Contents

[Purpose 2](#_Toc536536272)

[Components 2](#_Toc536536273)

[Invoking refresh\_db 2](#_Toc536536274)

[Example files and scripts 2](#_Toc536536275)

[Crontab setup 10](#_Toc536536276)

# Purpose

Korn Shell « refresh\_db.sh” script purpose is to duplicate an Oracle database using RMAN active duplication and customize the database copy according to specific needs. A set of files prepared for refresh of AXA’s UASS database is included as an example. Finally, this refresh\_db.sh script assumes identical folders structure for source and destination databases.

# Components

refresh\_db.sh is the main script, it takes 4 parameters

1. an executable file in which environment and duplication variables are defined
2. the duplicate database script to use. This script uses the previous parameter to define several variables used during the duplication.
3. a pre\_duplicate script in which actions before the duplication are performed like saving schemas, tables etc … These actions are specific to the environment being refreshed
4. a post\_duplicate script in which actions after the duplication are performed like stopping Oracle archive log mode, refreshing schemas, tables etc … These actions are specific to the environment being refreshed

# Invoking refresh\_db

**$PWD/refresh\_db.sh $PWD/init\_UASS.dup $PWD/duplicate\_db.sh $PWD/pre\_duplicate\_UASS.sh $PWD/post\_duplicate\_UASS.sh**

## Example files and scripts

|  |
| --- |
| **init\_UASS.dup** |
| SourceSid=PASS  SourceSidPwd=Ora-pass-6  #  DestSid=UASS  TMPDIR=/tmp  #  ORACLE\_HOME=/oracle/product/12.1  oracle\_admin=/oracle/admin |

