

# Oracle on Solaris Cheat Sheet

**Author: James L. Colestock**  
**Last Modified: 8/11/2006**

1. I have an Ultra 60 at home that I use in this example. First I obtained the Solaris 9 binaries from Sun and reinstalled the Operating System using a console cable. If you already have a clean install, then you are all set to begin.

2. Obtain any Oracle documentation you feel may be helpful; in my case:

[http://download-west.oracle.com/docs/cd/B19306\\_01/install.102/b15690.pdf](http://download-west.oracle.com/docs/cd/B19306_01/install.102/b15690.pdf)  
(Oracle® Database Installation Guide 10g Release 2 (10.2) for Solaris Operating System (SPARC 64-Bit) B15690-02)

3. Download the Oracle binaries from <http://otn.oracle.com> and transfer them to the target host; gunzip and cpio -idmv as appropriate:

```
# gunzip 10gr2_db_sol.cpio.gz
# cpio -idmv < 10gr2_db_sol.cpio
```

4. Determine the Operating System version and verify that it is certified for the version you are using by visiting Metalink and clicking on the 'Certify' tab; in my case:

```
# SunOS sun9 5.9 Generic_118558-11 sun4u sparc SUNW,Ultra-60
```

5. As root, determine that the host's hardware fits the installation requirements (as prescribed in the aforementioned documentation):

Atleast 1024MB of memory is required:

```
# /usr/sbin/prtconf | grep "Memory"
Memory size: 1024 Megabytes
```

For 1024MB of Physical Memory 1.5X in swap is recommended:

```
# /usr/sbin/swap -s
total: 17360k bytes allocated + 3840k reserved = 21200k used, 2949848k available
```

There must be 400MB or more free space in /tmp:

```
# df -h /tmp
Filesystem                size      used  avail capacity  Mounted on
/dev/dsk/c0t0d0s3          963M      1.0M   904M      1%      /tmp
```

I will store my Oracle binaries on a specific mount point called /oracle – I should have at least 3GB of free disk space there as well:

```
# df -h /oracle
Filesystem                size      used  avail capacity  Mounted on
/dev/dsk/c0t0d0s7          5.7G      5.8M   5.6G      1%      /oracle
```

Determine that the system architecture can support Oracle – the output of the command should be exactly as follows in order to proceed with the installation:

```
# /bin/isainfo -kv
64-bit sparcv9 kernel modules
```

6. As root, determine that the host's software fits the installation requirements (as prescribed in the aforementioned documentation):

For example, the following packages are needed for my distribution:

```
SUNWarc
SUNWbtool
SUNWhea
SUNWlibm
SUNWlibms
SUNWsprot
SUNWtoo
SUNWilof
SUNWilcs
SUNWi15cs
SUNWxwfont
SUNWsprox
```

Check via the following command:

```
# pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibm SUNWlibms SUNWsprot SUNWtoo
SUNWilof SUNWilcs SUNWi15cs SUNWxwfont SUNWsprox
```

```
system      SUNWarc      Archive Libraries
system      SUNWbtool    CCS tools bundled with SunOS
system      SUNWhea      SunOS Header Files
system      SUNWi15cs    X11 ISO8859-15 Codeset Support
system      SUNWilcs     X11 ISO8859-1 Codeset Support
system      SUNWilof     ISO-8859-1 (Latin-1) Optional Fonts
system      SUNWlibm     Forte Developer Bundled libm
system      SUNWlibms    Forte Developer Bundled shared libm
system      SUNWsprot    Solaris Bundled tools
system      SUNWsprox    Sun WorkShop Bundled 64-bit make library
system      SUNWtoo      Programming Tools
system      SUNWxwfont   X Window System platform required fonts
```

My installation lacks Sun One Studio C and C++ compilers, so I opted to install the gcc compiler from the following web site: <http://www.sunfreeware.com/indexsparc9.html> and downloaded the precompiled version of gcc 3.4.2 using the following note to install: <http://www.sunfreeware.com/gcc.html>. Specifically, I transferred the gzipped file to the target host unzipped it and then issued the `pkgadd -d <pkg>` command. I did the same for a dependent package named libiconv 1.9.2, which is also available on that site. After that I updated my `PATH` as appropriate and did a version check:

```
# PATH=$PATH:/usr/local/bin:/usr/ccs/bin; export PATH
# which gcc
/usr/local/bin/gcc
# gcc -version
gcc (GCC) 3.4.2
Copyright (C) 2004 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
# MANPATH=$MANPATH:/usr/local/man; export MANPATH
```

Obviously perform the above only if you lack the Sun Studio version of the C/C++ compilers.

