Oracle Setup for the Model Repository

This appendix explains how to install, configure, and maintain an Oracle database to support the Model Repository.

Oracle RDBMS Install Basics

The Model Repository is an SA Core Component that stores information in an Oracle database. It stores the following information:

- database users
- database user privileges
- schema information
- baseline data

The SA distribution Media includes a separate Oracle 11g RDBMS software and database installation dual layer DVD. You can simply mount this DVD on the server you plan to use to host the Model Repository and run the installation. See Using the HP-Supplied Oracle RDBMS Software and Database on page 12 in this chapter for information about the installation steps

You can also use the Oracle Universal Installer to manually install an Oracle 10g or 11g database, however, you will need to perform certain tasks that the HP-supplied database performs automatically on installation. If you plan to use an existing database installation, you must ensure that the database is configured correctly for use with the SA Model Repository.

If you plan to use the Oracle Universal Installer to install the Oracle RDBMS software and database, or will use an existing Oracle database, then you should read the following sections:

The Oracle database must be created before you install the Model Repository.

Supported Oracle Versions

Support for the Model Repository is limited to certain versions of Oracle running on certain versions of operating systems. HP strongly recommends that you also apply the latest Oracle CPU or PSU patches. For manual installations, SA supports the Oracle Standard Edition, Standard Edition One and the Oracle Enterprise Edition. Table 1 lists the supported Oracle versions.

Table 1 Supported Operating Systems and Oracle Versions

Operating System	Supported Oracle Versions (Standard or Enterprise Edition)
SunOS 10 (SPARC)-64 bits	10.2.0.4, 11.1.0.7, 11.2.0.2
Red Hat Enterprise Linux 4 AS x86_64	10.2.0.4, 11.1.0.7, 11.2.0.2
Red Hat Enterprise Linux 5 AS x86_64	11.1.0.7, 11.2.0.2
SUSE Linux Enterprise Server 10 x86_64	10.2.0.4, 11.1.0.7, 11.2.0.2
See the SA 9.10 Supported Platform Matrix for additional information.	

Oracle 10.2.0.3 is not supported by SA due to known incompatibilities.

Multiple Oracle Versions and Multimaster Cores

For the database export to succeed during the installation of a Multimaster core, the version of the target database cannot be 10.x if the source database is 11.x. Table 2 lists these allowed version combinations.

Table 2 Database Versions Allowed for Multimaster

Source Database Version	Target Database Version	Allowed?
10	10	Y
10	11	Y
11	10	N
11	11	Y

Hardware Requirements

The server that will host the Oracle database for the Model Repository must meet the hardware requirements listed in this section.

Linux Requirements

The following are hardware requirements for running Oracle 11g under Linux:

- The recommended physical memory is 16 GB or more of RAM. An HP-supplied Oracle installation will use around 1.8-2.0 GB of memory. The Oracle SGA memory can be increased after database installation. You can use the following command to check memory status:

  `grep MemTotal /proc/meminfo`

- Required available swap space is shown in Table 3:

Table 3 Required Available RAM Swap Space

RAM Between:	Available Swap Space
4 GB and 8 GB	2 times the size of RAM
8 GB and 32 GB	1.5 times the size of RAM
More than 32 GB	32GB

  You can use the following command to check swap space:

  `grep SwapTotal /proc/meminfo`

- Recommended shared memory available required for automatic memory management should be greater than 1GB.

  You can use the following command to check available shared memory:

  `df -h /dev/shm/`

- Free `tmp` space should be 1GB or more of `/tmp` directory space

  You can use the following command to check `tmp` space:

  `df -k /tmp`

- Required operating system version:

  — Red Hat Enterprise Linux 4 Update 7

  — Red Hat Enterprise Linux 5 Update 2

  — SUSE Linux Enterprise Server 10 SP2

  You can use the following command to determine the distribution and version of Linux installed, :

  `# cat /proc/version`

- Required Kernel version:
  - Red Hat AS 4: 2.6.9 or later
  - Red Hat AS 5: 2.6.18 or later
  - SUSE Linux 10: 2.6.16.21 or later

  You can use the following command to check the kernel versions:

  ```
  uname -r
  ```

  You can use the following command to check the platform:

  ```
  uname -mi
  ```

  You can use the following command to check the processor type:

  ```
  grep "model name" /proc/cpuinfo
  ```

Solaris Requirements

The following are hardware requirements for running Oracle 11g under Solaris:

- The recommended physical memory is 16 GB or more of RAM. An HP-supplied Oracle installation will use around 1.8-2.0 GB of memory. The Oracle SGA memory can be increased after database installation.

  You can use the following command to check the physical memory:

  ```
  /usr/sbin/prtconf | grep "Memory"
  ```

- Required available swap space is shown in Table 4:

Table 4 Required Available Swap Space

RAM Between:	Available Swap Space
4 GB and 16 GB	Equal to the size of the RAM
More than 16 GB	16 GB

  You can use the following command to check the swap space:

  ```
  /usr/sbin/swap -s
  ```

- Free `tmp` space should be 1GB `/tmp` directory space.

  You can use the following command to check `tmp` space:

  ```
  df -k /tmp | grep / | awk '{ print $3 }'
  ```

- Required operating system version is: 5.10

  You can use the following command to check the operating system version:

  ```
  uname -a
  ```

  To determine the update level of Oracle Solaris installed:

  ```
  $ cat /etc/release
  ```

- System architecture should show `64-bit sparcv9 kernel` modules

  You can use the following command to check system architecture:

  ```
  /bin/isainfo -kv
  ```

Model Repository (Database) Disk Space Requirements

Additional disk space is required for the Oracle software and the Model Repository data files. Keep in mind that storage requirements for the database grow as the number of managed servers and database activity grows.

As a benchmark figure, you should allow an additional 3.1 GB of database storage for every 1,000 servers in the facility that SA manages. When sizing the tablespaces, follow the general guidelines described in Table 5. If you need to determine a more precise tablespace sizing, contact your technical support representative.

Table 5 Tablespace Sizes

Tablespace	MB/1000 Servers	Minimum Size
AAA_DATA	256 MB	256 MB
AAA_INDX	256 MB	256 MB
AUDIT_DATA	256 MB	256 MB
AUDIT_INDX	256 MB	256 MB
LCREP_DATA	3500 MB	2000 MB
LCREP_INDX	1700 MB	900 MB
TRUTH_DATA	1400 MB	800 MB
TRUTH_INDX	500 MB	500 MB
STRG_DATA	1300 MB	700 MB
STRG_INDX	400	400 MB

Hostname Setup

You must be able to ping the database server hostname. To verify this, enter the following command:

Linux/Red Hat Linux

```
# ping <hostname>
```

or, on the database server, enter the following command:

```
# hostname
```

SUSE Linux

```
# hostname -f
```

If the hostname is not configured correctly, Oracle will not start and you will encounter the following error:

```
ORA-00600: internal error code, arguments: [keltnfy-ldmInit], [46], [1],
[], [], [], [], []
```

Operating System Requirements

The following sections list the operating system requirements for Oracle 11g. The SA Installer performs an automated check to ensure that these requirements are met on the Oracle host before proceeding with the installation of the Oracle 11g software and database.

If you create the database using the Oracle Universal Installer rather than the SA Installer, you must check for these packages and patches manually.

Required Packages for Red Hat Enterprise Linux AS 4 x_64

The following packages are required for Oracle 11g on Red Hat Enterprise Linux AS 4 64 x86_64. These packages must be the versions listed or higher.

As of Oracle Database 11g Release 2 (11.2.0.2), you are only required to install the 64-bit packages and the 32-bit package `gcc-32bit-4.3` listed in Table 6. All other 32-bit packages listed in the table (shaded rows) are no longer required for a database installed under Linux x86-64.

However, for any Oracle Database 11g release *prior to 11.2.0.2*, all the 32-bit and 64-bit packages listed in the table are still required.

Table 6 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64

Required Packages	Version
binutils	2.15.92.0.2
compat-libstdc++	33-3.2.3
compat-libstdc++	33-3.2.3 (32-bit)
elfutils-libelf	0.97
elfutils-libelf-devel	0.97
expat	1.95.7
gcc	3.4.6
gcc-c++	3.4.6
glibc	2.3.4-2.41
glibc	2.3.4-2.41 (32-bit)
glibc-common	2.3.4
glibc-devel	2.3.4
glibc-devel	2.3.4 (32-bit)
glibc-headers	2.3.4
libaio	0.3.105

Table 6 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64 (cont'd)

Required Packages	Version
libaio	0.3.105 (32-bit)
libaio-devel	0.3.105
libaio-devel	0.3.105 (32-bit)
libgcc	3.4.6
libgcc	3.4.6 (32-bit)
libstdc++	3.4.6
libstdc++	3.4.6 (32-bit)
libstdc++-devel	3.4.6
make	3.80
numactl	0.6.4.x86_64
pdksh	5.2.14
sysstat	5.0.5

Required Packages for Red Hat 5 Server x86_64, Asianux Server 3, and Oracle Linux 5

The following packages are required for Oracle 11g on Red Hat Enterprise Linux 5 Server x86_64, Asianux Server 3 and Oracle Linux 5. These packages must be the versions listed or higher (32-bit packages in shaded rows):.

Table 7 Required Packages for Red Hat 5 Server x86_64, Asianux Server 3, and Oracle Linux 5

Required Packages	Version
binutils	2.17.50.0.6
compat-libstdc++	33-3.2.3
compat-libstdc++	33-3.2.3 (32-bit)
elfutils-libelf	0.125
elfutils-libelf-devel	0.125
gcc	4.1.2
gcc-c++	4.1.2
glibc	2.5-24
glibc	2.5-24 (32-bit)
glibc-common	2.5

Table 7 Required Packages for Red Hat 5 Server x86_64, Asianux Server 3, and Oracle Linux 5 (cont'd)

Required Packages	Version
glibc-devel	2.5
glibc-devel	2.5 (32-bit)
glibc-headers	2.5
ksh	20060214
libaio	0.3.106
libaio	0.3.106 (32-bit)
libaio-devel	0.3.106
libaio-devel	0.3.106 (32-bit)
libgcc	4.1.2
libgcc	4.1.2 (32-bit)
libstdc++	4.1.2
libstdc++	4.1.2 (32-bit)
libstdc++-devel	4.1.2
make	3.81
numactl-devel	0.9.8.x86_64
sysstat	7.0.2

See your Oracle documentation to determine requirements for supported databases earlier than Oracle 11g Release 2.

Required Packages for SUSE Linux Enterprise Server 10 x86_64

The following packages are required for Oracle 11g on SUSE Linux Enterprise Server 10 x86_64. These packages must be the versions listed or higher (32-bit packages in shaded rows):

Table 8 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64

Required Packages	Version
binutils	2.16.91.0.5
compat-libstdc++	5.0.7
gcc	4.1.0
gcc-c++	4.1.2
glibc	2.4-31.63

Table 8 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64 (cont'd)

Required Packages	Version
glibc	2.4-31.63 (32-bit)
glibc-devel	2.4-31.63
glibc-devel	2.4-31.63 (32-bit)
ksh	93r-12.9
libaio	0.3.104
libaio	0.3.104 (32bit)
libaio-devel	0.3.104
libaio-devel	0.3.104 (32bit)
libelf	0.8.5
libgcc	4.1.2
libstdc++	4.1.2
libstdc++-devel	4.1.2
make	3.80
numactl	0.9.6.x86_64
sysstat	8.0.4

To verify whether these RPMs are installed on the OS, enter the following command:

```
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' <rpm_name>
```

See your Oracle documentation to determine requirements for supported databases earlier than Oracle 11g Release 2.

