

# Nieuwe features Oracle 11g (2)

## *Diagnostics, virtual columns en Data Recovery Advisor*

***Inmiddels zijn de nodige boeken over New Features in Oracle 11g op de markt verschenen, of ze verschijnen binnenkort. Ik heb ze nog niet gelezen, het is veel leuker eerst zelf eens een beetje rond te neuzen in zo'n nieuwe release en mijn belevenissen en meningen hier weer te geven. Het zal me niet lukken om in dit tempo in Optimize alle nieuwe features te bespreken voordat Oracle met een volgende release komt. In een Optimize in het volgende jaar zal ik daarom eens een paar van de New Features-boeken naast elkaar leggen in een vergelijkende boekbespreking.***

In deze aflevering wil ik twee totaal verschillende zaken behandelen. Ik begin met de diagnostics. Een 'verse' Oracle 11g database krijgt een hele andere set directories mee voor het bewaren van alle mogelijke trace- en logfiles. Als tweede ga ik in op een interessante nieuwe feature in de data-opslag, al kunnen we daarbij eigenlijk juist niet van opslag spreken.

Als laatste behandel ik een interessante uitbreiding van RMAN met een geautomatiseerde adviseur voor de oplossing van problemen met corrupte of ontbrekende datafiles.

### Diagnostics

Onder het begrip Diagnostics valt in een Oracle-omgeving deset logfiles waarmee de DBA de werking en toestand van een database in de gaten kan houden, en problemen kan analyseren.

Sinds jaar en dag wordt daarbij gebruik gemaakt van een directory-structuur die bekend is geworden onder de afkorting OFA, voor Optimal Flexible Architecture. De eerste versie van deze de-facto standaard werd in 1991 gepresenteerd en al snel opgepikt. In 1993 werd deze standaard al in de installatiesoftware voor Oracle 7 verwerkt. In september 1995 schreef Cary Millsap een whitepaper om de standaard goed te definiëren. Deze whitepaper is op veel plaatsen terug te vinden op het net: zoek op OFA Cary Millsap.

In het kort heeft de standaard tot doel om data, software en administratieve bestanden te scheiden, op een zodanige manier dat de bestanden van verschillende databases niet met elkaar worden verward. Ook kan door een goede scheiding van software en database bereikt worden dat een upgrade van de software een minimale impact heeft op de configuratie.

Bij de diagnostics gaat het om de trace- en logfiles die van een instance worden bijgehouden. Deze zijn traditioneel terug te vinden in een directory als:

```
/opt/oracle/admin/d10b/
```

In die directory wordt het gedeelte '/opt/oracle' ook wel aangeduid met ORACLE\_BASE. 'd10b' is hier de naam van de database (en meestal ook van de instance). Dat we meestal de SID (de naam van de instance) gebruiken is op basis van de standaard eigenlijk fout: in feite gaat het daar om de naam van de database. Als we even verder gaan, en we kijken naar de log- en tracefiles van een database d10b met twee instances d10b1 en d10b2 dan zouden deze eigenlijk de volgende namen moeten hebben:

```
/opt/oracle/admin/d10b/bdump/d10b1/alert_d10b1.log
/opt/oracle/admin/d10b/bdump/d10b2/alert_d10b2.log
```

Dat is zoals het in de oude situatie, voor 11g volgens de OFA standaard zou moeten worden opgezet. Naast de directory bdump (voor alert en trace files) zijn er nog een cdump (voor core-dumps), pfile (voor parameterbestanden), adump (voor audit bestanden) etc.

Voor dit artikel heb ik een database d11b gemaakt, voorzien van één instance d11b. Voor het maken van de database heb ik gebruik gemaakt van het grafische tool daarvoor, de dbca (database creation assistant). De dbca gaat uit van de instelling voor ORACLE\_BASE, en maakt dan de alert-logfile als volgt aan:

```
/opt/oracle/diag/rdbms/d11b/d11b/trace/alert_d11b.log
```

