

# RMAN voor snel transport

## Methode voor vullen datawarehouse

**Het vullen van datawarehouses vindt in de meeste gevallen logischerwijs plaats met behulp van ETL-tools. Het kan echter interessant of zelfs noodzakelijk zijn een staging-area of datawarehouse op een andere manier van source-data te voorzien. In dit artikel laat René Kundersma aan de hand van een voorbeeld laten zien dat één van die andere methoden sinds 10G ook steeds gemakkelijker een RMAN-methode kan zijn.**

Tablespaces worden daarbij uit de backup van een productiedatabase gerestored en gerecoverd met behulp van de zogenaamde 'Tablespace Point In Time Recovery' (TSPITR). Vervolgens kan met de transportable tablespace-optie de tablespace beschikbaar worden gemaakt binnen een andere database. Het voordeel van het fysiek verplaatsen van datafiles vanuit een back-up is dat je zonder SQL-overhead en dus zonder de source-database te belasten de data naar de target krijgt. Er zijn vanuit deze situatie vervolgens verschillende manieren waarop de transportable tablespaces 'gekoppeld' kunnen worden op aan de target-database. In dit voorbeeld gebruik ik daarvoor 'impdp' (datapump) vanuit de shell maar ook is het mogelijk een datapump import-script vanuit SQL\*Plus aan te roepen. Aan de hand van een concreet voorbeeld wil ik de uitgebreide stappen beschrijven die deze exercitie behelst bij het verplaatsen van een tabel. We gaan uit van een eenvoudige tabel die tot en met de laatste rij op de target-database beschikbaar moet worden gesteld. De helft van de tabel zit in de back-up, de andere helft niet, die staat als archived-redolog op disk. Houdt bij transportable tablespace acties altijd rekening met de restricties die de transportable optie kent. Eén daarvan is bijvoorbeeld dat objecten in een te transporteren tablespace nooit van 'SYS' mogen zijn.

### Aanmaken test-tabel op source database: DB1020

```
$ sqlplus rk/topsecret@DB1020

SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 12 11:06:11 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create table big_tab tablespace TRANSPORT_TBS
2 select * from dba_objects;

Table created.

SQL> select count(*) from big_tab;

COUNT(*)
-----
11269
```

### Create RMAN-catalog

Uitgangspunt is dat de database DB1020 in archived redolog mode draait. In een tweede database (DB920) wordt een catalog gemaakt ten bate van de RMAN back-up van DB1020. Er wordt begonnen met het aanmaken van de user RMAN. Een catalog is overigens niet per definitie noodzakelijk, deze actie zou ook middels de 'nocatalog' optie kunnen.

```
$ sqlplus "sys/topsecret@DB920 as sysdba"

SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 12 11:06:11 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create user rman identified by topsecret
2 default tablespace tools
3 temporary tablespace temp;
```

```
User created.

SQL> grant create session
2, recovery_catalog_owner
3, resource to rman;

Grant succeeded.
```

Vervolgens wordt daadwerkelijk de catalog aangemaakt en de target-database geregistreerd:

```
$ . oraenv
ORACLE_SID = [DB920] ? DB1020
$ rman target / catalog rman/topsecret@DB920

Recovery Manager: Release 10.2.0.1.0 - Production on Mon Dec 12
09:03:46 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

connected to target database: DB1020 (DBID=4063779737)
connected to recovery catalog database

RMAN> create catalog;

recovery catalog created

RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

## De reguliere back-up vindt plaats

```
RMAN> backup database;

Starting backup at 12-DEC-05
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=/u01/oradata/DB1020/datafile/o1_mf_system_
1f057xdy_.dbf
input datafile fno=00003 name=/u01/oradata/DB1020/datafile/o1_mf_sysaux_
1f058sp0_.dbf
input datafile fno=00002 name=/u01/oradata/DB1020/datafile/o1_mf_
undotbs1_1f058qd7_.dbf
input datafile fno=00005 name=/u01/oradata/DB1020/datafile/rk01.dbf
input datafile fno=00004 name=/u01/oradata/DB1020/datafile/o1_mf_users_
1f05bs6v_.dbf
input datafile fno=00006 name=/u01/oradata/DB1020/datafile/o1_mf_trans-
por_1s0174pl_.dbf
channel ORA_DISK_1: starting piece 1 at 12-DEC-05
channel ORA_DISK_1: finished piece 1 at 12-DEC-05
piece handle=/u01/oradata/flash_recovery_area/DB1020/backupset/2005_12_
12/o1_mf_nnndf_TAG20051212T090830_1std02cz_.bkp tag=TAG20051212T090830
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:59
Finished backup at 12-DEC-05
```

