

# **Exadime DBA Scripts**

**Version 1.0.0**

**Oracle**

---

---

## Revision History

---

Date	Version	Description	Author	Reviewer
Apr 12, 2016	1.0.0	Draft	EXADIME LLC	
Jul 06, 2017	1.0.0	Troubleshooting	EXADIME LLC	
Aug 08, 2018	1.0.0	Installation – UNIX	EXADIME LLC	
Mar 25, 2019	1.0.0	Software	EXADIME LLC	
Apr 01, 2019	1.0.0	Libraries	EXADIME LLC	
May 16, 2019	1.0.0	Golden Gate Parameters	EXADIME LLC	
May 04, 2020	1.0.0	Monitoring Scripts	EXADIME LLC	
Sep 03, 2020	1.0.0	Sample Cron Entry	EXADIME LLC	
Jan 13, 2020	1.0.0	Data Pump Export	EXADIME LLC	
Jan 15, 2020	1.0.0	Golden Gate Monitor	EXADIME LLC	
Feb 01, 2021	1.0.0	Removed title.ini and esp.ini	EXADIME LLC	
Feb 02, 2021	1.0.0	Host Ping Monitoring	EXADIME LLC	
Feb 07, 2021	1.0.0	Data Guard Monitoring	EXADIME LLC	
Feb 11, 2021	1.0.0	Directory Navigation	EXADIME LLC	
Apr 30, 2021	1.0.0	SQL Scripts	EXADIME LLC	
May 18, 2022	1.0.0	Installation & Navigation	EXADIME LLC	
Sep 12, 2023	1.0.0	TDE Wallet Backup	EXADIME LLC	
Nov 09, 2023	1.0.0	ASH Files	EXADIME LLC	
Dec 18, 2023	1.0.0	Java Files	EXADIME LLC	
May 20, 2024	1.0.0	Health Check	EXADIME LLC	

# Contents

---

<b>INTRODUCTION .....</b>	<b>5</b>
<b>INSTALLATION .....</b>	<b>5</b>
UNIX .....	5
LINUX .....	6
WINDOWS .....	6
<b>CONFIGURATION .....</b>	<b>7</b>
1. ORACLE FILES AND DIRECTORIES .....	7
2. TITLE .....	7
3. NOTIFICATIONS .....	7
4. DEFAULTS .....	8
7. AUTO START AND STOP .....	10
<b>GETTING STARTED.....</b>	<b>11</b>
DATABASE ENVIRONMENT .....	11
DIRECTORY NAVIGATION.....	12
<b>SETUP .....</b>	<b>13</b>
1. DATABASE.....	13
1.1. <i>stp_cr_db.ksh</i> .....	13
1.2. <i>stp_rm_db.ksh</i> .....	14
<b>MAINTENANCE .....</b>	<b>15</b>
1. ENV SETTINGS .....	15
1.1 <i>mnt_sw_db.ksh</i> .....	15
2. LISTENER.....	16
2.1. <i>mnt_st_lis.ksh</i> .....	16
2.2. <i>mnt_sp_lis.ksh</i> .....	16
3. INSTANCE .....	17
3.1. <i>mnt_st_db.ksh</i> .....	17
3.2. <i>mnt_sp_db.ksh</i> .....	17
4. AGENT.....	18
4.1. <i>mnt_st_agt.ksh</i> .....	18
4.2 <i>mnt_sp_agt.ksh</i> .....	18
5. OSWATCHER.....	19
5.1. <i>mnt_st_osw.ksh</i> .....	19
5.2. <i>mnt_sp_osw.ksh</i> .....	19
6. BACKUPS.....	20
6.1. <i>mnt_exp_dmp.ksh</i> .....	20
6.2. <i>mnt_imp_dmp.ksh</i> .....	20
6.3. <i>mnt_col_bkp.ksh</i> .....	21
6.4. <i>mnt_hot_bkp.ksh</i> .....	22
6.5. <i>mnt_tts_bkp.ksh</i> .....	22
6.6. <i>mnt_ctl_bkp.ksh</i> .....	23
6.7. <i>mnt_dp_exp.ksh</i> .....	24
6.8. <i>tde_wall_bkp.ksh</i> .....	24
7. HOUSEKEEPING .....	25

7.1. mnt_exa_hkf.ksh .....	25
7.2. mnt_ora_hkf.ksh .....	25
7.3. mnt_arc_hkf.ksh.....	26
7.4. mnt_aud_hkf.ksh.....	26
7.5. mnt_old_hkf.ksh.....	27
7.6. mnt_asm_aud.ksh .....	28
8. PERFORMANCE .....	28
8.1. mnt_anz_db.ksh .....	28
<b>MONITORING .....</b>	<b>29</b>
1. STATUS.....	29
1.1. mtr_ins_sts.ksh .....	29
1.3. mtr_host_ping.ksh .....	30
2. SPACE.....	30
2.1. mtr_tbs_spa.ksh.....	30
2.2. mtr_asm_disk.ksh .....	31
2.3. mtr_host_disk.ksh .....	32
3. LOG .....	32
3.1. mtr_alt_log.ksh.....	32
3.2. mtr_sys_log.ksh .....	33
4. REPLICATION.....	33
4.1. mtr_dg_gap.ksh .....	33
4.2. mtr_gg_lag.ksh .....	34
<b>REPORTS .....</b>	<b>35</b>
1. DAR_MEM_PAD.KSH .....	35
<b>SUPPORT .....</b>	<b>35</b>
1. RUN_CRN_JOB.KSH.....	35
2. RUN_SQL_FILE.KSH .....	36
3. RUN_SHE_FILE.KSH .....	36
4. RUN_AUT_SSH.KSH.....	37
<b>RECOVERY .....</b>	<b>37</b>
1. RCO_MNG_FILE.KSH .....	37
2. RCO_SYN_STB.KSH .....	38
<b>MISCELLANEOUS .....</b>	<b>39</b>
1. SHO_FILE_CNT.KSH .....	39
2. SHO_FILE_SUM.KSH.....	39
3. SHO_DISK_SUM.KSH .....	39
4. SHO_CRC_SUM.KSH .....	40
5. SHO_HOST_DISK.KSH.....	40
6. SHO_PAT_CON.KSH.....	41
7. SHO_AIX_NODE.KSH .....	41
<b>TROUBLESHOOTING .....</b>	<b>42</b>
PROBLEM #1.....	42
<b>APPENDIX .....</b>	<b>42</b>
1. INTERPRETATIONS .....	42
2. FINE PRINT .....	42

## Introduction

This document is a manual source for the usage of Exadime DBA Scripts for Oracle databases maintenance, monitoring, reporting, support and recovery.

