SA Oracle Setup for the Model Repository — Standalone Version

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

➤ This document is both an appendix in the SA Installation Guide and a standalone document. Sometimes the content of this document needs to be updated between releases. You can use the following URL to download the latest standalone version of this document from the HP Software Support portal after signing in with your HP Passport credentials:

> https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01253513

Supported Oracle Versions and Operating Systems

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.

See the *SA Support and Compatibility Matrix* associated with this release for a list of supported Oracle versions and operating systems.

➤ The *SA Support and Compatibility Matrix* is updated in every release, including minors and CORDs. You can use this URL to download the latest version from the HP Software Support portal after signing in with your HP Passport credentials:

> https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01253535

For additional versions, after logging into the HP Software Support portal, click the house icon to go to the home page, then use the SEARCH box to find any available product document.

System Requirements

The following sections list the system requirements for Oracle 11g and 12c. The SA Installer performs an automated check to ensure that these requirements are met on the Oracle host.

The system requirements and configurations listed in this section apply both to the SA-supplied Oracle RDBMS software as well as to non-SA-supplied Oracle RDBMS and software installations.

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

➤ The Oracle database must be installed either on its own host or on a server that has the SA Infrastructure Component bundle installed.

Database Server Time Requirements

Database servers must meet the following requirements. These time requirements do not apply to Managed Servers.

- All SA database 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
    ```

Hostname Setup

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

    ```
    # ping <hostname>
    ```

2   Check that the database server name is FQDN by using the following command:

    ```
    # 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], [],
[], [], [], []
```

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 and 12c under Linux.

- For detailed Linux requirements, see the *Oracle® Database Quick Installation Guide11g Release 2 (11.2) for Linux x86-64* (Part Number E24326-02) and *Oracle® Database Quick Installation Guide12c Release 1 (12.1) for Linux x86-64* (Part Number E17718-09) available at:

    http://docs.oracle.com

- Determine the processor type to verify that the processor's architecture matches the Oracle software release you will install. Use the following command to check system architecture:

    ```
    # uname -m
    ```

- The recommended physical memory is 16 GB or more of RAM. An HP-supplied Oracle installation will use a minimum of 2 GB 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 1:

table 1   Required Available RAM Swap Space

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

  You can use the following command to check swap space:

  ```
  grep SwapTotal /proc/meminfo
  ```

- As of Oracle 11g, Automatic Memory Management (ASMM) requires more shared memory (`/dev/shm`) and file descriptors. Shared memory should be sized to be at least the greater of `MEMORY_MAX_TARGET` and `MEMORY_TARGET` for each Oracle instance on a database server.

  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 –h /tmp
  ```

Solaris, HP-UX and AIX Requirements

See also Oracle Solaris, HP-UX and IBM AIX Version and Package Requirements on page 13.

HP-UX and IBM

For HP-UX and IBM requirements, refer to the *Checking the Hardware Requirements* section in the following Oracle documents:

- *Oracle® Database Quick Installation Guide 11g Release 2 (11.2) for HP-UX Itanium, Part Number E24342-03*

- *Oracle® Database Quick Installation Guide 12c Release 1 (12.1) for HP-UX Itanium, Part Number E56893-01*

- *Oracle® Database Quick Installation Guide 11g Release 2 (11.2) for IBM AIX on POWER Systems (64-Bit), Part Number E24335-03*

- *Oracle® Database Quick Installation Guide 12c Release 1 (12.1) for IBM AIX on POWER Systems (64-Bit), Part Number E56891-01*

Oracle Solaris

For Oracle Solaris requirements, refer to the *Checking the Hardware Requirements* section in the following Oracle documents:

- *Oracle® Database Quick Installation Guide 11g Release 2 (11.2) for Oracle Solaris on SPARC (64-Bit), Part Number E24349-03*

- *Oracle® Database Quick Installation Guide 12c Release 1 (12.1) for Oracle Solaris on SPARC (64-Bit), Part Number E17756-08*

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 3.5 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 2. If you need to determine a more precise tablespace sizing, contact your technical support representative.

table 2   Tablespace Sizes

Tablespace	MB/1000	Recommended Minimum Tablespace Size
AAA_DATA	256 MB	2000 MB
AAA_INDX	256 MB	2000 MB
AUDIT_DATA	256 MB	2000 MB
AUDIT_INDX	256 MB	2000 MB
LCREP_DATA	3000 MB	8000 MB
LCREP_INDX	2000 MB	8000 MB
TRUTH_DATA	1500 MB	4000 MB
TRUTH_INDX	500 MB	4000 MB
STRG_DATA	1300 MB	2000 MB
STRG_INDX	400 MB	2000 MB

Software Requirements

This section lists the requirements for running Oracle 11g and 12c under Red Hat Enterprise Linux, Oracle Enterprise Linux and SUSE Linux Enterprise Server.

Linux Requirements

The following are software requirements for running Oracle 11g and 12c under Red Hat Enterprise Linux, Oracle Enterprise Linux and SUSE Linux Enterprise Server:

- Required operating system version for 11g:
    - Oracle Linux 5 Update 2 (with Red Hat Compatible Kernel)
    - Oracle Linux 6 (with Red Hat Compatible Kernel)
    - Red Hat Enterprise Linux 5 Update 2
    - Red Hat Enterprise Linux 6
    - SUSE Linux Enterprise Server 10 SP2
    - SUSE Linux Enterprise Server 11
- Required operating system version for 12c:

- — Oracle Linux 5 Update 6 (with Red Hat Compatible Kernel)
- — Oracle Linux 6 (with Red Hat Compatible Kernel)
- — Red Hat Enterprise Linux 5 Update 6
- — Red Hat Enterprise Linux 6
- — SUSE Linux Enterprise Server 11 SP2

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

```
# cat /proc/version
```

- Required Kernel version for Oracle Database 11g Release 2 (11.2):
  - — Oracle Linux 5 Update 2

    2.6.18 or later (with Red Hat Compatible Kernel)
  - — Oracle Linux 6

    2.6.32-71.el6.x86_64 or later (with Red Hat Compatible Kernel)
  - — Red Hat Enterprise Linux 6

    2.6.32-71.el6.x86_64 or later
  - — Red Hat Enterprise Linux 5 Update 2

    2.6.18 or later
  - — SUSE Linux Enterprise Server 10

    2.6.16.21 or later
  - — SUSE Linux Enterprise Server 11

    2.6.27.19 or later

- Required Kernel version for Oracle Database 12c Release 1 (12.1):
  - — Oracle Linux 5 Update 6

    2.6.18-238.0.0.0.1.el5 or later
  - — Oracle Linux 6 (with Red Hat Compatible Kernel)