Daarbij verwijst de eerste d11b naar de databasenaam, en de tweede d11b naar de instance naam. Voor allerlei andere onderdelen van Oracle, zoals crs (cluster ready services, t.b.v. RAC), asm (automatic storage management) etc. worden ook directories aangemaakt. Ik zal me nu beperken tot de rdbms directory.

```
/opt/oracle
    ORACLE_BASE, te definiëren
    op operating system niveau in
    een environment variabele
    /diag
        De home-directory voor de
        diagnostics
    /rdbms
        De subdirectory voor rdbms-
        gerelateerde diagnostische data
    /d11b
        De naam van de database
    /d11b
        De naam van de instance
    /alert
        Bevat een XML-variant van de
        alert_d11a.log
    /cdump
        Nog steeds de plek voor core-
        dumps
    /hm
        ??
    /incident
        ??
    /incpkg
        Bevat waarschijnlijk packages
        die van incidenten kunnen
        worden gemaakt, om te uploa-
        den naar Oracle Support.
    /ir
        ??
    /lck
        ?? Bevat veel lege bestanden,
        waarschijnlijk lock-files
    /metadata
        Bevat (vermoedelijk) beschrij-
        vingen van de XML-formaten.
    /stage
        ??
    /sweep
        ??
    /trace
        Bevat de 'oude' variant van
        alert_d11b.log en de trace-files
        van de achtergrondprocessen.
```

Er komen een aantal vraagtekens in voor, omdat ik nog geen documentatie heb kunnen ontdekken die deze directories beschrijft. Verder valt op dat de nieuwe werkwijze nog niet helemaal consequent is doorgevoerd. De volgende directories werden aangemaakt volgens de 'oude' OFA standaard toen ik deze d11b database creëerde:

```
/opt/oracle
    ORACLE_BASE, te definiëren
    op operating system niveau in
    een environment variabele
    /admin
        De aloude admin directory
```

```
/d11b
    De naam van de database
/adump
    De directory met de audit-files
/dpdump
    De directory met logfiles van
    de datapump utility
/pfile
    De directory met de initiële
    parameterfile
/scripts
    De directory waarin de creatie-
    scripts van de database werden
    weggeschreven.
```

Van de laatste twee kan ik me voorstellen dat het geen bestanden zijn die met diagnose te maken hebben. Voor adump en dpdump vind ik dat niet opgaan, maar misschien zit er een ongedocumenteerde diepere reden achter.

Al met al zorgt deze nieuwe werkwijze wel voor een stevig zoekplaatje voor de in de aloude OFA-structuur 'vastgeroeste' DBA. Ik moet weer nieuwe kunstjes leren, en veel in mijn vingers vastgeroeste directorynamen moeten nu weer zeer bewust worden ingetypt. In de overgang tussen 10g en 11g gaat dat weer voor een hoop verwarring en 'aanpassingsproblemen' bij de command-line-georiënteerde DBA zorgen. Gelukkig heeft Oracle de nieuwe structuur echter ook voorzien van een heus diagnose-tool. Tik op de commandline in: adrci, en er verschijnt een command prompt waarachter een hele reeks commando's mag worden ingetypt. 'help' was het eerste commando dat ik intypte.

```
d11b@laphroaig:/opt/oracle/diag/rdbms>adrci

ADRCI: Release 11.1.0.6.0 - Beta on Mon Dec 3 01:21:18 2007

Copyright (c) 1982, 2007, Oracle. All rights reserved.

ADR base = "/opt/oracle"
adrci> help

HELP [topic]
Available Topics:
    CREATE REPORT
    ECHO
    EXIT
    HELP
    HOST
    IPS
    PURGE
    RUN
    SET BASE
    SET BROWSER
    SET CONTROL
    SET ECHO
    SET EDITOR
    SET HOMES | HOME | HOMEPATH
    SET TERMOUT
    SHOW ALERT
    SHOW BASE
```

```

SHOW CONTROL
SHOW HM_RUN
SHOW HOMES | HOME | HOMEPATH
SHOW INCDIR
SHOW INCIDENT
SHOW PROBLEM
SHOW REPORT
SHOW TRACEFILE
SPOOL

```

There are other commands intended to be used directly by Oracle, type "HELP EXTENDED" to see the list

adrci>

Met bijv. het commando IPS kan een zgn. incident package worden gemaakt. Dat is een package die alle tracefiles etc. die bij een opgegeven incident horen bevat. Het geheel kan naar Oracle Support gestuurd worden ter ondersteuning van een Support Request. Ik was benieuwd naar incidenten:

```
adrci> adrci> show incident
```

```
ADR Home = /opt/oracle/diag/rdbms/dlla_1/dlla:
```

```
*****
```