## Installation

### UNIX

1. Make the base directory for Exadime scripts in ~oracle/scripts.

```
$mkdir -p ~oracle/scripts/exa
```

2. Transfer Exadime scripts into the above directory.

S. No.	Transfer Mode	Directory/File
1	Binary	doc lib plb unix
2	Text/ASCII	lib/exa.lib
3	Text/ASCII	unix/Sora; mnt_sw_db.ksh; run_crn_job.ksh
4	Text/ASCII	All other folders.

3. Set RWX permission.

```
chmod -R 750 ~oracle/scripts/exa
```

4. Specify Exadime scripts path in the following files if required.

S. No.	File Name	Directory	Comments
1	exa.ini	\${exa_home}/ini	Initialization parameters.
2	unix.ini	\${exa_home}/ini	Unix scripts initialization.
3	oracle.ini	\${exa_home}/ini	Oracle scripts initialization.
4	exa.env	\${exa_home}/env	Environment setup.
5	run_crn_job.ksh	\${exa_home}/unix	Cron job execution.
6	Sora	\${exa_home}/unix	Database auto start/stop.

**Note:** Default scripts path is “~oracle/scripts/exa” and ORACLE\_BASE is “~oracle”.

5. Lib folder contains Korn Shell libraries for each O/S. Use the appropriate KSH file.

```
$cd lib  
$cp <ksh_os> ksh  
$chmod 750 ksh
```

6. Add the following line into .bash\_profile (Linux)/.profile (AIX/Solaris).

```
. ~oracle/scripts/exa/env/exa.env
```

## **Linux**

In case of LINUX, install the following rpm.

```
#yum install ksh  
#yum install bc
```

## **Windows**

1. Create folder "scripts" in c:\oracle and extract scripts.

2. Set the values for SQLPATH and TNS\_ADMIN.

a) Open command prompt.

```
Start>Run>cmd.exe
```

b) Open system properties.

```
C:\>control.exe %windir%\system32\sysdm.cpl
```

c) Select Advanced>Environment Variables>System Variables>New.

d) Enter the name and value as shown below.

```
SQLPATH=c:\oracle\scripts\exa\ini;c:\ora  
cle\scripts\exa\sql  
TNS_ADMIN=c:\oracle\scripts\exa\tns
```

3. Make necessary environment changes to

the following file. dba.env

**Note:** This is required only if the host is a database server.

# Configuration

## **1. Oracle files and directories.**

Specify the values for the following entries in exa.ini file.

- ✓ ora\_tab
- ✓ ORACLE\_BASE
- ✓ TNS\_ADMIN

## **2. Title**

Specify the following information in cnf.ini file.

- ✓ Company Name
- ✓ Company Address
- ✓ Author

## **3. Notifications**

The following notifications can be set in cnf.ini file.

- ✓ email
- ✓ sms
- ✓ pager

There are two types of notifications.

- ✓ Default
- ✓ Ad-hoc

### **Default**

Default notifications are regular notifications that will be sent to DBA group when production is live.

### **Ad-hoc**

Ad-hoc notifications are used when notifications are to be sent to a specific DBA while performing maintenance or support activities.

The anatomy of -esp (email, sms and pager) value is as shown below.

-esp|-e [{0|1}{0|1}{0|1}[a-z]].

0 - Turnoff

1 - Turnon

Contents of cnf.ini file related to notifications are shown below.

```
#Default DBA Notifications dba_email=<email> dba_sms=<sms> dba_pager=<pager>
return_recipient=<email>
```

```
#Ad-hoc DBA Notifications dba_email_x=<email> dba_sms_x=<sms>
dba_pager_x=<pager> return_recipient=<email>
```

Example

1. To send notifications to default group, specify at -esp|-e as below.

- e 100 for email.
- e 110 for email and mobile sms.
- e 111 for email, mobile sms and pager.

2. To send notifications to ad-hoc group, specify at -esp|-e as below.

- e 100x for email.
- e 110x for email and mobile sms.
- e 111x for email, mobile sms and pager.

## **4. Defaults**

Default values for different inputs in the scripts can be specified in cnf.ini file. The following are the list of entries that can be found in cnf.ini file. These are self-explanatory.

#2.File Permissions

umask 026

#3.Default Notification Codes

stp\_esp="000"

mmt\_esp="000"

mtr\_esp="000"



```
dar_esp="000"
run_esp="000"
rco_esp="000"
trs_esp="000"
sho_esp="000"
```

## 5. Cron

It is recommended to set the following jobs in Oracle cron as described below.

