Introduction

"... The Oracle DBA just left the project. Congratulations, YOU are now in charge of maintaining our production Oracle database until we hire a new DBA ..."

Sounds familiar? The purpose of this paper is to discuss basic Oracle database administration skills to administer and maintain an existing Oracle database in support of PDMLink. The objective is not to teach the reader how to become an Oracle DBA, but to give the PDMLink administrator, or others in a similar situation, the fundamental and basic knowledge, tips, techniques, and resources to help perform the daily maintenance tasks needed to support an Oracle database. The topics that will be covered in this paper are:

- Typical daily activities of a DBA
- Logging into the Oracle database
- Starting and stopping the Oracle database
- Oracle database and configuration files
- Backing up the Oracle database
- GUI tools – Oracle Enterprise Manager Database Control
- Troubleshooting Oracle database
- Technical resources
Typical daily activities of a DBA

If you find yourself in a position of having to take care of one or more Oracle databases, your first question might be “what did the DBA who just left the project do every day?” If I had to answer this question, as a DBA supporting a PDMLink installation, I would say that a typical day consists of:

- Checking the nightly Oracle database backup jobs
- Checking the Oracle alert log file for errors
- Supporting PDMLink administrators, specifically, creating a new Oracle database that will be used to rehost a production system for testing and development

I have listed just three items in a typical day for me as an Oracle DBA supporting PDMLink; however, each of these items leads to many other questions to someone who is not an Oracle DBA and finds him/herself tasked with taking care of the Oracle system.

For example, the item listed above “Checking the Oracle alert log file for errors” leads immediately to the following questions: What is and where is the alert log file? What does an Oracle error look like? If I find an error, how do I log into the database as an administrator? How do I start up the database if the error says that the database is down? Where are the Oracle configuration files? All these topics, and others, will be covered in this paper.
Logging into the Oracle database

Since the assumption is that the Oracle DBA left, and that you, a PDMLink administrator, have been asked to take care of the Oracle databases until a new Oracle DBA is hired, we must assume that you will be granted access to the host machines where the Oracle databases are installed, and you will be given access to the operating system user account who owns the Oracle database installations.

Using PDMLink as an analogy, typically a system administrator creates local user on the host where PDMLink is installed who is the “owner” of the Windchill installation. This user is also granted specific privileges on the host. Similarly, a system administrator also creates local user who is the “owner” of the Oracle installation on the host machines where the Oracle databases are installed, and this user is also given specific privileges.

Let’s use the production Oracle database and host server as a starting point, since this is the most important database that you will be taking care of. You will be given (or must insist!) the Oracle account user name and password on the host server where the Oracle database resides. This is a MUST if you are being asked to take care of the Oracle databases. For the purposes of this paper, and for following examples, this is the configuration on a test production system:

- Host machine: ORAPROD
- Host user name and password: oracle/oracle123
- Database name: TEST

When you log in on the host machine “ORAPROD” as the OS user “oracle”, you can log in directly into the database with the highest database administrative privilege, using SQLPlus from a command prompt, as follows:

```
C:\Windows\System32>sqlplus / as sysdba
SQL*Plus: Release 11.2.0.4.0 Production on Tue Feb 24 16:35:08 2015
Copyright (c) 1982, 2013, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
SQL> SHOW USER
USER is "SYS"
SQL> EXIT
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 
C:\Windows\System32>
```
Ideally, so that you don’t have to change any passwords in the database, you should be given the current passwords for the database users SYS and SYSTEM. These are the users in the database with the highest privileges. It is beyond the scope of this paper to discuss in detail the differences between these users. But since you will be in charge of this database temporarily, you will perform most of the database administration activities as the database user SYS.

The command “sqlplus/as sysdba” allows you to log into the database as user SYS, as shown in the picture above, without having to supply SYS’s password. The argument “as sysdba” indicates that you are logging in with SYSDBA privilege into the database, which is the highest privilege that you can have in the database.

Once you have been given the passwords for the database users SYS and SYSTEM, this is how you can log into the database using these passwords:

A note about SQLPlus - searching on Google, you get this:


SQL*Plus is an interactive and batch query tool that is installed with every Oracle Database Server or Client installation. It has a command-line user interface, a Windows Graphical User Interface (GUI) and the iSQL*Plus web-based user interface.

