
A Oracle Setup for the Model Repository

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

Oracle RDBMS Install Basics

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

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

The SA distribution Media includes a separate Oracle 11g RDBMS software and database installation dual layer DVD. You can simply mount this DVD on the server you plan to use to host the Model Repository and run the installation. See [Installing the HP-Supplied Oracle RDBMS Software and Database](#) on page 195 in this chapter for information about the installation steps

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

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

- [Supported Oracle Versions](#) on page 188
- [Hardware Requirements](#) on page 189
- [Operating System Requirements](#) on page 191
- [The SA Installer HP-Supplied RDBMS Installation Process](#) on page 197
- [Pre-Installation Tasks \(Oracle Universal Installer\)](#) on page 199
- [Manually Creating the Oracle Database](#) on page 201
- [Post-Oracle Installation Tasks](#) on page 205



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

Supported Oracle Versions

Support for the Model Repository is limited to certain versions of Oracle running on certain versions of operating systems. HP strongly recommends that you also apply the latest Oracle CPU patches. For manual installations, SA supports both the Oracle Standard Edition, Standard Edition One, and the Oracle Enterprise Edition. [Table 42](#) lists the supported Oracle versions.

Table 42 Supported Operating Systems and Oracle Versions

Operating System	Supported Oracle Versions (Standard or Enterprise Edition)
SunOS 10 x86_64	10.2.0.2, 10.2.0.4, 11.1.0.7
Red Hat Enterprise Linux AS 3 x86_32	10.2.0.2, 10.2.0.4
Red Hat Enterprise Linux AS 4 x86_64	10.2.0.2, 10.2.0.4, 11.1.0.7
Red Hat Enterprise Linux AS 5 x86_64	10.2.0.4, 11.1.0.7
SUSE Linux Enterprise Server 10 x86_64	10.2.0.4, 11.1.0.7



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

Multiple Oracle Versions and Multimaster Cores

For the database export to succeed during the installation of a Multimaster core, the version of the target database cannot be 10.x if the source database is 11.x. [Table 43](#) lists these allowed version combinations.

Table 43 Database Versions Allowed for Multimaster

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

Hardware Requirements

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

Linux Requirements

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

- The recommended physical memory is 8 GB. An HP-supplied Oracle installation will use around 1.6-1.7GB of memory. The Oracle SGA memory can be increased after database installation. You can use the following command to check memory status:

```
grep MemTotal /proc/meminfo
```

- Recommended swap space: between 1-2 GB RAM, (1.5 times the size of RAM, between 2-8 GB RAM, equal to the size of RAM).

You can use the following command to check swap space:

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

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

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

```
df -k /dev/shm/
```

- Free tmp space should be 400MB or more

You can use the following command to check tmp space:

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

- Required Kernel version:

— Red Hat AS 4: 2.6.9 or later

— Red Hat AS 5: 2.6.18 or later

— SUSE Linux 10: 2.6.16.21 or later

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

```
uname -r
```

You can use the following command to check the platform:

```
uname -mi
```

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

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

Solaris Requirements

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

- The recommended physical memory is 8 GB. An HP-supplied Oracle installation will use around 1.6-1.7GB of memory. The Oracle SGA memory can be increased after database installation.

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

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

- Recommended swap space: between 1-2 GB RAM, (1.5 times the size of RAM, between 2-8 GB RAM, equal to the size of RAM).

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

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

- Free tmp space should be 400MB or more

You can use the following command to check tmp space:

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

- Required operating system version is: 5.10

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

```
uname -a
```

- System architecture should show 64-bit sparcv9 kernel modules

You can use the following command to check system architecture:

```
isainfo -kv
```

Model Repository (Database) Disk Space Requirements

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

As a benchmark figure, you should allow an additional 3.1 GB of database storage for every 1,000 servers in the facility that SA manages. When sizing the tablespaces, follow the general guidelines described in [Table 44](#). If you need to determine a more precise tablespace sizing, contact your technical support representative.