A System Administrator buddy of mine, suggested that I try to download Sun Studio 11 (it includes the Sun compilers which negate using gcc), which is now freely available here: <http://www.sun.com/download/index.jsp?cat=Application%20Development&tab=3&subcat=Development%20Tools>

You will need bunzip2 -- if you don't already have it --  
<http://www.sunfreeware.com/programlistsparc9.html#bzip2>

I downloaded Sun Studio 11, transferred to my machine, and then installed via:

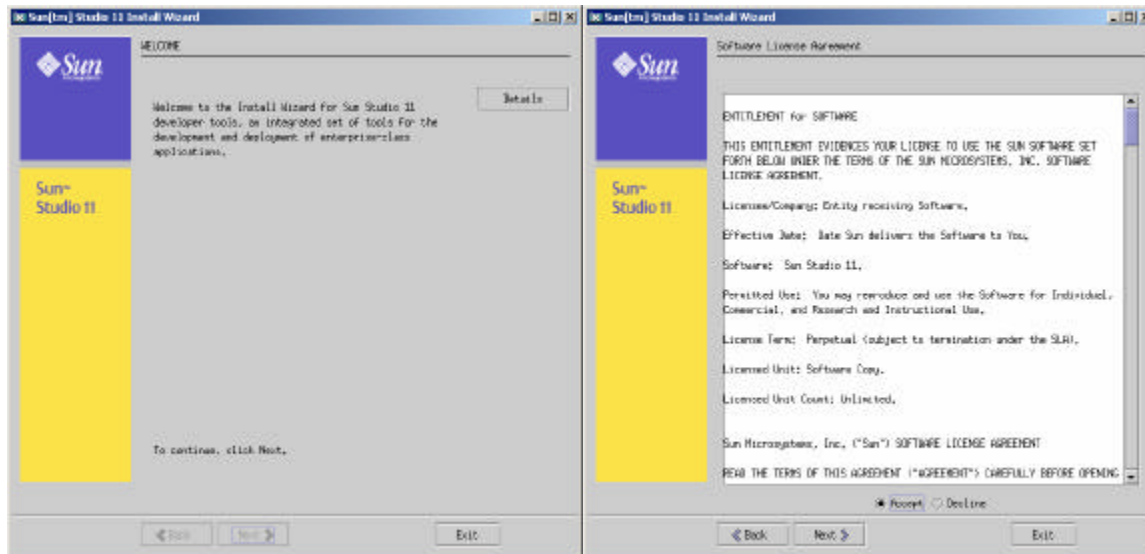
```
# /usr/local/bin/bunzip2 studioll-sol-sparc.tar.bz2
# tar -xvf studioll-sol-sparc.tar
```