The examples shown above are using the command-line user interface. Since you will be doing this job temporarily, I need to tell you that I have never used the GUI, and rarely have I used the
iTIP – How do you find the passwords for users SYS and SYSTEM if they are not documented? In the unlikely event that the outgoing Oracle DBA does not give you the passwords for users SYS or SYSTEM, which has happened to me before, the way that I have found these passwords was simple. It is almost certain that there are Windows scheduled task jobs (or cron jobs in the UNIX world) that run nightly to back up the Oracle database. Open the Task Scheduler on a Windows Server machine, or the crontab on a UNIX box, see which jobs are the ones that look like Oracle backup jobs (look for words such as RMAN, EXP, EXPDP, COLD BACKUP), and look at the scripts or programs that these scheduled task jobs, or cron jobs, run every night and read the contents of these scripts or programs carefully. Even though it is not a recommended practice, some DBAs create backup scripts where they have hard-coded the user names and passwords of either SYS or SYSTEM, and they have made these files not-readable by anyone else other than by the owner, which would be the OS “oracle” user. And we have already established that you have been given the password to log in as the OS “oracle” user. In the past I have been able to find out passwords for these, and other, users by looking in detail at the backup scripts. As mentioned earlier, hard coding passwords is not recommended, but some administrators do this for various reasons and they justify this practice by making these files not accessible to anybody else other than to the OS “oracle” user.
Starting up and shutting down an Oracle database

Besides the usual calls from the PDMLink administrators or from the users saying that they are getting error messages telling them that there is no access to the database, how do you check if the database is up and running, and how do you restart it if it is down?

There are two ways to quickly check if the database is up or down: 1) log into the database and look for a message from the database indicating that the database is down, or 2) check the status of the Oracle-related services on a MS Windows Server host, or if on a UNIX host check the processes running.

Shutting down a database

Let’s start by shutting down a running database. It’s very simple: log in as the user SYS and issue the command “shutdown immediate”, as follows:

```
Now that we know that we have a database that has been shut down, the following example will illustrate what you will see when you log into a database that has been shut down:

The phrase “Connected to an idle instance” indicates that the database is in a shutdown state.
Starting up a database

It is also simple to start up a database: log in as user **SYS** and issue the command “**startup**”, as shown in the following example:

![Database Startup Example](image_url)

Oracle services on a MS Windows Server

Another method to determine if the Oracle database is up or down is to check the status of the Oracle services in a MS Windows Server. In a typical Oracle installation on a MS Windows Server, you will see the following services:

![Oracle Services on Windows](image_url)

In this example, the database name is **TEST**, and the two services that you, as the Oracle DBA, are interested in are:

- **OracleOraDb11g_home2TNSListener**
- **OracleServiceTEST**
These are the two services that must be running (started) on a MS Windows server in order for the Oracle database to be accessible to client requests. The service `OracleServiceTEST` points to the memory structures allocated to the Oracle server on the host machine (the Oracle instance), while the service `OracleOraDb11g_home2TNSListener` points to the “LISTENER”, which is the process that manages and handles connection requests from clients, such as connection and query requests from PDMLink, to the database. Both of these processes must be running on the Oracle database host server in order for clients to access the database.

If any of these two processes shows a state other than “Started” (usually the Status column will be blank if the service has been stopped, as shown in the next example), then you must manually start the process by selecting it and clicking on the “Start” link, as shown in the next figure (in this example, the LISTENER service has been stopped and needs to be restarted):

![Oracle service running](image)

**Oracle processes on a UNIX server**

Below is a screen shot that shows the process running on a UNIX box when an Oracle database is running:

```
ps –ef | grep ora_ ; ps –ef | grep –i lsnr
```
The `ora_*` processes are the Oracle instance processes. The process listed at the bottom of the screen shot (…/bin/tnslsnr …) is the LISTENER process, as explained previously in the paper.
Oracle database and configuration files

At some point during the brief period of time when you will be in charge of maintaining one or more Oracle databases, you will ask yourself “well … where are the Oracle database files?, and which ones do I need to worry about?”

ORACLE_BASE and ORACLE_HOME

On the host server where the Oracle database is installed, you will find an environment variable called ORACLE_HOME:

On Windows: %oracle_home%
On UNIX: $ORACLE_HOME

This variable points to the folder/directory structure on the host server where the Oracle database executables reside. In a “typical” installation of an Oracle database, you could expect to see something like this for ORACLE_HOME:

On Windows: D:\oracle\product\11.2.0\dbhome_1
On UNIX: /app/oracle/product/11.2.0/dbhome_1

In the “dbhome_1” directory/folder is where you will find the Oracle database binaries and libraries. The important point to take away from this is that most of the Oracle documentation that you come across makes references to “ORACLE_HOME”, and most of the instructions that you will find when troubleshooting or when working with Oracle Support you will be expected to know where “ORACLE_HOME” is.

On a Windows server: echo %oracle_home%

On a UNIX box: echo $ORACLE_HOME
In some installations you may or may not see another environment variable defined, called `ORACLE_BASE`. `ORACLE_BASE` is the root folder/directory for all Oracle installations, and it contains diagnostic and trace/log files. You can find files such as the `alert.log` and `init.ora` parameter file. These files can be found under the `ADMIN` folder/directory if the database version is 10g, or `DIAG` folder/directory if the database version is 11g.

`ORACLE_BASE` is not a required environment variable, so you may not find it if you are looking for it. But in general, the `ORACLE_BASE` environment variable can be deduced from the `ORACLE_HOME` variable:

If `ORACLE_HOME = D:\oracle\product\11.2.0\dbhome_1`, then, in general, `ORACLE_BASE` will be the path before “\product\...”, in this case, `ORACLE_BASE=D:\oracle`

In `ORACLE_BASE`, the folders/directories that you should familiarize yourself with, especially when troubleshooting, are:

- `ORACLE_BASE/admin/<SID>/pfile`
  This folder contains one of the database’s initialization parameter files

- `ORACLE_BASE/diag/rdbms/<SID>/<SID>/trace`
  This folder contains the database’s trace/log files

**ALERT.LOG FILE**

The Oracle `alert.log` file is an important file that you must be familiar with. This is similar to a Method Server log file. This file contains entries of the many activities going on in the database. This file is the first file that you will check when troubleshooting an Oracle-related issue. If you have to work with Oracle Support, they will ask you for this file, so you need to know where it is.

This file is a plain-text file. There is only one such file in the database, and information gets appended to it through database shut downs and start ups. Note that, unlike a Method Server log file which gets created when Windchill stops or starts, this file will grown over time, since there is only one.

The `alert.log` file is located in `ORACLE_BASE/diag/rdbms/<SID>/<SID>/trace`. You can also find the location of this file by looking at the value of the database parameter “background_dump_dest”, as shown in the following example:
In the directory listed by the parameter “background_dump_dest”, you will find the file called alert_<database name>.log, along many trace files, which can be used for troubleshooting purposes when the need arises.

_TIP_ – Did you notice the line:

