

# Appendix A: Oracle Setup for the Model Repository

## IN THIS APPENDIX

This section discusses the following topics:

- Overview of Oracle Setup for Model Repository
- Supported Oracle Versions
- Hardware Requirements for the Oracle Database
- Required Operating System Packages and Patches
- The Opware Installer and the Oracle Database
- Tasks to Perform Before Installing the Oracle RDBMS Software
- Creating the Oracle Database
- Tasks to Perform After Creating the Oracle Database
- Database Monitoring for the Model Repository
- Oracle Database Backup Methods
- Upgrading the Database from Opware SAS 5.5 to 6.0

## Overview of Oracle Setup for Model Repository

The Model Repository (truth) is an Opware core component that stores information in an Oracle database. The process for installing Oracle and the Model Repository has three major steps:

- 1** Install the Oracle RDBMS software.
- 2** Create the Oracle database (instance).
- 3** Install the Model Repository.

You can perform both steps 1 and 2 either with or without Opware Installer. With the Opware Installer, steps 1 and 2 are performed as a single unit. You can perform step 3 only with the Opware Installer.

If you perform steps 1 and 2 with the Opware Installer, then you should read “The Opware Installer and the Oracle Database” on page 188

If you perform steps 1 and 2 without the Opware Installer, then you should read the following sections:

- “Tasks to Perform Before Installing the Oracle RDBMS Software” on page 191
- “Creating the Oracle Database” on page 192
- “Tasks to Perform After Creating the Oracle Database” on page 195

## Supported Oracle Versions

Support for the Model Repository is limited to specific versions of Oracle running on specific versions of operating systems. Table A-1 lists the supported Oracle versions.

*Table A-1: Supported Oracle Versions for Model Repository*

ORACLE EDITION	VERSIONS
Oracle Standard Edition	9.2.0.4 9.2.0.6 9.2.0.7 10.2.0.2
Oracle Enterprise Edition	9.2.0.4 9.2.0.6 9.2.0.7 10.2.0.2




---

Oracle version 9.2.0.5 is not supported with Opware SAS.

---

To be supported on the Model Repository, the Oracle versions listed in Table A-1 are limited to the operating systems listed in Table A-2.

Table A-2: Supported Operating Systems for Model Repository

SUPPORTED OPERATING SYSTEMS FOR MODEL REPOSITORY	VERSIONS	ARCHITECTURE
Sun Solaris	Solaris 8 Solaris 9	Sun SPARC Sun SPARC
Red Hat Linux	Red Hat Enterprise Linux 3 AS	32 bit x86

### Multiple Oracle Versions and Multimaster Cores

For the database export to succeed during the installation of a multimaster core, the version of the target (slave) database cannot be 9.x if the version of the source (master) database is 10.x. Table A-3 lists this allowed version combinations.

Table A-3: Database Versions Allowed for Multimaster

SOURCE DB VERSION	TARGET DB VERSION	ALLOWED?
9	9	Y
9	10	Y
10	9	N
10	10	Y

### Hardware Requirements for the Oracle Database

The server that will run the Oracle database for the Model Repository has the following hardware requirements.

### Physical Memory and Swap Space

Oracle requires at least 1024 MB of physical RAM. The amount of swap space required depends on the size of the physical RAM, as shown in Table A-4.

Table A-4: RAM and Swap Space

SIZE OF RAM (MB)	SWAP SPACE REQUIRED (MB)
1024 - 2048	1.5 times the size of RAM
2094 - 8192	equal to size of RAM
more than 8192	9

### Temporary Disk Space

The Oracle Universal Installer (OUI) requires up to 400 MB free space in the `/tmp` directory.

### Permanent Disk Space

The amount of disk space required depends on the Oracle edition and the number of servers managed by Opware SAS, as listed in Table A-5.

Table A-5: Database Versions Allowed for Multimaster

ORACLE EDITION	DISK SPACE REQUIRED BY ORACLE RDBMS SOFTWARE (GB)	ADDITIONAL DISK SPACE (FOR DATA AND INDEX TABLESPACES) REQUIRED FOR EVERY 1000 SERVERS MANAGED BY SAS (GB)
Enterprise	2.0	3.1
Standard	1.5	3.1

For the disk space requirements of an upgrade, see "Upgrading the Database from Opware SAS 5.5 to 6.0" on page 205.

## Required Operating System Packages and Patches

The sections that follow list the packages and patches required by the Oracle database. The Opsware Installer checks for these packages and patches before installing the Oracle database. If you create the database without the Opsware Installer, you must check for these packages and patches manually.

