

DBMS Naming Standards

Version: 1.0

Table of Contents

DOCUMENT CONTROL	3
1.1 Distribution List	3
1.2 Document History	3
1.3 Terminology	3
1.4 Document Location	4
EXECUTIVE SUMMARY	5
NEW SERVER BUILD – PRE-INSTALLATION TASKS	6
1.1 Oracle User IDs and Groups	6
1.2 Basic Directory Structures and Files	6
1.2.1 database vs instance	6
1.2.2 ORACLE_BASE and ORACLE_HOME	6
1.2.3 CRS_HOME and ASM	7
1.2.4 Unix oratab file	7
1.2.5 Oracle oraInventory Directory	7
1.3 Basic Naming Requirements	7
1.4 Database and Instance Names	7
1.5 Oracle Net	8
1.6 Listener Configuration	8
1.7 Domain Name	8
1.8 Database Files	9
1.8.1 Data Files Directory	9
1.8.2 Archive Log Directory	9
1.8.3 Control Files	9
1.8.4 Files Names	9
1.9 Instance Specific Files	10
1.10 DBA and Application Support Files	10
1.10.1 DBA Directory Types	10
1.10.2 File Extensions	11
1.11 Database Schema	12
1.11.1 Application Schema Owner	12
1.11.2 End User Names	12
1.11.3 Profiles	12
1.11.4 Roles	12
1.12 Database Objects	12
1.12.1 Name abbreviations	13
1.12.2 Tables and Views	13
1.12.3 Columns	13
1.12.4 Constraints	13
1.12.5 Indexes	14
1.12.6 Snapshots	14
1.12.7 Synonyms	14
1.12.8 Tablespaces	14
1.12.9 Executable Objects	15
1.12.10 Database Links	15
1.13 Documentation Standards	15
1.14 Scripts	15
APPENDIX A – SCRIPT FILE TEMPLATE	16

APPENDIX B – DATE AND TIME STANDARD.....	17
APPENDIX C – SAMPLE DIRECTORY STRUCTURE FOR UNIX	17

DOCUMENT CONTROL

The document control section describes the revision history and summary of the changes made to the document.

1.1 Distribution List

Name	Role	Representing

1.2 Document History

Version	Date	Who	Summary of Changes
1.0	May, 22 nd 2017	Ramiz Sarah	Initial Document

1.3 Terminology

Term	Definition
< >	Text enclosed in < > represent variables.
\$	The dollar sign before text indicates that this is a predefined variable
alias	An alternative name for an object
archive log	Log files associated with transactions (data changes) associated with a database; used for recovery operations or replication to standby database
control file	File used to record the physical structure of the database
database	Combination of physical files that contain logical objects; a database may be used by multiple instances, but each instance may only refer to one database; see also instance
database files	Physical space allocation; assigned to logical tablespaces
database name	Identifier used to differentiate groups of logically connected tables and other objects from one another; refers to a database (see also database)
dbname	See database name
global database name	Unique identifier for a database
instance	Background processes, logs and associated parameter files that maintains the connection to a database; a database may have multiple instances but each instance can only connect to one database; see also database
listener	Oracle Net process that provides initial connection to database from clients that are not connecting directly; configuration files may include the listener.ora and sqlnet.ora
net profile	A configuration used for connection information (sqlnet.ora)
net service name	The alias or assigned identifier for the connection information used by a client to connect to the instance; may be configured using tnsnames.ora file; see also service name
oratab	Oracle file containing information on the instance name, ORACLE_HOME location and whether or not an instance should be started automatically after a server reboot
profile	An object used to define security resource limitations; users are assigned to a profile

redo log	Records changes to the data; also called online redo logs; are written to archive logs if archivelog mode is set for the database; see also archive log
role	Used to associate a group of security permissions; roles are then assigned to users instead of assigning all permissions directly to each user
schema	A userid and the objects in the database (such as tables, views, nicknames, triggers, functions, and packages) owned by that userid
service name	The alias or assigned identifier for the connection information used by a client to connect to an instance; see also net service name
SID	Instance identifier; instance name is set with variable ORACLE_SID
snapshot	Collection of tables (sometimes including logs and indexes) that contain the results of a query of one or more tables or views usually located in a remote database
synonym	Alias for a database object
tablespace	Logical allocation of space in the database; made up of physical data files; database objects are assigned to tablespaces
userid	Login name associated with a session or schema

1.4 Document Location

The source of the document will be found under the [KEC ALLIANCE Documentation](#) section

EXECUTIVE SUMMARY

This document covers the required and recommended naming standards for Oracle DBMS related objects and files, including database name, instance name, tables, functions, scripts, etc. It also includes the required and recommended locations for files associated with an Oracle DBMS, including executables, configuration files, log files, database files, scripts and trace files.

