

# Mémento des commandes DBA ORACLE

Par Benjamin LISAN – créé le 2 janvier 2000- mise à jour le 28 octobre 2005.

1	Introduction .....	1
2	Fichiers de configuration SQLNET V2.0 avec Oracle 7.3.4 .....	2
3	Augmenter la taille d'un datafile.....	2
4	Réparation d'une base en cas de la perte d'un fichier datafile.....	2
5	Réparation d'une base si fichier ".ctl" est cassé (en Oracle 7.3.4.).....	2
6	Réparation d'une base dont l'un des fichiers de journalisation « ...log ...rdo » a été supprimé par erreur (Oracle 7.3.4) .....	3
7	Réparation d'une base en cas de perte du fichier redolog ONLINE .....	3
8	Problème d'extension de tablespace : Ajout d'un fichier à un tablespace en Oracle 7.1.6 & 7.3.44	
9	Export d'une liste de tables.....	5
10	Export/import d'une base d'une machine vers une autre, en changeant le nom des tablespaces.....	5
11	Effacement par erreur du datafile avant de faire le drop de la tablespace associée .....	6
12	Exemples de commandes DBA Oracle .....	6
13	Exemple de type de démarrages et d'arrêt d'une base Oracle.....	7
14	En cas d'une erreur ORACLE mentionnant RESETLOGS lors d'un database open.....	7
15	Mettre une base Oracle en archive log .....	7
16	Pour faire une sauvegarde manuelle de type archive log .....	8
17	Voir des sessions (leurs locks ...) .....	8
18	Erreurs lors du lancement de la base (fichiers de contrôle perdu).....	8
19	Changement de place d'un datafile.....	8
20	Si erreur Oracle : log on line non disponible (ORA-00313 ou ORA-00322).....	8
21	Rétablir le mot de passe du login "system" .....	9
22	Création d'un fichier mot de passe.....	9
23	Suppression de la limite « maxextents » pour tables et indexes .....	9
24	Déplacer un datafile d'un répertoire à un autre .....	9
25	Commandes diverses .....	10
25.1	Script DOS pour transfert ftp à partir de machine NT .....	10
26	Contourner une limite de taille à la création d'un fichier d'export.....	10
27	Vérification de l'état de la base de donnée .....	11
28	Commande NT pour savoir quell process lock un fichier .....	11
29	Installation d'un patch ORACLE .....	11
30	Tests sql*net.....	11
31	Divers : mount sur AIX.....	12
32	Mot de passe .....	12
33	Chargement de données par SQL*LOADER .....	12
34	Arrêt / démarrage du « listener » Oracle .....	13
35	Analyse de la base .....	13
36	Support Oracle .....	13
37	recompilation d'un package.....	13
38	Arrêter / relancer un base bloquée grâce à la commande "sysresv" .....	14
39	LANGUE : NLS_LANG ... ..	14
40	Dimensionner un tablespace d'annulation (undo) en v9 .....	15
41	Affichage de toutes les informations sur les rollback segments en V8.....	17

## 1 Introduction

Ce document a été prévu pour les cas de perte de données (datafile), du redolog (online), d'un fichier de contrôle (ctl), d'un rollback segment, du tablespace system, fichier, d'altération de bloc de données ....

L'organisation de ce document est purement historique.

## 2 Fichiers de configuration SQLNET V2.0 avec Oracle 7.3.4

a) listener.ora

(il se trouve dans /XXXX/oracle/network/admin ou dans /XXXX/oracle/tnsadmin ).

Ce fichier liste les bases de données visibles de SQLNET.

b) tnsname.ora

Ce fichier liste les clients Oracle (pour SQLNET) (un nom de service = une base ???) ???.

Dès qu'on modifie un de ces fichiers, il faut relancer le démon listener SQLNET V2.

lsnrctl reload (ou bien "lsnrctl stop" puis "lsnrctl start").

## 3 Augmenter la taille d'un datafile

```
set pages 10
```

```
select * from v$tablespace;
```

```
select name, bytes , ts# from v$datafile;
```

(pour calculer la taille du "datafile"), ou bien :

```
select tablespace_name, file_name, bytes from dba_data_files where  
tablespace_name='EPT_TS_DATA1'
```

```
alter database datafile '/ora_data_tbl/epitdbo/epit_data001.dbf' resize 800 M ;
```

(alter tablespace USER\_DATA autoextend unlimited;) (???)

## 4 Réparation d'une base en cas de la perte d'un fichier datafile

Voici la procédure de restauration à appliquer :

- open database : STARTUP MOUNT;
- fichier offline : ALTER DATABASE DATAFILE OFFLINE;
- ALTER DATABASE OPEN;
- Restauration fichier : RECOVER DATAFILE; (après message : recover complete).
- Activer fichier de données : ALTER DATABASE DATAFILE ONLINE;

Note : existe aussi « recover tablespace » et « recover database », pour le cas de la restauration de plusieurs fichiers de données (à utiliser avec prudence).

## 5 Réparation d'une base si fichier ".ctl" est cassé (en Oracle 7.3.4.)

En cas de perte d'un fichier de contrôle, restauration possible :

- => si existe un miroir du fichier de contrôle perdu.
- => si existe une trace du fichier de CTL => ce qui permet de recréer le fichier de CTL.

Exemple de procédure de restauration :

Par exemple, si on a, par exemple les messages :

ORA-01110 : data file 1 : '/data\_3/.../system\_1.dbf'

ORA-01207 : file is more recent than control file.

on ne démarre que sur le 2° control file, en supprimant le 1er « control file », précisé dans le fichier « initXXX.ora ».

```
svrmgrl
```

```
connect internal ;
```

```
startup mount;
```

```
select * from v$log;
```

```

group sequence
67
65
66
select * from v$logfile;
/data_1/oracle_gret/gret_log_3_1.rdo
(note : ici avec cette commande, on voit s'afficher une liste de "logfiles" (liste de chemins _ paths _
absolus et fichiers unix).
select * from v$recover_file;
sequence# change#
select * from v$log_history;
sequence#

```

1er essai de "recover" de la base :

```

recover database until cancel using backup controlfile ;
(note : prompt affiché : <RET> = suggested | filename | AUTO | CANCEL )
On choisit alors "AUTO" (et non "OK").

