Oracle

- BASH-like command line history für Oracle
- sqlplus
- Wichtige Dateien von Oracle
- Manueller Start/Stop der Oracle-Datenbank
- Oracle Mini-Tips
- Parameter-Files spfile pfile
- Passwort für User SYS (SYSDBA) ändern
- TNSNAMES und EZCONNECT
- Installation
    - Oracle 11g Installation
    - Oracle 10g2 Installation

- Backup und Recovery
    - Exportieren einer bestimmten Tabelle
    - Recovery eines Controlfiles
    - Recovery der Redo-Logs
    - Allgemeines zu RMAN
    - RMAN - Aufruf und Connect
    - RMAN Format Codes
    - Vollständiges Recovery

BASH-like command line history für Oracle

wer viel mit SQLPLUS oder RMAN an der Kommandozeile arbeitet vermisst sicherlich die aus der BASH bekannten History-Funktionen und so nützliche Tasten wie Pfeil-Hoch oder STRG+R (Suche in der History).

Folgende Tools bringen hier Abhilfe, beide sind Wrapper um die Oracle-Befehle sqlplus und rman und können parallel installiert werden:

- rlwrap - bringt History und Bash-ähnliche Edit-Funktionen
- gqlplus - bringt zusätzlich zu den Funktionen von rlwrap eine Tab-Completion für Tabellennamen usw. (das Tool wirft aber ab und an mal Fehler)

Installation von rlwrap (falls der Server down ist, hier der Download):

```
wget http://utopia.knoware.nl/~hlub/uck/rlwrap/rlwrap-0.28.tar.gz
tar xvfz rlwrap-0.28.tar.gz
cd rlwrap-0.28
./configure && make
```

jetzt muss man sich noch entscheiden, ob man das Tool systemweit braucht, oder ob eine Installation in das $ORACLE_HOME ausreicht.
 systemweite Installation (als root) installiert das Binary und die Man-Page:

`make install`

soll das nur in $ORACLE_HOME liegen, reicht folgendes (als User oracle):

`cp src/rlwrap $ORACLE_HOME/bin/`

um zukünftig mit dem Wrapper zu arbeiten muss er noch in die Datei ~/.bashrc des Oracle-Users als Alias eingetragen werden (einfach in die Datei einfügen):

```
alias sqlplus='rlwrap sqlplus'
alias rman='rlwrap rman'
```

Jetzt die Installation von gqlplus:

```
wget http://mesh.dl.sourceforge.net/sourceforge/gqlplus/gqlplus-1.12.tar.gz
tar xvfz gqlplus-1.12.tar.gz
cd gqlplus-1.12
./configure && make
cp gqlplus $ORACLE_HOME/bin/
```

gqlplus kann jetzt einfach als Ersatz zu sqlplus aufgerufen werden:

```
gqlplus / as sysdba
```

sqlplus

um als User SYSDBA mit sqlplus (CLI für Oracle) zu arbeiten:

```
[oracle@db1-ext oracle]$ sqlplus / as sysdba

SQL*Plus: Release 9.2.0.4.0 - Production on Thu Mar 1 07:32:29 2007

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.4.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.4.0 - Production

SQL>
```

bei Oracle 9i und kleiner muss man noch Anführungszeichen setzen:

```
[oracle@db1-ext oracle]$ sqlplus "/ as sysdba"
```

sqlplus Spaltenbreite und weitere Einstellungen festlegen:

sofern es im Userhome eine Datei login.sql gibt, wird dessen Inhalt gleich beim Starten von sqlplus gesetzt

```
define _editor=vi
set linesize 120
set pagesize 24
set timing on;
break on overload skip page
col overload format a8
col member form a40
#alter session set nls_date_format='dd-mm-yyyy hh24:mi:ss';
```

Ausführzeit eines Statements mit ausgeben:

```
set timing on;
```

Wichtige Dateien von Oracle

alles von einer frischen Standard-Installation ausgehend und dass:

```
$ORACLE_BASE=/u01/app/oracle
$ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
$ORACLE_SID=orcl
```

Alert-Logfile:

```
$ORACLE_BASE/admin/orcl/bdump/alert_$ORACLE_SID.log
```

Datenbank-, Control-, Redo-Log-, Undo-Log-Files:

```
$ORACLE_BASE/oradata/orcl
```

Konfigurationsfiles Listener, TNS-Names und sqlnet:

```
$ORACLE_HOME/network/admin
```

Archive- und Flashbacklogs:

```
$ORACLE_BASE/flash_recovery_area/$ORACLE_SID
```

Start-/Stop-Script von Oracle:

```
$ORACLE_HOME/bin/dbstart
$ORACLE_HOME/bin/dbshut
```

Manueller Start/Stop der Oracle-Datenbank

mit folgenden Befehlen werden:

