > /tmp/oraclestatus <<!
connect / as sysdba
!" > /dev/null
v_connected=$(cat /tmp/oraclestatus | grep 'Connected' | wc -l)
if [ $v_connected -ne 0 ]; then
  echo "Instance is running"
else
  echo "Instance is not running"
fi
su -l oracle -c "lsnrctl status listener > /tmp/oraclestatus" > /dev/null
v_not_connected=$(cat /tmp/oraclestatus | grep 'Connection refused' | wc -l)
if [ $v_not_connected -ne 0 ]; then
  echo "Listener is not running"
else
  echo "Listener is running"
fi
rm -f /tmp/oraclestatus
}

# Shuts down database and listeners
function stop(){
su -l oracle -c "sqlplus /nolog > /dev/null <<!
connect internal
shutdown immediate
!
" > /dev/null
su -l oracle -c "lsnrctl stop > /dev/null" > /dev/null
}

# Hard shutdown of all services
function abort(){
su -l oracle -c "sqlplus /nolog > /dev/null <<!
connect internal
shutdown abort
!
" > /dev/null
su -l oracle -c "lsnrctl stop > /dev/null" > /dev/null
}

case "$1" in
  start)
    echo $"Starting Oracle instance: "
    start
    ;;
  status)
    echo $"Checking status of Oracle instance: "
    status
    ;;
  stop)
    echo $"Stopping Oracle instance: "
    stop
    ;;
  abort)
    echo $"Aborting Oracle instance: "
    abort
    ;;
  restart)
    echo $"Restarting Oracle instance:"
    stop
    start
    ;;
  *)
```

```
        echo $"Usage: $0 {start|stop|restart|abort|status}"
        exit 1
        ;;
esac
exit 0
```

Set the file to executable more:

```
# chmod +x /etc/init.d/oracle
```

The Oracle instance and listener can be started, stopped and queried by user 'root' in the following way:

```
# /etc/init.d/oracle start
# /etc/init.d/oracle stop
# /etc/init.d/oracle restart
# /etc/init.d/oracle status
```

To get the Oracle instance and the listener to automatically start up on server boot up, do the following:

```
# chkconfig --add oracle
# chkconfig --level 345 oracle on
# chkconfig --list oracle
oracle              0:off   1:off   2:off   3:on    4:on    5:on    6:of
```

8.9.   Installing the Oracle DBI driver for Perl

To run Perl against the Oracle database, the Perl DBI driver needs to be installed. This installation can only be performed by user 'root' from the Common Perl Archive Network (CPAN), which is why the environment variable, ORACLE_HOME, need to be exported first. Also, the Oracle database needs to be running before proceeding.

```
# su -l oracle -c "sqlplus /nolog > /dev/null <<!
connect / as sysdba
startup
!
"
# export ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
# perl -MCPAN -e shell
```

The first time that this is run, a number of initial configuration options need to be set. Set the '.cpan' directory to '/usr/lib/perl5/.cpan', and accept the default values by hitting the Return key, especially where you do not know the answers for. Attempt to answer the obvious questions, such as "In which continent are you?" etc...

Once you have completed the setup, install the most recent collection of Perl modules:

```
cpan> install Bundle::CPAN
```

Accept all the defaults. This may take a few minutes. Reload the CPAN environment:

```
cpan> reload CPAN
```

Now Install DBI and the Oracle drivers

```
cpan> install DBI
cpan> install DBD::Oracle
```

Installing the DBD::Oracle module can be troublesome.

CPAN can also recommend which Perl modules should be updated on the server. By typing 'r' will display a list of modules that you might want to update if you feel that they will be used. To update them, follow the process described above from `Bundle::CPAN`:

```
cpan> r
Storable                  2.09      2.13  A/AM/AMS/Storable-2.13.tar.gz
Term::ANSIColor           1.07      1.08  R/RR/RRA/ANSIColor-1.08.tar.gz
Term::Cap                 1.08      1.09  J/JS/JSTOWE/Term-Cap-1.09.tar.gz
Test                      1.24      1.25  S/SB/SBURKE/Test-1.25.tar.gz
Test::Harness             2.40      2.42  P/PE/PETDANCE/Test-Harness-2.42.tar.gz
Text::Kakasi              1.05      2.04  D/DA/DANKOGAI/Text-Kakasi-2.04.tar.gz
Text::Soundex             1.01      3.02  M/MA/MARKM/Text-Soundex-3.02.tar.gz
Time::HiRes               1.55      1.61  J/JH/JHI/Time-HiRes-1.61.tar.gz
Time::Local               1.07      1.10  D/DR/DROLSKY/Time-Local-1.10.tar.gz
URI                       1.30      1.31  G/GA/GAAS/URI-1.31.tar.gz
Unicode::Collate          0.33      0.40  S/SA/SADAHIRO/Unicode-Collate-0.40.tar.gz
Unicode::Normalize        0.28      0.30  S/SA/SADAHIRO/Unicode-Normalize-0.30.tar.gz
XML::Checker              0.09      0.13  T/TJ/TJMATHER/XML-Checker-0.13.tar.gz
XML::DOM                  1.27      1.43  T/TJ/TJMATHER/XML-DOM-1.43.tar.gz
XML::Filter::DetectWS     undef     0.01  T/TJ/TJMATHER/XML-Filter-DetectWS-0.01.tar.gz
XML::Filter::Reindent     undef     0.03  T/TJ/TJMATHER/XML-Filter-Reindent-0.03.tar.gz
XML::Filter::SAXT         undef     0.01  T/TJ/TJMATHER/XML-Filter-SAXT-0.01.tar.gz
XML::Handler::Composer    undef     0.01  T/TJ/TJMATHER/XML-Handler-Composer-0.01.tar.gz
XML::Handler::PrintEvents undef     0.01  T/TJ/TJMATHER/XML-Handler-PrintEvents-0.01.tar.gz
XML::LibXML               1.56      1.58  P/PH/PHISH/XML-LibXML-1.58.tar.gz
XML::RegExp               undef     0.02  T/TJ/TJMATHER/XML-RegExp-0.03.tar.gz
XML::Twig                 3.13      3.15  M/MI/MIROD/XML-Twig-3.15.tar.gz
XML::UM                   undef     0.01  T/TJ/TJMATHER/XML-UM-0.01.tar.gz
XML::XQL                  0.63      0.68  T/TJ/TJMATHER/XML-XQL-0.68.tar.gz
bigint                    0.04      0.05  T/TE/TELS/math/bignum-0.15.tar.gz
if                        0.03    0.0401  I/IL/ILYAZ/modules/if-0.0401.tar.gz
4 installed modules have a version number of 0
228 installed modules have no parseable version number
```

You can leave the CPAN-shell by typing:

```
cpan> quit
```

Appendix

1. Troubleshooting Guide

1. Oracle service control script does not work

If you have elected during the installation to install the Oracle service control script 'oracle' in `/etc/init.d/,` then you should be able to start and stop all the Oracle instances on the server using this script:

```
$ su
Password:
$ /etc/init.d/oracle stop
$ /etc/init.d/oracle start
$ /etc/init.d/oracle status
$ exit
```

This control script does not only control the Oracle instance(s) on the server, but also controls the Oracle listener service.

If there are multiple Oracle instances on the server that need to be controlled by this script, their names (a.k.a. ORACLE_SID's) can be defined in an environment variable ORACLE_SIDS as a space-delimited list. This is best set up in user 'oracle''s .bash_profile file (assuming that user 'oracle''s preferred shell is /bin/bash):

```
$ nano ~oracle/.bash_profile
```

Add the following line and save the file:

```
export ORACLE_SIDS="<instance_name1> <instance_name2> etc.."
```

All instances will now simultaneously be started and stopped through this service control script. If you only want to control one instance at a time, specify it as a second parameter, e.g.

```
$ /etc/init.d/oracle start <instance_name2>
```

The Oracle control script can be set up to be run when the server boots up and to cleanly shut down the Oracle database when the server shuts down. This is done as follows:

```
$ su
Password:
$ chkconfig --list 345 --add /etc/init.d/oracle
$ exit
```

Various X-based utilities exist to view and control all the system services from a graphical front-end, such as Red Hat Linux's '`serviceconf`'.

2. Cannot start up the Oracle database

If the Oracle service control script is not able to start the selected database instance up, there may be a number of reasons for this, and to fix this would mostly require the assistance of an Oracle DBA. Some basic remedies are described below.

Ensure that the instance name set in the environment variable ORACLE_SID is correct. If not, correct it in the file `~oracle/.bash_profile` – you should have a line similar to the following:

```
export ORACLE_SID=<instance_name>
```

Check that you are can perform the following on the server logged in as user 'oracle':

```
$ export ORACLE_SID=<instance_name>
$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup
```

If on starting up you get the following error:

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

      or