```

2ème essai de "recover" de la base :

```

Au : prompt affiché : « <RET> = suggested ... », le réponds par le nom du fichier log, par exemple, par
le « filename » : /data_3/oracle_gret/gret_log_1_2.rdo
(note : si les messages suivants, ci-dessous, s'affichent, alors l'opération est réussie :
log appli
Media recovery completed ).
Alter database open resetlogs ;
(note : on doit alors voir s'afficher le message : « statement processed ).
Test de fonctionnement de la base :
Select * from dba_freespaces ;
Shutdown immediate ;
Copie du 2ème "control file" sur le 1er (sur celui qui était "vérolé").
Restauration du fichier « initXXX.ora », spécifiant les 2 fichiers « control file » employés.
Redémarrage de la base Oracle : startup open

```

## 6 Réparation d'une base dont l'un des fichiers de journalisation « ...log ...rdo » a été supprimé par erreur (Oracle 7.3.4)

```

svrmgrl
shutdown immediate ;
archive log liste ; (???)
alter system [disable | enable ] restricted session ;
recover database until cancel ;
alter database open resetlogs;
shutdown immediate ;
startup open;
Test fonctionnement de la base : select * from dual;

```

## 7 Réparation d'une base en cas de perte du fichier redolog ONLINE

=> procédure de restauration à appliquer :

- arrêt database : - shutdown
- suppression redolog : - rm ...
- relance et montage BD : - statup mount
- restauration de la BD : - recover database (archive fic. Log)
- ouvrir BD avec recreation de log : - alter database open resetlogs

- sauvegarde physique de la base arrêtée.

## 8 Problème d'extension de tablespace : Ajout d'un fichier à un tablespace en Oracle 7.1.6 & 7.3.4

```
svrmgrl
connect internal
alter tablespace TEMP_1 add datafile 'xxx/MIDV_temp_1.dbf' size 20M ;
alter tablespace TEMP_1 default storage (initial 1M next 1M pctincrease 0);
select * from dba_tablespaces;
select tablespace_name, filename, bytes from dba_data_files ;
```

Ajout d'un "datafile" à un tablespace existant :

```
Svrmgrl
connect / as sysdba ;
ALTER TABLESPACE nom_tablespace ADD DATAFILE 'xxx.dbf' SIZE 20M ;
ALTER TABLESPACE TEMP_1 DEFAULT STORAGE (INITIAL 1M NEXT 1M PCTINCREASE 0);
SELECT * FROM DBA_TABLESPACE ;
```

Recherche de doublons dans une table :

```
Select col from table having(col) > 1 group by col ;
```

=> Exemple d'options de l'import d'Oracle :

```
imp system/manager file=expdat.dmp destroy=y full=y ignore=y log=expdat.log
```

=> Exemple d'options de l'export d'Oracle :

(exemple du contenu d'un script d'export : export\_db ).

```
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
export ORACLE_HOME=/oracle/9.2.0
export PATH=$ORACLE_HOME/bin:/usr/bin:/etc:/usr/sbin:/usr/ucb:/sgbd/oracle/bin:/
usr/bin/X11:/sbin:
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export TNS_ADMIN=/oracle/9.2.0/network/admin
export ORACLE_SID=PILOAP
exp system/manager full=yes compress=yes file=/ora_data_tbl/piloap/exppilo.dmp
log=/util/admindb/pilo/ap/expilo.log statistics=none grants=yes indexes=yes
```

Sur NT, script exp\_PILDBO.BAT dans c:\UTIL\ADMINDB\PIL\OPER :

```
REM
set ORACLE_SID
dir E:\data\pildboPROD.dmp
pause
set ORACLE_SID=PILO
set ORACLE_SID
pause
c:\orant\bin\exp73.exe system/manager full=yes compress=yes file=E:\data\pildboPROD.dmp
log=C:\ORANT\expildboPROD.log statistics=none grants=yes indexes=yes
dir E:\data\pildboPROD.dmp
pause
```

=> Exemple de fichier 'parfile' pour l'import d'Oracle :

login/passe  
indexfile=creix.sql  
file=expdat.dmp  
(full=Y)  
(from use=c)  
(log=imp.log)

Exemple :  
c/c  
indexfile=ix.sql  
file=expC.dmp  
(full=Y)  
(from use=c)  
(log=imp.log)

La commande d'import sera par exemple : `imp sumeq=sumeq parfile=parfile`

Note : pour l'export Oracle (exp), on peut mettre l'option « `compress=N` ».

Faire un export/import d'une base sur une machine vers une autre base sur une autre machine (avec des noms de tablespaces différents).

Note : pour la volumétrie : il y a le logiciel Oracle : `oraflash2` (à partir du 'dba\_menu' ???).

Opérations sur la machine source :

faire un full export (de l'utilisateur administrateur).  
Par exemple : `exp c/c [full=y]` (export normal).

Opérations sur la machine cible :

créer le fichier « parfile » suivant :

login/passe  
indexfile=creix.sql  
file=expdat.dmp  
(full=Y)  
(from use=c)  
(log=imp.log) Exemple  
:c/c  
indexfile=ix.sql  
file=expC.dmp  
(full=Y)  
(from use=c)  
(log=imp.log)

faire un « `imp c/c` » normal (full). On importe les tables/données dans les tablespaces par défaut de l'utilisateur de l'import.

On génère un script de création d'index (avec la commande suivante et le parfile précédent) :

`imp c/c parfile=parfile`

Note : Il ne crée pas les tables mais crée le script : « `creix.sql` »

On modifie dans ce script « `creix.sql` », le nom du tablespace des index par celui qui est le « bon » (par exemple « `CAPA_1` » par « `C_2` »).