|  |
| --- |
| **duplicate\_db.sh** |
| #!/usr/bin/ksh  this\_script=$0  (( $# != 1 )) && { print "Usage is $this\_script duplication\_parameter\_file"; exit; }  # Checks on parameter file  init\_dup\_file=$1  [[ ! -f $1 ]] && { print "$init\_dup\_file not found", exit; }  [[ ! -x $1 ]] && { print "$init\_dup\_file must be executable", exit; }  ########################## Main ###############################################  #DEBUG=echo  DEBUG=''  grace\_delay='5'  if [[ $DEBUG == '' ]]  then  print 'scrip runs in LIVE mode, 15 sec to abort'  elif [[ $DEBUG == 'echo' ]]  then  print 'script runs in DEBUG mode'  else  print "Mode $DEBUG inconnu, exit"  exit  fi  print "Grace delay of $grace\_delay secs to kill script"  sleep $grace\_delay  # execute parameter file to set duplication variables  . $init\_dup\_file  [[ ! -d $TMPDIR ]] && { print "Folder $TMPDIR not found, exit."; exit; }  # Folders and parameters linked to destination database  passwordfile="$ORACLE\_HOME/dbs/orapw${DestSid}"  spfile="$ORACLE\_HOME/dbs/spfile${DestSid}.ora"  # connection string  cnxsys='/ as sysdba'  # Parameters must not be null  [[ ${SourceSid:="KO"} == "KO" ]] && { print "SourceSid is null, exit."; exit; }  [[ ${SourceSidPwd:="KO"} == "KO" ]] && { print "SourceSidPwd is null, exit."; exit; }  [[ ${DestSid:="KO"} == "KO" ]] && { print "DestSid is null, exit."; exit; }  # Abort instance $DestSid if found  (( instanceUp = $(ps -ef | grep pmon\_$DestSid | grep -v grep | wc -l ) ))  if (( instanceUp == 1 ))  then  print "About to abort instance $DestSid"  # Abort auxiliary instance  if [[ $DEBUG == '' ]]  then  $ORACLE\_HOME/bin/sqlplus -s $cnxsys <<!  shutdown abort  exit  !  fi  fi  # Cleanup  print 'Files cleanup'  [[ -d $oracle\_oradata/$DestSid ]] && find $oracle\_oradata/$DestSid -type f -exec ls -l {} \;  [[ -d $oracle\_oradata/$DestSid ]] && { $DEBUG find $oracle\_oradata/$DestSid -type f -exec rm {} \;; print 'removed'; }  [[ -d $oracle\_flash\_recovery/$DestSid ]] && find $oracle\_flash\_recovery/$DestSid -type f -exec ls -l {} \;  [[ -d $oracle\_flash\_recovery/$DestSid ]] && { $DEBUG find $oracle\_flash\_recovery/$DestSid -type f -exec rm {} \;; print 'removed'; }  [[ -d $oracle\_admin/$DestSid ]] && find $oracle\_admin/$DestSid -type f -exec ls -l {} \;  [[ -d $oracle\_admin/$DestSid ]] && { $DEBUG find $oracle\_admin/$DestSid -type f -exec rm {} \;; print 'removed'; }  [[ -f $spfile ]] && { $DEBUG rm $spfile; print "$spfile removed"; }  # set parameters for auxiliary instance init.ora file  db\_name=$DestSid  audit\_file\_dest="$oracle\_admin/$DestSid/adump"  control\_file\_1='control01.ctl'  control\_file\_2='control02.ctl'  control\_file\_dir\_1="$oracle\_oradata/$DestSid/controlfile"  control\_file\_dir\_2="$oracle\_flash\_recovery/$DestSid/controlfile"  # Fully qualified control filenames  ctl\_file\_1="$control\_file\_dir\_1/$control\_file\_1"  ctl\_file\_2="$control\_file\_dir\_2/$control\_file\_2"  control\_files="$ctl\_file\_1,$ctl\_file\_2"  db\_create\_file\_dest=$oracle\_oradata  db\_recovery\_file\_dest=$oracle\_flash\_recovery  db\_recovery\_file\_dest\_size=$recovery\_file\_dest\_size  remote\_login\_passwordfile='EXCLUSIVE'  # process number, to name files uniquely  pid=$$  # Create auxiliary instance init.ora file  init\_ora\_file="$TMPDIR/init\_${DestSid}\_4\_dup\_${pid}.ora"  cat <<! > $init\_ora\_file  db\_name='$db\_name'  audit\_file\_dest='$audit\_file\_dest'  control\_files='$control\_files'  db\_create\_file\_dest='$db\_create\_file\_dest'  db\_recovery\_file\_dest='$db\_recovery\_file\_dest'  db\_recovery\_file\_dest\_size='$db\_recovery\_file\_dest\_size'  remote\_login\_passwordfile='EXCLUSIVE'  !  # Create RMAN command file  rman\_cmd\_file="$TMPDIR/duplicate\_${SourceSid}\_2\_${DestSid}\_pull\_${pid}.rman"  rman\_log\_file="$TMPDIR/duplicate\_${SourceSid}\_2\_${DestSid}\_pull\_${pid}.log"  cat <<! > $rman\_cmd\_file  connect target sys/$SourceSidPwd@$SourceSid  connect auxiliary sys/$SourceSidPwd@$DestSid  run{  allocate channel prmy1 type disk;  allocate channel prmy2 type disk;  allocate channel prmy3 type disk;  allocate channel prmy4 type disk;  allocate channel prmy5 type disk;  allocate channel prmy6 type disk;  allocate auxiliary channel aux1 type disk;  allocate auxiliary channel aux2 type disk;  allocate auxiliary channel aux3 type disk;  allocate auxiliary channel aux4 type disk;  allocate auxiliary channel aux5 type disk;  allocate auxiliary channel aux6 type disk;  DUPLICATE TARGET DATABASE TO '$DestSid'  FROM ACTIVE DATABASE  spfile  PARAMETER\_VALUE\_CONVERT  '$SourceSid','$DestSid'  SET DB\_FILE\_NAME\_CONVERT  '$SourceSid','$DestSid'  SET LOG\_FILE\_NAME\_CONVERT  '$SourceSid','$DestSid'  USING compressed BACKUPSET SECTION SIZE 512M  NOFILENAMECHECK;  }  !  # Checks  print "\nCheck"  print "SourceSid is $SourceSid"  print "DestSid is $DestSid"  print "db\_name is $db\_name"  print "passwordfile is $passwordfile"  print "spfile is $spfile"  print "audit\_file\_dest is $audit\_file\_dest"  print "ctl\_file\_1 is $ctl\_file\_1"  print "ctl\_file\_2 is $ctl\_file\_2"  print "control\_files is $control\_files"  print "db\_create\_file\_dest is $db\_create\_file\_dest"  print "db\_recovery\_file\_dest is $db\_recovery\_file\_dest"  print "db\_recovery\_file\_dest\_size is $db\_recovery\_file\_dest\_size"  print "init\_ora\_file is $init\_ora\_file"  print "rman\_cmd\_file is $rman\_cmd\_file"  print "\n$init\_ora\_file content"  cat $init\_ora\_file  print "\n$rman\_cmd\_file content"  cat $rman\_cmd\_file  # Create mandatory directories  [[ ! -d $audit\_file\_dest ]] && $DEBUG mkdir -p $audit\_file\_dest  [[ ! -d $control\_file\_dir\_1 ]] && $DEBUG mkdir -p $control\_file\_dir\_1  [[ ! -d $control\_file\_dir\_2 ]] && $DEBUG mkdir -p $control\_file\_dir\_2  [[ ! -d $db\_create\_file\_dest ]] && $DEBUG mkdir -p $db\_create\_file\_dest  [[ ! -d $db\_recovery\_file\_dest ]] && $DEBUG mkdir -p $db\_recovery\_file\_dest  # Create password file  $DEBUG orapwd file=$passwordfile password=$SourceSidPwd entries=10 force=y  [[ -f $passwordfile ]] && ls -l $passwordfile  if [[ $DEBUG == '' ]]  then  # Start auxiliary instance  $ORACLE\_HOME/bin/sqlplus -s $cnxsys <<!  startup nomount pfile='$init\_ora\_file';  exit  !  fi  sleep 10  # Duplicate database with rman  print "About to duplicate ...."  #$DEBUG $ORACLE\_HOME/bin/rman cmdfile $rman\_cmd\_file log=$rman\_log\_file |