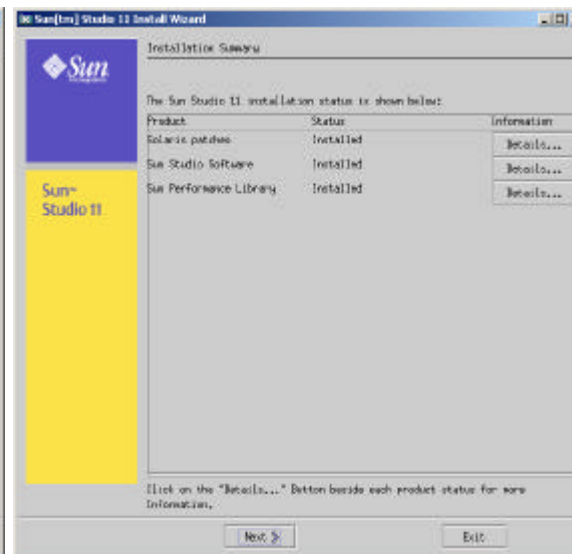
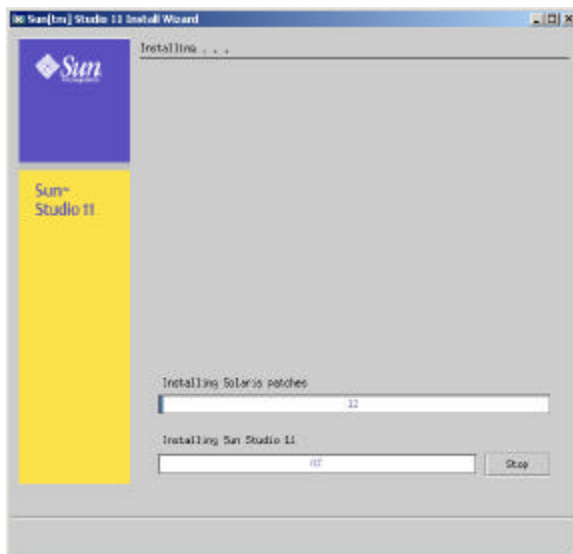
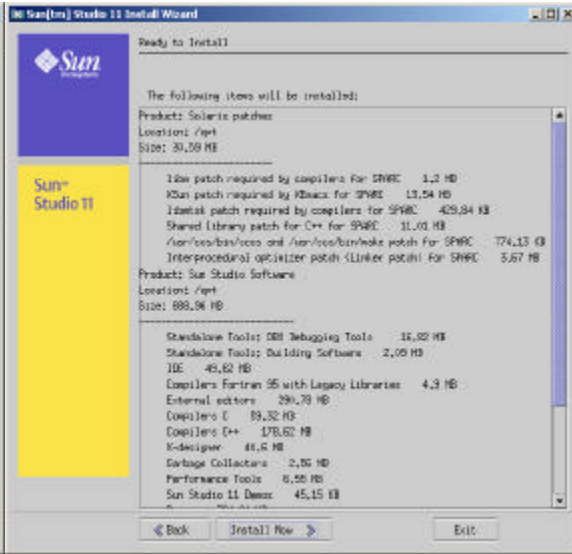
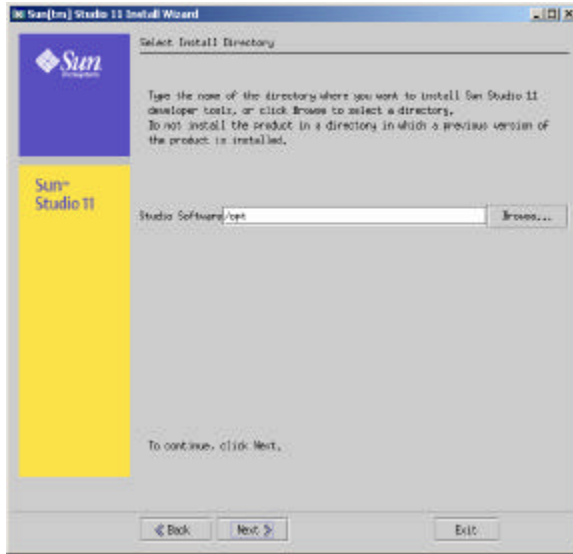
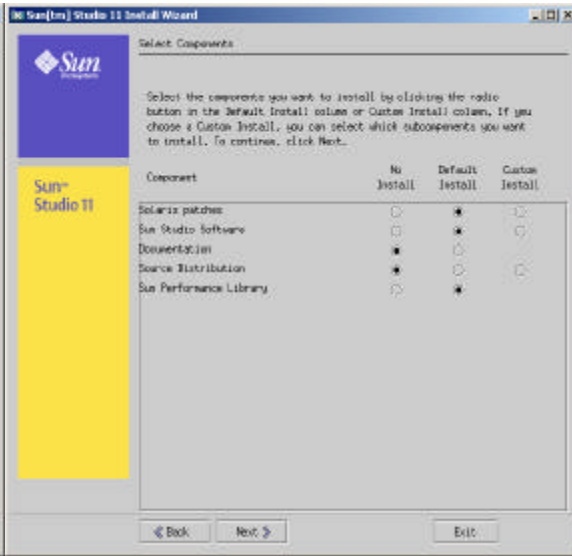
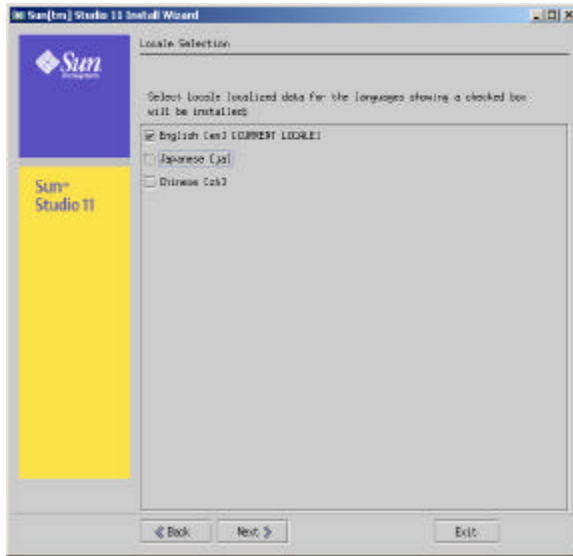
You will have to make sure your environment has all the appropriate references to java in it; here is my environment before launching the installer:

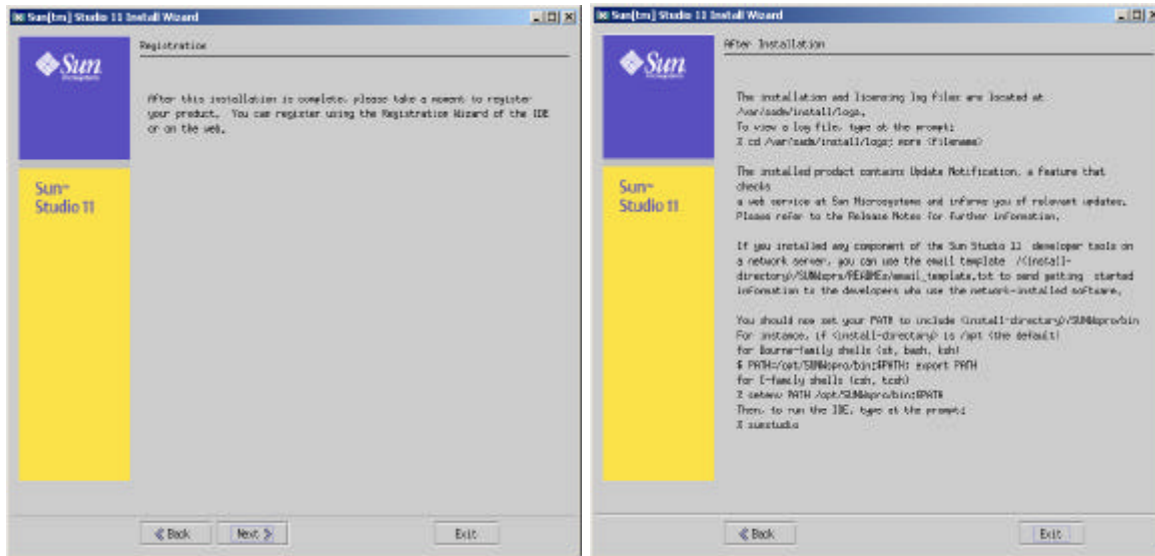
```
# env
CLASSPATH=/usr/j2se/jre/lib:/usr/j2se/lib
DISPLAY=192.168.1.2:0.0
HOME=/
JAVA_HOME=/usr/j2se/jre
LOGNAME=root
PATH=/usr/j2se/jre/bin:/usr/bin:/bin:/usr/sbin:/sbin
SHELL=/sbin/sh
SSH_CLIENT=192.168.1.2 2746 22
SSH_TTY=/dev/pts/1
TERM=vt100
TZ=US/Pacific
USER=root
# ./installer
```

```
Aug 11, 2006 1:52:58 PM java.util.prefs.FileSystemPreferences$2 run
INFO: Created user preferences directory.
```

Here are screenshots from my installation:







Afterwards, I check to see if sunstudio launches without fail via:

```
# PATH=/opt/SUNWspro/bin:$PATH; export PATH
# sunstudio
```

The following patches are required for my distribution. I check for their existence via:

```
# showrev -p | grep <patch>

112233-11, SunOS 5.9: Kernel Patch
111722-04, SunOS 5.9: Math Library (libm) patch
115675-01, SunOS 5.9: liblgrp API
113471-08, SunOS 5.9: Miscellaneous SunOS Commands Patch
115675-01, SunOS 5.9: /usr/lib/liblgrp.so Patch
```

If you lack the patch, download it at <http://sunsolve.sun.com> and install as follows:

```
# unzip 112233-11.zip
# /usr/sbin/patchadd 112233-11
```

#### 7. Check the Network Setup to insure compatibility with the Oracle installation:

The following command should yield an entry for 'files':

```
# cat /etc/nsswitch.conf | grep hosts
# "hosts:" and "services:" in this file are used only if the
hosts:      files
```