```
ORA-24324: service handle not initialized
ORA-01041: internal error. hostdef extension doesn't exist
cause: The cause of this problem is still undetermined.
```

- then the database's binary server parameter file (SPFILE) may have been corrupted. It can only be fixed by regenerating it.

Verify that the text-version of the database's parameter file (PFILE) is intact. This file usually resides in `/u01/app/oracle/admin/<instance_name>/pfile/init<instance_name>.ora`.

Start the database up using this PFILE:

```
$ sqlplus / as sysdba
SQL> startup pfile=/u01/app/oracle/admin/.../init<instance_name>.ora.
```

If the instance does not start up, then the content of the PFILE file is incorrect.

Regenerate the SPFILE:

```
SQL> create spfile from pfile;
File created.
```

Restart the database, this time without having to specify the PFILE:

```
SQL> shutdown immediate
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
```

3. Oracle system commands do not work

System commands that begin with `'alter system...'` and can normally only be executed by the Oracle system user, 'sys'. Either log on as user 'sys' or grant alter system commands to the user that you have in mind.

Note that if a system command produces the error:

```
ORA-27046: file size is not a multiple of logical block size
```

- then the SPFILE may be corrupted. Refer to the previous item for a remedy.

4. Cannot connect to remote database

Although it is possible to connect to the database *on* the server through a given schema, when attempting to connect from a remote client an error is displayed. In all cases, do the following:

- Check that the Oracle client software has been installed and that a path exists to the bin-directory in which sqlplus.exe and tnsping.exe reside.

- The `tnsnames.ora` file in the Oracle client installation's network/admin- directory should contain valid references the database and the server that it runs on.

- If there are multiple Oracle client installations on the client with that own `tnsnames.ora` files, remove any doubt for the time being by keeping them identical.

- Check that the Oracle server is running by pinging it:

```
C:\> ping <server_name>
```

- which should return:

```
Pinging <server_name>.<domain_name> <IP address> with 32 bytes of data:
Reply from <IP address>: bytes=32 time<1ms TTL=64
etc..
```

> If you cannot ping the server, then check that the server is running and there are no network issues that would prevent you from connecting to the server.

- You should be able to connect to the server over SSH (see section [11]), telnet, or X-Windows (see section [12]). Make sure you can establish one of these connections before proceeding.

- Check that the listener service on the server is running by typing the following on the server:

```
$ lsntctl status
```

- which returns the following if the listener is running:

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
STATUS of the LISTENER
  Alias                     LISTENER
  Version                   TNSLSNR for Linux: Version 10.1.0.2.0 - Production
  Start Date                28-JUL-2006 21:06:50
  Uptime                    0 days 0 hr. 0 min. 9 sec
  Trace Level               off
  Security                  ON: Local OS Authentication
  SNMP                      OFF
  Listener Parameter File   /opt/oracle/product/10g/network/admin/listener.ora
  Listener Log File         /opt/oracle/product/10g/network/log/listener.log
  Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))
  Services Summary...
  Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
```

- If the listener service on the database server is not running, start the listener up as follows:

```
$ lsnrctl start
```

- and verify that it is running:

```
$ lsntctl status
```

- Verify that there are no firewall or network problems by running the following from the client machine or a machine on the same network segment that as 'nmap' installed (www.insecure.org)

```
$ nmap -p 1521 <server_name>
```

If the listener is from the client network, you should get the following:

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-08-03 10:01
Interesting ports on <server_name>.<domain_name> (<IP address>):
PORT     STATE SERVICE
1521/tcp open  oracle
```

If the following is displayed:

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-08-03 10:03 BST
Interesting ports on <server_name>.<domain_name> (<IP address>):
PORT     STATE  SERVICE
1521/tcp closed oracle
```

- then, even though the Oracle listener service is running on the server, a firewall is blocking the port 1521 (the standard Oracle listener port). Refer to a network administrator to unblock port 1521 (ask for a '*Firewall policy change*').

```
ORA-12514 TNS listener could not resolve service_name in connect descriptor.
```

- From the remote client, tnsping the database:

```
C:\> tnsping <instance_name>
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTO
COL = TCP)(HOST = oracle1)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = my_db1)))
OK (10 msec)
```

This indicates that the listener on the database server is running. If the following response is received:

```
TNS-03505: Failed to resolve name
```

- then the instance name does not exist in the `tnsnames.ora` file as described above. Correct the file and retest this. If the following response is received:

```
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = oracle1)(PORT = 15:
(CONNECT_DATA = (SERVICE_NAME = my_db)))
TNS-12541: TNS:no listener
```