```
Starting Control File and SPFILE Autobackup at 12-DEC-05
piece handle=/u01/oradata/flash_recovery_area/DB1020/autobackup/2005_12_
12/o1_mf_s_576839499_1std5xpv_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 12-DEC-05
```

Na de reguliere back-up vinden er nieuwe transacties plaats, uiteindelijk staan er 22574 rijen in de tabel:

```
SQL> select count(*) from rk.big_tab;

COUNT(*)
-----
22574
```

## De actie

Vanaf dit punt kan de demo daadwerkelijk plaatsvinden. De target-database had aanvankelijk een tabel 'big\_tab' met 11269 rijen. Deze target-database is geback-up't en vervolgens zijn er in de 'big\_tab' tabel nog rijen toegevoegd totdat er uiteindelijk 22574 rijen in stonden. Om de betreffende tablespace waarin de tabel staat aan te kunnen brengen op een target-omgeving is dus de RMAN-backup nodig alsmede de archived redologs op disk van de source-database. Vanaf dit moment is het mogelijk middels het onderstaande script de betreffende data (de gehele tablespace) als het ware 'uit de back-up te halen' en klaar te zetten voor transport naar de target database.

Opm:

```
Source database : DB1020
Target database : DBDWH
RMAN catalog : DB920
```

```
# Filename: transport-tspitr.sh
# #####
rman << EOF
connect target /
connect catalog rman/topsecret@DB920
TRANSPORT TABLESPACE transport_tbs
TABLESPACE DESTINATION '/u01/oradata/DBDWH/datafile'
AUXILIARY DESTINATION '/u01/oradata/AUX';
EOF
```

## Resultaat

De beperkte set RMAN-commando's die ik bovenstaand opgegeven heb, worden door Oracle als het ware 'uitgevouwen' (zoals dat ook al bij een losse TSPITR gebeurde). De onderstaande scripting is dus automatisch gegenereerd. Op plaatsen waar de tekens \*\*\* staan vermeld, heb ik cursief commentaar toegevoegd.

```

$ ./transport-tspitr.sh

Recovery Manager: Release 10.2.0.1.0 - Production on Mon Dec 12
09:26:21 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

RMAN>
connected to target database: DB1020 (DBID=4063779737)

RMAN>
connected to recovery catalog database

RMAN> 2> 3>

```

Er wordt automatisch een initialisatie-parameter bestand gemaakt om een dummy-instance op te kunnen starten ten bate van de restore. Aansluitend wordt de instance ook gestart.

```

Creating automatic instance, with SID='tFbv'

initialization parameters used for automatic instance:
db_name=DB1020
compatible=10.2.0.0
db_block_size=8192
db_files=200
db_unique_name=tspitr_DB1020_tFbv
large_pool_size=1M
shared_pool_size=110M
#No auxiliary parameter file used
db_create_file_dest=/u01/oradata/AUX
control_files=/u01/oradata/AUX/cntrl_tspitr_DB1020_tFbv.f

starting up automatic instance DB1020

Oracle instance started

Total System Global Area      201326592 bytes

Fixed Size                    1218508 bytes
Variable Size                 146802740 bytes
Database Buffers              50331648 bytes
Redo Buffers                   2973696 bytes
Automatic instance created

```

Nu de instance 'in de lucht is' wordt de controlfile gerestored.