### Required Packages for RedHat Enterprise Linux 3 AS

RedHat Enterprise Linux 3 AS must have the following packages:

```
make-3.79.1
gcc-3.2.3-34
glibc-2.3.2-95.20
compat-db-4.0.14-5
compat-gcc-7.3-2.96.128
compat-gcc-c++-7.3-2.96.128
compat-libstdc++-7.3-2.96.128
compat-libstdc++-devel-7.3-2.96.128
openmotif21-2.1.30-8
setarch-1.3-1
libaio-0.3.96-5
```

### Required Packages for Solaris 8, 9, and 10

Solaris 8, 9 and 10 must have the following packages:

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

### Required Patches for Solaris 8

Solaris 8 must have the following patches (or later):

```
108528-23: SunOS 5.8: kernel update patch
108652-66: X11 6.4.1: Xsun patch
108773-18: SunOS 5.8: IIIM and X I/O Method patch
108921-16: CDE 1.4: dtwm patch
```

108940-53: Motif 1.2.7 and 2.1.1: Runtime lib. patch for Solaris 8  
108987-13: SunOS 5.8: Patch for patchadd and patchrm  
108989-02: /usr/kernel/sys/acctctl & /.../exacsys patch  
108993-18: SunOS 5.8: LDAP2 client, libc, libthread ... lib. patch  
109147-24: SunOS 5.8: linker patch  
110386-03: SunOS 5.8: RBAC Feature Patch  
111023-02: SunOS 5.8: /kernel/fs/mntfs and ... sparcv9/mntfs  
111111-03: SunOS 5.8: /usr/bin/nawk patch  
111308-03: SunOS 5.8: /usr/lib/libmtmalloc.so.1 patch  
111310-01: SunOS 5.8: /usr/lib/libdhcpageant.so.1 patch  
112396-02: SunOS 5.8: /usr/bin/fgrep patch  
111721-04: SunOS 5.8: Math Library (libm) patch  
112003-03: SunOS 5.8: Unable to load fontset in 64-bit Solaris 8 iso-1 or iso-15

### **Required Patches for Solaris 9**

Solaris 9 must have the following patches (or later):

112233-11: SunOS 5.9: Kernel Patch  
111722-04: SunOS 5.9: Math Library (libm) patch

## **The Opware Installer and the Oracle Database**

To administer a database created by the Opware Installer, you should know about the settings discussed in this section.

### **Database Settings and Files Changed by the Opware Installer**

When it installs the Oracle RDBMS software and creates the database, the Opware Installer makes the following changes:

- 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/10.2.0/db_1`

- Sets the `$ORACLE_SID` environment variable to `truth`.
- Gets the service name (TNS name) from the Opware Installer interview (`truth.servicename` prompt) and inserts it into the `tnsnames.ora` file.

- In the `/var/opt/oracle/tnsnames.ora` file, changes the value of the `host` parameter to the value returned by the Unix `hostname` command.

- Creates the data and index files under the following directories:

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

The system administrator can configure the `/u01`, `/u02`, `/u03` directories before installing the Oracle RDBMS software.

- 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`.) 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.

- For Solaris 8 and 9, modifies `/etc/system` and asks the user to reboot the server.

### Allowed Database Changes

When it creates the Oracle database, the Opsware Installer sets the values for parameters in various files. This section lists the parameters set by the Opsware Installer that can be changed without adversely affecting Opsware SAS.

### Allowed Kernel Parameter Changes for RedHat Enterprise Linux 3 AS

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

```
kernel.shmmax=2147483648
kernel.shmall=2097152
kernel.shmmni=4096
kernel.sem=256 32000 256 256
net.core.rmem_default=262144
net.core.wmem_default=262144
net.core.rmem_max=262144
net.core.wmem_max=262144
fs.file-max=65536
net.ipv4.ip_local_port_range=1024 65000
```

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

```
oracle soft nofile 4096
oracle hard nofile 63536
oracle soft nproc 2047
oracle hard nproc 16384
session required /lib/security/pam_limits.so
```

### **Allowed Kernel Parameter Changes for Solaris 8 and 9**

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

```
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmns=2058
set semsys:seminfo_semmsl=256
set semsys:seminfo_semmni=100
set semsys:seminfo_semvmx=32767
set noexec_user_stack=1
```

You can increase the value for the following parameter in `/etc/system`:

```
set shmsys:shminfo_shmmax=2147483648
```

You can remove the following parameters in `/etc/system`:

```
forceload: sys/shmsys
forceload: sys/semsys
forceload: sys/msgsys
```

### **Allowed Changes to `init.ora`**

You can increase values for the following parameters in `init.ora`:

```
db_cache_size=629145600
shared_pool_size=262144000
java_pool_size=52428800
large_pool_size=52428800
log_buffer=1048576
```

### **Location of Additional Oracle Data Files**

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

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



## Tasks to Perform Before Installing the Oracle RDBMS Software

If you create the database with the Opsware Installer, you do not need to perform the tasks in this section.