Required Packages for Solaris 10

The following packages are required for Oracle 11g on Solaris 10 servers. These packages must be the versions listed or higher:

Oracle 11g

The following packages (or later versions) are required for Oracle Database 11g Release 2 (11.2) for Oracle Solaris:

```
SUNWarc
SUNWbtool
SUNWhea
SUNWlibC
SUNWlibm
SUNWlibms
```

```
SUNWsprot
SUNWtoo
SUNWi1of
SUNWi1cs (ISO8859-1)
SUNWi15cs (ISO8859-15)
SUNWxwfnt
SUNWcsl
```

To determine if the required packages are installed, enter commands similar to the following:

```
# pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibC SUNWlibms SUNWsprot \
 SUNWtoo SUNWi1of SUNWi1cs SUNWi15cs SUNWxwfnt
```

See your Oracle documentation to determine requirements for supported databases earlier than Oracle 11g Release 2.

Required Patches for Solaris 10

The following patches are required on Solaris 10 for Oracle 11g:

Table 9 Required Patches for Solaris 10

Patch	Version	Comment
120753	06	SunOS 5.10: Microtasking libraries (libmtsk) patch
139555	08	SunOS 5.10 (139555-08 obsoletes 139574-03)
141444	09	
141414	02	

To determine if an operating system patch is installed, enter a command similar to the following:

```
# /usr/sbin/patchadd -p | grep patch_number(without version number)
```

Required Patch for Manual Oracle 11g Installations

If you plan to install Oracle 11.1.07 (for the Model Repository) using the Oracle Universal Installer or use an existing Oracle 11.1.07 installation, you must ensure that Oracle Bug Patch 8300752 has been applied to that database.

Oracle 10g on Solaris 10 Servers

When Oracle 10.2 is installed on T2000 hardware with the Solaris 10 operating system, the SA Installer hangs during the installation of the Model Repository. The Oracle alert.log includes errors, such as the following:

```
MMNL absent for 28552 secs; Foregrounds taking over
Wed Aug 2 12:45:57 2006
MMNL absent for 28853 secs; Foregrounds taking over
Wed Aug 2 12:50:57 2006
MMNL absent for 29151 secs; Foregrounds taking over
```

Customers should look at Bug 6385446 from Sun Microsystems and apply Patches 118833-18, 119578-24 and 119254-24 as per:

http://sunsolve.sun.com/search/document.do?assetkey=1-26-102289-1

Database Server Time Requirements

Core Servers (either Single Core or Multimaster) and Satellite Core Servers must meet the following requirements. These time requirements do not apply to Managed Servers.

- All SA Core Servers must have their time zone set to Coordinated Universal Time (UTC).
- All SA Core Servers must maintain synchronized system clocks. Typically, you will synchronize the system clocks through an external server that uses NTP (Network Time Protocol) services.

Linux Time Configuration

To configure the time zone on a Linux server, perform the following tasks:

1. Copy or link

   `/usr/share/zoneinfo/UTC`

   to

   `/etc/localtime`.

2. Ensure that the `/etc/sysconfig/clock` file contains the following lines:

   ```
   ZONE="UTC"
   UTC=true
   ```

Solaris Time Configuration

To configure the time zone on a Solaris server, verify that the `/etc/TIMEZONE` file contains the following line:

```
TZ=UTC
```

Oracle RDBMS State Checker Script

Previous SA versions provided the standalone Oracle RDBMS State Checker script `/tools/truth_oracle_state_checker.sh` which was manually invoked before installs and upgrades to verify that the Oracle database was configured correctly. The script verified that certain required parameters were specified in the database and issued warnings when the parameters were not specified according to SA requirements.

As of SA 9.10, this script has been encorporated into the SA Installer. The script runs during SA Core installation and perfoms database checks before installs or upgrades. The standalone script is no longer available to customers.

Using the HP-Supplied Oracle RDBMS Software and Database

If you are manually installing the Oracle RDBMS software and database, skip this section and go to Pre-Installation Tasks (Oracle Universal Installer) on page 16.

The SA distribution Media includes an Oracle 11g database installation DVD. You can simply mount this DVD on the server you plan to use to host the Model Repository and enter the following command to begin the Oracle installation.

```
/<distro>/oracle_sas/install_opsware.sh --verbose
```

You will see a screen similar to this:

```
Install Type: "Oracle RDBMS for SAS"

The Opsware Installer will now interview you to obtain the installation
parameters it needs. You can use the following keys to navigate forward
and backward through the list of parameters:

Control-P - go to the previous parameter
Control-N - go to the next parameter
Return - accept the default (if any) and go to the next parameter
Control-F - finish parameter entry
Control-I - show this menu, plus information about the current parameter

Press Control-F when you are finished. The Opsware Installer will perform
a final validation check and write out a response file that will be
used to install the Opsware components.

Parameter 1 of 2 (truth.oaPwd)Please enter the password for the opsware_admin
user. This is the password used to connect to the Oracle database.:
<enter_password>
Validating... OK.

Parameter 2 of 2 (truth.dcNm)Please enter the short name of the facility where
Opsware Installer is being run (no spaces): <enter_data_center_name>
Validating... OK.
```

You are then asked to supply the name of the response file in which to store your interview responses. The default is `oiresponse.oracle_sas`:

```
Name of response file to write [/usr/tmp/oiresponse.oracle_sas]:
Response file written to /usr/tmp/oiresponse.oracle_sas.

Would you like to continue the installation using this response file? (y/n): y
```

You will then see this screen:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Oracle RDBMS for SAS
Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select 1 and press c to begin the database installation. When the installation is complete, a message to that effect is displayed. You can now continue with the SA installation.

HP-Supplied RDBMS Configuration Details

The HP-supplied Oracle database sets certain system and Oracle environment variables.

When you install the HP-supplied Oracle RDBMS using the SA Installer Oracle installation option, the installer:

- Checks that all requirements are met on the host server (see Hardware Requirements on page 3 and Operating System Requirements on page 6).
- Sets certain kernel parameters to required values. For details about these parameter changes, see Allowed Manual Kernel Parameter Changes on page 14.
- Creates the Unix user `oracle` locally in `/etc/passwd`.
- Creates the Unix groups `dba` and `oinstall` locally in `/etc/group`.
- Sets the `$ORACLE_HOME` environment variable to the following directory:

  `/u01/app/oracle/product/11.2.0/db_1`
- Sets the `$ORACLE_SID` environment variable to `truth`.
- Creates an Oracle instance with the required `init.ora` parameters.
- Creates the tablespaces and data and index files under the following directories:

  ```
  /u01/oradata/truth
  /u02/oradata/truth
  /u03/oradata/truth
  /u04/oradata/truth
  ```
- The system administrator can configure the `/u01`, `/u02`, `/u03`, `/u04` directories before installing the Oracle RDBMS software.
- Gets the service name (TNS name) from the SA Installer interview (`truth.servicename` prompt) and inserts it into the `tnsnames.ora` file in `$ORACLE_HOME/network/admin` and `/var/opt/oracle`. The SA Installer changes the value of the `host` parameter in `tsnames.ora` to the value returned by the Unix `hostname` command.
- In the `/$ORACLE_HOME/network/admin/listener.ora` file, changes the value of the `host` parameter to the value returned by the Unix `hostname` command.

  The listener is password protected and OS authenticated. (The default password is `opsware`.) By default, it listens on port 1521.
- Creates the `/etc/init.d/opsware-oracle` script, which you can use to start up and shut down the database and listener.

  This script is linked to corresponding scripts in the `/etc/rc*.d` directories.
- Creates the user `opsware_admin` with the required privileges.
- After installation is complete, you can examine the logs that are created here:

  `/var/log/opsware/install_opsware`

Allowed Manual Kernel Parameter Changes

If you manually install the Oracle database, or use an existing database, you must insure that all kernal parameter values are specified correctly for your environment but also within the limitations required by SA.

Modifiable Kernel Parameter Values for Linux

This section identifies the kernel parameters you can change for Red Hat Linux Enterprise Server AS 4 x86_64, Red Hat Linux Enterprise Server AS 5 x86_64, and SUSE Linux Enterprise Server 10 x86_64.

You can change values for the following parameters in `/etc/sysctl.conf`:

```
#SA Oracle parameters begin
fs.aio-max-nr=1048576
fs.file-max=6815744
kernel.shmmax=2147483648
kernel.shmall=2097152
kernel.shmmni=4096
kernel.sem=250 32000 100 128
net.core.rmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_default=262144
net.core.wmem_max=1048586
net.ipv4.ip_local_port_range=9000 65500
net.ipv4.tcp_wmem=262144 262144 262144
net.ipv4.tcp_rmem=4194304 4194304 4194304
#SA Oracle parameters end
```

You can change values for the following parameters in `/etc/security/limits.conf`:

```
#SA Oracle parameters begin
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft stack 10240
oracle hard stack 32768
#SA Oracle parameters end
```

You can change values for the following parameters in `/etc/pam.d/login`:

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

You can change values for the following parameters in `/etc/fstab`:

```
shmfs /dev/shm tmpfs size=4g 0
```

You can change values for the following parameters in `/etc/selinux/config`:

```
#SAS Oracle parameters begin
SELINUX=disabled
#SAS Oracle parameters end
```

Additional Modifiable SUSE Kernel Parameter Values

This section identifies additional required settings for SUSE Linux Enterprise Server 10 x86_64 when running Oracle 11g:

- Enter the following command to cause the system to read the `/etc/sysctl.conf` file when it restarts:

  ```
  # /sbin/chkconfig boot.sysctl on
  ```

- You must enter the GID of the `oinstall` group as the value for the parameter `/proc/sys/vm/hugetlb_shm_group`. Doing this grants members of `oinstall` a group permission to create shared memory segments.

  For example, where the oinstall group GID is 501:

  ```
  # echo 501 > /proc/sys/vm/hugetlb_shm_group
  ```

  After running this command, use vi to add the following text to /etc/sysctl.conf, and enable the boot.sysctl script to run on system restart:

  ```
  vm.hugetlb_shm_group=501
  ```

Only one group can be defined as the `vm.hugetlb_shm_group`.