INCIDENT_ID	PROBLEM_KEY	CREATE_TIME
1066	ORA 4031	2007-10-06 22:47:59.012929 +02:00
1065	ORA 4031	2007-10-06 22:47:54.740350 +02:00
993	ORA 4031	2007-10-06 22:47:55.248371 +02:00
962	ORA 4031	2007-10-06 22:47:56.440382 +02:00
961	ORA 4031	2007-10-06 22:47:55.060981 +02:00
436	ORA 600 [kzxcparentread-8]	2007-10-06 21:54:27.537897 +02:00
435	ORA 600 [kzxcparentread-8]	2007-10-06 21:54:24.575420 +02:00
434	ORA 600 [kzxcparentread-8]	2007-10-06 21:54:21.759235 +02:00
433	ORA 600 [kzxcparentread-8]	2007-10-06 21:54:19.636060 +02:00
432	ORA 600 [kzxcparentread-8]	2007-10-06 21:54:15.711115 +02:00
209	ORA 4031	2007-10-06 21:44:23.021116 +02:00
145	ORA 4031	2007-10-06 21:44:28.210814 +02:00
115	ORA 4031	2007-10-06 21:44:38.387001 +02:00
114	ORA 4031	2007-10-06 21:44:31.912186 +02:00
113	ORA 4031	2007-10-06 21:44:28.649568 +02:00

15 rows fetched

adrci>

De hele reeks incidenten blijkt terug te voeren te zijn op de voorbereiding van het vorige artikel, toen ik een database db1a\_1 aan het maken was voor een Data Guard test. Ik had

toen in eerste instantie te weinig geheugen aan mijn virtual machine toegewezen om een database te kunnen maken. Dat resulteerde in de foutmeldingen die hierboven worden getoond.

Het diagnostics tool werkt dus niet alleen voor de actuele database, maar op alle databases waarvan logfiles onder de betreffende \$ORACLE\_BASE/diag beschikbaar zijn. Dat kan erg handig zijn als er gebruik gemaakt wordt van RAC, en bovendien de diagnostics directory op een gedeeld filesystem staat. In dat geval kunnen alle instances tegelijkertijd worden gediagnosticeerd. Het voert te ver om in het kader van dit artikel het hele adrci tool te bespreken. Hoewel documentatie over de directorystructuur niet te vinden is, wordt adrci wel uitvoerig beschreven in de 'Oracle database utilities' manual.

Samenvattend: de documentatie is nog niet compleet, de gewijzigde directorystructuur lijkt nog niet helemaal consequent doorgevoerd, maar het geheel ziet er interessant uit. Er worden wel wat resources verspild door de alert logfile twee keer op te slaan, één keer op de klassieke manier en één keer in xml-formaat, maar dat zal in de praktijk niet gauw tot problemen leiden. De mogelijkheid om snel alle relevante informatie bij een incident in te pakken om naar Oracle Support te sturen maakt het leven van de DBA bij het oplossen van incidenten een stukje makkelijker, en voorkomt (als het goed is) een hoop ergernis bij de samenwerking met Oracle Support. We hebben allemaal de ervaringen van moeizame supportrequests, waarbij de supportmedewerker vraagt om al eerder verstrekte informatie. Nu een Oracle-tool die informatie automatisch verzamelt mogen we hopen dat alle relevante informatie meteen bekend is, en een supportrequest voortvarend inhoudelijk kan worden opgepakt.

## Virtual columns

De mogelijkheid om virtuele kolommen in een tabel te kunnen opnemen is een lang gekoesterde wens. Met een virtuele kolom wordt een van andere gegevens afgeleide kolom bedoeld.

Eigenlijk is een tabel met virtuele kolommen een kruising tussen een tabel en een view. Een voorbeeld waar ik in de luchthavenindustrie veel mee te maken had was de tijdrepresentatie. Tijd, tijdzones en zeker zomertijdperiodes zijn ondingen in softwareontwikkeling als er internationaal moet worden gewerkt. Het gegeven '28-10-2007 02:15:00' is onmogelijk eenduidig te interpreteren. Zelfs als ik er bij zeg dat het de lokale tijd in Nederland betreft maakt interpretatie onmogelijk. Om 03:00:00 is de klok namelijk een uur terug gezet. Het is op 28 oktober in ons land dus twee keer kwart over twee 's nachts geweest. Welke keer wordt hierboven nu bedoeld? Dat is niet te bepalen. Om deze reden wordt in internationaal georiënteerde systemen niet met lokale tijd gewerkt, maar met de wereldwijde standaard Coordinated Universal Time, op z'n Frans afgekort tot UTC.