|  |
| --- |
| **pre\_duplicate\_uass.sh** |
| #!/usr/bin/ksh  print "this is $0"  # This script is run before refreshing the UASS database  # UASS specific passwords are saved in a file  # Schema ASSUROL\_SEC is dumped  # Table assurol.t\_parametres is dumped  # Environment variables  export ORACLE\_SID=UASS  export ORACLE\_HOME=/oracle/product/12.1  # Connection string  connect='/ as sysdba'  # Save UASS db passwords, excludes usernames with string SYS  cat <<! | $ORACLE\_HOME/bin/sqlplus -s $connect  set lines 200  set pages 0  set trimspool on  set feedback off  set heading off  spool savepwdUASS.cmd  select  'alter user "'||username||'" identified by values '''||extract(xmltype(dbms\_metadata.get\_xml('USER',username)),'//USER\_T/PASSWORD/text()').getStringVal()||''';' old\_password  from dba\_users  order by user  /  spool off  !  # Prepare for datapump dumps  # Check on filesystem directory associated with Oracle Directory  [[ ! -d /oracle/oradata/dupdump ]] && mkdir -p /oracle/oradata/dupdump  # Create directory DUP\_PUMP\_DIR  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  drop directory DUP\_PUMP\_DIR;  create directory DUP\_PUMP\_DIR as '/oracle/oradata/dupdump';  grant read,write on directory DUP\_PUMP\_DIR to SYS;  !  # Dump schema ASSUROL\_SEC  $ORACLE\_HOME/bin/expdp \"$connect\" directory=DUP\_PUMP\_DIR \  dumpfile=UASS\_ASSUROL\_SEC.dmp \  LOGfile=UASS\_ASSUROL\_SEC.txt \  schemas=ASSUROL\_SEC \  reuse\_dumpfiles=y \  content=all  # Dump parameter table for schema assurol  $ORACLE\_HOME/bin/expdp \"$connect\" directory=DUP\_PUMP\_DIR \  dumpfile=UASS\_ASSUROL\_t\_parametres.dmp \  logfile=UASS\_ASSUROL\_t\_parametres.txt \  tables=assurol.t\_parametres \  reuse\_dumpfiles=y \  content=all |

|  |
| --- |
| **post\_duplicate\_uass.sh** |
| #!/usr/bin/ksh  print "this is $0"  # This script is run after refreshing the UASS database  # UASS specific passwords are recreated  # Schema ASSUROL\_SEC is reloaded  # Table assurol.t\_parametres is reloaded  # Database statistics are recomputed  export ORACLE\_SID=UASS  export ORACLE\_HOME=/oracle/product/12.1  connect='/ as sysdba'  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  shutdown immediate;  startup mount;  alter database noarchivelog;  alter database open;  !  # restore schemas passwords  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  @savepwdUASS.cmd  exit  !  # Create directory DUP\_PUMP\_DIR  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  drop directory DUP\_PUMP\_DIR;  create directory DUP\_PUMP\_DIR as '/oracle/oradata/dupdump';  grant read,write on directory DUP\_PUMP\_DIR to public  !  # Existing ASSUROL\_SEC schema is dropped and recreated with dump created before db duplication  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  set timing on  drop user ASSUROL\_SEC cascade;  exit;  !  $ORACLE\_HOME/bin/impdp \"$connect\" directory=DUP\_PUMP\_DIR \  dumpfile=UASS\_ASSUROL\_SEC.dmp \  LOGfile=impUASS\_ASSUROL\_SEC.txt \  schemas=ASSUROL\_SEC \  logtime=all  # Parameter table is reloaded  $ORACLE\_HOME/bin/impdp \"$connect\" directory=DUP\_PUMP\_DIR \  dumpfile=UASS\_ASSUROL\_t\_parametres.dmp \  logfile=impUASS\_ASSUROL\_t\_parametres.txt \  tables=assurol.t\_parametres \  TABLE\_EXISTS\_ACTION=truncate \  logtime=all  # Statistics are recomputed for the whole database  $ORACLE\_HOME/bin/sqlplus -s $connect <<!  set timing on  execute dbms\_stats.gather\_database\_stats(estimate\_percent => 100, degree=> 8, cascade=> true, options=>'GATHER AUTO');  exit;  ! |

# Crontab setup

To keep crontab job definition simple a script refresh\_db\_uass.sh launches refresh\_db.sh script with all the required parameters.

|  |
| --- |
| **Refresh\_db\_uass.sh** |
| #!/usr/bin/ksh  this\_script=$0  # This script launches refresh operations for uass databases  # to be performed before and after the database duplication  WRKDIR=/export/home/oracle/scripts\_dbdup  db=uass  cd $WRKDIR  date  time echo "./refress\_db.sh ./init\_${db}.dup ./duplicate\_db.sh ./pre\_duplicate\_${db}.sh ./post\_duplicate\_${db}.sh"  date |
| **Crontab entry for a refresh occurring on the 10/01/2019 at 15h45** |
| **45 15 10 1 \* /export/home/oracle/scripts\_dbdup/refresh\_db\_uass.sh > /export/home/oracle/scripts\_dbdup/refresh\_db\_uass.log 2>&1** |