```

contents of Memory Script:
{
# set the until clause
set until scn 255985;
# restore the controlfile
restore clone controlfile;
# mount the controlfile

sql clone 'alter database mount clone database';
# archive current online log for tspitr to a resent until time
sql 'alter system archive log current';
# avoid unnecessary autobackups for structural changes during TSPITR
sql 'begin dbms_backup_restore.AutoBackupFlag(FALSE); end;';
# resync catalog after controlfile restore
resync catalog;
}

```

```

executing Memory Script

executing command: SET until clause

Starting restore at 12-DEC-05
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: sid=37 devtype=DISK

channel ORA_AUX_DISK_1: starting datafile backupset restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece /u01/oradata/flash_
recovery_area/DB1020/autobackup/2005_12_12/o1_mf_s_576839499_1std5xpv_
.bkp
channel ORA_AUX_DISK_1: restored backup piece 1
piece handle=/u01/oradata/flash_recovery_area/DB1020/autobackup/2005_12_
12/o1_mf_s_576839499_1std5xpv_.bkp tag=TAG20051212T091139
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:03
output filename=/u01/oradata/AUX/cntrl_tspitr_DB1020_tFbv.f
Finished restore at 12-DEC-05

sql statement: alter database mount clone database

sql statement: alter system archive log current

sql statement: begin dbms_backup_restore.AutoBackupFlag(FALSE); end;

starting full resync of recovery catalog
full resync complete
released channel: ORA_AUX_DISK_1

```

Vervolgens worden de minimaal benodigde datafiles gerestored en gerecoverd.

```

contents of Memory Script:
{
# generated tablespace point-in-time recovery script
# set the until clause
set until scn 255985;
# set an omf destination filename for restore
set newname for clone datafile 1 to new;
# set an omf destination filename for restore
set newname for clone datafile 2 to new;
# set an omf destination filename for restore
set newname for clone datafile 3 to new;
# set an omf destination tempfile
set newname for clone tempfile 1 to new;
# set a destination filename for restore
set newname for datafile 6 to
"/u01/oradata/DBDWH/datafile/o1_mf_transpor_ls0174pl_.dbf";
# rename all tempfiles
switch clone tempfile all;
# restore the tablespaces in the recovery set plus the auxilliary
tablespaces
restore clone datafile 1, 2, 3, 6;
switch clone datafile all;
#online the datafiles restored or flipped
sql clone "alter database datafile 1 online";
#online the datafiles restored or flipped
sql clone "alter database datafile 2 online";
#online the datafiles restored or flipped
sql clone "alter database datafile 3 online";
#online the datafiles restored or flipped
sql clone "alter database datafile 6 online";
}

```

```

# make the controlfile point at the restored datafiles, then recover
them
recover clone database tablespace "TRANSPORT_TBS", "SYSTEM",
"UNDOTBS1", "SYSAUX" delete archivelog;
alter clone database open resetlogs;
# PLUG HERE the creation of a temporary tablespace if export fails due
to lack
# of temporary space.
# For example in Unix these two lines would do that:
#sql clone "create tablespace aux_tspitr_tmp
#         datafile '/tmp/aux_tspitr_tmp.dbf' size 500K";
}
executing Memory Script

executing command: SET until clause

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

renamed temporary file 1 to /u01/oradata/AUX/TSPITR_DB1020_TFBV/
datafile/ol_mf_temp_%u_.tmp in control file

Starting restore at 12-DEC-05
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: sid=39 devtype=DISK

channel ORA_AUX_DISK_1: starting datafile backupset restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup
set
restoring datafile 00001 to /u01/oradata/AUX/TSPITR_DB1020_TFBV/data-
file/ol_mf_system_%u_.dbf
restoring datafile 00002 to /u01/oradata/AUX/TSPITR_DB1020_TFBV/data-
file/ol_mf_undotbs1_%u_.dbf
restoring datafile 00003 to /u01/oradata/AUX/TSPITR_DB1020_TFBV/data-
file/ol_mf_sysaux_%u_.dbf
restoring datafile 00006 to /u01/oradata/DBDWH/datafile/ol_mf_transpor-
ls0174pl_.dbf
channel ORA_AUX_DISK_1: reading from backup piece /u01/oradata/flash_
recovery_area/DB1020/backupset/2005_12_12/ol_mf_nnndf_
TAG20051212T090830_1std02cz_.bkp
channel ORA_AUX_DISK_1: restored backup piece 1
piece handle=/u01/oradata/flash_recovery_area/DB1020/backupset/2005_12_
12/ol_mf_nnndf_TAG20051212T090830_1std02cz_.bkp tag=TAG20051212T090830
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:02:36
Finished restore at 12-DEC-05