Any exception to specifications in this document must be based on application specific requirements and not on individual preferences.

Where an application and/or 3rd party hosting has standards of their own that conflicts, every effort should be made to comply with these standards to the extent possible. A reference to the location of the application and/or 3rd party standards document should be included in the database documentation.

This document addresses only the Oracle DBMS. It is meant as a supplement to the Oracle documentation for those seeking additional clarification as to how the standard is used for an Oracle environment.

ORACLE BUILD

This section sets the standards that can help database administrator with the setup of a new Oracle server. The following section will detail guidelines, which should be followed to assure a successful and comprehensive build.

1.1 Oracle User IDs and Groups

Oracle executables will always be installed with the oracle id.

The dba group must be defined for all UNIX servers. The oracle userid will be created with the primary group of dba. The home directory for the oracle user on UNIX servers will be /u01/app/oracle/home; this is equivalent to \$ORACLE_BASE/home. If individual userids are assigned to each DBA, these must also have the primary group of dba.

On Windows servers, the oracle user will be assigned to the admin group.

1.2 Basic Directory Structures and Files

This section of the standards document provides information on the directory structures that must be in place for the Oracle executables, database-specific files and application-specific files.

1.2.1 database vs instance

A database may have multiple instances; however, each instance can only point to one database. The database name to be used for an instance is defined in the parameter file with the dbname value. The database name should be equivalent to the instance name for a single instance configuration. For RAC configurations, the same database name will be used for multiple instances. The instance names will consist of the database name followed by a number. Database names must not end with a numeric. This would cause confusion RAC instances.

Instance-specific files are placed under \$ORACLE_BASE/admin/<database name>. Refer to the section on Instance Specific Files for additional information.

Scripts and files that are used for all instances are placed under \$ORACLE_BASE/common/scripts.

Database files are placed under /u###/oradata/<database name> on UNIX servers. If /u### cannot be used, then /<dbname>/oradata/u### should be used. Refer to the section on Database Files for additional information.

1.2.2 ORACLE_BASE and ORACLE_HOME

The directory layout for Oracle-related files is based on two variables:

- ORACLE_BASE
- ORACLE_HOME

Only files directly related to the set of executables are located in the ORACLE_HOME. Any files that are shared by multiple sets of Oracle executables should be in a subdirectory off of ORACLE_BASE. This location is required in order to prevent deletion of common files when upgrading Oracle from one version to another or when removing any version of Oracle that is installed on a server.

On UNIX servers, the ORACLE_BASE should be set to /u01/app/oracle. On Windows servers, the ORACLE_BASE should be set to <drive>:\oracle.

ORACLE_HOME is the location of a set of Oracle executables (a specific version). The ORACLE_HOME should be set to \$ORACLE_BASE/product/<version> where version is at least the first three sections of the version installed in that directory. The following are examples for the ORACLE_HOME:

- /u01/app/oracle/product/11.2.0
- /u01/app/oracle/product/11.2.0.4
- /u01/app/oracle/product/12.1.0
- /u01/app/oracle/product/10.1.0.2

1.2.3 CRS_HOME and ASM

When the Cluster Ready Services (crs) must be installed for Oracle, a CRS_HOME directory must be created. For UNIX servers, the CRS_HOME must be /u01/crs/oracle/product/<version>. If using the Automatic Storage Management (ASM), then a separate ORACLE_HOME for the ASM instance will need to be created. The ORACLE_HOME for the ASM instance must be /u01/asm/oracle/product/<version>.

1.2.4 Unix oratab file

For UNIX servers, the oratab file should be placed in either /etc or /var/opt/oracle depending upon the Operating System. If possible, this should be open for write to the UNIX dba group.

An alternative is to place the oratab file in the \$ORACLE_BASE/common directory and create a link to it from /etc/oratab or /var/opt/oracle/oratab.

