

RMAN in praktijk

Deel 3: restore

In dit artikel over de Oracle Recovery Manager behandelen we restore, waar een backup initieel voor wordt gemaakt. Aan de orde komen: restore van een database, tablespace, datafile en een datablock. Verder wordt de 'Point In Time Restore' behandeld.

Restore database

In dit voorbeeld gaan we uit van een situatie dat er geen files meer bestaan van de database. De catalog database is niet bereikbaar en de backup bestaat uit aparte backup sets voor datafiles, archive logs en controlfile. De spfile is met de autobackup van de controlfiles gemaakt.

Als het environment is gezet, dan kan je met RMAN de instance starten ook al heb je geen spfile. Na de initiële melding dat het pfile niet wordt gevonden, komt de melding 'starting Oracle instance without parameter file for retrieval of spfile'. De instance is dan gestart met een temporary spfile, waarna de controlfile en spfile kunnen worden teruggezet. Voordat de restore kan plaatsvinden moet eerst het DBID, database identifier, worden gezet. Deze kan worden teruggevonden in de naam van de backupset van de controlfile backup. Het tweede deel van de naam is de DBID. De volledige naam van het path en file moeten aangegeven worden bij het restoren,.

Let op: in de voorbeelden is alleen de meest relevante output weergegeven. De rest is vervangen door (...).

```
RMAN> set dbid=1198421209 ;
RMAN> @restore_cf_sp.rmn
run{
allocate channel GETIT
  device type disk format
    '/opt/oracle/app/oracle/backup/orcl/%F' ;
restore controlfile from '/opt/oracle/app/oracle/backup/
orcl/c-1198421209-20090203-01';
restore spfile from '/opt/oracle/app/oracle/backup/
orcl/c-1198421209-20090203-01';
}
(...)
/*
the files are restored in $ORACLE_HOME/dbs
check desired location and correct this.
*/
(...)
```

Hierna is het originele spfile en controlfile weer tot de beschikking, de database kan hiermee in mount status worden gestart. In dit geval zijn de default locaties niet herkend dus de files zijn geplaatst in de sw tak. Deze moeten eerst met de hand gekopieerd worden naar de correcte locatie. De locatie van de controlfiles kun je uit de spfile filteren (strings spfileMyDB.ora | grep control_file). Als de instance is gestart, met de juiste parameter en controlfile, kunnen de andere database files worden gerestored. Om te weten welke backupsets er nodig zijn en of deze aanwezig zijn dan kan met het restore commando een 'preview' meegeven. De restore wordt dan niet uitgevoerd maar dan krijg je wel een listing met de benodigde backupset en hun status. Ook wordt dan aangegeven tot welke transactie gerecovered moet worden om alle datafiles consistent te krijgen. Indien de backupsets met RMAN van disk naar tape geschreven zijn geeft de preview ook aan waar de betreffende backupset zich bevindt.

Een voorbeeld van restore:

```
run
{
allocate channel repare device type disk format '/opt/oracle/app/
oracle/backup/MyDB/%U';
restore database ;
recover database ;
alter database open ;
}
```

Tijdens de restore word hier een channel aangemaakt, de default channel heeft bandwidth limiet. Het doel is de restore tijd zo kort mogelijk te houden. Als een backup afgebroken is en opnieuw wordt opgestart worden de datafiles, die bruikbaar zijn en de juiste status hebben, overgeslagen. Dit doet RMAN zelf, daar hoeft je niets voor te doen.

Algemene opmerking

Indien het deel dat moet worden gerestored niet kritisch is voor het laten draaien van de database kan men dit online uitvoeren. Het deel dat defect is wordt daarbij offline gezet en is dan ook niet voor de gebruikers te benaderen. Delen die niet offline zijn kunnen gewoon gebruikt worden. Afhankelijk van

wat er offline gezet wordt betekent dit downtime voor de eindgebruiker. Maar omdat men de schaal klein houdt is de restoretijd en dus ook de downtime beperkt. Doelstelling is iedere keer de database voor zoveel mogelijk gebruikers beschikbaar te hebben en de downtime zoveel mogelijk te beperken. Downtime is afhankelijk van de snelheid van het medium en de grootte van de objecten die worden gerestored.