```
SQL> host dir C:\ORACLE\diag\rdbms\test\test\trace | find "alert"
```

When you are in a SQLPlus session, you can issue OS commands by typing the keyword “host”, followed by the OS command. You can also type the keyword “host” and hit return, which will take you to the OS prompt, where you can issue OS commands. Once you are done working in the OS prompt, you can type “exit” and it will return you to the SQLPlus prompt. Take a look at the following example, when you issue the command “host” at the SQL> prompt:
TNSNAMES.ORA FILE

The file called tnsnames.ora is one that you may or may not need to interact with; however, it will be useful to know what it is and where it is. First, where is it? You can find it in:

On Windows: %oracle_home%\network\admin
On UNIX: $ORACLE_HOME/network/admin

It is a plain text file, so you can read its contents. You can find the full technical definition in the Oracle documentation (or on Google!), but in a nut shell, this file contains the alias descriptors that point to one or more databases. In this file you will find one alias that maps to one database on a server, not necessarily the local host where you are working, but it can contain aliases to databases that reside on other networked hosts, such as development or testing hosts. The SQLPlus executable that is installed on the machine that you are currently logged in will read the contents of this file upon initialization; therefore, you can use your local SQLPlus installation to access databases that are not on your local host. Typically, the tnsnames.ora file is a local file
on each machine where there is Oracle software installed. Below is an example of what a tnsnames.ora file might look like:

```oracle
WIND-PROD =
<DESCRIPTION =
<ADDRESS_LIST =
<ADDRESS = (PROTOCOL = TCP)(SERVICE_NAME = wind)
</ADDRESS_LIST>
</CONNECT_DATA>
</DESCRIPTION>
</ENTRY>
</ENTRY>
</tnsnames.ora>
```

You see two alias entries: WIND-PROD and WIND-DEVL. The alias WIND-PROD points to a server host called ORA-PROD where there is an Oracle database called WIND, and the LISTENER process on that host is listening for database requests on port 1521.

The alias WIND-DEVL points to a server host called ORA-DEVL where there is an Oracle database, also called WIND, and the LISTENER process on that host is listening on port 1521.

Note that each host has its own Oracle installation and in each host there is a database called WIND. From this example one can see that one of the WIND databases is the production database on the PROD host, while the other WIND database is a development database on the DEVL host.

The tnsnames.ora file takes care of connecting you to the correct database on the correct host, depending on the alias that you use to connect to the database. Take a look at the following examples:
In this example you can see how the `tnsnames.ora` file connected the database user SYSTEM to the WIND database on the ORA-PROD server by using the alias “@wind-prod” in the SQLPlus connect string. Similarly, `tnsnames.ora` file resolved the “@wind-devl” alias to connect the database user SYSTEM to the WIND database on the ORA-DEVL server. Both of these connections were made from the same SQLPlus session on a client machine.

This is an important concept to understand, especially if you are managing multiple database installations, whether they are on one or multiple servers.

**INIT.ORA / SPFILE.ORA PARAMETER FILES**

Analogous to a Windchill installation where you have parameter files such as `site.xconf` and `wt.properties` files where the properties of the Windchill server are defined upon start up of the Windchill application, the Oracle database also has a parameter file that is read when the database instance is started up that defines properties for the Oracle instance and the database.

There are two types of parameter files: a plain text, parameter file called `INIT<SID>.ORA` file, and a binary file called `SPFILE<SID>.ORA`. These files contain parameters that define the properties of the database and of the instance upon start up. These files include properties such as: RAM memory allocation for the Oracle instance, location of the database CONTROL files (the database CONTROL files, usually in the form `CONTROL01.CTL`, are essential to the
database – they record the physical structure of the database; the database will not work without the CONTROL files), physical limits of the database, and many more.

You can find these parameter files in the following locations:

On UNIX:  
$ORACLE_HOME/dbs/init<SID>.ora  (plain text file)  
$ORACLE_HOME/dbs/spfile<SID>.ora  (binary file)

On Windows:  
$ORACLE_HOME\database\init<SID>.ora  
$ORACLE_HOME\database\spfile<SID>.ora

Note:  
In some cases, you will find the most recent, manually edited, plain-text initialization parameter file, in ORACLE_BASE/admin/<SID>/pfile/init<SID>.ora.

It is beyond the scope of this paper to discuss in detail the differences between the plain text init<SID>.ora file and the binary spfile<SID>.ora files. What YOU, a PDMLink administrator taking care of Oracle databases temporarily, need to know is that when you start up a database by issuing the command (from a SQLPrompt) “SQL> startup;” (notice no arguments after the word “startup”) the database instance will use the binary spfile<SID>.ora initialization file to get its properties from.

If you are instructed (by Oracle Support for example) to manually make changes to one or more initialization parameter files and use these new changes to start up the database, then you will be making changes to the plain text init<SID>.ora file, in which case you will start up the database by issuing the command:

```
SQL> startup pfile='c:\oracle\admin\<SID>\pfile\init<SID>.ora'
```

Notice the argument “pfile=…” after the keyword “startup”

An extremely important aspect about shutting down and starting up the Oracle database is to know what the standard procedure used by the former Oracle DBA was: did the former DBA use to start up the Oracle database routinely using the SPFILE (i.e. SQL> startup;), or did he routinely used to start up the database by always using a parameter file (i.e. SQL> startup pfile=’…/init<SID>.ora’). Since you are doing this job temporarily, there is no need to change procedures; therefore, you must find out how the Oracle database is routinely started up after being shutdown. The easiest way to find out is to look at the daily backup scripts, ideally one that does a COLD backup where the database has to be shutdown, and inspect the script and look for the piece of SQL code where the database is started up. Once you find it you will see whether it is the former or the latter.
Back up the Oracle database

One of the most important, if not THE most important, tasks that you, the PDMLink administrator, will have to perform on a daily basis while serving as the temporary Oracle DBA is to check the health of the backups of the Oracle databases that you are now in charge of. If you only have the time to do one thing as the interim Oracle DBA, make it to take care of backing up the production Oracle database.

It is impossible to know the backups that your former Oracle DBA has in place; however, I will describe the backups that we have implemented on a typical production Oracle database supporting a PDMLink installation. If your previous Oracle DBA didn’t leave you documentation describing the daily backups of the Oracle database, it is relatively simple to figure it out.

On a Windows server, it is likely that database backup jobs are being run through the Windows Task Scheduler, as shown on the figure below:

From the screen shot above you can see that there are jobs such as “rman backup”, “expdp”, “export”. The “rman backup” refers to the Oracle RMAN (Recovery Manager) hot backup. The “expdp” job refers to the Oracle Export Data Pump backup utility. The “export” job refers to the traditional Oracle Export backup utility. These different Oracle utilities will be explained below.

On a UNIX server, you can inspect the contents of the CRON TAB and search for Oracle-related cron jobs. As an example, you can run the following commands and inspect the output:

```
[oraprod ~]$ crontab -l | more
or ... 
[oraprod ~]$ crontab -l > crontab.out
```
If there is no specific documentation regarding your Oracle database backups, it will be your duty to look carefully at the Task Scheduler on Windows, or at the CRONTAB on UNIX, figure out which are the database backup jobs based on certain key words (such as “rman”, “exp”, “expdp”), track these scripts down, and look at their contents to figure out the folder/directory where the backup files are being written to, and then familiarize yourself with the output. We will attempt to describe, at a very high level, these utilities so that you know what to look for.

RMAN – Recovery Manager

This is a built-in Oracle database utility that allows you to do a hot (i.e no database shut down), online, backup of the database files. RMAN will back up the files on disk that make up the database (referred to as a ‘physical’ backup), as compared to the Export or Data Pump Export utilities which will back up the actual data within the database (referred to as a ‘logical’ backup). This utility is also used to restore files from backup and recover the database from a failure.

The contents of a typical RMAN backup script might look something like this:

```
%oracle_home%/bin/rman

RMAN utility executable

target '/ as sysdba'

you are logging in to the “test” database as user SYS (/ as sysdba)

nocatalog

you are not using an external metadata repository for RMAN operations

cmdfile D:\BackupScripts\rman\rman_backup.run

you are using another file called rman_backup.run which contains the actual RMAN commands
```
The output of the RMAN command will be recorded in a log file; **THIS is the log file that you need to inspect daily to verify the backup job!**

The **D:\BackupScripts\rman\rman_backup.log.temp** log file will be pretty large and will contain a lot of information pertaining to what was backed up and the result of the operation. Below we list a few of the lines in the log file that will be of interest to you:

```
log D:\BackupScripts\rman\rman_backup.log.temp

The output of the RMAN command will be recorded in a log file; **THIS is the log file that you need to inspect daily to verify the backup job!**