1.2.5 Oracle oraInventory Directory

The global inventory directory (oraInventory) should be \$ORACLE_BASE/oraInventory. In the /etc or /var/opt/oracle directory (depending upon the operating system), the oraInst.loc file should point to this directory. The oraInst.loc file should contain the following line:

```
inventory_loc=/u01/app/oracle/oraInventory
```

1.3 Basic Naming Requirements

All database objects must adhere to the following requirements:

- Must be longer than one character
- Should be meaningful
- Should be a maximum of 30 characters

If there are additional naming requirements developed by or for the business or a particular application, that standard should be referenced as part of the documentation for support of that application.

1.4 Database and Instance Names

All database and instance names must also adhere to the following minimum requirements:

- Must have a unique name
- Must contain at least one letter that reflects the purpose of the database (i.e., d for development, t for test, etc.)

The database and instance name should be the same for single instance databases.

The global database name is meant to be unique globally. Therefore, the database name should be associated with a domain. The typical global database name should be a combination of the database name with the domain name.

1.5 Oracle Net

There should be a \$ORACLE_BASE/network directory for the files associated with Oracle Net. This location is required in order to prevent deletion of common files when upgrading Oracle from one version to another or when removing any version of Oracle that is installed on a server.

This directory should contain the following subdirectories:

Subdirectory	Contents
admin	listener.ora, sqlnet.ora, tnsnames.ora links to these files should be created under \$ORACLE_HOME/network/admin
log	Oracle Net log files
scripts	scripts related to Oracle Net links to these files should be created under /usr/local/bin
trace	Oracle Net trace files

A services file should be created for each server. This services file should contain information on all the port assignments for that server. By listing the ports and the associated process, duplication and the resultant problems can be avoided. The location of the services file is operating system dependent:

- Unix: /etc/services
- OS/2: \tcip\etc\services
- Windows: \WINNT\system32\drivers\etc

1.6 Listener Configuration

Depending upon the requirements of the application, there may be one or multiple listeners. If at all possible, all listeners on a server should use one version of the Oracle executables; the newest version is recommended. If the default listener configuration is used, then no listener configuration (listener.ora file) is necessary.

Where it is necessary to have a different listener for each set of executables, the \$ORACLE_BASE/network directory should not be used for the location of the listener.ora. In this situation, a listener.ora should be under each \$ORACLE_HOME/network directory.

When using multiple listeners, the listener name should be meaningful. The listener name should be a name that is associated with an application or database. The recommendation is to include lsnr as the suffix in the listener name. An example is: finance_lsnr.

The net service name can be any alias and there can be multiple net service names defined in a tnsnames.ora file for any database instance.

The sqlnet.ora file is optional. The recommendation is to create a sqlnet.ora file instead of using the defaults.

1.7 Domain Name

NAMES.DEFAULT_DOMAIN sets the domain to be used as a default if no domain is specified. When this parameter is set, the default domain name is automatically appended to any unqualified net service name. For example, if the default domain is set to livingstonintl.com, the connect string CONNECT scott/pwd@sales searches tnsnames.ora for sale and appends the domain (livingstonintl.com) to the net service name if it finds a net service name of "sales". If the net service name includes the domain extension (e.g. sales.world or sales.livingstonintl.com), the default domain is not appended. It is recommended that this be set to the most used domain.

1.8 Database Files

Oracle database files can use an operating system filesystem or Oracle Automatic Storage Management (ASM) instance. If using an Oracle ASM instance, please refer to the latest Oracle manuals for additional information.

1.8.1 Data Files Directory

If files are created in standard operating system file systems, then they should be placed in directories created with the following pattern:

- UNIX Server - /u###/oradata/<dbname>

Where u### represents the letter u followed by three digits (i.e., u001, u101, u200, etc.) and dbname is the database name.

Alternatively, if the above pattern cannot be used, use the following:

/<dbname>/oradata/u###

- Windows servers - <drive>:\oradata\<dbname>

Where <drive> represents the letter associated with that drive and dbname represents the database name.