Job#	Group	Description	Frequency	esp		Script	Environment			
				P&S	U&D		PRD	STB	UAT	DEV
1.1	MNT	Hot backup.	Weekly	100	100	mnt_rmn_bkp.ksh	Y	N/A	Y	Y
1.2	MNT	Archive Log backup	Daily	100	100	mnt_rmn_bkp.ksh	Y	Y	Y	Y
1.3	MNT	Control file backup.	Daily	100	100	mnt_rmn_bkp.ksh	Y	N/A	Y	Y
1.4	MNT	Structure export.	Daily	100	100	mnt_exp_dp.ksh	Y	N/A	Y	Y
1.5	MNT	Data export.	Daily	100	100	mnt_exp_dp.ksh	Y	N/A	Y	Y
1.6	MNT	File housekeeping.	Daily	100	100	mnt_exa_hkf.ksh	Y	Y	Y	Y
1.7	MNT	Oracle housekeeping.	Daily	100	100	mnt_ora_hkf.ksh	Y	Y	Y	Y
1.8	MNT	Audit housekeeping.	Daily	100	100	mnt_aud_hkf.ksh	Y	Y	Y	Y
1.9	MNT	Block corruption.	Weekly	100	100	mnt_anz_db.ksh	Y	N/A	Y	Y
1.10	MNT	Database Analyze.	Daily	100	100	mnt_anz_db.ksh	Y	N/A	Y	Y
2.1	MTR	Instance status.	10 Min	101	100	mtr_ins_sts.ksh	Y	Y	Y	Y
2.2	MTR	Listener status.	10 Min	101	100	mtr_lis_sts.ksh	Y	Y	Y	Y
2.3	MTR	Server status.	10 Min	101	100	mtr_host_ping.ksh	Y	Y	Y	Y
2.4	MTR	Tablespace	Daily	100	100	mtr_tbs_spa.ksh	Y	N/A	Y	Y
2.5	MTR	File System	Daily	100	100	mtr_host_disk.ksh	Y	Y	Y	Y
2.6	MTR	Alert log.	1 Hour	101	100	mtr_alt_log.ksh	Y	Y	Y	Y
2.7	MTR	Sys log.	1 Hour	101	100	mtr_sys_log.ksh	Y	Y	Y	Y

## Example

```
#####
####                      OS Watcher Monitoring                      #####
#####
00 * * * * /home/oracle/scripts/exa/unix/run_crn_job.ksh -f "mtr_osw_sts.ksh -a y -r 728
-e 100" > /dev/null 2>&1
```

## 6. Job Schedule

It is recommended to schedule the above daily jobs as described below.

1.00:00 - 06:00	- Backup
2.06:00 - 12:00	- Monitoring (Regular)
3.18:00 - 00:00	- Housekeeping

Schedule weekly jobs during the time when there is minimal business activity in the database.

## 7. Auto Start and Stop

Create the following soft links in /etc/rc?.d directory. These will perform auto start and stop of databases and listeners.

S.No.	Link	Description
1	ln -s ~oracle/scripts/exa/unix/Sora Sora	Auto Start
2	ln -s ~oracle/scripts/exa/unix/Sora Kora	Auto Stop

Once the link is created, specify the database role in oratab file at the end of every entry separated by colon (:).

S.No.	Server	Database Role
1	Primary	P
2	Standby	S
3	Data Guard	D

### Syntax

<SID>:<ORACLE\_HOME>:<AUTO\_START>:<DATABASE\_ROLE>

### Example

ORCL:~oracle/product/10.2.0.1:N:P

STBY:~oracle/product/10.2.0.1:N:D

**Note:** 1. Standard oratab file does not contain <DATABASE\_ROLE>.  
2. Always specify AUTO\_START=N.

## Getting Started

### ***Database Environment***

After scripts setup is completed, run exa.env script to prepare the environment.

```
$. ~oracle/scripts/exa/env/exa.env
```

**Note:** If scripts are copied in different location, then specify the appropriate path. When you run exa.env, it will set to first database environment listed in oratab file.

```
$Switched to db <DB Name> environment on host <Hostname>!
```

If no entries exist in oratab, it leaves message as shown below.

```
$oratab file is blank!
```

Now you are ready to run any unix script from any directory. To see the help of the script of your choice, simply type the script name with -h or -help argument.

```
$ksh run_sql_file.ksh -help
```

## Directory Navigation

You can navigate to the following directories just by typing following short name at unix prompt.

S.No.	Directory Name	Short Name	Description
0	exa	exa	Scripts
1	alt	alt	Alert Files
2	ash	ash	ASH Files
3	awr	awr	AWR Files
4	bat	bat	Windows Scripts
5	crn	crn	Cron Backup
6	doc	doc	Documents
7	env	en	Environment Files
8	ini	ini	Ini Files
9	java	jav	Java Files
10	job	job	User Cron Job Files
11	lib	lib	Library Files
12	log	log	Log Files
13	mas	mas	Master Files
14	oby	oby	Golden Gate Obey Files
15	ora	ora	Oracle Support Scripts
16	osw	osw	OSWatcher
17	out	out	Troubleshooting Logs*
18	pal	pal	3 <sup>rd</sup> Party Scripts
19	par	par	Parameter Files
20	pie	pie	Unix Pipes
21	plb	plb	Encrypted PL/SQL Code
22	pls	pls	PL/SQL Scripts
23	prm	prm	Golden Gate Parameters
24	psu	psu	PSU Patching
25	rda	rda	Remote Diagnostic Agent
26	rmn	rmn	RMAN Scripts
27	rpt	rpt	Report Files
28	rsp	rsp	Response Files
29	sch	sch	Schema Changes
30	shc	shc	Health Check
31	spl	spl	Special Files
32	sql	sql	SQL Scripts
33	sqt	sqt	SQLT
34	tmp	tmp	Temp Files
35	tns	tns	tnsnames.ora
36	unx	unx	Unix Scripts

\*As out files are intended for troubleshooting purpose e.g. Oracle Support, housekeeping scripts do not remove files from this folder.

# Setup

## 1. Database

### 1.1. stp\_cr\_db.ksh

This script creates database.

Syntax

```
stp_cr_db.ksh -n <db name> -v <db ver> -m <ora home> -x <mp pfx> -p <par  
file> -l <label> -t <title> -e <esp>
```

Parameters

- n Database Name.
- v Database Version(9i|10g|11g|12c).
- m Oracle Home.
- lp [o] Listener Port.
- x [o] Mount Point Directory prefix.
- p [o] Name of par file.
- l [o] Label for rpt/alt/log file. \*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

Defaults

- x "u"
- p \${par\_dir}/cr\_db\_{-v}.par

Example

```
stp_cr_db.ksh -n ORCL -v 10g -m ~oracle/product/10.2.0.1 -x u -p  
${par_dir}/cr_db_9i.par -l ORCL -t ORCL-CREATE-DB -e 100x
```

## 1.2. stp\_rm\_db.ksh

This script removes database.