Oracle recommends that the following command return a fully-qualified hostname, however, mine does not:

```
# hostname
sun9
```

Oracle recommends that the following command return nothing:

```
# domainname
```

Here is the contents of my /etc/hosts file:

```
# more /etc/hosts
#
```

```
# Internet host table
#
192.168.1.10    sun9.colestock.com  sun9  loghost
127.0.0.1      localhost
```

8. Perform the following required UNIX tasks:

Create the required O/S groups:

```
# groupadd -g 200 oinstall
# groupadd -g 201 dba
```

Create the Oracle user and any dependencies:

```
# mkdir /home/oracle
```

In order to perform the aforementioned I first had to comment out the following line in `/etc/auto_master` as reboot the system:

```
#/home          auto_home          -nobrowse
```

Create the Oracle user:

```
# /usr/sbin/useradd -u 175 -g dba -G oinstall -d /home/oracle -s /bin/bash -c
"Oracle Software Owner" oracle
```

I opted not to use the `oinstall` group as the primary group, since this will be a standalone server with a single Oracle installation.

```
# chown oracle:dba /home/oracle
# passwd -r files oracle
New Password:
Re-enter new Password:
passwd: password successfully changed for oracle
```

Verify that the user 'nobody' exists:

```
# id nobody
uid=60001(nobody) gid=60001(nobody)
```

Configure the system-level semaphore and file descriptor limits considering the platform and physical memory installed in the server. Ensure that swap 2X the physical memory on the server exists. Normally `shmmax` should be half of physical memory, but I wouldn't be too concerned about this on development and/or test systems. Production systems with higher process counts and the like will need a more refined configuration. The directives not explicitly modified below, leave at the default value. (Note: Only change these items if you have to; in the case that you do, make sure to test thoroughly, etc.):

```
# cp /etc/system /etc/system.orig
```

Edit the `/etc/system` with the appropriate values for your system:

```
# vi /etc/system

set noexec_user_stack=1
set semsys:seminfo_semmni=100
set semsys:seminfo_semmns=1024
set semsys:seminfo_semmsl=256
set semsys:seminfo_semvmx=32767
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
```

```
# reboot
```

### Create necessary Oracle-owned directories:

```
# mkdir /oracle/product
# mkdir /var/opt/oracle
# chown oracle:dba /oracle/product
# chown oracle:dba /var/opt/oracle
```

