**RMAN (Recovery Manager) Commands in Oracle**

**rman commands**  
  
rman  
[ TARGET [=] ['] [userid][/[password]][@net\_service\_name] [']  
| {CATALOG [=] ['] [userid][/[password]][@net\_service\_name] [']  
| LOG [=] [']filename['] [APPEND]  
...  
]...  
  
$ rman  
$ rman NOCATALOG  
$ rman TARGET SYS/pwd@target  
$ rman TARGET SYS/pwd@target NOCATALOG  
$ rman TARGET SYS/pwd@target LOG $ORACLE\_HOME/dbs/my\_log.log APPEND  
$ rman CATALOG rman/pwd@catdb  
$ rman TARGET=SYS/pwd@target CATALOG=rman/pwd@cat  
$ rman TARGET / CATALOG rman/rman@cat  
$ rman TARGET / SCRIPT dwh LOG /tmp/dwh.log  
$ rman PIPE newpipe TARGET / TIMEOUT 90  
$ rman @/my\_dir/my\_commands.txt  
$ rman @backup\_ts\_generic.rman "/tmp" USERS  
$ rman CMDFILE=backup\_ts\_users.rman  
$ rman TARGET / @backup\_db.rman  
$ rman TARGET / CATALOG rman/pwd@cat CMDFILE cmdfile.rcv LOG outfile.txt  
$ rman TARGET / CATALOG rman/pwd@cat DEBUG TRACE trace.log  
$ rman TARGET SYS/pwd@prod CATALOG rman/rman@rcat @'/oracle/dbs/whole.rcv'  
$ rman TARGET user/pwd CMDFILE=takefulldb.cmd @@takefulldb.cmd  
$ rman CHECKSYNTAX @'/tmp/backup\_db.cmd'  
$ rman MSGNO  
$ rman | tee rman.log  
$ rman help=yes

@ (at sign) Run a command file.  
@@(double at sign) Run a command file in the same directory as another command file that is currently running. The @@ command differs from the @ command only when run from within a command file.  
RMAN> @backup\_db.rman  
RMAN> @/my\_dir/my\_command\_file.txt  
RMAN> @/tmp/bkup\_db.rman whole\_db  
RMAN> @backup\_ts\_generic.rman "/tmp" $1  
RMAN> RUN {@backup\_db.rman}

**CONNECT command**  
Establish a connection between [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) and a target, auxiliary, or recovery catalog database.  
RMAN> CONNECT TARGET;  
RMAN> CONNECT TARGET /  
RMAN> CONNECT TARGET sys@tgt;  
RMAN> CONNECT TARGET sys/pwd@tgt;  
RMAN> CONNECT CATALOG rman@catdb;  
RMAN> CONNECT CATALOG rman/pwd@catdb;  
RMAN> CONNECT AUXILIARY /  
RMAN> CONNECT AUXILIARY rman@auxdb;  
RMAN> CONNECT AUXILIARY rman/pwd@auxdb;  
  


**CREATE CATALOG command**  
Create Oracle schema for the recovery catalog.  
RMAN> CREATE CATALOG;  
RMAN> CREATE CATALOG TABLESPACE rmants;  
RMAN> CREATE VIRTUAL CATALOG; -- [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
SQL> EXEC rman.DBMS\_RCVCAT.CREATE\_VIRTUAL\_CATALOG; -- [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)

RMAN> SQL "EXEC catown.DBMS\_RCVCAT.CREATE\_VIRTUAL\_CATALOG"; -- [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)

**DROP CATALOG command**  
Remove Oracle schema from the recovery catalog.  
RMAN> DROP CATALOG;

**RESYNC CATALOG command**  
Perform a full resynchronization, which creates a snapshot control file and then copies any new or changed information from that snapshot control file to the recovery catalog.  
RMAN> RESYNC CATALOG;  
RMAN> RESYNC CATALOG FROM DB\_UNIQUE\_NAME prod\_db;  
RMAN> RESYNC CATALOG FROM DB\_UNIQUE\_NAME ALL;

**UPGRADE CATALOG command**  
Upgrade the recovery catalog schema from an older version to the version required by the RMAN executable.  
RMAN> UPGRADE CATALOG;

**IMPORT CATALOG command**  
Import the metadata from one recovery catalog into another recovery catalog.  
RMAN> IMPORT CATALOG cat@srcdb;  
RMAN> IMPORT CATALOG rcat@inst DBID=2871507123;  
RMAN> IMPORT CATALOG cat@srcdb DBID=1844750987, 61738563;  
RMAN> IMPORT CATALOG cat@srcdb DB\_NAME=prod2;  
RMAN> IMPORT CATALOG cat@srcdb DB\_NAME=prod3, prod4;  
RMAN> IMPORT CATALOG rman/rman@catdb1 DB\_NAME=prod1 NO UNREGISTER;  
RMAN> IMPORT CATALOG rman/oracle@catdb1 NO UNREGISTER;

**REGISTER command**  
Register the target database in the recovery catalog.  
RMAN> REGISTER DATABASE;  
RMAN> REGISTER CATALOG;  
RMAN> REGISTER CATALOG TABLESPACE tbs-name;

**UNREGISTER command**  
Unregister a Oracle database from the recovery catalog.  
RMAN> UNREGISTER DATABASE;  
RMAN> UNREGISTER DATABASE NOPROMPT;  
RMAN> UNREGISTER DATABASE prod1;  
RMAN> UNREGISTER DATABASE prod2 NOPROMPT;  
RMAN> UNREGISTER DB\_UNIQUE\_NAME prod2;  
RMAN> UNREGISTER DB\_UNIQUE\_NAME prod1 NOPROMPT;  
RMAN> UNREGISTER DB\_UNIQUE\_NAME prod2 INCLUDING BACKUPS;  
RMAN> UNREGISTER DB\_UNIQUE\_NAME prod3 INCLUDING BACKUPS NOPROMPT;

**GRANT command**  
Grant privileges to a recovery catalog user.  
RMAN> GRANT CATALOG FOR DATABASE prod1 TO vpc1; -- [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
RMAN> GRANT REGISTER DATABASE TO bckop2;  
RMAN> GRANT RECOVERY\_CATALOG\_OWNER TO rmanop1, rmanop3;

**REVOKE command**  
Revoke privileges from a recovery catalog user.  
RMAN> REVOKE CATALOG FOR DATABASE prod1 FROM vpc1; -- [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
RMAN> REVOKE REGISTER DATABASE FROM bckop2;  
RMAN> REVOKE RECOVERY\_CATALOG\_OWNER FROM bckop;

**RESET DATABASE command**  
Inform [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) that the SQL statement ALTER DATABASE OPEN RESETLOGS has been executed and that a new incarnation of the target database has been created, or reset the target database to a prior incarnation.  
RMAN> RESET DATABASE TO INCARNATION 3;

**STARTUP command**  
Startup the target database. This command is equivalent to the [SQL\*Plus STARTUP](http://satya-dba.blogspot.com/2009/01/startupshutdown-options.html) command.  
RMAN> STARTUP;  
RMAN> STARTUP PFILE=’/u01/app/oracle/admin/pfile/initsid.ora’  
RMAN> STARTUP NOMOUNT;  
RMAN> STARTUP MOUNT;  
RMAN> STARTUP FORCE;  
RMAN> STARTUP FORCE DBA;  
RMAN> STARTUP FORCE DBA PFILE=c:\Oracle\Admin\pfile\init.ora;  
RMAN> STARTUP FORCE NOMOUNT;  
RMAN> STARTUP FORCE MOUNT DBA PFILE=/tmp/inittrgt.ora;

RMAN> STARTUP AUXILIARY nomount;

**SHUTDOWN command**  
Shutdown the target database. This command is equivalent to the [SQL\*Plus SHUTDOWN](http://satya-dba.blogspot.com/2009/01/startupshutdown-options.html) command.  
RMAN> SHUTDOWN;  
RMAN> SHUTDOWN NORMAL;  
RMAN> SHUTDOWN TRANSACTIONAL;  
RMAN> SHUTDOWN IMMEDIATE;  
RMAN> SHUTDOWN ABORT;