This section summarizes some of the pre-requisites for the Oracle RDBMS software. For details on these pre-requisites, see the *Oracle Installation Guide*. 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:

- 1** Verify that the server has the software listed in “Required Operating System Packages and Patches” on page 187.

- 2** Download and unzip the sample files.

The sample files are available in the support area of the Opsware, Inc. web site at [www.opsware.com](http://www.opsware.com). See “Overview of the Sample Scripts and Configuration Files” on page 192.

- 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 /bin/bash oracle
chown oracle:oinstall /u01/app/oracle
```

- 5** Set the environment variables for the `oracle` user.

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

```
bash_profile
profile
```

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

## Creating the Oracle Database

If you create the database with the Opware Installer, you do not need to perform the tasks in this section.

### Overview of the Sample Scripts and Configuration Files

Opware, Inc. provides a bundle of sample files for you to copy and edit. Referenced throughout the instructions in this document, the sample files include SQL scripts, database configuration files, and kernel parameter settings.

The sample files are available in the support area of the Opware, Inc. web site at [www.opware.com](http://www.opware.com).

The following list summarizes the sample scripts and configuration files:

- `truth.sh`: A shell script that creates directories and then launches the `truth.sql` script.
- `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 Opware SAS), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.
- `CreateDBfiles.sql`: Creates the following tablespaces, which are required by Opware SAS:

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

- `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 Opware SAS).
- `postDBCcreation.sql`: Creates the `spfile` from the `pfile` (parameter file).
- `init.ora`: Contains initialization parameters for the database. See “Required and Suggested Parameters for `init.ora`” on page 193.
- `tnsnames.ora`: Enables resolution of database names used internally by Opware SAS.
- `listener.ora`: Contains configuration parameters for the listener. Opware SAS requires the listener to listen on port 1521.
- `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 RedHat Enterprise Linux 3 AS.
- `kernel_params_solaris.txt`: Contains kernel parameters for for Solaris 8 and 9.
- `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 SAS components, does not start and stop the database and listener. For more information on the `opware-sas` script, see the *Opware® SAS Administration Guide*.

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

### Required and Suggested Parameters for `init.ora`

For Opware SAS, 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
log_buffer>=1048576
db_block_size>=8192
open_cursors >=300
job_queue_processes >=10
nls_length_semantics=CHAR
nls_sort=GENERIC_M
processes >=1024
sessions >=1152
pga_aggregate_target >=104857600
workarea_size_policy=auto
undo_management=AUTO (Suggested)
undo_tablespace=UNDO (Suggested)
query_rewrite_integrity=TRUSTED
query_rewrite_enabled=true
optimizer_mode=CHOOSE
optimizer_index_cost_adj=20
optimizer_index_caching=80
cursor_sharing=similar
recyclebin=OFF (Suggested, for Oracle 10g only)
```

A bug in Oracle10g regarding DML containing inline views and certain types of subqueries causes Oracle to throw an ORA-00600 exception. Until the bug is fixed in Oracle 10g, the workaround is the following entry in `intit.ora`:

```
_complex_view_merging = false
```

### **File Locations in the Sample Scripts**

In the sample scripts and configuration files, `ORACLE_HOME` environment variable is set to the following value:

```
/u01/app/oracle/product/10.2.0/db_1
```

The sample `init.ora` file has the following settings for files:

```
db_create_file_dest=/u01/oradata/truth
db_create_online_log_dest_1=/u02/oradata/truth
db_create_online_log_dest_2=/u03/oradata/truth
```

```
control_files=(/u02/oradata/truth/control01.ctl,/u03/
oradata/truth/control02.ctl)
```

If your organization has policies that do not match these settings, then you should modify the sample files accordingly.

## Creating the Database With the Sample Scripts

To create the database with the sample scripts, perform the following steps:

- 1 Download and unzip the sample files.

The sample files are available in the support area of the Opware, Inc. web site at [www.opware.com](http://www.opware.com). See "Overview of the Sample Scripts and Configuration Files" on page 192.