### Syntax

```
stp_rm_db.ksh -s <sid> -p <parameters> -l <label> -t <title> -e <esp>
```

### Parameters

-s Instance Name.

-x [o] Mount Point Directory prefix.

-p Parameters (act={run|gen}:adm=[y|n]).

-l [o] Label for rpt/alt/log file. \*

-t [o] Title for report/alert.

-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Brace Parameters

act Action.

run -Run.

gen -Generates script.

adm Remove admin directories.

y [d] -Removes.

n -Does not remove.

### Defaults

-x "u"

### Example

```
stp_rm_db.ksh -s ORCL -p {act=run} -l ORCL -t ORCL-REM-DB -e 100x
```

# Maintenance

## 1. Env Settings

### 1.1 mnt\_sw\_db.ksh

This script sets database environment of the given sid or selected sid. This script can be used either in interactive mode or non-interactive mode.

#### Syntax

##### 1. Interactive Mode

```
mnt_sw_db.ksh
```

##### 2. Non-Interactive Mode

```
mnt_sw_db.ksh <sid>
```

#### Alias

```
swdb
```

#### Example

##### 1. Interactive Mode

```
mnt_sw_db.ksh or swdb
```

##### 2. Non-Interactive Mode

```
mnt_sw_db.ksh smpl
```

#### Output

```
$ swdb
```

Here are the databases on <hostname>, your sid is currently ORCL

- 1) ORCL
- 2) GNRL

Which database would you like to switch to?

## **2. Listener**

### **2.1. mnt\_st\_lis.ksh**

This script starts listener(s).

#### **Syntax**

```
mnt_st_lis.ksh -i <lis> -s <sid> -l <label> -t <title> -e <esp>
```

#### **Parameters**

- i Listener name|ALL.
- s Instance name (Only for single listener).
- l [o] Label for rpt/alt/log file. \*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### **Example**

```
mnt_st_lis.ksh -i LISTENER -s ORCL -l ORCL -t LISTENER-START -e 100  
mnt_st_lis.ksh -i ALL -l ALL -t ALL-LISTENER-START -e 100
```

### **2.2. mnt\_sp\_lis.ksh**

This script stops listener(s).

#### **Syntax**

```
mnt_sp_lis.ksh -i <lis> -s <sid> -l <label> -t <title> -e <esp>
```

#### **Parameters**

- i Listener name|ALL.
- s Instance name (Only for single listener).
- l [o] Label for rpt/alt/log file. \*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### **Example**

```
mnt_sp_lis.ksh -i LISTENER -s ORCL -l ORCL -t LISTENER-STOP -e 100  
mnt_sp_lis.ksh -i ALL -l ALL -t ALL-LISTENER-STOP -e 100
```



### 3. Instance

#### 3.1. mnt\_st\_db.ksh

This script starts database(s).

##### Syntax

```
mnt_st_db.ksh -s <sid> -p <parameters> -l <label> -t <title> -e <esp>
```

##### Parameters

- s Instance name|ALL.
- p [o] Parameters (opt=[pri|stb|odg]) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file. \*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

##### Brace Parameters

- opt Startup option.
- pri [d] -Primary database.
- stb -Standby database.
- odg -Oracle Data Guard.

##### Example

```
mnt_st_db.ksh -s ORCL -p {opt=pri} -l ORCL -t ORCL-START -e 100
mnt_st_db.ksh -s ORCL -p {opt=odg} -l ORCL -t ORCL-START -e 100
```

#### 3.2. mnt\_sp\_db.ksh

This script stops database(s).

##### Syntax

```
mnt_sp_db.ksh -s <sid> -p <parameters> -l <label> -t <title> -e <esp>
```

##### Parameters

- s Instance name|ALL.
- p [o] Parameters (opt=[nor|imm|abo]) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file. \*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

##### Brace Parameters

opt     Shutdown option.  
nor [d] -Normal.  
imm    -Immediate.  
abo    -Abort.

Example

```
mnt_sp_db.ksh -s ORCL -p {opt=imm} -l ORCL -t ORCL-STOP -e 100
```

## 4. Agent

### 4.1. mnt\_st\_agt.ksh

This script starts agent.

Syntax

```
mnt_st_agt.ksh -d <agent home directory> -l <label> -t <title> -e <esp>
```

Parameters

-d       EM Agent home directory.  
-l [o]   Label for rpt/alt/log file.\*  
-t [o]   Title for report/alert.  
-e [o]   email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

Example

```
mnt_st_agt.ksh -d ~oracle/product/12.1.0.5/agent/agent_inst -e 100
```

### 4.2 mnt\_sp\_agt.ksh

This script stops agent.

Syntax

```
mnt_sp_agt.ksh -d <agent home directory> -l <label> -t <title> -e <esp>
```

Parameters

-d       EM Agent home directory.  
-l [o]   Label for rpt/alt/log file.\*  
-t [o]   Title for report/alert.  
-e [o]   email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

Example

```
mnt_sp_agt.ksh -d ~oracle/product/12.1.0.5/agent/agent_inst -e 100
```

## 5. OSWatcher

### 5.1. mnt\_st\_osw.ksh

This script starts OSWatcher.

#### Syntax

```
mnt_st_osw.ksh -d <OSWatcher directory> -l <label> -t <title> -e <esp>
```

#### Parameters

- d OSWatcher directory.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
mnt_st_osw.ksh -d ~oracle/scripts/osw -e 100
```

### 5.2. mnt\_sp\_osw.ksh

This script stops OSWatcher.

#### Syntax

```
mnt_sp_osw.ksh -d <OSWatcher directory> -l <label> -t <title> -e <esp>
```

#### Parameters

- d OSWatcher directory.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
mnt_sp_osw.ksh -d ~oracle/scripts/osw -e 100
```

## 6. Backups

### 6.1. mnt\_exp\_dmp.ksh

This script does logical export of the given database.