- die Datenbank
- der Listener
- die isql-Weboberfläche
- und der Oracle-Enterprise-Manager gestartet

```
$ sqlplus / as sysdba
SQL> startup
SQL> exit
$ lsnrctl start
$ isqlplusctl start
$ emctl start dbconsole
```

stoppen mit diesen Befehlen:

```
$ emctl stop dbconsole
$ isqlplusctl stop
$ lsnrctl stop
$ sqlplus / as sysdba
SQL> shutdown immediate
SQL> exit
```

Datenbank Schritt für Schritt starten:

```
1. $ sqlplus / as sysdba
2. SQL> startup nomount          NOMOUNT-Phase
3. SQL> alter database mount     MOUNT-Phase
4. SQL> alter database open      OPEN-Phase
```

Kurzüberblick:

```
DB starten:                  SQL> startup;
DB stoppen:                  SQL> shutdown immediate;
DB in nomount-Phase starten: SQL> startup nomount;
DB mounten:                  SQL> alter database mount;
DB öffnen:                   SQL> alter database open;
```

```
DB restricted öffnen:        SQL> startup open restrict;
DB wieder normal öffnen:     SQL> alter system disable restricted session;
Controlfiles zeigen:         SQL> show parameter control_files;

Einen User trotz restricted Mode die Nutzung der Datenbank erlauben:
SQL> grant restricted session to <user>;
```

Oracle Mini-Tips

Zeitangaben mit SYSDATE:

```
SYSDATE-1 = 1 Tag zurück
SYSDATE-1/24 = 1 Stunde zurück
SYSDATE-3/24 = 3 Stunden zurück
```

Flashback und Restorepunkte:

```
SQL> create restorepoint db-ok;
SQL> select * from v$restorepoint;
SQL> flashback to restorepoint db-ok;
SQL> drop restorepoint db-ok;
```

aktueller Stand DB-File und Angabe aus den DB-Fileheadern:

```
select file#, checkpoint_change# from v$datafile;
select file#, checkpoint_change# from v$datafile_header;
```

Flashback-Beispiel:

```
SQL> update scott.emp T1 set sal = (select sal from scott.emp as of scn 699417 where T1.empno
= empno);
```

Redo-Logfiles spiegeln:

```
alter database add logfile member '/u01/app/oracle/oradata/orcl/redo04a.log' reuse to group 4;
alter system switch logfile;
alter system checkpoint;
```

Parameter im spfile ändern:

```
SQL> alter system set param=bla [scope=both|spfile|memory];
```

Neues Passwort für SYS:

```
$ cd $ORACLE_HOME/dbs
$ orapwd file=orapw<$ORACLE_SID> password=<akt.passwort>
```

neue Datenbank anlegen mit:

```
$ dbca
```

Liste der vergebenen Ports ankucken (Ports können mit dem Tool 'emca' geändert werden):

```
$ cat $ORACLE_HOME/install/portlist.ini
```

Einen User unlocken:

```
SQL> alter user <user> account unlock;
```

Passwort eines Users ändern:

```
SQL> alter user <user> identified by '<pass>';
```

Alert-Logfile der Datenbank:

```
$ORACLE_BASE/admin/<$ORACLE_SID>/bdump/alert_<$ORACLE_SID>.log
```

Status der Datenbank:

```
SQL> select open_mode from v$database;
```

Status der Tablespaces:

```
SQL> select STATUS from dba_tablespaces;
```

Im Usernamen erlaubte Zeichen:

```
A-Z, 0-9, $, _, -
```

Password eines Users ändern (in sql):

```
SQL> alter user scott identified by "<neues Passwort>";
```

Nachfragen beim Recovery verhindern:

```
SQL> set autorecovery on;
```

aktuelle Uhrzeit ausgeben:

```
SQL> select sysdate from dual;
```

Parameter-Files spfile pfile

> die in diesem Beitrag verwendeten Variablen $ORACLE_HOME und $ORACLE_SID müssen durch die richtige Pfade und Namen ersetzt werden, da diese Variablen in SQLPLUS nicht bekannt sind!

Parameterfiles liegen normalerweise in folgendem Ordner:

```
$ORACLE_HOME/dbs/init<$ORACLE_SID>.ora
$ORACLE_HOME/dbs/spfile<$ORACLE_SID>.ora
```

die sp-files sind binär und sollten nicht via Editor geändert werden.

mit folgendem Befehl kann man bei laufender Instanz ein p-file erzeugen (aus dem aktuellen sp-file):

```
SQL> create pfile from spfile;
```

mit diesem Befehl wird aus einem p-file wieder ein sp-file erzeugt (der Dateiname kann auch weggelassen werden):

```
SQL> create spfile='<$ORACLE_HOME>/dbs/spfile<$ORACLE_SID>.ora' from
pfile='<$ORACLE_HOME>/dbs/init<$ORACLE_SID>.ora';
```