How to Change Kernel Parameter Values for Solaris 10

To change a kernel parameter for Solaris 10, perform the following steps:

1. Enter set `noexec_user_stack=1` in `/etc/system`.
2. Run the following commands:

   ```
   projadd -U oracle -K "project.max-shm- memory=(priv,4GB,deny)" user.oracle

   projmod -s -K "project.max-sem-ids=(priv,100,deny)" user.oracle
   projmod -s -K "process.max-sem-nsems=(priv,256,deny)" user.oracle

   projmod -s -K "project.max-shm-ids=(priv,100,deny)" user.oracle
   projmod -s -K "process.max-file-descriptor=(priv,65536,deny)" user.oracle

   echo "oracle::::project=user.oracle"  >> /etc/user_attr
   ```

3. Use the `vi` editor for `/etc/project` and `/etc/user_attr` to verify the changes made in step 2.
4. Run the following commands:

   ```
   ndd -set /dev/udp udp_smallest_anon_port 9000
   ndd -set /dev/udp udp_largest_anon_port 65500
   ndd -set /dev/tcp tcp_smallest_anon_port 9000
   ndd -set /dev/tcp tcp_largest_anon_port 65500
   ```

Pre-Installation Tasks (Oracle Universal Installer)

If you plan to install the HP-supplied Oracle RDBMS software and database using the SA Installer, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle databases installed using the Oracle Universal Installer and are required for compatibility with SA.

This section discusses the prerequisites for an installation of the Oracle RDBMS using the Oracle Universal Installer for use with SA. For more detailed information about installing Oracle, see the *Oracle Installation Guide* for your operating system. Each operating system and Oracle version has a different guide. The Oracle documentation is available at the following URL:

```
http://www.oracle.com/technology/documentation/index.html
```

Before installing the Oracle RDBMS software, perform the following steps:

The sample files referenced in these steps can be obtained from your HP Support representative. See Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 18.

1. Verify that the server has the hardware and software listed in Hardware Requirements on page 3 and Operating System Requirements on page 6.
2. Obtain the sample files and unzip them.
3. Manually set kernel parameters.

   For information about modifying kernal parameters, see Allowed Manual Kernel Parameter Changes on page 14.
4. Create the required Unix users and groups by running the following commands. (If you use a directory different than `/u01/app/oracle`, modify the commands accordingly):

   ```
   mkdir -p /u01/app/oracle
   groupadd oinstall
   groupadd dba
   groupadd dboper
   useradd -g oinstall -G dba \
     -d /u01/app/oracle -s /usr/bin/sh oracle
   chown oracle:oinstall /u01/app/oracle
   Set the environment variables for the oracle user.
   ```

   The easiest way to set these variables is by obtaining and editing the following sample files:

   ```
   bash_profile
   profile
   ```

   Now you should be ready to install the Oracle RDBMS. For instructions, see the *Oracle Installation Guide* for your operating system.

Baseline Data Installation

The following steps are required for Red Hat Enterprise Linux AS 4 x86_64, Red Hat Enterprise Linux AS 5 x86_64, Sun Solaris x86_64, and SUSE Enterprise Linux 10 x86_64 because, during a Model Repository fresh install, baseline data is not inserted completely. Oracle does not insert some of the baseline data in `role_classes` and other tables and there are no errors, failures or trace files generated by Oracle. This is a silent failure and an intermittent problem. The Model Repository installs successfully because there are no error messages returned from Oracle, but later the Data Access Engine (Spin) install fails due to missing baseline data.

Before installing the Model Repository, run the following commands:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> ALTER SYSTEM SET EVENT='12099 trace name context forever,
level 1' SCOPE=SPFILE;
SQL> Shutdown immediate;
SQL> Startup
SQL> Exit
```

You can now run the SA Installer and install the Model Repository.

Oracle XDB Component Installation Requirements

During a Multimaster installation, SA exports the database using Oracle's Export utility. Due to an Oracle bug, the Export utility fails if the XDB component is installed *and* NLS_LENGTH_SEMANTICS=CHAR (as required for SA). To avoid this error, you must install Oracle excluding the XDB component.

Manually Creating the Oracle Database

If you will install the HP-supplied Oracle RDBMS software and database, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle installed using the Oracle Universal installer and are required for compatibility with SA.

When the SA Installer installs the Oracle RDBMS software and database, it runs certain scripts that do configuration tasks, create users, set password and parameter values, etc.

When you manually install the Oracle RDBMS, certain of these scripts must be run, others are optional (you can manually make the required modifications to Oracle settings for SA or you can run the script `truth.sh` which will automatically run all the required scripts in the correct order).

The SQL scripts that must be run or edited are:

- `CreateDB.sql`
- `CreateDBFiles.sql`
- `CreateUserOpsware_Admin.sql`
- `init.ora`

Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files

The following describes the script files, SQL scrips, and configuration files that are run or edited when you run the truth.sh script. These files are available from your HP Support representative.

- **truth.sh**: A shell script that creates directories and then launches the `truth.sql` script. Running this script causes all the tasks performed in this list to be performed automatically.
- **truth.sql**: Prompts for passwords of the SYS and SYSTEM users and then launches the remainder of the SQL scripts in this list.
- **CreateDB.sql**: Creates a database with the UTF8 character set (as required by SA), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.

The database must have the character set UTF8 available.

- **CreateDBFiles.sql**: Creates the following tablespaces that are required by SA:

  LCREP_DATA
  LCREP_INDX
  TRUTH_DATA
  TRUTH_INDX
  AAA_DATA
  AAA_INDX
  AUDIT_DATA
  AUDIT_INDX
  STRG_DATA
  STRG_INDX

  See Tablespace Sizes on page 35 in Chapter 2 for information about for additional tablespace sizing information.
- **CreateDBCatalog.sql**: Runs Oracle scripts to create data system catalog objects.
- **JServer.sql**: Sets up the Oracle Java environment.
- **CreateAdditionalDBFiles.sql**: Adds data and index files to certain tablespaces and allocates additional disk space. This script is optional, but recommended.
- **CreateUserOpsware_Admin.sql**: Creates the `opsware_admin` database user and grants permissions (privileges) to this user (required by SA).
- **postDBCreation.sql:** Creates the `spfile` from the `pfile` (parameter file).
- **init.ora**: Contains initialization parameters for the database. Certain parameter values are required by SA. See Required and Suggested Parameters for init.ora on page 20.
- **tnsnames.ora**: Enables resolution of database names used internally by SA.
- **listener.ora**: Contains configuration parameters for the listener. SA by default listens on port 1521. You can change the default port during installation or by editing the tsnames.ora file.

The SA-supplied Oracle 11.2.02 database has a new `listener.ora` parameter:

```
SUBSCRIBE_FOR_NODE_DOWN_EVENT_LISTENER=
```

Default is OFF. This parameter must be set to OFF for non-RAC installations. For more information about this parameter, see the Oracle documents IDs 372959.1 and 437598.1.

- **bash_profile** or **profile**: Sets environment variables and sets shell limits for the `oracle` Unix user.
- **opsware-oracle**: A script residing in `/etc/init.d` that starts up and shuts down the database and listener.

The `/etc/init.d/opsware-sas` start script, which starts and stops the SA components, does not start and stop the database and listener. For more information on the `opsware-sas` start script, see "Start Script for SA" in the *SA Administration Guide*.

Files that Must be Run or Edited for a Manual Oracle Installation

Even if you plan to configure your Oracle installation manually and not run `truth.sh` to automatically configure the Oracle installation, the following scripts must be run and `init.ora` must have certain parameter values edited or added as shown in Required and Suggested Parameters for init.ora on page 20.

- **CreateDB.sql**: Creates a database with the UTF8 character set (as required by SA), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.
- **CreateDBFiles.sql**: Creates the following tablespaces that are required by SA:

  ```
  LCREP_DATA
  LCREP_INDX
  TRUTH_DATA
  TRUTH_INDX
  AAA_DATA
  AAA_INDX
  AUDIT_DATA
  AUDIT_INDX
  STRG_DATA
  STRG_INDX
  ```

  See Tablespace Sizes on page 35 in Chapter 2 for information about for additional tablespace sizing information.
- **CreateUserOpsware_Admin.sql**: Creates the `opsware_admin` database user and grants permissions (privileges) to this user (required by SA). If you plan to create the opsware_admin without running this script, see below.
- **init.ora**: Must be edited as shown in Required and Suggested Parameters for init.ora on page 20.

Create the User Opsware_Admin

The following explains how to create this user:

To create the `opsware_admin` user after a manual Oracle installation, log in to SQL*Plus and enter the following:

```
# Su - oracle
# Sqlplus "/ as sysdba"


create user opsware_admin identified by opsware_admin
        default tablespace truth_data
        temporary tablespace temp
        quota unlimited on truth_data;

grant alter session to opsware_admin with admin option;
grant create procedure to opsware_admin with admin option;
grant create public synonym to opsware_admin with admin option;
grant create sequence to opsware_admin with admin option;
grant create session to opsware_admin with admin option;
grant create table to opsware_admin with admin option;
grant create trigger to opsware_admin with admin option;
grant create type to opsware_admin with admin option;
grant create view to opsware_admin with admin option;
grant delete any table to opsware_admin with admin option;
grant drop public synonym to opsware_admin with admin option;
grant select any table to opsware_admin with admin option;
grant select_catalog_role to opsware_admin with admin option;
grant query rewrite to opsware_admin with admin option;
grant restricted session to opsware_admin with admin option;

grant execute on dbms_utility to opsware_admin with grant option;
grant analyze any to opsware_admin;
grant insert, update, delete, select on sys.aux_stats$ to opsware_admin;
grant gather_system_statistics to opsware_admin;
grant create job to opsware_admin;
grant create any directory to opsware_admin;
grant drop any directory to opsware_admin;

grant alter system to opsware_admin;
grant create role to opsware_admin;
grant create user to opsware_admin;
grant alter user to opsware_admin;
grant drop user to opsware_admin;
grant create profile to opsware_admin;
grant alter profile to opsware_admin;
grant drop profile to opsware_admin;
```

Required and Suggested Parameters for init.ora

All Oracle 11g

```
db_file_multiblock_read_count := suggested to be >= 16
job_queue_processes := required to be >= 1000
remote_login_passwordfile := required to be = EXCLUSIVE
optimizer_mode := 'required to be = ALL_ROWS
recyclebin := required to be = OFF
event := required to be = 12099 trace name context forever, level 1
_complex_view_merging := required to be = FALSE
```

Oracle 11.2.0

```
deferred_segment_creation := required to be = FALSE
compatible := required to be >= 11.2.0
open_cursors := required to be >= 1500 (
cursor_sharing := required be = FORCE
memory_target := required to be >= 1879048192 (1.75GB)
log_buffer := required to be >= 5242880
db_file_multiblock_read_count := required to be = 16
```

Oracle 11.1.0

```
compatible := >= 11.1.0
compatible := required to be >= 11.1.0
open_cursors := required to be >= 1000
cursor_sharing := should be = SIMILAR (preferred) or
  EXACT (recommended only if you encounter Oracle Bug No. 3102053)
memory_target := required to be >= 1694498816 (1616M)
log_buffer := required to be >= 1048576
```

Oracle 10g

```
cursor_sharing := should be = SIMILAR (preferred) or
  EXACT (recommended only if you encounter Oracle Bug No. 3102053)
db_file_multiblock_read_count := 'suggested to be >= 16
log_buffer := required to be >= 1048576
open_cursors := required to be >= 300
sga_max_size := required to be >= 1073741824 (1G)
db_cache_size := required to be >= 629145600 (600M)
shared_pool_size := required to be >= 262144000 (250M)
java_pool_size := required to be >= 52428800 (50M)
large_pool_size := required to be >= 52428800 (50M)
pga_aggregate_target := required to be >= 104857600 (100M)
workarea_size_policy := required to be = AUTO
job_queue_processes := required to be >= 10
sessions := required to be >= 1152
remote_login_passwordfile := required to be = EXCLUSIVE
optimizer_mode := required to be = ALL_ROWS
recyclebin := required to be = OFF
event := required to be = 12099 trace name context forever, level 1
_complex_view_merging := required to be = FALSE
compatible := required to be >= 10.2.0
```

All Oracle Versions:

```
audit_trail := required to be = none
db_block_size := required to be >= 8192
processes := required to be >= 1024
undo_management := should be = AUTO
session_cached_cursors := required to be >= 50
undo_tablespace := should be = UNDO or other UNDO tablespace
optimizer_mode:= required to be = all_rows
optimizer_index_cost_adj := required to be = 20
optimizer_index_caching := required to be = 80
```

```
nls_length_semantics := required to be = CHAR
nls_sort := required to be = GENERIC_M
_complex_view_merging := required to be = false
event := required to be = 12099 trace name context forever, level 1'
remote_login_passwordfile := required to be = EXCLUSIVE
```

Creating the Database using the HP-Supplied Scripts

To create the Oracle database using the HP-supplied scripts, perform the following steps:

1. Obtain the database creation scripts from your HP support representative.
2. Configure the scripts. See Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 18
3. Log in to the server as the Unix user `oracle`.
4. Copy the HP-supplied `init.ora` file to the following directory:

   `$ORACLE_BASE/admin/truth/create`
5. Examine the SQL scripts that you will run in **step 7**. If necessary, edit the scripts to conform to your organization's policies.
6. Log on to the server as the `oracle` user and change the mode of the HP-supplied `truth.sh` script:

   `chmod 755 truth.sh`
7. Launch the SQL scripts that create the database by running the `truth.sh` script:

   `./truth.sh`
8. After the scripts launched by `truth.sh` complete, check the log files in the following directory for errors:

`/u01/app/oracle/admin/truth/scripts/*.log`

Post-Oracle Installation Tasks

If you will install the HP-supplied Oracle database, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle databases installed using the Oracle Universal installer and are required for compatibility with SA.

After creating the database, but before installing the Model Repository with the SA Installer, perform the following steps:

1. Create the `tnsnames.ora` file in the following directory:

   `$ORACLE_HOME/network/admin`

   Verify that the file conforms to the rules listed in tnsnames.ora File Requirements on page 23.
2. If it does not exist, create the following directory:

   `mkdir -p /var/opt/oracle`
3. Create the following symbolic link:

```
ln -s $ORACLE_HOME/network/admin/tnsnames.ora \
/var/opt/oracle/tnsnames.ora
```

4. Make sure that the oracle Unix user has read-write permission on the `tnsnames.ora` file.
5. For Red Hat Enterprise Linux 4 or 5 AS, create another symbolic link:

   ```
   ln -s /etc/oratab /var/opt/oracle/oratab
   ```

6. Copy the sample `opsware-oracle` script to `/etc/init.d/`.
7. Link `/etc/init.d/opsware-oracle` to corresponding scripts in the `/etc/rc*` directories. For example:

   ```
   ln -s /etc/init.d/opsware-oracle \
         /etc/rc0.d/K02opsware-oracle
   ln -s /etc/init.d/opsware-oracle \
         /etc/rc1.d/K02opsware-oracle
   ln -s /etc/init.d/opsware-oracle \
         /etc/rc2.d/S60opsware-oracle
   ln -s /etc/init.d/opsware-oracle \
         /etc/rcS.d/K02opsware-oracle
   ```

8. Copy the sample `listener.ora` file to `$ORACLE_HOME/network/admin`.
9. In `listener.ora`, change the value of the `host` parameter to the host name of server running the database.

Location of Additional Oracle Data Files

If you want to add data files to a database created with the SA Installer, you can add them to the following directories:

```
/u01/oradata/truth
/u02/oradata/truth
/u03/oradata/truth
/u04/oradata/truth
```

tnsnames.ora File Requirements

The `tnsnames.ora` file enables resolution of database names used internally by the core components. SA has the following requirements for the `tnsnames.ora` file:

- The file must reside in the following location:

  `/var/opt/oracle/tnsnames.ora`

- If the core is installed across multiple servers, a copy of the file must reside on the servers hosting the following components:
  - Model Repository
  - Infrastructure Component bundle (required by the Data Access Engine, Model Repository Multimaster Component, Software Repository Store)
  - Slice Component bundle (required by the Command Center, Web Services Data Access Engine, Global File System)
- For a core installed on multiple servers, the directory path of the `tnsnames.ora` file must be the same on each server.

- In a Single Core installation, the tnsnames.ora file must contain an entry for the Model Repository, as in the following example:

  ```
  truth = (DESCRIPTION= (ADDRESS=(HOST=magenta.example.com)(PORT=1521)
  (PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
  ```

tnsnames.ora: Multimaster Mesh Requirements

In a Multimaster Mesh, the tnsnames.ora file must be set up for a Source Core and a Destination Core using the following guidelines.

Source Core

The tnsnames.ora file must contain an entry for its own Model Repository. The port number must be set to the port that you have designated that the Oracle listener process use, such as 1521 (default), 1526, and so on.

The tnsnames.ora file must also contain an entry that specifies the Source Core Management Gateway. This port is used by the Data Access Engine for Multimaster traffic. The port number is derived from the following formula: (20000) + (facility ID of the Destination Core).

Example: In the following example, the TNS service name of the Source Core is orange_truth, which runs on the host orange.example.com. The TNS name of the Destination Core is cyan_truth, which has a facility ID of 556. Note that the entry for cyan_truth specifies orange.example.com, which is the host running the Source Core's Management Gateway.

```
orange_truth=(DESCRIPTION=(ADDRESS=(HOST=orange.example.com)(PORT=1521)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
cyan_truth=(DESCRIPTION=(ADDRESS=(HOST=orange.example.com)(PORT=20556)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Destination Core

The tnsnames.ora file must contain an entry for its own Model Repository. The port number must be set to the port that you have designated that the Oracle listener process use, such as 1521 (default), 1526, and so on. The tnsnames.ora file does not require any entries for other cores in the mesh.

Example: In the following example, the TNS service name of the Destination Core is cyan_truth, and the core runs on the host, cyan.example.com.

```
cyan_truth=(DESCRIPTION=(ADDRESS=(HOST=cyan.example.com)(PORT=1521)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Requirements for Enabling Oracle Daylight Saving Time (DST)

To enable Daylight Saving Time for the Oracle database, you must apply database tier patches. To apply these patches, perform the following steps:

1. Verify that your database is running on Oracle 10g or higher. If you are on an earlier database release, use one of the following MetaLink Notes to upgrade your database:

   10gR2 Database: MetaLink Note 362203.1

2. Use MetaLink Note 359145.1 to apply Oracle Database time zone fixes specific to your database version.

3 Use MetaLink Note 359145.1 to apply time zone fixes to the Oracle Java Virtual Machine (JVM) in the Oracle Database specific to your E-Business Suite database version.

Installing the Model Repository Database on a Remote Server

To install or upgrade the Model Repository Oracle database on a remote server, perform the following steps:

1 Perform the following tasks on the server on which you will run the SA Installer:

a Install the Oracle client software. The software must be owned by the OS user `oracle`. For a Multimaster install, you need the Oracle full client; for a Single Core (standalone) installation, the Oracle Instant client will suffice. The client software must be owned by the OS user `oracle`. Install the Oracle client in a location like

`/opt/opsware/oracle_client_model_repo`.

The Oracle full client must be the same version as your Oracle database.

— You can copy the Oracle instant client from an existing core. The Oracle Instant client is by default installed under `/opt/opsware/oracle_client` on the Web Services Data Access Engine (`twist`) host.

— Otherwise, download the appropriate Oracle full client from

http://www.oracle.com/technology/software/products/database/index.html

b (For an Oracle Full client install, this step is not necessary.) Copy the database server's `$ORACLE_HOME/jdbc/lib/classes12.zip` to the client's Oracle home, for example:

```
# scp oracle@truth:$ORACLE_HOME/jdbc/lib/classes12.zip \
/opt/opsware/oracle_client_model_repo/jdbc/lib
```

c Copy the database server's `/var/opt/oracle/tnsnames.ora` file to the client host and ensure that the hostname in the file resolves properly.

d Ensure that the SA Installer response file has the correct path to the client `tnsnames.ora` file (`%truth.tnsdir`), oracle client home (`%db.orahome`), database server name/IP (`%db.host`), listener port (`%db.port`), SA Installer machines subdomain (`%truth.dcSubDom`), etc. Based on the above steps your parameter values will be:

– `%truth.tnsdir=/var/opt/oracle`

– `%db.orahome=/opt/opsware/oracle_client_model_repo`

– `%db.port=1521`

– `%truth.dcSubDom=prod.example.com`

– `db.host=192.168.9.99 (server where the Oracle RDBMS is installed)`

e Ensure that the COMPATIBLE parameter is set correctly and that it matches the database version. For example, for database software that is version 11.2.0.1 ensure that COMPATIBLE=11.2.0.1. SA uses Oracle's Export Data Pump and Import Data Pump utilities during secondary core creation. These utilities require on the COMPATIBLE parameter.

2 Set up the following on the Model Repository host (Oracle database server):

a Log in as user `oracle`

b cd `$ORACLE_HOME/network/admin`

c Ensure that the listener is started with the command

```
lsnrctl start <your_listener_name>
```

Oracle RAC Support

SA supports Oracle Real Application Clusters (RAC).

Oracle RAC support requires a new installation of both Oracle and SA. Therefore, in order to enable Oracle RAC support in SA, you must first install SA and Oracle 10.2.0.4 or 11.1.0.7 configured as described in the following sections.

Supported Oracle Versions

SA support for Oracle RAC includes these Oracle Enterprise Edition versions:

- 10.2.0.4
- 11.1.0.7

Supported Operating Systems

- Red Hat Enterprise Linux AS 4 x86_64
- Red Hat Enterprise Linux AS 5 x86_64

Set up the Oracle RAC Database / Instances

SA supports any valid Oracle RAC configuration, such as any number of nodes, ASM or regular disks, and so on.

However, SA requires that the Oracle database be configured for use with SA. You will require your Oracle DBA's help to configure the Oracle RAC/instances, the required initialization parameters, the required tablespaces, the `opsware_admin` database user, and the `listener.ora` and `tnsnames.ora` files.

Create the Database with the Required Initialization Parameters

Before installing Oracle, the following scripts must be run and `init.ora` must have certain parameter values edited or added as shown in Required and Suggested Parameters for init.ora on page 20.

1 Create a database with the UTF8 character set (as required by SA), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.

Create the Required Tablespaces

2 Create the following tablespaces that are required by SA:

```
LCREP_DATA
LCREP_INDX
TRUTH_DATA
TRUTH_INDX
AAA_DATA
AAA_INDX
AUDIT_DATA
AUDIT_INDX
STRG_DATA
STRG_INDX
```

See “Tablespace Sizes” in the *SA Simple/Advanced Installation Guide* for additional tablespace sizing information.

Required and Suggested Parameters for init.ora

3 The file `init.ora` must be edited as shown in Required and Suggested Parameters for init.ora on page 20.

Create the User opsware_admin

4 You can use the script, `CreateUserOpsware_Admin.sql`, to create the `opsware_admin` database user and grant permissions (privileges) to the user (required by SA) or create the user manually.

If you plan to create the `opsware_admin` user manually, follow the procedure below:

Manual Creation of the User Opsware_Admin

To create the `opsware_admin` user after a manual Oracle installation, log in to SQL*Plus and enter the following:

```
# Su - oracle
# Sqlplus "/ as sysdba"


SQL>create user opsware_admin identified by opsware_admin
        default tablespace truth_data
        temporary tablespace temp
        quota unlimited on truth_data;

grant alter session to opsware_admin with admin option;
grant create procedure to opsware_admin with admin option;
grant create public synonym to opsware_admin with admin option;
grant create sequence to opsware_admin with admin option;
grant create session to opsware_admin with admin option;
```

```
grant create table to opsware_admin with admin option;
grant create trigger to opsware_admin with admin option;
grant create type to opsware_admin with admin option;
grant create view to opsware_admin with admin option;
grant delete any table to opsware_admin with admin option;
grant drop public synonym to opsware_admin with admin option;
grant select any table to opsware_admin with admin option;
grant select_catalog_role to opsware_admin with admin option;
grant query rewrite to opsware_admin with admin option;
grant restricted session to opsware_admin with admin option;

grant execute on dbms_utility to opsware_admin with grant option;
grant analyze any to opsware_admin;
grant insert, update, delete, select on sys.aux_stats$ to opsware_admin;
grant gather_system_statistics to opsware_admin;
grant create job to opsware_admin;
grant create any directory to opsware_admin;
grant drop any directory to opsware_admin;

grant alter system to opsware_admin;
grant create role to opsware_admin;
grant create user to opsware_admin;
grant alter user to opsware_admin;
grant drop user to opsware_admin;
grant create profile to opsware_admin;
grant alter profile to opsware_admin;
grant drop profile to opsware_admin;
```

Installing the Model Repository

In most production environments with Oracle RAC, the Model Repository installation can be done from any SA server. The database server or RAC nodes in this case are considered to be remote.

The examples used in the following sections assume this configuration:

Two (active-active) Node RAC environment:

```
# Public Network
192.168.173.210 rac1pub rac1pub.dev.opsware.com (instance_name=truth1,
db name=truth)
192.168.173.211 rac2pub rac2pub.dev.opsware.com (instance_name=truth2,
db name=truth)

# Private network
172.16.1.100 rac1prv rac1prv.dev.opsware.com
172.16.1.101 rac2prv rac2prv.dev.opsware.com

# Public Virtual IP (VIP)
192.168.173.212 rac1-vip rac1-vip.dev.opsware.com
192.168.173.213 rac2-vip rac2-vip.dev.opsware.com

SA server:
192.168.173.214 rac1sa.dev.opsware.com
```

Model Repository Installation on a Remote Database (truth) RAC Server

In an Oracle RAC environment, only one of the RAC nodes is used during the SA installation/upgrade process. The SA Installer connects to only one Oracle RAC instance to install/modify the Model Repository. During the regular SA operations, all RAC nodes are used.

Perform the following tasks on the SA server on which you will run the SA Installer, for example `rac1sa.dev.opsware.com`.

Model Repository Hostname Resolution

1. On the server where you will run the SA Installer, ensure that the Model Repository hostname `truth` resolves to the remote database server, not to the server on which you will be running the SA Installer:

   In `/etc/hosts`, enter the public IP address of one of the RAC nodes/instances. For example the `/etc/hosts` file on `rac1sa.dev.opsware.com` would have the following entry:

   ```
   192.168.173.210    truth     rac1pub     rac1pub.dev.opsware.com
   ```

Install the Oracle 11g Full Client on the SA server

1. The SA Installer will use the Oracle Full Client to connect to the SA server and install the Model Repository. Below are sample commands for installing the Oracle full client.

   Create user `oracle` for the Oracle Full Client installation:

   ```
   root@rac1sa ~]# mkdir -p /u01/app/oracle
   root@rac1sa ~]# mkdir -p /u01/app/oraInventory
   root@rac1sa ~]# groupadd oinstall
   root@rac1sa ~]# groupadd dba
   root@rac1sa ~]# useradd -c "Oracle Client software owner" -g oinstall -G
   dba -d /u01/app/oracle -s /bin/bash oracle
   root@rac1sa ~]# chown -R oracle:oinstall /u01/app
   root@rac1sa ~]# chmod -R 775 /u01/app
   root@rac1sa ~]#passwd oracle  (change oracle user password )
   ```

2. Create the .bash_profile file

   In `/u01/app/oracle` create the `.bash_profile` file.

Temporarily comment out `ORACLE_HOME` and `ORACLE_PATH`. You will uncomment these entries after the Oracle client installation is complete.

Sample .bash_profile file

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi

# User specific environment and startup programs
PATH=$PATH:$HOME/bin
export PATH

#SA-OracleRAC parameters begin
#unset USERNAME
export ORACLE_BASE=/u01/app/oracle
```

```
#export ORACLE_HOME=$ORACLE_BASE/product/11.1.0/client_1
#PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$PATH
export PATH

if [ -t ]; then
stty intr ^C
fi

umask 022
#SA-OracleRAC parameters end
```

3 Install the Oracle Full Client

Install the Oracle Full Client as described in your Oracle documentation. You can create a share to access the Oracle Full Client binaries.

4 Set Up Terminals

You will need two X window terminals to install the Oracle Full Client:

Terminal 1: log in as root and enter the commands:

```
Terminal 1> xhost +
Terminal 2: ssh –X oracle@<new_oracle_full_client_host>
```

5 Start Oracle Full Client installation

From Terminal 2 run the Oracle Universal Installer (OUI) installer. The Oracle Full Client is installed in:

`/u01/app/oracle/product/11.1.0/client_1`

6 Run the Oracle Universal Installer to install Oracle Full Client. The directories in this example assume an Oracle 11g Full Client on Linux.

a `cd /location_of_oracle_full_client`

b `/runInstaller`

c At the **Welcome** Screen, click **Next**.

d Specify the Inventory Directory and Credentials (`/u01/app/oraInventory` and `/u01/app/oinstall`)

e For **Select Installation Type**, choose Administrator, click **Next**.

f For `ORACLE_BASE` select: `/u01/app/oracle`, click **Next**.

g The Oracle Universal Installer performs some checks. If the checks are not successful, fix the issue and re-run this step. If the checks are successful the click on 'Next'

h Oracle OUI will list of products that will be installed. Click on 'Install'

i OUI will show the progress bar when installing

j On the 'Welcome to Oracle Net Configuration Assistant' window click on 'Next'

k Click on 'Finish' once the installation is complete.

l The following two configuration scripts need to be executed as "root" upon installation being complete:

- `/u01/app/oraInventory/orainstRoot.sh`
- `/u01/app/oracle/product/11.1.0/client_1/root.sh`

7 Verify that the `.bash_profile` file for user `oracle` is correct.

8 Uncomment $ORACLE_HOME and $ORACLE_PATH.

Making changes to tnsnames.ora on an SA server

By default the tnsnames.ora file is located in /var/opt/oracle.

1 Login as root on the SA Server.

2 Enter the command:

```
mkdir -p /var/opt/oracle
```

3 Copy tnsnames.ora from the remote database server to the directory you created above. For the RAC environment, copy it from RAC Node 1 (for example, rac1pub.dev.opsware.com).

To accommodate the remote Model Repository installation process, two sets of tnsnames.ora are required on the SA server.

- tnsnames.ora-install_upgrade – this copy of tnsnames.ora is used during SA installation/upgrade. The file can be renamed.
- tnsnames.ora-operational – this copy of tnames.ora is used during normal SA operation. The file can be renamed.

  You can use softlinks to point tnsnames.ora to either tnsnames.ora-install_upgrade or tnsnames.ora-operational.

  ```
  ln –s tnsnames.ora-install_upgrade tnsnames.ora
  ```

tnsnames.ora-install_upgrade sample file

Make a note of the text that is in BOLD letters. The tnames.ora file should contain the SID and not the service name. These examples have TRUTH as the truth.servicename. Ensure that the HOST references the same server as the truth entry in /etc/hosts file. truth.servicename is case sensitive.

```
# Generated by Oracle configuration tools.

TRUTH =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac1pub.dev.opsware.com)(PORT = 1521))
    (CONNECT_DATA =
            (SID = truth1)
    )
  )

LISTENER_TRUTH =
  (ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP)(HOST = rac1pub.dev.opsware.com)(PORT = 1521))
  (CONNECT_DATA =
      (SID = truth1)
    )
  )
```

Use softlinks to link the file to tnsnames.ora.ora file. Do this before you start the SA Model Repository installation or upgrade

```
ln –s tnsnames.ora-install_upgrade tnsnames.ora
```

During installation the SA Installer adds a SA Gateway entry into tnsnames.ora (linked to tnsnames.ora.install-upgrade) file on the First SA Core. When installation completes, copy this entry into the tnsname.ora.operational file. If this entry is not present in

tnsname.ora.operational, Multimaster Mesh transactions will not flow. Below is a sample gateway entry from tnsnames.ora:

```
Rac2sa_truth=(DESCRIPTION=(ADDRESS=(HOST=192.168.173.214)(PORT=20002)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Testing the connection from the SA host to the database

Before starting the Model Repository installation/upgrade, you can perform the following tests to verify that your tnsnames.ora fileis configured correctly and if the SA Installer can connect to the database in restricted mode.

1. Verify that the SA server's /var/opt/oracle/tnsnames.ora file is configured correctly as described in *Making changes to tnsnames.ora on an SA Server*.
2. On the SA server:
   a. Login as oracle or root or su – twist/spin – if these users exist
   b. export ORACLE_HOME=/u01/app/oracle/product/11.1.0/client_1 (or where you installed the Oracle Full Client)
   c. export LD_LIBRARY_PATH=$ORACLE_HOME/lib
   d. export TNS_ADMIN=/var/opt/oracle
   e. set $PATH $ORACLE_HOME/bin path
   f. sqlplus sys/password@truth as sysdba;
   where truth is the service_name or entry from the tnsnames.ora file
   g. connect opsware_admin/<password>@truth

If you are able to logon to the database then all files are configured correctly.

SA Installer Response File

You can now start the installation of the SA Model Repository. Ensure that you have the correct parameters values for the installation interview or that you have a previous response file.

Verify the paths to the client's tnsnames.ora file (%truth.tnsdir), oracle client home (%db.orahome), database server name/IP (%db.host), listener port (%db.port), and so on.

- %truth.tnsdir=/var/opt/oracle
- %db.orahome=/u01/app/oracle/product/11.1.0/client_1
- %db.port=1521
- %db.host=192.168.9.99 (server where the Oracle RDBMS is installed)

You can now install the SA Core as described in the *SA Simple/Advanced Installation Guide*.

Post SA installation process

After you install the SA Core, perform the following tasks in order to use all the nodes in the Oracle RAC environment.

Making changes to tnsnames.ora on the SA server

After SA install is complete, the tnsnames.ora file should point/link to the tnsnames.ora-operational file.

In an Oracle RAC environment, only one of the RAC nodes or instances is used during the installation/upgrade process. The SA Installer connects to only one Oracle instance to modify the Model Repository. During the normal SA operations, all the RAC nodes are used. To accommodate the remote truth installation process, two sets of tnsnames.ora are required on the SA server.

- `tnsnames.ora-install_upgrade` – this copy of `tnsnames.ora` is used during SA installation/upgrade. You can rename the file.
- `tnsnames.ora-operational` – this copy of `tnames.ora` is used during normal SA operation. You can rename the file.

You can use softlinks to point `tnsnames.ora` to either `tnsnames.ora-install_upgrade` or `tnsnames.ora-operational`:

```
ln –s tnsnames.ora-operational tnsnames.ora
```

tnsnames.ora-operational sample file

Make a note of the text that is in BOLD letters. This `tnsnames.ora` file is used during normal SA operation and contains the RAC parameters.

