Maintaining Oracle Standards

As a DBA you will be expected to draft and control different standards, as they pertain to an Oracle database. Some of the best standards are the ones that have a wide adoption rate, and which are easily understood and well-documented. Standards are intended to grease the various gears of a team so that they work together with less friction. Standards will also make a DBA's life easier in the long run, safer for all those involved, and more efficient, because time isn't spent reinventing or rewriting a process that wasn't based on a standard in the first place.

This chapter outlines several standards for the major tasks that most DBAs are expected to perform: installation, configuration, and the maintenance of Oracle software. Expect the standards that you adopt to change gradually over time as technology improves and to reflect changes within your organization.

Adapting to constant change

You can have more than one ORACLE_HOME for every node, primarily for migration projects that will be spread over an extended period of time. There is an inherent risk in running multiple ORACLE_HOMES, in that you may mistakenly use the wrong ones. It is my personal recommendation that the ongoing risk is worth it when using the Step-Ordered Approach to Migrating, because it reduces the overall downtime (see Chapter 7).

An ORACLE_HOME consists of installed binary files along with Oracle-supplied scripts. The database comprises memory structures, background processes, control files, parameter files, data files, and temporary and undo files. There are other types of ORACLE_HOME(s) that contain ORACLE installed software, as each of them is only an environmental variable pointing to a specific disk location. Other types will have other names like: CRS_HOME or AGENT_HOME that may be identified as an ORACLE_HOME in the Oracle documentation. This means you can have multiple ORACLE_HOMES on the same node running different versions of a database and/or different versions of other Oracle software.
Database concepts

You can have multiple instances of a single database. An instance consists of multiple background OS processes that are executed by the Oracle binary located in $ORACLE_HOME/bin/oracle. Multiple instances are known as Real Application Clusters (RAC) that can be deployed on a single server or distributed on multiple servers, but are associated with a single shared set of data files.

Refer to the installation guide for your operating system as well as the recommendations found on My Oracle Support. MOS also contains a Certifications Area, which you can use to determine if the operating system is compatible with the version you are installing, as well as to determine compatibility with other Oracle software that will be used with an Oracle database.

Database startup initialization Parameter files (pfile), as well as the new Server Parameter file (spfile), control each database instance, providing adjustable instructions. This flexibility allows for different instance parameters that are based on an application’s need for distributing the available resources across multiple servers (also known as nodes).

A database control file connects all of the physical components found on a storage device with the correct database, tracking and synchronizing all changes. Since a database has its own controls, called a Relational Database Management System (RDBMS), making changes to the physical files at the operating system level can't be done when the database is open without incurring some sort of damage or data loss. Changes such as moving data files, renaming, or copying the physical components at an OS level may corrupt or make them otherwise unrecoverable.

Multiple ORACLE_HOME(s)

In this book, an ORACLE_HOME is the home that is currently in use. This may or may not be the production home. There is usually a different ORACLE_HOME that you are migrating to at any one point in time. This migration can happen on a production, test, or development server depending on which one you are working on at the time.

Oracle releases Patch Set Updates (PSU) and Critical Patch Updates (CPU) quarterly. CPU is the quarterly security release. Oracle recommends that clients use PSUs since a PSU is a superset of a CPU. There is continued ongoing debate on whether Oracle’s recommendation should be followed because there are problems if you need to revert to applying CPUs. This is where it would be best to ask the Oracle community for feedback on this important decision.
When it is time to upgrade the Oracle database software, it is recommended not to upgrade the home that is currently in use, because that leaves you with a longer downtime should the upgrade process fail. It would be much faster as part of the downgrade process to shutdown the database and bring it back up in the previous ORACLE_HOME. This means you are only reversing any database changes rather than trying to reverse both software and database changes, which increases the length of downtime.

**Keeping the environment clean**

Multiple ORACLE_HOME(s) accessing different memory structures and background processes can cause the environment to become cloudy. Confusion as to which executables or scripts should be run can cause outages, core dumps, and human errors. There are some procedures and practices to adopt in order to keep the switching process as clean as possible.

How do you keep multiple installations independent of each other while reducing contention at all levels? Multiple operating system accounts are the cleanest way if implemented along with the **Optimal Flexible Architecture (OFA)** standards. Check out the installation guide as part of the documentation for your OS, which has quite a bit of information on how to run Oracle products with multiple accounts.

Here are a few notes about variations on the multi-user system of installing Oracle software on a single server:

- **One OS user account per database**, which would result in an increased complexity when shell scripting tasks run against a particular database. The script would need to verify the ORACLE_HOME and the correct database combination, as there is the possibility of several combinations. Each database would require their own ORACLE_HOME instead of sharing the same home between multiple databases. Setting environmental variables can be done with local customizations instead of using the Oracle-supplied oraenv file.

- **One OS user per Oracle release**, which would result in less ORACLE_HOME installs than one OS user account/database type. Running OS shell scripts against this type of database would more than likely be a combination of oraenv and custom environmental variables. It is easy to modify oraenv to source a custom variable, as shown later in this chapter.

While that may be the cleanest, most secure way to implement Oracle software installs, it is not necessarily the easiest method. There is more administrative overhead involved with configuring the operating system user and group accounts as well as the appropriate file permissions so that each is independent on a single server.
Oracle's Optimal Flexible Architecture (OFA)

In the following section, you will find a small summary of the OFA standard(s) that were written by Cary Millsap. His article titled Oracle for Open Systems was first published in 1995. It is still used today and is widely adopted on Unix systems by DBAs, no matter the database size. This standard that has been expanded and revised to embrace the newer hardware technologies, and the Automatic Diagnostic Repository found in 11g is in the installation guide of every operating system.


1. Name the Unix mount points with this pattern /mountpoint+numbered string and start numbering with a left-padded zero to keep the list in numerical order. For example, /u01, /u02, /u03.

2. The Oracle operating system account that owns ORACLE_HOME with a home directory of /mountpoint/directory/user. For example, /u01/app/oracle.

3. Remove all hardcoded references in shell scripts to exact path names except for the few key Unix files that require such an entry. Use environmental variables instead.

4. Each ORACLE_HOME is recommended to be installed with a pattern matching /oracle_user_home_directory/product/version. For example, /u01/app/oracle/product/11.2.0/dbhome_1.

5. In 11g, the Diagnosability Framework changes the older default location for cdump, udump, and bdump database parameters (see the next section 11g differences in the OFA standard). It would be a recommended location for each of the following directories that are not part of the 11g Diagnosability Framework—adump, create, logbook, pfile, and scripts. There are other files related to archivelogs, data pump, or export files that are now Oracle-recommended to be put in the FLASH_RECOVERY_AREA (FRA). For example, /u01/app/oracle/admin/newdb.

6. Use the Unix profile and the Oracle-supplied files that set the environment—oraenv, coraenv, and dbhome. See later in this chapter for specific recommendations for these files, ORACLE_SID or just SID (System Identifier).
7. Identify an instance with the combination of $ORACLE_HOME, $ORACLE_SID and $HOSTNAME, which makes it unique and usually defaults to the database name.

8. Name data files with /mount point/specific_to_data/ORACLE_SID/control.ctl (control files), redo+number.log (redo logs, change .log to .rdo, see Major Blunder list) and tablespace_name+number.dbf (data files). A personal recommendation includes adding the ORACLE_SID to a data filename. For example, /u01/oradata/newdb/contrl01_newdb.ctl.

9. Database objects that are backed up at the same time or that have a similar purpose or lifespan should be grouped in the same tablespace. For example, temporary segments should be in a temporary tablespace, read only objects should be in their own tablespace, system objects in the system tablespace, and so forth.

10. Limit tablespace names to eight characters with a matching data filename to simplify administrative tasks. For example, USERS tablespace with the data file: /u01/oradata/newdb/users01_newdb.dbf.

Steps 8, 9, and 10 will not be covered here (a lot has changed in data storage), as the original document refers to the obsolete Oracle Parallel Server and one data file per disk implementation.

11g differences in the OFA standard

The 10g release of the Oracle Database introduced the use of the new $ORACLE_BASE directory as the primary starting point for installations. ORACLE_BASE will be required in future installations, but it currently brings up a warning box during the GUI install if it is not set in 11g. It is highly recommended to go ahead and set ORACLE_BASE before installation in order to control the location of the new Diagnosability Framework components.

These details can be found in the Oracle Database Pre-installation Requirements section of the installation guide for your operating system. Oracle also recommends locating the Flash Recovery Area (FRA) and database files under $ORACLE_BASE. The FRA is meant for the storage of any type of backup, recovery, or flashback technologies when used in conjunction with an Oracle database. This type of implementation puts the majority of the files on a single originating mount point, such as /u01.