Table 44 **Tablespace Sizes**

Tablespace	Minimum Size
AAA_DATA	256 MB
AAA_INDX	256 MB
AUDIT_DATA	256 MB
AUDIT_INDX	256 MB
LCREP_DATA	1,500 MB
LCREP_INDX	800 MB
TRUTH_DATA	700 MB
TRUTH_INDX	400 MB
STRG_DATA	700 MB
STRG_INDX	400 MB

Hostname Setup

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

Linux/Red Hat Linux

```
# ping <hostname>
```

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

```
# hostname
```

SUSE Linux

```
# hostname -f
```

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

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

Operating System Requirements

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



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

Required Packages for Red Hat Enterprise Linux AS 4 x_64

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

Table 45 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.15.92.0.2
compat-libstdc++-33	i386	3.2.3
compat-libstdc++-33	x86_64	3.2.3
elfutils-libelf	x86_64	0.97
elfutils-libelf-devel	x86_64	0.97
gcc	x86_64	3.4.5
gcc-c++	x86_64	3.4.5
glibc	i686	2.3.4-2.19
glibc	x86_64	2.3.4-2.19

Table 45 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64

Required Packages	Architecture	Version
glibc-common	x86_64	2.3.4
glibc-devel	i386	2.3.4
glibc-devel	x86_64	2.3.4
libaio	i386	0.3.105
libaio	x86_64	0.3.105
libaio-devel	x86_64	0.3.105
libgcc	i386	3.4.5
libgcc	x86_64	3.4.5
libstdc++	x86_64	3.4.5
libstdc++	i386	3.4.5
libstdc++-devel	x86_64	3.4.5
make	x86_64	3.80
sysstat	x86_64	5.0.5

Required Packages for Red Hat 5 Server x86_64

The following packages are required for Oracle 11g on Red Hat Enterprise Linux 5 Server x86_64. These packages must be the versions listed or higher.

Table 46 Required Packages for Red Hat 5 Server x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.17.50.0.6
compat-libstdc++-33	i386	3.2.3
compat-libstdc++-33	x86_64	3.2.3
elfutils-libelf	x86_64	0.125
elfutils-libelf-devel	x86_64	0.125
gcc	x86_64	4.1.1
gcc-c++	x86_64	4.1.1
glibc	i686	2.5-12
glibc	x86_64	2.5-12
glibc-common	x86_64	2.5
glibc-devel	i386	2.5-12
glibc-devel	x86_64	2.5

Table 46 Required Packages for Red Hat 5 Server x86_64

Required Packages	Architecture	Version
libaio	i386	0.3.106
libaio	x86_64	0.3.106
libaio-devel	x86_64	0.3.106
libgcc	i386	4.1.1
libgcc	x86_64	4.1.1
libstdc++	i386	4.1.1
libstdc++	x86_64	4.1.1
libstdc++-devel	x86_64	4.1.1
make	x86_64	3.81
sysstat	x86_64	7.0.0

Required Packages for SUSE Linux Enterprise Server 10 x86_64

The following packages are required for Oracle 11g on SUSE Linux Enterprise Server 10 x86_64. These packages must be the versions listed or higher

Table 47 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.16.91.0.5
compat-libstdc++	x86_64	5.0.7-22.2
gcc	x86_64	4.1.0
gcc-c++	x86_64	4.1.0
glibc	x86_64	2.4-31.2
glibc-32bit	x86_64	2.4-31.2
glibc-devel	x86_64	2.4
glibc-devel-32bit	x86_64	2.4
libaio	x86_64	0.3.104
libaio-32bit	x86_64	0.3.104
libaio-devel	x86_64	0.3.104
libelf	x86_64	0.8.5
libgcc	x86_64	4.1.0
libstdc++	x86_64	4.1.0

Table 47 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64

Required Packages	Architecture	Version
libstdc++-devel	x86_64	4.1.0
make	x86_64	3.80
sysstat	x86_64	6.0.2

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

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

Required Packages for Solaris 10

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

Oracle 11g

Solaris 10 must have the following packages for Oracle 11g:

```
SUNWarc
SUNWbash
SUNWbtool
SUNWhea
SUNWlibC
SUNWlibm
SUNWlibms
SUNWsprt
SUNWtoo
SUNWilof
SUNWilcs
SUNWil5cs
SUNWxfnt
SUNWpool
SUNWpoolr
```

Required Patches for Solaris 10

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

Table 48

Patch	Version	Comment
127111	02	#libc patch
137111	04	#kernel patch

Required Patch for Manual Oracle 11g Installations

If you plan to install Oracle 11g (for the Model Repository) using the Oracle Universal Installer or use an existing Oracle 11g installation, you must ensure that Oracle Bug Patch 8300752 has been applied to that database. This patch has already been applied to the HP-supplied Oracle RDBMS.

Oracle 10g on Solaris 10 Servers

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

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

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

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

Installing the HP-Supplied Oracle RDBMS Software and Database



If you are manually installing the Oracle RDBMS software and database, skip this section and go to [Pre-Installation Tasks \(Oracle Universal Installer\)](#) on page 199.