    2.6.32-71.el6.x86_64 or later
  - — Red Hat Enterprise Linux 5 Update 6

    2.6.18-238.0.0.0.1.el5 or later
  - — Red Hat Enterprise Linux 6

    2.6.32-71.el6.x86_64 or later
  - — SUSE Linux Enterprise Server 11 SP2

    3.0.13-0.27 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
```

Linux Package Requirements

Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 for Oracle 11g

The following or later package versions for Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 (with Red Hat compatible kernel) must be installed (shaded rows indicate 32-bit packages):

➤ Starting with Oracle Database 11g Release 2 (11.2.0.2), all the *32-bit packages*, excepting `gcc-32bit-4.3`, listed in the following table are no longer required for installing a database on Linux x86-64. Only the 64-bit packages are required. However, for any Oracle Database 11g release before 11.2.0.2, both the 32-bit *and* 64-bit packages listed in the following table are required..

table 3   Required Packages for Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 for Oracle 11g

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
glibc-devel	2.5
glibc-devel	2.5 (32-bit)
glibc-headers	2.5
ksh	NA
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

Required Packages	Version
make	3.81
sysstat	7.0.2

Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 for Oracle 12c

The following or later package versions for Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 (with Red Hat compatible kernel) must be installed (shaded rows indicate 32-bit packages):

table 4   Required Packages for Red Hat Enterprise Linux 5 and Oracle Enterprise Linux 5 for Oracle
              12c

Required Packages	Version
binutils	2.17.50.0.6
compat-libstdc++	33-3.2.3
compat-libstdc++	33-3.2.3 (32-bit)
gcc	4.1.2
gcc-c++	4.1.2
glibc	2.5-58
glibc	2.5-58 (32-bit)
glibc-common	2.5
glibc-devel	2.5-58
glibc-devel	2.5-58 (32-bit)
ksh	NA
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
libXext	1.0.1
libXext	1.0.1 (32-bit)
libXtst	1.0.1

Required Packages	Version
libXtst	1.0.1 (32-bit)
libX11	1.0.3
libX11	1.0.3 (32-bit)
libXau	1.0.1
libXau	1.0.1 (32-bit)
libXi	1.0.1
libXi	1.0.1 (32-bit)
make	3.81
sysstat	7.0.2

Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 for Oracle 11g

The following or later package versions for Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 (with Red Hat compatible kernel) must be installed:

table 5   **Required Packages for Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 for Oracle 11g**

Required Packages	Version
binutils	2.20.51.0.2–5.11.el6 (x86_64)
compat-libcap1	1–1.10-1 (x86_64)
compat-libstdc++	33–3.2.3-69.el6 (x86_64)
compat-libstdc++	33–3.2.3-69.el6.i686
gcc	4.4.4-13.el6 (x86_64)
gcc-c++	4.4.4-13.el6 (x86_64)
glibc	2.12-1.7.el6 (x86_64)
glibc	2.12-1.7.el6 (i686)
glibc-devel	2.12-1.7.el6 (x86_64)
glibc-devel	2.12-1.7.el6.i686
ksh	NA
libaio	0.3.107-10.el6 (x86_64)
libaio	0.3.107-10.el6.i686
libaio-devel	0.3.107-10.el6 (x86_64)
libaio-devel	0.3.107-10.el6.i686
libgcc	4.4.4-13.el6 (x86_64)

Required Packages	Version
libgcc	4.4.4-13.el6 (i686)
libstdc++	4.4.4-13.el6 (x86_64)
libstdc++	4.4.4-13.el6.i686
libstdc++-devel	4.4.4-13.el6 (x86_64)
libstdc++-devel	4.4.4-13.el6.i686
make	3.81-19.el6
sysstat	9.0.4-11.el6 (x86_64)

Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 for Oracle 12c

The following or later package versions for Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 (with Red Hat compatible kernel) must be installed:

table 6   Required Packages for Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 for Oracle 12c

Required Packages	Version
binutils	2.20.51.0.2-5.11.el6 (x86_64)
compat-libcap1	1.10-1 (x86_64)
compat-libstdc++	33-3.2.3-69.el6 (x86_64)
compat-libstdc++	33-3.2.3-69.el6 (i686)
gcc	4.4.4-13.el6 (x86_64)
gcc-c++	4.4.4-13.el6 (x86_64)
glibc	2.12-1.7.el6 (x86_64)
glibc	2.12-1.7.el6 (i686)
glibc-devel	2.12-1.7.el6 (x86_64)
glibc-devel	2.12-1.7.el6 (i686)
ksh	NA
libaio	0.3.107-10.el6 (x86_64)
libaio	0.3.107-10.el6 (i686)
libaio-devel	0.3.107-10.el6 (x86_64)
libaio-devel	0.3.107-10.el6 (i686)
libgcc	4.4.4-13.el6 (x86_64)
libgcc	4.4.4-13.el6 (i686)
libstdc++	4.4.4-13.el6 (x86_64)

table 6 **Required Packages for Red Hat Enterprise Linux 6 and Oracle Enterprise Linux 6 for Oracle 12c (cont'd)**

Required Packages	Version
libstdc++	4.4.4-13.el6 (i686)
libstdc++-devel	devel-4.4.4-13.el6 (x86_64)
libstdc++-devel	4.4.4-13.el6 (i686)
make	3.81-19.el6
sysstat	9.0.4-11.el6 (x86_64)

SUSE Linux Enterprise Server 10 for Oracle 11g

The following or later package versions for SUSE Linux Enterprise Server 10 must be installed:

table 7 **Required Packages for SUSE Linux Enterprise Server 10 for Oracle 11g**

Required Packages	Version
binutils	2.16.91.0.5
compat-libstdc++	5.0.7
gcc	4.1.0
gcc-c++	4.1.2
glibc	4.1.2
glibc-devel	2.4-31.63
glibc-devel	32bit-2.4-31.63
ksh	93r-12.9
libaio	0.3.104
libaio	32bit-0.3.104
libaio-devel	0.3.104
libaio-devel	32bit-0.3.104
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

SUSE Linux Enterprise Server 11 for Oracle 11g

The following or later package versions for SUSE Linux Enterprise Server 11 must be installed:

Required Packages	Version
binutils	2.19
gcc	4.3
gcc	32bit-4.3
gcc-c++	4.3
glibc	2.9
glibc	32bit-2.9
glibc-devel	2.9
glibc-devel	32bit-2.9
ksh	93t
libaio	0.3.104
libaio	32bit-0.3.104
libaio-devel	0.3.104
libaio-devel	32bit-0.3.104
libgcc43	4.3.3_20081022
libstdc++-devel	4.3
libstdc++33	3.3.3
libstdc++33	32bit-3.3.3
libstdc++43	4.3.3_20081022
libstdc++43	32bit-4.3.3_20081022
libstdc++43-devel	4.3.3_20081022
libstdc++43-devel	32bit-4.3.3_20081022
make	3.81
sysstat	8.1.5

SUSE Linux Enterprise Server 11 for Oracle 12c

The following or later package versions for SUSE Linux Enterprise Server 11 must be installed:

table 9   Required Packages for SUSE Linux Enterprise Server 11 for Oracle 12c

Required Packages	Version
binutils	2.21.1-0.7.25
gcc	4.3-62.198
gcc-c++	4.3-62.198
glibc	2.11.3-17.31.1
glibc-devel	2.11.3-17.31.1
ksh	93u-0.6.1
libaio	0.3.109-0.1.46
libaio-devel	0.3.109-0.1.46
libcap1	1.10-6.10
libgcc46	4.6.1_20110701-0.13.9
libstdc++33	3.3.3-11.9
libstdc++33	3.3.3-11.9 (32-bit)
libstdc++43-devel	4.3.4_20091019-0.22.17
libstdc++46	4.6.1_20110701-0.13.9
make	3.81
sysstat	8.1.5-7.32.1
xorg-x11-libs	7.4 (x86_64)
xorg-x11-libs	7.4 (32-bit)
xorg-x11-libX11	7.4 (x86_64)
xorg-x11-libX11	7.4 (32-bit)
xorg-x11-libXau	7.4 (x86_64)
xorg-x11-libXau	7.4 (32-bit)
xorg-x11-libxcb	7.4 (x86_64)
xorg-x11-libxcb	7.4 (32-bit)
xorg-x11-libXext	7.4 (x86_64)
xorg-x11-libXext	7.4 (32-bit)

Verifying that Packages are Installed

To verify that RPMs are installed under Linux, enter the following command:

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

Oracle Solaris, HP-UX and IBM AIX Version and Package Requirements

For Oracle Solaris, HP-UX and IBM AIX operating system, compiler, patch and any additional software requirements, see the *Checking the Software Requirements* section in the *Oracle® Database Quick Installation Guide* for your operating system.

Non-SA-Supplied Oracle Software and Database Setup

➤ If you plan to install the SA-supplied Oracle RDBMS software and database, you do not need to perform the tasks in this section. The SA Installer performs all the tasks discussed below. For information about installing the SA-supplied Oracle software and database, see SA-Supplied Oracle RDBMS Software and Database Setup on page 21.

If you plan to use a non-SA-supplied Oracle database with the SA Model Repository, the following steps are required for compatibility with SA. You should also review System Requirements on page 1 before preceding with this section.

Modifiable Kernel Parameters

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

You can find additional information about kernel parameter configuration in the *Configuring Kernel Parameters* section of the *Oracle® Database Quick Installation Guide.*

Modifiable Kernel Parameter Values for Linux

This section identifies the kernel parameters you can change for supported Linux operating systems.

You can change values for the following parameters in /etc/sysctl.conf. If the current value of any parameter is higher than the value listed in this table, then do not change the value of that parameter:

```
#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=9081 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 parameter:

For **64-bit Linux** in `/lib64/security/pam_limits.so`

```
session required pam_limits.so
```

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

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

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

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

Modifiable Kernel Parameter Values for SUSE Linux x86_64

This section identifies additional required settings for SUSE Linux x86_64 when running Oracle 11g or 12c:

- 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`.

Modifiable Kernel Parameter Values for Oracle SPARC Solaris (64 bit), HP-UX and IBM AIX

Refer to the *Configuring Kernel Parameters* section in the following Oracle documents:

- *Database Quick Installation Guide for Oracle Solaris on SPARC (64 Bit)*

- *Database Quick Installation Guide for HP-UX Itanium*

- *Database Quick Installation Guide for IBM AIX on POWER Systems (64-Bit)*

Oracle Database Installation Steps

In order to install an Oracle database for use with the SA Model Repository, you must perform the following tasks which are explained in more detail in the following sections:

1 Create the database with the UTF8 database character set

2 Set the database with `TIME_ZONE` to `'+00:00'`

3 Create the database with the required initialization (`init.ora`) parameters

4 Create the database with required tablespaces

5 Create the database user `opsware_admin`

6 `tnsnames.ora` file requirements

7 File linking requirements

8 Enable Oracle Daylight Savings Time (DST)

9 `sqlnet.ora` file requirements

1. UTF8 Database Character Set

Create the database with the UTF8 database character set:

```
CHARACTER SET UTF8
```

2. Set the Database TIME_ZONE

Create the database with `TIME_ZONE` set to `'+00:00'`:

```
SET TIME_ZONE = '+00:00'
```

3. Specify the Required Initialization (init.ora) Parameters

Create the database instance with the following initialization (`init.ora`) parameters. For parameters not listed, SA assumes that the default Oracle parameters are used.