If you implement Oracle's Automatic Storage Management (ASM), then it would make sense to install all of the components in a single disk location or any other storage method for using multiple disks as a single unit of space. ASM is a logical volume manager on top of the physical disks; it presents those logical volumes as disk groups.
ASM is a volume management tool in that it simplifies data storage for a growing system. It gives you the ability to add additional storage, as the database needs it, without interruptions. While it provides additional flexibility in adding storage, most often it requires additional resources (time, personnel expertise) to install, configure, and maintain. There are also ASM-related changes to account for in most of the database maintenance tasks, backup and recovery routines, and Data Guard, among others.

**XWINDOWS and GUI displays**

Nowadays Unix hardware is more often headless in racks, and because of the missing components, there is no need to sit at the server anymore to run any type of GUI display such as the OUI, NETCA, or DBCA utilities. Simply run an XWindows display on your desktop or another server that has a GUI display. You must export the $DISPLAY variable to your desktop IP address to start Oracle's Universal Installer (OUI) or any other GUI Oracle utility. See your system administrator for assistance with XWindows issues.

Most Unix desktops such as the widely adopted Ubuntu will have XWindows built in, or packages are available for installation. XWindows software is needed for any GUI sessions if your desktop is Windows-based, because it will not have the X support built in. Other open source versions for Windows that are available include Cygwin/X, Xming, and WeirdX, and products that will require the purchase of a license include Xmanager, Exceed, MKS X/Server, Reflection X, and X-Win32.

System administrators will most often require the use of a secure shell (ssh) encrypted command-line utility to connect remotely to a server. PUTTY is a freeware telnet/ssh client and does this job very well; it can be downloaded from [http://www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/).

**Automating day-to-day tasks**

Options to automate the DBA tasks mentioned in the previous chapter include some of the following:

- **DBMS_SCHEDULER**: Oracle-supplied PL/SQL package that comes preinstalled
- **OS scheduling commands**: Certain tasks such as monitoring file space usage and removing old trace/log files are often scheduled using something like Unix `cron` or the Windows `at` command. The best options include tasks that need to run no matter the database state—down, up, or mounted.
Chapter 2

- OEM Grid Control Intelligent Agent: Requires installation and configuring to run OS type commands. A very useful Enterprise-wide monitoring tool but quite an investment in time to configure and maintain for multiple servers.

Let's look at an example task and apply it to each of the three types of automation mentioned above: Monitoring the Database Alert Log.

**DBMS_SCHEDULER**

*DBMS_SCHEDULER*, along with the power of PL/SQL, provides a mechanism for automating some of the daily tasks mentioned earlier in this chapter. Using a scheduler from within the database instead of an external scripting has some advantages as well as disadvantages.

Advantages of using DBMS_SCHEDULER:

- It is another tool to make your environment the same across the enterprise.
- Runs on any operating system that the database can.
- Capable to run a program, anonymous PL/SQL blocks, stored procedure, executables or even a chain of commands.
- It will execute operating system commands along with shell scripts, which can run external Oracle utilities, such as RMAN, export, adrci, or datapump.
- All of your tasks and PL/SQL (called metadata) associated with these scheduled jobs are backed up along with the database.
- Jobs run only when the database is up. There are many times when you have to disable certain jobs when the database is down if scheduled at the operating system level.
- It is relatively easy to grant or revoke access for each specific job to other users.
- It removes the need to store a password for job execution, as you have to do with scripting at the operating system level.

Disadvantages of DBMS_SCHEDULER:

- The fact that jobs only run when the database is open may be considered a disadvantage as well as an advantage of DBMS_SCHEDULER.
- It can be harder to manage due to the PL/SQL nature of the interface.
- Third-party interfaces to other applications, schedulers, assorted operating systems, or even databases can be challenging and hard to troubleshoot when things aren't working as expected.
- The OEM console is one GUI method of monitoring DBMS_SCHEDULER jobs, which takes quite a few clicks to determine status, failure, past runs, among others.
The following DBMS_SCHEDULER task will give you a critical piece of information for certain recovery situations (see Chapter 6, Extended RMAN for more information). See the downloaded code for this chapter for more examples of DBMS_SCHEDULER commands for certain DBA tasks.

Note how this job will be repeated on a daily basis. The following is an example of a recommended daily task, writing the database ID to the alert log:

```sql
BEGIN
sys.dbms_scheduler.create_job (
  job_name => "SYS"."DBID_TOALERT",
  job_type => 'PLSQL_BLOCK',
  job_action => 'declare l_dbid number;
                begin
                select dbid into l_dbid from v$database;
                dbms_system.ksdwrt (2, ''DBID='' || l_dbid);
                end;
  repeat_interval => 'FREQ=DAILY;BYHOUR=14',
  start_date => to_timestamp_tz('2007-12-20 America/Denver', 'YYYY-MM-DD TZR'),
  job_class => "DEFAULT_JOB_CLASS",
  comments => 'Write DBID to Alert Log for Recovery',
  auto_drop => FALSE,
  enabled => FALSE);
sys.dbms_scheduler.set_attribute( name => "SYS"."DBID_TOALERT",
  attribute => 'job_weight', value => 1);
sys.dbms_scheduler.set_attribute( name => "SYS"."DBID_TOALERT",
  attribute => 'restartable', value => TRUE);
sys.dbms_scheduler.enable( "SYS"."DBID_TOALERT" );
END;
/
```

**OS cron utility executing a scheduled task on a Unix server**

See the code provided for this chapter `dbid_toalert_os.ksh` Unix file for a complete script. The script includes the appropriate shell script commands to be able to run SQL code. The script also contains example code that contains suggestions on writing Unix shell scripting for DBAs found later in this chapter.
OEM Console plus the Intelligent Agent

To schedule the same job of writing the database ID to the alert log, see the following screenshot for details. This particular job is scheduled to run once daily with no expiration date.

If this job was resource intensive, it would be wise to use the following method: How to Incorporate Pre-Defined jobs into your Resource Manager Plan [ID 971991.1]. Resource Manager Plans would allow the DBA to control database jobs that need to run with certain priorities for allocating resources appropriately.

This basically takes the exact information as using the DBMS_SCHEDULER in the preceding section for utilizing a GUI screen to input the job information.
11g Diagnosability Framework

The Oracle 11g Database includes a full suite of diagnostic tools and advisors that are at least partially integrated with My Oracle Support (MOS), automatically uploading metrics via the Oracle Configuration Manager. This new diagnosability infrastructure includes the monitoring of the RDBMS, Automatic Storage Management (ASM), Oracle Call Interface (OCI), SQL*Net, and Oracle Application Server 11.1 products.

This new version of Oracle introduces the Automatic Diagnostic Repository (ADR), which is a flat file structure containing all alert logs, core files, trace files, and incident information. The \texttt{ADR\_BASE = $ORACLE\_BASE}, which is controlled by the database \texttt{diagnostic\_dest} parameter, replaces \texttt{background\_dump\_dest}, \texttt{core\_dump\_dest}, and \texttt{user\_dump\_dest}. The entire ADR repository can be moved to a different location than the initial install, at the same time resetting \texttt{$ORACLE\_BASE}. Just removing or moving the directories will not disable the ADR. The TNS components can be disabled by following this document from MOS: Using and Disabling the Automatic Diagnostic Repository (ADR) with Oracle Net for 11g [ID 454927.1].

A database parameter is used from the time a database is started. This parameter can be changed manually with what is known as a \texttt{pfile} or dynamically with an \texttt{spfile}. A database parameter is found by the following example query:

```
SYS@ORCL11> show parameter base
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_database</td>
<td>boolean</td>
<td>FALSE</td>
</tr>
<tr>
<td>cluster_database_instances</td>
<td>integer</td>
<td>1</td>
</tr>
<tr>
<td>optimizer_capture_sql_plan_baselines</td>
<td>boolean</td>
<td>FALSE</td>
</tr>
<tr>
<td>optimizer_use_sql_plan_baselines</td>
<td>boolean</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Notice how it finds all parameters that contain the word base and also why \texttt{$ORACLE\_BASE} is not here. It is an OS environmental variable, which is controlled outside the SQL*Plus prompt. The outside environment can be queried or manipulated by using host (use \texttt{!} or the keyword \texttt{host}) commands within SQL*Plus. See the following query that uses the Unix \texttt{echo} command to determine where \texttt{$ORACLE\_BASE} is set:

```
SYS@ORCL11> !echo $ORACLE_BASE
/u01/app/oracle
```
You will also see in the documentation that the combination of the following features is known as the Fault Diagnosability Infrastructure:

- ADR plus ADRCI (ADR Command Interpreter) command-line utility
- Alert Log
- Trace Files, Dumps, and Core Files
- Enterprise Manager Support Workbench

It is also important to evaluate the default database jobs and tasks that are created and enabled as of 11g. They are tightly coupled with the new Diagnosability Framework. See the MOS documents: New 11g Default Jobs [ID 755838.1] and 11g: Scheduler Maintenance Tasks or Autotasks [ID 756734.1]. Become familiar with the new 11g Weekday Windows versus the 10g Weekend/Weeknight method of scheduling. This new method gives the DBA more flexibility in determining which day of the week is best for scheduling database maintenance tasks. Adjust the Resource Plan of each Window to allocate hardware and database resources as needed for prioritizing maintenance tasks.

**Advisors and checkers**

Tools within OEM (Oracle Enterprise Manager) known as advisors and checkers monitor and troubleshoot within the Diagnosability Framework. These tools include the Incident Packaging and Reporting, Support Workbench, and Health Monitor. There is a Trace Assistant for Oracle Net Services. Repair Advisors include the SQL Test Case Builder, SQL Repair, and one for Data Recovery used with RMAN.

While this entire new framework is a start towards diagnosing issues, will it really change the way in which Oracle supports and their customers interact? Oracle Configuration Manager, the software piece that uploads your local information to MOS, has been promised to make Service Requests (SR) easier to enter at the same time, resulting in Oracle's support personnel responding faster.

There are database packages that correspond to the diagnosability utilities available on the OEM dashboard. More information on the OEM GUI monitoring console can be found in Chapter 4, Achieving Maximum Uptime of this book. These are health check monitor and policies—DBMS_HM, V$HM_CHECK and Diagnostics—DBMS_SQLDIAG.
The screenshot below will only show a few components as part of the Diagnosability Framework, as each one is found on a different OEM tab. Several of these checkers were run in the example below to search for problems related to a missing temp file.

Out of the box OEM monitoring an 11.2.0.1 database could only partially detect what a DBA would consider a major error—missing temp files. A missing temp file would affect queries that need sort space. Initially, nothing in the OEM dashboard indicated anything was wrong; no alert log error entries appeared.

I knew there was a problem because I intentionally caused it by manually removing the actual temp file from the disk. Only after doing one of the workarounds mentioned in the following MOS document did the alert log error entries become viewable using OEM: Monitoring 11g Database Alert Log Errors in Enterprise Manager [ID 949858.1]. The step that was done was to copy (or create a softlink) the ojdl.jar file from <AGENT_HOME>/diagnostics/lib to the <AGENT_HOME>/sysman/jlib. See the document if this fix doesn't work for your environment.

**Missing temp file resolution**

Since Oracle Database Version 10g, temp files are automatically created on database startup and opened when they cannot be found. I verified that this behavior still exists in 11g by inspecting the alert log after opening the database (where I had removed the temp file), as seen in the following entry:

```
SMON: enabling cache recovery
Successfully onlineed Undo Tablespace 2.
Verifying file header compatibility for 11g tablespace encryption...
Verifying 11g file header compatibility for tablespace encryption completed
SMON: enabling tx recovery
Re-creating tempfile /u01/oradata/orcl/temp01_orcl.dbf
```

This probably explains why the advisors and checkers didn’t detect the missing files, as there is an automatic method of recreation. This makes me wonder: What else does Oracle consider not to be critical?
There were two possible solutions to this issue: recreate the missing temp files while the database was still open, or shut it down and have them automatically recreated on opening. I chose the latter in order to verify the automatic recreate functionality. In this case, the newer diagnosability features inserted an event in the Oracle database spfile, which prevented the database from even mounting, as seen in the following screenshot:

```
I entered a Support Request on MOS because the database would not open (or even mount). This problem was caused by the automatic insertion of an event into the corresponding spfile for the database. Once the following event was removed from the spfile (by creating a pfile and then editing to comment out the event), the database would then open normally:

*.event='1110 incident(missing file)'
```

When monitoring an 11g+ database with OEM Server and Agent Version 10.2.0.5+, the mechanism/metrics for monitoring the alert log have changed significantly. There are two different classes of error—incidents and operational errors:

- Operational errors include Data Block Corruption, Archiver Hung, or Media Failure.
- Incidents include Access Violation, Session Terminated, Out of Memory, Redo Log Corruption, Inconsistent DB State, Deadlock, File Access Error, and Internal SQL Error.

Be aware of these monitoring changes and how they might affect your current processes. At this point I would recommend holding on to any user-created monitoring (such as shell scripting or DBMS_SCHEDULER) that you currently have running.

An alert_[ORACLE_SID].log file and a log.xml file are written to the ADR location, but the alert_[ORACLE_SID].log file is now deprecated as of 11g. This means that there are plans to remove it in future releases.

See Monitoring 11g Database Alert Log Errors in Enterprise Manager Doc [ID 949858.1]
Maintaining Oracle Standards

The following screenshot shows the alert log entries indicating the missing file, but running several of the checkers didn't cause alarms or create an incident (the missing file error ORA-01110 is a generic incident). I expected an incident to be automatically created and viewable by the Support Workbench. At this writing, an open SR had not been resolved to understand why no incident was produced.

The MOS document 949858.1, which was mentioned earlier in this section, has more information on this problem and a couple of workarounds for non-detected events. You can create what is known as a User-Defined Metric for OEM monitoring or set a system-wide event to make sure incidents are created for specific errors raised in the alert log. Both of these workarounds add up to more work for you, but think of them as a change to the baseline functionality. Changing baseline functionality is a configuration change that you will have to track and maintain over time.

The Diagnosability Framework is meant to make the interactions with MOS and entering Service Requests for critical errors easier. For incidents to be tracked automatically, you must configure OEM using Oracle Configuration Manager (OCM) to package and upload the results to Oracle support. Incidents can be manually packaged and uploaded using OEM's Support Workbench or the adrci command-line utility.
In the past, MOS depended on customers running an RDA (Remote Diagnostic Agent) to upload system information for each Service Request. Now MOS is moving towards the model of daily updated configurations from a locally installed collector, so it is purported that RDAs won't be used as often.

Oracle service personnel will still ask for the output from an RDA for troubleshooting complex problems. Its basic job is to gather extensive environment information. Running an RDA before entering an SR would give insight into what information the Service Engineer from MOS would be looking at, and might even allow you to solve the problem without entering an SR. See the following MOS document: Subject: Remote Diagnostic Agent (RDA) 4 – Getting Started [Doc ID: 314422.1 and Doc ID: 330760.1]. Most DBAs don't know about the more advanced features of the RDA utility.

Some of the most underused features of RDA include:

1. RDA report files are viewable by using a browser on the following file:
   
   `<rda_directory>output_directory>/report_group>__start.htm`

2. Security filtering is available to remove sensitive information such as IP addresses, domain names, and user names. This information is not really needed for most Service Requests.

3. Oracle Configuration Manager and RDA can be installed together at the same time.

4. There is a Testing Option (-T) available for certain modules. See the Remote Diagnostic Agent (RDA) 4 – Content Modules Man Page [ID 330760.1] document for specifics.

5. RDA profiles are provided, which may or may not fit your needs. It basically keeps you from having to answer the long list of yes/no questions. A more viable option would be to customize an RDA profile for future use, which can be transferred from one server to another.

6. User Defined Collection is available and it will collect custom files that you can add.

7. An RAC Cluster Guide is available for collecting multi-instance RDAs.

Don't forget to occasionally check for the newest release, as MOS constantly improves this utility.
If you use any of the 11g products, including the client, be sure to set the environment variable $ORACLE_BASE. This will control where the alert, trace, and log files are located. Utilize the built-in environmental variable within your own scripts and maintenance routines to become used to the new adrci command-line utility. If you don't set the variable, the log files will appear in $ORACLE_HOME/log even for client installations.

The default editor for adrci is vi and since a standard Windows install won't have the editor vi, make a copy of notepad.exe or wordpad.exe. Move that copy to a location in the Windows $PATH renaming that exe file to vi.exe. The adrci command-line utility has a set editor command but may yield unexpected results in Windows.

To script an automatic purging of date-aged logs, use adrci in a shell script with a text file that contains the commands as below. You will need to substitute the ORACLE_SID for all of the sid entries in the following parameter file:

```
#adrci_commands.par
#Change the nodename and listener_sid for your environment
set echo on
set homepath diag/tnslsnr/nodename/listener_sid
purge -age 10080 -type alert
purge -age 10080 -type incident
purge -age 10080 -type trace
purge -age 10080 -type cdump
set homepath diag/rdbms/sid/SID
purge -age 10080 -type alert
purge -age 10080 -type incident
purge -age 10080 -type trace
purge -age 10080 -type cdump
quit
```

See the file adrci_maint.ksh in the code provided for this chapter on the shell script that calls the adrci_commands.par found earlier.

As a side note, since the ADRCI interface (adrci command-line utility) will only change the XML-formatted alert file (log.xml), you will need to manually schedule log rotation, archiving, and eventual removal of the text-based alert logs. This behavior is similar to the older, manual method where files are put according to the database parameters *dump directories as mentioned earlier.
Environmental variables and scripting

Unix scripting will be one of the most powerful tools in your arsenal and only with constant use will you become proficient. Your standards for scripting need to produce code that is robust, modular, consistent, and scalable. This book won’t cover everything you need to know about scripting. There are many excellent lessons available from your operating system vendor. See the following documents for more information specific to how Oracle products behave in a Unix environment:

- Note: 131207.1 Subject: How to Set Unix Environment Variables
- Note: 1011994.6 Subject: How do Applications Act in Different Unix Shells.
- How to Integrate the Shell, SQL*Plus Scripts and PL/SQL in any Permutation? [ID 400195.1] (Excellent document)

Usually cron is used for scheduling in Unix and the at command with Windows.

For Oracle utilities that run outside the database (or which must only run when the database is down), Unix shell scripting is best used. A well-written script would parse /etc/oratab for a list of ORACLE_SIDS on a single server for tasks such as parsing log files, consistent backups, exports, dbverify, and RMAN. If you have a solitary database, then DBMS_SCHEDULER can be used with a combination of SQL and PL/SQL integrated with OS commands.

Occasionally, DBAs rename the oraenv located in $ORACLE_HOME/bin when they have added custom code to the /usr/local/bin/oraenv (default location on Linux), so that they can make sure they know exactly which one is executed at run time. If you have any problems related to running a modified oraenv file, check which one is being executed and adjust the $PATH variable as needed.

The following is a list of some of the Oracle-provided Unix commands with a quick synopsis of their purpose. Most of them are located in $ORACLE_HOME/bin:

- wrap—encrypts stored procedures for advanced use
- oerr—displays oracle errors. Syntax: oerr ora 12154
- Sysresv—instance and shared memory segments
- Tkprof—formats output trace file into readable format
- Dbshut—shell script to shut down all instances
- Dbstart—shell script to start up all instances at boot
- Dbhome—sets ORACLE_HOME
- Oraenv—sets environmental variables for ORACLE_SID
- trcasst—trace assistant
Guidelines for scripting

These are general suggestions for someone with some experience with a Unix operating system. You will need more background information than what is covered in this book to understand this section. The best sources of information will be the software vendor of your operating system, because there are small differences between the different versions and flavors of Unix. As with any code, you must test on a non-production system first, as inexperience may lead to unexpected results.

Separating the configuration file

Use the OS-provided default profile for environmental variables, but use a separate configuration file to set the $ORACLE_SID variable. There are several key environmental variables that are commonly used with Oracle software, all of which are found in the documentation specific to the operating system. Optimal Flexible Architecture (OFA) mentions setting the ORACLE_SID in the profile, but if you have more than one database or Oracle product installed on a system, it is best done interactively. A default profile is loaded when you first log in to Unix. So if the ORACLE_SID is loaded when you log on, what happens when you want to change ORACLE_SID(s)? This is when the environment becomes mixed. It just keeps appending the $PATH variable each time you execute the oraenv script. Which set of executables will you find first? It will be those executables for which you ran oraenv the first time.

At this point I wanted to execute SQL*Plus out of the 11g directory, but was able to determine that the 10gR2 client was first in the $PATH. How did I know that? Use the which Unix command to find out.

```
oracle@yournode.domain:/u01/app/oracle[orcl]
$ which sqlplus
/u03/home/jobs/product/10gR2/client/bin/sqlplus
```

It is also a good practice to use a separate terminal window for each ORACLE_HOME. Normally, you will operate with multiple windows open, one for each ORACLE_HOME or ORACLE_SID in different colors to visually remind you which one is production.
The example profile is provided in the code section for this chapter: example_profile.txt. The profile sets the entire user environment at first logon. This one is specific to the ksh or korn shell on the Linux operating system and will also work for bash. Differences in bash include that the line history is scrollable with the up and down arrows instead of vi commands.

To set the ORACLE_SID and activate all of the variables located in profile, source the file oraenv (bash, Bourne, or korn shell) or coraenv (C shell). Source means the variables will be in effect for the entire session and not just the current line in the command window. You indicate source by putting a '.' (dot) in front of the file. As the oraenv file is located in /usr/local/bin (on Linux) and this location is in the $PATH, typing it at the command line works. Putting key Oracle files, such as oraenv, oratab, and oraInst.loc, in locations that will not be affected by standard Oracle installations is also part of the OFA. The oraenv script is installed into /usr/local/bin/ automatically when running .runInstaller for the first time. Notice the prompt that you will see if you use the command as in the profile listed above:

```
ora@hostname.domain:/u01/app/oracle[]
> . oraenv
ora@hostname.domain:/u01/app/oracle[newdb]
> . oraenv
```

A note about prompts: Every person will want to customize their own prompt so; look around for various versions that tickle your fancy. This one is better than most examples to compare to. Notice the small difference in the prompt before and after? ORACLE_SID is now defined because oraenv is executed, which also runs dbhome (also located in /usr/local/bin), but these scripts require a valid entry in /etc/oratab. If you type the ORACLE_SID incorrectly on Unix, this will be case sensitive. It will ask where the ORACLE_HOME is if there is no entry in oratab. Making an ORACLE_SID lowercase conforms to the OFA standard (see the install guide for your OS for more information). Some DBAs use an uppercase ORACLE_SID because it makes it more prominent for any type of naming convention and is meant to reduce human error.
You can use an **ORACLE_SID** entry in the `oratab` file to reference other Oracle products such as the Grid Control Intelligent Agent. The ea, which is an **ORACLE_SID** in the following `oratab` example, is what I use to indicate the Enterprise Manager Agent. The third letter after the **ORACLE_HOME** (N) indicates when Oracle-supplied utilities (like `db_shut` and `db_start`) are *not* to execute against this **ORACLE_HOME**. I personally use the N for my own scripting to indicate which utilities shouldn't run against this **ORACLE_HOME**. What this does take advantage of Oracle-provided files—`oratab` and `oraenv`—to accomplish other types of tasks. This is only a suggested use. There are other ways of setting environmental variables for non-database products.

You will need to create a test database to work through all of the examples and practice scenarios in this book. How should you create the test database? Use the Oracle-provided `Database Configuration Assistant (DBCA)` utility to create a test database. There are default templates provided that will work for most of the tasks outlined in this book. If you are interested in duplicating some of the advanced tasks (like Data Guard), then it will require the installation of the Enterprise Edition of Oracle Database. All tasks in this book were done with 11.1.0.7 version of Oracle Database with some references to 11.2.0.1, which had just been released.

**Host commands relative location**

This will be important as you begin scripting. Host commands are relative to the location of the executable. As a general rule, you should execute database-specific utilities (imp, exp, datapump, RMAN, and so forth) on the server where the database is located in the correct **ORACLE_HOME**. This reduces the amount of issues such as core dumps and version compatibilities. This is different from what is usually thought of as a client utilities such as SQL*Plus.
There are exceptions to this rule, for it is recommended to run a compiled code (C, C++, Cobol) on a separate server rather than a database. See the following document for setting the `TWO_TASK` variable when using a separate node for compiled programs. `TWO_TASK` is an environmental variable. Subject: How to Customize Pro*C and Pro*Cobol Makefiles demo_proc.mk And demo_procob.mk On Linux/Unix [Doc ID: 602936.1].

Another exception to running a utility on a different node is when you are performing migration to a different version or patchset, or running the DATAGUARD broker. Both of these situations are discussed later in this book in the Chapter 7, and Chapter 5.

Notice the WARNING! message that is set using the new 11g `sqlnet.ora` parameter `SEC_USER_UNAUTHORIZED_ACCESS_BANNER`. The `sqlnet.ora` file is part of the SQL*Net components of Oracle RDBMS, which handle the communication between clients and the database.

```
oracle@nodename:/u01/app/oracle/admin/newdb[newdb]
> sqlplus /nolog
SQL*Plus: Release 11.1.0.7.0 - Production on Thu Nov 5 19:00:29 2009
Copyright (c) 1982, 2008, Oracle. All rights reserved.
@> connect / as sysdba
#################################################################
### WARNING! This computer system is the property of YOUR
ORGANIZATION
and may be accessed only by authorized users.
Unauthorized use of this system is strictly prohibited and may be
subject to criminal prosecution.
Connected.
SYS@newdb>
```

If you wanted to execute something that is available on the operating system level, then you would use a `host` command (either Windows or Unix), or on Unix the `!` symbol. The output below shows that I am logged into the `newdb` as `sys` and lists (`ls` command) the files located in the `$ORACLE_HOME/sqlplus/admin` directory:

```
SYS@newdb>  ls $ORACLE_HOME/sqlplus/admin
glogin.sql  help  libsqlplus.def  plusroe.sql  publd.sql
SYS@newdb>  vi $ORACLE_HOME/sqlplus/admin/glogin.sql
```
Notice how the SQL prompt is populated with the `ORACLE_SID` and the username that is connected to the database. This is done by adding a line to the `glogin.sql` file, which can be done within SQL*Plus as shown below (I used the text editor `vi`):