Restore tablespace (online)

In de casus die we hier behandelen gaan we uit van een tablespace met een single big file. Hierbij moeten we wel noteren dat de betreffende tablespace niet te benaderen is vanuit oracle en het betreft niet de system tablespace. De betreffende datafile die bij de tablespace hoort is verwijderd. Het herstellen gebeurt in drie stappen. De tablespace wordt offline gezet, een restore en recovery vindt plaats, de tablespace wordt weer online gezet. \

```
[oracle@oralinet orcl]$ mv users01.dbf users01.org1
RMAN> sql 'alter tablespace users offline immediate ' ;
sql statement: alter tablespace users offline immediate
RMAN> restore tablespace users ;
Starting restore at 04-FEB-09
using channel ORA_DISK_1
...
Finished restore at 04-FEB-09
RMAN> recover tablespace users ;
Starting recover at 04-FEB-09
using channel ORA_DISK_1
starting media recovery
(...)
RMAN> sql 'alter tablespace users online ' ;
sql statement: alter tablespace users online

Recover datafile (online)
```

Eigenlijk komt dit dicht bij het herstellen van een tablespace. Alleen hebben we hier een tablespace met meerdere datafiles. En een van de datafiles is defect, de betreffende datafile wordt offline gezet. Een restore en recovery van het betreffende datafile wordt uitgevoerd. Vervolgens wordt deze weer online gezet en is de data uit de betreffende datafile voorhanden.

```
[oracle@oralinet orcl]$ mv users02.dbf users02.org
SQL rick@ORCL> alter database
  2 datafile '/opt/oracle/app/oracle/oradata/orcl/users02.dbf' offline
;
Database altered.
RMAN> restore datafile '/opt/oracle/app/oracle/oradata/orcl/users02.dbf'
;
Starting restore at 04-FEB-09
(...)
Finished restore at 04-FEB-09
RMAN> sql 'alter database datafile 6 online ' ;
sql statement: alter database datafile 6 online
RMAN-00571: =====
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of sql command on default channel at 02/04/2009
13:17:03
RMAN-11003: failure during parse/execution of SQL statement: alter
database datafile 6 online
ORA-01113: file 6 needs media recovery
ORA-01110: data file 6: '/opt/oracle/app/oracle/oradata/orcl/users02.
```

```
dbf'
RMAN> recover datafile 6 ;
Starting recover at 04-FEB-09
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 04-FEB-09
RMAN> sql 'alter database datafile 6 online ' ;
sql statement: alter database datafile 6 online
```

In het bovenstaande voorbeeld ziet men dat een recovery echt nodig is ook al zijn er geen veranderingen geweest sinds de backup. Het betrof hier een nieuw aangemaakt file die verder leeg was. Toch wil rman een recovery laten uitvoeren om te controleren of het file consistent is. Recovery en consistentie check zijn dus essentiële onderdelen die bij elkaar horen. Het onderstaande script geeft een goed voorbeeld daarvan.

```
run
{
  sql "alter database datafile x offline";
  restore datafile x;
  recover datafile x;
  sql "alter database datafile x online";
}
```