datafile 1 switched to datafile copy
input datafile copy recid=5 stamp=576840569 filename=/u01/oradata/AUX/
TSPITR_DB1020_TFBV/datafile/ol_mf_system_1stf2gd7_.dbf
datafile 2 switched to datafile copy
input datafile copy recid=6 stamp=576840569 filename=/u01/oradata/AUX/
TSPITR_DB1020_TFBV/datafile/ol_mf_undotbs1_1stf2gkx_.dbf
datafile 3 switched to datafile copy
input datafile copy recid=7 stamp=576840570 filename=/u01/oradata/AUX/
TSPITR_DB1020_TFBV/datafile/ol_mf_sysaux_1stf2gh2_.dbf
datafile 6 switched to datafile copy
input datafile copy recid=8 stamp=576840570 filename=/u01/oradata/AUX/
TSPITR_DB1020_TFBV/datafile/ol_mf_transpor_1stf2gss_.dbf

```

```

sql statement: alter database datafile 1 online

sql statement: alter database datafile 2 online

sql statement: alter database datafile 3 online

sql statement: alter database datafile 6 online

Starting recover at 12-DEC-05
using channel ORA_AUX_DISK_1

starting media recovery

archive log thread 1 sequence 126 is already on disk as file /u01/
oradata/flash_recovery_area/DB1020/archivelog/2005_12_12/ol_mf_1_126_
1stdl15d_.arc
archive log thread 1 sequence 127 is already on disk as file /u01/
oradata/flash_recovery_area/DB1020/archivelog/2005_12_12/ol_mf_1_127_
1stdt89z_.arc
archive log filename=/u01/oradata/flash_recovery_area/DB1020/
archivelog/2005_12_12/ol_mf_1_126_1stdl15d_.arc thread=1 sequence=126
archive log filename=/u01/oradata/flash_recovery_area/DB1020/
archivelog/2005_12_12/ol_mf_1_127_1stdt89z_.arc thread=1 sequence=127
media recovery complete, elapsed time: 00:00:11
Finished recover at 12-DEC-05

database opened

```

Oracle voert nu zelf de 'loskoppel-actie' uit van het transportable-tablespace stuk. Hiervoor wordt datapump gebruikt, voordat datapump bestond kon dit alleen maar met 'exp'.

```

contents of Memory Script:
{
#mark read only the tablespace that will be exported
sql clone "alter tablespace TRANSPORT_TBS read only";
# create directory for datapump export
sql clone "create or replace directory STREAMS_DIROBJ_DPDIR as ''
/u01/oradata/DBDWH/datafile''";
# export the tablespaces in the recovery set
host 'expdp userid="/@/(DESCRIPTION=(ADDRESS=(PROTOCOL=beq))
(PROGRAM=/u01/app/oracle/product/10.2.0/bin/oracle))\ (ARGV0=oracletFbv\
)\ (ARGS=^\ (DESCRIPTION=(LOCAL=YES)\ (ADDRESS=(PROTOCOL=beq)\)\)\)^\
(ENVS=^ORACLE_SID=tFbv^\)\)\ (CONNECT_DATA=(SID=tFbv)\)\)' as sysdba\
transport_tablespaces=
TRANSPORT_TBS dumpfile=
dmpfile.dmp directory=
STREAMS_DIROBJ_DPDIR logfile=
explog.log';
}
executing Memory Script

sql statement: alter tablespace TRANSPORT_TBS read only

sql statement: create or replace directory STREAMS_DIROBJ_DPDIR as ''
u01/oradata/DBDWH/datafile''