Note : Attention, ôter, dans ce script, l'ordre : « `connect c` ».

On lance ce script modifié : `sqlplus c/c @creix.sql`

## 9 Export d'une liste de tables

```
script "export_T_RAPPORT.sh" :  
export ORACLE_SID=TTTDBT  
exp system/manager file=/ora_arch_log/tttdbt/exp_TTT_T_RAPPORT.dmp indexes=y constraints=no  
grants=no tables= TTTDBA.T_RAPPORT,TTTDBA.T_RAPPORT_FOURN,  
TTTDBA.T_RAPPORT_PALIER log=log/export_TTT_T_RAPPORT.log
```

## 10 Export/import d'une base d'une machine vers une autre, en changeant le nom des tablespaces

a) machine source :

a.1) lancer le script `export_chmdbaw` contenant les lignes suivantes :

```
#!/bin/ksh
echo "init variables oracle"
export ORACLE_HOME=/sgbd/oracle/8.1.7
export PATH=/sgbd/oracle/8.1.7/bin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
export ORACLE_SID=CHMDBO
echo "etape 1 : export no row : export schema base"
exp chmdbaw/chmdbaw file=/ora_arch_log/exp_chmdbaw_norow.dmp rows=no
log=exp_chmdbaw_norow.log

echo "etape 2 : export indexfile (pour créer un script de creation d'indexes)"

imp chmdbaw/chmdbaw file=/ora_arch_log/exp_chmdbaw_norow.dmp indexfile=/ora_arch_log/
exp_chmdbaw_ixfile.sql log=exp_chmdbaw_ixfile.log
```

b) transférer par ftp , les 3 fichiers : exp\_chmdbaw\_ixfile.sql et exp\_chmdbaw\_norow.dmp

c) machine cible :

c1.) Modifier les noms des tablespaces (sources par ceux cible) dans le fichier :  
exp\_chmdbaw\_ixfile.sql  
Lancer l'exécution de ce script sql : sqlplus system/Mdp @exp\_chmdbaw\_ixfile.sql

c.2) Lancer le script import\_chmdbaw contenant les lignes suivantes :

```
# maj par B. LISAN 20 oct 05
export ORACLE_SID=CHMDBO
export NLS_LANG=french_france.WE8ISO8859P15

imp chmdbaw/chmdbaw file=/ora_data_tbl/chmdbo/exp_chmdbaw_norow.dmp fromuser=CHMDBAW
touser=CHMDBAW grants=y indexes=y commit=y ignore=y log=imp_chmdba.log
#imp system/xxx file=/ora_data_tbl/chmdbo/exp_chmdbaw_norow.dmp fromuser=chmdbaw
touser=chmdbaw ignore=y
```

## 11 Effacement par erreur du datafile avant de faire le drop de la tablespace associée

Message d'erreur : ORA-07368 : ...  
(note pas d'entrée pour cette erreur avec « oerr ora 07368 »).

```
Copier la tablespace (si possible)
sqlplus system/manager
alter tablespace TOTO offline;
Si OK, faire alors : alter database datafile '/xxx/datafilex.dbf' offline drop ;
Puis : drop tablespace C1 including contents ;
```

Une règle : en général, toujours lancer les commandes oracle avant les commandes unix.

## 12 Exemples de commandes DBA Oracle

```
Select * from v$database ;
Select sum(blocks), owner from dba_segments group by owner ;
Alter table TOTO deallocate unused ;
Select sum(blocks) from dba_data_files ;
Select sum(blocks), tablespace_name from dba_free_spaces group by tablespace_name ;
```

```
Set page 0
Select 'alter table SUMEQ.'||[[table_name|index_name]]| deallocate unused keep 10k ;' from
[dba_tables|dba_indexes] where owner = 'SUMEQ';
Select count(*) from dba_indexes where table_owner='SUMEQ';
Truncate table TABLE1 ; (???)
```

**DANGER !** : drop tablespace TABSPACE1 including contents ;

```
Alter user SUMEQ default tablespace TABSPACE2;
```

### 13 Exemple de type de démarrages et d'arrêt d'une base Oracle

Lancements d'une base :

```
Startup mount ;
Startup open ;
Alter database open ;
```

Mode restreint : startup restrict ;  
Alter system disable restricted session ;

Arrêt d'une base :

```
Shutdown ;
Shutdown immediate ;
Shutdown abort ;
```

### 14 En cas d'une erreur ORACLE mentionnant RESETLOGS lors d'un database open

Message d'erreur :  
ORA-01589 : must user RESETLOGS or NORESETLOGS option for database open

Note : diagnostic pas "bon".

```
svrmgrl
connect internal ;
alter database open [resetlogs | noresetlogs ] ;
```

### 15 Mettre une base Oracle en archive log

```
shutdown immediate ;
faire une sauvegarde à froid de la base
modifier le fichier « initXXX.ora » et y mettre :
Log_archive_start=true
Relancer "svrmgrl".
Se connecter avec l'instance sysdba : connect a/a as sysdba ;
Startup mount ;
Alter database archive log ;
Alter database open ;
"archive log list;" affiche alors "mode: archive automatical archival enable".
Il faut sauvegarder le "control file" :
Alter database backup controlfile to trace ;
Notes : . On doit voir apparaître le répertoire « userdumpdest ».
. L'archive est au format « control file ».
```

Startup mount ;  
Alter system archive log start ;

## 16 Pour faire une sauvegarde manuelle de type archive log

Alter tablespace begin backup tablespace XXX ;  
( alter tablespace unbackup tablespace XXX ; ).