Restore datablock

Om een datablock te repareren moet men deze eerst signaleren en daarna kan men deze online repareren. Wel gelden de gebruikelijke randvoorwaarden, de backups die hier voor nodig zijn moeten aanwezig en consistent zijn. Een defect block kan men terug vinden in de view `V$DATABASE_BLOCK_CORRUPTION`. Bij twijfel kan men een controle forceren door het uitvoeren van `rman 'validate database'` of `'validate tablespace HACKME;'`. Het repareren gaat met het commando `'recover corruption list'`, rman zal dan alle entries in `V$DATABASE_BLOCK_CORRUPTION` repareren. Aangezien dit een procedure is die men niet gauw in de praktijk tegenkomt. Hebben we een casus gemaakt waarbij een tabel genaamd filler kapot is gemaakt. Hieronder volgen de stappen en de getrimde output van de procedure.

```
SQL rick@MYDB> select count(*) from filler ;
select count(*) from filler
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 6, block # 1982)
ORA-01110: data file 6: '/opt/oracle/app/oracle/oradata/MyDB/hackme01.
dbf'
SQL rick@MYDB> select * from v$database_block_corruption ;

  FILE#      BLOCK#      BLOCKS CORRUPTION_CHANGE#  CORRUPTIO
-----
        6         1982            1              0 CORRUPT
```

We hebben niets anders gedaan dan een block van een tabel kapot gemaakt. Er zijn geen controles uitgevoerd anders dan dat de database is gestart. De check is gedaan tijdens het starten van de database en kan in de alert.log file terug gevonden worden.

```
Reread of blocknum=1982, file=/opt/oracle/app/oracle/oradata/MyDB/
hackme01.dbf. found same corrupt data
2009-03-11 09:33:15.481000 +01:00
Checker run found 1 new persistent data failures
2009-03-11 09:36:47.695000 +01:00
...
Corrupt Block Found
  TSN = 8, TSNAME = HACKME
  RFN = 6, BLK = 1982, RDBA = 25167806
  OBJN = 70958, OBJD = 70958, OBJECT = FILLER, SUBOBJECT =
  SEGMENT OWNER = RICK, SEGMENT TYPE = Table Segment
```

Indien de 'validate database' onder rman wordt uitgevoerd dan is het volgende segment in de logfile/output terug te vinden:

```
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
6 FAILED 0 214 3104 859342
File Name: /opt/oracle/app/oracle/oradata/MyDB/hackme01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data 1 2072
Index 0 748
Other 0 70
validate found one or more corrupt blocks
See trace file /opt/oracle/app/oracle/diag/rdbms/mydb/MyDB/trace/MyDB_
ora_5427.trc for details
```

Voordat we met een recovery beginnen doen we een check om te zien of we alle backupsets die we nodig hebben voorhanden zijn.

```
RMAN> recover corruption list preview summary ;
Starting recover at 11-MAR-09
using channel ORA_DISK_1
List of Backups
=====
Key TY LV S Device Type Completion Time #Pieces #Copies Compressed
Tag
---
66 B F A DISK 10-MAR-09 1 1 YES
TAG20090310T112540
using channel ORA_DISK_1
List of Archived Log Copies for database with db_unique_name MYDB
=====
Key Thrd Seq S Low Time
-----
93 1 22 A 04-MAR-09
Name: /opt/oracle/app/oracle/flash_recovery_area/MYDB/
archivelog/2009_03_10/o1_mf_1_22_4vdjk8mq_.arc
95 1 23 A 10-MAR-09
Name: /opt/oracle/app/oracle/flash_recovery_area/MYDB/
archivelog/2009_03_11/o1_mf_1_23_4vgwwf0k_.arc
Media recovery start SCN is 945483
Recovery must be done beyond SCN 945483 to clear datafile fuzziness
Finished recover at 11-MAR-09
```

De controle op aanwezigheid van de backups laat de full en archivelogfiles zien die nodig zijn voor het herstellen van het datablock. Aangezien alle randvoorwaarden zijn vervuld kunnen we een recovery uitvoeren.

```
RMAN> recover corruption list ;
Starting recover at 11-MAR-09
using channel ORA_DISK_1
channel ORA_DISK_1: restoring block(s)
channel ORA_DISK_1: specifying block(s) to restore from backup set
restoring blocks of datafile 00006
(...)
channel ORA_DISK_1: restored block(s) from backup piece 1
```

```
channel ORA_DISK_1: block restore complete, elapsed time: 00:00:08
starting media recovery
media recovery complete, elapsed time: 00:00:07
Finished recover at 11-MAR-09
```