Oracle 11.2.0.x

```
compatible := required to be >= 11.2.0
cursor_sharing := required to be = FORCE
db_file_multiblock_read_count := suggested to be >= 16
db_block_size := required to be >= 8192
deferred_segment_creation := required to be = FALSE
event := required to be = 12099 trace name context forever, level 1
job_queue_processes := required to be >= 1000
log_buffer := required to be >= 5242880
memory_target := required to be >= 1879048192 (1.75GB)
nls_length_semantics := required to be = CHAR
nls_sort := required to be = GENERIC_M
open_cursors := required to be >= 1500
optimizer_index_cost_adj := required to be = 20
optimizer_index_caching := required to be = 80
optimizer_mode := 'required to be = ALL_ROWS
processes := required to be >= 1024
recyclebin := required to be = OFF
remote_login_passwordfile := required to be = EXCLUSIVE
```

```
session_cached_cursors := required to be >= 50
undo_tablespace := should be = UNDO or other UNDO tablespace
undo_management := should be = AUTO
_complex_view_merging := required to be = FALSE
```

Oracle 12.1.0.x

```
compatible := required to be >= 12.1.0
cursor_sharing := required to be = FORCE
db_block_size := required to be >= 8192
db_file_multiblock_read_count := suggested to be >= 16
deferred_segment_creation := required to be = FALSE
job_queue_processes := required to be >= 1000
max_string_size := required to be = STANDARD
memory_target := required to be >= 2684354560 (2.5GB)
nls_length_semantics := required to be = CHAR
nls_sort := required to be = GENERIC_M
open_cursors := required to be >= 1500
optimizer_index_cost_adj := required to be = 100
optimizer_index_caching := required to be = 0
optimizer_mode := 'required to be = ALL_ROWS
processes := required to be >= 1024
recyclebin := required to be = OFF
remote_login_passwordfile := required to be = EXCLUSIVE
session_cached_cursors := required to be >= 50
undo_tablespace := should be = UNDO or other UNDO tablespace
```

Note: The parameters `_complex_view_merging` and `event` are no longer required for Oracle 12c.

4. Create the Required Tablespaces

The following tablespaces must be created to support SA. For tablespace disk space requirements, see
Model Repository (Database) Disk Space Requirements on page 4.

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

5. Create the Database User opsware_admin

Create the database user 'opsware_admin' with the following privileges.

```
SQL> create user opsware_admin identified by opsware_admin
default tablespace truth_data temporary tablespace temp
```

```
                    quota unlimited on truth_data;
                    SQL> grant alter session to opsware_admin with admin option;
                    SQL> grant create procedure to opsware_admin with admin option;
                    SQL> grant create public synonym to opsware_admin with admin option;
                    SQL> grant create sequence to opsware_admin with admin option;
                    SQL> grant create session to opsware_admin with admin option;
                    SQL> grant create table to opsware_admin with admin option;
                    SQL> grant create trigger to opsware_admin with admin option;
                    SQL> grant create type to opsware_admin with admin option;
                    SQL> grant create view to opsware_admin with admin option;
                    SQL> grant delete any table to opsware_admin with admin option;
                    SQL> grant drop public synonym to opsware_admin with admin option;
                    SQL> grant select any table to opsware_admin with admin option;
                    SQL> grant select_catalog_role to opsware_admin with admin option;
                    SQL> grant query rewrite to opsware_admin with admin option;
                    SQL> grant restricted session to opsware_admin with admin option;
                    SQL> grant execute on dbms_utility to opsware_admin with grant option;
                    SQL> grant analyze any to opsware_admin;
                    SQL> grant insert, update, delete, select on sys.aux_stats$ to opsware_admin;
                    SQL> grant gather_system_statistics to opsware_admin;
                    SQL> grant create job to opsware_admin with admin option;
                    SQL> grant create any directory to opsware_admin;
                    SQL> grant drop any directory to opsware_admin;
                    SQL> grant alter system to opsware_admin;
                    SQL> grant create role to opsware_admin;
                    SQL> grant create user to opsware_admin;
                    SQL> grant alter user to opsware_admin;
                    SQL> grant drop user to opsware_admin;
                    SQL> grant create profile to opsware_admin;
                    SQL> grant alter profile to opsware_admin;
                    SQL> grant drop profile to opsware_admin;
```

6. 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 locations:

  ```
  /var/opt/oracle/tnsnames.ora
  $ORACLE_HOME/network/admin
  ```

- 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)))
```

7. File Linking Requirements:

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

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

    `/var/opt/oracle`

    Verify that the file conforms to the rules listed in

2   If it does not exist, create the following directory:

    `mkdir -p /var/opt/oracle`

3   Create the following symbolic link:

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

4   Ensure that the oracle Unix user has read-write permission on the `tnsnames.ora` file.

For Red Hat Enterprise Linux:

1   Create another symbolic link:

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

Next:

1   Copy the sample `opsware-oracle` script to `/etc/init.d/`.

2   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. Enable Oracle Daylight Savings 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 11g, 12c or higher.

2   Use MetaLink Note 412160.1 to apply Oracle Database time zone fixes specific to your database
    version.

    Use MetaLink Note 412160.1 to apply time zone fixes to the Oracle Java Virtual Machine (JVM) in the
    Oracle Database specific to your E-Business Suite database version.

9. sqlnet.ora requirements

Some applications in Server Automation use the oracle `classes12.jar` file to connect to the database.
To enable these utilities to connect to the Oracle 12C database, create a `sqlnet.ora` in the
`$ORACLE_HOME/network/admin` folder in *both* the SA Client system and the SA Core Database server
with the following contents:

```
# File:         sqlnet.ora
# Certified:    Oracle 12.1.0
# Purpose:      Configuration File for all Net8 Clients
# Notes:        None