The **D:\BackupScripts\rman\rman_backup.log.temp** log file will be pretty large and will contain a lot of information pertaining to what was backed up and the result of the operation. Below we list a few of the lines in the log file that will be of interest to you:

```

```
```
TIP – It is important to note that these backup files (or backup pieces) are binary files that are created by the RMAN utility and can only be read by the RMAN utility. You cannot manually manipulate or use these files without using RMAN.

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Deleting the following obsolete backups and copies:

These lines in the log file show that only 1 copy of the backup files will be retained, and upon successful completion of the current RMAN job, the previous backup files will be deleted. This is the mechanism by which RMAN automatically cleans up older backup, obsolete, files, thereby keeping the backup destination folder/directory from filling up.

RMAN errors

What does an RMAN error look like in the log file? Below is one example of what you should be looking for in the output log file after an RMAN backup runs. If and when you encounter this, you will have to contact Oracle Support, or try to research the error using other resources:
As you can see, these messages stand out and are pretty clear to see. Usually you will see these messages at the bottom of the log file, since it indicates a problem with the backup and the RMAN backup job would have most likely ended or aborted at this point.

**Data Pump (EXPDP) export and traditional export (EXP)**

If you are a PDMLink administrator and you are reading this paper, then you are probably familiar with the re-hosting procedure in PDMLink. One of the steps in re-hosting a PDMLink installation is to copy the database table data from the source system to the target system. You accomplish this task by copying an Oracle export dump file (a .dmp file) containing the database tables from the source system to the target system, and then importing (IMP) these tables into the new system.
The Oracle export dump file created in the source system, such as the production database, can also be used as a backup for the Oracle database. The Oracle export dump file (*.dmp) is also referred to as a “logical” backup, because it contains the actual data from the database. This type of “logical” backup complements the “physical” backup explained in the previous section, done through Recover Manager (RMAN). By having both a “physical” backup through RMAN, and a “logical” backup through the Oracle export utility (either traditional export, EXP, or the newer Data Pump export, EXPDP), you will have the ability to restore and recover the database, or the Windchill schema, to a point in time before failure.

As mentioned in the previous section, if the former DBA did not leave behind any documentation pertaining to the database backups, it is now your job as the interim DBA to find the backup job that does the nightly export of the Oracle database, and understand what it is doing and, more importantly, what to look for in the output log file.

If you are on Windows, you will look at the Scheduled Task jobs, and if you are on UNIX, you will look through the CRONTAB. Hopefully the previous DBA gave the export backup job a name that will be obvious to you, with the words “exp” or “export” in the name of the job. If not, then the following example illustrates a sample script that will export the entire database:

```bash
## Traditional export utility ##
set oracle_home=C:\oracle\product\11.2.0\dbhome_1
set oracle_sid=test
%oracle_home%/bin/exp '/ as sysdba' file=/oraback/export/test.dmp log=/oraback/export/test.log full=y compress=y statistics=none direct=n

## New Data Pump export utility ##
set oracle_home=C:\oracle\product\11.2.0\dbhome_1
set oracle_sid=test
%oracle_home%/bin/expdp '/ as sysdba' full=y content=all job_name=expdp_test_1 parallel=8 filesize=3G directory=datapump_dir dumpfile=/oraback/export/expdp_test.dmp logfile=/oraback/export/expdp_test.log
```
There are many parameters that can be used in both the traditional export (EXP) utility and the newer Data Pump (EXPDP) utility. The description of these parameters is beyond the scope of this paper. Earlier in the paper we established that, assuming that these export backup jobs are working properly, you should not need to, or be required to, make any changes or modifications to these scripts.

The information that is important to you as the temporary Oracle DBA, are the portions highlighted in yellow: the output log files. These are the files that you will need to inspect, if not every day, relatively frequently to ensure that the export backup jobs are working correctly.

In the examples shown above you can see the output of the export backup job will be recorded in a plain text log file. If there is no documentation regarding the Oracle backups, then it is up to you to look at these scripts in detail in order to figure out where to find these log files.

The following examples illustrate good, clean, export backup jobs, using the traditional EXP and the newer EXPDP utilities. These are the messages that you are looking for at the end (or at the bottom) of the log files:

**Traditional export (EXP):**