Oracle-Instanz von p-file starten:

```
$ sqlplus / as sysdba
SQL> startup pfile=/<$ORACLE_HOME>/dbs/init<$ORACLE_SID>.ora
```

Änderungen im sp-file (default ist: scope=both, um die Änderungen nicht fest einzutragen: scope=memory, um die Änderungen im sp-file fest einzutragen: scope=spfile), z.B.:

```
SQL> alter system set open_cursors=300 scope=spfile;
```

Wie kann ich feststellen, ob die Datenbank via sp-file oder via p-file gestartet wurde?

```
SQL> show parameter spfile; --> wenn gesetzt wurde normal gestartet, wenn nicht dann wurde die
DB via pfile gestartet
```

Hilfe, mein sp-file ist kaputt und ich habe auch kein aktuelles p-file oder Backup mehr! Sofern das sp-file nicht komplett weg ist, und z.B. nur ein Parameter nicht gesetzt/geladen werden kann, kann man auch einfach den Binärteil des sp-files mittels Editor entfernen, danach die Datenbank mit dem eben erzeugten p-file starten und danach ein neues spfile erzeugen.

Passwort für User SYS (SYSDBA) ändern

Solange man direkten Zugriff auf den Oraclehost hat, läßt sich das Kennwort für den User SYS problemlos ändern.

```
# su - oracle
# sqlplus / as sysdba
SQL> alter user sys identified by "<new password>";
SQL> quit
```

TNSNAMES und EZCONNECT

Verbinden zur Datenbank

via TNSNAMES:

```
$ sqlplus scott/tiger@$ORACLE_SID
```

via EZCONNECT (etwa EasyConnect):

```
$ sqlplus scott/tiger@hostname.domain.de[:1521]/$ORACLE_SID
```

Installation

Oracle 11g Installation

Diese Anleitung wurde für CentOS 5 geschrieben und getestet. Sie sollte aber auf andere Systeme übertragbar sein.

- Anforderungen: mind. 1GB MEM, Swap mind. 1,5*MEM, mind. 4GB HDD rein für die Grundinstallation, kein DHCP.
- System für die Installation vorbereiten (als root ausführen)

```
#!/bin/sh

yum -y install gcc binutils make libaio compat-libstdc++-296 compat-libstdc++-33 libaio-devel
sysstat unixODBC unixODBC-devel

groupadd -g 1000 oracle
groupadd -g 1001 oinstall
groupadd -g 1002 dba
useradd  -u 1000 -p oracle -g oinstall -G dba oracle

mkdir /oracle
mkdir /oracle/11g
chown -R oracle.oinstall /oracle
mkdir /osource

cat >> /etc/sysctl.conf << EOF

#Added for Oracle Installation
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 4194304
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 262144
EOF
/sbin/sysctl -p
```

```
cat >> /home/oracle/.bash_profile << EOF

#Added for Oracle Installation
umask 022
ORACLE_BASE=/oracle
ORACLE_HOME=/oracle/11g
ORACLE_SID=orcl
export ORACLE_BASE ORACLE_HOME ORACLE_SID
PATH=/oracle/11g/bin:$PATH:.
LD_LIBRARY_PATH=$ORACLE_HOME/jdk/jre/lib/i386:$ORACLE_HOME/jdk/jre/lib/i386/server:$ORACLE_HOM
E/rdbms/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH LD_LIBRARY_PATH

# koennte Probleme machen
export NLS_LANG=american;
export NLS_DATE_FORMAT='DD.MM.YYYY HH24:MI:SS';
EOF

passwd oracle
Oracle Init-Script (Datenbank vorher in /etc/oratab aktivieren)
#!/bin/bash
# chkconfig: 345 99 10
# description: Startup Script for Oracle Databases
# /etc/init.d/oracle

export ORACLE_HOME=/oracle/11g
export PATH=$PATH:$ORACLE_HOME/bin

case "$1" in
start)
  su - oracle -c "$ORACLE_HOME/bin/dbstart $ORACLE_HOME"
  touch /var/lock/oracle
  echo "OK"
  ;;
stop)
  echo -n "Shutdown Oracle: "
  su - oracle -c "$ORACLE_HOME/bin/dbshut $ORACLE_HOME"
  rm -f /var/lock/oracle
  echo "OK"
```

```
  ;;
  *)
  echo "Usage: 'basename $0' start|stop"
  exit 1
esac

exit 0
```

wichtige URLs nach der Installation:

```
iSQL*Plus URL:
http://<domain.tld>:5560/isqlplus

iSQL*Plus DBA URL:
http://<domain.tld>:5560/isqlplus/dba

Enterprise Manager 11g Database Control URL:
http://<domain.tld>:1158/em

Database Control URL:
http://<domain.tld>:5500/em
```