LOG_DIRECTORY_SERVER=/u01/app/oracle/product/12.1.0/db_1/network/log
LOG_FILE_SERVER=sqlnet.log
TRACE_DIRECTORY_SERVER=/u01/app/oracle/product/12.1.0/db_1/network/trace
TRACE_FILE_SERVER=sqlnet.trc
NAMES.DIRECTORY_PATH= (TNSNAMES)
SQLNET.INBOUND_CONNECT_TIMEOUT=180
SQLNET.ALLOWED_LOGON_VERSION_SERVER=8
SQLNET.ALLOWED_LOGON_VERSION_CLIENT=8
```

SA Database Installation Sample Scripts

HP Support can provide sample scripts for steps 1 through 5 of the Oracle Database Installation Steps.

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

- **truth.sh**: A shell script that creates directories and then launches the `truth.sql` script. Running this script causes all the scripts to be run automatically, in the correct order.

- **truth.sql**: Prompts for passwords of the `SYS` and `SYSTEM` users and launches the remainder of the SQL scripts in this list.

- **CreateDB.sql**: Creates a database with the UTF8 character set and `TIME_ZONE` set to `'+00:00'`

- **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 "Model Repository (Database) Disk Space Requirements" in the *SA Installation Guide* 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. See 3. Specify the Required Initialization (init.ora) Parameters on page 15.

- **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 12.1.0.1 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*.

Creating the Database using the SA-Supplied Scripts

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

1. Obtain the database creation scripts from your HP support representative.

2. Make any required changes to the scripts.

3. As `root`, create the Unix user `oracle` and log in to the server as the user `oracle`.

4. Copy the SA-supplied files to the following directory:

   `$ORACLE_BASE/admin/truth/create`

5. Change the mode of the SA-supplied `truth.sh` script:

   `chmod 755 truth.sh`

6. Launch the SQL scripts that create the database by running the `truth.sh` script:

   `./truth.sh`

7. 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`

SA-Supplied Oracle RDBMS Software and Database Setup

➤ If you plan to install the Oracle RDBMS software and database yourself, you do not need to perform the tasks in this section. See Non-SA-Supplied Oracle Software and Database Setup on page 13.

If you plan to use a SA-supplied Oracle database with the SA Model Repository, you should read the following sections for information about what the SA Installer does when installing the Oracle software and database during SA installation. The SA Installer performs all the tasks discussed below. You should also review System Requirements on page 1 before proceeding with this section.

SA-Supplied RDBMS Configuration Details

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

- Checks that all requirements are met on the host server (see System Requirements on page 1).

- Sets certain kernel parameters to required values (see Modifiable Kernel Parameters on page 13).

- 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/12.1.0/db_1`

- Sets the `$ORACLE_SID` environment variable to `truth`.

- Creates a database with the UTF8 character set, `TIME_ZONE` set to `'+00:00'` and with 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 `tnsames.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 `$ORACLE_HOME/network/admin/sqlnet.ora` file that allows older client versions to connect to the Oracle 12C database.

- 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`

Security

SA recommends that you change the default passwords for the following:

- the Unix user `oracle`

- the Oracle database users SYS and SYSTEM

  SA does not use the SYS and SYSTEM users.

- the Oracle listener

  In the `/$ORACLE_HOME/network/admin/listener.ora` file, SA sets 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, the Oracle listener uses port 1521.

SA-Supplied Oracle Installation Procedure

SA supports the following SA/Oracle database configurations:

- SA Core and Oracle database on a single host

- SA Core with the Oracle database on a remote database server

In the *SA Installation Guide*, see Chapter 2, "SA Core Configurations Supported For Customer Installation" for a description of supported SA Core/Oracle database configurations and Chapter 5, "SA Core Installation" for installation instructions.

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 tasks:

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

   a   Install the Oracle Full Client software.

      The steps below use `/u01/app/oracle/product/12.1.0/client_1` as the Oracle Full Client home.

➤   The Oracle Full Client must be the same version as the Oracle database.

   b   Ensure that the Oracle Full Client software is owned by the OS user `oracle`.

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

   d   If it does not exist, create the following directory:

      `mkdir -p /var/opt/oracle`

   e   Create the following symbolic link:

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

   f   Ensure that the Unix user `oracle` has read-write permission on the `tnsnames.ora` file.

   g   Ensure that the `$ORACLE_HOME/network/admin/sqlnet.ora` file is created in both the client as well as the Oracle Database Server as described in step 9 under the section Oracle Database Installation Steps on page 15.

   h   Ensure that the SA Installer Core Definition File (CDF) 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`), and so on. Based on the above steps your parameter values will be:

      –   `%truth.tnsdir=/var/opt/oracle`

      –   `%db.orahome=/u01/app/oracle/product/12.1.0/client_1`

      –   `%db.port=1521`

      –   `%truth.dcSubDom=prod.example.com`

      –   `db.host=192.168.9.99` (server on which the Oracle database is installed)

   i   Ensure that the `COMPATIBLE` parameter is set correctly and that it matches the database version. For example, for database software that is version 12.1.0.1 ensure that `COMPATIBLE=12.1.0.1`. SA uses Oracle's Export Data Pump and Import Data Pump utilities during secondary core creation. These utilities require the `COMPATIBLE` parameter be specified correctly.

2   Perform the following tasks on the Model Repository host:

   a   Log in as the user `oracle`.

   b   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 Oracle RAC 11g or Oracle RAC 12c, configured as described in the following sections.

Supported Oracle Versions and Operating Systems

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.

- See the *SA 10.2 Support and Compatibility Matrix* for a list of supported Oracle versions and operating systems.

➤ The *SA Support and Compatibility Matrix* is updated in every release, including minors and CORDs. You can use this URL to download the latest SA10.2x version of the from the HP Software Support portal after signing in with your HP Passport credentials:

https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01253535

For additional versions, after logging into the HP Software Support portal, click the house icon to go to the home page, then use the SEARCH box to find any available product document.

System Requirements

See

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, the Oracle database must be configured for use with SA. You may 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`, `sqlnet.ora`, and `tnsnames.ora` files.

Create the Database with the Required Initialization Parameters

Follow the procedure described in

You should perform the following tasks listed in the section:

- Modifiable Kernel Parameters
- Oracle Database Installation Steps

    a Create the database with the UTF8 database character set

    b Set the database with `TIME_ZONE` to `'+00:00'`

    c Create the database with the required initialization (`init.ora`) parameters

    d Create the database with required tablespaces

e Create the database user `opsware_admin`

➤ Use the `tnsnames.ora` file and file linking requirements listed in the following sections since they differ for the Oracle RAC environment from those listed in Non-SA-Supplied Oracle Software and Database Setup.

Installing the Model Repository

In most production environments with Oracle RAC, you can perform the Model Repository installation 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 an SA server (`rac1sa.dev.opsware.com`) on which SA will be installed and a 2 node RAC configuration shown in Table 10

table 10   **Example RAC Configurations**

Identity	Host Note	Name	Type	Address	Address Static or Dynamic	Resolved By
Node 1 Public	rac1pub	rac1pub	Public	192.168.173.210	Static	DNS
Node 1 Virtual	Selected by Oracle Clusterware	rac1-vip	Virtual	192.168.173.212	Static	DNS and/or host file
Node 1 Private	rac1pub	rac1prv	Private	172.16.1.100	Static	DNS, host file or none
Node 2 Public	rac2pub	rac2pub	Public	192.168.173.211	Static	DNS
Node 2 Virtual	Selected by Oracle Clusterware	rac2-vip	Virtual	192.168.173.213	Static	DNS and/or host file
Node 2 Private	rac2pub	rac2prv	Private	172.16.1.101	Static	DNS, host file or none
SCAN vip 1	Selected by Oracle Clusterware	sa_cluster1-scan	Virtual	192.168.173.216	Static	DNS
SCAN vip 2	Selected by Oracle Clusterware	sa_cluster1-scan	Virtual	192.168.173.217	Static	DNS
SCAN vip 3	Selected by Oracle Clusterware	sa_cluster1-scan	Virtual	192.168.173.218	Static	DNS

Installing the Model Repository in a RACed Environment

➤ 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`.

1 **Model Repository Hostname Resolution**

Ensure that all the public, VIP and SCAN addresses are resolvable by using `nslookup`. If you are not using `nslookup`, then update the `/etc/hosts` file with those names.

Example:

```
# nslookup rac1-vip.dev.opsware.com

# nslookup sa_cluster1-scan.dev.opsware.com
```

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
```

➤ If you have set up Oracle Clusterware, you should use the Clusterware IP address rather than a single database node IP address. For example:

```
192.168.173.216 truth sa_cluster1-scan sa_cluster1-scan.dev.opsware.com
```

If you have set up SCAN name, you should use the SCAN address rather than the database node IP address.

2   Install the Oracle 11g or Oracle 12c Full Client on the SA Server

➤ For Oracle 12.1.0.1, use the Oracle Full Client Version 12.1.0.1.

a   The SA Installer uses 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 the database 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)
```

b   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=$ORACLE_BASE/product/11.2.0/client_1 (for 11g Client) or
# export ORACLE_HOME=$ORACLE_BASE/product/12.1.0/client_1 (for 12c Client)
#PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$PATH
export PATH

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

umask 022
#SA-OracleRAC parameters end
```

c   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.

d   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>
```

e   Start Oracle Full Client installation

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

```
/u01/app/oracle/product/12.1.0/client_1
```

f   Run the Oracle Universal Installer to install Oracle Full Client. The directories in this example assume an Oracle 12c Full Client on Linux.

1. `cd /<location_of_oracle_full_client>`

2. `./runInstaller`

3. At the Welcome Screen, click Next.

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

5. For Select Installation Type, choose Administrator, click Next.

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

7. 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 click Next.

8. The Oracle OUI will list the products that are to be installed. Click Install.

9. The OUI shows the progress bar while installing.

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

11. Click Finish when the installation completes.

12. You must run the following two configuration scripts as `root` after installation completes:

   `-/u01/app/oraInventory/orainstRoot.sh`

```
~/u01/app/oracle/product/12.1.0/client_1/root.sh
```

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

h Uncomment `$ORACLE_HOME` and `$ORACLE_PATH`.

3 **Making Changes to tnsnames.ora on an SA Server (Use tnsnames.ora-install_upgrade File)**

By default SA expects the `tnsnames.ora` file to be located in `/var/opt/oracle`.

a Login as `root` on the SA server from which the installer will be run.

b Enter the command:

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

c Copy `tnsnames.ora` from the remote database server to the directory you created above.

For the RAC environment, copy `tnsnames.ora` from RAC Node 1 (for example, `rac1pub.dev.opsware.com`).

To accommodate the remote Model Repository installation process, two sets of `tnsnames.ora` files 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 `tnsnames.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`. For example:

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

tnsnames.ora-install_upgrade Sample File

```
# tnsnames.ora Network Configuration File: /u01/app/oracle/product/
12.1.0/db_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.
RAC1SA_TRUTH =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.dev.opsware.com)
(PORT = 1521))
(CONNECT_DATA =
(UR=A)
(SERVER = DEDICATED)
(SERVICE_NAME = truth)
)
)
RAC2SA_TRUTH=(DESCRIPTION=(ADDRESS=(HOST=192.168.173.214)(PORT=20002)(
PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

d Ensure that the `$ORACLE_HOME/network/admin/sqlnet.ora` file is created in both the client as well as the Oracle Database Server as described in step 9 under the section Oracle Database Installation Steps on page 15.

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` file is configured correctly and if the SA Installer can connect to the database.

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 (Use tnsnames.ora-install_upgrade File) on page 28.

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/12.1.0/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@RAC1SA_TRUTH as sysdba;`

        where `rac1sa_truth` is the `service_name` or entry from the `tnsnames.ora` file

    g   `connect opsware_admin/<password>@RAC1SA_truth`

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

SA Installer Core Definition File (CDF)

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 Core Definition File (CDF).

- `%db.sid: truth1` (Oracle SID of the instance where SA installer is going to connect to.)
- `%db.orahome: /u01/app/oracle/product/12.1.0/client_1`(Oracle client home)
- `%db.port: 1521`(Oracle listener port)
- `%db.host: 192.168.173.210` (IP address of a node where ORACLE RDBMS is installed)
- `%truth.servicename: rac1sa_truth` (value of service name from `tnsnames.ora` file)

You can now install the SA Core as described in the Chapter 5, "SA Core Installation" in the *SA 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 (Use tnsnames.ora-operational File)

After SA Core installation 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 normal SA operations, all the RAC nodes are used.

To accommodate the remote database installation process, two sets of `tnsnames.ora` files 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

➤ If you have set up Oracle Clusterware, you should use the Clusterware IP address rather than a single database node IP address. If you have set up SCAN name, you should use the SCAN address rather than the database node IP address.