**ALTER DATABASE command**  
Mount or open a database.  
RMAN> ALTER DATABASE MOUNT;  
RMAN> ALTER DATABASE OPEN;  
RMAN> ALTER DATABASE OPEN RESETLOGS;

**SHOW command**  
Display the current CONFIGURE settings.  
  
SHOW  
{ RETENTION POLICY  
| BACKUP OPTIMIZATION  
| [DEFAULT] DEVICE TYPE  
| CONTROLFILE AUTOBACKUP [FORMAT]  
| [AUXILIARY] CHANNEL [FOR DEVICE TYPE deviceSpecifier]  
| MAXSETSIZE  
| DATAFILE BACKUP COPIES  
| ARCHIVELOG [BACKUP COPIES|DELETION POLICY]  
| AUXNAME  
| EXCLUDE  
| ENCRYPTION {ALGORITHM | FOR [DATABASE|TABLESPACE]}  
| COMPRESSION ALGORITHM  
| SNAPSHOT CONTROLFILE NAME  
| DB\_UNIQUE\_NAME  
| ALL  
} FOR [DB\_UNIQUE\_NAME [‘db\_unique\_name’|ALL]];  
  
  
RMAN> SHOW ALL;  
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default  
CONFIGURE BACKUP OPTIMIZATION OFF; # default  
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default  
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default  
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default  
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default  
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default  
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DISK TO 1; # default  
CONFIGURE MAXSETSIZE TO UNLIMITED; # default  
CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default  
CONFIGURE DATAFILE BACKUP COPIES FOR SBT TO 1; # default  
CONFIGURE ARCHIVELOG BACKUP COPIES FOR SBT TO 1; # default  
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default  
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default  
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE; # default -- [Oracle 11g R2](http://satya-dba.blogspot.com/2009/09/whats-new-in-11g-release-2.html)  
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default  
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '.../dbs/snapcf\_sid.f'; # default  
  
%F = dbid, day, month, year and sequence  
%U = %u\_%p\_%c  
%u = eight characters of the backup set and time ...  
%p = piece number within the backupset  
%c = copy number of the backup piece ...  
  
RMAN> SHOW RETENTION POLICY;  
RMAN> SHOW RETENTION POLICY FOR DB\_UNIQUE\_NAME ALL;  
RMAN> SHOW DEVICE TYPE;  
RMAN> SHOW DEVICE TYPE FOR DB\_UNIQUE\_NAME prod3;  
RMAN> SHOW DEFAULT DEVICE TYPE;  
RMAN> SHOW CHANNEL;  
RMAN> SHOW MAXSETSIZE;  
RMAN> SHOW BACKUP OPTIMIZATION;  
RMAN> SHOW SNAPSHOT CONTROLFILE NAME;  
RMAN> SHOW CONTROLFILE AUTOBACKUP;  
RMAN> SHOW COMPRESSION ALGORITHM;  
RMAN> SHOW ENCRYPTION ALGORITHM;  
RMAN> SHOW ALL FOR DB\_UNIQUE\_NAME ALL;  
RMAN> SHOW ALL FOR DB\_UNIQUE\_NAME 'STANDBY';

**CONFIGURE command**  
To configure persistent RMAN settings. These settings apply to all RMAN sessions until explicitly changed or disabled.  
  
CONFIGURE deviceConf;  
CONFIGURE backupConf;  
CONFIGURE AUXNAME FOR DATAFILE datafileSpec {TO 'filename' | CLEAR};  
CONFIGURE SNAPSHOT CONTROLFILE NAME {TO 'filename' | CLEAR};  
CONFIGURE cfauConf;

CONFIGURE ARCHIVELOG DELETION POLICY

{CLEAR | TO {APPLIED ON [ALL] STANDBY | BACKED UP integer TIMES TO DEVICE TYPE deviceSpecifier | NONE | SHIPPED TO [ALL] STANDBY}

[{APPLIED ON [ALL] STANDBY | BACKED UP integer TIMES TO DEVICE TYPE deviceSpecifier | NONE | SHIPPED TO [ALL] STANDBY}] …

}

deviceConf::=  
{ DEFAULT DEVICE TYPE { TO deviceSpec | CLEAR }  
| DEVICE TYPE deviceSpec { PARALLELISM integer | CLEAR }  
| [AUXILIARY] CHANNEL [integer] DEVICE TYPE deviceSpec {allocOperandList|CLEAR}  
}  
  
allocOperandList::=  
{ PARMS [=] 'channel\_parms'  
| FORMAT [=] 'format\_string' [, 'format\_string']...  
| { MAXPIECESIZE [=] integer | RATE [=] integer } [K | M | G]  
...  
}...  
  
connectStringSpec::=  
['] [userid] [/[password]] [@net\_service\_name] [']  
  
backupConf::=  
{RETENTION POLICY {TO {RECOVERY WINDOW OF integer DAYS  
| REDUNDANCY [=] integer | NONE  
}  
| CLEAR  
}  
| MAXSETSIZE {TO {integer [K | M | G]| UNLIMITED}  
| CLEAR  
}  
| {ARCHIVELOG | DATAFILE}  
BACKUP COPIES FOR DEVICE TYPE deviceSpec {TO integer | CLEAR}  
| BACKUP OPTIMIZATION {ON | OFF | CLEAR}  
| EXCLUDE FOR TABLESPACE tablespace\_name [CLEAR]  
}  
  
cfauConf::==  
CONTROLFILE AUTOBACKUP {ON | OFF | CLEAR | FORMAT FOR DEVICE TYPE deviceSpec {TO 'format string'|CLEAR}}  
  
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;  
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP OFF;  
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'cf%F';  
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '+BACKUP';  
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK CLEAR;  
RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY 3;  
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;  
RMAN> CONFIGURE RETENTION POLICY CLEAR;  
RMAN> CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 2;  
  
RMAN> CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 2;  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY CLEAR; --11g  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO SHIPPED TO STANDBY;  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO SHIPPED TO ALL STANDBY;  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY;  
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;

RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 2 TIMES TO sbt;

RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 3 TIMES TO disk;

RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;  
RMAN> CONFIGURE DEVICE TYPE sbt PARALLELISM 3;  
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 4;

RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 3 BACKUP TYPE TO BACKUPSET;

RMAN> CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET;  
  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt;  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=mml\_env\_settings';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'ENV=(NSR\_SERVER=bksrv1)';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMS 'BLKSIZE=1048576';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt FORMAT 'bkup\_%U';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt CLEAR;  
RMAN> CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'SYS/pwd@node2' PARMS 'ENV=(NSR\_SERVER=bksrv2)';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/tmp/%U';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT 'C:\backup\df%t\_s%s\_s%p';  
RMAN> CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/backup/db\_%s%d\_%p';  
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT CLEAR;  
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK DEBUG 5;  
  
RMAN> CONFIGURE BACKUP OPTIMIZATION ON;  
RMAN> CONFIGURE BACKUP OPTIMIZATION OFF;  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO ‘/backup/snapcf\_%d.f‘;  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO ‘+[FRA](http://satya-dba.blogspot.com/2009/02/flash-recovery-area.html)/snap/snapcf\_%d.f‘;  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO ‘/ocfs/oradata/snapcf‘;  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO ‘/dev/sda‘;  
RMAN> CONFIGURE MAXSETSIZE TO 100M;  
RMAN> CONFIGURE MAXSETSIZE TO UNLIMITED;  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt MAXPIECESIZE 1G;  
RMAN> CONFIGURE EXCLUDE FOR TABLESPACE example;  
RMAN> CONFIGURE EXCLUDE CLEAR;  
RMAN> CONFIGURE AUXNAME FOR DATAFILE 4 TO '/oracle/auxfiles/aux\_4.f';  
RMAN> CONFIGURE AUXNAME FOR DATAFILE 2 CLEAR;  
  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'BZIP2';  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'ZLIB'; --[Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'LOW'; --[11g R2](http://satya-dba.blogspot.com/2009/09/whats-new-in-11g-release-2.html),corresponds to LZO  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'MEDIUM'; --[11g R2](http://satya-dba.blogspot.com/2009/09/whats-new-in-11g-release-2.html),corresponds to ZLIB  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'HIGH'; --[11g R2](http://satya-dba.blogspot.com/2009/09/whats-new-in-11g-release-2.html),corresponds to unmodified BZIP2  
RMAN> CONFIGURE COMPRESSION ALGORITHM 'BASIC'; --[Oracle 11g R2](http://satya-dba.blogspot.com/2009/09/whats-new-in-11g-release-2.html),corresponds to BZIP2  
  