1.8.2 Archive Log Directory

Archivelog files are associated with an instance. Archivelog files should be placed in the /u###/oradata/<database name>/arch or /u###/arch/<database name> directory. If these cannot be used, then /<DATABASE NAME>/oradata/u###/arch can be used instead. Consider using a numeric subset for the archive log files (e.g., /u900 through /u999). The following are examples of archive log directories, where "prdorcl" is the SID:

- /u900/oradata/prdorcl/arch
- /u095/oradata/prdorcl/arch
- /u095/arch/prdorcl/

1.8.3 Control Files

Control files are associated with an instance. As long as the control file includes the instance name, it may be placed in the same location as the data files (refer to Data Files Directory). An alternative is to place them in the \$ORACLE_BASE/admin/<database name>/control directory (refer to Instance Specific Files).

1.8.4 Files Names

The database file name should adhere to the following pattern:

File Type	Pattern	Examples
archive logs	<text>%s(sequence)%t(thread)%r(resetlogsID).arc	orcl%s%t%r.arc
control files	control<number>.ctl or ctl<SID><number>.ctl	control01.ctl
data file	<tablespace name><number>.dbf	users01.dbf
temporary	temp<number>.dbf	temp01.dbf
redo logs	redo<number><letter>.log	redo01a.log redo01b.log
system file	system<number>.dbf	system01.dbf

1.9 Instance Specific Files

Each instance should have a directory named with the following pattern \$ORACLE_BASE/admin/diag/rdbms<database name>. The table below lists the subdirectories which may be created under this directory.

Subdirectory	Contents
adump	audit trail files (excluding the files ending with .aud that are automatically created in \$ORACLE_HOME/rdbms/audit at startup and shutdown of the instance)
bdump	background dump destination
cdump	core file dumps related to an instance
control	optional location for the control files (refer to Control Files for more information)
create	programs used to create the database
exp	optional location for database export control files, scripts and output; this may be a link to a separate filesystem used for backups
pfile	parameter files associated with instance, including init.ora, spfile.ora and orapwd files; create links to the initialization files for the instance to these from the \$ORACLE_HOME/dbs directory For the init<SID>.ora, an alternative to having a link is to place an ifile reference in the init<SID>.ora file that points to a file (init.ora or spfile) in pfile.
scripts	scripts specific to the instance
udump	user dump destination

1.10 DBA and Application Support Files

If the Oracle DBA requires a directory for scripts and other files, this should be created either in the individual user home directory or in a subdirectory under \$ORACLE_BASE/<username> (where username is the individual login or name of the DBA).

Application support files that are specifically for an application should be placed either in the application-required directory or in \$ORACLE_BASE/<application> (where application is the name associated with the application). If different files are associated with various instances, then they should be placed in \$ORACLE_BASE/<application>/<database name> (where application is the name associated with the application).

1.10.1 DBA Directory Types

The following directories should be created (as needed):

Type	Directory	Contents
DBA	\$ORACLE_BASE/<dba username>	scripts and files for the DBA
application	\$ORACLE_BASE/<application name>	scripts and files associated with the application
database	\$ORACLE_BASE/common	scripts and files associated with all instances associated with the database (other than data files)
instance	\$ORACLE_BASE/admin/<database name>	Refer to Instance Specific Files

The following table lists the names for the subdirectories with the type of contents for each. Not all the following subdirectories need to be created under each directory. Only those that are needed should be created. For example, if there are no executable files for an application, then there would be no need to have a \$ORACLE_BASE/<application>/bin subdirectory.

Type	Contents
bin	executable programs
data	shared input and shared parameter files for scripts
ddl	data definition language; scripts used to create objects and schemas
doc	site or database documentation or instructions
ftp	location for ftp parameter files
output	location of files which are created from scripts or scheduled jobs
sched	scheduling or job information
scripts	shell or program scripts which are not more appropriately placed in any of the previously defined directories
sql	sql scripts which are not more appropriately placed in any of the previously defined directories
src	source code which is not placed in an application specific directory or under \$ORACLE_HOME

1.10.2 File Extensions

The following table provides a list of standard file extensions (not all inclusive):

Extension	Contents
.arc	archive logs
.bak	backup of a file
.bat	batch script (for older versions of Windows)
.cfg	configuration file referenced by a program or script
.cmd	batch script (for newer versions of Windows)
.ctl	control file
.dat	input/output data files used by programs and scripts
.dbf	database file
.dll	executable function called by a program
.exe	binary executable files (Windows platforms)
.htm	HTML file (web viewer)
.inc	include file referenced by a script
.job	job files (Windows Scheduled Task)
.ksh	korn shell script
.log	database log files
.lst	default extension on sql output files
.ora	Initialization files (listener.ora, init<SID>.ora, etc.)
.out	summary of script execution
.pip	named pipe
.pl	perl script
.prm	parameter file
.rpt	report file
.sh	shell script
.sql	sql script
.trc	trace file
.txt	text document (notepad) or sql output (can be used instead of .lst to make it easier to read in text editor)
no extension	binary executable on UNIX

1.11 Database Schema

This section covers the minimum requirements for the application owner and end users. The naming standards for the application schema owner and objects are not applicable for commercial off-the-shelf applications.