```
[many lines of output above this]
. . exporting table MGMT_V_CLUSTER_MEMBER_LIST
. . exporting table MGMT_V_CLUSTER_RAC_POL_DETAIL
. . exporting table MGMT_V_RAC_ASSOC_MEMBER_LIST
. . exporting table MGMT_V_RAC_MEMBER_LIST
. about to export MGMT_VIEW's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting referential integrity constraints
. exporting stored procedures
. exporting operators
. exporting indextypes
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting user history table
. exporting default and system auditing options
. exporting statistics
```
Export terminated successfully without warnings.

Data Pump export (EXPDP):

[many lines of output above this]
. . exported "SYSTEM"."REPCAT$_USER_PARM_VALUES" 0 KB 0 rows
. . exported "SYSTEM"."SQLPLUS_PRODUCT_PROFILE" 0 KB 0 rows
Master table "SYS"."EXPDP_20150306_0200AM" successfully loaded/unloaded
************************************************************************
Dump file set for SYS.EXPDP_20150306_0200AM is: \oraback\export\expdp_test.dmp
Job "SYS"."EXPDP_TEST_1" successfully completed at Fri Mar 6 02:16:23 2015 elapsed 0 00:16:16

Export errors

What does a typical EXP or EXPDP error look like in the log file? Below is one example of what you should be looking for in the output log file after an export backup job completes. If and when you encounter this, you will have to contact Oracle Support, or try to research the error using other resources:

Traditional export (EXP):

[many lines of output above this]
. . exporting table FIX_ASSOC
EXP-00008: ORACLE error 1008 encountered
ORA-01008: not all variables bound
EXP-00056: ORACLE error 1013 encountered
ORA-01013: user requested cancel of current operation
EXP-00000: Export terminated unsuccessfully

Data Pump export (EXPDP):

[many lines of output above this]
. . exported "USER1"."TABL1_NOCOMPRESS" 797.9 MB 6749198 rows
. . exported "USER1"."TABL1_COMPRESS" 797.9 MB 6749198 rows
::: Export> help
ORA-39065: unexpected master process exception in SEND
ORA-39079: unable to enqueue message
RP,KUPC$C_1_20150309162839,MCP,KUPC$A_1_162840280190000,13,Y
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 86
ORA-06512: at "SYS.KUPCSQUE_INT", line 969
ORA-24033: no recipients for message
Job "SYS"."DPEXP_USER1" stopped due to fatal error at Mon Mar 9 16:31:05 2015 elapsed 0 00:02:25

As you can clearly see from the examples above, when an export job fails it will produce lines of output with “ORA-“ or “EXP-“ errors in it.
GUI tools – Oracle Enterprise Manager Database Control

In this paper we are going to discuss only one graphical user interface: Enterprise Manager Database Control (DBControl). This is a web-based tool that you can use to manage the Oracle database. This application is extremely powerful; it allows the Oracle DBA to perform hundreds of tasks, from basic user creation and password change, to cloning a database on another host. This tool is an effective mechanism to perform routine, simple, tasks without the need to know the corresponding SQL commands from the command prompt. We believe that a PDMLink administrator serving in a temporary capacity of performing routine daily database administration tasks will find this tool useful.

The screen shot below shows the main page of DBControl.
How do I know if DBControl is installed?

On Windows, you check the contents of the following file:

C:\Windows\System32> **type %oracle_home%\install\portlist.ini**

If the **portlist.ini** file is empty, then this is an indication that DBControl has not been installed yet. We will cover the installation of DBControl in the following pages. If DBControl is installed, you will see a message similar to the message shown in the example below (on UNIX).

On UNIX, you check the contents of the following file:

[oracle@oraprod install]$ **cat $ORACLE_HOME/install/readme.txt**

```
Enterprise Manager Database Control URL - (test) :  
https://oraprod.lanl.gov:1158/em
```

The **readme.txt** file shows you that DBControl is installed on this server, and it shows you the URL to access it:  https://oraprod.lanl.gov:1158/em
You will log into DBControl either as user SYSTEM (Normal) or as user SYS (SYSDBA from the “Connect As” drop down list).

If DBControl is installed, how do I check if it is running? How do I start it if it is not?

On UNIX, you can check if DBControl is running by typing the following command:

```
[oracle@oraprod install]$ $ORACLE_HOME/bin/emctl status dbconsole
```

Oracle Enterprise Manager 11g Database Control Release 11.2.0.4.0
Copyright (c) 1996, 2013 Oracle Corporation. All rights reserved.
https://oraprod.lanl.gov:1158/em/console/aboutApplication
Oracle Enterprise Manager 11g is running.

If DBControl is not running (the message will be obvious), you can start the DBControl process like this:

```
[oracle@oraprod install]$ $ORACLE_HOME/bin/emctl start dbconsole
```

The commands on Windows to check if DBControl is running and to start it if it is not running are the same as shown above. On Windows you use `%oracle_home%` instead of `$ORACLE_HOME`.

If DBControl is not installed, how do I install it?

You will encounter one of two situations: 1) DBControl has not been installed on the Oracle database server, or 2) DBControl is installed but you get error messages and you have to re-install it. Both of these cases are easily solved, as the installation and re-installation of DBControl is very easy.

You can install or re-install DBControl on Windows and UNIX by issuing the following command from a command prompt (remember, `%oracle_home%` on Windows, or `$ORACLE_HOME` on UNIX):