- then the listener service on the database server is not running. Start the listener up as follows:

```
$ lsnrctl start
```

- and verify that the listener is running:

```
$ lsntctl status
```

- Check the contents of the file on the server `/u01/app/oracle/product/10.0.1/db_1/network/admin/listener.ora`. If the file contains a `(HOST = <DNS name>)` statement, change the DNS name to the server's IP address. This is a way of circumventing a bug in Oracle Release 10.1 and should be removed in the 10.2 release. If DHCP provides the IP address, set it to: `(HOST = 0.0.0.0)` and restart the listener service:

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

- If the following error appears when attempting to connect to the database from a remote client

```
ORA-12516: TNS:listener could not find available handler with matching protocol stack
```

- verify the content of the file `/u01/app/oracle/product/10.0.1/db_1/network/admin/sqlnet.ora`

- Desperate course of Action #1

Elegantly restart the database from the server console:

```
$ su – oracle
Password:
$ sqlplus / as sysdba
SQL> shutdown immediate
ORACLE instance shut down
SQL> startup
ORACLE instance started
```

- Desperate course of Action #2

Abort the database and restart from the server console if an elegant shutdown seems to hang:

```
$ su – oracle
Password:
```

```
$ sqlplus / as sysdba
SQL> shutdown abort
ORACLE instance shut down
SQL> startup
ORACLE instance started
```

- and restart it elegantly

```
SQL> shutdown immediate
ORACLE instance shut down
SQL> startup
ORACLE instance started
```

- **Desperate course of Action #3**

Delete the `sqlnet.ora` file on both the server and the clients. This will cause both server and client to unambiguously connect using TNSNAMES definitions, as set up in the `tnsnames.ora` file on both client and server.

- **Desperate course of Action #4**

Check that the server's `tnsnames.ora` file refers to the hostnames or IP addresses, rather than 'localhost'. Then, copy the server's `tnsnames.ora` to the client's `tnsnames.ora` file. This will remove a further element of doubt.

5. I get an error message that I don't understand

- Oracle error messages are in the form of <group>-<number>, where <group> is a 3-letter code and <number> is a 5-digit code. To get the meaning and (sometimes) the cause of an Oracle error message, enter the following:

```
$ oerr <group> <number>
```

6. Cannot connect remotely over SSH

If you are connecting from Windows, a frequently-used Windows program to connect to UNIX servers of SSH is putty.exe. It is very robust and free.

- Check that the username and the password are valid by trying it out on the server console.

- Check that the SSH daemon on the server is running:

```
$ su –
Passwd:
# /etc/init.d/sshd status
```

- Start the SSH daemon if it is not running

```
$ su –
Passwd:
# /etc/init.d/sshd start
```

- The `/etc/ssh/ssh_config` file may have been incorrectly configured. It should contain at least the following lines:

```
Host *
ForwardX11 yes
```

- The `/etc/ssh/sshd_config` file may have been incorrectly configured. It should contain at least the following line:

```
X11Forwarding yes
```

If in doubt, ensure that this is the only line in this file in case of conflicting settings by issuing the following command:

```
$ su -
Password:
# echo "X11Forwarding yes" > /etc/ssh/sshd_config
```

- Restart the SSH daemon to effect any changes to the sshd_config file:

```
$ su -
Passwd:
# /etc/init.d/sshd restart
```

7. I cannot start a remote X-Windows session on a Windows PC

   A frequently-used MS-Windows program to run X-Windows sessions on the PC is XWin32 from www.starnet.com. This causes the Win32 PC to act as an X-Windows server that receives display commands from the UNIX server. The following fault finding technique is based on this product.

   - Check that the username and the password that you log through the X11 session are valid by trying the username and password out on a text-based console session.

   - The preferred way to connect to a remote server is over SSH. To resolve SSH connectivity issues, refer to the above item.

   - The host name or IP address of the server should be set up in the X-host list on the Security tab of the XWin32 configuration program. Also tick the Access Control box.

   - Restart the XWin32 server and accept all connections from localhost(127.0.0.1).

8. Perl or KORN shell scripts that I wrote (on a Windows machine) do not want to run

   - If you get the error when running a script

```
: bad interpreter: No such file or directory
```

   - then the file is probably still in DOS-mode and needs to be converted to Unix mode using the following command:

```
$ dos2unix <filename>
```

   - If you get the error

```
-bash: ./<filename>: Permission denied
```

   - then the file has not been set to executable mode. This can be set as follows:

```
$ chmod +x <filename>
```