Jetzt sollte man die Installation noch absichern (in der mount-Phase):

- Archivelog einschalten (alter database archivelog;)
- Flashback einschalten (alter database flashback on;)
- Flashback Zeitraum einstellen (alter system set db_flashback_retention_target = 4320;)
- Flashback Filegröße einstellen (alter system set db_recovery_file_dest_size=20G;)
- Redologs in verschiedene Verzeichnisse spiegeln
- Controlfiles an verschiedene Orte legen
- RMAN-Backup einrichten

Installation

Oracle 10g2 Installation

Diese Anleitung wurde für CentOS 4.5 geschrieben und getestet. Sie sollte aber auf andere Systeme übertragbar sein. Die Anleitung funktioniert auch mit CentOS 5, dort muss der Installer allerdings mit der Option „-ignoreSysPrereqs“ gestartet werden (thx @ http://www.oliver-schaef.de)

- Anforderungen: mind. 1GB MEM, Swap mind. 1,5*MEM, mind. 2GB HDD rein für die Grundinstallation, kein DHCP.
- System für die Installation vorbereiten (als root ausführen)

```
#!/bin/sh

yum -y install gcc binutils make libaio compat-libstdc++-296 compat-libstdc++-33

groupadd -g 1000 oracle
groupadd -g 1001 oinstall
groupadd -g 1002 dba
useradd  -u 1000 -p oracle -g oinstall -G dba oracle

mkdir /oracle
mkdir /oracle/10g
chown -R oracle.oinstall /oracle
mkdir /osource

cat >> /etc/sysctl.conf << EOF

#Added for Oracle Installation
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_max = 262144
EOF
/sbin/sysctl -p
```

```
cat >> /home/oracle/.bash_profile << EOF

#Added for Oracle Installation
umask 022
ORACLE_BASE=/oracle
ORACLE_HOME=/oracle/10g
ORACLE_SID=orcl
export ORACLE_BASE ORACLE_HOME ORACLE_SID
PATH=/oracle/10g/bin:$PATH:.
LD_LIBRARY_PATH=$ORACLE_HOME/jdk/jre/lib/i386:$ORACLE_HOME/jdk/jre/lib/i386/server:$ORACLE_HOM
E/rdbms/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH LD_LIBRARY_PATH

# koennte Probleme machen
export NLS_LANG=american;
export NLS_DATE_FORMAT='DD.MM.YYYY HH24:MI:SS';
EOF

passwd oracle
```

Oracle Init-Script (Datenbank vorher in /etc/oratab aktivieren)

```
#!/bin/bash
# chkconfig: 345 99 10
# description: Startup Script for Oracle Databases
# /etc/init.d/oracle

export ORACLE_HOME=/oracle/10g
export PATH=$PATH:$ORACLE_HOME/bin

case "$1" in
start)
  su oracle -c $ORACLE_HOME/bin/dbstart
  touch /var/lock/oracle
  echo "OK"
  ;;
stop)
  echo -n "Shutdown Oracle: "
  su oracle -c $ORACLE_HOME/bin/dbshut
```

```
  rm -f /var/lock/oracle
  echo "OK"
  ;;
  *)
  echo "Usage: 'basename $0' start|stop"
  exit 1
esac

exit 0
```

wichtige URLs nach der Installation:

```
iSQL*Plus URL:
http://<domain.tld>:5560/isqlplus

iSQL*Plus DBA URL:
http://<domain.tld>:5560/isqlplus/dba

Enterprise Manager 10g Database Control URL:
http://<domain.tld>:1158/em

Database Control URL:
http://<domain.tld>:5500/em
```

im Script $ORACLE_HOME/bin/dbstart, welches aus dem init-File aufgerufen wird, ist ein Fehler in Zeile 78:

```
ORACLE_HOME_LISTNER=/ade/vikrkuma_new/oracle
```

der richtige Pfad lautet z.B. (i.d.R. das $ORACLE_HOME):

```
ORACLE_HOME_LISTNER=/oracle/10g
```

das Script dbshut sollte auch noch gepatched werden, um den Listener beim Shutdown zu stoppen

```
cd $ORACLE_HOME/bin
```