Deze tijd is gelijk aan de tijd op de nulmeridiaan, ofwel de Greenwich tijd, zonder zomer- of wintertijd correcties. Samen met de tijdzone kan van UTC altijd de juiste lokale tijd worden afgeleid, andersom is niet mogelijk zonder toegevoegde informatie.

In de luchthavensystemen waar ik bij betrokken was is er uiteindelijk voor gekozen om de tijd in de twee representaties redundant op te slaan: naast de kolom met UTC-representatie werd een extra kolom gedefinieerd die de lokale tijd bevat. Die extra kolom werd dan via een trigger met de gecorrigeerde tijd gevuld, rekening houdend met zomer en wintertijd. Die begin- en einddatum van de zomertijd is overigens ook niet zomaar in een formule te vangen: deze datums zijn onderhevig aan politieke besluitvorming. In 1996 is de einddatum van de zomertijd in o.a. Nederland van september naar oktober verschoven, en dit jaar is de begindatum in de V.S. van de eerste zondag in april naar de tweede zondag in maart verplaatst. Maar ik dwaal af, het gaat om virtuele kolommen.

Als ik een tabel maak met een kolom van het 'date' type, en ik vul deze met UTC-tijden, dan kan ik in een virtuele kolom de lokale tijd 'opslaan'.

Even voorbijgaand aan tijdzonefunctionaliteit van Oracle zelf zou zo'n tabel als volgt worden gedefinieerd:

```
ALTER SESSION SET NLS_DATE_FORMAT='ddmmyyyy hh24miss'
/

DROP TABLE utc_lt
/

CREATE OR REPLACE FUNCTION utc2lt(utc IN DATE) RETURN DATE
DETERMINISTIC IS
  l_lt DATE;
BEGIN
  IF utc BETWEEN TO_DATE('26032006 020000')
    AND TO_DATE('29102006 025959')
  THEN l_lt := utc + 2/24;
  ELSIF utc BETWEEN TO_DATE('29102006 030000')
    AND TO_DATE('25032007 015959')
  THEN l_lt := utc + 1/24;
  ELSIF utc BETWEEN TO_DATE('25032007 020000')
    AND TO_DATE('28102007 025959')
  THEN l_lt := utc + 2/24;
  ELSIF utc BETWEEN TO_DATE('28102007 030000')
    AND TO_DATE('30032008 015959')
  THEN l_lt := utc + 1/24;
  END IF;
  RETURN l_lt;
END;
/

CREATE TABLE utc_lt
(id NUMBER NOT NULL
,utc DATE NOT NULL
```

```
,lt AS (utc2lt(utc)
)
)
/

INSERT INTO utc_lt (id ,utc)
VALUES (1 , '01012007 020000');

INSERT INTO utc_lt (id ,utc)
VALUES (2 , '01042007 020000');

INSERT INTO utc_lt (id ,utc)
VALUES (3 , '01012008 020000');

INSERT INTO utc_lt (id ,utc)
VALUES (4 , '01102008 020000');
```

Als de tabel met het script dat hierboven is afgebeeld is aangeemaakt en gevuld is de inhoud als volgt:

```
SQL> SELECT * FROM utc_lt;

          ID UTC                LT
-----
1 01012007 020000 01012007 030000
2 01042007 020000 01042007 040000
3 01012008 020000 01012008 030000
4 01102008 020000
```

De eerste en derde rij vallen in de wintertijd: het verschil tussen UTC en lokale tijd (LT) is 1 uur. De tweede rij valt in de zomertijd met een verschil van 2 uur tussen UTC en LT. De vierde rij heeft geen LT: de gedefinieerde functie bevat geen definitie voor 2008.

Dit is maar een eenvoudig voorbeeld van de inzet van een virtual column. Met een view zou het ook kunnen, of, zoals genoemd, door een trigger te definiëren om een echte kolom te vullen.

Lucas Jellema beschrijft in de technology blog van AMIS (<http://technology.amis.nl/blog/?p=2427>) uitgebreid hoe virtual columns nog meer kunnen worden ingezet zoals voor het definiëren van bijzonder krachtige constraints.