```
#This entry is for connecting to RAC virtual machines.
TRUTH =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.dev.opsware.com)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.dev.opsware.com)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = truth)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = Preconnect)
        (RETRIES = 180)
        (DELAY = 5))
    )
  )

LISTENERS_TRUTH =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.dev.opsware.com)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.dev.opsware.com)(PORT = 1521))
  )

#This entry is for connecting to node2 via service_name. This entry is
optional
TRUTH2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.dev.opsware.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = truth)
      (INSTANCE_NAME = truth2)
    )
  )
LISTENER_TRUTH2 =
   (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.dev.opsware.com)(PORT = 1521))
```

```
#This entry is for connecting to node1 via service_name. This entry is
optional
TRUTH1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.dev.opsware.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = truth)
      (INSTANCE_NAME = truth1)
    )
  )
LISTENER_TRUTH1 =
   (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.dev.opsware.com)(PORT = 1521))
```

Use softlinks to link the file to `tnsnames.ora.ora` file after SA installation is complete and you are ready to start SA in operational mode.

```
ln –s tnsnames.ora- operational tnsnames.ora
```

During installation the SA Installer adds an SA Gateway entry into `tnsnames.ora` (linked to `tnsnames.ora.install-upgrade`) file on the First SA Core. When installation is complete, copy that entry into `tnsname.ora.operational`. If this entry is not present in the `tnsname.ora.operational`, Multimaster Mesh transactions will not flow. The following is a sample gateway entry from `tnsnames.ora`:

```
Rac2sa_truth=(DESCRIPTION=(ADDRESS=(HOST=192.168.173.214)(PORT=20002)(PROTOCO
L=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Vault.conf File Changes

In an Oracle RAC environment, the `vault.conf` file must be modified after SA installation is complete. Modify `/etc/opt/opsware/vault/vault.conf` to specify the complete `tnsname` definition instead of the `SID`. For example:

- Before:

  ```
  db.sid: truth
  ```

- After:

  ```
  db.sid:(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
  rac1-vip.dev.opsware.com)(PORT = 1521)) (ADDRESS = (PROTOCOL = TCP)
  (HOST = rac2-vip.dev.opsware.com)(PORT = 1521)) (LOAD_BALANCE = yes)
  (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = truth)
  (FAILOVER_MODE = (TYPE = SELECT) (METHOD = Preconnect) (RETRIES = 180)
  (DELAY = 5))))

  db.port: 1521
  ```

- Restart the vaultdaemon:

  ```
  /etc/init.d/opsware-sas restart vaultdaemon
  ```

Changing the da.conf Configuration FIle

In SA 9.10, the Application Deployment Manager reads database connection information from the `tnsnames.ora` file.

In SA 9.0 and 9.0x, the default was `SID =Truth` unless changed by the user, for example, in `/etc/opt/opsware/da/da.conf`:

```
truth.sid=<new SID>
```

Upgrading the Model Repository

To upgrade the Model Repository in an Oracle RAC environment, follow the same procedure as Installing the Model Repository on page 28. If you are doing a remote database installation, ensure that you modify your `tnsnames.ora` on the server where the SA Installer is run. It is recommended that you test the connection as suggested in section *Testing the connection from the SA host to the database.*

Garbage Collection

The Garbage Collector (GC) is a stored procedure written in PL/SQL that runs in the database on a schedule. The GC procedures look at the `AUDIT_PARAMS` table to determine the retention period to use to delete the old data. The GC PL/SQL procedures are managed by Oracle's `dba_jobs`.

Data Retention Period

When GC runs, it looks at the values in the `AUDIT_PARAMS` table to determine what retention period to use when deleting objects.

The `AUDIT_PARAMS` table is not replicated, so there is a possibility that these retention periods may become out of synch, which can cause severe Multimaster conflict issues. You must ensure that the values in the `AUDIT_PARAMS` table are exactly the same for all the cores in a mesh.

```
# Sqlplus "/ as sysdba"
SQL> col name format a20;
SQL> col value format a20;
SQL> col AUDIT_PARAM_ID format a15;
SQL> select AUDIT_PARAM_ID, NAME, VALUE from audit_params;
```

The parameters from `AUDIT_PARAMS` table and their default values are:

```
AUDIT_PARAM_ID  NAME                 VALUE
--------------- -------------------- --------------------
2               DAYS_WAY             30         (These are the completed way sessions)
3               DAYS_CHANGE_LOG      180        (These are the server history events)
4               LAST_DATE_WAY        04-APR-09
5               LAST_DATE_CHANGE_LOG 05-NOV-08
6               DAYS_AUDIT_LOG       180        (These are the audit logs)
7               LAST_DATE_AUDIT_LOG  05-NOV-08
```

As of SA 9.0, the `DAY_TRAN` parameter that controlled retention time for transactions has been removed. To control transaction retention time, instead use the system configuration parameter `vault.garbageCollector.daysToPreserve`.

From the SAS Web Client **Navigation** panel, select **System Configuration ➤ Model Repository Multimaster Component** to change the value for the parameter (the default is 7).

Modifying the Retention Period Values

To update the data, run a SQL command similar to the following example as user LCREP:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> grant create session to lcrep;
SQL> connect lcrep/<password>
SQL> update AUDIT_PARAMS set value=30 where name = 'DAYS_AUDIT_LOG';
```

The values in the AUDIT_PARAMS table must be exactly the same for all the cores in a mesh.

Viewing GC DBA_JOBS

When the Model Repository is installed, the SA Installer sets up these jobs, which perform garbage collection.

GC jobs can be viewed by logging in to SQL*Plus and running the following SQL commands:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> col schema_user format a10
SQL> col what format a50
SQL> set line 200
SQL> select job, schema_user, last_date, this_date, next_date, broken,
what from dba_jobs where priv_user= 'GCADMIN';

JOB SCHEMA_USE LAST_DATE THIS_DATE NEXT_DATE B WHAT
--- ---------- --------- --------- --------- - ----------------------------------------
  4 GCADMIN    04-MAY-09           05-MAY-09 N WAYPURGE.GC_SESSIONS;
  5 GCADMIN    04-MAY-09           05-MAY-09 N CHANGELOGPURGE.GC_CHANGELOGS;
  6 GCADMIN    04-MAY-09           05-MAY-09 N AUDITPURGE.GC_AUDITLOGS;
  7 GCADMIN    04-MAY-09           05-MAY-09 N STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;
```

where:

WAYPURGE.GC_SESSIONS - Performs a sessions garbage collection

CHANGELOGPURGE.GC_CHANGELOGS - Performs a changelogs garbage collection

AUDITPURGE.GC_AUDITLOGS - Performs auditlogs garbage collection

STORAGEINITIATORPURGE.GC_STORAGEINITIATORS - Performs storage data garbage collection

Manually Running GC Jobs

You can run GC jobs by logging in to SQL*Plus and entering the following:

```
# Su - oracle
# Sqlplus "/ as sysdba"
```

```
SQL> grant create session to gcadmin
SQL> connect gcadmin/<password>
SQL> exec dbms_job.run(<job no>);
```

For example, this sample command runs the `waypurge_gc` job:

```
SQL> exec dbms_job.run(4);
```

Monitor the ERROR_INTERNAL_MSG Table

The garbage collection jobs write exceptions to the `truth.ERROR_INTERNAL_MSG` table. You can monitor this table for errors (checking daily is recommended). For example:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> set line 200
SQL> col ERR_ID format 999999
SQL> col ERR_USER format a8
SQL> col ERR_TABLE format a25
SQL> col ERR_TABLE_PK_ID format a10
SQL> col ERR_CODE format 9999999
SQL> col ERR_TEXT format a20
SQL> col ERR_INFO format a30

SQL> select ERROR_INTERNAL_MSG_ID ERR_ID,
      ERR_DATE,
      ERR_USER,
      ERR_TABLE,
      ERR_TABLE_PK_ID,
      ERR_CODE,
      ERR_TEXT,
      DELETE_FLG,
      ERR_INFO
     from ERROR_INTERNAL_MSG
     order by ERR_DATE;
```

Database Monitoring Strategy

Because the Model Repository is a critical component of SA, the DBA should implement a monitoring strategy. The DBA can write custom monitoring scripts or use third-party products.

This section contains example commands for monitoring the Oracle database used by the Model Repository. When issuing the commands shown in this section, you must be logged on to the server as the user `oracle`:

```
$ su - oracle
```

The SQL commands shown in this section are entered in the `sqlplus` command-line utility. To run `sqlplus`, log on as `oracle` and enter the following command:

```
$ sqlplus "/ as sysdba"
```

Verify that the Database Instances are Up and Responding

To verify that the Database Instances are up and running, perform the following steps:

1 Check to see if the Oracle processes are running by entering the following command:

```
ps -ef | grep ora_
```

This ps command should generate output similar to the following lines:

```
oracle   14674     1  0 Apr18 ?        00:00:00 ora_pmon_truth
oracle   14676     1  0 Apr18 ?        00:00:00 ora_psp0_truth
oracle   14678     1  0 Apr18 ?        00:00:00 ora_vktm_truth
oracle   14682     1  0 Apr18 ?        00:00:00 ora_gen0_truth
oracle   14684     1  0 Apr18 ?        00:00:00 ora_diag_truth
oracle   14686     1  0 Apr18 ?        00:00:00 ora_dbrm_truth
oracle   14688     1  0 Apr18 ?        00:05:57 ora_dia0_truth
oracle   14690     1  0 Apr18 ?        00:00:00 ora_mman_truth
oracle   14692     1  0 Apr18 ?        00:00:00 ora_dbw0_truth
oracle   14694     1  0 Apr18 ?        00:00:01 ora_lgwr_truth
oracle   14696     1  0 Apr18 ?        00:00:28 ora_ckpt_truth
oracle   14698     1  0 Apr18 ?        00:00:04 ora_smon_truth
oracle   14700     1  0 Apr18 ?        00:00:00 ora_reco_truth
oracle   14702     1  0 Apr18 ?        00:00:13 ora_mmon_truth
oracle   14704     1  0 Apr18 ?        00:00:13 ora_mmnl_truth
oracle   14728     1  0 Apr18 ?        00:00:00 ora_qmnc_truth
oracle   14775     1  0 Apr18 ?        00:00:01 ora_cjq0_truth
oracle   14779     1  0 Apr18 ?        00:00:00 ora_q000_truth
oracle   14781     1  0 Apr18 ?        00:00:00 ora_q001_truth
oracle   14832     1  0 Apr18 ?        00:00:00 ora_smco_truth
oracle   22619     1  0 22:38 ?        00:00:00 ora_w000_truth
```

2 Verify that the database status is ACTIVE by entering the following command in sqlplus:

```
select database_status from v$instance;
```

3 Verify that the open mode is READ WRITE by entering the following command in sqlplus:

```
select name, log_mode, open_mode from v$database;
```

Verify that the Data Files are Online

To verify that the data files are online, in SQL*Plus, enter the following commands:

```
Col file_name format a50
Col status format a10
Set line 200
Select file_id, status, bytes, file_name from dba_data_files order by
tablespace_name;
```

The status should be AVAILABLE for all the data files.

Verify That the Listener is Running

To verify that the listener is running, perform the following steps:

1 Check to see if the Oracle listener processes are running by entering the following command:

```
ps -ef | grep tns

oracle   11664     1  0 Mar22 ?        00:08:05 /u01/app/oracle/product/
11.2.0/db_1/bin/tnslsnr LISTENER -inherit
oracle   22725 22706  0 22:44 pts/2    00:00:00 grep tns
```

2 Check the status of the listener with the `lsnrctl` command:

```
lsnrctl status
```

The listener should be listening on port 1521 (default), or on the port that you have designated that the Oracle listener process use, with the TCP protocol, and should be handling the instance named truth. The `lsnrctl` command should generate output similar to the following lines:

```
. . .
Connecting to (ADDRESS=(PROTOCOL=tcp)
(HOST=per1.performance.qa.example.com)(PORT=1521))
. . .
Instance "truth", status READY, has 1 handler(s) for this service...
```

3 Test connectivity to the instance from the Data Access Engine (spin) and Web Services Data Access Engine (twist) hosts by running the `tnsping` utility:

```
tnsping truth
```

The `OK` statement displayed by the `tnsping` utility confirms that the listener is up and can connect to the instance. The `tnsping` utility should generate output similar to the following lines:

```
. . .
Used parameter files:

Used HOSTNAME adapter to resolve the alias
Attempting to contact
(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=truth.performance.qa.example.com
))(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.165.178)(PORT=1521)))
OK (0 msec)


Attempting to contact
(DESCRIPTION=(ADDRESS=(HOST=localhost)(PORT=1521)(PROTOCOL=tcp))(CONNECT_
DATA=(SERVICE_NAME=truth)))
OK (0 msec)
```

As an alternative to running the `tnsping` utility in this step, you can check the connectivity by running `sqlplus` and connecting to the database instance with the service name (TNS alias), for example:

```
sqlplus myuser/mypass@truth
```

Examine the Log Files

To examine the log files, perform the following steps:

1 Look for errors in the `alert.log` file.

For each instance, locate the `alert.log` file in the background dump destination directory:

Oracle 10g

$ORACLE_BASE/admin/<SID>/bdump

Oracle 11g

$ORACLE_BASE/diag/rdbms/<SID>/<SID>/trace/

Here is an example bdump directory for an instance with the truth SID:

Oracle 10g

/u01/app/oracle/admin/truth/bdump

Oracle 11g

/u01/app/oracle/diag/rdbms/truth/truth/trace/

2 Look for errors in the other log and trace files, located in the following directories:

Oracle 10g

$ORACLE_BASE/admin/<SID>/cdump
$ORACLE_BASE/admin/<SID>/adump
$ORACLE_BASE/admin/<SID>/udump

Oracle 11g

Various directories under:

$ORACLE_BASE/diag/rdbms/<SID>/<SID>

Check for Sufficient Free Disk Space in the Tablespaces

To check for sufficient disk space, perform the following steps:

1 Enter the following commands in sqlplus:

```
set line 200
column dummy noprint
column  pct_used format 999.9       heading "Pct|Used"
column  name    format a16      heading "Tablespace Name"
column  Kbytes   format 999,999,999    heading "Current|File Size|MB"
column  used    format 999,999,999   heading "Used MB "
column  free    format 999,999,999  heading "Free MB"
column  largest    format 999,999,999  heading "Largest|Contigous|MB"
column  max_size format 999,999,999 heading "Max Possible|MB"
column  pct_max_used format 999.999       heading "Pct|Max|Used"
break   on report
compute sum of kbytes on report
compute sum of free on report
compute sum of used on report

select nvl(b.tablespace_name,
             nvl(a.tablespace_name,'UNKOWN')) name,
       kbytes_alloc Kbytes,
       kbytes_alloc-nvl(kbytes_free,0) used,
       nvl(kbytes_free,0) free,
       ((kbytes_alloc-nvl(kbytes_free,0))/
                         kbytes_alloc)*100 pct_used,
       nvl(largest,0) largest,
       nvl(kbytes_max,kbytes_alloc) Max_Size,
```

```
       ((kbytes_alloc-nvl(kbytes_free,0))/kbytes_max)*100 pct_max_used
from ( select sum(bytes)/1024/1024 Kbytes_free,
              max(bytes)/1024/1024 largest,
              tablespace_name
       from  sys.dba_free_space
       group by tablespace_name ) a,
     ( select sum(bytes)/1024/1024 Kbytes_alloc,
              sum(decode(maxbytes,0,bytes,maxbytes))/1024/1024 Kbytes_max,
              tablespace_name
       from sys.dba_data_files
       group by tablespace_name
       union all
      select sum(bytes)/1024/1024 Kbytes_alloc,
              sum(decode(maxbytes,0,bytes,maxbytes))/1024/1024 Kbytes_max,
              tablespace_name
       from sys.dba_temp_files
       group by tablespace_name) b
where a.tablespace_name (+) = b.tablespace_name
order by 1
/
```

In the output generated by the preceding commands, compare the numbers under the Used and Free headings.

2 To list the existing data, index, and temporary files, enter the following commands in sqlplus:

```
Select file_id, bytes, file_name from dba_data_files;
Select file_id, bytes, file_name from dba_temp_files;
```

3 If a tablespace has auto-extended to its maximum size and is running out of disk space, then add new data files by entering the ALTER TABLESPACE command in sqlplus.

The following example commands add data files to four of the tablespaces. For a full list of tablespaces and data files, see the output generated by the commands in the preceding two steps.

```
ALTER TABLESPACE "AAA_DATA"
ADD DATAFILE '/u01/oradata/truth/aaa_data10.dbf'
SIZE 32M AUTOEXTEND ON NEXT 128M MAXSIZE 4000M ;

ALTER TABLESPACE "AAA_INDX"
ADD DATAFILE '/u02/oradata/truth/aaa_indx11.dbf'
SIZE 32M AUTOEXTEND ON NEXT 128M MAXSIZE 4000M ;

ALTER TABLESPACE "UNDO"
ADD DATAFILE '/u03/oradata/truth/undo12.dbf' SIZE 32M AUTOEXTEND ON NEXT
128M MAXSIZE 4000M ;

ALTER TABLESPACE "TEMP" ADD
TEMPFILE '/u04/oradata/truth/temp14.dbf' SIZE 32M AUTOEXTEND ON NEXT 128M
MAXSIZE 4000M ;
```

Verify that the Database Jobs (System / Schema Statistics and Garbage Collection) Ran Successfully

When the Model Repository is installed, the SA Installer sets up the System/Schema Statistics and the Garbage Collection jobs in Oracle's dba_jobs. dba_jobs runs these jobs at specified time-intervals. The jobs perform system/schema statistics collection and garbage collection. If the system/schema statistics collection jobs do not run successfully, database performance will degrade. If the garbage collection jobs do not run, then old data will accumulate requiring additional disk space. Performance can also be affected.

To verify that the Jobs in DBA_JOBS ran successfully, perform the following steps:

1 Enter the following commands in SQL*Plus:

```
# Su - oracle
# Sqlplus "/ as sysdba"
set line 200
col priv_user format a14
col last format a17
col next format a17
col this format a17
col what format a50
col broken format a1

select job, priv_user, to_char(LAST_DATE, 'MM/DD/YY HH:MI:SS') last,
to_char(NEXT_DATE, 'MM/DD/YY HH:MI:SS') next, broken, what from dba_jobs;
```

In the output generated from the preceding statement, the value of the WHAT column indicates the type of job. If the value of WHAT is DBMS_STATS* or GATHER_*, the job performs statistics collection. The jobs owned by GCADMIN perform the garbage collection. Sample output looks like this:

```
 JOB PRIV_USER       LAST                NEXT                B WHAT
---- -------------- ----------------    ----------------- - --------------------------------------------
       41 TRUTH          04/19/11 11:00:04 04/21/11 11:00:00 N gather_truth_stats;
       43 LCREP          04/19/11 11:00:04 04/21/11 11:00:00 N gather_lcrep_stats;
       42 AAA            04/19/11 11:00:04 04/21/11 11:00:00 N DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', est
                                                               imate_percent=>dbms_stats.auto_sample_size, degree
                                                               =>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO', opt
                                                               ions=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);

       44 GCADMIN        04/20/11 09:00:02 04/21/11 09:00:00 N WAYPURGE.GC_SESSIONS;
       45 GCADMIN        04/20/11 09:00:02 04/21/11 09:00:00 N CHANGELOGPURGE.GC_CHANGELOGS;
       46 GCADMIN        04/20/11 09:00:02 04/21/11 09:00:00 N AUDITPURGE.GC_AUDITLOGS;
       47 GCADMIN        04/20/11 10:39:26 04/20/11 11:39:26 N STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;

      JOB PRIV_USER      LAST              NEXT              B WHAT
--------- -------------- ----------------- ----------------- - --------------------------------------------------
       48 OPSWARE_ADMIN  04/18/11 06:00:01 04/25/11 06:00:00 N DBMS_STATS.GATHER_SYSTEM_STATS(gathering_mode=>'IN
                                                               TERVAL', interval=>30);


8 rows selected.
```

where:

JOB - job id

SCHEMA_USER - the user who with permissions to run the job

LAST_DATE - last date-time when the job was run

NEXT_DATE - next date the job will run

BROKEN - value N = job was successful, value = Y - job failed

WHAT - the type of job

Changes to the Database Statistics Job

The following changes have been made to the database statistics collection jobs. These jobs can be found in the dba_jobs table. These changes are only relevant to upgraded SA Cores.

To view the jobs you can run the following from SQL*plus

```
# Su - oracle
# Sqlplus "/ as sysdba"
set line 200
col priv_user format a14
col what format a50
col job format 999
select job, priv_user, what from dba_jobs where priv_user in ('AAA','TRUTH');
```

Your output should be as follows:

SA 7.50:

```
## TRUTH DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH', options=>'GATHER AUTO');
## AAA   DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', options=>'GATHER AUTO');
```

SA 7.80 and above:

```
 ## TRUTH DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH',
          estimate_percent=>dbms_stats.auto_sample_size,
          degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
          options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
 ## AAA   DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA',
          estimate_percent=>dbms_stats.auto_sample_size,
          degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
          options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
```

SA 9.0 and above:

```
 JOB PRIV_USER      WHAT
---- -------------- --------------------------------------------------
  41 TRUTH          gather_truth_stats;
  42 AAA            DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA',
                    estimate_percent=>dbms_stats.auto_sample_size,
                    degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
                    options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
```

Running the dba_jobs manually

If you need to run the System/Schema Statistics and the Garbage Collection jobs manually, you must first grant the following privilege.