#### Syntax

```
mnt_exp_dmp.ksh -s <sid> -o <loc sid> -d <dmp directory> -p <par file> -c <cut files> -z <file size> -n <db env> -k <keep days> -l <label> -t <title> -e <esp>
```

#### Parameters

- s Instance name.
- o Local instance name. Use this if connected from remote machine.
- d Dump file directory.
- p Par file name or parfile contents enclosed in curling braces ({}).
- c [o] Split option [y/n].
- z Split file size. Required only if split option is "y".
- n [o] Database environment report [y/n].
- k [o] # of days to keep old backups.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}]{0|1}{0|1}[a-z]].

#### Defaults

- c "n"
- n "n"

#### Example

```
mnt_exp_dmp.ksh -s ORCL -d /u02/oradata/ORCL/expbkp -p  
~oracle/scripts/exa/par/exp_cmpd_9i.par -c y -z 8192m -n n -k 2 -l ORCL-cmpd -  
t ORCL-DATA-EXP -e 100
```

### 6.2. mnt\_imp\_dmp.ksh

This script does logical import from the given compressed dump file/directory.

#### Syntax

```
mnt_imp_dmp.ksh -s <sid> -o <loc sid> -f <dmp file> -d <dmp directory> -p  
<par file> -c <cut files> -l <label> -t <title> -e <esp>
```

#### Parameters

- s Instance name.

- o Local instance name. Use this if connected from remote machine.
- f Dump file name.
- d Dump file directory. Required only if split option is "y".
- p Par file name or parfile contents enclosed in curling braces ({}).
- c [o] Split option [y/n].
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Defaults

- c "n"

#### Example

```
mnt_imp_dmp.ksh -s ORCL -d /u02/oradata/ORCL/expbkp-p "{fromuser=pie  
touser=pie ignore=y}" -c y -l ORCL-PIE-IMP -t ORCL-PIE-IMP -e 100
```

### 6.3. mnt\_col\_bkp.ksh

This script does cold backup of the given database.

#### Syntax

```
mnt_col_bkp.ksh -s <sid> -d <bkp directory> -p <parameters> -l <label> -t  
<title> -e <esp>
```

#### Parameters

- s Instance name.
- d Backup file directory.
- p Parameters (act={run|gen}:bsz=[n]:up=[y|n]) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

- act Action.
- run -Run.
- gen -Generates script.
- bsz Backup set size.
- up Startup option after backup.
- y [d] -Brings database up after backup.
- n -Does not bring database up after backup.

#### Example

```
mnt_col_bkp -s ORCL -d /u02/oradata/ORCL/colbkp -p {act=run:bsz=10} -l  
ORCL -t ORCL-COL-BKP -e 100
```

## 6.4. mnt\_hot\_bkp.ksh

This script does hot backup of the given database.

### Syntax

```
mnt_hot_bkp.ksh -s <sid> -d <bkp directory> -p <parameters> -l <label> -t  
<title> -e <esp>
```

### Parameters

- s Instance name.
- d Backup file directory.
- p Parameters (act={run|gen}) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Brace Parameters

- act Action.
- run -Run.
- gen -Generates script.

### Example

```
mnt_hot_bkp -s ORCL -d /u02/oradata/ORCL/hotbkp -p {act=run} -l ORCL -t  
ORCL-HOT-BKP -e 100
```

## 6.5. mnt\_tts\_bkp.ksh

This script does transportable tablespaces backup of the given database.

### Syntax

```
mnt_tts_bkp.ksh -s <sid> -d <bkp directory> -p <parameters> -l <label> -t  
<title> -e <esp>
```

### Parameters

- s Instance name.
- d Backup file directory.

-p Parameters (act={run|gen}:bsz=[n]) separated by colon (:) and enclosed in curling braces ({}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act Action.  
run -Run.  
gen -Generates script.  
bsz Backup set size.

#### Example

```
mnt_tts_bkp -s ORCL -d /u02/oradata/ORCL/ttsbkp -p {act=run:bsz=10} -l  
ORCL -t ORCL-TTS-BKP -e 100
```

## 6.6. mnt\_ctl\_bkp.ksh

This script does controlfile backup of the given database.

#### Syntax

```
mnt_ctl_bkp.ksh -s <sid> -d <bkp directory> -p <parameters> -l <label> -t  
<title> -e <esp>
```

#### Parameters

-s Instance name.  
-d Backup file directory.  
-p Parameters (act={run|gen}) separated by colon (:) and enclosed in curling braces ({}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act Action.  
run -Run.  
gen -Generates script.

#### Example

```
mnt_ctl_bkp -s ORCL -d /u02/oradata/ORCL/ctlbkp -p {act=run} -l ORCL -t  
ORCL-CTL-BKP -e 100
```

## 6.7. mnt\_dp\_exp.ksh

This script does data pump export of the given database.

### Syntax

```
mnt_dp_exp.ksh -s <sid> -o <loc sid> -d <dmp directory name> -p <par file> -r  
<data pump recovery template> -k <keep days> -l <label> -t <title> -e <esp>
```

### Parameters

- s Instance name.
- o Local instance name. Use this if connected from remote machine.
- d [o] Data Pump directory name.
- p [o] Par file name or parfile contents enclosed in curling braces ({}).
- r [o] Data pump recovery template [y/n].
- k [o] # of days to keep old backups.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Defaults

- d DATA\_PUMP\_DIR
- p dp.par
- r "y"

### Example

```
mnt_dp_exp.ksh -s ORCL -l ORCL -t ORCL-DATA-PUMP-EXP -e 100
```

## 6.8. tde\_wall\_bkp.ksh

This script does backup of TDE wallet files.

### Syntax

```
tde_wall_bkp.ksh -s <sid> -d <backup directory name> -k <keep days> -t <title>  
-e <esp>
```

### Parameters

- s Instance name.
- d Backup directory name.
- k # of days to keep old backups.
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Example

```
tde_wall_bkp.ksh -s ORCL -d /tmp -k 10 -t ORCL-TDE-WALL-BKP -e 100
```



## 7. Housekeeping

### 7.1. mnt\_exa\_hkf.ksh

This script removes files present in rpt, alt, log, tmp, spl and pie directories.