- 2 Log on to the server as the Unix user `oracle`.

- 3 Copy the sample `init.ora` file to the following directory:

```
$ORACLE_BASE/admin/truth/create
```

- 4 Examine the sample SQL scripts that you will run in step 6. If necessary, edit the scripts to conform to your organization's policies.

- 5 Log on to the server as the `oracle` user and change the mode of the sample `truth.sh` script:

```
chmod 755 truth.sh
```

- 6 To launch the sample SQL scripts that create the database, run the `truth.sh` script:

```
./truth.sh
```

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

```
$ORACLE_HOME/assistants/dbca/logs
```

## Tasks to Perform After Creating the Oracle Database

If you create the database with the Opware Installer, you do not need to perform the tasks in this section, except for step 1.

After creating the database, but before installing the Model Repository with the Opware 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 "Requirements for the `tnsnames.ora` File" on page 196.

- 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 RedHat 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/rc5.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.

### Requirements for the `tnsnames.ora` File

The `tnsnames.ora` file enables resolution of database names used internally by the core components. Opware SAS 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 running the following components:
  - Model Repository
  - Data Access Engine
  - Web Services Data Access Engine
  - Opware Command Center

- Global File System
- Model Repository Multimaster Component
- For a core installed on multiple servers, the directory path of the `tnsnames.ora` file must be the same on each server.
- In a standalone core, the `tnsnames.ora` file must contain an entry for the Model Repository, as in the following example:

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

### **Multimaster Mesh Requirements for `tnsnames.ora`**

In a multimaster mesh, the `tnsnames.ora` file of the central (source, master) core must contain an entry for its own Model Repository. The file must also have entries for the Model Repositories of the other cores in the mesh. For the entries of the other (non-central) cores, the host specifies the central core Gateway, and the port number is derived from this formula:  $(20000) + (\text{facility ID of the non-central core})$ .

The following `tnsnames.ora` example is for the central core of a multimaster mesh. In this example, the TNS service name of the central core is `orange_truth`, which runs on the host `orange.opsware.com`. The TNS name of the non-central core is `cyan_truth`, which has a facility ID of 556. Note that the entry for `cyan_truth` specifies `orange.opsware.com`, the host running the central core's Gateway.

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

In a multimaster mesh, the `tnsnames.ora` file of a non-central (non-master) core must contain an entry for its own Model Repository, but does not require entries for other cores in the mesh. In the following `tnsnames.ora` example, the TNS service name of the non-central core is `cyan_truth`, and the core runs on the host, `cyan.opsware.com`.

```
cyan_truth =
  (DESCRIPTION=
```

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

## Database Monitoring for the Model Repository

Because the Model Repository is a critical component of Opware SAS, 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"
```

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

- 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_qmnc_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;
```

### Verifying 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.

### Verifying That the Listener is Running

- 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 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.opsware.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.opsware.com)) (ADDRESS=(PROTOCOL=TC
P) (HOST=192.168.165.178) (PORT=1521)))
OK (0 msec)
```

```
Attempting to contact
(DESCRIPTION=(ADDRESS=(HOST=localhost) (PORT=1521) (PROTOCOL=t
cp)) (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
```

## Examining the Log Files

- 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/admin/<SID>/bdump
```

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

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

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

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

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

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

## Checking for Sufficient Free Disk Space in the Tablespaces

- 1 Enter the following commands in `sqlplus`:

```
column pct_used format 999.9          heading "%|Used"
column name      format a16           heading "Tablespace Name"
column Kbytes    format 999,999,999   heading "KBytes"
column used      format 999,999,999   heading "Used"
```

```

column free      format 999,999,999 heading "Free"
column largest   format 999,999,999 heading "Largest"
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
from ( select sum(bytes)/1024 Kbytes_free,
            max(bytes)/1024 largest,
            tablespace_name
      from sys.dba_free_space
      group by tablespace_name ) a,
     ( select sum(bytes)/1024 Kbytes_alloc,
            tablespace_name
      from sys.dba_data_files
      group by tablespace_name )b
where a.tablespace_name (+) = b.tablespace_name
order by 1
/
set line 250
col file_name format a75
col tablespace_name format a15
select tablespace_name, maxbytes, AUTOEXTENSIBLE,
       file_name from dba_data_files;

```

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 temp 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 ;
```

### Verifying That the Jobs in DBA\_JOBS Ran Successfully

When it installs the Model Repository, the Opware Installer sets up these jobs, which perform statistics and garbage collection. If these jobs do not run successfully, database performance will degrade.

- 1** To see if the jobs have run successfully, enter the following commands in `sqlplus`:

```
Col schema_user format a10  
Col what format a50  
Set line 200  
Select job, schema_user, last_date, this_date, next_date,  
broken, what from da_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*`, the job performs statistics collection. If the value is `*PURGE.GC*`, the job performs garbage collection.