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.

tnsnames.ora-operational sample File - with Clusterware Setup

If you have set up Oracle Clusterware, use the following:

#This entry is for connecting to RAC virtual machines. This entry is used by SA during operation of SA.

```
RAC1SA_TRUTH =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = sa_cluster1-scan.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))
)
)
```

#This entry is for connecting to node2 via `service_name`. This is for DBA convenience. This is not used by SA.

```
TRUTH2 =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = rac2pub.dev.opsware.com)(PORT = 1521))
(CONNECT_DATA =
(UR=A)
(SERVER = DEDICATED)
(SERVICE_NAME = truth)
)
)
```

#This entry is for connecting to node1 via `service_name`. This is for DBA convenience. This is not used by SA.

```
TRUTH1 =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = rac1pub.dev.opsware.com)(PORT = 1521))
(CONNECT_DATA =
```

```
(UR=A)
(SERVER = DEDICATED)
(SERVICE_NAME = truth)
)
)
```

During installation, the SA Installer adds an SA Gateway entry into `tnsnames.ora` file (linked to `tnsnames.ora.install-upgrade`) on the Primary SA Core.

After installation completes, copy that entry into the `tnsname.ora.operational` file. If this entry is not present in the `tnsname.ora.operational` file, 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)
(PROTOCOL=tcp))
(CONNECT_DATA=(SERVICE_NAME=truth)))
```

tnsnames.ora-operational Sample File - Without Clusterware Setup

If you have not set up Oracle Clusterware, use the following:

#This entry is for connecting to RAC virtual machines.

```
RAC1SA_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. This is for DBA convenience. This is not used by SA.

```
RAC2SA_TRUTH =
(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 using `service_name`. This entry is optional. This is for DBA convenience. This is not used by SA.

```
RAC1SA_TRUTH =
(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))
```

During installation, the SA Installer adds an SA Gateway entry into `tnsnames.ora` file (linked to `tnsnames.ora.install-upgrade`) on the Primary SA Core.

After installation completes, copy that entry into the `tnsname.ora.operational` file. If this entry is not present in the `tnsname.ora.operational` file, 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) (PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

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

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

vault.conf File Changes

> If you have set up Oracle Clusterware, you should use the Clusterware IP address rather than a single database node IP address. If you have set up SCAN name, you should use the SCAN address rather than the database node IP address.

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.ora` definition instead of the SID. For example:

- *If you have set up Oracle Clusterware, use the following:*

  — Before:

    ```
    db.sid: truth
    ```

  — After:

    ```
    #truth.sid: truth1
    truth.sid: (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
    (HOST = sa_cluster1-scan.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))))
    ```

- *If Oracle Clusterware is not set up, use the following:*

    ```
    #truth.sid: truth1
    truth.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))))
```

- Also, ensure that these values are correct:

```
truth.port: 1521
truth.host: 192.168.173.210 (database server IP)
truth.servicename: rac1sa_truth (tnsnames.ora enter)
```

- Restart the vaultdaemon:

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

da.conf File Changes

As of SA 9.10 and later, the Application Deployment Manager reads database connection information from the `tnsnames.ora` file.

In SA 9.10 and 9.1x, the default was `SID =Truth` unless changed by the user, for example, in `/etc/opt/opsware/da/da.conf`:

```
truth.sid=truth1 (this is the Oracle SID of the instance on RAC node)
```

opsware_start.config File Changes

This file is located in:

```
/opt/opsware/oi_util/startup/opsware_start.config
```

➤ If you have set up Oracle Clusterware, you should use the Clusterware IP address rather than a single database node IP address. If you have set up SCAN name, you should use the SCAN address rather than the database node IP address.

- `TRUTH_HOST="192.168.173.210"` - If Clusterware is not set up, then set the `TRUTH_HOST` value to one of the node's hostnames or public IPs.

- `TRUTH_HOST="sa_cluster1-scan.dev.opsware.com"` - If Clusterware is set up, then set the `TRUTH_HOST` value to the Clusterware scan name.

Setting up a Secondary SA core in an Oracle RACed environment.

During the interview process the installer asks for the secondary cores' database host information. Enter the IP or host name of the secondary cores single RACed node. During the install process the installer connects to the database via a single node only.

Upgrading the Model Repository in a RACed Environment

To upgrade the Model Repository in an Oracle RAC environment, follow the same procedure as Installing the Model Repository on page 25. HP recommends that you test the connection as described in Testing the Connection from the SA host to the Database on page 29.

Setting the Oracle Database Server OS To Non-UTC Time

HP recommends that all servers including the Oracle database server be set to UTC. However, if you must set the database server to a non-UTC time zone, you can use the following procedure.

Requirements

- The steps in this section apply only to the Oracle *database server* used for SA. All other SA servers are required to be set to the UTC time zone.

- The process listed in this section is only for Oracle Standard and Enterprise Edition. *It is not certified for Oracle RAC.*

- This section is applicable for SA10.20. It is not applicable for lower versions of SA.

- The process is applicable to fresh core installations.

- Once the database is set up for non-UTC support as described in this section, the DBA must log on to the database using the `tnsnames.ora`'s service name/aliases. For example:

  `sqlplus <userid>/password@<tns_service_name>`

  Do not use local connections to log into the database. Refer to the section Logging into the Database on page 41 for more details.

- The SA application and Oracle RDBMS software must be installed on a supported OS. See the *SA Support and Compatibility Matrix* for supported OS's and versions.

- The Oracle database server's OS time zone can be set to a non-UTC time zone, for example PST, EST etc.

- The Primary and Secondary Core database servers can be set to different time zones.

Non-UTC Support Set Up for SA Oracle Database Server

The steps in this section refer to various machines, IP's and host names. The following table lists the Core environment used for the steps including Core names and machine information.

table 11   Core Names

Core Name	Facility ID/Core ID
Core 1	1
Core 2	2

table 12   Host Details

Host Name	IP Address	Purpose
Core 1		
db.pri.cpe.opsware.com	192.168.209.41	Primary Core Oracle database server
pri.aus.cpe.opsware.com	192.168.209.42	Primary Core Host
prisat1.aus.cpe.opsware.com	192.168.209.43	Primary Core Satellite
Core 2		
db.sla.cpe.opsware.com	192.168.210.24	Secondary Core Oracle database server
sla.aus.cpe.opsware.com	192.168.210.25	Secondary Core Host
slasat1.aus.cpe.opsware.com	192.168.210.26	Secondary Core Satellite

Non-UTC Setup Procedure

The following steps must be performed to set up a non-UTC time zone for the SA Oracle database server.

➤ If you allow the SA Installer to install the SA-supplied Oracle database, you can skip Steps 1 and 2.

1   Install the Oracle Software and create SA Oracle database

2   Create the database user `opsware_admin`.

➤ See System Requirements on page 1for Oracle database set up Requirements. You can install Oracle Software, and create the SA database using Oracle Corporation's install utilities or you can allow the SA Installer to install the SA-supplied Oracle database. The requirements for database user `opsware_admin` are listed in 5. Create the Database User opsware_admin on page 16.

3   Create a logon trigger and modify the `listener.ora` file.

table 13

Step	Action	Details	Comments
1	Create the logon trigger to set SA database user sessions to UTC.	This command must be executed from the database server. See Trigger for Setting SA DB User's Session Time Zone on page 42.	Example showing correct `CURRENT_TIMESTAMP` and `LOCALTIMESTAMP` after the `HPSA_LOGON_ZONE_UTC` trigger is created. `$sqlplus opsware_admin/ opsware_admin@ truth.AusPri` `SQL> SELECT DBTIMEZONE, SESSIONTIMEZONE FROM DUAL;` `DBTIME SESSIONTIMEZONE` `------ ------------` `-07:00 +`**00:00** `SQL> select CURRENT_TIMESTAMP, LOCALTIMESTAMP from dual;` `CURRENT_TIMESTAMP LOCALTIMESTAMP` `------------------------ --------------- ------------------------ -----` `30-JUN-14 05.48.51.146919 PM +00:00    30-JUN-14 05.48.51.146919 PM (correct time in UTC)`

table 13

Step	Action	Details	Comments

2	Modify the `listener.ora` file	**Note**: You can see a sample `listener.ora` in Sample listener.ora File for a Non-UTC Environment on page 41.
		
		a Shutdown the Oracle listener:
		`$lsnrctl stop`
		
		b Add ENVS variable to the file i.e. (`ENVS='TZ=UTC'`), UTC is required by SA.
		
		c Specify the PORT to be other than the default1521.
		
		d Restart the listener:
		`$lsnrctl start`
			a The database server's listener files was changed based on the Oracle document 'How To setup TNS listener to Show More Than one Time zone (Doc ID 399448.1)'
			
			b `listerner.ora`'s PORT number must be something other than the SA default 1521.
			
			c `listener.ora`'s connection must be via SID.
			
			d `listener.ora`'s connections are made using the static handler (`SID_DESC`).
			
			e Ensure that the database parameter `LOCAL_LISTENER` is not set to dynamically register any database that requires a different `TZ` or statically defined `SID_DESC` in the `listener.ora` file. To verify use `show parameter local_listener` in SQL*Plus.

table 13

Step	Action	Details	Comments

3	Verify the listener settings	To verify that the `listener.ora` settings:
		
		``
		$ su - oracle
		$ lsnrctl
		LSNRCTL> set displaymode verbose
		LSNRCTL> services
		Connecting to
		(ADDRESS=(PROTOCOL=tcp)(HOST=db.
		pri.cpe.opsware.com)(PORT=1524))
		Services Summary...
		Service "truth" has 1
		instance(s).
		  Instance "truth", status
		UNKNOWN, has 1 handler(s) for
		this service...
		    Handler(s):
		      "DEDICATED" established:2
		refused:0
		        LOCAL SERVER
		
		(ADDRESS=(PROTOCOL=beq)(PROGRAM=
		/u01/app/oracle/product/11.2.0/
		db_2/bin/
		oracle)(ENVS='TZ=UTC,ORACLE_HOME=/
		u01/app/oracle/product/11.2.0/
		db_2,ORACLE_SID=truth')(ARGV0=or
		acletruth)(ARGS='(LOCAL=NO)'))
		The command completed
		successfully
		LSNRCTL> exit
		``
			This step ensures that all the remote connections going through the listener will have the TZ=UTC setting.