eine Datei dbshut.patch mit folgendem Inhalt erzeugen:

```
--- dbshut.old  2007-06-20 08:59:29.000000000 +0200
+++ dbshut      2007-06-20 08:58:23.000000000 +0200
@@ -55,6 +55,23 @@
 # Save LD_LIBRARY_PATH
```

```
 SAVE_LLP=$LD_LIBRARY_PATH

+# Set this to stop Oracle Net Listener
+ORACLE_HOME_LISTNER=/oracle/10g
+if [ ! $ORACLE_HOME_LISTNER ] ; then
+  echo "ORACLE_HOME_LISTNER is not SET, unable to auto-start Oracle Net Listener"
+else
+  LOG=$ORACLE_HOME_LISTNER/listener.log
+
+  # Start Oracle Net Listener
+  if [ -f $ORACLE_HOME_LISTNER/bin/tnslsnr ] ; then
+    echo "$0: Stopping Oracle Net Listener" >> $LOG 2>&1
+    $ORACLE_HOME_LISTNER/bin/lsnrctl stop >> $LOG 2>&1 &
+    #export VER10LIST=`$ORACLE_HOME_LISTNER/bin/lsnrctl version | grep "LSNRCTL for " | cut -
d' ' -f5 | cut -d'.' -f1`
+  else
+    echo "Failed to stop Oracle Net Listene using $ORACLE_HOME_LISTNER/bin/tnslsnr"
+  fi
+fi
+
 # Set this in accordance with the platform
 ORATAB=/etc/oratab
 if [ ! $ORATAB ] ; then
```

danach einfach ein:

```
patch -p0 < dbshut.patch
```

Jetzt sollte man die Installation noch absichern (in der mount-Phase):

- Archivelog einschalten (alter database archivelog;)
- Flashback einschalten (alter database flashback on;)
- Flashback Zeitraum einstellen (alter system set db_flashback_retention_target = 4320;)
- Flashback Filegröße einstellen (alter system set db_recovery_file_dest_size=20G;)
- Redologs in verschiedene Verzeichnisse spiegeln
- Controlfiles an verschiedene Orte legen
- RMAN-Backup einrichten

Backup und Recovery

Exportieren einer bestimmten Tabelle

Mit dem Befehl „exp“ läßt sich eine bestimmte Tabelle in eine Datei exportieren. Es handelt sich allerdings um ein Binärformat, welches nur mit dem „imp“-Tool wieder in eine Oracle-Datenbank imporiert werden kann.

```
[oracle@oradb]$ /oracle/product/9.2.0/bin/exp user/pass file=dumpfile.dmp log=dumpfile.log
tables=TABELLE1
```

Recovery eines Controlfiles

1. CTL-File aus Backup wiederherstellen
2. Datenbank in mount-Phase starten:
   `SQL> startup mount`
3. Recover der Datenbank durchführen:
   `SQL> recover database until cancel using backup controlfile;`
   ? Recovery schlägt fehl, da Oracle im Archivelog sucht, aber die benötigten Infos in den Redo-Logs zu finden sind
   ? manuelle Angabe des Filenamens der Redo-Logs (evtl. müssen alle nacheinander durchprobiert werden!)
4. Wenn erfolgreich kann die Datenbank geöffnet werden:
   `SQL> alter databse open resetlogs;`

Recovery der Redo-Logs

- Verlust eines online Redo-Logs (nicht das current und inactive):

kann einfach aus einem Backup wiederhergestellt werden, danach ein

```
alter database clear logfile '<filename>';
```

- Verlust des current Redo-Logs: Worst Case: Es kann kein Recovery dieses Redo-Logs durchgeführt werden, einzig möglicher Weg ist die Wiederherstellung der kompletten Datenbank aus einem Backup.

Allgemeines zu RMAN

Alle RMAN-Einstellungen anzeigen:

```
RMAN> show all;
```

Automatisches Backup der Controlfiles einschalten:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

Schnellverbindung zu RMAN und zur Katalog-Datenbank:

```
[oracle@student3 ~]$ rman target / catalog rman/rman@repo
```

RMAN konfigurieren und einen Recovery-Katalog einrichten:

1. User anlegen und berechtigen in ext. Oracle-Datenbank für den RMAN-Catalog

```
SQL> create user rman identified by rman default tablespace rman_ts;
SQL> grant connect,resource,recovery_catalog_owner to rman;
```

2. Verbinden zur lokalen Datenbank und zum entfernten Katalog-Server:

```
[oracle@server ~]$ rman target / catalog rman/rman@repo
...
connected to target database: ORCL (DBID=1153117770)
connected to recovery catalog database
```

3. Katalog erzeugen und Datenbank mit dem Katalog registrieren:

```
RMAN> create catalog;

recovery catalog created

RMAN> register database;

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

RMAN - Aufruf und Connect

das Folgende wurde durch Hermann Brunner für unseren Oracle DBA-Kurs zusammengefasst:

Aufruf / Connect

```
$ rman
RMAN> connect target /
RMAN> connect catalog user/password@catdb
```

oder direkt beim Aufruf

```
$ rman target /
$ rman target / catalog user/password@catdb
$ rman target / log='/pfad/zum/log.log' [append] (mit Angabe einer Logdatei)
```

Einstellungen

```
RMAN> show all;
RMAN> configure [...]
```

Syntax wie in der SHOW Anzeige. Beispiele:

```
RMAN> configure retention policy to redundancy 3;
RMAN> configure controlfile autobackup on;
```

komplexere Beispiele:

```
RMAN> configure device type disk parallelism 3 backup type backupset;
RMAN> configure datafile backup copies for device type sbt to 2;
```

Channels vorkonfigurieren:

```
RMAN> configure channel device type disk format '/pfad/%U';
RMAN> configure channel n device type disk format '/home/backup_m/%d_%t_%s.bck'
```

Einstellungen auf default zurücksetzen:

```
RMAN> configure retention policy clean;
RMAN> configure controlfile autobackup clear;
```

Recovery Catalog

Recovery Catalog einrichten:

```
auf dem Katalog-Server:
SQL> create tablespace ts_rman datafile '/pfad/rman01.dbf' size 100m; ## Richtgröße 100MB pro Zi
SQL> create user rman identified by 'passwd' default tablespace ts_rman;
SQL> grant connect, resource, recovery_catalog_owner to rman;
SQL> grant select any dictionary to rman;

auf dem Ursprungs-Server:
$ rman target / catalog rman/rman@catdb ## (evtl. tnsnames.ora anpassen)
RMAN> create catalog;
RMAN> register database;

RMAN> report schema;
RMAN> report schema at time 'sysdate -3' ## (wie war die DB-Struktur vor 3 Tagen?)
```

Maintenance commands für den Katalog:

```
SQL> connect rman/rman@catdb;
SQL> select * from cat;

RMAN> resync catalog;
RMAN> unregister database;
RMAN> upgrade catalog; ## Katalog auf den Stand des rman-Clients bringen
RMAN> drop catalog;

RMAN> catalog backuppiece 'filename';
RMAN> catalog controlfilecopy 'filename';
RMAN> catalog datafilecopy 'filename';
RMAN> catalog archivelog 'filename';
RMAN> catalog recovery area noprompt;
RMAN> catalog start with '/u01/fra/archive' ## Start-Pfad, ab dem gesucht werden soll
```

Backups erstellen

```
RMAN> backup database;
RMAN> backup tablespace ts_name;
RMAN> backup datafile n;
RMAN> backup datafile df_name;
RMAN> backup current controlfile;
RMAN> backup spfile;
RMAN> backup archivelog [from sequence=nnn] [delete {ALL} input];

RMAN> backup as backupset [...];
RMAN> backup as compressed backupset [...];
RMAN> backup as copy [...];
RMAN> backup validate [...]; ## füllt die v$database_block_corruption Liste
RMAN> backup check logical [...];

RMAN> backup [...] plus archivelog;
RMAN> backup [...] include controlfile;
RMAN> backup [...] format '/pfad/zum/backup';
RMAN> backup [...] tag='freier Text';
RMAN> backup [...] not backed up nnn times;
RMAN> backup [...] duration hh:mm minimize load;
RMAN> backup [...] delete [ALL] input;
```

Alte Version 8i/9i Syntax:

```
RMAN> run {
            ALLOCATE CHANNEL c1 DEVICE TYPE disk FORMAT='/home/oracle/backup1/datafile_5.bck';
            BACKUP DATAFILE 5;
            RELEASE CHANNEL c1; }  ## nur in rman8 nötig
```

Sinnvolle Technik für Full Backup mit anschließender Sicherung der Archive-Logs:

```
RMAN> backup database;
RMAN> sql 'alter system archive log current';
```

```
RMAN> backup archivelog all;
```

Seit Oracle 9i gehts einfacher:

```
RMAN> backup database plus archivelog; ## alternativ mit Option: delete [all] input
```

Zweistufiges Backup - z.B. Sicherung aller DISK Backups auf Band - Variante 1:

```
RMAN> backup device type disk as backupset database;
Optional:
RMAN> [...] plus archivelog;
RMAN> [...] delete input;
```

Zweistufiges Backup - z.B. Sicherung aller DISK Backups auf Band - weitere Varianten:

```
RMAN> backup backupset completed before 'sysdate -1';
RMAN> backup copy of database;
```

Trickreichere rman Commands:

```
RMAN> configure device type disk parallelism 4 backup type to copy;
RMAN> backup database; ## erzeugt einzelne Datafiles in fra/datafile...
```

Incrementelles Backup:

```
RMAN> backup [as backupset] incremental level 0 [database|tablespace|datafile];
RMAN> backup [as backupset] incremental level n [cumulative] [database|tablespace|datafile];
```

Incrementelles Updating einer Copy die mit <> gekennzeichneten Optionen sind nur in Oracle 9 nötig!