1.11.1 Application Schema Owner

The SYS and SYSTEM accounts should never be used as the application object owner. The application owner should be an account used only for the application objects and should not be used as the login for users (unless required by the application).

The schema owner should be a meaningful name. In many cases, the schema name to be used is part of the application requirements. The following are the basic ORACLE requirements:

- Limited to thirty characters
- Must be unique within the database
- Cannot be the same as a default oracle schema (e.g., perfstat, mdsys)

Application schemas must be granted privileges in accordance with the Oracle Security Baseline. In no case should any application schema be granted the Oracle DBA role.

1.11.2 End User Names

In most cases, Infrastructures will have a required pattern for the identifier associated with each employee or contractor. The identifier should be used for the end user names whenever possible. Avoid using actual names for database user accounts. NEVER use an employee name or userid for an application schema.

1.11.3 Profiles

Profiles should be assigned meaningful names that indicate the type of user privileges associated with the users. Typically one profile is created for all end users, another for the application schema owner and a third for users with high level privileges. The sys and system users should always have the default profile.

1.11.4 Roles

Roles should be created to meet the Oracle Security Baseline standards. Roles should be assigned meaningful names that indicate the type of privileges associated with that role. For example, a hr_read_only role could be created for users with only select privileges on the human resource tables.

1.12 Database Objects

This section covers the minimum requirements for the application objects which are developed in-house. The naming standards for the application schema owner and objects are not applicable for commercial off-the-shelf applications which typically require specific object names.

All objects must adhere to the following naming requirements:

- Do not use reserved words
- Names must be more than one character
- Names cannot exceed 30 characters
- Names should be meaningful

If it is necessary to reduce the size of the column name, consider removing vowels and duplicate letters.

Reserved words should always be checked when naming database objects. Since the reserved words will vary, an underscore in the name can be used to ensure that no reserved words are used. Even if development is being done with an older version of Oracle, always include as reserved words those for the most recent version of Oracle. Including the most recent version will enable easier upgrades for the database.

1.12.1 Name abbreviations

Name abbreviations should follow the application's team standards. If no application team standard is available, the recommendation is to use an agreed upon source, such as an acronym/abbreviation web site (e.g., the Word search function on the [Acronyma](#)).

1.12.2 Tables and Views

The following requirements should be followed for in-house developed applications:

- Singular not plural (e.g., employee instead of employees)
- Use a suffix or prefix to indicate the type of data (e.g., hst or hist for historical data)
- Consider including a prefix or suffix of v or vw for viewsddf

1.12.3 Columns

The following are recommended naming conventions for columns:

- Same name should not be used in other tables where the column data type or domain is different (e.g., do not use app_time for a varchar2 column in one table and for a date column in another table)
- Columns generated using a sequence should contain seq as the prefix or suffix and the name of the sequence used
- Foreign key columns should have the same name as the key column name
- Parent and child columns should consider including the word parent or child (or an appropriate abbreviation) in the column name

1.12.4 Constraints

Constraints (with the exception of not null constraints) should not use system-generated names. Constraints that have associated indexes should have the same name for both the constraint and index. If the constraint is associated with a single column, consider using the column name in the constraint name. For primary key constraints, consider using the table name in the constraint/index name.

Constraints should include a prefix or suffix to indicate the type of constraint. The following table lists the constraint types supported by Oracle.

Type of Constraint	Abbreviation	Definition
Alternate Key	ak	Can be used to uniquely identify each row in a table; but is not the primary key.
Check	ck	Performs verification on the value in a column
Foreign Key	fk	Ensure that a value in a column or group of columns exists in a parent table.
Not Null	nn	Ensures a column contains a value for every row inserted.
Primary Key	pk	Unique identifier for each row in a table; one primary key per table
Unique Key	uk or uq	Ensures that a value for a column or group of columns only appears once in the table.