## 17 Voir des sessions (leurs locks ...)

Select sid, username from v\$sessions ;  
Select \* from v\$process;

## 18 Erreurs lors du lancement de la base (fichiers de contrôle perdu)

Messages d'erreur : ORA-01157 : cannot identify datafile 14 - file not found  
ORA-01110 : '/data\_26/oracle\_sumeq/sumeq\_sumeq\_3\_3.dbf'

```
select * from v$recover_file ;
FILE    ON LINE      ERROR
14      ON LINE      FILE NOT FOUND
select name from v$datafile ;
alter database datafile '/data_26/oracle_sumeq/sumeq_sumeq_3_3.dbf' offline drop ;
alter database open ;
select * from dba_data_files;
drop tablespace TEMP_1 ;
Message d'erreur : « not empty ».
Select * from dba_segments where tablespace_name = 'TEMP_1' ;
On trouve 2 tables : PRODUCT_FILE et USER_FILE
DANGER ! : drop tablespace TEMP_1 including contents ;
Create tablespace TEMP_1 ... ;
```

## 19 Changement de place d'un datafile

Select file\_name from dba\_data\_files ; (ou "from v\$logfile" ou « from v\$database" ).  
arrêt de la base (shutdown immediate ?).  
cp -p /partition\_de\_départ/FIC.dbf /partition\_cible  
svrmgrl  
connect internal ;  
startup mount ;  
select \* from v\$datafile ;  
alter database rename file 'ancien\_datafile' to 'nouveau\_datafile';  
(alter database open ; )  
shutdown immediate ;  
startup mount ; (et start DB Oracle) ;  
select \* from v\$datafile ;  
Eventuellement, modifier le chemin du "control file" dans le fichier « iniXXX.ora ».

## 20 Si erreur Oracle : log on line non disponible (ORA-00313 ou ORA-00322)

Messages d'erreurs : ORA-00313 : open failed for members of log group %s of thread %s  
ORA-00322 : log %s of thread %s is not current copy



a) Erreur ORA-00313 :

Cause: La « log online » ne peut pas être ouverte. La base semble incapable de trouver le fichier.

Action: Voir les erreurs accompagnant le message d'erreur et faire en sorte que la log soit de nouveau disponible.

b) Erreur ORA-00322 :

Cause: La vérification de l'entête (header) de la "log file" à l'ouverture de la database (« open database ») montre que la « online log » est un « backup » restauré incorrect.

Action: Restaurer les correct "file" ou "reset logs" corrects.

```
connect internal ;
startup mount ;
alter database open resetlogs;
shutdown immediate ;
startup pmount=initXXX.ora
```

## 21 Rétablir le mot de passe du login "system"

```
sqlplus /nolog
connect / as sysdba
alter user system identified by manager ;
```

## 22 Création d'un fichier mot de passe

Mot de passe ayant été modifié par la commande :

```
orapwd file=/oracle/9.2.0/dbs/orapwPILOdbo password=Lfst2f$lcm#eaX entries=5
```

## 23 Suppression de la limite « maxextents » pour tables et indexes

Il faut d'abord trouver le propriétaire ([owner]) à qui appartient l'objet (table, index ...).

Par exemple :

```
select owner, index_name from dba_indexes where index_name like '%I_COL_SESSIONS%';
select owner, table_name from dba_tables where table_name = 'TTDPUR051950';
```

Alter [table|index] [ [owner.]nom\_table | [owner.]nom\_index ] storage ( maxextents unlimited ) ;

Exemple : alter table BAAN.TTDPUR051950 storage ( maxextents unlimited );

## 24 Déplacer un datafile d'un répertoire à un autre

Actions à faire pour renommer un fichier qui n'est pas dans le bon répertoire (avec copie du fichier dans le bon répertoire cible).

1.Mettre le tablespace offline en utilisant la commande ALTER TABLESPACE avec la clause OFFLINE :

```
ALTER TABLESPACE accounting OFFLINE NORMAL;
```

2.Copier le fichier de 'diska:pay1.dat' vers 'diskb:receive1.dat'.

3.Renommer le datafile en utilisant ALTER TABLESPACE avec la clause RENAME DATAFILE :

```
ALTER TABLESPACE accounting
RENAME DATAFILE 'diska:pay1.dbf'
```

```
TO 'diskb:receive1.dbf';
```

4. Remettre la tablespace en utilisant ALTER TABLESPACE avec la clause ONLINE :

```
ALTER TABLESPACE accounting ONLINE;
```

## 25 Commandes diverses

### 25.1 Script DOS pour transfert ftp à partir de machine NT

```
REM copie FTP de NT a UNIX
echo open h443111m > c:\util\system\put_stat.cmd
echo user oracle machin >> c:\util\system\put_stat.cmd
echo cd /tmp >> c:\util\system\put_stat.cmd
echo put e:\data\pildboPROD.dmp >> c:\util\system\put_stat.cmd
echo bye >> c:\util\system\put_stat.cmd
cat c:\util\system\put_stat.cmd
pause
ftp -n < c:\util\system\put_stat.cmd
del c:\util\system\put_stat.cmd
pause
```

## 26 Contourner une limite de taille à la création d'un fichier d'export

Exemple de script shell à lancer :

```
#!/bin/ksh
# script shell : expSYMDBO.sh
# export full de la base SYMDBO (Symbad ope).
#

export nopipe=410 # ce numero de pipe est pris comme exemple
# creation des fichiers pipes nommes
# mknod /dev/pipe_${nopipe} p
# mknod /dev/compress_${nopipe} p

export DUMPDIR=/work_oracle/backup
export TAILLE_SPLIT=1000
export ORACLE_SID=SYMDBO

# === Test de l'existence des pipes
if [ ! -p /dev/pipe_${nopipe} -o ! -p /dev/compress_${nopipe} ]
then
    echo "\nATTENTION, au moins un des pipes nommes n'existe pas !!\n"
    exit 1
fi

nohup split -b${TAILLE_SPLIT}m /dev/pipe_${nopipe} $DUMPDIR/exp${ORACLE_SID}.dmp &

# === compress recupere les donnees du pipe compress_pipe_${nopipe},les compresse
# === et renvoie le tout dans le pipe split_pipe_${nopipe}

nohup compress < /dev/compress_${nopipe} > /dev/pipe_${nopipe} &

exp system/<MdP> file=/dev/compress_${nopipe} full=y compress=y constraints=y indexes=y
log=$DUMPDIR/exp${ORACLE_SID}.log
```

Note : Pour créer un fichier pipe nommé : `mknod /dev/nom_pipe p`

## 27 Vérification de l'état de la base de donnée

```
select trigger_name, trigger_type, owner, status from dba_triggers where status = 'DISABLED';
select object_name, object_type, owner, status from dba_objects where status = 'INVALID';
```

## 28 Commande NT pour savoir quell process lock un fichier

HANDLE.EXE (disponible sur le site : SYSINTERNAL).

