

**NAME**

oracold.ksh – make Oracle database cold backup, with Solaris 8 snapshot option.

**SYNOPSIS**

```
oracold.ksh help=y
```

```
oracold.ksh MySID |sid:=MySid [mount=y umount=n |mount=n umount=y] [debug=y]
```

**DESCRIPTION**

When Solaris 8 snapshot option is not available or not used, oracold.ksh generates a list of datafiles, logfiles, and controlfiles (“Oracle files”) to be backed up, shutdown the database, backup the Oracle files, and then start the database. The database has to remain down for the duration of the cold backup without Solaris 8 snapshot option.

With Solaris 8 snapshot option, oracold.ksh generates a list of Oracle files, shutdown the database, create snapshots of the file systems for the database, start the database immediately. The database is only required to be down for the duration of snapshots creation, which is a matter of a few minutes. Then the backup is made against the snapshots, and when it is finished, the snapshot is deleted.

oracold.ksh can run the entire show if there are available file systems for backup, or it can cooperate with central Unix file system backup. In the latter case, it runs in two stages. First it runs with “mount=y umount=n” option just to create and mount the snapshots, and then leave the snapshot file systems to the central Unix backup. When the central backup finishes, we can schedule to run oracold.ksh again with “mount=n umount=y” to umount and delete the snapshots. The options can be specified either as command line options, or in the configuration file. The command line options have higher precedence over the configuration file.

As only root can create and delete snapshots, oracold.ksh include a small portion of the code for root to run. To make backup, the script needs to run by both Oracle and root user at same time. oracle will do oracle’s job, root will do root’s job. Oracle sends root instruction via dummy files in the file system. The one way communication is illustrated below:

```
$ ./oracold.ksh MySID # (Oracle user)
# ./oracold.ksh MySID # (root user)

root: (waiting for order from oracle.)
oracle: shutdown database.
oracle: root, please create snapshots.
root: (verify order) ok.
oracle: startup database.
oracle: backup snapshots.
oracle: root, please delete snapshots.
root: (verify order) ok.
root: bye.
oracle: bye.
```

The backup job with Solaris 8 snapshot option can be completed interactively by oracle user and root user, or by using both oracle and root’s crontabs, or by using root’s crontab alone with “su” command, or best of all, by creating a autosys box job:

```
insert_job: MYSID_COLD_BACKUP_BX job_type: b
owner: oracle
...
description: "BOX JOB that backs up MYSID with Solaris 8 snapshot."
```

```

insert_job: MYSID_COLD_BACKUP_ORACLE job_type: f
box_name: MYSID_COLD_BACKUP_BX
command: <job_dir>/ora_cold/script/oracold.ksh sid:=MySID
owner: oracle
...
description: "oracle's job in MYSID_COLD_BACKUP_BX"
insert_job: MYSID_COLD_BACKUP_ROOT job_type: f
box_name: MYSID_COLD_BACKUP_BX
command: <job_dir>/ora_cold/script/oracold.ksh sid:=MySID
owner: root
...
description: "root's job in MYSID_COLD_BACKUP_BX"

```

oracold.ksh also backs up init<SID>.ora, and “text copy of the controlfile” generated with “alter database controlfile to trace” as a courtesy (renamed “gen\_controlfile.sql” in the backup set) if it backup the Oracle files by itself.

To coordinate with central backup, then we need to setup two box jobs:

```

MYSID_COLD_BACKUP_A_BX
MYSID_COLD_BACKUP_B_BX

```

Both boxes are identical as MYSID\_COLD\_BACKUP\_BX except that the “A\_BX” has “MOUNT=Y UMOUNT=N” option, and “B\_BX” has “MOUNT=N UMOUNT=Y” option. The options can be dynamically changed in the configuration file per day of the week condition as shown in the sample configuration file below, or specified as command line options, which override the configuration file.

The job dependencies are the central backup job depends on the successful completion of A\_BX job, and the B\_BX depends on the successful completion of central backup job. The dependencies can be coded in autosys JIL code if central backup can be scheduled using autosys, or just by using time windows.

Other features include:

- keeps multiple (NUM\_PROCS) “compress” processes running at the same time (the compress utility can be compress, gzip, bzip2, or whatever which can takes stdin and compress to stdout). The large file (>2GB) support on the compress utility is *not* required.
- keeps multiple copies online (NUM\_COPIES).
- it uses multiple file systems including remote file systems, and dynamically balance the storage usage among the file systems.
- startup the database via signal handler when an exception is caught, and clean up the backup processes.
- maximum allowed run time, or backup skip (MAX\_SECS).
- Prevent duplicate backup processes from running.

## INSTALLATION

To install, follow these easy 1–2–3 steps:

1. Create an install directory for cold backup <job dir>/ora\_cold. And under ora\_cold, create following sub-directories:

```

script
etc
operlog/<SID>
temp/<SID>

```

Where SID is the actual ORACLE\_SID name. If you have multiple SIDs, then just create them under operlog and temp.