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

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

You will see a screen similar to this:

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

```
Please select the interview mode. Simple mode uses default values for many of
the configuration parameters. Advanced mode allows you to fully configure the
installation.
```

- ```
1 - Simple Interview Mode
2 - Advanced Interview Mode
```

```
Please select the interview mode from the menu, type 'h' for help, 'q' to
quit: 1
```

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

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

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

Enter 1 for the Simple Interview or 2 for the Advanced Interview (which allows you to modify additional parameters, if necessary). For example, if you choose the Simple Interview, you will see a screen similar to this:

```
Parameter 1 of 3 (truth.oePwd)Please enter the password for the opware_admin
user. This is the password used to connect to the Oracle database.:
opware_admin
Validating... OK.
```

```
Parameter 2 of 3 (truth.servicename)Please enter the service name (aka TNS
name) of the Model Repository instance in the facility where Opware Installer
is being run [truth]:
Validating... OK.
```

```
Parameter 3 of 3 (truth.port)Please enter the port on which the Model
Repository database is listening. [1521]:
Validating... OK.
```

All parameters have values. Do you wish to finish the interview? (y/n): y

Concluding interview.

Interview complete.

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

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

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

You will then see this screen:

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

Selection:

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

# The SA Installer HP-Supplied RDBMS Installation Process

The SA Installer uses the following process when installing the HP-supplied Oracle RDBMS .

## SA Installer Changes to Database Configuration and Files

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

- Checks that all requirements are met on the host server (see [Hardware Requirements](#) on page 189 and [Operating System Requirements](#) on page 191).
- Sets certain kernel parameters to required values. For details about these parameter changes, see [Kernel Parameter Values](#) on page 198.
- Creates the Unix user `oracle` locally in `/etc/passwd`.
- Creates the Unix groups `dba` and `oinstall` locally in `/etc/group`.
- Sets the `$ORACLE_HOME` environment variable to the following directory:  
`/u01/app/oracle/product/11.1.0/db_1`
- Sets the `$ORACLE_SID` environment variable to `truth`.
- Creates an Oracle instance with the required `init.ora` parameters.
- Creates the tablespaces and data and index files under the following directories:  
`/u01/oradata/truth`  
`/u02/oradata/truth`  
`/u03/oradata/truth`  
`/u04/oradata/truth`
- The system administrator can configure the `/u01`, `/u02`, `/u03`, `/u04` directories before installing the Oracle RDBMS software.
- Gets the service name (TNS name) from the SA Installer interview (`truth.servicename` prompt) and inserts it into the `tnsnames.ora` file in `$ORACLE_HOME/network/admin` and `/var/opt/oracle`. The SA Installer changes the value of the `host` parameter to the value returned by the Unix `hostname` command.
- In the `/$ORACLE_HOME/network/admin/listener.ora` file, changes the value of the `host` parameter to the value returned by the Unix `hostname` command.  
  
The listener is password protected and OS authenticated. (The default password is `opsware`.) By default, it listens on port 1521.
- Creates the `/etc/init.d/opsware-oracle` script, which you can use to start up and shut down the database and listener.  
  
This script is linked to corresponding scripts in the `/etc/rc*.d` directories..
- Creates the user `opsware_admin` with the required privileges.
- After installation is complete, you can examine the logs that are created here:  
`/var/log/opsware/install_opsware`

## Kernel Parameter Values

During the SA Installer installation of the HP-supplied Oracle RDBMS , the installation also sets the values for certain parameters in various files. If you manually install the Oracle database, or use an existing database, you must insure that these values are specified correctly. This section lists the parameters set by the installer that can be changed without adversely affecting SA.

### Changing Kernel Parameter Values for Linux

This section identifies the kernel parameters you can change for Red Hat Linux Enterprise Server AS 4 x86\_64, Red Hat Linux Enterprise Server AS 5 x86\_64, and SUSE Linux Enterprise Server 10 x86\_64.

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

```
kernel.shmmax=2147483648
kernel.shmall=2097152
kernel.shmmni=4096
kernel.sem=250 32000 100 128
net.core.rmem_default=4194304
net.core.wmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_max=262144
fs.file-max=65536
net.ipv4.ip_local_port_range=1024 65000
net.ipv4.tcp_wmem=262144 262144 262144
net.ipv4.tcp_rmem=4194304 4194304 4194304
```

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

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

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

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

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

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

### Additional Modifiable SUSE Kernel Parameter Values

This section identifies additional required settings for SUSE Linux Enterprise Server 10 x86\_64 when running Oracle 11g:

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

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

For example, where the oinstall group GID is 501:

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

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

```
vm.hugetlb_shm_group=501
```



---

Only one group can be defined as the vm.hugetlb\_shm\_group.