```sql
set sqlprompt "$_user@"_connect_identifier> "
```

Host commands work based on the location of SQL*Plus. If you want to execute these same commands from a Windows desktop connecting to a remote Unix database, then it would require a Windows equivalent command like Notepad or another editor. If you have Cygwin installed and configured on the Windows desktop, then it would allow you to run Unix-equivalent commands like `vi`.

**Separating the variable part of the script into its own configuration file**

There are scripts that will need some variables set, but you don't necessarily want to use the `profile` to set a variable at every login. The variables may need to contain commands specific to applications such as RMAN, SQL*Plus or specific to certain tasks. This is where a configuration file comes in handy, which is a personal preference as to what you call them. Be sure not to use reserved keywords for variables, as that leads to unexpected results. In the example below, we use `email` instead of the word `email`.

To call this configuration file, you source it starting with the prerequisite `. (dot) within a script. This file can also be used in the custom code section of the `oraenv` file.
Don't hardcode values; reference a configuration file and password file at runtime

Values include items such as a directory path, ORACLE_SID, e-mail, or file locations as shown in the above newdb.conf file. Passwords should be in a separate hidden file-protected location (chmod 400, which is only readable by Oracle) to be read at runtime by simply inserting a small piece of code and an appropriate variable in each script:

```bash
#.oracle.passwd
system/password

PASS=`grep system /u01/app/oracle/.oracle.passwd | cut -f 2`;
exr PASS
$ORACLE_HOME/bin/exp userid=$PASS parfile=$HOME/export$db.par
```

The following line actually pulls the ORACLE_SID out of the oratab file. This is useful for executing against multiple databases on a single node. See the script labeled coalesce_tablespace.ksh for a complete example.

```bash
cat /etc/oratab | while read LINE
do
   case $LINE in
      
"*"
   ) ; #comment-line in oratab
      *
   )
      # Proceed only if third field is 'Y'.
      if [ "\"echo $LINE | awk -F: \" '{print $3}' -\" = "\"Y\" \" ]; then
         ORACLE_SID=\"echo $LINE | awk -F: \" '{print $1}' -\"
         export ORACLE_SID
         ORACLE_HOME=\"echo $LINE | awk -F: \" '{print $2}' -\"
         export ORACLE_HOME
```
This small script shows how a local script on a database node can be run without a password, as the database has a password file and the startup initialization parameter remote_login_passwordfile=EXCLUSIVE. A password file is created for a specific database with a default file name format usually found in $ORACLE_HOME/dbs/orapw<SID>.

```
#!/bin/ksh
#Source the configuration file as follows.
. $OTILS/newdb.conf
$ORACLE_HOME/bin/sqlplus -s <<EOF
connect / as sysdba
set pagesize 0 feedback off termout off linesize 200
spool output.txtselect tablespace_name from dba_tablespaces;
spool off
EOF
exit
```

Using a hidden password file for scripting is more secure than embedding clear text passwords in the script itself. There is also a relatively new feature in Oracle Database 10gR2 and above called the External Password Store. This component is also part of the Advanced Networking Option (options are always an additional license cost) and is another way to store passwords. If used for password functionality only, the External Password Store is free of charge.

http://download.oracle.com/docs/cd/E11882_01/license.112/e10594/editions.htm#CJACGHEB: This document details the differences between the different editions of Oracle. Search for Secure External Password Store; it is included with the Enterprise Edition of Oracle.

It requires the creation of an Oracle Wallet that stores encrypted username and password combinations for batch jobs, scripts, and even application code. This is one of the most secure ways to connect to the database while providing a single location for changing passwords on a regular basis. See the Oracle Documentation, Oracle Database, and Security Guide for your release for instructions on creating and maintaining a password store. See the following white paper for instructions that don't require a MOS account:

Putting variables at the top of the script with curly braces

This makes it easier to spot for editing and troubleshooting. Mixed case would make them stand out even more. Curly braces will resolve to the value of a variable that had already been defined at the top of a particular script.

```
$ORACLE_HOME # variable at the top of a script
${ORACLE_HOME} # Refers back to the variable
```

Moving functions to a centralized file to be reused

All scripts can use any of the functions within by referencing the function name. The example below illustrates this:

```
#!/bin/ksh
# otils.fnc
function verify_directories {
    if [ ! -d ${MY_BASEDIR} ];then
        mkdir ${MY_BASEDIR}
    fi

    if [ ! -d ${MY_BASEDIR}/scripts ];then
        mkdir ${MY_BASEDIR}/scripts
    fi

    if [ ! -d ${MY_BASEDIR}/archive ];then
        mkdir ${MY_BASEDIR}/archive
    fi
}
```

And an example shell script that will call the separate function is simply run and sourced by putting a dot in front of the filename. In this case, the otils.fnc file can be found in the $PATH environmental variable.

```
------
#!/bin/ksh

. otils.fnc
```
Validating the use of the script

To validate a script that should only be run by a certain user ID or on a certain host, only use `whoami` and `uname`. See the Move functions to a centralized file to be reused section for the idea of a centralized function file, as this would be a good candidate.

```bash
if [ `whoami` != "oracle" ]; then
  echo "error. Must be oracle to run this script"
  exit 1
fi

if [ `uname -a | awk '{print $1}'` != "databasenode" ]; then
  echo "error. This script only to run on databasenode server"
  exit 1
fi
```

The `if-fi` part of this code is used to evaluate something. If it is true, then do the next step. The `fi` indicates where the end of this section of code should stop.

Using SQL to generate code

The script provided for this chapter called `cleanout.sql` generates a file that is then run by SQL*Plus, which demonstrates the capability to run certain host commands from within the database. This is done by using the keyword `host` within the script, and on Unix you can also use the exclamation point (!) to indicate host commands. The Oracle user will be limited to executables that are in the `$PATH` and have the appropriate file permissions.

This ability to use SQL*Plus to generate SQL commands as well as execute OS commands by embedding SQL syntax in a script is a powerful tool for a DBA. This type of execution will dynamically generate spooled SQL commands. The resulting spooled SQL file can then run in the same script or edit it as needed before running manually.

The following scripts are especially handy for one-time mass updates, such as switching users to a new profile or expiring passwords. The single ampersand symbol (`&`) will substitute the value for that variable you provide a single time; two ampersands will substitute every time that variable is found in a script.
The first script, `grant_table_role.sql`, grants all privileges for a list of tables to a role that belongs to a certain schema. The second script inserts a username into a table generated by pulling the first part of an e-mail address, that is, everything before the @ symbol. The `chr(10)` puts in a newline and the pipe symbols `||` append each item.

```sql
--grant_table_role.sql
set echo      off
set feedback  off
set heading   off
set pagesize  0
set timing    off
set trimspool on
set verify    off
spool grant_table_&USERROLE.&TABLEOWNER..sql
select 'grant select,insert,update,delete,references on ' ||table_name || ' to &&USERROLE; ' from dba_tables where owner = '&&TABLEOWNER';
spool off
exit

--update_email.sql
set echo      off
set feedback  off
set heading   off
set pagesize  0
set timing    off
set trimspool on
set verify    off
spool update_email_users.sql
select 'update table_name set table_name_pidm = ||EMAIL_PIDM || ', ' ||'table_name_username' ||' = ' || chr (10) || (select substr(email_address,1, instr(email_address,'@')-1) from email_table)|| 'table_name_user_id= BATCH, table_name_activity_date =sysdate;' from general_emal;
spool off
@ update_email_users.sql
exit
```
In the e-mail example above, the script that was created in the first step is automatically called after the spool-off command.