Er moet bij het geheel echter wel een kanttekening gemaakt worden, die ik bij de verschillende op internet aanwezige beschrijvingen van de virtual column-functionaliteit niet of nauwelijks tegenkom: de inhoud van een virtual column wordt iedere keer als deze wordt gelezen opnieuw bepaald. Des te complexer de achterliggende definitie, des te meer (CPU)-resources worden verbruikt om die waarde te bepalen. Met name in situaties waar ontwikkelaars gemakzuchtig terugvallen

op het helaas veel te vaak voorkomende 'select \* from <tabel>', ook als ze maar in één of een paar kolommen van de rij zijn geïnteresseerd, zal heel veel overbodig werk moeten worden gedaan door de database om de waarde van de virtuele kolom te bepalen.

Virtuele kolommen bieden een welkome aanvulling op de mogelijkheden die views en triggers bieden. De mogelijkheid partitioning toe te passen op basis van virtual columns is handig voor omvangrijke data warehouses, waar virtual columns voor aanzienlijke ruimtebesparing kunnen zorgen. Voor het gebruik ter voorkoming van redundante, met triggers bijgewerkte kolommen zal een zorgvuldige afweging moeten worden gemaakt m.b.t. het resourcegebruik. Voor een veelvuldig geraadpleegde, maar niet zo vaak wijzigende kolom zal mogelijk de eenmalig te berekenen en redundante vastgelegde kolom beter uitpakken dan de virtuele variant die bij iedere query weer opnieuw moet worden berekend.

## Data Recovery Advisor

Rman is uitgebreid met een geautomatiseerde Data Recovery Advisor. Een verloren gegane datafile of een corrupt blok in een datafile is voor veel DBA's een hachelijke situatie. Dat komt vooral om dat het veel te weinig voorkomt. Hoe positief dat ook is voor de beschikbaarheid van het systeem, het is er wel de oorzaak van dat bij veel DBA's de routine met betrekking tot restore- en recovery-operaties ontbreekt. Dat bleek ook toen ik medio 2006 docent was bij Miracle in Kopenhagen tijdens het 3-daagse Miracle High Performance Downtime Bootcamp. In het kort: er waren 20 ervaren DBA's, die met virtual machines op laptops in teams hun systeem zoveel mogelijk in de lucht moesten houden, met zo weinig mogelijk downtime. Er was een nadrukkelijk competitie-element ingebouwd, om de druk vanuit het management tijdens recovery operaties zo goed mogelijk na te bootsen. Presentaties door de 5 docenten werden afgewisseld met oefeningen. Helaas voor de deelnemers verliepen die niet altijd even soepel: Via achterdeurtjes in de systemen werd door de docenten steeds weer een nieuwe corruptie of verstoring geïntroduceerd. De teams moesten daarbij steeds afwegingen maken tussen snel herstel van de functionaliteit, en achteraf toevoegen van verloren gegane data, of direct volledig herstel met een mogelijk langer niet beschikbare database als consequentie. De cursisten waren zoals genoemd ervaren tot zeer ervaren, en vaak verantwoordelijk voor vele databases van hun werkgever. Toch bleek ook hier steeds weer dat snel en accuraat gegevensherstel in hoge mate afhangt van de ervaring en routine van de DBA. Verstoringen zijn vaak 'nieuw' voor de beheerder, en het zoeken van de juiste oplossing onder druk leidt vaak tot fouten. Een cursist, werkzaam bij één van de grotere Oraclegebruikers in Europa, beklaagde zich halverwege de eerste ochtend al over het

gebrek aan niveau van de training. Hij bleef, en was bij de afsluiting van dag drie zo sportief om toe te geven dat hij zijn vaardigheden schromelijk had overschat. Onder druk had hij steeds weer fouten gemaakt, en uiteindelijk eindigde het team onder zijn leiding ver onderaan.

Volgend jaar gaan we weer zo'n training doen, maar ik zal voorstellen geen gebruik te maken van Oracle 11g, want dan is er geen eer meer aan te behalen.

Om de nieuwe Data Recovery Advisor te testen heb ik de database 11g die ik heb gemaakt in Archived Log Mode gezet, en vervolgens heb ik met RMAN een full backup op schijf gemaakt. Tot zover niets spannends. Daarna heb ik bewust twee problemen gecreëerd: ik heb datafile 4 (de datafile van de tablespace USERS weggegooid, en ik heb blok 840 van de datafile sysaux01.dbf corrupt gemaakt door byte 1025 – 3072 van dat blok de waarde 0 te geven..