1.12.5 Indexes

The following are recommended when naming indexes:

- if associated with a constraint, the name should be the same
- include the column or table name in the index name
- include a prefix or suffix to indicate the type of index
 - refer to the Constraint Type Table
 - ix for index
 - bm for bitmapped
 - fn for function-based indexes
 - pt for partition key
 - gl for global partition index
 - lc for local index

1.12.6 Snapshots

The recommendation for snapshots is to use sn as the prefix or suffix of the name.

1.12.7 Synonyms

In many cases, synonyms are required to be the same as the object name in order to point the application code to the objects owned by the application schema. If this is not required, then the following recommendations should be considered when creating synonyms:

- use sy as the prefix or suffix
- use a variation of the table name in the synonym name
- if there are multiple tables in different schemas with the same name, consider including the schema name as part of the synonym name

1.12.8 Tablespaces

Tablespace names should have meaningful names to indicate the purpose of the tablespace or application. The following are recommended for all databases:

- create separate tablespaces for the following:
 - rbs for rollback segments
 - sysaux is required (10g and higher versions)
 - system is required for the sys objects
 - temp tablespace for temporary segments or tempfiles
 - undo for undo tablespaces
 - users as the default tablespace for end users (not for application schema)
- Index tablespace names should enable the DBA to identify the associated application schema and include a prefix or suffix to indicate that these are for indexes (e.g., finance_idx)
- Data tablespaces names should enable the DBA to identify the associated application schema and include a prefix or suffix to indicate this is for data (e.g., finance_dat)
- Tablespaces with a LOB should consider using l as the prefix or suffix in the name (e.g., emp_pict_l)
- When installing Oracle options, a separate tablespace is recommended for each (e.g. wksys for ultrasearch)

1.12.9 Executable Objects

Including a suffix or prefix to identify the type of object is recommended, as follows:

- pkg for package
- prc or sp for procedure
- fct or fn for function
- trg or tr for trigger

For database triggers associated with a table, the recommendation is to include the table name in the name of the trigger (e.g., emp_data_trg for a trigger associated with the emp_data table).

1.12.10 Database Links

Database links must conform to the Oracle Security Baseline standards. Since database links can create a security vulnerability, it is important that they be created under a separate userid with limited privileges.

1.13 Documentation Standards

Livingston [Know-net](#) is the central repository for information on the servers, databases, and instances. It provides information on problems, changes, and metrics for the databases. It is also the central Oracle repository for database specific documentation.

For documentation that is specific to an instance, the instance name should be included in the document name. For documentation that is database-specific, the database name should be included in the document name. For documentation that is application-specific, the application name should be included in the document name.

1.14 Scripts

Scripts may be written in Korn shell, Perl, C shell, or Bourne shell. Script names must reflect the actions performed by the script. If a word separator is not used, then a capital letter can be used to make the name more readable. For UNIX environments, which are case sensitive, a separator is preferred. For example, a script used to perform cleanup actions should be given a name such as cln_up.sh (preferred for UNIX servers) or clnUp.sh.

Generic scripts that are applicable to all instances/databases on a server; but, are not specific to an application, should be placed in the \$ORACLE_BASE/common/scripts directory. Scripts applicable to only one instance should be in \$ORACLE_BASE/admin/<database name>/scripts. Scripts applicable to a specific application should be in \$ORACLE_BASE/<application>/scripts.

Scripts should include the full path name and file name for all output scripts. Where multiple copies of the output file are to be kept, a date-time variable should be included in the name of the output file. Where multiple copies are maintained, a process should be in place to periodically remove files in order to prevent the filesystem from filling up and causing an error.

Any parameter files referenced in a script should include the full path and file name.

Some systems use special code in scripts to create unique job steps. For those scripts each job step must be documented with one or more lines that describe exactly what the job step is doing. Additionally, the job script must echo to a log file the time, date, and job step that is executing.

The systems that do not use special codes to create unique job steps must still break the script into logical units of work that are documented.

All job scripts must be completely documented so that other DBAs can easily interpret the purpose and logic involved in the process. When documenting job scripts, a concise description must be given. Documentation must be detailed enough to understand the process. The recommended template is in Script File Template in the Appendices.