#### Syntax

```
mnt_exa_hkf.ksh -k <keep days> -p <parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

- k # of days to keep files.
- p Parameters (act={run|gen}:hst={day|mon}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

- act Action.
- run -Run.
- gen -Generates script.
- hst History of mnt\_exc\_crm.log.

#### Example

```
mnt_exa_hkf.ksh -k 30 -p {act=run:hst=mon} -l ATLANTIC -t ATLANTIC-  
EXA-FILES -e 100
```

### 7.2. mnt\_ora\_hkf.ksh

This script removes files present in bdump, udump and cdump directories.

#### Syntax

```
mnt_ora_hkf.ksh -k <keep days> -p <parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

- k # of days to keep files.
- p Parameters (act={run|gen}:hst={n}:arc=[y|n]).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act     Action.  
run     -Run.  
gen     -Generates script.  
hst     History of alert log in days.  
arc     Archival of history.  
y [d]   -Archives history.  
n       -Does not archive history.

#### Example

```
mnt_ora_hkf.ksh -k 30 -p {act=run:hst=7:arc=n} -l ATLANTIC -t ATLANTIC-ORA-FILES -e 100
```

### 7.3. mnt\_arc\_hkf.ksh

This script compresses and removes Oracle archive log files.

#### Syntax

```
mnt_arc_hkf.ksh -k <keep days> -p <parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

-k       # of days to keep files.  
-p       Parameters (act={run|gen}).  
-l [o]   Label for rpt/alt/log file.\*  
-t [o]   Title for report/alert.  
-e [o]   email/sms/pager notification [{0|1} {0|1} {0|1} [a-z]].

#### Brace Parameters

act     Action.  
run     -Run.  
gen     -Generates script.

#### Example

```
mnt_arc_hkf.ksh -k 7 -p {act=run} -l ATLANTIC -t ATLANTIC-ARC-FILES -e 100
```

### 7.4. mnt\_aud\_hkf.ksh

This script compresses and removes Oracle audit files.

#### Syntax

```
mnt_aud_hkf.ksh -k <keep days> -p <parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

-k # of days to keep files.  
-p Parameters (act={run|gen}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act Action.  
run -Run.  
gen -Generates script.

#### Example

```
mnt_aud_hkf.ksh -k 30 -p {act=run} -l ATLANTIC -t ATLANTIC-AUD-FILES  
-e 100
```

## 7.5. mnt\_old\_hkf.ksh

This script removes old files.

#### Syntax

```
mnt_old_hkf.ksh -d <dir> -x <file ext> -k <keep days> -p <parameters> -l  
<label> -t <title> -e <esp>
```

#### Parameters

-d Directory of files.  
-x Extension of file names. Specify "ALL" for all files.  
-k # of days to keep files.  
-p Parameters (act={run|gen}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act Action.  
run -Run.  
gen -Generates script.

#### Example

```
mnt_rm_fls.ksh -d ~oracle -x ALL -k 30 -p {act=run} -l ATLANTIC -t  
ATLANTIC-OLD-FILES -e 100
```

## 7.6. mnt\_asm\_aud.ksh

This script removes old ASM audit files.

### Syntax

```
mnt_asm_aud.ksh -k <keep days> -l <label> -t <title> -e <esp>
```

### Parameters

- k # of days to keep files.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Example

```
mnt_asm_aud.ksh -k 90 -l ATLANTIC -t ATLANTIC-ASM-AUD-FILES -e 100
```

## 8. Performance

### 8.1. mnt\_anz\_db.ksh

This script analyzes the given database(s).

### Syntax

```
mnt_anz_db.ksh -s <sid> -o <loc sid> -p <par file> -l <label> -t <title> -e <esp>
```

### Parameters

- s Instance name|ALL.
- o Local instance name. Use this if connected from remote machine.
- p Par file name or parfile contents separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Example

```
mnt_anz_db.ksh -s ORCL -p ~oracle/scripts/exa/par/anz_db.par -l ORCL -t  
ORCL-ANZ-DB -e 100  
mnt_anz_db.ksh -s ORCL -p {act=a:lev=d:pct=20:opt=4} -l ORCL -t ORCL-  
ANZ-DB -e 100
```

#### Brace Parameters

act	Action.
v	-Validate structure.
g	-Gather statistics.
b	-Both.
lev	Level.
d	-Database.
s	-Schema.
pct [o]	Percent.
opt [o]	Option.
1 [d]	-Gather.
2	-Gather empty.
3	-Gather stale.
4	-Gather auto.

## Monitoring

### 1. Status

#### 1.1. mtr\_ins\_sts.ksh

This script checks instance status.

##### Syntax

```
mtr_ins_sts.ksh -s <sid> -l <label> -t <title> -e <esp>
```

##### Parameters

-s	Instance name ALL.
-l [o]	Label for rpt/alt/log file.*
-t [o]	Title for report/alert.
-e [o]	email/sms/pager notification [{0 1}{0 1}{0 1}[a-z]].

##### Example

```
mtr_ins_sts.ksh -s ORCL -l ORCL -t ORCL-INSANCE-STATUS -e 101
```

#### 1.2. mtr\_lis\_sts.ksh

This script checks listener status.