Eerst heb ik datafile 4 weggegooid:

```
d11b@laphroaig:/u02/oradata/d11b>ls -l *dbf
-rw-r----- 1 oracle dba 450306048 Dec  4 01:22 sysaux01.dbf
-rw-r----- 1 oracle dba 639639552 Dec  4 01:22 system01.dbf
-rw-r----- 1 oracle dba 40902656 Dec  2 20:19 temp01.dbf
-rw-r----- 1 oracle dba 309338112 Dec  4 01:22 undotbs01.dbf
-rw-rw---- 1 oracle dba 17047552 Dec  4 01:22 users01.dbf

d11b@laphroaig:/u02/oradata/d11b>rm users01.dbf

d11b@laphroaig:/u02/oradata/d11b>ls -l *dbf
-rw-r----- 1 oracle dba 450306048 Dec  4 01:22 sysaux01.dbf
-rw-r----- 1 oracle dba 639639552 Dec  4 01:22 system01.dbf
-rw-r----- 1 oracle dba 40902656 Dec  2 20:19 temp01.dbf
-rw-r----- 1 oracle dba 309338112 Dec  4 01:22 undotbs01.dbf

d11b@laphroaig:/u02/oradata/d11b>
```

Omdat dit Linux is, blijft een door Oracle geopende file voor Oracle zelf gewoon beschikbaar. Om de file ook daadwerkelijk kwijt te raken moet óf de instance worden herstart, óf de datafile offline worden gebracht. Ik heb gekozen voor het laatste. Daarna blijkt de file ook inderdaad niet meer online te brengen.

```
SQL> alter database datafile 4 offline;

Database altered.

SQL> alter database datafile 4 online;
alter database datafile 4 online
*
ERROR at line 1:
ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: '/u02/oradata/d11b/users01.dbf'

SQL>
```

Om blok 840 van sysaux01.dbf corrupt te maken heb ik gebruik gemaakt van het Linux commando dd. Daarmee kan een bestand blok voor blok worden gekopieerd.

Opmerking: dd begint te tellen bij blok 1, Oracle start met blok 0. Daardoor zijn de bloknummers in de beschrijving van het corruptie-proces 1 hoger.

Eerst kopieer ik de eerste 840 blokken van 8kB naar een bestand 'f1'. Vervolgens wordt omgeschakeld naar een blok-grootte van 1kB, en wordt de eerste kB van blok 841 naar een bestand 'f2' gekopieerd. Dan wordt 2kB aan nullen weggeschreven naar 'f3'. Daarbij wordt gebruik gemaakt van het Linux pseudobestand /dev/zero, dat een oneindige hoeveelheid nullen kan leveren. De laatste 5kB van blok 841 wordt in bestand 'f4' gezet. Tenslotte wordt sysaux01.dbf vanaf blok 842 tot het eind naar bestand 'f5' gekopieerd.

Na deze losse kopieeracties worden de bestanden f1-f5 weer achter elkaar gezet en over sysaux01.dbf heen gekopieerd. Het script ziet er als volgt uit:

```
d11b@laphroaig:/u02/oradata/d11b>cat corrupt_sysaux
# Neem de eerste 840 8k blokken
dd if=sysaux01.dbf of=f1 skip=0 bs=8192 count=840

# Neem de eerste 1k van blok 841
dd if=sysaux01.dbf of=f2 skip=6720 bs=1024 count=1

# Schrijf 2k met zeros
dd if=/dev/zero of=f3 bs=1024 count=2

# Neem de resterende 5k van blok 841
dd if=sysaux01.dbf of=f4 skip=6723 bs=1024 count=5

# Schrijf de rest vanaf blok 842
dd if=sysaux01.dbf of=f5 skip=841 bs=8192

# Zet het resultaat terug over sysaux01.dbf
cat f* >sysaux01.dbf
```