```sql
--This script is a generic create user script.
--create_users.sql
set echo off
set feedback off
set heading off
set pagesize 0
set timing off
set trimspool on
set verify off
spool create_mass_users.sql
select 'create user ' ||username || ' identified by ' ||'"'||TO_CHAR(BIRTH_DATE,'MMDDYY')||'"'||' ;'|| chr(10)||
 'alter user ' || username || chr(10) ||
 ' default tablespace USERS;' || chr(10) ||
 'alter user ' || username || chr(10) ||
 ' temporary tablespace TEMP;' || chr(10) ||
 ' grant CONNECT to ' || username || chr(10) ||
 'grant USR_ROLE to ' || username || chr(10) ||
 'grant USR_QUERY to ' || username || chr(10) ||
 'grant USR_STAFF to ' || username || chr(10) ||
 ' alter user '||username||' default role CONNECT, USR_STAFF ;'
 from table_name, person_table
where table_name_pidm=person_table_pidm;
spool off
-- commented out the last statement to double check before running.
-- @ create_mass_users.sql
exit
```

The only difficult part about this process is that the number of quotes needed depends on the data retrieved. If it is a literal string (data that has to have quotes to do the insert), then you will need a corresponding set of two single quotes to do this. See the example create_users.sql.

All of the queries used in these examples may not work for you, depending on the data you are trying to retrieve. They are intended to provide examples of how to use the embedded SQL technique for different types of DBA tasks.

It takes some practice, but this is a tool that will make your job easier. The various `set` commands are specific to SQL*Plus. These settings remove extra characters that end up in the final script. Removing those extra characters from the final script allows it to be run in SQL*Plus without editing the file first.
Unix "ps" Command Exposes Password When Using Export/Import, SQL*Plus, Tkprof [ID 372964.1] provides important information about concealing passwords, which is easily found with the Unix `ps` command when running scripts.

Helpful Unix commands

The following table lists some helpful Unix commands:

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Unix commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show files greater than (&gt;2) 2 GB</td>
<td><code>find /u02/oradata -size +200000000c -ls</code></td>
</tr>
<tr>
<td>Show files less than (&lt;) 2 GB</td>
<td><code>find /u02/oradata -size -200000000c -ls</code></td>
</tr>
<tr>
<td>Remove trace files not modified in the last 30 days</td>
<td><code>find /u01/app/oracle -name &quot;*.trc&quot; -mtime +30 -exec rm {} \;</code></td>
</tr>
<tr>
<td>Find scripts where new users are created</td>
<td><code>find $ORACLE_HOME -type f -exec grep -il &quot;identified by&quot; {} \;</code></td>
</tr>
<tr>
<td>Find sqlplus scripts that might have passwords in them</td>
<td><code>find /u01/app/oracle/admin -type f -exec grep -il &quot;sqlplus&quot; {} \;</code></td>
</tr>
<tr>
<td>Find all files that are owned by Oracle</td>
<td><code>find /u01 /u02 /u03 -user oracle -print</code></td>
</tr>
<tr>
<td>Remove all export files that are larger than 1 MB and older than 30 days</td>
<td><code>find $ADMIN -name &quot;*.dmp&quot; -size +1048576c -mtime +30 -exec rm {} \;</code></td>
</tr>
<tr>
<td>Find linked files</td>
<td><code>find /u01/app/oracle /u*/oradata -type l -ls</code></td>
</tr>
<tr>
<td>Find files modified in the last 24 hours</td>
<td><code>find /u01/app/oracle/admin/orcl -type f -mtime -1 -print</code></td>
</tr>
<tr>
<td>Show total size of all subdirectories with Disk Usage command on a disk (sorted in reverse order)</td>
<td><code>cd /u02/oradata</code></td>
</tr>
<tr>
<td></td>
<td>`du -s *</td>
</tr>
<tr>
<td></td>
<td><code>du -sh</code> will provide a shortened readout in gigabytes</td>
</tr>
<tr>
<td></td>
<td><code>cd /u02/oradata/orcl</code></td>
</tr>
<tr>
<td></td>
<td>`du -k *</td>
</tr>
<tr>
<td></td>
<td><code>date; who</code></td>
</tr>
</tbody>
</table>
Maintaining Oracle Standards

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Unix commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>To debug a script with sh or bash and display the environmental variables</td>
<td>sh -vx script.sh</td>
</tr>
<tr>
<td>Bash shell debugging set command, a negative in front turns it on, a plus sign turns it off</td>
<td>set-x: Display commands and their arguments as they are executed. set-x set+x</td>
</tr>
<tr>
<td>set-v: Display shell input lines as they are read</td>
<td>.#!/bin/bash -xv</td>
</tr>
</tbody>
</table>


Reducing operating system differences with common tools

When your enterprise environment contains several different operating systems and database versions, it is best to have tools that work across scopes. We have already mentioned scripting as well as using DBMS_SCHEDULER and PL/SQL. There are also several software packages that offer GUI interfaces for real-time monitoring capability.

Oracle has a product called Grid Control (GC); it is the Enterprise Manager for the Enterprise. You can install a standalone version of Enterprise Manager in each database (called Database Control) or use GC with a centralized repository for all of your Oracle products. Grid Control is a free product from Oracle, the Management Packs have an additional license cost. Most customers purchase the DIAGNOSTICS & TUNING to take advantage of the extensive performance tuning features found in the Automatic Workload Repository, and not just for the OEM Interface.

The trade-off is a superb enterprise-wide tool, but one that requires an additional repository database for the infrastructure components of GC. The database or repository is not completely self-managed, but has not shown itself to be a large burden in terms of overhead. GC is a complex product that is now maturing in terms of scalability and flexibility. Challenges include limited migration paths when moving the different components to new hardware and/or hardware platforms.

GC version 10.2.0.5 has some of the best features so far: mass deployment of the agent, and installing the software and configuring later options for several of the components. Both of these new features directly address past migration and deployment issues.
Other third-party vendors with widely adopted database monitoring tools include Veritas, Quest, EMC, Embarcadero, and CA. Each of them have their own positive and negative features. Each one should be available for a full trial testing period before investing.

**Chapter 2**

**Configuration management, release management, and change control**

One of the largest changes to Oracle is the recent acquisition of several other software lines and technologies. Oracle has combined all of these technologies and customers under a single support site called My Oracle Support at [http://support.oracle.com](http://support.oracle.com), effective from Fall 2009. Along the way, Oracle also completely redesigned the interface, making it flash-based in order to provide a personalized GUI.

To take full advantage of the personalization features, you will need to install a free utility on each node and each `ORACLE_HOME` you would like to monitor. The following paragraphs outline several reasons for use and suggestions for getting started. Please review and discuss with the management the security implications of uploading critical information to a support website before proceeding.

**Configuration management**

Are you the only Oracle DBA in your company? How do you provide disaster recovery and redundancy for personnel in that situation?

MOS has a tool that provides an *Automatic Document Repository* (my words) called Oracle Configuration Manager (OCM). This tool has been mentioned several times in this chapter, but its real purpose is to manage all of your configurations (different systems, servers, databases, application servers) when dealing with Oracle support.

It is automatic in the sense that if you are out of the office, temporarily or permanently, the system configurations are available for viewing by anyone with the same Oracle *Customer Support Identifier* (CSI) number. The information is also available to Oracle support personnel. The repository is located on My Oracle Support. The systems are for you to choose, whether you want to only include production and/or non-production systems.
What information does OCM collect and upload? It contains extensive hardware details, software installs (not just Oracle products), databases, and Oracle application servers. There is enough information to help in recreating your site if there is a complete disaster. The GUI interface allows managers and other IT personnel to see how nodes and applications are related and how they fit into your architectural framework. The information can only be updated by the upload process.

**Using OCM in disconnected mode with masking**

There is sensitive information being collected from the OCM tool. If you are employed by an organization that doesn't allow you to reveal such information or allow direct access by the servers to the Internet, there are steps to improve the security of this upload process. This section is highly recommended to be reviewed before enabling OCM. You must know what types of information are there and how that information is used before enabling uploading capabilities to a support website.

To disable the collection of IP and MAC addresses, you add the following entries to the $ORACLE_HOME/ccr/config/collector.properties file.

To disable the collection of network addresses, add the following entry:

```
ccr.metric.host.ecm_hw_nic.inet_address=false
```

To disable the collection of the MAC address, add the following entry:

```
ccr.metric.host.ecm_hw_nic.mac_address=false
```

The OCM collector collects the schema usernames for databases configured for configuration collections. The collection of this information is filtered or masked when `ccr.metric.oracle_database.db_users.username` is assigned the value of 'mask' in the $ORACLE_HOME/ccr/config/collector.properties file. The default behavior of the collector is to not mask this data.

MOS customers may request deletion of their configuration information by logging a Service Request (SR) indicating the specific configuration information and scope of the deletion request.

Disconnected mode is carried out with something called Oracle Support Hub, which is installed at your site. This hub is configured as a local secure site for direct uploads from your nodes, which the hub can then upload to MOS through the Internet. This protects each of your nodes from any type of direct Internet access.
Finally, there is a way to do a manual upload of a single node using the method outlined in the MOS document 763142.1: *How to upload the collection file ocmconfig.jar to My Oracle Support for Oracle Configuration Manager (OCM) running in Disconnected Mode.* This is probably the safest method to use for OCM. Run it for a specific purpose with appropriate masking built-in and then request the information to be deleted by entering a SR request.

These tips came from these locations as well as the OCM licensing agreement found on MOS:

- [http://download.oracle.com/docs/html/E12881_01/toc.htm](http://download.oracle.com/docs/html/E12881_01/toc.htm)


Each node with an installed OCM collector can be automated to upload any changes on a daily basis or interval of your choice. OCM is now an optional part of any of the 10.2.0.4+ Oracle Product GUI installs. The OCM collector is also found by logging into MOS and selecting the collector tab. It is recommended to use at least the 3.2 version for ease of installation across the enterprise.

Be aware! The collector install actually creates the Unix `cron` entry to automatically schedule the uploads.

**Mass deployment utility**

The OCM collector utility has been out for over a year, but a recent enhancement makes installation easier with a mass deployment utility. On the MOS collector tab, find *Configuration Manager Repeater & Mass Deployment Tools* and the OCM Companion Distribution Guide.

The template file required to install the collector on multiple servers is in `csv` format, which you may find difficult to edit using `vi` or `vim`. The template doesn't have an initial entry and the length is wider than the average session window. Once the first entry is filled out (try using desktop spreadsheet software), editing this file with a command-line tool is easier. It has a secure password feature so that no password is stored in clear text. You can enter a password at the prompt or allow the password utility to encrypt the open text passwords in the template file during the install run.
Running the utility runs very quickly from a single node that has SSH access to all entries in the template. It auto detects if OCM was already installed and bypasses any of those entries. You may encounter an issue where the required JAVA version is higher than what is installed. Other prerequisites include SSH on Linux or CYGWIN for Windows.

A downside is that all configuration information is available to everyone with the same CSI number. In a small IT shop, this isn't a problem as long as MOS access is maintained properly when personnel changes. Providing granular group access within a CSI number to your uploaded configurations is a highly anticipated feature.

**Release management**

As a DBA you must be consistent in the different aspects of administration. This takes dedication to keep all of your installed Oracle products up-to-date on critical patches. Most DBAs keep up-to-date with production down issues that require a patch install. But what about the quarterly security fixes? The operating systems that your system admin is in charge of will probably be patched more regularly than Oracle. Why is that the case? It seems to take an inordinate amount of effort to accomplish what appears to be a small task.

Newer versions of Oracle are associated with major enhancements—as shown by the differences between versions 11.1 and 11.2. Patch sets contain at least all the cumulative bug fixes for a particular version of Oracle and an occasional enhancement as shown in the version difference between 11.1.0.6 and 11.1.0.7. Oracle will stop supporting certain versions, indicating which is the most stable version (labeling it as the terminal release). For example, the terminal release of Oracle 10.1.x is 10.1.0.5, as that was the last patch set released. See the following document on MOS for further information on releases—*Oracle Server (RDBMS) Releases Support Status Summary* [Doc ID: 161818.1].

In addition to applying patch sets on a regular basis (usually an annual event) to keep current with bug fixes, there are other types of patches released on a regular basis. Consider these to be post-patch set patches. There is some confusing information from MOS, with two different methods of patching on a quarterly basis (Jan, April, July, Oct.)—Patch Set Updates and Critical Patch Updates. CPUs only contain security bug fixes. The newer method of patching—PSU—includes not only the security fixes but other major bugs. These are tested as a single unit and contain bug fixes that have been applied in customers' production environments.
See the following for help in identifying a database version in relationship to PSUs:

MOS Doc ID 850471.1
1st digit—Major release number
2nd digit—Maintenance release
3rd digit—Application server release
4th digit—Release component specific
5th digit—Platform specific release
First PSU for Oracle Database Version—10.2.0.4.1
Second PSU for Oracle Database Version—10.2.0.4.2

While either PSUs or CPUs can be applied to a new or existing system, Oracle recommends that you stick to one type. If you have applied CPUs in the past and want to continue—that is one path. If you have applied CPUs in the past and now want to apply a PSU, you must now only apply PSUs from this point to prevent conflicts. Switching back and forth will cause problems and ongoing issues with further installs, and it requires significant effort to start down this path. You may need a merge patch when migrating from a current CPU environment, called a Merge Request on MOS.

Important information on differences between CPUs and PSUs can be found in the following locations. If there is a document number, then that is found on the MOS support site:

http://blogs.oracle.com/gridautomation/
http://www.oracle/technology/deploy/security/alerts.htm
Doc 864316.1 Application of PSU can be automated through Deployment Procedures
Doc 854428.1 Intro to Patch Set Updates
Doc 756388.1 Recommended Patches
Upgrade Companions 466181.1, 601807.1
Error Correction Policy 209768.1
Now to make things even more complicated for someone new to Oracle; let's discuss recommended patches. These are released between the quarterly PSUs and CPUs with common issues for targeted configurations. The following are targeted configurations:

- Generic—General database use
- Real Application Clusters and CRS—For running multiple instances on a single database with accompanying Oracle Clusterware software
- DataGuard (and/or Streams)—Oracle Redo Apply technology for moving data to a standby database or another read/write database
- Exadata—Vendor-specific HP hardware storage solution for Oracle
- Ebusiness Suite Certification—Oracle's version of Business Applications, which runs on an Oracle Database

Recommended patches are tested as a single combined unit, reducing some of the risk involved with multiple patches. They are meant to stabilize production environments, hopefully saving time and cost with known issues starting with Oracle Database Release 10.2.0.3—see Doc ID: 756671.1.

**DBA issues with patching**

Here is a list of the most frequently experienced issues with the patching process for Oracle products:

- What patches should be applied?
- No advance notification if there is a conflict.
- Patch conflicts that cause rollbacks.
- Merge requests for patch conflicts.
- Troubleshooting patch application failure.
- Conflicting and resolving differences in CPUs and PSUs.
- Should I apply the recommended patches?
- Am I the first person to deploy this patch?
- Documenting versions and patches across all systems.
Oracle recently added a new functionality to their support website called Patch Plans. It requires the 3.2+ version of the MOS Configuration Manager Collector (which is the collector component of OCM). It was just mentioned a few sections earlier and is available for download from the My Oracle Support website. Once the collector is installed and your system configuration information is uploaded to MOS, you can create a patch plan.

The following screenshot contains my list of patch recommendations for all of the ORACLE_HOME(s) that have a collector installed and configured for uploading to Metalink.

There is an alternate login page for non-flash users—http://supporthtml.oracle.com, but this will not include any of the personalization features such as Patch Plans or uploaded configuration details.

There is one patch plan associated with the patch number 8534378, as shown by the folder icon. I purposely moved the sliding components of this dashboard so that the sensitive information is covered. You can also right-click to select a patch to add to a new plan or existing plan, download, or suppress this patch. The existence of a patch plan is indicated by the folder icon next to a certain patch, as seen in the very first entry of the list of patch recommendations. If you don’t have any configurations, you will not see a list of patch recommendations.
Maintaining Oracle Standards

The next screenshot is the full screen of a selected patch 8534378, which is the CPU for July 2009. Notice on the right the Coming Soon! banner, which is reserved for the community functionality that wasn't available at the time this book was written. This area of MOS will include end users' feedback on this particular patch and will also list the number of times it was downloaded over the past few months.

Applying a patch without integrating MOS with OCM

Here are generic instructions for applying a patch without MOS and OCM:

1. Download the patch, PSU, CPU, or patch set.
2. Create a new ORACLE_HOME for testing if alternating between different ORACLE_HOMES. Otherwise, all Oracle software running in the home to be patched will have to be shut down.
3. Install if there aren't any conflicts. If there is a conflict, then the patch is rolled back.
4. Check if any key patches are also rolled back.
5. Submit a merge request on MOS for key patches.
6. Download the merge request to start the patch process again.

Using the new patch plan functionality with OCM installed and uploaded to MOS

1. Patch Plan automatically checks for missing prerequisites. This functionality has not been shown to be consistent across the board yet.
2. It checks any conflicts before downloading.
3. It request a merge if required before downloading.
4. It will also check recommended patches for your configuration.
The following documents from MOS contain the manual methods for determining whether a patch will conflict using the Oracle supplied opatch utility. More details about opatch use are outlined in Chapter 7. If your organization doesn't want to install or configure the OCM collector for security reasons, then the following is a viable option for determining patch conflicts.

Refer to How to find whether the one of the patches will conflict or not? [Doc ID: 458485.1]. Here are a couple of examples of how to use this command-line utility outlined in the document:

```
opatch prereq CheckConflictAmongPatchesWithDetail -phBaseDir $ORACLE_HOME/patches