Export: Release 10.2.0.1.0 - Production on Monday, 12 December, 2005
9:30:12

Copyright (c) 2003, 2005, Oracle. All rights reserved.

```

```

Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0
- Production
With the Partitioning, OLAP and Data Mining options
Starting "SYS"."SYS_EXPORT_TRANSPORTABLE_01": userid="/
*****@(DESCRIPTION=(ADDRESS=(PROTOCOL=beq)(PROGRAM=/u01/app/oracle/
product/10.2.0/bin/oracle)(ARGV0=oracleTfBv)(ARGS=(DESCRIPTION=\
(LLOCAL=YES)\(ADDRESS=(PROTOCOL=beq)\)\)))(ENVS=ORACLE_
SID=tFbv))(CONNECT_DATA=(SID=tFbv)) AS SYSDBA" transport_tablespaces=
TRANSPORT_TBS dumpfile=dmpfile.dmp directory=STREAMS_DIROBJ_DPDIR
logfile=explog.log
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Master table "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
*****
Dump file set for SYS.SYS_EXPORT_TRANSPORTABLE_01 is:
/u01/oradata/DBDWH/datafile/dmpfile.dmp
Job "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at
09:30:52

host command complete

```

Vervolgens maakt Oracle ook alvast de benodigde scripts aan om de 'losgekoppelde' datafile behorende bij de tablespace ook weer aan de target-database te koppelen.

```

/*
The following command may be used to import the tablespaces.
Substitute values for <logon> and <directory>.
impdp <logon> directory=<directory> dumpfile= 'dmpfile.dmp'
transport_datafiles=/u01/oradata/AUX/TSPITR_DB1020_TFBV/datafile/o1_mf_
transpor_1stf2gss_.dbf
*/
-----
-- Start of sample PL/SQL script for importing the tablespaces
-----
-- creating directory objects
CREATE DIRECTORY STREAMS$DIROBJ$1 AS '/u01/oradata/DBDWH/datafile/';
CREATE DIRECTORY STREAMS$DIROBJ$DPDIR AS '/u01/oradata/DBDWH/
datafile';
/* PL/SQL Script to import the exported tablespaces */
DECLARE
-- the datafiles
tbs_files dbms_streams_tablespace_admin.file_set;
cvt_files dbms_streams_tablespace_admin.file_set;
-- the dumpfile to import
dump_file dbms_streams_tablespace_admin.file;
dp_job_name VARCHAR2(30) := NULL;
-- names of tablespaces that were imported
ts_names dbms_streams_tablespace_admin.tablespace_set;
BEGIN
-- dump file name and location
dump_file.file_name := 'dmpfile.dmp';
dump_file.directory_object := 'STREAMS$DIROBJ$DPDIR';
-- forming list of datafiles for import
tbs_files(1).file_name := 'o1_mf_transpor_1s0174p1_.dbf';
tbs_files(1).directory_object := 'STREAMS$DIROBJ$1';
-- import tablespaces
dbms_streams_tablespace_admin.attach_tablespaces(
datapump_job_name => dp_job_name,
dump_file => dump_file,
tablespace_files => tbs_files,

```

```

converted_files => cvt_files,
tablespace_names => ts_names);
-- output names of imported tablespaces
IF ts_names IS NOT NULL AND ts_names.first IS NOT NULL THEN
FOR i IN ts_names.first .. ts_names.last LOOP
dbms_output.put_line('imported tablespace ' || ts_names(i));
END LOOP;
END IF;
END;
/
-- dropping directory objects
DROP DIRECTORY STREAMS$DIROBJ$1;
DROP DIRECTORY STREAMS$DIROBJ$DPDIR;
-----
-- End of sample PL/SQL script
-----

```

Nadat het bestand behorende bij de tablespace netjes is losgekoppeld en aan de target-database kan worden gekoppeld ruimt Oracle de bestanden op die nodig waren voor de TSPITR-acties.

```

Removing automatic instance
shutting down automatic instance
Oracle instance shut down
Automatic instance removed
auxiliary instance file /u01/oradata/AUX/cntrl_tspitr_DB1020_tFbv.f
deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/datafile/o1_
mf_system_1stf2gd7_.dbf deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/datafile/o1_
mf_undotbs1_1stf2gkx_.dbf deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/datafile/o1_
mf_sysaux_1stf2gh2_.dbf deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/datafile/o1_
mf_temp_1stf82h3_.tmp deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/online/
o1_mf_1_1stf7sgz_.log deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/online/
o1_mf_2_1stf7tq9_.log deleted
auxiliary instance file /u01/oradata/AUX/TSPITR_DB1020_TFBV/online/
o1_mf_3_1stf7wfl_.log deleted

RMAN>

Recovery Manager complete.