#### Syntax

```
mtr_lis_sts.ksh -i <listener> -l <label> -t <title> -e <esp>
```

#### Parameters

- i Listener Name|ALL.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
mtr_lis_sts.ksh -i ORCL_LISTENER -l ORCL -t ORCL-LISTENER-STATUS -e 100
```

### 1.3. mtr\_host\_ping.ksh

This script pings the given host(s) and alerts if response not received from the host(s).

#### Syntax

```
mtr_host_ping.ksh -p <par file> -l <label> -t <title> -e <esp>
```

#### Parameters

- p [o] Par file name.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Default

Parameter file is ping.ini.

#### Example

```
mtr_host_ping.ksh -l smg -t SMG-DOMAIN-SERVER(S)-STATUS -e 101
```

## 2. Space

### 2.1. mtr\_tbs\_spa.ksh

This script checks tablespace space and reports the tablespace(s) of each or individual database(s) that are exceeding the regular space limit.

#### Syntax

```
mtr_tbs_spa.ksh -s <sid> -o <loc sid> -p <parameters> -l <label> -t <title> -e  
<esp>
```

#### Parameters

- s Instance name|ALL.
- o Local instance name. Use this if connected from remote machine.
- p Parameters (rts\_lmt={nn}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

rts\_lmt Threshold limit (% used).

#### Example

```
mtr_tbs_spa.ksh -s ORCL -p {rts_lmt=90} -l ORCL -t ORCL-TABLESPACES-  
SPACE -e 100
```

## 2.2. mtr\_asm\_disk.ksh

This script checks asm file system space and reports the file systems that are exceeding the specified threshold.

#### Syntax

```
mtr_asm_disk.ksh -p <par file> -l <label> -t <title> -e <esp>
```

#### Parameters

- p [o] Par file name.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Default

Parameter file is asm.ini.

#### Example

```
mtr_asm_disk.ksh -l ATLANTIC -t ATLANTIC-FILE-SYSTEM-SPACE -e 111
```

## 2.3. mtr\_host\_disk.ksh

This script checks file system space and reports the file systems that are exceeding the specified threshold.

### Syntax

```
mtr_host_disk.ksh -p <par file> -l <label> -t <title> -e <esp>
```

### Parameters

- p [o] Par file name.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Default

Parameter file is disk.ini.

### Example

```
mtr_host_disk.ksh -l ATLANTIC -t ATLANTIC-FILE-SYSTEM-SPACE -e 100
```

## 3. Log

### 3.1. mtr\_alt\_log.ksh

This script reports errors that occur in database alert log. It can be fired on each day or each hour.

### Syntax

```
mtr_alt_log.ksh -s <sid> -p <parameters> -l <label> -t <title> -e <esp>
```

### Parameters

- s Instance name|ALL.
- p Parameters (fire={day|hour}) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Brace Parameters

fire Error checking interval.



day -Checks errors once in a day.  
hour -Checks errors once in an hour.

#### Example

```
mtr_alt_log.ksh -s ORCL -p {fire=day} -l ORCL -t ORCL-ALT-LOG -e 111
```

### 3.2. mtr\_sys\_log.ksh

This script reports errors that occur in server log. It can be fired on each day or each hour.

#### Syntax

```
mtr_sys_log.ksh -p <parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

-p Parameters (fire={day|hour}) separated by colon (:) and enclosed in curling braces ({}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

fire Error checking interval.  
day -Checks errors once in a day.  
hour -Checks errors once in an hour.

#### Example

```
mtr_sys_log.ksh -s ORCL -p {fire=day} -l L000000 -t L000000-SYS-LOG -e 111
```

## 4. Replication

### 4.1. mtr\_dg\_gap.ksh

To monitor Data Guard gap. Run on primary database.

#### Syntax

```
mtr_dg_gap.ksh -s <sid> -g <lag> -l <label> -t <title> -e <esp>
```

#### Parameters

-s Instance name.  
-p Process names separated by colon (:) and enclosed in curling braces ({}).  
-g Gap.

-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
mtr_dg_gap.ksh -s ORCL -g 20 -e 100
```

#### Defaults

-g 10

## 4.2. mtr\_gg\_lag.ksh

This script monitors Golden Gate processes and replication lag.

#### Syntax

```
mtr_gg_lag.ksh -s <sid> -p <process> -g <lag> -l <label> -t <title> -e <esp>
```

#### Parameters

-s Instance name|ALL.  
-p Process names separated by colon (:) and enclosed in curling braces ({}).  
-g Lag in minutes.  
-d Golden Gate home directory.  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
mtr_gg_lag.ksh -s ORCL -p ALL -g 30 -d /home/oracle/product/ggs/19.1.0 -e 100
```

#### Defaults

-p ALL  
-g 60 Minutes

## Reports

### 1. *dar\_mem\_pad.ksh*

This script provides memory stats.

#### Syntax

```
dar_mem_pad.ksh -s <sid> -o <loc sid> -l <label> -t <title> -e <esp>
```

#### Parameters

- s Instance name|ALL.
- o Local instance name. Use this if connected from remote machine.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Example

```
dar_mem_pad.ksh -s ORCL -l ORCL -t ORCL-MEM-PAD -e 100
```

## Support

### 1. *run\_crn\_job.ksh*

This script executes the given script from cron. This script does not use .profile of the cron user.

#### Syntax

```
run_crn_job.ksh -a <scripts_home> -f <script_name>
```

#### Parameters

- a [o] Scripts home path.
- f Script file name.

#### Example

```
12 19 * * * ~oracle/scripts/exa/unix/run_crn_job.ksh -f "mnt_exp_dmp.ksh -s  
ORCL -d  
/u02/oradata/ORCL/dump -p ~oracle/scripts/exa/par/exp_cmpd_9i.par -c y -z  
8192m -n n -k 2 -l ORCL-cmpd -t ORCL-DATA-EXP -e y" > /dev/null 2>&1
```

## 2. *run\_sql\_file.ksh*

This script runs the given SQL file in one or all databases. This can be run on one database or set of databases as mentioned in SID file and or databases listed in oratab file by specifying ALL in SID.

### Syntax

```
run_sql_file.ksh -s <sid> -o <loc sid> -S <sid file> -f <sql file> -l <label> -t <title>
-e <email>
```

### Parameters

- s Instance name|ALL.
- o Local instance name. Use this if connected from remote machine.
- S SID file name.
- f SQL file name.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Example

```
run_sql_file.ksh -s ALL -f $ora_dir/db_env.sql -l ATLANTIC -t DB-ENV -e 100
```

## 3. *run\_she\_file.ksh*

This script runs the given shell file.