APPENDIX A – SCRIPT FILE TEMPLATE

```
#####
# SCRIPT                : <name>                                     #
#                                                                #
# DESCRIPTION           : <High Level Description >                #
#                                                                #
# INPUT FILES           : <path, name and describe all input files> #
#                                                                #
# OUTPUT FILES          : <path, name and describe all output files> #
#                                                                #
# CALLED OBJECT         : <path, name and type (function, stored    #
#                        procedure, module ...)                    #
#                                                                #
# CALLED BY             : <path, name and type (trigger, procedure, etc.) #
#                        of code that calls this                  #
#                                                                #
# TABLE EFFECTED       : <name and how (can use SIUD for Select, insert, #
#                        update and/or delete)                    #
#                                                                #
# SETUP INSTRUCTIONS    : <Explain dependencies for executing script> #
#                                                                #
#####
# COMMAND LINE EXECUTION:                                         #
# Script_name           input parameters (flags)                  #
# -----               #
# <script name>    parm1 parm2 etc showing optional parms in []  #
#                                                                #
# parm record layout:                                           #
# Parm 1 =                                                       #
# Parm 2 =                                                       #
#                                                                #
# EXAMPLE: <script name> <actual parms>                          #
#####
# RESTART INSTRUCTIONS: <if the script dies for any reason what  #
#                        steps must someone take to run it again> #
#                                                                #
# SPECIAL CONSIDERATIONS: <Is there anything special that someone #
#                        supporting the script should know>       #
#                                                                #
#####
# Modification Log:                                              #
# Date              Name              Description                  #
#####
#                                                                #
# CCYY_MM_DD  <Author Name>    Initial Creation                  #
# CCYY_MM_DD  <Changers Name>  <Description of change, change number, #
#                               problem resolved, etc. >          #
#####
```

APPENDIX B – DATE AND TIME STANDARD

Scripts backups, reports, and log files should have a date/time stamp amended to them. The “yymmdd_hhmmss” layout should be used. Use numbers NOT characters for month’s names, e.g:

20170525_143055

Use the following command to reproduce the layout in UNIX when an underscore is used between the date and time:

```
`date "+%Y%m%d_%H%M%S"``
```

Use the following code you to reproduce the layout in UNIX when a period is used between the date and time:

```
`date "+%Y%m%d%.H%M%S"``
```

APPENDIX C – SAMPLE DIRECTORY STRUCTURE FOR UNIX

This is an example of the most commonly configured directories. For this example the database name is “PRDORCL”, the instance name is “PRDORCL1”, the application is “FINANCE”, and the DBA has an individual UNIX id of “Scott”.

Directory	Contents
/u01/app/oracle	ORACLE_BASE
/u01/app/oracle/home	home directory for oracle UNIX id (\$HOME)
/u01/app/oracle/orainventory	oracle inventory directory
/u01/app/oracle/network	Oracle Net files
/u01/app/oracle/network/admin	listener.ora, tnsnames.ora, sqlnet.ora
/u01/app/oracle/network/log	network log files
/u01/app/oracle/network/trace	network trace files
/u01/app/oracle/product/11.2.0	ORACLE_HOME
/u01/app/oracle/scott	files owned by scott (the DBA)
/u01/app/oracle/admin/PRDORCL	files associated with PRDORCL1 instance
/u01/app/oracle/admin/PRDORCL/adump	PRDORCL1 audit trail files
/u01/app/oracle/admin/PRDORCL/bdump	PRDORCL1 background dump destination
/u01/app/oracle/admin/PRDORCL/cdump	PRDORCL1 core dump destination
/u01/app/oracle/admin/PRDORCL/control	PRDORCL1 control file, copy one
/u01/app/oracle/admin/PRDORCL/pfile	PRDORCL1 parameter files
/u01/app/oracle/admin/PRDORCL/scripts	PRDORCL1 instance-specific scripts
/u01/app/oracle/admin/PRDORCL/udump	PRDORCL1 user dump destination
/u01/app/oracle/common/scripts	common database scripts
/u001/oradata/PRDORCL	database files for PRDORCL database
/u001/oradata/PRDORCL	second copy of controlfile
/u900/oradata/PRDORCL/arch	archive files for PRDORCL1 instance
/u002/app/oracle/FINANCE	Application (FINANCE) files