9. As oracle, modify the /home/oracle/.bash\_profile profile to include at least the following:

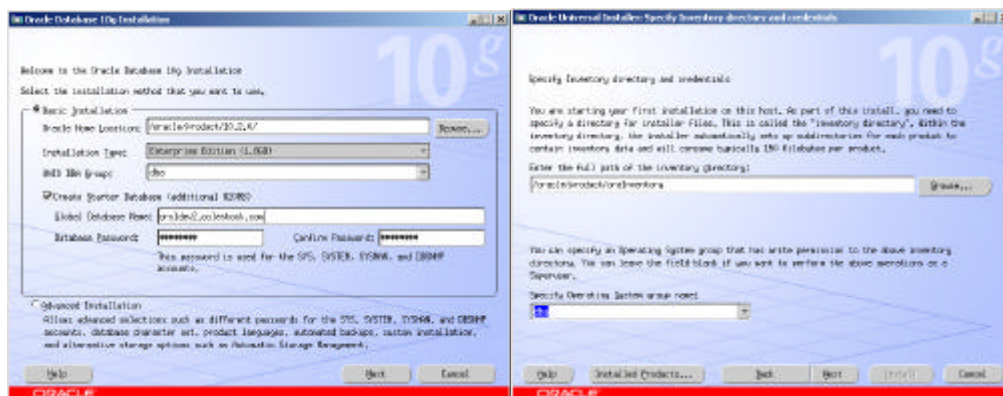
```
# vi $HOME/.bash_profile

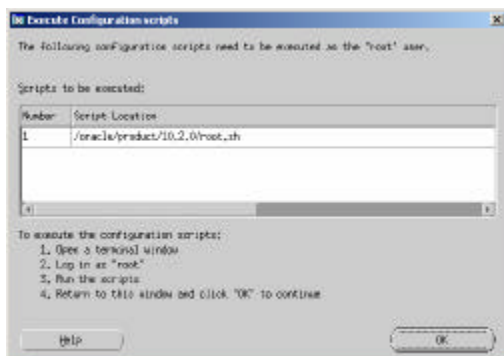
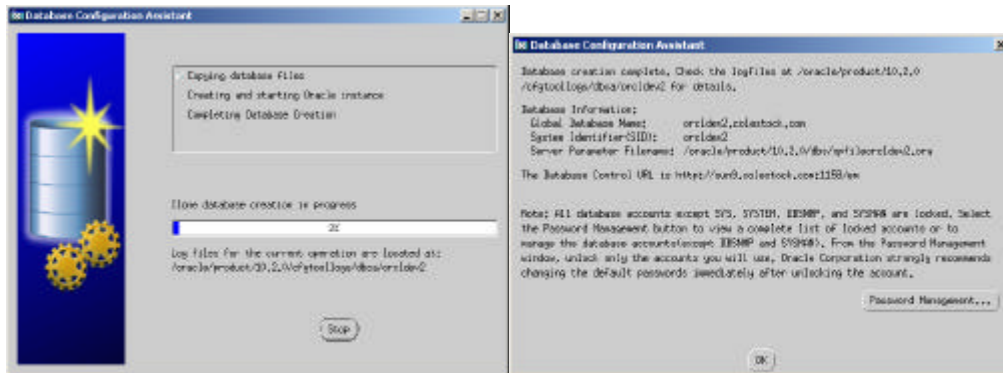
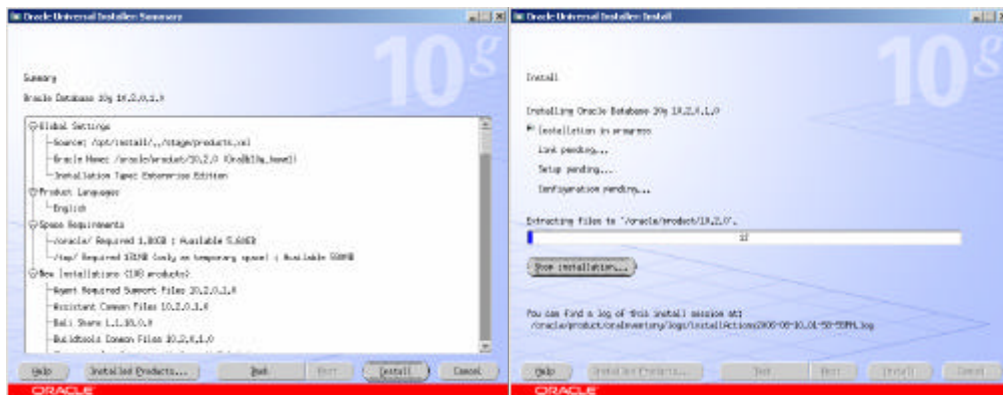
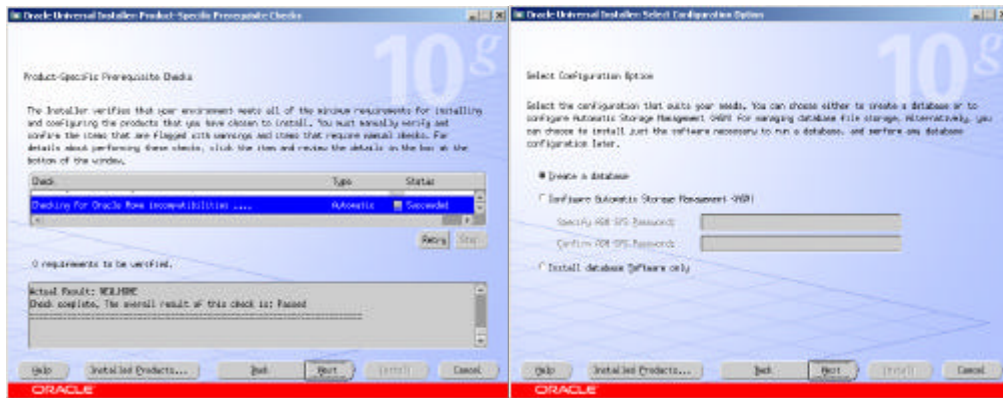
umask 022
export
PATH=/usr/SUNWSpro/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/ccs/bin:/usr/local/bin:/u
sr/openwin/bin
export MANPATH=/usr/local/man:$MANPATH
export ORACLE_BASE=/oracle/product
export DISPLAY=192.168.1.2:0.0
```