2.

```
cp oracold.ksh script/oracold.ksh
chmod 755 script/oracold.ksh
```

3. Edit etc/oracold\_SID.cf from the sample configuration shown below. If you have multiple SIDs, then just create multiple files.

```
# *
# * Where do you want the cold backup files to go?
# * It can be a directory on a remote server.
# *
BKUP_DIRS="/bkup01/cold oracle@castor:/bkup02/cold"

# *
# * How many simutaneous compress processes you want to run?
# * Rule of thumb is to set it to the number of processors.
# * Default is 3.
# *
NUM_PROCS=3

# *
# * How many copies you want to keep online?
# * ( If you keep only 1 copy, make sure the file system backup
# * occurs after the Oracle backup finishes. )
# *
NUM_COPIES=2

# *
# * What is the maximum time you allow the backup to run?
# * Backup will kill itself exceeding the maximum allowed time.
# * Default is 999999.
# *
# * If it is set to 0, then backup is skipped.
# *
MAX_SECS=999999

# *
# * What command do you use to compress a file to standard output?
# * Default is "compress -c".
# *
COMPRESS_C="/usr/bin/compress -c"
# COMPRESS_C="/usr/local/bin/gzip -c"
# COMPRESS_C="/usr/local/bin/bzip2 -c"

# *
# * Whether to copy files under each DIRECTORY_PATH
# * from dba_directories.
# *
COPY_DIRECTORY_PATH=Y

# *
# * FILTER_DIRECTORY_PATH: to include or exclude certain
# * OWNER, DIRECTORY_NAME, or DIRECTORY_PATH from DBA_DIRECTORIES.
# *
FILTER_DIRECTORY_PATH="where DIRECTORY_NAME not in ('AUDIT_DIR')"
```

```

# *
# * Where can you find submit?
# * /usr/local/bin/submit is the default.
# *

SUBMIT=<job_dir>/common/submit

# *
# * What is the directory where you can find oracle-SID?
# * /etc/init.d is the default.
# *

INIT_D=<job_dir>/common

# *
# * Do you need pre/post processing?
# *

# PRE_PROCESSING="/bin/true"
# POST_PROCESSING="/bin/true"

# *
# * Solaris 8 snapshot cold backup?
# * (1) shutdown the database;
# * (2) make a snapshot;
# * (3) startup the database immediately;
# * (4) backup the snapshot.
# *

SNAPSHOT=Y                # Y or N
[[ $SNAPSHOT = Y ]] && {  # if backup snapshot, then
    if [[ $(date +%A) = "_Saturday_" ]]; then
        MOUNT=Y UMount=N    # "Saturday" mount snapshot central backup
    elif [[ $(date +%A) = "_Sunday_" ]]; then
        MOUNT=N UMount=Y    # "Sunday" umount snapshot when done
    else
        # delete "_" chars in actual use
        MOUNT=Y UMount=Y
    fi
    [[ $MOUNT = Y ]] && {
        SNAPMTP=/snapshot    # what is the top level snapshot mount point?
        SNAPBKST=/var/tmp    # what is the backing store for the snapshot?
        (( WAIT_MOUNT = 300 )) # seconds to wait for root to mount snapshot?
    }
    [[ $UMOUNT = Y ]] && {
        (( WAIT_UMOUNT = 120 )) # seconds to wait for root to umount snapshot?
    }
}
}

```

**RESTTORE**

The ora\_cold.ksh copies Oracle data files, control files, and redo log files. The text copy of the control file gen\_controlfile.sql, and parameter file initSID.ora are also backed up gratuitously.

A sample hot backup file set is shown below:

```

/bkup01/cold/ora901/ora10/oradata/ora901/control01ctl.gz
/bkup01/cold/ora901/ora10/oradata/ora901/control02ctl.gz
/bkup01/cold/ora901/ora10/oradata/ora901/control03ctl.gz
/bkup01/cold/ora901/ora10/oradata/ora901/cwmlite01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/drsys01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/example01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/indx01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/redo01.log.gz
/bkup01/cold/ora901/ora10/oradata/ora901/redo02.log.gz
/bkup01/cold/ora901/ora10/oradata/ora901/redo03.log.gz
/bkup01/cold/ora901/ora10/oradata/ora901/system01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/tools01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/undotbs01.dbf.gz
/bkup01/cold/ora901/ora10/oradata/ora901/users01.dbf.gz
/bkup01/cold/ora901/gen_controlfile.sql
/bkup01/cold/ora901/initora901.ora

```

To restore, first copy Oracle data files, control files, and redo log files to where they were. As the original file path are recorded in the backup set, it is very easy to automate the task. The following *ad hoc* script does this.

```

#!/bin/ksh
find /bkup01/cold/ora901/ora* -type f | while read i; do
    j=${i#*ora901}
    dir=${j%/*}
    file=${j%.gz}

    CAT=cat
    [[ $i = *.gz ]] && CAT="gzip -dc"

    mkdir -p $dir
    $CAT $i > $file
done

```

What it does is something like,

```

mkdir -p /ora10/oradata/ora901
gzip -dc /bkup01/cold/ora901/ora10/oradata/ora901/system01.dbf.gz \
> /ora10/oradata/ora901/system01.dbf

```

Once all the files are in place, we can go ahead to start the database. For example,

```
rc.oracle sid=ora901 action=start
```

That is it, happy restore!

## SEE ALSO

submit help=y, rc.oracle help=y.

## AUTHOR

Michael Wang <xw73@columbia.edu> with help from many others acknowledged in version notes.

This is free software. You may copy or redistribute it under the same terms as Perl. However, if you modify it, you need to send the modification to the author via email.