---

## Changing Kernel Parameter Values for Solaris 10

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

- 1 Enter set noexec\_user\_stack=1 in /etc/system.
- 2 Run the following commands:

```
projadd -U oracle -K "project.max-shm-memory=(priv,2048MB,deny) "
user.oracle

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

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

echo "oracle::::project=user.oracle" >> /etc/user_attr
```
- 3 Use the vi editor for /etc/project and /etc/user\_attr to verify the changes made in step 2.

## Pre-Installation Tasks (Oracle Universal Installer)



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

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

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

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



The sample files referenced in these steps can be obtained from your HP Support representative. See [Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files](#) on page 201.

- 1 Verify that the server has the hardware and software listed in [Hardware Requirements](#) on page 189 and [Operating System Requirements](#) on page 191.
- 2 Obtain the sample files and unzip them.

- 3 Set the kernel parameters.

The easiest way to set these parameters is by copying and editing the following sample files:

```
kernel_params_redhat.txt
kernel_params_solaris.txt
```

These two files contain instructions, Unix commands, and lines of text for configuration files.

- 4 Create the required Unix users and groups by running the following commands. (If you use a directory different than `/u01/app/oracle`, modify the commands accordingly):

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

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

```
bash_profile
profile
```

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

## Baseline Data Installation

The following steps are required for Red Hat Enterprise Linux AS 4 x86\_64, Red Hat Enterprise Linux AS 5 x86\_64, Sun Solaris x86\_64, and SUSE Enterprise Linux 10 x86\_64 because, during a Model Repository fresh install, baseline data is not inserted completely. Oracle does not insert some of the baseline data in `role_classes` and other tables and there are no errors, failures or trace files generated by Oracle. This is a silent failure and an intermittent problem. The Model Repository installs successfully because there are no error messages returned from Oracle, but later the Data Access Engine (Spin) install fails due to missing baseline data.

Before installing Oracle on Red Hat Enterprise Linux AS 4 x86\_64, Red Hat Enterprise Linux AS 5 x86\_64, Sun Solaris x86\_64, or SUSE Enterprise Linux 10 x86\_64 on the Model Repository (`truth`)/database host, run the following commands:

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

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

Now, you can run the SA Installer and install the Model Repository.

## Oracle XDB Component Installation Requirements

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

# Manually Creating the Oracle Database



---

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

---

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

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

The SQL scripts that must be run or edited are:

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

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

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

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



---

The database must have the character set UTF8 available.

---

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

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

See [Tablespace Sizes](#) on page 40 in Chapter 2 for information about for additional tablespace sizing information.

- **CreateDBCatalog.sql:** Runs Oracle scripts to create data system catalog objects.

- **JServer.sql:** Sets up the Oracle Java environment.
- **CreateAdditionalDBFiles.sql:** Adds data and index files to certain tablespaces and allocates additional disk space. This script is optional, but recommended.
- **CreateUserOpware\_Admin.sql:** Creates the `opware_admin` database user and grants permissions (privileges) to this user (required by SA).
- **postDBCreation.sql:** Creates the `spfile` from the `pfile` (parameter file).
- **init.ora:** Contains initialization parameters for the database. Certain parameter values are required by SA. See [Required and Suggested Parameters for init.ora](#) on page 203.
- **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 `tnsnames.ora` file.
- **bash\_profile:** Sets environment variables and sets shell limits for the `oracle` Unix user.
- **profile:** Sets environment variables for the `oracle` Unix user.
- **kernel\_params\_redhat.txt:** Contains kernel parameters for Red Hat Enterprise Linux 3 AS.
- **kernel\_params\_solaris.txt:** Contains kernel parameters for Solaris 10.
- **opware-oracle:** A script residing in `/etc/init.d` that starts up and shuts down the database and listener.

Note that the `/etc/init.d/opware-sas` script, which starts and stops the SA components, does not start and stop the database and listener. For more information on the `opware-sas` script, see the *SA Administration Guide*.

- **Export-Import:** A directory that contains parameter files and instructions for performing full database exports and imports.

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

Even if you plan to configure your Oracle installation manually and not run `truth.sh` to automatically configure the Oracle installation, the following scripts must be run and `init.ora` must have certain parameter values edited or added as shown in [Required and Suggested Parameters for init.ora](#) on page 203.

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

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

See [Tablespace Sizes](#) on page 40 in Chapter 2 for information about for additional tablespace sizing information.

- **CreateUserOpware\_Admin.sql:** Creates the `opware_admin` database user and grants permissions (privileges) to this user (required by SA). If you plan to create the `opware_admin` without running this script, see [Create the User Opware\\_Admin](#) on page 203.