$ ^ABV/datafile/o1_mf_sysaux_1stf2gh2_.dbf deleted
$ UX/TSPITR_DB1020_TFBV/datafile/o1_mf_temp_1stf82h3_.tmp deleted
<
$ UX/TSPITR_DB1020_TFBV/online/o1_mf_1_1stf7sgz_.log deleted
<
$ UX/TSPITR_DB1020_TFBV/online/o1_mf_2_1stf7tq9_.log deleted
<
$ UX/TSPITR_DB1020_TFBV/online/o1_mf_3_1stf7wfl_.log deleted
<

```

## De import

De tablespace wordt gekoppeld aan de target database (DBDWH). Teneinde de datafile(s) behorende bij de tablespace te koppelen aan de target-database heeft Oracle twee scripts

aangemaakt tijdens de bovenstaande actie: een SQL-script dat vanuit de database gedraaid kan worden en een shell-script dat van buiten de database de impdp (datapump import) utility aanroept. In dit voorbeeld gebruik ik het script met de minste code, namelijk het shell-script dat datapump aanroept (de genoemde directory 'rkdump' heb ik overigens wel zelf moeten aanmaken):

```
impdp \sys/topsecret@DBDWH as sysdba\ directory=rkdump
dumpfile='dmpfile.dmp' transport_datafiles=/u01/oradata/AUX/TSPITR
DB1020_TFBV/datafile/ol_mf_transpor_1stf2gss_.dbf

$ ./imprk.sh

Import: Release 10.2.0.1.0 - Production on Monday, 12 December, 2005
10:04:53

Copyright (c) 2003, 2005, Oracle. All rights reserved.

Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0
- Production
With the Partitioning, OLAP and Data Mining options
Master table "SYS"."SYS_IMPORT_TRANSPORTABLE_03" successfully loaded/
unloaded
Starting "SYS"."SYS_IMPORT_TRANSPORTABLE_03": "sys/*****@DBDWH AS
SYSDBA" directory=rkdump dumpfile=dmpfile.dmp transport_datafiles=/u01/
oradata/AUX/TSPITR_DB1020_TFBV/datafile/ol_mf_transpor_1stf2gss_.dbf
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYS"."SYS_IMPORT_TRANSPORTABLE_03" successfully completed at
10:05:00
```

## Ter controle

Ter controle bekijk ik of de tablespace is 'overgekomen':

```
$ . oraenv
ORACLE_SID = [DBDWH] ?
$ sq

SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 12 10:05:09 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> select tablespace_name from dba_tablespaces;

TABLESPACE_NAME
-----
SYSTEM
UNDOTBS1
SYSaux
TEMP
USERS
RK
TRANSPORT_TBS

7 rows selected.
```

Vervolgens wordt ook gecontroleerd of de tabel meegekomen is.

```
SQL> select table_name from dba_tables where owner = 'RK'

TABLE_NAME
-----
BIG_TAB
```

Tenslotte controleer ik of het aantal rijen ook overeenkomt:

```
SQL> select count(*) from rk.big_tab;

COUNT(*)
-----
22574
```

## Samenvattend

RMAN en zeker een TSPITR is niet het eerste waaraan ik zou denken als mij gevraagd zou worden een datawarehouse of staging-area te vullen. De situatie kan zich echter voordoen dat deze methodiek in bepaalde omstandigheden een goede oplossing blijkt te zijn. Het is direct al duidelijk dat een synchronisatie op dit niveau een te zware load op productie-omgevingen kan voorkomen. Back-ups worden immers al dagelijks gemaakt en de data zijn daarmee dus al een keer gekopieerd en dus kan deze methodiek een aanzienlijke hoeveelheid performance-overhead besparen. Een niet te moeilijk RMAN-script ontpopt zich tot een reeks commando's die deze actie uiteindelijk geheel automatisch doet laten verlopen.

**René Kundersma** is software-ingenieur en Oracle Certified Master.