4	Modify all `tnsnames.ora` files on database host and Core server hosts	Ensure that the `tnsnames.ora` files on the database server and all the SA Core component hosts have the correct PORT specified. ``` truth.AusPri=(DESCRIPTION=(ADDRE SS=(HOST=db.pri.cpe.opsware.com) (PORT=1524)(PROTOCOL=tcp))(CONNE CT_DATA=(SERVICE_NAME=truth))) ```	

4 Run the SA Installer and install the Model Repository and other SA components using the "Expert Interview" mode. The Expert interview mode allows you to set the Oracle listener port to a non-default port. By default the Oracle listener is set to 1521. When setting the database server to other than UTC time, `listener.ora`'s PORT number must be a value other than the SA default 1521.

➤ **Note**: If you install the Model Repository without setting the `listener.ora` file, the SA Installer issues a warning that the database server is not set to the required UTC time zone. Also, the data inserted in the database will have incorrect time. It is necessary to make the changes listed in the previous section before installing any of the SA components.

After installing all the SA components, logon to the database as the `spin` or `twist` database user and verify the time.:

table 14   Verify Time

Step	Actions	Details	Comments

1	Verify the time.
	
	Log on as one of the SA database users and verify the time from any of the SA Core component hosts or the database server.
		```set line 150```
		```col SESSIONTIMEZONE format a25```
		```col SYSTIMESTAMP format a40```
		```col CURRENT_TIMESTAMP format a40```
		```col LOCALTIMESTAMP format a30```
		```col SYSDATE_FORMATTED format a30```
		
		```SELECT DBTIMEZONE,```
		```SESSIONTIMEZONE FROM DUAL;```
		
		```SELECT```
		```TO_CHAR(SYSDATE,'DD-MON-YY```
		```HH:MI:SSAM') SYSDATE_FORMATTED,```
		```SYSTIMESTAMP FROM DUAL;```
		```Select CURRENT_TIMESTAMP,```
		```LOCALTIMESTAMP from dual;```
		
		```! date```
			Example of verifying the date-time from an SA component host:
			
			```$ su - oracle```
			```$sqlplus spin/```
			```<spin_password>@```
			```truth.AusPri```
			
			```SQL> set line 150```
			```SQL> col SESSIONTIMEZONE```
			```format a25```
			```SQL> col SYSTIMESTAMP```
			```format a40```
			```SQL> col CURRENT_TIMESTAMP```
			```format a40```
			```SQL> col LOCALTIMESTAMP```
			```format a30```
			```SQL> col SYSDATE_FORMATTED```
			```format a30```
			
			```SQL> SELECT DBTIMEZONE,```
			```SESSIONTIMEZONE FROM DUAL;```

table 14   Verify Time

Step	Actions	Details	Comments

2			Ensure that the SESSIONTIMEZONE **is** +00:00
			
			`DBTIME SESSIONTIMEZONE`
			`------`
			`------------------------`
			`-07:00 +`**`00:00`**
			
			`SQL> SELECT`
			`TO_CHAR(SYSDATE,'DD-MON-YY`
			`HH:MI:SSAM')`
			`SYSDATE_FORMATTED,`
			`SYSTIMESTAMP FROM DUAL;`
			
			`SYSDATE_FORMATTED`
			`SYSTIMESTAMP`
			`---------------------------`
			`-`
			`---------------------------`
			`-----------`
			`30-JUN-14 06:32:22PM`
			`30-JUN-14 06.32.22.609631 PM`
			`+00:00`
			
			`SQL> select`
			`CURRENT_TIMESTAMP,`
			`LOCALTIMESTAMP from dual;`
			
			`CURRENT_TIMESTAMP`
			`LOCALTIMESTAMP`
			`---------------------------`
			`-----------`
			`---------------------------`
			`-`
			`30-JUN-14 06.32.22.612416 PM`
			`+00:00      30-JUN-14`
			`06.32.22.612416 PM`

Post Installation Steps and Notes

In addition to the above, perform these tasks or be aware of the following after the installation.

Oracle's dba_scheduler jobs

SA has several jobs that are run using Oracle's `dba_scheduler`. See also Garbage Collection on page 43 and Changes to the Database Statistics Job on page 52 . The `dba_scheduler` jobs are run based on the database server time, but the job timings can be easily changed to a suitable time. Garbage Collection on page 43 lists the commands that can be used to change the job run time.

Oracle's alert.log

Oracle's `alert.log` and other log files will continue to show the machine/OS time.

SA Logs

All SA logs will show the time in UTC.

Logging into the Database

The DBA should always connect to the database using the `tnsnames.ora` entry, which connects to the database using the Oracle listener. You should not connect the database using a local connection. The `listener.ora` file has been modified to include the TZ=UTC parameter. The listener connections set the correct time zone value.

Correct:

```
sqlplus <userid>/password@<tns_service_name>
```

For example:

```
$sqlplus opsware_admin/opsware_admin@truth.AusPri
```

Incorrect:

```
sqlplus <userid>/password
```

For example:

```
$sqlplus opsware_admin/opsware_admin
```

Setting up a Secondary Core

After the primary core is setup in a non-UTC environment, it is important to wait for a day before setting up a secondary core. During the secondary core creation, if 'add_dc_to_mesh' runs into errors similar to the following in Oracle's alert.log file:

```
ORA-31693: Table data object "TRUTH"."<table_name>" failed to load/unload
and is being skipped due to error:
ORA-02354: error in exporting/importing data
ORA-08186: invalid timestamp specified
ORA-06512: at "SYS.TIMESTAMP_TO_SCN", line 1
```

...then, wait for a day, and then give the following grants in sqlplus and kickoff the installer's add_dc_to_mesh command once again:

```
$ su - oracle
$ sqlplus  "/ as sysdba"
SQL> grant create session to vault, gcadmin, twist, spin, lcrep, aaa,
truth;
SQL> exit;
```

Sample listener.ora File for a Non-UTC Environment

The following is a sample `listener.ora` file from a primary core:

```
Sample listener.ora from primary core
# File:      listener.ora
# Purpose:   Listener specifications
# Notes:     None

LISTENER =
 (ADDRESS_LIST=
```

```
            (ADDRESS=(PROTOCOL=tcp)(HOST=db.pri.cpe.opsware.com)(PORT=1524))
            (ADDRESS=(PROTOCOL=ipc)(KEY=PNPKEY)))

SID_LIST_LISTENER=
   (SID_LIST=
     (SID_DESC=
         (ENVS='TZ=UTC')
         (SID_NAME=truth)
         (ORACLE_HOME=/u01/app/oracle/product/12.1.0/db_1)
     )
   )
LOG_DIRECTORY_LISTENER=/u01/app/oracle/product/12.1.0/db_1/network/log
LOG_FILE_LISTENER=listener.log
TRACE_DIRECTORY_LISTENER=/u01/app/oracle/product/12.1.0/db_1/network/trace
TRACE_FILE_LISTENER=listener.trc
INBOUND_CONNECT_TIMEOUT_LISTENER=120
# subscribe_for_node_down_event entry should be removed for RAC env.
SUBSCRIBE_FOR_NODE_DOWN_EVENT_LISTENER=OFF
#LOCAL_OS_AUTHENTICATION_LISTENER=OFF

#----ADDED BY TNSLSNR 04-OCT-2005 23:06:20---
SAVE_CONFIG_ON_STOP_LISTENER = ON
#-------------------------------------------

#----ADDED BY TNSLSNR 04-OCT-2005 23:09:20---
PASSWORDS_LISTENER = CEE9A6452943F605
#-------------------------------------------
```

Sample tsnames.ora File for a Non-UTC Environment

The following is sample `tnsnames.ora` file from primary core:

```
truth.AusPrimary=(DESCRIPTION=(ADDRESS=(HOST=db.pri.cpe.opsware.com)(PORT=152
4)(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
truth.AusSecondary=(DESCRIPTION=(ADDRESS=(HOST=192.168.209.42)(PORT=20002)(PR
OTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

➤ **Note**: The `truth.AusSecondary` entry is added by the SA installer and is used for the Multimaster Mesh replication process.

Trigger for Setting SA DB User's Session Time Zone