10. As oracle, verify your environment, install and patch Oracle software as appropriate (you will need an X windowing mechanism handy):

```
# su - oracle
Sun Microsystems Inc. SunOS 5.9 Generic May 2002
bash-2.05$ echo $DISPLAY
192.168.1.2:0.0
bash-2.05$ /opt/runInstaller
Starting Oracle Universal Installer...
Checking installer requirements...
Checking operating system version: must be 5.8, 5.9 or 5.10. Actual 5.9
Passed
Checking Temp space: must be greater than 250 MB. Actual 800 MB Passed
Checking swap space: must be greater than 500 MB. Actual 2911 MB Passed
Checking monitor: must be configured to display at least 256 colors
```

Follow the Installer and select options appropriate for your environment. This is my test server so I am doing a basic configuration with a starter database. The following are screen shots of my installation:





```
# id
uid=0(root) gid=1(other)
```

```
# /oracle/product/10.2.0/root.sh
Running Oracle10 root.sh script...
```

The following environment variables are set as:

```
ORACLE_OWNER= oracle
ORACLE_HOME= /oracle/product/10.2.0
```

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 ...
```

Creating /var/opt/oracle/oratab file...

Entries will be added to the /var/opt/oracle/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.



11. As oracle, update `/var/opt/oracle/oratab` as appropriate (indicating 'Y' for a service that you want started at boot time)
12. As oracle, update the `$HOME/.bash_profile` to source the database (optional), You may use a combination of `export` commands and calls to `/usr/local/bin/oraenv`. For example, I add the following lines:

```
export ORAENV_ASK=NO
export ORACLE_SID=orcldev2
. /usr/local/bin/oraenv
```

13. Generate the Client Static Library:

```
bash-2.05$ genclntst
ar: creating /oracle/product/10.2.0/lib/libclntst10.a
Created /oracle/product/10.2.0/lib/libclntst10.a
ar: creating /oracle/product/10.2.0/lib32/libclntst10.a
Created /oracle/product/10.2.0/lib32/libclntst10.a
```

14. Configure command file for native compilation of PL/SQL (optional):

```
# su - oracle
Sun Microsystems Inc. SunOS 5.9 Generic May 2002
bash-2.05$ vi $ORACLE_HOME/plsql/spnc_commands
```

If you wish to use `gcc`, comment out the line that uses `cc` and uncomment the line that uses `gcc`. In either case, be sure to update the path to `cc` or `gcc` to reflect the true path on you system; for example,

mine is `/opt/SUNWspro/bin/cc`; save and close.

15. Make all necessary modifications to strap the oracle services into the system's runtime faculties:

- a. Modify the `$ORACLE_HOME/bin/dbstart` and `$ORACLE_HOME/bin/dbshut` scripts to not 'shutdown abort' the databases if they are already running and also update the `ORATAB` environment variable found in each to point to the Solaris-specific location, `/var/opt/oracle/oratab`. Test and alter as necessary before continuing.

- b. As root:

```
# touch /etc/init.d/dbora
```

- c. Edit the file as appropriate, using the following as an example:

```
#!/bin/sh
#
# Set ORACLE_HOME to be equivalent to the $ORACLE_HOME
# from which you wish to execute dbstart and dbshut;
#
# Set ORA_OWNER to the user id of the owner of the
# Oracle database in ORACLE_HOME.
#
ORACLE_HOME=/oracle/product/10.2.0
ORA_OWNER=oracle

case "$1" in
'start')
# Start the Oracle databases:
su - $ORA_OWNER -c $ORACLE_HOME/bin/dbstart
# Start the TNS Listener
su - $ORA_OWNER -c "$ORACLE_HOME/bin/lsnrctl start"
;;
'stop')
# Stop the TNS Listener
su - $ORA_OWNER -c "$ORACLE_HOME/bin/lsnrctl stop"
# Stop the Oracle databases:
# The following command assumes that the oracle login
# will not prompt the user for any values
su - $ORA_OWNER -c $ORACLE_HOME/bin/dbshut
;;
esac
# End of script dbora
```

- d. Change the permissions on the file and then create the appropriate symbolic links:

```
# chmod u+x dbora
# ln -s /etc/init.d/dbora /etc/rc0.d/K01dbora
# ln -s /etc/init.d/dbora /etc/rc3.d/S99dbora
```

- e. As root, test the `dbora` service by either issuing a shutdown reboot or an `init` command, similar to the following:

```
# reboot
# init 6
```