```
<oracle_home>/bin/emca --config dbcontrol db --repos [create | recreate]
```

You will use the parameter “create” if DBControl as not been installed, and you will use the parameter “recreate” if it has already been installed (as determined by inspecting either the portlist.ini or readme.txt files) and you need to re-create it.
Be aware that when you run this command, you will be asked to provide the passwords for the following users: **SYS, SYSMAN, and DBSNMP**. The database user **SYSMAN** owns the database tables that are created by EM DBControl. These database tables are the metadata repository for information pertaining to DBControl and the database it is monitoring. The database user **DBSNMP** is the database user account used by the Enterprise Manager Agent process running on the database server to monitor the database.

If you are in the unfortunate situation of not knowing the passwords for user **SYSMAN** or **DBSNMP**, due to lack of documentation, then you will have to change their passwords and assign them new passwords prior to installing, or reinstalling, DBControl. Be sure to consult with the application administrators and others in your team to ensure that changing the password for the database user **DBSNMP** will not have a negative impact on other parts of the system.

You change a database user’s password through SQLPlus like this, from a command prompt:

```sql
SQL> alter user sysman identified by abc123 ;
SQL> alter user dbsnmp identified by abc123 ;
```

You will also be asked to provide the LISTENER port number (the port where the LISTENER process is running on and listening for incoming connections). In a default database installation and network configuration, the default LISTENER port number is 1521. You can verify this information by looking at the contents of the `listener.ora` configuration file, located in `%oracle_home%/network/admin`:

```sql
C:\Windows\System32>C:\Windows\System32>type %oracle_home%/network/admin/listener.ora

# Listener.ora Network Configuration File: c:\oracle\product\11.2.0\dbhome_2\network\ad
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = c:\oracle\product\11.2.0\dbhome_2)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=ONLY:c:\oracle\product\11.2.0\dbhome_2\bin\oraclr11.dll")
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ianl.gov)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

ADR_BASE_LISTENER = c:\oracle
```

**UNCLASSIFIED**
Below is a screen shot from a Windows server when re-installing DBControl in a database where it has already been previously installed.
STARTED EMCA at Mar 30, 2015 3:05:18 PM
EM Configuration Assistant, Version 11.2.0.3.0 Production
Copyright (c) 2003, 2011, Oracle. All rights reserved.

Enter the following information:
Database SID: test
Listener port number: 1521
Listener ORACLE_HOME [c:\oracle\product\11.2.0\dbhome_2 ]:
Password for SYS user:
Password for DBSNMP:
Password for SYSMAN user:
Password for SYSMAN user: Email address for notifications (optional): jachang@lanl.gov
Outgoing Mail (SMTP) server for notifications (optional):

You have specified the following settings
Database ORACLE_HOME ...................... c:\oracle\product\11.2.0\dbhome_2
Local hostname .............................. .lanl.gov
Listener ORACLE_HOME ...................... c:\oracle\product\11.2.0\dbhome_2
Listener port number .......................... 1521
Database SID ................................. test
Email address for notifications .................. jachang@lanl.gov
Outgoing Mail (SMTP) server for notifications

-------- WARNING : While repository is dropped the database will be put in quiesce mode. -------

Do you wish to continue? [yes<y>/no<n>): Y
Mar 30, 2015 3:05:50 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at c:\oracle\cfytool\log\emca\test\emca_2015_03_30_03_05_50 EMLog_20150330_030550
INFO: Dropping the EM repository (this may take a while) ...
Mar 30, 2015 3:07:59 PM oracle.sysman.emcp.EmreposConfig invoke
INFO: Repository successfully dropped
Mar 30, 2015 3:07:59 PM oracle.sysman.emcp.EMreposConfig createRepository
INFO: Creating the EM repository (this may take a while) ...
Mar 30, 2015 3:11:20 PM oracle.sysman.emcp.EmreposConfig invoke
INFO: Repository successfully created
Mar 30, 2015 3:11:23 PM oracle.sysman.emcp.EMreposConfig uploadConfigDataToRepository
INFO: Uploading configuration data to EM repository (this may take a while) ...
Mar 30, 2015 3:12:08 PM oracle.sysman.emcp.EmreposConfig invoke
INFO: Uploaded configuration data successfully
Mar 30, 2015 3:12:15 PM oracle.sysman.emcp.util.DBCtrlUtil secureDBCConsole
INFO: Securing Database Control (this may take a while) ...
Mar 30, 2015 3:12:28 PM oracle.sysman.emcp.util.DBCtrlUtil secureDBCConsole
INFO: Database Control secured successfully
Mar 30, 2015 3:12:28 PM oracle.sysman.emcp.util.DBCtrlUtil startOms
INFO: Starting Database Control (this may take a while) ...
Mar 30, 2015 3:13:34 PM oracle.sysman.emcp.EMDBPostConfig performConfiguration
INFO: Database Control started successfully
Mar 30, 2015 3:13:36 PM oracle.sysman.emcp.EMDBPostConfig performConfiguration
INFO: >>>>>>>>>>>>>> The Database Control URL is https:// .lanl.gov:55000/em <<<<<<<<<
Mar 30, 2015 3:13:42 PM oracle.sysman.emcp.EMDBPostConfig invoke
WARNING: """""""""""""""""

Management Repository has been placed in secure mode wherein Enterprise Manager data will remain placed in the file: c:\oracle\product\11.2.0\dbhome_2\ .lanl.gov_test\sysman\ ed up as the encrypted data will become unusable if this file is lost.

Enterprise Manager configuration completed successfully
FINISHED EMCA at Mar 30, 2015 3:13:37 PM
Note:
When you run the command to install DBControl, you will see the following message:

WARNING : While repository is dropped the database will be put in quiesce mode.

Putting the database in quiesce mode will not allow regular users to connect to the
database. This is done typically when you must perform maintenance, usually as user
SYS. On a 11g database, this message can be ignored because the database will not be
put in quiesce mode. However, on a 10g database, it will be put in quiesce mode and
regular users will not be able to connect to the database while DBControl is being
installed.

Note the information at the bottom of the screen output. It will show you the URL that you will
use to access EM DBControl on this server:

You can also get the URL information to access the newly installed DBControl from the
%oracle_home%\install\readme.txt file:
On a Windows server, you will see a new **OracleDBConsole<SID>** service after the creation, or re-installation, of EM DBControl:

![Windows Server Services](image)

On UNIX, you can “grep” for “emagent”, as shown below:

```
LA-UR-15-23637
```

![Grep Command Output](image)
Troubleshooting the Oracle database

One of the very first things you need to do in order to get help with troubleshooting issues in the Oracle database is to get the Oracle Customer Support Identifier (CSI) number from the previous DBA, or from someone in your IT organization. The CSI number will allow you to create an account with Oracle Support and create tickets to get help with a problem.

Once you have the CSI number, you will go the Oracle Support web page and register with them:

support.oracle.com
Once you have established an account with Oracle Support, you can log in and do a number of things: create a service request (SR), i.e. a ticket, look for documentation, look for answers to questions, etc. Below is the home screen when you go to support.oracle.com:
There are two main sections in the home page that we believe will be useful to you as you serve in a temporary DBA capacity: the Service Request tab, and the Search toolbar.

The Service Request tab

This is the place where you will create a service request (SR), i.e. a ticket, in order to engage Oracle Support in helping you troubleshoot issues with the database. The steps to create a SR are pretty straightforward, but you will need to know some key information, such as the operating system (and version) of the host where the Oracle database is installed, the version of the Oracle database, whether it’s a 32-bit or 64-bit installation, and the general type of problem that you are having (i.e., is it related to performance, or is it related to RMAN errors, etc.). Below is a screen shot of this tab:
The Search toolbar

The Search toolbar is where we do most of our troubleshooting. When you enter key words in this toolbar, you will be presented with a large number of results, including official how-to documents, bug reports, community conversations associated with your keyword, and much more. Below is a screen shot of the search toolbar and the results you get when you enter the keyword “RMAN errors”:

![Search toolbar and results](image-url)
You notice a large number of results coming back from your search. As with any other search tool, the more specific that you are with your search key words, the more relevant the results you get will be.
Oracle Enterprise Manager Database Control (EM DBControl)

Another powerful tool that you will use to troubleshoot and fix issues with the database is the GUI tool that we have already discussed: EM DBControl. In the following set of screen shots, we will show DBControl’s home page, and we will go through some of the tabs available from the home page and what you can do from within each tab.

Log into DBControl:
This is the “Home” page, and in it you can see an “Alerts” area where the most recent issues with the database are being reported. In this example, you can see that there are two tablespaces reported as being full.
This is the “Availability” page, where you can inspect and manage the Recovery Manager (RMAN) backups. The “Backup Reports” hyperlink will show you the output logs from the most recent RMAN backup jobs. You can inspect these reports for potential problems. The “Manage Current Backups” hyperlink will take you to the page where you can delete RMAN backups, if, for example, that is what needs to be done in order to free up disk space.

Below is an example of the “Backup Reports” page, where you can see that the most recent RMAN back up job completed with errors, which needs investigating:
The next screen shot is the “Server” tab, where you have access to a variety of powerful database administration tasks, such as database storage, configuration, user management, and much more.

As mentioned before, it is beyond the scope of this paper to discuss in detail each of the functions available in this page. However, we hope that this brief discussion of EM DBControl will at least get you to the starting point if and when you need to engage Oracle Support.
Technical resources

Besides the Oracle Support (support.oracle.com) web site that we have already discussed, one of the most useful resources available to you is the Oracle Technology Network (http://www.oracle.com/technetwork/index.html). This web site is home to all of Oracle’s software downloads and documentation. You will need an account to log in; however, you can use the account that you previously created to access Oracle Support’s web site to log into this web site. Below is a screen shot of the home page:

When you follow the hyperlink to “Documentation & APIs”, you will get to the Oracle Database documentation web page. Some of the most useful documents that you will find are the “2 Day” documents, which include: “2 Day DBA”, “2 Day+Security Guide”, and “2 Day+Performance
Tuning Guide”. These documents are short and concise, and will provide you with a strong foundation of Oracle database administration. You can find these documents at the following url:

http://docs.oracle.com/cd/E11882_01/nav/portal_4.htm