```
SQL> grant create session to truth, aaa, lcrep;
```

To run the statistics collection jobs manually in SQL*Plus, use the commands shown below.

If you copy and paste the following command examples, replace the variables like schema_user_value with the values of the schema_user column displayed by the preceding select statement. Substitute the variables such as job_no_value with the values of the job column displayed by the same select statement.

```
SQL> connect <schema_user_value>/<password>
SQL> exec dbms_job.run(<job_no_value>)
```

After you are done running the jobs, you should revoke the privileges granted above. Log in to SQL*Plus and enter the following command:

```
SQL> revoke create session from truth, aaa, lcrep;
```

Changing the Time Jobs are Run

dba_jobs are run at UTC time. To change the time at which the jobs are run, follow these instructions:

```
sqlplus "/ as sysdba"
set line 300
col what format a40
col next format a17
col last format a17
col interval format a30
col priv_user format a10

select job,broken, priv_user,to_char(last_date,'MM-DD-YY HH:MI:SS') last,
to_char(next_date,'MM-DD-YY HH:MI:SS') next , interval, what from dba_jobs;
```

The above statement provides infomation about a job. Note the job number and the user that has the privilege to run this job.

The output of the above statement is similar to the following (formatting is compressed due to space limitations):

```
JOB B PRIV_USER  LAST              NEXT              INTERVAL                  WHAT
---- - ---------- ----------------- ----------------- ------------------------- -------------------------
  41 N TRUTH      04-19-11 11:00:04 04-21-11 11:00:00 TRUNC(SYSDATE+2)+11/24    gather_truth_stats;
  43 N LCREP      04-19-11 11:00:04 04-21-11 11:00:00 TRUNC(SYSDATE+2)+11/24    gather_lcrep_stats;
  42 N AAA       04-19-11 11:00:04 04-21-11 11:00:00 TRUNC(SYSDATE+2)+11/24    DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>
                                                                                'AAA', estimate_percent=>dbms_stats.auto
                                                                                _sample_size, degree=>10, method_opt=>'F
                                                                                OR ALL COLUMNS SIZE AUTO', options=>'GAT
                                                                                HER', cascade=>TRUE, gather_temp=>TRUE);

  44 N GCADMIN    04-20-11 09:00:02 04-21-11 09:00:00 TRUNC(SYSDATE+1)+9/24     WAYPURGE.GC_SESSIONS;
  45 N GCADMIN    04-20-11 09:00:02 04-21-11 09:00:00 TRUNC(SYSDATE+1)+9/24          CHANGELOGPURGE.GC_CHANGELOGS;
  46 N GCADMIN    04-20-11 09:00:02 04-21-11 09:00:00 TRUNC(SYSDATE+1)+9/24     AUDITPURGE.GC_AUDITLOGS;

 JOB B PRIV_USER  LAST              NEXT              INTERVAL                  WHAT
---- - ---------- ----------------- ----------------- ------------------------- ------------------------
  47 N GCADMIN    04-20-11 10:39:26 04-20-11 11:39:26 SYSDATE+1/24
STORAGEINITIATORPURGE.GC_STORAGEINITIATO
                                                                                RS;

  48 N OPSWARE_AD 04-18-11 06:00:01 04-25-11 06:00:00 TRUNC(SYSDATE+1) + 18/24 + mod
DBMS_STATS.GATHER_SYSTEM_STATS(gathering
       MIN                                          (abs(to_number(to_char(sysdate _mode=>'INTERVAL', interval=>30);
                                                     + 1,'D')) - 7) + 2, 7)
```

In this example user truth changes the time/interval at which the job is run. Any other user can be substituted for user truth.

```
sqlplus "connect / as sysdba"
SQL> grant create session to truth;
Grant succeeded.
```

In the example:

- job no=92
- priv_user = truth

Job 92 is currently running at 9:00a.m.UTC time. Tto change this to 7:00a.m. UTC time, the command is:

```
SQL> connect truth/<password_for_truth>
Connected.
SQL> exec dbms_job.interval(92,'TRUNC(SYSDATE+2)+7/24');
```

Monitor Database Users

To monitor database users, perform the following steps:

1 To check the database users, enter the following command in `sqlplus`:

```
Select username, account_status, default_tablespace,
temporary_tablespace from dba_users;
```

Monitoring the ERROR_INTERNAL_MSG Table

Various SA internal PL/SQL procedures write exceptions to the `truth.ERROR_INTERNAL_MSG` table. You should monitor this table for errors (daily checks are recommended) on all Model Repository (Oracle) databases.

Executing the SQL below lists the data in `error_internal_msg` from the last fifteen days.

You can remove the `WHERE` clause if you want to display all data in the `truth.ERROR_INTERNAL_MSG` table.

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> set line 200
SQL> col ERR_ID format 999999
SQL> col ERR_USER format a8
SQL> col ERR_TABLE format a25
SQL> col ERR_TABLE_PK_ID format a10
SQL> col ERR_CODE format 9999999
SQL> col ERR_TEXT format a20
SQL> col ERR_INFO format a30

SQL> select ERROR_INTERNAL_MSG_ID ERR_ID,
ERR_DATE,
ERR_USER,
ERR_TABLE,
ERR_TABLE_PK_ID,
ERR_CODE,
ERR_TEXT,
DELETE_FLG,
ERR_INFO
from ERROR_INTERNAL_MSG
where ERR_DATE > sysdate - 15
order by ERR_DATE;
```

Rebuilding the SHADOW_FOLDER_UNIT Table

The procedure SHADOW_FOLDER_UNIT_RELOAD is provided in case the contents of SHADOW_FOLDER_UNIT table becomes out of synchronization or there are multiple records of the type (`shadow_folder_unit.folder_id = -1`).

The table can be rebuilt without stopping the system. Simply connect as user `TRUTH`, `TWIST`, `SPIN`, or `OPSWARE_ADMIN` and issue the command:

```
exec SHADOW_FOLDER_UNIT_UTIL.SHADOW_FOLDER_UNIT_RELOAD
```

Check the results from monitoring the `ERROR_INTERNAL_MSG` table. If the results contain:

```
'ERR_TABLE' = 'UNIT_RELATIONSHIPS'
```

do the following:

1. Check if there are records in `truth.SHADOW_FOLDER_UNIT` of the type (`folder_id = -1`).

   ```
   SQL> connect / as sysdba
   SQL>  select count(*) from shadow_folder_unit where folder_id = -1;
   ```

2. If the above SQL returns more than zero rows, then run the following during low database usage time:

   ```
   SQL> grant create session to truth;
   SQL> connect truth/<password>
   SQL> exec SHADOW_FOLDER_UNIT_UTIL.SHADOW_FOLDER_UNIT_RELOAD;
   ```

3. Run the SQL from Monitoring the ERROR_INTERNAL_MSG Table on page 45 and check if the procedure has listed any faulty records. `SHADOW_FOLDER_UNIT_UTIL.SHADOW_FOLDER_UNIT_RELOAD` is idem potent therefore the faulty records can be fixed and you can rerun `SHADOW_FOLDER_UNIT_UTIL.SHADOW_FOLDER_UNIT_RELOAD`.

   HP recommends that you gather table statistics after the data reload:

   ```
   SQL> connect truth/<password>
   SQL> exec dbms_stats.gather_table_stats (
                  ownname=> 'TRUTH',
                  tabname=> 'SHADOW_FOLDER_UNIT',
                  estimate_percent=> DBMS_STATS.AUTO_SAMPLE_SIZE,
                  cascade => true);
   ```

4. Revoke the permissions given to user `truth`:

   ```
   SQL> connect / as sysdba
   SQL> revoke create session to truth;
   ```

Troubleshooting System Diagnosis Errors

If an additional privilege (permission) has been made manually to the database, when SA performs a system diagnosis on the Data Access Engine, an error message might be generated. For example, if an additional grant has been made to the `truth.facilities` table, the following error appears:

```
Test Information
Test Name: Model Repository Schema
Description: Verifies that the Data Access Engine's version of the schema
matches the Model Repository's version.
Component device: Data Access Engine (spin.blue.qa.example.com)
Test Results: The following tables differ between the Data Access Engine
and
the Model Repository:  facilities.
```

To fix this problem, revoke the grant. For example, if you need to revoke a grant on the `truth.facilities` table, log on to the server with the database and enter the following commands:

```
su - oracle
sqlplus "/ as sysdba"
grant create session to truth;
```

```
connect truth/<truth passwd>;
revoke select on truth.facilities from spin;
exit
sqlplus "/ as sysdba"
revoke create session from truth;
```

Oracle Database Backup Methods

It is important that you back up the database on a regular basis. Be sure to use more than one backup method and to test your recovery process.

You can use the following methods to back up the Oracle database:

- **Export-Import**: An export extracts logical definitions and data from the database and writes the information to a file. Export-import does not support point-in-time recoveries. Do not use Export-Import as your only backup and recovery strategy.

  See the information on the `Export-Import` subdirectory in Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 18.

- **Cold or Off-Line Backups**: This procedure shuts the database down and backs up all data, index, log, and control files. Cold or off-line backups do not support point-in-time recoveries.

- **Hot or Online Backups**: During these backups, the database must be available and in `ARCHIVELOG` mode. The tablespaces are set to backup mode. This procedure backs up tablespace files, control files, and archived redo log files. Hot or online backups support point-in-time recoveries.

- **RMAN Backups**: While the database is either off-line or on-line, use the `rman` utility to back up the database.

Regardless of your backup strategy, remember to back up all required Oracle software libraries, parameter files, password files, and so forth. If your database is in `ARCHIVELOG` mode, you also need to back up the archived log files.

For more information on backing up Oracle databases, see the following Oracle documents:

- *Oracle Database 2 Day DBA*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*

These guides are on the Oracle web site at the following URL:

```
http://www.oracle.com/technology/documentation/index.html
```

Useful SQL

The following SQL commands help you manage information in the Oracle database that the Model Repository uses.

Locked and Unlocked User

A user in Oracle 10.2.0.2 will be locked out after ten unsuccessful log on attempts.

To verify whether the user has been locked or unlocked, enter the following SQL command:

```
select username, account_status from dba_users;
```

To unlock the user, enter the following SQL command:

```
>ALTER USER <username> ACCOUNT UNLOCK;
```

GATHER_SYSTEM_STATS

Sometimes the GATHER_SYSTEM_STATS job will be suspended. To remove this from 'AUTOGATHERING" mode, perform the following steps:

1. Select PNAME, pval2 from SYS.AUX_STATS$ where pname ='STATUS';.
2. If the PVAL2 status is "AUTOGATHERING", run GATHER_SYSTEM_STATS with gathering_mode=('STOP');.
3. Run your job 'exec dbms_job.run(xxx);.

BIN$ Objects

If the SA Installer discovers the existence of BIN$ objects in the database, enter the following SQL commands:

```
show parameter recyclebin;

SELECT owner,original_name,operation,type FROM dba_recyclebin;
connect <owner>/password
purge recyclebin; or purge table BIN$xxx;
```

By default, recyclebin is set to OFF.