RMAN> CONFIGURE DB\_UNIQUE\_NAME 'standby' CONNECT IDENTIFIER 'standby\_cs';  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK FOR DB\_UNIQUE\_NAME 'standby';  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK FOR DB\_UNIQUE\_NAME ALL;  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO SBT FOR DB\_UNIQUE\_NAME po;

**SET command**  
Set the value of various attributes that affect RMAN behaviour for the duration of a RUN block or a session.  
  
SET {set\_rman\_option [;] | set\_run\_option;}  
  
set\_rman\_option::=  
{ECHO {ON|OFF} | DBID [=] integer  
| CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpec TO 'frmt\_string'  
  
set\_run\_option::=  
{ NEWNAME FOR DATAFILE datafileSpec TO {'filename' | NEW}  
| ARCHIVELOG DESTINATION TO 'log\_archive\_dest'  
| untilClause  
| COMMAND ID TO 'string'  
| CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE deviceSpec TO 'frmt\_string'  
...  
}  
  
ECHO - Controls whether RMAN commands are displayed in the message log.  
  
DBID - A unique 32-bit identification number computed when the database is created. RMAN displays the DBID upon connection to the target database. We can obtain the DBID by querying V$DATABASE or RC\_DATABASE.  
  
NEWNAME FOR DATAFILE - The default name for all subsequent RESTORE or SWITCH commands that affect the specified datafile.  
  
MAXCORRUPT FOR DATAFILE - A limit on the number of previously undetected physical block corruptions that Oracle will allow in the datafile(s).  
  
AUTOLOCATE - Force RMAN to automatically discover which nodes of an [Oracle Real Application Clusters](http://satya-racdba.blogspot.com/) configuration contain the backups that you want to restore.  
  
RMAN> SET ECHO ON;  
RMAN> SET ECHO OFF;  
RMAN> SET DATABASE prod;  
RMAN> SET DBID=4240978820;  
RMAN> SET DBID 591329635;  
RMAN> SET COMMAND ID TO 'rman';  
RMAN> SET MAXCORRUPT FOR DATABASE TO 2;  
RMAN> SET MAXCORRUPT FOR DATAFILE 13 TO 200;  
RMAN> SET BACKUP COPIES = 2;  
  
RMAN> SET NEWNAME FOR DATABASE TO '/oradata1/%b';  
RMAN> SET NEWNAME FOR TABLESPACE users TO '/oradata2/%U';  
RMAN> SET NEWNAME FOR DATAFILE 1 to ‘/oradata/system01.dbf’;  
RMAN> SET NEWNAME FOR DATAFILE '/disk7/tbs11.f' TO '/disk9/tbs11.f';  
RMAN> SET NEWNAME FOR TEMPFILE 1 TO '/newdisk/dbs/temp1.f';  
  
RMAN> SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE sbt TO 'cf\_%F';  
RMAN> SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'cf\_%F.bak';  
RMAN> SET UNTIL TIME ’04-23-2010:23:50:04’;  
RMAN> SET ARCHIVELOG DESTINATION TO '/oracle/temp\_restore';  
RMAN> SET COMPRESSION ALGORITHM 'LOW';  
RMAN> SET COMPRESSION ALGORITHM 'LOW' OPTIMIZE FOR LOAD FALSE;  
RMAN> SET COMPRESSION ALGORITHM 'MEDIUM';  
RMAN> SET COMPRESSION ALGORITHM 'HIGH';

  
**BACKUP command**  
Backs up Oracle database files, copies of database files, archived logs, or backup sets.  
  
BACKUP FULL Options  
BACKUP FULL AS (COPY | BACKUPSET) Options  
BACKUP INCREMENTAL LEVEL [=] integer Options  
BACKUP INCREMENTAL LEVEL [=] integer AS (COPY | BACKUPSET) Options  
BACKUP AS (COPY | BACKUPSET) Options  
BACKUP AS (COPY | BACKUPSET) (FULL | INCREMENTAL LEVEL [=] integer) Options  
  
Options::=  
[backupOperand [backupOperand]...] backupSpec [backupSpec]...  
[PLUS ARCHIVELOG [backupSpecOperand [backupSpecOperand]...]];  
  
backupOperand::=  
{ FORMAT [=] 'format\_string' [, 'format\_string']...  
| CHANNEL ['] channel\_id [']  
| CUMULATIVE  
| MAXSETSIZE [=] integer [K | M | G]  
| TAG [=] ['] tag\_name [']  
| keepOption  
| SKIP {OFFLINE | READONLY | INACCESSIBLE}  
| VALIDATE  
| NOT BACKED UP [SINCE TIME [=] 'date\_string']  
| COPIES [=] integer  
| DEVICE TYPE deviceSpecifier  
...  
}  
  
backupSpec::=  
[(]  
{ BACKUPSET  
{ {ALL | completedTimeSpec } | primary\_key) [, primary\_key]... }  
| COPY OF { DATABASE  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| DATAFILE datafileSpec [, datafileSpec]...  
}  
| DATAFILE datafileSpec [, datafileSpec]...  
| DATAFILECOPY 'filename' [, 'filename']...  
| DATAFILECOPY FROM TAG [=] ['] tag\_name ['] [, ['] tag\_name [']]...  
| DATAFILECOPY { ALL | LIKE 'string\_pattern' }  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| DATABASE  
| archivelogRecordSpecifier  
| CURRENT CONTROLFILE [FOR STANDBY]  
| CONTROLFILECOPY 'filename'  
| SPFILE  
}  
[backupSpecOperand [backupSpecOperand]...]  
  
backupSpecOperand::=  
{ FORMAT [=] 'format\_string' [, 'format\_string']...  
| CHANNEL ['] channel\_id [']  
| CUMULATIVE  
| MAXSETSIZE [=] integer [K | M | G]  
| TAG [=] ['] tag\_name [']  
| keepOption  
| SKIP {OFFLINE | READONLY | INACCESSIBLE}  
| NOT BACKED UP [SINCE TIME [=] 'date\_string' | integer TIMES]  
| DELETE [ALL] INPUT  
...  
}  
  
RMAN> BACKUP DATABASE;  
RMAN> BACKUP DATABASE TAG=’test backup’;  
RMAN> BACKUP DATABASE COMMENT=’full backup’;  
RMAN> BACKUP TAG 'weekly\_full\_db\_bkup' DATABASE MAXSETSIZE 10M;  
RMAN> BACKUP MAXSETSIZE 500M DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP DURATION 00:60 DATABASE;  
RMAN> BACKUP DURATION 00:30 MINIMIZE TIME DATABASE;  
RMAN> BACKUP DURATION 00:45 MINIMIZE LOAD DATABASE;  
  