Het volgende is opmerkelijk, we hebben niets offline gezet. Dit in tegenstelling tot een tablespace of een datafile recovery. De andere datablocks van de betreffende table blijven toegankelijk. In de restore wordt alleen het specifieke block uit de backup gehaald. Gedurende het recovery process wordt gekeken of er wijzigingen zijn voor het block indien dit het geval is worden die doorgevoerd. De casus is natuurlijk een redelijk ideale situatie omdat dit netjes is gemanipuleerd. De doelstelling is om de procedure te laten zien en de afwijkingen t.o.v. De andere restore procedures.

Database Point In Time Restore

Het doel is om de database tot een bepaald tijdstip te herstellen, in een consistente situatie. De noodzaak daarvan kan verschillend zijn, een mislukte update, een human error of een software error. Dit kan op hetzelfde systeem zijn maar ook op een ander systeem. Het tijdstip tot waar gerestored wordt kan op drie manieren aangegeven worden. Namelijk op basis van de real time, scn of het sequence nummer van een archivelog.

```
connect target /
spool msglog to '/opt/oracle/app/oracle/admin/general/log/DBPITR.log'
run {
set until time="to_date('23-02-2009 09:00:00','dd-mm-yyyy hh:mi:ss')";
restore database ;
recover database ;
alter database open resetlogs ;
}
===> output
connected to target database: MYDB (DBID=2594099710, not open)
Spooling started in log file: /opt/oracle/app/oracle/admin/general/log/
DBPITR.log
Recovery Manager11.0.6.0
executing command: SET until clause
using target database control file instead of recovery catalog
(...)
Finished restore at 23-FEB-09
Starting recover at 23-FEB-09
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:03
Finished recover at 23-FEB-09
database opened
Recovery Manager complete.
```

In het bovenstaande voorbeeld zal de database gerecovered worden tot de timestamp in het archivelog file overeenkomt met de aangegeven tijd of de eerst volgende timestamp daarna. Men moet dus goed overwegen tot waar gerecovered moet worden. In de v\$log_history kan men een overzicht vinden van de archivelogs die aangemaakt zijn.

```
[oracle@localhost sql]$ cat loghist.sql
select sequence# as "Sequence"
, first_change# as "SCN"
, to_char(first_time,'hh:mi:ss DD-MM-YY')
from v$log_history
```

```

/
====> output
Sequence          SCN_TO_CHAR(FIRST_TIM
-----
3                810467 10:31:23 23-02-09
4                810472 10:31:29 23-02-09
(....)
21               918835 03:10:50 03-03-09
22               927018 09:09:55 04-03-09
    
```

De bovenstaande query geeft een overzicht van de log sequences , SCN en de timestamp op basis waarvan men een PITR kan uitvoeren.

Tablespace point in time recovery

Het is ook mogelijk om een PITR van een table space te doen, dit wordt een TSPITR genoemd. Een aantal zaken dient men wel in de gaten te houden. Als een TSPITR doet moet men wel in de gaten houden welk control file voor handen is. Het makkelijkste is een TSPITR te doen met een recovery catalog omdat deze een historie heeft van de bestaande structuur. Een controlfile heeft wat dat betreft alleen maar een moment opname. Als men dan terug wil moet men ook een controlfile hebben van die periode met de juiste informatie. Een aantal stappen wordt bij een TSPITR onder water uitgevoerd. Er word namelijk een clone instance gemaakt die een PITR is. Men moet voor de clone instance een locatie toewijzen, hou er rekening mee dat er ruimte is voor system, sysaux, undo en dependend tablespaces minus

de tablespace(s) die gerecovered moeten worden. De benodigde tablespaces worden naar de juiste locatie gekopieerd en de andere files naar de opgegeven destination. Na recovery worden de tablespace(s) doormiddel van 'transportable tablespace' procedure toegevoegd aan de database. De niet benodigde datafiles worden verwijderd van het systeem.