- 2** If you need to run the statistics and collection jobs manually, start by entering the following command in `sqlplus`:  

```
grant create session to truth, aaa, lcrep;
```
- 3** To run the statistics collection jobs manually in `sqlplus`, enter `exec` commands similar to the example shown in this step.

If you copy and paste the following `exec` command examples, substitute the variables such as `schema_user_1` with the values of the `schema_user` column displayed by the preceding `select` statement. Substitute the variables such as `job_no_1` with the values of the `job` column displayed by the same `select` statement.

```
connect <schema_user_1>/<password>
exec dbms_job.run(<job_no_1>)
```

```
connect < schema _user_2>/<password>
exec dbms_job.run(<job_no_2>);
```

```
connect < schema _user_3>/<password>
exec dbms_job.run(<job_no_3>)
```

```
connect < schema _user_4>/<password>
exec dbms_job.run(<job_no_4>);
```

- 4** To run the garbage collection jobs manually, enter the following commands in `sqlplus`, substituting the job ID variables such as `job_no_1`:

```
grant create session to gadmin;
connect gadmin/<password_of_gadmin>
```

```
exec dbms_job.run(<job_no_1>);
exec dbms_job.run(<job_no_2>);
exec dbms_job.run(<job_no_3>);
exec dbms_job.run(<job_no_4>);
```

- 5** If you entered the `grant` command in step 2, enter the following command in `sqlplus`:

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

### Monitoring the `ERROR_INTERNAL_MSG` Table

The garbage collection jobs write exceptions to the `truth.ERROR_INTERNAL_MSG` table. Monitor this table daily for errors.

### Monitoring Database Users

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

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

The preceding `select` command should display the following users:

```
OPSWARE_PUBLIC_VIEWS
TRUTH
```

```
AAA_USER  
LCREP  
GCADMIN  
TWIST  
SPIN  
AAA  
OPSWARE_ADMIN  
VAULT
```

(The `VAULT` user is for multimaster databases only.)

The `default_tablespace` of the Opware SAS users should not be `SYSTEM` or `SYSAUX`. The `temporary_tablespace` of all users should be `TEMP`.

- 2** If a database user listed in the preceding step has the `account_status` of `LOCKED`, then unlock the user by entering the following command in `sqlplus`:
- ```
ALTER USER <username> ACCOUNT UNLOCK;
```

## 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 “Overview of the Sample Scripts and Configuration Files” on page 192.
- **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 On-Line 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 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>

## Upgrading the Database from Opware SAS 5.5 to 6.0

If you are upgrading an existing installation of Opware SAS 5.5 to 6.0, then you must make the following changes to the Oracle database that implements the Model Repository:

- 1** Before upgrading the database, make sure that the tablespaces have enough disk space.

Allocate 10% additional disk space for the current data and index tablespaces. Most of the growth will be in the TRUTH\_DATA and TRUTH\_INDX tablespaces. See "Checking for Sufficient Free Disk Space in the Tablespaces" on page 200.

- 2** Create the following new tablespaces:

```
AUDIT_DATA
AUDIT_INDX
```

You can create these tablespaces by issuing SQL statements in the `sqlplus` utility. To run `sqlplus`, log on to the server as `oracle` and enter the following command:

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

Here are some example SQL statements for creating the new tablespaces:

```
CREATE TABLESPACE "AUDIT_DATA" LOGGING DATAFILE
'/u03/oradata/truth/audit_data01.dbf' SIZE 32M
AUTOEXTEND ON NEXT 128M MAXSIZE 4000M
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO ;
```

```
CREATE TABLESPACE "AUDIT_INDX" LOGGING DATAFILE
'/u02/oradata/truth/audit_indx01.dbf' SIZE 32M
```

```
AUTOEXTEND ON NEXT 128M MAXSIZE 4000M  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO ;
```

- 3** In `sqlplus`, grant the following privileges (permissions) to the `opsware_admin` database user:

```
grant analyze any to opsware_admin;  
grant insert, update, delete, select on sys.aux_stats$  
to opsware_admin;  
grant gather_system_statistics to opsware_admin;
```

For Oracle version 10 (not 9), issue the following command:

```
grant create job to opsware_admin;
```