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP DATABASE KEEP FOREVER;  
RMAN> BACKUP DATABASE KEEP UNTIL TIME=’SYSDATE+30’;  
RMAN> BACKUP DATABASE UNTIL 'SYSDATE+365' NOLOGS;  
RMAN> BACKUP DATABASE NOEXCLUDE;  
RMAN> BACKUP DATABASE NOEXCLUDE KEEP FOREVER TAG=’abc’;  
RMAN> BACKUP DATABASE SKIP READONLY;  
RMAN> BACKUP DATABASE SKIP OFFLINE;  
RMAN> BACKUP DATABASE SKIP INACCESSIBLE;  
RMAN> BACKUP DATABASE SKIP READONLY SKIP OFFLINE SKIP INACCESSIBLE;  
RMAN> BACKUP DATABASE FORCE; -- backup read only database also  
RMAN> BACKUP DATABASE NOT BACKED UP;  
RMAN> BACKUP DATABASE NOT BACKED UP SINCE TIME=’SYSDATE–3’;  
RMAN> BACKUP NOT BACKED UP SINCE TIME 'SYSDATE-10' MAXSETSIZE 500M DATABASE PLUS ARCHIVELOG;  
  
RMAN> BACKUP DATABASE COPIES=2;  
RMAN> BACKUP DATABASE FORMAT '/disk1/backups/db\_%U.bck'  
TAG quarterly KEEP UNTIL TIME 'SYSDATE+365' RESTORE POINT Q1FY12;  
RMAN> BACKUP DEVICE TYPE DISK DATABASE;  
RMAN> BACKUP DEVICE TYPE sbt DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP DEVICE TYPE sbt DATAFILECOPY FROM TAG 'latest' FORMAT 'df%f\_%d';  
RMAN> BACKUP DEVICE TYPE sbt ARCHIVELOG LIKE '/disk%arc%' DELETE ALL INPUT;  
RMAN> BACKUP DEVICE TYPE sbt BACKUPSET COMPLETED BEFORE 'SYSDATE-14'DELETE INPUT;  
RMAN> BACKUP CHECK LOGICAL DATABASE;  
RMAN> BACKUP VALIDATE CHECK LOGICAL DATABASE;  
RMAN> BACKUP VALIDATE DATABASE;  
RMAN> BACKUP VALIDATE DATABASE ARCHIVELOG ALL;  
  
RMAN> BACKUP TABLESPACE test;  
RMAN> BACKUP TABLESPACE system, users, tools;  
RMAN> BACKUP TABLESPACE 4;  
RMAN> BACKUP TABLESPACE gld PLUS ARCHIVELOG;  
RMAN> BACKUP TABLESPACE invd INCLUDE CURRENT CONTROLFILE;  
RMAN> BACKUP TABLESPACE appsd INCLUDE CURRENT CONTROLFILE PLUS ARCHIVELOG;  
RMAN> BACKUP TABLESPACE dwh SECTION SIZE 100M;  
RMAN> BACKUP SECTION SIZE 250M TABLESPACE datamart;  
  
RMAN> BACKUP DATAFILE 1;  
RMAN> BACKUP DATAFILE 3, 2, 14;  
RMAN> BACKUP DATAFILE ‘/u01/data/...’;  
RMAN> BACKUP DATAFILE 1 PLUS ARCHIVELOG;  
RMAN> BACKUP KEEP FOREVER FORMAT '?/dbs/%U\_longterm.cpy' TAG longterm\_bck DATAFILE 1 DATAFILE 2;  
RMAN> BACKUP SECTION SIZE 500M DATAFILE 6;  
  
RMAN> BACKUP ARCHIVELOG ALL;  
RMAN> BACKUP ARCHIVELOG ALL DELETE INPUT;  
RMAN> BACKUP ARCHIVELOG LIKE '/arch%' DELETE ALL INPUT;  
RMAN> BACKUP ARCHIVELOG FROM TIME ‘SYSDATE–3’;  
RMAN> BACKUP ARCHIVELOG FROM SEQUENCE 100;  
RMAN> BACKUP ARCHIVELOG FROM SEQUENCE 999 DELETE INPUT;

RMAN> BACKUP ARCHIVELOG FROM SEQUENCE 123 DELETE ALL INPUT;

RMAN> BACKUP ARCHIVELOG FROM SEQUENCE 21531 UNTIL SEQUENCE 21590 FORMAT '/tmp/archive\_backup.bkp';

RMAN> BACKUP ARCHIVELOG ALL FROM SEQUENCE 1200 DELETE ALL INPUT;

RMAN> BACKUP ARCHIVELOG NOT BACKED UP 2 TIMES;  
RMAN> BACKUP ARCHIVELOG COMPLETION TIME BETWEEN 'SYSDATE-28' AND 'SYSDATE-7';  
RMAN> BACKUP FORMAT='AL\_%d/%t/%s/%p' ARCHIVELOG LIKE '%arc\_dest%';  
  
RMAN> BACKUP CURRENT CONTROLFILE;  
OR  
RMAN> SQL “ALTER DATABASE BACKUP CONTROLFILE TO ’’/u01/ .../bkctl.ctl’’ “;  
  
RMAN> BACKUP CURRENT CONTROLFILE TO '/backup/cntrlfile.copy';  
RMAN> BACKUP CONTROLFILE COPY ‘/u10/backup/control.bkp’;  
RMAN> BACKUP SPFILE;  
RMAN> BACKUP DEVICE TYPE sbt SPFILE ARCHIVELOG ALL;  
RMAN> BACKUP DEVICE TYPE sbt DATAFILECOPY ALL NODUPLICATES;  
  
RMAN> BACKUP RECOVERY FILES;  
  
**BACKUP set**  
RMAN> BACKUP BACKUPSET ALL;  
RMAN> BACKUP BACKUPSET ALL FORMAT = ‘/u01/.../backup\_%u.bak’;  
RMAN> BACKUP BACKUPSET COMPLETED BEFORE ‘SYSDATE-3’ DELETE INPUT;  
RMAN> BACKUP DEVICE TYPE sbt BACKUPSET COMPLETED BEFORE 'SYSDATE-14' DELETE INPUT;  
RMAN> BACKUP COPIES 2 DEVICE TYPE sbt BACKUPSET ALL;  
RMAN> BACKUP AS COMPRESSED BACKUPSET;  
RMAN> BACKUP AS COMPRESSED BACKUPSET DEVICE TYPE DISK COPIES 2 DATABASE FORMAT '/disk1/db\_%U', '/disk2/db\_%U';

RMAN> BACKUP AS COMPRESSED BACKUPSET INCREMENTAL FROM SCN 4111140000000 DATABASE TAG 'RMAN\_RECOVERY';

RMAN> BACKUP AS BACKUPSET DATAFILE '$ORACLE\_HOME/oradata/users01.dbf','$ORACLE\_HOME/oradata/tools01.dbf';  
RMAN> BACKUP AS BACKUPSET DATAFILECOPY ALL;  
RMAN> BACKUP AS BACKUPSET DATAFILECOPY ALL NODUPLICATES;  
  
**IMAGE copy**  
RMAN> BACKUP AS COPY DATABASE;  
RMAN> BACKUP AS COPY COPY OF DATABASE FROM TAG 'test' CHECK LOGICAL TAG 'duptest';  
RMAN> BACKUP AS COPY TABLESPACE 8;  
RMAN> BACKUP AS COPY TABLESPACE test;  
RMAN> BACKUP AS COPY TABLESPACE system, tools, users, undotbs;  
RMAN> BACKUP AS COPY DATAFILE 1;  
RMAN> BACKUP AS COPY DATAFILE 2 FORMAT '/disk2/df2.cpy' TAG my\_tag;  
RMAN> BACKUP AS COPY CURRENT CONTROLFILE;  
RMAN> BACKUP AS COPY CURRENT CONTROLFILE FORMAT ‘/....’;  
RMAN> BACKUP AS COPY ARCHIVELOG ALL;  
RMAN> BACKUP AS COPY KEEP FOREVER NOLOGS CURRENT CONTROLFILE FORMAT '?/oradata/cf\_longterm.cpy', DATAFILE 1 FORMAT '?/oradata/df1\_longterm.cpy', DATAFILE 2 FORMAT '?/oradata/df2\_longterm.cpy';  
RMAN> BACKUP AS COPY DATAFILECOPY 'bar' FORMAT 'foobar';  
RMAN> BACKUP AS COPY DATAFILECOPY '/disk2/df2.cpy' FORMAT '/disk1/df2.cpy';