```
$ su - oracle
$ sqlplus / as sysdba
CREATE OR REPLACE TRIGGER HPSA_LOGON_ZONE_UTC
AFTER LOGON ON DATABASE

DECLARE
            v_user varchar2(30):=user;
            sql_stmt1 varchar2(256) :='ALTER SESSION SET
TIME_ZONE=''+00:00''';
begin
            if
            (
```

```
                                        USER IN
                                        (
                                                    'AAA',
                                                    'AAA_USER',
                                                    'GCADMIN',
                                                    'LCREP',
                                                    'OPSWARE_ADMIN',
                                                    'OPSWARE_PUBLIC_VIEWS',
                                                    'TRUTH',
                            'SPIN',                 'TWIST',

                                                    'VAULT'
                                        )
                            )
                            THEN
                            execute immediate sql_stmt1;
            end if;
end;
/
/
```

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_scheduler_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 unsynchronized, 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        02-NOV-14
5                 LAST_DATE_CHANGE_LOG 05-JUN-14
6                 DAYS_AUDIT_LOG       180          (These are the audit logs)
7                 LAST_DATE_AUDIT_LOG  05-JUN-14
8                 DAYS_WLM             30           (These are completed WLM jobs)
9                 LAST_DATE_WLM        02-NOV-14
```

► As of SA 9.10, the `DAY_TRAN` parameter that controlled retention time for transactions was removed. To control transaction retention time, instead use the system configuration parameter `vault.garbageCollector.daysToPreserve`.

Select the **Administration** tab in the SA Client, then select System Configuration in the navigation panel. Select Model Repository Multimaster Component. Locate and change the value of the above parameter. The default value is 7. Select the Save button to save your change.

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_SCHEDULER_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> set line 200
SQL> col job_name format a50
SQL> col owner format a14
SQL> col last_date format a17
SQL> col next_date format a17
SQL> col job_action format a50

SQL>select job_name, owner, to_char(LAST_START_DATE, 'MM/DD/YY HH:MI:SS')
last_date,to_char(next_run_date, 'MM/DD/YY HH:MI:SS') next_date, job_action
from dba_scheduler_jobs where owner='GCADMIN';
```

```
JOB_NAME                 OWNER    LAST_DATE          NEXT_DATE          JOB_ACTION
------------------------ -------  -----------------  -----------------  ---------------------------
WLMPURGE_GC              GCADMIN  04/02/12 09:00:02  04/04/12 09:00:00  WLMPURGE.GC_JOBS
STORAGEINITIATORPURGE_GC GCADMIN  04/02/12 09:47:30  04/03/12 10:47:30  STORAGEINITIATORPURGE.GC_STORAGE
                                                                        INITIATORS
AUDITPURGE_GC            GCADMIN  04/02/12 09:00:02  04/04/12 09:00:00  AUDITPURGE.GC_AUDITLOGS
CHANGELOGPURGE_GC        GCADMIN  04/02/12 09:00:02  04/04/12 09:00:00  CHANGELOGPURGE.GC_CHANGELOGS
WAYPURGE_GC             GCADMIN  04/02/12 09:00:02  04/04/12 09:00:00  WAYPURGE.GC_SESSIONS
```

where:

`WAYPURGE.GC_SESSIONS` - Performs `sessions` garbage collection

`CHANGELOGPURGE.GC_CHANGELOGS` - Performs `changelogs` garbage collection

`AUDITPURGE.GC_AUDITLOGS` - Performs `auditlogs` garbage collection

`STORAGEINITIATORPURGE.GC_STORAGEINITIATORS` - Performs `storage data` garbage collection

`WLMPURGE.GC_JOBS` - Performs WLM 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_scheduler.run_job('<job_name_value>');
```

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

```
SQL> exec dbms_scheduler.run_job('WAYPURGE_GC');
```

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`:

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

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

```
SQL>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:

```
SQL>Col file_name format a50
SQL>Col status format a10
SQL>Set line 200
SQL>Select file_id, status, bytes, file_name from dba_data_files order by
SQL>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/
12.1.0/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_BASE/diag/rdbms/<SID>/<SID>/trace/
```

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

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

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

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


SQL>SELECT
    nvl(df.tablespace_name,'UNKOWN') name, df.mbytes_alloc Mbytes,
    df.mbytes_alloc-nvl(fs.mbytes_free,0) used, nvl(fs.mbytes_free,0)
free,
    ((df.mbytes_alloc-nvl(fs.mbytes_free,0)) / df.mbytes_alloc) * 100
pct_used,
    nvl(df.largest,0) largest, nvl(df.mbytes_max,df.mbytes_alloc)
Max_Size,
    ((df.mbytes_alloc-nvl(fs.mbytes_free,0)) / df.mbytes_max) * 100
pct_max_used
FROM
    (  SELECT tablespace_name, sum(bytes)/1024/1024 Mbytes_alloc,
max(bytes)/1024/1024 largest,

sum(decode(autoextensible,'YES',greatest(bytes,maxbytes),bytes))/1024/
1024 Mbytes_max
        FROM
            dba_data_files GROUP BY tablespace_name
    ) df,
    ( SELECT tablespace_name, sum(bytes)/1024/1024 Mbytes_free
        FROM dba_free_space GROUP BY tablespace_name
    ) fs
WHERE
    df.tablespace_name = fs.tablespace_name(+)
UNION
SELECT
    D.tablespace_name name, D.mbytes_alloc Mbytes, ((ss.used_blocks *
F.block_size) / 1024 / 1024) used,
    D.mbytes_alloc - ((ss.used_blocks * F.block_size) / 1024 / 1024) free,
    ((D.mbytes_alloc-nvl((D.mbytes_alloc - ((ss.used_blocks *
F.block_size) / 1024 / 1024)),0)) / D.mbytes_alloc) * 100 pct_used,
    nvl(((G.max_blocks * F.block_size) / 1024 / 1024),0) largest,
Max_Mbytes Max_Size,
```

```
        ((D.mbytes_alloc-nvl((D.mbytes_alloc - ((ss.used_blocks *
F.block_size) / 1024 / 1024)),0)) / D.Max_Mbytes) * 100 pct_pct_used
FROM
    (   SELECT tablespace_name, used_blocks, free_blocks, max_size
        FROM v$sort_segment
    ) ss,
    (   SELECT tablespace_name, sum(bytes)/1024/1024 Mbytes_alloc,

sum(decode(autoextensible,'YES',greatest(bytes,maxbytes),bytes))/1024/
1024 Max_Mbytes
        FROM dba_temp_files GROUP BY tablespace_name
    ) D,
    (   SELECT B.name, C.block_size, SUM (C.bytes) / 1024 / 1024 mb_total
        FROM v$tablespace B, v$tempfile C
        WHERE B.ts#= C.ts# GROUP BY B.name, C.block_size
    ) F,
    (   SELECT B.name, max(blocks) max_blocks, sum(blocks) total_blocks
        FROM v$tablespace B, v$tempfile C
        WHERE B.ts#= C.ts# GROUP BY B.name
    ) G
WHERE ss.tablespace_name = D.tablespace_name and ss.tablespace_name =
F.name and ss.tablespace_name = G.name;
```

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`:

```
SQL>Select file_id, bytes, file_name from dba_data_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.

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

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

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

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

Enable Oracle Automatic Optimizer Statistics Collection

As of SA 10.0 the schema and index statistics collection for SA database user `AAA`, `TRUTH` etc. has been moved from `dba_jobs` to Oracle's Automatic Optimizer Statistics Collection.

SA relies on Oracle's Automatic Optimizer statistics collection to collect schema statistics used to avoid database performance degradation. By default, Oracle's Automatic optimizer statistics collection should be enabled.

To verify that the Oracle Automatic optimizer statistics collection is enabled, perform the following steps:

1   Enter the following commands in SQL*Plus:

```
# su - oracle
# sqlplus "/ as sysdba"

SQL>set line 200
SQL>col status format a10
SQL>SELECT status FROM dba_autotask_client where client_name='auto
optimizer stats collection';
```

The output from the above statement should be as follows:

```
STATUS
----------
ENABLED
```

2   If the status is not ENABLED, execute the following statement to enable Oracle's Automatic Optimizer statistics collection.