## 29 Installation d'un patch ORACLE

1) arrêter toutes les bases oracle :

a) `export ORACLE_SID=<sid_base>`

b) `svrmgrl` (ou `sqlplus /nolog`).

c) `connect internal`

d) `shutdown immediate`

[ e) `shutdown abort` (si le « `shutdown immediate` » d'avant ne fonctionne pas) ]

[ f) faire le `kill [-9]` le reste des derniers process oracle restant en mémoire ]

g) lancer `dtterm` ou `xterm` (en Xwindows / X11 sous un émulateur X\_ par ex. Humingbird ... \_ ou sur une station graphique Unix).

(`xterm ls -display @d`

`aixterm -ls -sb -sl 1000 -display @d -T 'nom_du_serveur' -fn 6x13 -fg black -bg white` ).

h) `export DISPLAY=adr_IP_de_son_PC :0`

i) `mount /cdrom` (être `root`).

[ j) `rootpre.sh` ]

k) `./runIntaller`

[ k2) éventuellement, modifier "l'init<sid>.ora" ]

l) `svrmgrl` (ou `sqlplus /nolog`).

m) `connect internal`

n) `startup` (doit faire le `mount` et l'`open` dans la base).

o) vérification par : `select * from dba_users ;`

p) lancer `@chemin_ORACLE_HOME/catalog`

q) lancer `@chemin_ORACLE_HOME/catproc`

r) `shutdown immediate`

[ s) éventuellement, modifier "l'init<sid>.ora" ]

t) `startup`

## 30 Tests sql\*net

a) Test listener : `lsnrctl status`

b) Test TNS : `tnsping [ nom_machine | nom_listener_base ]`

### 31 Divers : mount sur AIX

Note : mount sur AIX

```
mount -rv cdrfs /dev/cd0 /cdrom
```

### 32 Mot de passe

```
orapwd file=oraNOCIDBO password=Lfst2flcmeap entries=5
```

Et placer, dans le fichier "init<sid>.ora" :  
remote\_login\_passwordfile=EXCLUSIVE

```
select username , LAST_CALL_ET , PROGRAM from sys.v_$session;
```

### 33 Chargement de données par SQL\*LOADER

a) script "load\_T\_Rapport.sh" :

```
export ORACLE_SID=TTTDBT
```

```
sqlldr userid=system/manager control=load_T_Rapport.ctl log=log/load_T_Rapport.log
```

b) script load\_T\_Rapport.ctl :

```
-- *****
-- *          load_T_Rapport.ctl          *
-- *          Creation controle SQLLOADER de table la table T_RAPPORT  *
-- *****
--
-- *****transfert de donnees sqlserver
--
--
NUMFOUR;NUMFOURNEE;DATES;DATEDEBUTTRAIT;HEUREDEBUTTRAIT;DATEFINTRAIT;HEUREFINTRAIT;USERNAME;USERDESC;TOTALTAP;COMMENTAIRESFAB;NOMFAB;USERFAB;VALIDFAB;COMMENTAIRESQUALITE;NOMQUALITE;USERQUALITE;VALIDQUALITE;TTHPILOTE;IDFOURNEE
-- F31;1;08/10/2005 00:00;08/10/2005 00:00;01/01/1900 16:30;10/10/2005
00:00;01/01/1900
16:30;S086508;MACHTELINCKX;100,8;RAS;TROUSSELIER;S086972;1;;;1;00464-09-
11-6376;F31-2005/10/08-1
--
options (rows=15000,errors=9999)
load data
infile 'T_Rapport.csv'
-- SKIP 1
badfile 'log/t_rapport.bad'
discardfile 'log/t_rapport.dsc'
discardmax 999
replace
-- preserve blanks
into table TTTGENIO.T_RAPPORT
fields terminated by ";" optionally enclosed by '"'
TRAILING NULLCOLS
(
  NUMFOUR
  NUMFOURNEE
  DATES          DATE 'DD/MM/YYYY HH24:MI' ,
  DATEDEBUTTRAIT DATE 'DD/MM/YYYY HH24:MI' ,
  HEUREDEBUTTRAIT DATE 'DD/MM/YYYY HH24:MI' ,
  DATEFINTRAIT   DATE 'DD/MM/YYYY HH24:MI' ,
```

```

HEUREFINTRAIT      DATE 'DD/MM/YYYY HH24:MI' ,
USERNAME           ,
USERDESC           ,
TOTALTAP           INTEGER EXTERNAL ,
-- TOTALTAP DECIMAL(7,2) EXTERNAL ,
COMMENTAIRESFAB    ,
NOMFAB             ,
USERFAB            ,
VALIDFAB           ,
COMMENTAIRESQUALITE ,
NOMQUALITE         ,
USERQUALITE        ,
VALIDQUALITE       ,
TTHPILOTE         ,
IDFOURNEE         ,
)

```

### 34 Arrêt / démarrage du « listener » Oracle

```

./lsnrctl start
./lsnrctl stop

```

### 35 Analyse de la base

1) En V9, pour un propriétaire d'un schéma de la base :

```
exec DBMS_STATS.GATHER_SCHEMA_STATS('DATAMART',degree=> 20,cascade=>true);
```

2) En V8 et V9, pour une table donnée :

```
analyze nom_table compute statistics ;
```