Deze methode kan problemen geven als:

- De tablespace in de tussenliggende periode is gerename of gedropped.
- Als er dependencies zijn in andere tablespaces .
- Er tussentijds een nieuwe incarnation is gemaakt.

Deze dependencies kan men terug vinden in de table van sys genaamd TS_PITR_CHECK. Men kan dan de objecten indien mogelijk dan ook een PITR geven. Ook kunnen er objected aangemaakt zijn na het moment dat we terug willen. Deze zijn te vinden in een tabel van sys genaamd TS_PITR_OBJECTS_TO_BE_DROPPED.

```

Script :
connect target /
spool msglog to '/opt/oracle/app/oracle/admin/general/log/TSPITR.log'
run {
sql 'alter tablespace HACKME offline immediate';
recover tablespace HACKME
until scn 811926
auxiliary destination '/opt/oracle/app/oracle/aux';
    
```

(Advertentie)

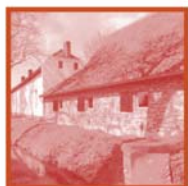
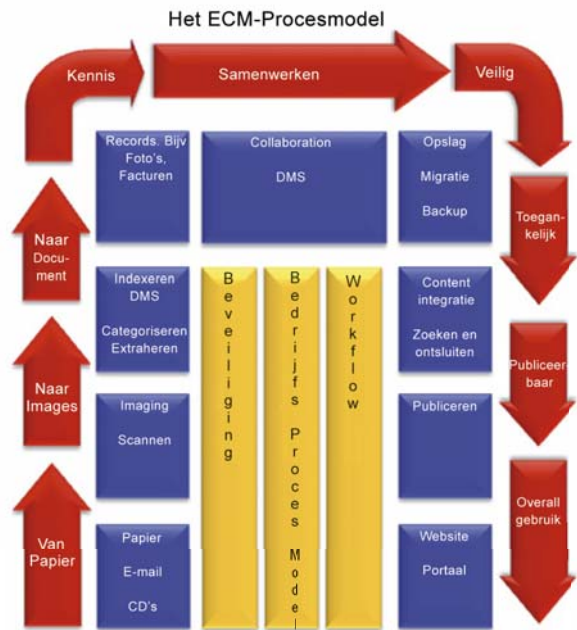


OraVision, De integratie specialist

- ✓ De bruggenbouwer tussen uw front - en backoffice
 - Waar Enterprise Application Integration en Enterprise Content Management elkaar ontmoeten
 - Waar primaire en documentprocessen naadloos samenwerken

OraVision, De ECM Specialist

- ✓ De specialist in het Oracle ECM platform
 - Oracle Collaboration Suite
 - Oracle Content Services
 - Oracle Content Database Suite
 - Oracle Enterprise Content Management Suite
 - Oracle Universal Content Management (voorheen Stellent)
- ✓ Biedt producten en diensten voor het hele ECM - Procesmodel
- ✓ Ons team van ervaren ICT - en ECM - specialisten maakt het verschil en is de sleutel tot uw succes!
- ✓ Member of the BCT Group