```
SQL>EXEC DBMS_AUTO_TASK_ADMIN.ENABLE(client_name => 'auto optimizer stats
collection',operation => NULL, window_name => NULL);
```

Manually Collecting Schema Statistics

Although you have moved to the Oracle's Automatic Optimizer Statistics Collection, there are times when you may need to collect statistics manually.

To collect the statistics manually:

Enter the following commands in SQL*Plus:

```
# su - oracle
# sqlplus "/ as sysdba"


SQL> begin
    dbms_stats.gather_schema_stats (ownname=>'TRUTH',
        estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE,
        cascade=>TRUE,
        options=>'GATHER AUTO');
end;
/
```

Replace the ownname from TRUTH to the other SA schemas like AAA, LCREP, GCADMIN, OPSWARE_ADMIN.

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

When the Model Repository is installed, the SA Installer sets up the System/Index Statistics and the Garbage Collection jobs in Oracle's `dba_scheduler_jobs` which then runs these jobs at specified time-intervals. The jobs perform system/ index statistics collection and garbage collection. If the system/ index statistics collection jobs do not run successfully, database performance degrades. If the garbage collection jobs do not run, old data accumulates and requires additional disk space. Performance can also be affected.

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

1   Enter the following commands in SQL*Plus:

```
SQL>set line 200
SQL>col job_name format a50
SQL>col owner format a14
SQL>col last format a17
SQL>col next format a17
SQL>col state format a10
SQL>col job_action format a50

SQL>select job_name, owner, to_char(LAST_START_DATE, 'MM/DD/YY HH:MI:SS')
last, to_char(next_run_date, 'MM/DD/YY HH:MI:SS') next, state, job_action
from dba_scheduler_jobs where owner in ('OPSWARE_ADMIN', 'LCREP',
'GCADMIN');
```

In the output generated from the preceding statement, the value of the `JOB_ACTION` column indicates the type of job. The jobs owned by `GCADMIN` perform the garbage collection. The job owned by `LCREP` performs index statistics collection and the job owned by `OPSWARE_ADMIN` performs system statistics collection. Sample output looks like this:

```
JOB_NAME                     OWNER          LAST              NEXT              STATE      JOB_ACTION
-------------------          ------------   ----------------  ----------------  ---------  ----------------------
WLMPURGE_GC                  GCADMIN        12/02/14 09:00:01 12/03/14 09:00:00 SCHEDULED  WLMPURGE.GC_JOBS
STORAGEINITIATORPURGE_GC     GCADMIN        12/02/14 09:00:01 12/03/14 09:00:00 SCHEDULED  STORAGEINITIATORPURGE.GC
                                                                                           _STORAGEINITIATORS
AUDITPURGE_GC                GCADMIN        12/02/14 09:00:01 12/03/14 09:00:00 SCHEDULED  AUDITPURGE.GC_AUDITLOGS
CHANGELOGPURGE_GC            GCADMIN        12/02/14 09:00:01 12/03/14 09:00:00 SCHEDULED  CHANGELOGPURGE.GC_CHANGE
                                                                                           LOGS
WAYPURGE_GC                  GCADMIN        12/02/14 09:00:01 12/03/14 09:00:00 SCHEDULED  WAYPURGE.GC_SESSIONS
LCREP_INDEX_STATS            LCREP          12/02/14 11:00:00 12/04/14 11:00:00 SCHEDULED  gather_lcrep_stats
OPSWARE_ADMIN_SYSTEM_STATS   OPSWARE_ADMIN  12/01/14 06:00:02 12/08/14 06:00:00 SCHEDULED  gather_opsware_admin_sys
                                                                                           _stats

7 rows selected.
```

where:

- JOB_NAME - name of the job
- OWNER - 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
- STATE - The status of the scheduled job:
  - disabled - The job is disabled
  - scheduled - The job is scheduled to be executed

- — running - The job is currently running

- — completed - The job has completed, and is not scheduled to run again

- — broken - The job is broken

- — failed - The job was scheduled to run once and failed

- — retry scheduled - The job has failed at least once and a retry has been scheduled to be executed

- — succeeded - The job was scheduled to run once and completed successfully

- — JOB_ACTION - the procedure that the job runs

Changes to the Database Statistics Job

Starting with Oracle 10g, the DBMS_JOB package was superceded by the improved Oracle Scheduler (dbms_scheduler) package. Although Oracle still supports the DBMS_JOB package for backward compatibility, Oracle will make no further enhancements to the package. Since the DBMS_SCHEDULER provides better functionality, all the SA jobs that used the DBMS_JOB package have been redesigned in this release to use the DBMS_SCHEDULER package. The affected jobs can be found in the dba_scheduler_jobs table. These changes are only relevant to new SA 10.x Cores and cores upgraded to SA 10.x.

To view the jobs and changes made, you can run the following from SQL*Plus:

```
# su - oracle
# sqlplus "/ as sysdba"
SQL>set line 200
SQL>col owner format a14
SQL>col job_action format a50
SQL>col job_name format a50
SQL>select job_name, owner, job_action from dba_scheduler_jobs where owner in
('OPSWARE_ADMIN', 'LCREP', 'GCADMIN');
```

Your output should be as follows:

```
JOB_NAME                        OWNER          JOB_ACTION
--------------------------      -----------    -------------------------
WLMPURGE_GC                     GCADMIN        WLMPURGE.GC_JOBS
STORAGEINITIATORPURGE_GC        GCADMIN        STORAGEINITIATORPURGE.GC_STORAGEINITIATORS
AUDITPURGE_GC                   GCADMIN        AUDITPURGE.GC_AUDITLOGS
CHANGELOGPURGE_GC               GCADMIN        CHANGELOGPURGE.GC_CHANGELOGS
WAYPURGE_GC                     GCADMIN        WAYPURGE.GC_SESSIONS
LCREP_INDEX_STATS               LCREP          gather_lcrep_stats
OPSWARE_ADMIN_SYSTEM_STATS      OPSWARE_ADMIN  gather_opsware_admin_sys_stats

        7 rows selected.
```

Running dba_scheduler_jobs manually

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

```
SQL> grant create session to lcrep, gcadmin;
```

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_name_value` with the values of the `job` column displayed by the same select statement.

```
SQL> connect <schema_user_value>/<password>
SQL> exec dbms_scheduler.run_job('<job_name_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 lcrep, gcadmin;
```

Changing the Time Jobs are Run

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

```
# su - oracle
$ sqlplus "/ as sysdba"
SQL>set line 300
SQL>col job_name format a30
SQL>col owner format a14
SQL>col last format a17
SQL>col next format a17
SQL>col repeat_interval format a40
SQL>col job_action format a30

SQL>select job_name, owner, to_char(LAST_START_DATE, 'MM/DD/YY HH:MI:SS')
last, to_char(next_run_date, 'MM/DD/YY HH:MI:SS') next, repeat_interval,
job_action from dba_scheduler_jobs where owner in ('OPSWARE_ADMIN', 'LCREP',
'GCADMIN');
```

The above statement provides information about a job. Note the job name and the owner 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_NAME              OWNER    LAST             NEXT             REPEAT_INTERVAL        JOB_ACTION
-----------           -------  --------------   --------------   --------------------   ---------------------

WLMPURGE_GC           GCADMIN  12/02/14 09:00:01 12/03/14 09:00:00 TRUNC(SYSDATE+1)+9/24   WLMPURGE.GC_JOBS
STORAGEINITIATORPURGE GCADMIN  12/02/14 09:00:01 12/03/14 09:00:00 TRUNC(SYSDATE+1)+9/24   STORAGEINITIATORPURGE.GC
_GC                                                                                      _STORAGEINITIATORS
AUDITPURGE_GC         GCADMIN  12/02/14 09:00:01 12/03/14 09:00:00 TRUNC(SYSDATE+1)+9/24   AUDITPURGE.GC_AUDITLOGS
CHANGELOGPURGE_GC     GCADMIN  12/02/14 09:00:01 12/03/14 09:00:00 TRUNC(SYSDATE+1)+9/24   CHANGELOGPURGE.GC_CHANGE
                                                                                         LOGS
WAYPURGE_GC           GCADMIN  12/02/14 09:00:01 12/03/14 09:00:00 TRUNC(SYSDATE+1)+9/24   WAYPURGE.GC_SESSIONS
LCREP_INDEX_STATS     LCREP    12/02/14 11:00:00 12/04/14 11:00:00 TRUNC(SYSDATE+2)+11/24  gather_lcrep_stats
OPSWARE_ADMIN_SYSTEM_ OPSWARE_ 12/01/14 06:00:02 12/08/14 06:00:00 TRUNC(SYSDATE+1) + 18/  gather_opsware_admin_sys
STATS                 ADMIN                                       24 +                   _stats
                                                                  mod(abs(to_number(to_c
                                                                  har(sysdate + 1,'D'))
                                                                  - 7) + 2,  7")
```

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

```
# su - oracle
$ sqlplus "/ as sysdba"
SQL> grant create session to lcrep;
Grant succeeded.
```

In the example:

- job name=LCREP_INDEX_STATS

- owner = lcrep

In this example, the job LCREP_INDEX_STATS runs at 11:00 a.m. UTC. To change this to 9:00 a.m. UTC, the command is:

```
SQL> connect lcrep/<password_for_lcrep>
Connected.

SQL> exec dbms_scheduler.set_attribute('LCREP_INDEX_STATS',
attribute=>'REPEAT_INTERVAL', value=>'TRUNC(SYSDATE+2)+9/24');
```

Monitoring Database Users

To monitor database users, perform the following steps:

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

```
# su - oracle
$ sqlplus "/ as sysdba"
SQL>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:

SQL>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 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;
```

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.

- **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
```

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

Useful SQL

The following SQL commands help you manage information in the Oracle database.

BIN$ Objects

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

```
SQL>show parameter recyclebin;
SQL>SELECT owner,original_name,operation,type FROM dba_recyclebin;
connect <owner>/password
SQL>purge recyclebin;
```

or

```
SQL>purge table BIN$xxx;
```

By default, `recyclebin` is set to `OFF`.