```
RMAN> backup as copy <incremental level 0> [database|tablespace|datafile] <format='somewhere/som
RMAN> backup incremental level 1 <for recover of copy with tag='TEST'> [database|tablespace|data
RMAN> recover copy of [database|tablespace|datafile] <with tag='TEST'>;
```

Restore / Recover Commands

```
RMAN> restore [database|tablespace|datafile];
RMAN> restore [...] from tag 'xxx';
RMAN> restore [...] to '/somewhere/somefile';
RMAN> restore validate [database|tablespace|datafile]; ## prüft ob korrekte Backups vorhanden si

RMAN> recover database;
RMAN> recover [tablespace|datafile]; ## database muss open, tablespace oder datafile muss offlin
```

Weitere Recovery-Optionen Für Recovery ist nur beschränkter Platz für Archivelogs vorhanden:

```
RMAN> recover [...] delete archivelog maxsize 100M; ## braucht nur max. 100MB
```

Recovery an einen anderen Ort, weil Original-Location nicht erreichbar (z.B. weil neue Disk):

```
RMAN> set newname for datafile n to '/pfad/file.dbf';
RMAN> restore datafile n;
RMAN> switch datafile all; ## führt das SET NEWNAME im Katalog durch, Datenbank wird auf die neu
RMAN> recover datafile n;
```

```
RMAN> sql 'alter database datafile n online';
```

Unvollständiges Recovery Prinzip im rman: „UNTIL“-Klausel muss VOR dem RESTORE gesetzt werden!

```
RMAN> run { set until time 'time'; ## oder: set until change 'scn' | until sequence 'seq#' threa
            restore database;
            recover database;
            alter database open resetlogs; }
```

Alles Futsch? --> komplettes Recovery einer Datenbank

Alle zur Datenbank gehörenden Dateien sind verloren…
 Prinzip:

```
RMAN> recover spfile from autobackup; ## nomount Phase
RMAN> recover controlfile from autobackup; ## nomount Phase
RMAN> restore database; ## mount Phase
RMAN> recover database; ## mount Phase
RMAN> alter database open resetlogs; ## Datenbank sollte jetzt wieder laufen
```

im Detail:

```
RMAN> startup nomount force; ## startup mit default-Parametern. Geht nur im RMAN!
RMAN> restore spfile from autobackup; ## funktioniert nur, wenn an default-locations zu finden)
oder:
RMAN> set dbid=1234567890
RMAN> restore spfile from 'location_of_autobackups';

dann weiter:
RMAN> shutdown abort;
RMAN> startup nomount; ## Datenbank startet wieder mit eigenem spfile
RMAN> restore controlfile from autobackup;
RMAN> alter database mount;

ab hier sind viele LIST-Commands möglich:
RMAN> list incarnation;
RMAN> list backup of database; ## etc..

also einfach den Rest restoren und recovern:
RMAN> restore database;
RMAN> recover database; ## führt zu Fehler bei letzter Log-Seq#, da das online Redo-Log ja futsc
daher am einfachsten/besten:
SQL> recover database until cancel using backup controlfile; ## bis zum Schluss, falls zufällig
SQL> alter database open resetlogs;

Fertig! An dieser Stelle sollte dann natürlich, wegen der neuen Datenbank-Incarnation, nochmal e
RMAN> backup database;
```

Umgang mit RMAN-Scripts

```
RMAN> create [global] script name {backup database;};
RMAN> create [global] script name from file 'file mit rman-commands';

RMAN> print script name;
RMAN> print script name to file 'filename';
RMAN> list [global] script names;

RMAN> replace script name { new script commands; }
```

```
RMAN> run { execute script name; }
```

Report, List und Maintenance Commands

```
RMAN> report schema;
RMAN> report obsolete;
RMAN> report obsolete orphan; ## aus der vorletzten incarnation oder älter
RMAN> report need backup [redundancy nn | days nn];
RMAN> report unrecoverable [...];

RMAN> list backup of [database|tablespace|datafile];
RMAN> list backup of [archivelog|controlfile|spfile];
RMAN> list backup of [...] summary; ## zeigt übersichtliches Kurzformat
RMAN> list copy of [database|tablespace|datafile];
RMAN> list incarnation [of database];

RMAN> crosscheck [backup of database|archivelog]; ## setzt nicht vorhandene Backupsets|Archivelo
RMAN> delete [noprompt] expired backupset; ## löscht expired Eintragungen aus dem Katalog
RMAN> delete expired archivelog;
RMAN> delete [noprompt] obsolete;
RMAN> delete obsolete [redundancy n];
RMAN> delete obsolete [orphan];
RMAN> delete backupset nnn;
```

Weitere Tips und Tricks mit RMAN

Direkte SQL-Commands, die RMAN selbst 'kann':

```
RMAN> startup;
RMAN> shutdown [...];
RMAN> alter database open [resetlogs];
```

Alle SQL Commands, die keinen Output benötigen, können über die SQL-Klausel gestartet werden