- **init.ora:** Must be edited as shown in [Required and Suggested Parameters for init.ora](#) on page 203.

### Create the User Opware\_Admin

The following explains how to create this user:

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

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

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

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

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

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

## Required and Suggested Parameters for init.ora

(Both Oracle 10g and 11g) For SA, the following init.ora entries are either suggested or required:

```
log_buffer>=1048576
db_block_size>=8192
open_cursors >=1000
```

```

session_cached_cursors=50
nls_length_semantics=CHAR
nls_sort=GENERIC_M
processes >=1024
undo_management=AUTO (Suggested)
undo_tablespace=UNDO (Suggested)
query_rewrite_integrity=TRUSTED
query_rewrite_enabled=true
optimizer_mode=choose or all_rows
optimizer_index_cost_adj=20
optimizer_index_caching=80
cursor_sharing=SIMILAR (value can be set to SIMILAR(preferred) or EXACT,
recommended only if you encounter an Oracle error)
recyclebin=OFF
event="12099 trace name context forever, level 1"
_complex_view_merging=false

```

(Oracle 10g only) For SA, the following `init.ora` entries are either suggested or required:

```

sga_max_size >=1GB
db_cache_size>=629145600
shared_pool_size>=262144000
java_pool_size>=52428800
large_pool_size>=52428800
job_queue_processes >=10
sessions >=1152
pga_aggregate_target >=104857600
workarea_size_policy=auto
remote_login_passwordfile=SHARED

```

(Oracle 11g only) For SA, the following `init.ora` entries are either suggested or required:

```

memory_target >= 1.7GB
job_queue_processes >=1000 (default)
remote_login_passwordfile=EXCLUSIVE

```

## Creating the Database using the HP-Supplied Scripts

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

- 1 Obtain the database creation scripts from your HP support representative..
- 2 Configure the scripts. See [Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files](#) on page 201
- 3 Log in to the server as the Unix user `oracle`.
- 4 Copy the HP-supplied `init.ora` file to the following directory:  
`$ORACLE_BASE/admin/truth/create`
- 5 Examine the SQL scripts that you will run in **step 7**. If necessary, edit the scripts to conform to your organization's policies.
- 6 Log on to the server as the `oracle` user and change the mode of the HP-supplied `truth.sh` script:  
`chmod 755 truth.sh`
- 7 Launch the SQL scripts that create the database by running the `truth.sh` script:  
`./truth.sh`

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

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

## Post-Oracle Installation Tasks



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

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

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

```
$ORACLE_HOME/network/admin
```

Verify that the file conforms to the rules listed in [tnsnames.ora File Requirements](#) on page 206.

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

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

- 3 Create the following symbolic link:

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

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

- 5 For Red Hat Enterprise Linux 3 AS, create another symbolic link:

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

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

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

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

- 8 Copy the sample `listener.ora` file to `$ORACLE_HOME/network/admin`.

- 9 In `listener.ora`, change the value of the `host` parameter to the host name of server running the database.

### Location of Additional Oracle Data Files

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

```
/u01/oradata/truth
```

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

## tnsnames.ora File Requirements

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

- The file must reside in the following location:  
`/var/opt/oracle/tnsnames.ora`
- If the core is installed across multiple servers, a copy of the file must reside on the servers hosting the following components:
  - Model Repository
  - Infrastructure Component bundle (required by the Data Access Engine, Model Repository Multimaster Component, Software Repository Store)
  - Slice Component bundle (required by the Command Center, Web Services Data Access Engine, Global File System)
- For a core installed on multiple servers, the directory path of the `tnsnames.ora` file must be the same on each server.
- In a Single Core installation, the `tnsnames.ora` file must contain an entry for the Model Repository, as in the following example:

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

## tnsnames.ora: Multimaster Mesh Requirements

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

### Source Core

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

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

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

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

## Destination Core

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

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

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

## Requirements for Enabling Oracle Daylight Saving Time (DST)

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

- 1 Verify that your database is running on Oracle 10g or higher. If you are on an earlier database release, use one of the following MetaLink Notes to upgrade your database:  
10gR2 Database: MetaLink Note 362203.1
- 2 Use MetaLink Note 359145.1 to apply Oracle Database time zone fixes specific to your database version.
- 3 Use MetaLink Note 359145.1 to apply time zone fixes to the Oracle Java Virtual Machine (JVM) in the Oracle Database specific to your E-Business Suite database version.