### 36 Support Oracle

En cas d'un gros problème ORACLE, utiliser / consulter :

- a) la base [www.metalink.oracle.com](http://www.metalink.oracle.com)
- b) l'assistance de M. Rui CORREIA : [rui.correia@oracle.com](mailto:rui.correia@oracle.com) et 05.62.72.97.88, 06.07.15.65.09, fax : 05.62.72.97.72

### 37 recompilation d'un package

```
alter package <owner>.<package> compile [body] ;
alter view <view> compile [body] ; (vue compilée).
```

Exemples :

```
alter package DBMS_AQ_IMPORT_INTERNAL compile body ;
alter package ecoldba.pkg_ecol_master compile ;
```

### 38 Arrêter / relancer un base bloquée grâce à la commande "sysresv"

1) Avec la commande oracle « shutdown immediate » ou « shutdown abort » on ne peut ni arrêter, ni relancer une base. Par exemple, on a le message :

- a) ORA-27100: shared memory realm already exists  
[IBM AIX RISC System/6000 Error: 2: No such file or directory ]
- b) ORA-01033: ORACLE initialization or shutdown in progress
- c) ORA-00445: background process "PMON" did not start after 120 seconds
- d) ORA-27120: unable to removed shared memory segment  
IBM AIX RISC System/6000 Error: 22: Invalid argument

Solution :

1) rechercher la mémoire IPC (shared memory, semaphore ...) qui n'a pas été arrêtée par la base, grâce à la commande Oracle « sysresv », de la façon suivante :  
\$ export ORACLE\_SID=PLOBJ # <= choisir ici l'instance Oracle à supprimer, ici l'instance « PLOBJ ».  
\$ **sysresv** # lancer la commande « sysresv »

IPC Resources for ORACLE\_SID "PLOBJ" : # <= affiche les infos des mémoires IPC à supprimer.

Shared Memory:

ID	KEY
103	0x53bc73d4

Semaphores:

ID	KEY
196610	0x72c86d78

Oracle Instance alive for sid "PLOBJ"

\$ ipcrm -m 103 # <= ensuite, on supprime les mémoires IPC en question  
\$ ipcrm -s 196610

Puis on supprime les processus de la base Oracle :

```
ps -ef | grep PLOBJ [ grep -v grep
oracle 23808 1 0 07:20:03 - 0:00 ora_pmon_PLOBJ
oracle 31996 1 0 07:20:03 - 0:00 ora_reco_PLOBJ
oracle 32000 1 0 07:20:03 - 0:00 ora_smon_PLOBJ
oracle 32258 1 0 07:20:03 - 0:00 ora_ckpt_PLOBJ
oracle 35394 1 0 07:20:03 - 0:00 ora_dbw0_PLOBJ
oracle 35758 1 0 07:20:03 - 0:00 ora_lgwr_PLOBJ
oracle 67842 1 0 13:19:37 - 0:00 oraclePLOBJ (LOCAL=NO)
oracle 71990 1 0 13:40:31 - 0:00 oraclePLOBJ (LOCAL=NO)
```

et on tue tous ces processus de la base "PLOBJ" : kill -9 23808 31996 32000 32258 35394 35758 ...

### 39 LANGUE : NLS\_LANG ...

```
select value from v$nls_parameters where parameter='NLS_LANGUAGE' or
parameter='NLS_TERRITORY' or parameter='NLS_CHARACTERSET';
```

La réponse à cette requête comporte 3 lignes

Il faut rajouter au niveau du script d'export, le paramètre suivant:

**export NLS\_LANG=value1\_value2.value3**

## 40 Dimensionner un tablespace d'annulation (undo) en V9

```
# -----
# script shell : cal_taille_undo.sh
# -----
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
sqlplus system/manager @cal_taille_undo.sql

# -----
# script SQL : cal_taille_undo.sql
# ce script ne peut fonctionner qu'avec un NLS_LANG américain
# -----
col nb_blk_undo_per_sec new_val nb_blk_undo
col val_db_blk          new_val val_db_blk_size
col undo_ret_val       new_val undo_ret_val2

SELECT SUM(undoblks) / SUM ( (end_time - begin_time) * 86400) nb_blk_undo_per_sec
FROM v$undostat;

prompt "nb_blk_undo='&nb_blk_undo'"

select name, value val_db_blk  from v$parameter where name = 'db_block_size';

prompt "val_db_blk_size='&val_db_blk_size'"

select name, value undo_ret_val from v$parameter where name = 'undo_retention';

prompt "undo_ret_val2='&undo_ret_val2'"

select  (&undo_ret_val2 * &nb_blk_undo * &val_db_blk_size + &val_db_blk_size *
24) / 1048576 Bytes from dual;

[ou bien :

SELECT (UR * (UPS * DBS)) + (DBS * 24) Bytes
FROM (SELECT value AS UR
FROM v$parameter
WHERE name = 'undo_retention'),
(SELECT SUM(undoblks) / SUM ((end_time - begin_time) * 86400 ) AS UPS
FROM v$undostat ),
(SELECT value AS DBS
FROM v$parameter
WHERE name = 'db_block_size');
]
```

## 41 Suppression d'un tablespace UNDO décrite dans le cours Oracle V9

Pour supprimer un tablespace UNDO (rollback), il faudrait suivre la procédure suivante :

1) Créer un second tablespace UNDO, par exemple : **TBSGENIO2**

```
CREATE UNDO TABLESPACE TBSGENIO2
DATAFILE '/ora_data_tbl/geniodbo/rbsGENIO2.ora' SIZE nnnM ;
[ ou /genio_RBS2/geniodbo/rbsGENIO.ora ]
```

La taille "nnn" de ce table space est obtenue par le script suivant (fonctionnant en V9) :

```
SELECT (UR * (UPS * DBS)) + (DBS * 24) Bytes
FROM (SELECT value AS UR
FROM v$parameter
WHERE name = 'undo_retention'),
```

```
(SELECT SUM(undoblks) / SUM ((end_time - begin_time) * 86400 ) AS UPS
FROM v$undostat ),
(SELECT value AS DBS
FROM v$parameter
WHERE name = 'db_block_size');
```

2) il faut ensuite rendre actif le 2° tablespace UNDO (donc basculer dessus) :

```
ALTER SYSTEM SET undo_tablespace = TBSGENIO2 ;
```

3) lancer :

```
SELECT a.name, b.status
FROM v$rollname a, v$rollstat b
WHERE a.name IN (SELECT segment_name
FROM dba_segments )
AND a.usn = b.usn ;
```

Si on a quelque chose comme :

```
NAME                STATUS
-----
_SYSSMU4$          PENDING OFFLINE
```

Un segment d'annulation, possédant le statut « PENDING OFFLINE » a encore des transactions en cours.