RMAN> BACKUP AS COPY REUSE TARGETFILE '/u01/oracle/11.2.0.2/dbs/orapwcrd' AUXILIARY FORMAT '/u01/oracle/11.2.0.2/dbs/orapwcrd';

RMAN> BACKUP AS COPY CURRENT CONTROLFILE FOR STANDBY AUXILIARY format '+DATA/crd/data1/control01.ctl';

**Incremental backups**  
RMAN> BACKUP INCREMENTAL LEVEL=0 DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL=1 DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL=2 DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL 2 CUMULATIVE DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL 2 DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL=0 DATABASE PLUS ARCHIVELOG;  
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE SKIP INACCESSIBLE DATABASE;  
RMAN> BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr' DATABASE;  
RMAN> BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'oltp' DATABASE;  
RMAN> BACKUP DEVICE TYPE DISK INCREMENTAL FROM SCN 351986 DATABASE FORMAT '/tmp/incr\_standby\_%U';  
RMAN> BACKUP INCREMENTAL FROM SCN 629184 DATAFILE 5 FORMAT '/tmp/ForStandby\_%U' TAG 'FORSTANDBY';  
  
RMAN> BACKUP INCREMENTAL LEVEL = --- tablespace/datafile

RMAN> BACKUP BLOCKS ALL CHECK LOGICAL VALIDATE DATAFILE 1398;



**LIST command**  
Produce a detailed listing of backup sets or copies.  
  
LIST  
{ INCARNATION [OF DATABASE [[']database\_name[']]]  
| [EXPIRED] {listObjectSpec  
[ maintQualifier | RECOVERABLE [untilClause] ]... | recordSpec}  
};  
  
listObjectSpec::=  
{BACKUP [OF listObjectList] [listBackupOption] | COPY [OF listObjectList] | archivelogRecordSpecifier}  
  
listObjectList::=  
[ DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']]...  
| archivelogRecordSpecifier  
| DATABASE [SKIP TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| CONTROLFILE  
| SPFILE  
]...  
  
listBackupOption::=  
[[BY BACKUP] [VERBOSE] | SUMMARY | BY {BACKUP SUMMARY|FILE}]  
  
RMAN> LIST INCARNATION;  
RMAN> LIST INCARNATION OF DATABASE;  
RMAN> LIST INCARNATION OF DATABASE vis;  
RMAN> LIST DB\_UNIQUE\_NAME ALL;  
RMAN> LIST DB\_UNIQUE\_NAME OF DATABASE;  
  
RMAN> LIST BACKUP;  
RMAN> LIST BACKUP SUMMARY;  
RMAN> LIST BACKUP BY FILE;  
RMAN> LIST BACKUP OF DATABASE;  
RMAN> LIST BACKUP OF DATABASE BY BACKUP;  
RMAN> LIST BACKUP OF TABLESPACE test SUMMARY;  
RMAN> LIST BACKUP OF DATAFILE 65;  
RMAN> LIST BACKUP OF DATAFILE 11 SUMMARY;  
RMAN> LIST BACKUP OF CONTROLFILE;  
RMAN> LIST BACKUP OF ARCHIVELOG FROM SEQUENCE 2222;

RMAN> LIST BACKUP OF ARCHIVELOG FROM TIME 'sysdate-1';

RMAN> LIST BACKUP OF ARCHIVELOG ALL COMPLETED BEFORE 'sysdate-2';

RMAN> LIST BACKUP RECOVERABLE;  
RMAN> LIST EXPIRED BACKUP;  
RMAN> LIST EXPIRED BACKUP OF ARCHIVELOG ALL SUMMARY;  
  
RMAN> LIST COPY;  
RMAN> LIST COPY OF DATABASE ARCHIVELOG ALL;  
RMAN> LIST COPY OF TABLESPACE appl\_idx;  
RMAN> LIST COPY OF DATAFILE 11, 60, 98;  
RMAN> LIST COPY OF CONTROLFILE;  
RMAN> LIST EXPIRED COPY;  
  
RMAN> LIST BACKUPSET SUMMARY;  
RMAN> LIST BACKUPSET 109;  
RMAN> LIST BACKUPSET OF DATAFILE 1;  
RMAN> LIST ARCHIVELOG;

RMAN> LIST ARCHIVELOG ALL LIKE '%5515%';

RMAN> LIST CONTROLFILECOPY "/tmp/cntrlfile.copy";  
  
RMAN> LIST SCRIPT NAMES;  
RMAN> LIST ALL SCRIPT NAMES;  
RMAN> LIST GLOBAL SCRIPT NAMES;  
  
RMAN> LIST FAILURE; -- [11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
RMAN> LIST FAILURE 420 DETAIL; -- [11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)  
RMAN> LIST FAILURE ALL; -- [11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)

RMAN> LIST RESTORE POINT ALL;

**REPORT command**  
Report backup status: database, files, and backups. Perform detailed analyses of the content of the recovery catalog.  
  
REPORT  
{{NEED BACKUP [{INCREMENTAL | DAYS} [=] integer| REDUNDANCY [=] integer | RECOVERY WINDOW OF integer DAYS)]  
| UNRECOVERABLE  
}  
reportObject  
| SCHEMA [atClause]  
| OBSOLETE [obsOperandList]  
}  
[DEVICE TYPE deviceSpecifier [,deviceSpecifier]... ]  
  
reportObject::=  
[ DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...  
| DATABASE [SKIP TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
]  
  
atClause::=  
{AT TIME [=] 'date\_string' | AT SCN [=] integer|AT SEQUENCE [=] integer THREAD [=] integer  
}  
  
obsOperandList::=  
[REDUNDANCY [=] integer | RECOVERY WINDOW OF integer DAYS | ORPHAN]...  
  
RMAN> REPORT OBSOLETE;  
RMAN> REPORT NEED BACKUP;  
RMAN> REPORT NEED BACKUP DAYS=5;  
RMAN> REPORT NEED BACKUP REDUNDANCY=3;  
RMAN> REPORT NEED BACKUP RECOVERY WINDOW OF 7 DAYS;  
RMAN> REPORT NEED BACKUP DATABASE;  
RMAN> REPORT NEED BACKUP INCREMENTAL 1;  
RMAN> REPORT UNRECOVERABLE;  
RMAN> REPORT SCHEMA;  
RMAN> REPORT SCHEMA AT TIME 'sysdate-20/1440';

**CHANGE command**  
Update the status of a backup in the RMAN repository. Mark a backup piece, image copy, or archived redo log as having the status UNAVAILABLE or AVAILABLE; remove the repository record for a backup or copy; override the retention policy for a backup or copy; update the recovery catalog with the DB\_UNIQUE\_NAME for the target database.  
  
CHANGE {BACKUP | COPY} [OF listObjList] [maintQualifier [maintQualifier]...]  
{AVAILABLE | UNAVAILABLE | UNCATALOG | keepOption}  
[DEVICE TYPE deviceSpecifier [, deviceSpecifier]...];  
  
CHANGE archivelogRecordSpecifier {AVAILABLE | UNAVAILABLE | UNCATALOG | keepOption}  
[DEVICE TYPE deviceSpecifier [, deviceSpecifier]...];  
  
CHANGE recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...  
{AVAILABLE | UNAVAILABLE | UNCATALOG | keepOption}  
[DEVICE TYPE deviceSpecifier [, deviceSpecifier]...];  
  
listObjList::=  
[DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| archivelogRecordSpecifier  
| DATABASE [SKIP TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| CONTROLFILE  
| SPFILE  
]...  
  
recordSpec::=  
{{BACKUPPIECE | PROXY}  
{'media\_handle' [, 'media\_handle']... | primary\_key [, primary\_key]... | TAG [=] ['] tag\_name [']  
}  
| BACKUPSET primary\_key [, primary\_key]...  
| {CONTROLFILECOPY | DATAFILECOPY}  
{{primary\_key [, primary\_key]... | 'filename' [, 'filename']...}  
| TAG [=] ['] tag\_name ['] [, ['] tag\_name [']]...  
}  
| ARCHIVELOG {primary\_key [, primary\_key]... | 'filename' [, 'filename']...}  
}  
  