### Syntax

```
run_she_file.ksh -f <sql file> -l <label> -t <title> -e <email>
```

### Parameters

- f Shell file name.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Example

```
run_sql_file.ksh -f $unx_dir/svr_stat.sh -l stats -t SERVER-STATS -e 100
```

## 4. *run\_aut\_ssh.ksh*

This script transfers files via ssh.

### Syntax

```
run_aut_ssh.ksh -rs <rmt svr> -sd <src dir> -td <tar dir> -x <file ext> -p  
<parameters> -l <label> -t <title> -e <esp>
```

### Parameters

- rs Remote server.
- sd Source directory.
- td Target directory.
- x Extension of file names. Specify "ALL" for all files.
- p Parameters (act={run|gen}:tsz=[n]) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*  
(Should be alpha numeric with special characters - and \_ only. Blank space is not allowed.)
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

### Brace Parameters

- act Action.
- run -Run.
- gen -Generates script.
- tsz Transfer set size.

### Example

```
run_aut_ssh.ksh -rs 150.110.177.219 -sd /u02/oradata/ORCL/colbkp -td  
/u02/oradata/ORCL/colbkp -x ALL -p {act=run:tsz=20} -l ORCL -t ORCL-COL-  
AUT-SSH -e 111
```

# Recovery

## 1. *rco\_mng\_file.ksh*

This script manages file(s).

### Syntax

```
rco_mng_file.ksh -sd <src dir> -td <tar dir> -x <file ext> -p <parameters> -l  
<label> -t <title> -e <esp>
```

#### Parameters

-sd Source directory.  
-td Target directory.  
-x Extension of file names. Specify "ALL" for all files.  
-p Parameters (act={run|gen}:cmd={c|u|p|m}:fsz=[n]) separated by colon (:) and enclosed in curling braces ({}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act Action.  
run -Run.  
gen -Generates script.  
cmd Command.  
c -Compress  
u -Uncompress  
p -Copy  
m -Move  
fsz File set size.

#### Example

```
rco_mng_file.ksh -sd /u03/oradata/ORCL -x ALL -p {act=run:cmd=u:fsz=10} -l  
ORCL -t ORCL-UNC-ARC -e 100x
```

## 2. *rco\_syn\_stb.ksh*

This script sync standby database.

#### Syntax

```
run_syn_stb.ksh -s <sid> -rs <rmt svr> -sd <src dir> -td <tar dir> -p  
<parameters> -l <label> -t <title> -e <esp>
```

#### Parameters

-s Instance name.  
-rs Remote server.  
-sd Source directory.  
-td Target directory.  
-p Parameters (act={run|gen}:fsz=[n]) separated by colon (:) and enclosed in curling braces ({}).  
-l [o] Label for rpt/alt/log file.\*  
-t [o] Title for report/alert.  
-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

act	Action.
run	-Run.
gen	-Generates script.
fsz	File set size.

#### Example

```
rco_syn_stb.ksh -s ORCL -rs gwmrutidbpd1 -sd /u03/oradata/ORCL -td  
/u03/oradata/ORCL -p {act=run:fsz=10} -l ORCL -e 100
```

## Miscellaneous

### **1. *sho\_file\_cnt.ksh***

Shows count of files in current working directory.

#### Syntax

```
sho_file_cnt.ksh
```

#### Example

```
sho_file_cnt.ksh
```

### **2. *sho\_file\_sum.ksh***

Shows sum of file sizes in current working directory in megabytes.

#### Syntax

```
sho_file_sum.ksh
```

#### Example

```
sho_file_sum.ksh
```

### **3. *sho\_disk\_sum.ksh***

Shows sum of file system sizes in the server in giga bytes.

#### Syntax

```
sho_disk_sum.ksh
```

#### Example

```
sho_disk_sum.ksh
```

### **4. sho\_crc\_sum.ksh**

Shows cyclical redundancy check sum.

#### Syntax

```
sho_crc_sum.ksh -d <file dir> -o <file order> -p <parameters> -l <label> -t  
<title> -e <esp>
```

#### Parameters

- d Directory of file(s).
- o [o] Order of files.
- p Parameters (act={run|gen}:csz=[n]) separated by colon (:) and enclosed in curling braces ({}).
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

#### Brace Parameters

- act Action.
- run -Run.
- gen -Generates script.
- csz Crc set size.

#### Example

```
sho_crc_sum.ksh -d /u02/oradata/ORCL/colbkp -p {act=run:csz=10} -l ORCL -t  
ORCL-COL-BKP-CRC-SUM -e 100
```

### **5. sho\_host\_disk.ksh**

Shows file system space.

#### Syntax

```
sho_host_disk.ksh -n <nas > -l <label> -t <title> -e <esp>
```

#### Parameters

- n Include NAS.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.



-e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

Example

```
sho_host_disk.ksh -n y -l <hostname> -t HOST-DISK-REPORT -e 100
```

## **6. sho\_pat\_con.ksh**

To check one off patch conflict.

Syntax

```
sho_pat_con.ksh -d <patch directory > -l <label> -t <title> -e <esp>
```

Parameters

- d Patch directory.
- l [o] Label for rpt/alt/log file.\*
- t [o] Title for report/alert.
- e [o] email/sms/pager notification [{0|1}{0|1}{0|1}[a-z]].

Example

```
sho_pat_con.ksh -d ~oracle/patches -l <hostname> -t PATCH-CONFLICT -e 100
```

## **7. sho\_aix\_node.ksh**

Shows nodes in AIX cluster.

Syntax

```
sho_aix_node.ksh
```

Example

```
sho_aix_node.ksh
```

# Troubleshooting

## ***Problem #1***

Error: ": command not found"

Resolution: 1. Copy the directory contents in ASCII format. env, ini, lib/exa.lib, par, unix/mnt\_sw\_db.ksh.

2. Make sure ksh is present in lib folder. If not, copy the respective ksh located in lib folder.

# Appendix

## ***1. Interpretations***

[o] -Optional.

[d] -Default.

P&S -Production and Standby U&D -UAT and Development

## ***2. Fine Print***

-l [o] Label for rpt/alt/log file.\*

\* Should be alpha numeric with special characters - and \_ only. Blank space not allowed.