*Donc, on ne peut pas faire cette opération.*

*Sinon, on peut continuer à faire cette suppression.*

4) Si le SELECT précédent ne ramène aucune ligne, on peut alors lancer (1) la suppression du 1er tablespace :

```
DROP TABLESPACE UNDOTBS1 ;
```

Note : Dans certains cas, il faudrait ensuite faire l'opération inverse, pour avoir [de nouveau] le tablespace de départ (ici RBSGENIO), mais avec un datafile dans /rbs\_GENIO/geniodbo ... d'une taille plus petite.

## Création ou modification d'un rollback segment en V8

1) créer le rollback segment :

```
CREATE PUBLIC ROLLBACK SEGMENT rb1 TABLESPACE tablespace ;
```

2) mettre le rollback segment ONLINE :

mettre le paramètre ROLLBACK\_SEGMENTS dans le fichier PFILE, comme ci-après :  
 ROLLBACK\_SEGMENTS = (RB1, RB2) ;

3) spécifier les paramètres du « rollback segment » avec O.E.M. ou avec l'ordre SQL :

```
CREATE PUBLIC ROLLBACK SEGMENT data1_rs
TABLESPACE tablespace (INITIAL 50K NEXT 50K OPTIMAL 750K MINEXTENTS 15
MAXEXTENTS 100) ;
```

4) changer le paramétrage du « rollback segment » avec O.E.M. ou avec l'ordre SQL :

```
ALTER PUBLIC ROLLBACK SEGMENT data1_rs STORAGE (MAXEXTENTS 120) ;
```

5) Ajuster (décroître) la taille du « rollback segment » :

```
ALTER ROLLBACK SEGMENT rbs1 SHRINK TO 100K ;
```

6) Mettre le « rollback segment » OFFLINE ou ONLINE :



```
ALTER ROLLBACK SEGMENT rbs1 OFFLINE ;
```

7) Assigner une transaction à un « rollback segment » particulier :  
SET TRANSACTION USE ROLLBACK SEGMENT rbs2 ;

## 42 Suppression d'un rollback segment en V8

On peut supprimer un rollback segments quand les extents d'un segment deviennent trop fragmentés sur le disque, ou si le segment doit être délocalisés dans un tablespace différent.

Avant de supprimer un ROLLBACK SEGMENT, s'assurer que le status du rollback segment est OFFLINE.

Si le rollback segment, est actuellement ONLINE, PARTLY AVAILABLE, NEEDS RECOVERY ou INVALID, on ne peut le supprimer.

Si le status est INVALID, le segment a déjà été supprimé.

Pour supprimer le rollback segment, on doit avoir le privilège DROP ROLLBACK SEGMENT.

On peut le supprimer par la commande : DROP PUBLIC ROLLBACK SEGMENT RB1 ;  
(dans cette commande, il est important de préciser s'il est privé ou public).

Ensuite, il faut mettre à jour la ligne « ROLLBACK\_SEGMENTS » dans le fichier PFILE.

Après qu'il a été supprimé, son status change en 'INVALID'.

Il n'apparaîtra plus dans la vue DBA\_ROLLBACK\_SEGS ;

## 43 Affichage de toutes les informations sur les rollback segments en V8

### Toutes infos sur les rollback segments en V8

```
SELECT segment_name, tablespace_name, status
FROM sys.dba_rollback_segs;
set lines 132
col segment_name format a20
SELECT segment_name, tablespace_name, bytes, blocks, extents
FROM sys.dba_segments
WHERE segment_type = 'ROLLBACK' ;
```

### Rollback segments OFFLINE en V8

```
SELECT name, xacts "ACTIVES TRANSACTIONS"
FROM v$rollname, v$rollstat
WHERE status = 'PENDING OFFLINE'
AND v$rollname.usn = v$rollstat.usn ;
```

### Affichage de tous les rollback segments deferrés en v8

prompt affiche les RBS avec owners publics (PUBLIC) ou prives (SYS)  
prompt -----

```
SELECT segment_name, tablespace_name, owner
FROM sys.dba_rollback_segs;
```

prompt Displaying all deferred rollback segments  
prompt -----

```
SELECT segment_name, segment_type, tablespace_name
FROM sys.dba_segments
WHERE segment_type = 'DEFERRED ROLLBACK';
```

## 44 Resize d'un tablespace utilisé par les rollback en V8

0) Je fais une fiche de changement, pour programmer cette opération un vendredi soir (avec accord utilisateurs de GENIO).