RMAN> CHANGE BACKUPSET 666 KEEP FOREVER;  
RMAN> CHANGE BACKUPSET 431 KEEP FOREVER NOLOGS;  
RMAN> CHANGE BACKUPSET 100 UNAVAILABLE;  
RMAN> CHANGE BACKUPSET 123 NOKEEP;  
RMAN> CHANGE BACKUPSET 121,122,127,203,300 UNCATALOG;  
RMAN> CHANGE BACKUP OF DATABASE TAG=’abc’ UNAVAILABLE;  
RMAN> CHANGE BACKUP OF DATABASE DEVICE TYPE DISK UNAVAILABLE;

RMAN> CHANGE COPY OF DATABASE CONTROLFILE NOKEEP;

RMAN> CHANGE BACKUP OF SPFILE COMPLETED BEFORE 'SYSDATE-3' UNAVAILABLE;  
RMAN> CHANGE BACKUP TAG 'consistent\_db\_bkup' KEEP FOREVER;

RMAN> CHANGE BACKUP TAG 'consistent\_db\_bkup' DATABASE KEEP FOREVER;

RMAN> CHANGE BACKUP TAG 'consistent\_db\_bkup' KEEP FOREVER NOLOGS;  
RMAN> CHANGE BACKUP TAG 'consistent\_db\_bkup' NOKEEP;  
  
RMAN> CHANGE ARCHIVELOG ALL UNCATALOG;  
RMAN> CHANGE CONTROLFILECOPY '/tmp/cf.cpy' UNCATALOG;  
RMAN> CHANGE FAILURE 5 PRIORITY LOW;

RMAN> CHANGE BACKUP FOR DB\_UNIQUE\_NAME standby1 RESET DB\_UNIQUE\_NAME;

RMAN> CHANGE BACKUP FOR DB\_UNIQUE\_NAME standby3 RESET DB\_UNIQUE\_NAME TO standby2;

RMAN> CHANGE DB\_UNIQUE\_NAME FROM rdbms4 TO rdbms\_dev;

**CROSSCHECK command**  
Check whether files managed by RMAN, such as archived logs, datafile copies, and backup pieces, still exist on disk or tape.  
  
CROSSCHECK  
{{BACKUP [OF listObjList] | COPY [OF listObjList] | archivelogRecordSpecifier} [maintQualifier [maintQualifier]...]  
| recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]  
};  
  
listObjList::=  
[ DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| archivelogRecordSpecifier  
| DATABASE [SKIP TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| CONTROLFILE  
| SPFILE  
]...  
  
recordSpec::=  
{{ BACKUPPIECE | PROXY }  
{ 'media\_handle' [, 'media\_handle']...| primary\_key [, primary\_key]... | TAG [=] ['] tag\_name ['] }  
| BACKUPSET primary\_key [, primary\_key]...  
| { CONTROLFILECOPY | DATAFILECOPY }  
{ {primary\_key [, primary\_key]... | 'filename' [, 'filename']...}  
| TAG [=] ['] tag\_name ['] [, ['] tag\_name [']]...  
}  
| ARCHIVELOG { primary\_key [, primary\_key]... | 'filename' [, 'filename']... }  
}  
  
RMAN> CROSSCHECK BACKUP;  
RMAN> CROSSCHECK BACKUP TAG=’full db’;  
RMAN> CROSSCHECK BACKUP COMPLETED BETWEEN ‘SYSDATE-7’ AND ‘SYSDATE–1’;  
RMAN> CROSSCHECK BACKUP COMPLETED BETWEEN '01-JAN-10' AND '14-FEB-10';  
RMAN> CROSSCHECK BACKUP DEVICE TYPE sbt COMPLETED BETWEEN '01-AUG-09' AND '31-DEC-09';  
RMAN> CROSSCHECK BACKUP DEVICE TYPE DISK COMPLETED BETWEEN '01-JAN-10' AND '23-MAR-10';  
  
RMAN> CROSSCHECK BACKUP OF DATABASE;  
RMAN> CROSSCHECK BACKUP OF TABLESPACE warehouse;  
RMAN> CROSSCHECK BACKUP OF TABLESPACE userd COMPLETED BEFORE 'SYSDATE-14';  
RMAN> CROSSCHECK BACKUP OF TABLESPACES gld, invd;  
RMAN> CROSSCHECK BACKUP OF DATAFILE 9;  
RMAN> CROSSCHECK BACKUP OF DATAFILE 4 COMPLETED AFTER 'SYSDATE-14';  
RMAN> CROSSCHECK BACKUP OF DATAFILE "?/oradata/dwh/system01.dbf" COMPLETED AFTER 'SYSDATE-14';  
RMAN> CROSSCHECK BACKUP OF CONTROLFILE;  
RMAN> CROSSCHECK BACKUP OF SPFILE;  
RMAN> CROSSCHECK BACKUP OF ARCHIVELOG ALL;  
RMAN> CROSSCHECK BACKUP OF ARCHIVELOG ALL SPFILE;  
  
RMAN> CROSSCHECK COPY;  
RMAN> CROSSCHECK COPY OF DATABASE;  
RMAN> CROSSCHECK DATAFILECOPY 113, 114, 115;  
RMAN> CROSSCHECK CONTROLFILECOPY '/tmp/control01.ctl';  
RMAN> CROSSCHECK ARCHIVELOG ALL;  
RMAN> CROSSCHECK BACKUPSET;  
RMAN> CROSSCHECK BACKUPSET 1338, 1339, 1340;  
RMAN> CROSSCHECK BACKUPPIECE TAG = 'nightly\_backup';  
RMAN> CROSSCHECK PROXY 789;

  
**SQL command**  
Execute a SQL statement from within [Recovery Manager](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html).  
  
SQL [CHANNEL ‘channel\_id’] ‘command’;  
  
RMAN> SQL 'ALTER TABLESPACE users ONLINE';  
RMAN> SQL 'ALTER DATABASE DATAFILE 64 OFFLINE';  
RMAN> SQL "ALTER SYSTEM ARCHIVE LOG CURRENT";  
RMAN> SQL "ALTER SYSTEM SWITCH LOGFILE";  
RMAN> SQL "ALTER DATABASE BACKUP CONTROLFILE TO TRACE";  
RMAN> SQL "ALTER TABLESPACE users ADD DATAFILE ''/disk1/ora/users02.dbf'' SIZE 100K AUTOEXTEND ON NEXT 10K MAXSIZE 100K";

**RESTORE command**  
Restore files from backup sets or from disk copies to the default or a new location.  
  
RESTORE  
[(] restoreObject [(restoreSpecOperand [restoreSpecOperand]...] [)]...  
[ CHANNEL ['] channel\_id [']  
| PARMS [=] 'channel\_parms'  
| FROM { BACKUPSET | DATAFILECOPY }  
| untilClause  
| FROM TAG [=] ['] tag\_name [']  
| VALIDATE  
| DEVICE TYPE deviceSpecifier [, deviceSpecifier]...  
]...;  
  
restoreObject::=  
{ CONTROLFILE [TO 'filename']  
| DATABASE [SKIP [FOREVER] TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| archivelogRecordSpecifier  
| SPFILE [TO [PFILE] 'filename']  
}  
  
restoreSpecOperand::=  
{ CHANNEL ['] channel\_id ['] | FROM TAG [=] ['] tag\_name ['] | PARMS [=] 'channel\_parms'  
| FROM {AUTOBACKUP [{MAXSEQ | MAXDAYS} [=] integer)]... | 'media\_handle'}  
}  
  
RMAN> RESTORE DATABASE;  
RMAN> RESTORE DATABASE VALIDATE;  
RMAN> RESTORE DATABASE PREVIEW;  
RMAN> RESTORE DATABASE PREVIEW SUMMARY;  
RMAN> RESTORE DATABASE SKIP TABLESPACE temp, history;  
RMAN> RESTORE DATABASE UNTIL SCN 154876;  
  