```
RMAN> sql 'alter system switch logfile';
RMAN> sql 'alter system archive log all';
```

Wenn man NLS_DATE_FORMAT außerhalb von RMAN setzt, kann man innerhalb von RMAN bessere Anzeigen von Backup-Zeitpunkten bekommen (z.B. in LIST Kommandos):

```
$ export NLS_LANG=german_germany ## für RMAN in 10g nicht mehr nötig
$ export NLS_DATE_FORMAT=yyy-mm-dd:hh24:mi:ss
RMAN> list backup of database;
```

Wenn man mit Hausmitteln die CTL-Files ausgetauscht hat und RMAN nicht durchblickt, weil die restaurierte CTL-File aus einer alten Incarnation stammt:

```
RMAN> reset incarnation to 2; ## ohne catalog DB
RMAN> reset database to incarnation 2; ## mit catalog DB
```

RMAN Format Codes

Restrictions and Usage Notes Any name that is legal as a sequential filename on the platform is allowed, so long as each backup piece or copy has a unique name. If backing up to disk, then any legal disk filename is allowed, provided it is unique.

Keywords and Parameters

Syntax Element	Description
%a	Specifies the activation ID of the database
%c	Specifies the copy number of the backup piece within a set of duplexed backup pieces. If you did not duplex a backup, then this variable is 1 for backup sets and 0 for proxy copies. If one of these commands is enabled, then the variable shows the copy number. The maximum value for %c is 256.
%d	Specifies the name of the database
%D	Specifies the current day of the month from the Gregorian calender in format DD
%e	Specifies the archived log sequence number
%f	Specifies the absolute file number
%F	Combines the DBID, day, month, year, and sequence into a unique and repeatable generated name. This variable translates into **c-IIIIIIIIII-YYYYMMDD-QQ**, where **IIIIIIIIII** stands for the DBID. The DBID is printed in decimal so that it can be easily associated with the target database. **YYYYMMDD** is a time stamp in the Gregorian calendar of the day the backup is generated and **QQ** is the sequence in hexadecimal number that starts with 00 and has a maximum of FF (256)
%h	Specifies the archived redo log thread number
%I	Specifies the DBID
%M	Specifies the month in the Gregorian calendar in format MM
%N	Specifies the tablespace name
%n	Specifies the name of the database, padded on the right with x characters to a total length of eight characters. For example, if the prod1 is the database name, then the padded name is prod1xxx.
%p	Specifies the piece number within the backup set. This value starts at 1 for each backup set and is incremented by 1 as each backup piece is created. If you specify PROXY, then the %p variable must be included in the FORMAT string either explicitly or implicitly within %U
%s	Specifies the backup set number. This number is a counter in the control file that is incremented for each backup set. The counter value starts at 1 and is unique for the lifetime of the control file. If you restore a backup control file, then duplicate values can result. Also, CREATE CONTROLFILE initializes the counter back to 1
%t	Specifies the backup set time stamp, which is a 4-byte value derived as the number of seconds elapsed since a fixed reference time. The combination of %s and %t can be used to form a unique name for the backup set

Syntax Element	Description
%T	Specifies the year, month and day in the Gregorian calendar in this format: <tt>YYYYMMDD</tt>
%u	Specifies an 8-character name constituted by compressed representations of the backup set or image copy number and the time the backup set or image copy war created
%U	Specifies a system-generated unique filename (default). The meaning of %U is different vor image copies and backup pieces. For a backup piece %U specifies a conveniant shorthand for %u_%p_%c that guarantees uniqueness in generated backup filenames. If you do not specify a format when making a backup, then RMAN uses %U by default. For an image copy of a datafile %U means **data-D-%d_id-%I_TS-%N_FNO-%f_%u**. For an image copy of an archived redo log %U means **arch-D_%d-id-%I_S-%e_T-%h_A-%a_%u**. For an image copy of a control file %U means **cf-D_%d-id-%I_%u**
%Y	Specifies the year in the format **YYYY**
% %	Specifies the '%' character. For example, %%Y translates to the string %Y

Vollständiges Recovery

Ach du Scheiße! Alles in oradata (ctl-, db-, redo-files) ist weg! Aber zum Glück haben wir ja ein Backup:

```
RMAN> connect target /
RMAN> connect catalog rman/rman@repo
RMAN> startup nomount [force]
RMAN> restore controlfile from autobackup;
RMAN> alter database mount;
RMAN> restore database;
RMAN> recover database;
```

Der letzte Befehl bricht mit Fehler ab. Die Archivlogs sind zwar vorhanden und wurden bis zum letzten logswitch wiederhergestellt. Die online Redo-Logs sind leider futsch, d.h. wir müssen die Datenbank mit 'resetlogs' öffnen:

```
RMAN> alter database open resetlogs;
```