Pour réaliser cette procédure : être connecté « SYSTEM » :

1) Ce vendredi soir 17h, je crée un second tablespace pour les rollback segments, par exemple : **RBSGENIO2**

```
CREATE TABLESPACE RBSGENIO2
DATAFILE '/ora_data_tbl/geniodbo/rbsGENIO2.ora' SIZE 200M REUSE
AUTOEXTEND ON NEXT 50M MAXSIZE 500M;
[ ou bien si Dominique Vrinat avait créé la partition "genio_RBS2" :
/genio_RBS2/geniodbo/rbsGENIO.ora ]
```

ou bien je crée ce tablespace **RBSGENIO2** par la commande :

```
create tablespace RBSGENIO2
    datafile '/ora_data_tbl/geniodbo/rbsGENIO2.ora' size 200M reuse
    default storage (initial 200M next 20M pctincrease 0 minextents 20);
```

2) Puis je crée 2 nouveaux rollback segments **RBS3** et **RBS4** :

```
-- Création de segments rollback
CREATE ROLLBACK SEGMENT RBS3
TABLESPACE (INITIAL 20M NEXT 20M [OPTIMAL 200M] MINEXTENTS 20 [MAXEXTENTS 40] )
tablespace RBSGENIO2;
CREATE ROLLBACK SEGMENT RBS4
TABLESPACE (INITIAL 20M NEXT 20M [OPTIMAL 200M] MINEXTENTS 20 [MAXEXTENTS 4] )
tablespace RBSGENIO2;
```

3) Je rends ces 2 ROLLBACK **RBS3** et **RBS4** "ONLINE" :

```
alter rollback segment RBS3 online;
alter rollback segment RBS4 online;
```

*Ensuite, mais peut-être que ce point 4) ci-après n'est pas nécessaire (?) => :*

4) je note les infos sur les ROLLBACK SEGMENTS **RBS1** et **RBS2** à supprimer et recréer :

```
SELECT segment_name, tablespace_name, status
FROM sys.dba_rollback_segs;
```

SEGMENT_NAME	TABLESPACE_NAME	STATUS
SYSTEM	SYSTEM	ONLINE
<b>RB1</b>	<b>RBSGENIO</b>	<b>ONLINE</b>
<b>RB2</b>	<b>RBSGENIO</b>	<b>ONLINE</b>

```
set lines 132
```

```
col segment_name format a20
```

```
SELECT segment_name, tablespace_name, bytes, blocks, extents
FROM sys.dba_segments WHERE segment_type = 'ROLLBACK' ;
```

SEGMENT_NAME	TABLESPACE_NAME	BYTES	BLOCKS	EXTENTS
--------------	-----------------	-------	--------	---------

SYSTEM	SYSTEM	409600	50	5
RB1	RBSGENIO	666558464	81367	5408
RB2	RBSGENIO	297746432	36346	2423

4bis) je vérifie qu'il n'y a pas de transaction actives, grâce à cet ordre SQL ci-après :

```
SELECT name, xacts "ACTIVES TRANSACTIONS"
FROM v$rollname, v$rollstat
WHERE status = 'PENDING OFFLINE'
AND v$rollname.usn = v$rollstat.usn ;
```

no rows selected

5) Je rends mes 2 rollback segments **RBS1** et **RBS2** "OFFLINE" :

```
alter rollback segment RBS1 offline;
alter rollback segment RBS2 offline;
```

6) je supprime ces 2 rollback segments **RBS1** et **RBS2** :

```
DROP ROLLBACK SEGMENT RB1 ;
DROP ROLLBACK SEGMENT RB2 ;
```

7) je mets OFFLINE et supprime la tablespace **RBSGENIO** :  
**ALTER TABLESPACE RBSGENIO OFFLINE NORMAL ;**  
**DROP TABLESPACE RBSGENIO INCLUDING CONTENTS;**

7bis) puis je supprime le datafile associé :  
**rm /genio\_RBS/geniodbo/rbsGENIO.ora**

8) Je recrée le 1er tablespace pour les Rollback **TBSGENIO** :

```
CREATE TABLESPACE TBSGENIO
DATAFILE '/genio_RBS/geniodbo/rbsGENIO.ora' SIZE 200M REUSE
AUTOEXTEND ON NEXT 50M MAXSIZE 500M;
```

ou par la commande :

```
create tablespace RBSGENIO
    datafile '/genio_RBS/geniodbo/rbsGENIO.ora' size 200M reuse
    default storage (initial 200M next 20M pctincrease 0 minextents 20);
```

9) Je recrée les 2 rollback segments **RBS1** et **RBS2** :

```
-- Création de segments rollback
CREATE ROLLBACK SEGMENT RBS1
TABLESPACE (INITIAL 20M NEXT 20M [OPTIMAL 200M] MINEXTENTS 20 [MAXEXTENTS 40] )
    tablespace RBSGENIO;
CREATE ROLLBACK SEGMENT RBS2
TABLESPACE (INITIAL 20M NEXT 20M [OPTIMAL 200M] MINEXTENTS 20 [MAXEXTENTS 4] )
    tablespace RBSGENIO;
```

10) Je rends ces 2 ROLLBACK **RBS1** et **RBS2** "ONLINE" :

```
alter rollback segment RBS1 online;
alter rollback segment RBS2 online;
```

11) Je rends les 2 rollback segments **RBS3** et **RBS4** "OFFLINE" :

```
alter rollback segment RBS3 offline;  
alter rollback segment RBS4 offline;
```

12) je supprime ces 2 rollback segments RBS3 et RBS4 :

```
DROP ROLLBACK SEGMENT RB3 ;  
DROP ROLLBACK SEGMENT RB4 ;
```

13) je mets OFFLINE et supprime la tablespace RBSGENIO2 :  
ALTER TABLESPACE RBSGENIO2 OFFLINE NORMAL ;  
DROP TABLESPACE RBSGENIO3 INCLUDING CONTENTS;

14bis) puis je supprime le datafile associé :

```
rm /ora_data_tbl/geniodbo/rbsGENIO2.ora
```

#### **45 Supprimer des doublons dans une table**

```
DELETE FROM EMP A  
WHERE EXISTS  
(SELECT * FROM EMP B WHERE B.EMPNO = A.EMPNO AND A.ROWID > B.ROWID) ;
```

#### **46 Infos sur l'archivage log**

Col name format a40

```
select name, FIRST_TIME, status, archived from v$archived_log;  
select NAME, ARCHIVE_CHANGE# , REMOTE_ARCHIVE from v$database;
```