RMAN> RESTORE TABLESPACE users;  
RMAN> RESTORE TABLESPACE dwh1, dwh2;  
RMAN> RESTORE TABLESPACE tbs1 PREVIEW;  
RMAN> RESTORE TABLESPACE users VALIDATE;  
  
RMAN> RESTORE DATAFILE 45;  
RMAN> RESTORE DATAFILE 23 PREVIEW;  
RMAN> RESTORE DATAFILE 12 VALIDATE;  
  
RMAN> RESTORE CONTROLFILE;  
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;  
RMAN> RESTORE CONTROLFILE FROM TAG 'monday\_cf\_backup';

RMAN> RESTORE CONTROLFILE FROM '/u01/control01.ctl';

RMAN> RESTORE CONTROLFILE VALIDATE;  
RMAN> RESTORE CONTROLFILE TO '/tmp/autobkp.dbf' FROM AUTOBACKUP MAXSEQ 20 MAXDAYS 150;  
  
RMAN> RESTORE SPFILE;  
RMAN> RESTORE SPFILE FROM AUTOBACKUP;  
RMAN> RESTORE ARCHIVELOG ALL VALIDATE;  
RMAN> RESTORE ARCHIVELOG ALL PREVIEW;  
RMAN> RESTORE ARCHIVELOG ALL PREVIEW RECALL;  
RMAN> RESTORE ARCHIVELOG ALL DEVICE TYPE sbt PREVIEW;

RMAN> RESTORE ARCHIVELOG LOW LOGSEQ 78311 HIGH LOGSEQ 78340 THREAD 1 ALL;

RMAN> RESTORE ARCHIVELOG FROM LOGSEQ=21531 UNTIL LOGSEQ=21590;

RMAN> RESTORE STANDBY CONTROLFILE FROM TAG 'forstandby';

RMAN> RESTORE CLONE CONTROLFILE TO '+DATA/pcrd/data2/control02.ctl' FROM '+DATA/pcrd/data1/control01.ctl';

Restore the control file, (to all locations specified in the parameter file) then restore the database, using that control file:  
STARTUP NOMOUNT;  
RUN  
{  
ALLOCATE CHANNEL c1 DEVICE TYPE sbt;  
RESTORE CONTROLFILE;  
ALTER DATABASE MOUNT;  
RESTORE DATABASE;  
}

**RECOVER command**  
Perform media recovery from RMAN backups and copies. Apply redo log files and incremental backups to datafiles or data blocks restored from backup or datafile copies, to update them to a specified time.  
  
RECOVER [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]  
recoverObject [recoverOptionList];  
  
recoverObject::=  
{ DATABASE  
[ untilClause  
| [untilClause] SKIP [FOREVER] TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']]...  
| DATAFILE datafileSpec [, datafileSpec]...  
}  
  
recoverOptionList::=  
{ DELETE ARCHIVELOG [MAXSIZE {integer [K | M | G]}]  
| CHECK READONLY  
| NOREDO  
| {FROM TAG | ARCHIVELOG TAG} [=] ['] tag\_name [']  
...  
}  
  
RMAN> RECOVER DATABASE;  
RMAN> RECOVER DATABASE NOREDO;  
RMAN> RECOVER DATABASE SKIP TABLESPACE temp;  
RMAN> RECOVER DATABASE SKIP FOREVER TABLESPACE exam;  
RMAN> RECOVER DATABASE UNTIL SCN 154876;  
  
RMAN> RECOVER TABLESPACE users;  
RMAN> RECOVER TABLESPACE dwh DELETE ARCHIVELOG MAXSIZE 2M;  
RMAN> RECOVER DATAFILE 33;

RMAN> RECOVER DATAFILE 3 BLOCK 116 DATAFILE 4 BLOCK 10;

RMAN> RECOVER DATAFILE 2 BLOCK 204 DATAFILE 9 BLOCK 109 FROM TAG=sundaynight;

RMAN> RECOVER DATAFILECOPY '/disk1/img.df' UNTIL TIME 'SYSDATE-7';  
RMAN> RECOVER COPY OF DATABASE WITH TAG 'incr';  
RMAN> RECOVER COPY OF DATABASE WITH TAG 'upd' UNTIL TIME 'SYSDATE - 7';  
RMAN> RECOVER CORRUPTION LIST;  
  
Restore and recover the whole database  
RMAN> STARTUP FORCE MOUNT;  
RMAN> RESTORE DATABASE;  
RMAN> RECOVER DATABASE;  
RMAN> ALTER DATABASE OPEN;  
  
Restore and recover a tablespace  
RMAN> SQL 'ALTER TABLESPACE users OFFLINE';  
RMAN> RESTORE TABLESPACE users;  
RMAN> RECOVER TABLESPACE users;  
RMAN> SQL 'ALTER TABLESPACE users ONLINE';  
  
Restore and recover a datafile  
RMAN> SQL 'ALTER DATABASE DATAFILE 64 OFFLINE';  
RMAN> RESTORE DATAFILE 64;  
RMAN> RECOVER DATAFILE 64;  
RMAN> SQL 'ALTER DATABASE DATAFILE 64 ONLINE';  
  
Steps for media recovery:  
1. [Mount or open](http://satya-dba.blogspot.com/2009/01/startupshutdown-options.html) the Oracle database. Mount the database when performing whole database recovery, or open the database when performing online tablespace/datafile recovery.  
2. To perform incomplete recovery, use the SET UNTIL command to specify the time, SCN, or log sequence number at which recovery terminates. Alternatively, specify the UNTIL clause on the RESTORE and RECOVER commands.  
3. Restore the necessary files with the RESTORE command.  
4. Recover the datafiles with the RECOVER command.  
5. Place the database in its normal state. For example, open it or bring recovered tablespaces/datafiles online.

**DELETE command**  
Delete backups and copies, remove references to them from the recovery catalog, and update their control file records to status DELETED.  
  
DELETE [FORCE] [NOPROMPT]  
{[EXPIRED]  
{  
{BACKUP [OF listObjectList] | COPY [OF listObectjList] | archivelogRecordSpecifier  
} [maintQualifier [maintQualifier]...]  
| recordSpec [DEVICE TYPE deviceSpecifier [, deviceSpecifier]...]  
}  
| OBSOLETE [REDUNDANCY [=] integer | RECOVERY WINDOW OF integer DAYS | ORPHAN] [DEVICE TYPE (deviceSpecifier [, deviceSpecifier]...]  
};  
  
recordSpec::=  
{ { BACKUPPIECE | PROXY }  
{ 'media\_handle' [, 'media\_handle']...| primary\_key [, primary\_key]...| TAG [=] ['] tag\_name ['] }  
| BACKUPSET primary\_key [, primary\_key]...  
| { CONTROLFILECOPY | DATAFILECOPY }  
{ {primary\_key [, primary\_key]... | 'filename' [, 'filename']...}  
| TAG [=] ['] tag\_name ['] [, ['] tag\_name [']]...  
}  
| ARCHIVELOG { primary\_key [, primary\_key]... | 'filename' [, 'filename']... }  
  
listObjectList::=  
[ DATAFILE datafileSpec [, datafileSpec]...  
| TABLESPACE ['] tablespace\_name ['] [, ['] tablespace\_name [']]...  
| archivelogRecordSpecifier  
| DATABASE [SKIP TABLESPACE [']tablespace\_name['] [, [']tablespace\_name[']] ...]  
| CONTROLFILE  
| SPFILE  
]...  
  