Het script wordt uitgevoerd:

```
40+0 records in
840+0 records out
6881280 bytes (6.9 MB) copied, 0.264001 seconds, 26.1 MB/s
1+0 records in
1+0 records out
1024 bytes (1.0 kB) copied, 0.000198273 seconds, 5.2 MB/s
2+0 records in
2+0 records out
2048 bytes (2.0 kB) copied, 0.000937869 seconds, 2.2 MB/s
5+0 records in
5+0 records out
5120 bytes (5.1 kB) copied, 0.000249992 seconds, 20.5 MB/s
54128+0 records in
54128+0 records out
443416576 bytes (443 MB) copied, 27.9073 seconds, 15.9 MB/s
```

Nu starten we RMAN via de command line interface en gebruiken het commando 'list failure' om te kijken of er problemen zijn:

```
d11b@laphroaig:/home/oracle>rman target=/

Recovery Manager: Release 11.1.0.6.0 - Production on Tue Dec 4 01:41:00
2007

Copyright (c) 1982, 2007, Oracle. All rights reserved.

connected to target database: D11B (DBID=134166940)

RMAN> list failure;

using target database control file instead of recovery catalog
List of Database Failures
=====

Failure ID Priority Status Time Detected Summary
-----
222 HIGH OPEN 04-DEC-07 One or more non-system
datafiles are missing

RMAN>
```

De instance heeft nog helemaal geen weet van het corrupte blok, totdat er misschien iets uit gelezen of in geschreven moet worden. RMAN heeft echter de mogelijkheid de integriteit van de datafiles te controleren.

Omdat we weten dat niet alle datafiles aanwezig zijn, moeten we ontbrekende datafiles overslaan. Het commando voor deze controle wordt dan :

```
RMAN> validate check logical skip inaccessible database;

Starting validate at 04-DEC-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=128 device type=DISK
could not access datafile 4
skipping inaccessible file 4
RMAN-06060: WARNING: skipping datafile compromises tablespace USERS
recoverability
RMAN-06060: WARNING: skipping datafile compromises tablespace USERS
recoverability
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00001 name=/u02/oradata/d11b/system01.dbf
input datafile file number=00002 name=/u02/oradata/d11b/sysaux01.dbf
input datafile file number=00003 name=/u02/oradata/d11b/undotbs01.dbf
channel ORA_DISK_1: validation complete, elapsed time: 00:00:55
List of Datafiles
=====

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
2 FAILED 0 20220 54968 538314
File Name: /u02/oradata/d11b/sysaux01.dbf
```

```

Block Type Blocks Failing Blocks Processed
-----
Data          0          7637
Index         1          6331
Other         0          20780

validate found one or more corrupt blocks
See trace file /opt/oracle/diag/rdbms/d11b/d11b/trace/d11b_ora_3561.trc
for details
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
including current control file for validation
including current SPFILE in backup set
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
List of Control File and SPFILE
=====
File Type      Status Blocks Failing Blocks Examined
-----
SPFILE         OK          0          2
Control File OK          0          594
Finished validate at 04-DEC-07

RMAN>

```

(Uit de resultaten zijn de datafiles zonder corruptie weggelaten om ruimte te besparen)

Als ik nu opnieuw het commando 'list failure' uitvoer is de blok-corruptie ook bekend:

```

RMAN> list failure;

List of Database Failures
=====

Failure ID Priority Status      Time Detected Summary
-----
306
HIGH
OPEN      04-DEC-07      Datafile 2: '/u02/oradata/d11b/sysaux01.dbf'
contains one or more corrupt blocks
222      HIGH      OPEN      04-DEC-07      One or more non-system
datafiles are missing

RMAN>

RMAN> RMAN> advise failure;

List of Database Failures
=====

```

De volgende stap is het comand 'advise failure'. RMAN bekijkt dan welke backups beschikbaar zijn om de problemen te verhelpen, en genereert meteen een script dat kan worden gebruikt om de oplossing uit te voeren. Eventueel kan de DBA het script nog aanpassen.

```

Failure ID Priority Status      Time Detected Summary
-----
306
HIGH
OPEN      04-DEC-07      Datafile 2: '/u02/oradata/d11b/sysaux01.dbf'
contains one or more corrupt blocks
222      HIGH      OPEN      04-DEC-07      One or more non-system
datafiles are missing

analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
1. If file /u02/oradata/d11b/users01.dbf was unintentionally renamed or
moved, restore it

Automated Repair Options
=====
Option Repair Description
-----
1      Restore and recover datafile 4; Perform block media recovery of
block 840 in file 2
Strategy: The repair includes complete media recovery with no data
loss
Repair script: /opt/oracle/diag/rdbms/d11b/d11b/hm/reco_4043853707.hm

RMAN>