## Installing the Model Repository Database on a Remote Server

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

- 1 Perform the following tasks on the server on which you will run the SA Installer:
  - a Ensure that the hostname `truth` resolves to the remote database server, not to the server on which you'll be running the SA Installer.
  - b Install the Oracle client. For a Multimaster install, you need the full Oracle client; for a Single Core (standalone) installation, the Oracle Instant client will suffice. The client software must be owned by the OS user `oracle`. Install the Oracle client in a location like

```
/opt/opsware/oracle_client_model_repo.
```

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

— Otherwise, download the appropriate Oracle full client from

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

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

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

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

- e Ensure that the SA Installer response file has the correct path to the client `tnsnames.ora` file (`%truth.tnsdir`), oracle client home (`%truth.orahome`), listener port (`%truth.port`), SA Installer machines subdomain (`%truth.dcSubDom`), etc. Based on the above steps your parameter values will be:
    - `%truth.tnsdir=/var/opt/oracle`
    - `%truth.orahome=/opt/opsware/oracle_client_model_repo`
    - `%truth.port=1521`
    - `%truth.dcSubDom=prod.example.com`
- 2 Set up the following on the Model Repository host (Oracle database server):
- a Log in as user `oracle`
  - b `cd $ORACLE_HOME/network/admin`
  - c Ensure that the `listener.ora` file has the following `SID_LIST_*` section:
 

```
SID_LIST_<your_listener_name> =
 (SID_LIST =
 (SID_DESC=
 (SID_NAME=truth)
 (ORACLE_HOME=<oracle_home>
)
)
```
  - d Ensure that the listener is started with the command
 

```
lsnrctl start <your_listener_name>
```

## Troubleshooting Remote Model Repository Installation

When you install or upgrade the Model Repository on a remote database server, Oracle gives the following error and the installer aborts:

```
Error: ORA-12526: TNS:listener: all appropriate instances are in restricted mode
```

### Problem

When SA installs or upgrades the schema in the Oracle database, it puts the database in a *restricted mode*. In Oracle 10g and 11g, the standard listener will reject connections if the database is in a restricted mode. In Oracle 10g and 11g, a database administrator can only access the restricted instance locally from the machine that the instance is running on.

### Solution

In Oracle 10g and 11g, if the listener has the `SID_LIST_*` paragraph in the `listener.ora` file, then the users with *restricted session* privilege are able to connect to a remote database, even if the database is in restricted mode. If the `listener.ora` file does not have the `SID_LIST_*` paragraph, then the listener rejects the client connections and gives an `ORA-12526: TNS: listener: all appropriate instances are in restricted mode error`.

## Example: A listener.ora Entry

---

```
OPSCORE1 =
 (DESCRIPTION_LIST =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = opscore1.mycompany.com)(PORT = 1521))
 (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
)
)

SID_LIST_OPSCORE1 =
 (SID_LIST =
 (SID_DESC=
 (SID_NAME=truth)
 (ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1)
)
 (SID_DESC =
 (SID_NAME = PLSExtProc)
 (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
 (PROGRAM = extproc)
)
)
)
```

---

In this example, the listener alias is OPSCORE1.

To start, stop, or check the status of the listener, enter the following commands:

```
su - oracle
```

To start the listener:

```
> lsnrctl start opscore1
```

To stop the listener:

```
> lsnrctl stop opscore1
```

To check the status of the listener:

```
> lsnrctl status opscore1
```

## Garbage Collection

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

### Data Retention Period

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



---

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

---

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

The parameters from AUDIT\_PARAMS table and their default values are:

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



As of SA 7.80, the DAY\_TRAN parameter that controlled retention time for transactions has been removed. To control transaction retention time, instead use the system configuration parameter `vault.garbageCollector.daysToPreserve`.

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

## Modifying the Retention Period Values

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

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



The values in the AUDIT\_PARAMS table must be exactly the same for all the cores in a mesh.

## Viewing GC DBA\_JOBS

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

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

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

| JOB | SCHEMA_USE | LAST_DATE | THIS_DATE | NEXT_DATE | B | WHAT                          |
|-----|------------|-----------|-----------|-----------|---|-------------------------------|
| 4   | GCADMIN    | 04-MAY-09 |           | 05-MAY-09 | N | WAYPURGE.GC_SESSIONS;         |
| 5   | GCADMIN    | 04-MAY-09 |           | 05-MAY-09 | N | CHANGELOGPURGE.GC_CHANGELOGS; |



```

6 GCADMIN 04-MAY-09 05-MAY-09 N AUDITPURGE.GC_AUDITLOGS;
7 GCADMIN 04-MAY-09 05-MAY-09 N STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;

```

where:

WAYPURGE.GC\_SESSIONS - Performs a sessions garbage collection

CHANGELOGPURGE.GC\_CHANGELOGS - Performs a changelogs garbage collection

AUDITPURGE.GC\_AUDITLOGS - Performs auditlogs garbage collection

STORAGEINITIATORPURGE.GC\_STORAGEINITIATORS - Performs storage data garbage collection

## Manually Running GC Jobs

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

```

Su - oracle
Sqlplus "/ as sysdba"

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

```

For example, this sample command runs the waypurge\_gc job:

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

## Monitor the ERROR\_INTERNAL\_MSG Table

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

```

Su - oracle
Sqlplus "/ as sysdba"

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

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

```

## Database Monitoring Strategy

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

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

```
$ su - oracle
```

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

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

### Verify that the Database Instances are Up and Responding

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

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

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

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

```
oracle 1883 1 0 Jul24 ? 00:00:00 ora_pmon_truth
oracle 1885 1 0 Jul24 ? 00:00:00 ora_psp0_truth
oracle 1887 1 0 Jul24 ? 00:00:00 ora_mman_truth
oracle 1891 1 0 Jul24 ? 00:00:45 ora_dbw0_truth
oracle 1895 1 0 Jul24 ? 00:01:11 ora_lgwr_truth
oracle 1897 1 0 Jul24 ? 00:00:02 ora_ckpt_truth
oracle 1899 1 0 Jul24 ? 00:00:24 ora_smon_truth
oracle 1901 1 0 Jul24 ? 00:00:00 ora_reco_truth
oracle 1903 1 0 Jul24 ? 00:00:02 ora_cjq0_truth
oracle 2391 1 0 Jul24 ? 00:00:00 ora_gmnc_truth
oracle 2513 1 0 Jul24 ? 00:00:00 ora_q000_truth
oracle 2515 1 0 Jul24 ? 00:00:00 ora_q001_truth
oracle 18837 1 0 03:04 ? 00:00:00 ora_mmon_truth
oracle 18839 1 0 03:04 ? 00:00:00 ora_mmln_truth
oracle 25184 24635 0 21:35 pts/1 00:00:00 grep ora_
```

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

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

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

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

### Verify that the Datafiles are Online

To verify that the datafiles are online, in `sqlplus`, enter the following commands:

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

The status should be AVAILABLE for all the data files.

## Verify That the Listener is Running

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

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

```
ps -ef | grep tns
```

```
oracle 1762 1 0 Jul24 ? 00:00:01 /u01/app/oracle/product/
10.2.0/db_1/bin/tnslsnr LISTENER -inherit
oracle 25231 25189 0 21:39 pts/1 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=perl.performance.qa.example.com)(PORT=1521))
. . . .
Instance "truth", status READY, has 1 handler(s) for this service...
```

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

```
tnsping truth
```

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

```
. . . .
Used parameter files:

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

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

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

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

## Examine the Log Files

To examine the log files, perform the following steps:

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

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

**Oracle 10g**

```
$ORACLE_BASE/admin/<SID>/bdump
```

**Oracle 11g**

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

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

**Oracle 10g**

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

**Oracle 11g**

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

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

**Oracle 10g**

```
$ORACLE_BASE/admin/<SID>/cdump
```

```
$ORACLE_BASE/admin/<SID>/adump
```

```
$ORACLE_BASE/admin/<SID>/udump
```

**Oracle 11g**

Various directories under:

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

## Check for Sufficient Free Disk Space in the Tablespaces

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

- 1 Enter the following commands in `sqlplus`:

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

select nvl(b.tablespace_name,
 nvl(a.tablespace_name, 'UNKOWN')) name,
 kbytes_alloc kbytes,
```

```

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

```

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

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

```

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

```

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

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

```

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

```

```

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

```

```

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

```

```

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

```

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

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

To verify that the Jobs in DBA\_JOBS ran successfully, perform the following steps:

- 1 Enter the following commands in SQL\*Plus:

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

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

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

| JOB | PRIV_USER     | LAST              | NEXT              | B | WHAT                                                                                                                                                                                                       |
|-----|---------------|-------------------|-------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21  | TRUTH         | 05/04/09 11:00:03 | 05/06/09 11:00:00 | N | DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH', estimate_percent=>dbms_stats.auto_sample_size, degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO', options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE); |
| 22  | AAA           | 05/04/09 11:00:03 | 05/06/09 11:00:00 | N | DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', estimate_percent=>dbms_stats.auto_sample_size, degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO', options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);   |
| 23  | LCREP         | 05/04/09 11:00:03 | 05/06/09 11:00:00 | N | gather_lcrep_stats;                                                                                                                                                                                        |
| 24  | GCADMIN       | 05/05/09 09:00:04 | 05/06/09 09:00:00 | N | WAYPURGE.GC_SESSIONS;                                                                                                                                                                                      |
| 25  | GCADMIN       | 05/05/09 09:00:04 | 05/06/09 09:00:00 | N | CHANGELOGPURGE.GC_CHANGELOGS;                                                                                                                                                                              |
| 26  | GCADMIN       | 05/05/09 09:00:04 | 05/06/09 09:00:00 | N | AUDITPURGE.GC_AUDITLOGS;                                                                                                                                                                                   |
| 27  | GCADMIN       | 05/05/09 05:25:08 | 05/05/09 06:25:08 | N | STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;                                                                                                                                                                |
| 28  | OPSWARE_ADMIN | 05/04/09 06:00:02 | 05/11/09 06:00:00 | N | DBMS_STATS.GATHER_SYSTEM_STATS(gathering_mode=>'INTERVAL', interval=>30);                                                                                                                                  |

8 rows selected.

where:

JOB - job id

SCHEMA\_USER - the user who with permissions to run the job

LAST\_DATE - last date-time when the job was run

NEXT\_DATE - next date the job will run

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

WHAT - the type of job

## Changes to the Database Statistics Job

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

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

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

Your output should be as follows:

### SA 7.50:

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

### SA 7.80:

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

## Backups of the `dba_jobs.what` information Table

During the SA 7.80 Model Repository Guide upgrade, the SA 7.50 `dba_jobs.what` information table is backed up and then replaced by the SA 7.8 `dba_jobs.what` table. You can view the backed up information by logging in to SQL\*Plus and entering the following commands:

```
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 a10
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,
SQL> ERR_DATE,
SQL> ERR_USER,
SQL> ERR_TABLE,
SQL> ERR_TEXT,
SQL> ERR_INFO
SQL> from ERROR_INTERNAL_MSG where ERR_TEXT = 'SA7.8 Model Repository Upgrade'
order by ERR_DATE;
```

Output will look similar to the following:

| ERR_ID | ERR_DATE  | ERR_USER | ERR_TABLE | ERR_TEXT                       | ERR_INFO                                                                                                     |
|--------|-----------|----------|-----------|--------------------------------|--------------------------------------------------------------------------------------------------------------|
| 6      | 07-MAY-09 | TRUTH    | DBA_JOBS  | SA7.8 Model Repository Upgrade | Pre SA7.8 dba_jobs.what value was: DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH', options=>'GATHER AUTO'); |
| 5      | 07-MAY-09 | AAA      | DBA_JOBS  | SA7.8 Model Repository Upgrade | Pre SA7.8 dba_jobs.what value was: DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', options=>'GATHER AUTO');   |

## Running the dba\_jobs manually

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

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

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

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

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

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

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

## Monitor Database Users

To monitor database users, perform the following steps:

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

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

## Troubleshooting System Diagnosis Errors

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

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



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

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

## Oracle Database Backup Methods

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

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

- **Export-Import:** An export extracts logical definitions and data from the database and writes the information to a file. Export-import does not support point-in-time recoveries. Do not use Export-Import as your only backup and recovery strategy.  
  
See the information on the `Export-Import` subdirectory in [Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files](#) on page 201.
- **Cold or Off-Line Backups:** This procedure shuts the database down and backs up all data, index, log, and control files. Cold or off-line backups do not support point-in-time recoveries.
- **Hot or Online Backups:** During these backups, the database must be available and in ARCHIVELOG mode. The tablespaces are set to backup mode. This procedure backs up tablespace files, control files, and archived redo log files. Hot or online backups support point-in-time recoveries.
- **RMAN Backups:** While the database is either off-line or on-line, use the `rman` utility to back up the database.

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

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

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

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

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

## Useful SQL

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

## Locked and Unlocked User

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

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

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

To unlock the user, enter the following sql command:

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

## GATHER\_SYSTEM\_STATS

Sometimes the GATHER\_SYSTEM\_STATS job will be suspended. To remove this from 'AUTOGATHERING' mode, perform the following steps:

- 1 Select PNAME, pval2 from SYS.AUX\_STATS\$ where pname = 'STATUS' ; .
- 2 If the PVAL2 status is "AUTOGATHERING", run GATHER\_SYSTEM\_STATS with gathering\_mode=( 'STOP' ); .
- 3 Run your job 'exec dbms\_job.run(xxx) ; .

## BIN\$ Objects

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

```
show parameter recyclebin;

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

By default, recyclebin is set to OFF.