Informatie ? www.oravision.nl

Of bel: 045 - 564 55 80

```

sql 'alter tablespace HACKME online ';
}
=====>  OUTPUT
connected to target database: MYDB (DBID=2594099710)
Spooling started in log file: /opt/oracle/app/oracle/admin/general/log/
TSPITR.log
Recovery Manager11.1.0.6.0
(...)
RMAN-05026: WARNING: presuming following set of tablespaces applies to
specified point in time
(...)
Creating automatic instance, with SID='kmtE'
initialization parameters used for automatic instance:
db_name=MYDB
(...)
db_unique_name=tsplitr_MYDB_kmtE
(...)
#No auxiliary parameter file used
db_create_file_dest=/opt/oracle/app/oracle/aux
control_files=/opt/oracle/app/oracle/aux/cntrl_tsplitr_MYDB_kmtE.f
starting up automatic instance MYDB
(...)
contents of Memory Script:
{
# set the until clause
set until scn 811926;
# restore the controlfile
restore clone controlfile;
# mount the controlfile
sql clone 'alter database mount clone database';
# archive current online log for tsplitr 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;';
}
executing Memory Script
executing command: SET until clause
(...)
sql statement: alter database mount clone database
sql statement: alter system archive log current
sql statement: begin dbms_backup_restore.AutoBackupFlag(FALSE); end;
contents of Memory Script:
{
# generated tablespace point-in-time recovery script
# set the until clause
set until scn 811926;
# set an omf destination filename for restore
set newname for clone datafile 1 to new;
(...)
set newname for datafile 6 to
"/opt/oracle/app/oracle/oradata/MyDB/hackme01.dbf";
# rename all tempfiles
switch clone tempfile all;
# restore the tablespaces in the recovery set plus the auxiliary
tablespaces
restore clone datafile 1, 3, 2, 6;
switch clone datafile all;
(...)
# make the controlfile point at the restored datafiles, then recover
them
recover clone database tablespace "HACKME", "SYSTEM", "UNDOTBS1",
"SYSAUX" delete archivelog;
alter clone database open resetlogs;
}
executing Memory Script
executing command: SET until clause
(...)
creating datafile file number=6 name=/opt/oracle/app/oracle/oradata/
MyDB/hackme01.dbf
(...)
sql statement: alter database datafile 6 online
Starting recover at 23-FEB-09
using channel ORA_AUX_DISK_1
starting media recovery
archived log for thread 1 with sequence 2 is already on disk as file /
opt/oracle /app/oracle/flash_recovery_area/MYDB/archivelog/2009_02_23/

```

```

o1_mf_1_2_4t4soxk4_.arc
(...)
media recovery complete, elapsed time: 00:00:23
Finished recover at 23-FEB-09
database opened
contents of Memory Script:
{
# export the tablespaces in the recovery set
host 'exp userid = '*****'
point_in_time_recover=y tablespaces= HACKME file= tsplitr_a.dmp';
# shutdown clone before import
shutdown clone immediate
# import the tablespaces in the recovery set
host 'imp userid = '*****'
point_in_time_recover=y file=tsplitr_a.dmp';
# online/offline the tablespace imported
sql "alter tablespace HACKME online";
sql "alter tablespace HACKME offline";
# enable autobackups in case user does open resetlogs from RMAN after
TSPITR
sql 'begin dbms_backup_restore.AutoBackupFlag(TRUE); end;';
}
executing Memory Script
(...)
Automatic instance removed
*****Volgende melding komt niet indien een catalog gebruikt zou
zijn.*****
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of recover command at 02/23/2009 11:32:05
RMAN-03015: error occurred in stored script Memory Script
RMAN-03009: failure of sql command on default channel at 02/23/2009
11:32:05
RMAN-11003: failure during parse/execution of SQL statement: alter
tablespace H ACKME online

```

ORA-01190: control file or data file 6 is from before the last
RESETLOGS

ORA-01110: data file 6: '/opt/oracle/app/oracle/oradata/MyDB/
hackme01.dbf'

Recovery Manager complete.

Samenvatting

Indien het nodig is kan men een restore/recovery uitvoeren zonder dat de database gestopt wordt. Het is niet noodzakelijk alles te herstellen/recoveren als er iets niet goed gaat. Delen die gerecovered worden zijn niet beschikbaar. Indien alles van die ene bigfile datafile van die enkele tablespace afhankelijk is. Dan betekent het nog steeds totale uitval voor de eindgebruiker. Bij PITR is het prettig een catalog te kunnen gebruiken met name als er verschillende incarnations zijn. Bij een totale database restore is de locatie waar de spfile en controlfile geplaatst worden, afhankelijk van de informatie waar RMAN over beschikt. Een catalog bevat meer historische informatie dan een controlfile.



Rick van Ek is Oracle Senior DBA.
Van Ek Consultancy B.V.