opatch prereq CheckConflictAgainstOHWithDetail -phBaseDir $ORACLE_HOME/patches
```

**Change control**

To keep from constantly writing the same pieces of code over and over again and reconciling differences between servers, a common storage area for DBA code needs to be established. It is most likely that your programming staff already has access to a code repository. The two largest open source versions are CVS (Open Source Version Control) and Subversion (SVN). They are both equally efficient and powerful tools with slightly different features. If version control software is already deployed in your enterprise, then ask for a repository for database administration and limit the access to that repository.

Do not store passwords in a repository that means 'currently in use' or 'previously used'. Find an enterprise-wide password utility such as KeePass that will store encrypted passwords for all of the servers and databases. Integrate SSH PUTTY with KeePass so that it launches a terminal window without having to type the password. See the following blog for additional information on other password utilities like KeePass: http://princessleia.com/journal/?p=1235.

One way to use version control software is to check out the repository on each node and use the command-line version on that node. While you can check out code on your desktop, you will still have to shuttle any code changes via SSH or SFTP between your desktop and the servers, which adds another step to the process. There are also issues when editing the files with a Windows utility; it inserts the carriage return character that has to be removed to run on a Unix box. The Unix utility that removes special Windows characters is called dos2unix.
Maintaining Oracle Standards

`dos2unix` removes any links (symbolic or hard) that a file currently has. This will affect how the file is referenced in the `$PATH`, so the links must be recreated. See the Unix help pages for `ln`, which is done by issuing a `man ln`. This command opens up the OS documentation for the `ln` command on Unix.

There is a way to eliminate the extra shuttling of code while keeping the GUI interface by using the open source ECLIPSE on the server. It is most closely associated with Java programming as an Independent Development Environment (IDE). There are additional plugins, software extensions, and code additions for almost any programming you would need to accomplish. Check out the Data Tools Platform Project for downloads specific to database development.

Version Control Software will take some time to learn to use safely. It can easily overwrite or remove critical pieces of code. So work closely with the code repository system admin to make sure there are backups! A safer way to use the repository might be to check it out in a temporary staging area and migrate the code safely to a protected code tree.

**Where, when, and who to call for help**

Okay, now that you know the major responsibilities as a DBA, *when* should you ask for help? When would you know something is wrong? One issue that every DBA dreads is when end users complain that the database is slow. It doesn't matter if you are sitting in your office and the database seems to be running perfectly well. Every script you run reveals no issues or distinguishable slowness. So what are those end users talking about? If the end users have an issue, then you now have a problem. And that particular problem is called response time; this subject will be explored further in Chapter 8.

Components (at several levels of the technology stack) that slow things down for the database can include the network, operating system, application servers, and of course hardware-specific problems. Start an SQL trace to help come up with an error message that will give more clues to the problem. Please don't assume it is someone else's problem.

If you encounter corruption at any level, automatically open a Service Request, even if you solve it completely yourself. Unless you know exactly how extensive the corruption is, it is best to take the database offline while it is being fixed because you may have to restore from previous backups, possibly losing the current transactions being entered.
The best defense against corruption is preventing it in the first place. 11g has a new startup initialization parameter `DB_ULTRA_SAFE`. Certain types of database corruption can also be prevented by implementing DATAGUARD for failing over when something disastrous occurs to the primary production database. In earlier versions of Oracle, the two database parameters `db_block_checksum` and/or `db_block_checking` were used to help in corruption detection and prevention.

Now is the time to point you to Oracle's website for the Maximum Availability Architecture (MAA). They have worked extensively with vendors and Very Large Database (VLDB) customers to improve performance at every level of the technology stack. Most of the concepts, techniques, and tips will apply to any company that runs Oracle. You need to download all of the white papers from the Oracle Technology website, print them out, and go over all of the details thoroughly. There is additional information available on MOS about MAA. Only by testing with your data can you be assured that true performance improvement or failover capabilities from these recommendations are suitable for your environment.

Critical tuning information presented in Chapter 8 will come from the practices outlined in the MAA white papers. The following URL comes from the Oracle Technology Network site, which contains extensive forums, white papers, online documentation, and software downloads for Oracle products.

http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

**My Oracle Support**

As a new DBA, I would recommend starting a search for a problem or issue on the My Oracle Support website. As you gain experience in researching issues that other DBAs have helped with, then you may decide to start with an Internet search first. On MOS there are database administration scripts, documentation, customer-written articles as well as complete checklists for any migration project. Look for FAQs on most subjects as they often have the complete list of all current documents that are related. The new **Upgrade Companion** found on MOS should be shown to your DBA manager so that they can see the complex process needed just to upgrade the database—it will help and scare you at the same time.

Double check whether the licenses listed on your MOS profile are correct; contact your Oracle account representative if they aren't.
If there is an error message involved, even if it isn't a specific Oracle message, MOS also has answers to errors related to the operating system, desktop, network, client browsers, compilers, and even common programming issues. The trick is learning to do advanced searching within MOS and to find the information you need. If you use an Internet software search site they will strip any non-alphanumerical characters, but not MOS. It will do exact, error code type of searches like "ORA-00600".

Check out the code download for this chapter. It contains a small plugin for Mozilla and information on downloading a Microsoft Deskbar, making searches on MOS easier. MOS_search_mozilla.txt and MOS_search_msdeskbar.txt each contain the code source and credit to the author. It will ask for your MOS username and password to complete the search.

Remember that if you have an emergency, MOS may take hours (if not several days) to respond, depending on the level of support purchased. There is a phone number to call MOS, but that will result in waiting until there is someone available to help you at the time you call. In the past, fairly good results came from telephoning Oracle support but these days you are summarily directed back to the MOS website to enter a Service Request. If you have system configurations uploaded to MOS (see Change Control, Release, and Configuration Management section) that makes it easier to enter a Service Request, as it automatically preloads a lot of the required information.

**Documentation library**

Of course, the first place you should start when you have a question on migrating to a new release is the documentation. Be sure to match the version of the documentation to the database version! Download the entire library found on the Oracle Technology Network website—http://otn.oracle.com. You will notice that even the downloaded local copy will have the search box returning to the Oracle website. There are occasions when the website is not available.

If you only want to search using desktop search indexing capabilities (see information in the preceding paragraphs), then open up the index.htm and comment out the following lines that reference the Oracle website. This will disable the feature completely and will show it has already commented out by adding ! in front and in the last line of the code just inside the > signs;

```html
<!--div class="simple_search_form_container" style="margin-bottom: 5px;">
  <div>
    <form class="simple_search_form" action="http://www.oracle.com/pls/db112/search" method="get">
      <input type="hidden" name="remark" value="quick_search" />
    </form>
  </div>
</div>
```

[64]
The Oracle community is world-wide and most people are willing to help even novice DBAs. One of the best ways to get a question answered is by knowing how to ask the question in the first place. See the following for a few pointers:

- Searching the forum archives first before asking a question—most often novices are asking a question someone else has asked.
- If forum searching is not successful then use your favorite Internet search engine. Be careful that the information found may not actually apply to your situation, as Oracle problems are time-sensitive and/or version specific.
- There is the old standby—Oracle documentation or searching MOS.
- Frequently asked questions or FAQ's on MOS (use both forms of the words FAQ to search with) will be more specific and up-to-date as compared to the documentation.
- Once you have some preliminary information, try to figure out the answer on your own in a test environment.
- Ask someone you know for the answer, but try not too overwhelm them with too many questions, they may stop responding.
The Oracle code (PL/SQL and executables) is wrapped, making the source non-viewable by any utility. Look within the Oracle-supplied DBA views for information. Search on the Internet for a free utility to learn these views. Descriptions for the columns for the data dictionary are included from the 
\texttt{dba\_col\_comments} view, but using a utility makes it easier to read.

- You should really make an effort to figure this out for yourself. Take an Oracle University class to get started. Every certified Oracle instructor that I have had the chance to meet has been excellent and the group as a whole comes highly recommended.

- Check out your local community college or university for Oracle classes. Taking any type of instruction will give you a pool of people to ask for help when you need it.

The following recommended URL list is not comprehensive. It is a jumping-off place to get you started. What it does is eliminate a lot of the bad websites that have incorrect information or are just strictly commercial:

\begin{itemize}
  \item http://asktom.oracle.com
  \item http://www.orafaq.com
  \item Oracle-L email lists, part of ORAFAQ
  \item http://www.freelists.org/archive/oracle-l
  \item http://www.eclipse.org
  \item Usenet Groups
  \item http://groups.google.com/groups/dir?sel=33583151&expand=1
  \item http://support.oracle.com
  \item http://forums.oracle.com
  \item OraNA :: Oracle News Aggregator
  \item http://orana.info
  \item Google Directories
  \item http://directory.google.com/Top/Computers/Software/Databases/Oracle/
  \item http://www.oracletips.info/
\end{itemize}
Summary

We looked at many things related to implementing standards across the enterprise in this chapter. Let’s take a moment to run through them.

Multiple ORACLE_HOME(s) and multiple databases on multiple servers can be kept under control by maintaining consistent standards across the enterprise. These standards include both OFA and the methods used to automate DBA tasks. Don’t let small mistakes keep you squirreled away in your office manually fixing problems that can be avoided in the first place.

Start off by writing your own scripts. Feel free to search on My Oracle Support or the Internet when you need some inspiration. But don’t just implement what someone else has written; change it slightly to make it yours. Keep working on scripting as and when you have time.

This chapter contained an introduction to all of the latest manageability features found in 11g Diagnosability Framework. Several of the newer features are an additional licensed option, which are available as a link on the OEM console, as a database package, or as a command-line utility.

Installing the Oracle Configuration Manager allows uploading configurations, which will open up an entire new world on the My Oracle Support website. Be aware that it does reveal information about your site to support personnel and to anyone else with the same CSI number.

Ask everyone when an issue can't be resolved quickly. Compare answers from different sources. Looking for similarity between the responses is one way to gauge credibility. Test several different scenarios before implementing the best solution. There are so many tools on the Internet to keep you connected with the large DBA community: user groups, Twitter, blogs, IT conferences, e-mail lists, and forums. Remember everyone is a novice at some point.

The next chapter takes us into the most detailed level of the database: the data block. You will see how to use special commands to dump the data in its hexadecimal format and convert back to a character. Understanding how data travels from one database structure to the next will give you a solid background of technical information to build on as a DBA. It will also give you more confidence in your troubleshooting abilities.