```

De laatste stap is het uitvoeren van de reparatie met het commando 'repair failure':

```

RMAN> repair failure;

Strategy: The repair includes complete media recovery with no data loss
Repair script: /opt/oracle/diag/rdbms/d11b/d11b/hm/reco_4043853707.hm

contents of repair script:
# restore and recover datafile
sql 'alter database datafile 4 offline';
restore datafile 4;
recover datafile 4;
sql 'alter database datafile 4 online';
# block media recovery
recover datafile 2 block 840;

Do you really want to execute the above repair (enter YES or NO)? YES
executing repair script

sql statement: alter database datafile 4 offline

Starting restore at 04-DEC-07
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set

```

```

channel ORA_DISK_1: restoring datafile 00004 to /u02/oradata/d11b/
users01.dbf
channel ORA_DISK_1: reading from backup piece /u02/oradata/d11b/rman/
rman_D11B_06j2naj3_1_1.bus
channel ORA_DISK_1: piece handle=/u02/oradata/d11b/rman/rman_
D11B_06j2naj3_1_1.bus tag=TAG20071203T234435
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 04-DEC-07

Starting recover at 04-DEC-07
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 54 is already on disk as file
/u02/oradata/d11b/arch/1_54_640288412.dbf
archived log for thread 1 with sequence 55 is already on disk as file
/u02/oradata/d11b/arch/1_55_640288412.dbf
.
.
archived log for thread 1 with sequence 67 is already on disk as file
/u02/oradata/d11b/arch/1_67_640288412.dbf
archived log file name=/u02/oradata/d11b/arch/1_54_640288412.dbf
thread=1 sequence=54
.
.
archived log file name=/u02/oradata/d11b/arch/1_64_640288412.dbf
thread=1 sequence=64
media recovery complete, elapsed time: 00:01:12
Finished recover at 04-DEC-07

sql statement: alter database datafile 4 online

Starting recover at 04-DEC-07

```

```

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 00002
channel ORA_DISK_1: reading from backup piece /u02/oradata/d11b/rman/
rman_D11B_06j2naj3_1_1.bus
channel ORA_DISK_1: piece handle=/u02/oradata/d11b/rman/rman_
D11B_06j2naj3_1_1.bus tag=TAG20071203T234435
channel ORA_DISK_1: restored block(s) from backup piece 1
channel ORA_DISK_1: block restore complete, elapsed time: 00:00:16

starting media recovery
media recovery complete, elapsed time: 00:00:07

Finished recover at 04-DEC-07
repair failure complete

RMAN>

```

Daarmee is de hele hersteloperatie uitgevoerd en de database weer volledig beschikbaar, zonder dataverlies.

De Data Recovery Advisor zal heel wat consultancy overbodig maken. Een simpel lijstje commando's, 'list failure', 'advise failure', 'repair failure', eventueel aangevuld met het 'validate database' commando zal veel DBA's een gerust gemoed en een goede nachtrust garanderen, tenminste, zolang er goede backups gemaakt worden...

OraVision bouwt Oracle-oplossingen waarin documenten, transacties en bestaande systemen samenwerken. OraVision staat bekend als *the mid-office company*. OraVision bouwt vanuit haar geheel eigen visie: kwaliteit staat centraal.



### Kwaliteit in kennis

OraVision beschikt over enorme ervaring in Oracle-, Java- en integratietechnologieën. Bij ons staat de techniek echter nooit op zichzelf. Juist bij mid-office en document-integratie toepassingen laten we de technologie tot volle bloei komen.

### Kwaliteit in werk

Klanten geven OraVision al jaren het vertrouwen om geavanceerde ICT-toepassingen te realiseren die tegelijk gebruikersvriendelijk zijn. Onze mid-office oplossingen bevinden zich immers in het hart van elke bedrijfsvoering.

### Kwaliteit in samenwerking

Bij OraVision staat niet alleen technische kwaliteit hoog in het vaandel, ook onze stijl is onderscheidend. Vanuit onze Limburgse basis investeren we nadrukkelijk in persoonlijke relaties en genieten van het goede leven.

Geïnteresseerd in de visie van OraVision op Oracle, Java, integratie en mid-office? Bezoek [www.oravision.com](http://www.oravision.com) en abonneer u gratis op de OraVisionair.