RMAN> DELETE OBSOLETE;  
RMAN> DELETE NOPROMPT OBSOLETE;  
RMAN> DELETE NOPROMPT OBSOLETE RECOVERY WINDOW OF 7 DAYS;  
RMAN> DELETE EXPIRED BACKUP;  
RMAN> DELETE EXPIRED BACKUP DEVICE TYPE sbt;  
RMAN> DELETE BACKUP OF DATABASE LIKE '/tmp%';  
RMAN> DELETE NOPROMPT EXPIRED BACKUP OF TABLESPACE userd COMPLETED BEFORE 'SYSDATE-14';  
RMAN> DELETE BACKUP OF SPFILE TABLESPACE users DEVICE TYPE SBT;  
  
RMAN> DELETE ARCHIVELOG ALL;

RMAN> DELETE ARCHIVELOG ALL COMPLETED BEFORE 'sysdate-2';

RMAN> DELETE ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE TYPE SBT;

RMAN> DELETE ARCHIVELOG ALL LIKE '%755153075%';

RMAN> DELETE ARCHIVELOG UNTIL SEQUENCE=79228;

RMAN> DELETE FORCE ARCHIVELOG ALL COMPLETED BEFORE 'sysdate-1.5';

RMAN> DELETE FORCE ARCHIVELOG UNTIL SEQUENCE=16364;

RMAN> DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE = 7300;  
RMAN> DELETE EXPIRED ARCHIVELOG ALL;  
RMAN> DELETE NOPROMPT EXPIRED ARCHIVELOG ALL;  
RMAN> DELETE BACKUPSET 101, 102, 103;  
RMAN> DELETE NOPROMPT BACKUPSET TAG weekly\_bkup;  
RMAN> DELETE FORCE NOPROMPT BACKUPSET TAG weekly\_bkup;  
  
RMAN> DELETE DATAFILECOPY "+DG\_DATA/db/datafile/system.259.699468079";  
RMAN> DELETE CONTROLFILECOPY '/tmp/cntrlfile.copy';  
RMAN> DELETE BACKUP DEVICE TYPE SBT;  
RMAN> DELETE BACKUP DEVICE TYPE DISK;  
RMAN> DELETE COPY;  
RMAN> DELETE EXPIRED COPY;  
RMAN> DELETE COPY TAG 'lastest';

**DROP DATABASE command**  
Delete the target database from disk and unregisters it.  
RMAN> DROP DATABASE;  
RMAN> DROP DATABASE NOPROMPT;  
RMAN> DROP DATABASE INCLUDING BACKUPS;  
RMAN> DROP DATABASE INCLUDING BACKUPS NOPROMPT;

**DUPLICATE command**  
Use backups of the target database to create a duplicate database that we can use for testing purposes or to create a standby database.  
RMAN> DUPLICATE TARGET DATABASE;  
RMAN> DUPLICATE TARGET DATABASE TO dwhdb;  
RMAN> DUPLICATE TARGET DATABASE TO test PFILE=/u01/apps/db/inittest.ora;  
RMAN> DUPLICATE TARGET DATABASE TO devdb NOFILENAMECHECK;  
RMAN> DUPLICATE DATABASE 'prod' DBID 139525561 TO 'dupdb' NOFILENAMECHECK;

RMAN> DUPLICATE DATABASE TO "cscp" NOFILENAMECHECK BACKUP LOCATION '/apps/oracle/backup';

RMAN> DUPLICATE TARGET DATABASE TO dup FROM ACTIVE DATABASE NOFILENAMECHECK PASSWORD FILE SPFILE;  
  
RMAN> DUPLICATE TARGET DATABASE TO dupdb  
LOGFILE GROUP 1 ('?/dbs/dupdb\_log\_1\_1.f','?/dbs/dupdb\_log\_1\_2.f') SIZE 200K, GROUP 2 ('?/dbs/dupdb\_log\_2\_1.f','?/dbs/dupdb\_log\_2\_2.f') SIZE 200K REUSE;  
RMAN> DUPLICATE TARGET DATABASE TO dup FOR STANDBY FROM ACTIVE DATABASE PASSWORD FILE SPFILE PARAMETER\_VALUE\_CONVERT '/disk1', '/disk2'  
SET DB\_FILE\_NAME\_CONVERT '/disk1','/disk2'  
SET LOG\_FILE\_NAME\_CONVERT '/disk1','/disk2'  
SET SGA\_MAX\_SIZE 200M SET SGA\_TARGET 125M;

RMAN> DUPLICATE TARGET DATABASE FOR STANDBY NOFILENAMECHECK;

RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE;

RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE NOFILENAMECHECK;

RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE

SPFILE PARAMETER\_VALUE\_CONVERT '/stg/','/muc/'

SET "DB\_UNIQUE\_NAME"="muc"

SET LOG\_FILE\_NAME\_CONVERT '/stg/','/muc/'

SET DB\_FILE\_NAME\_CONVERT '/stg/','/muc/'

DORECOVER;

RMAN> DUPLICATE DATABASE TO newdb

UNTIL RESTORE POINT firstquart12

DB\_FILE\_NAME\_CONVERT='/u01/prod1/dbfiles/','/u01/newdb/dbfiles'

PFILE = '/u01/newdb/admin/init.ora';

**SWITCH command**  
Specify that a datafile copy is now the current datafile, i.e. the datafile pointed to by the control file. This command is equivalent to the SQL statement ALTER DATABASE RENAME FILE as it applies to datafiles.  
RMAN> SWITCH DATABASE TO COPY;  
RMAN> SWITCH TABLESPACE users TO COPY;  
RMAN> SWITCH DATAFILE ALL;  
RMAN> SWITCH DATAFILE '/disk1/tols.dbf' TO DATAFILECOPY '/disk2/tols.copy';  
RMAN> SWITCH DATAFILE "+DG\_OLD/db/datafile/sysaux.260.699468081" TO COPY;  
RMAN> SWITCH TEMPFILE 1;  
RMAN> SWITCH TEMPFILE 1 TO '/newdisk/dbs/temp1.f';  
RMAN> SWITCH TEMPFILE ALL;

RMAN> SWITCH CLONE DATAFILE ALL;

**CATALOG command**  
Add information about file copies and user-managed backups to the catalog repository.  
RMAN> CATALOG DATAFILECOPY '/disk1/old\_datafiles/01\_10\_2009/users01.dbf';  
RMAN> CATALOG DATAFILECOPY '/disk2/backup/users01.bkp' LEVEL 0;  
RMAN> CATALOG CONTROLFILECOPY '/disk3/backup/cf\_copy.bkp';  
RMAN> CATALOG ARCHIVELOG '/disk1/arch1\_731.dbf', '/disk1/arch1\_732.dbf';  
RMAN> CATALOG BACKUPPIECE '/disk1/c-874220581-20090428-01';  
RMAN> CATALOG LIKE '/backup';  
RMAN> CATALOG START WITH '/fs2/arch';  
RMAN> CATALOG START WITH '/disk2/archlog' NOPROMPT;  
RMAN> CATALOG START WITH '+dg2';  
RMAN> CATALOG [RECOVERY AREA](http://satya-dba.blogspot.com/2009/02/flash-recovery-area.html);

**ALLOCATE command**  
Establish a channel, which is a connection between [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) and a database instance.  
RMAN> ALLOCATE CHANNEL c1 DEVICE TYPE sbt;  
RMAN> ALLOCATE CHANNEL ch DEVICE TYPE DISK FORMAT ‘C:\ORACLEBKP\DB\_U%’;  
RMAN> ALLOCATE CHANNEL t1 DEVICE TYPE DISK CONNECT 'sys/pwd@bkp1’;  
RMAN> ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS 'ENV=(OB\_MEDIA\_FAMILY=wholedb\_mf)';  
RMAN> ALLOCATE CHANNEL t1 DEVICE TYPE sbt PARMS 'ENV=(OB\_DEVICE\_1=tape1, OB\_DEVICE\_2=tape3)';  
RMAN> ALLOCATE CHANNEL t1 TYPE 'sbt\_tape' PARMS='SBT\_LIBRARY=/usr/openv/netbackup/bin/libobk.so.1';  
RMAN> ALLOCATE CHANNEL t1 TYPE 'sbt\_tape' SEND "NB\_ORA\_CLIENT=CLIENT\_MACHINE\_NAME";  
RMAN> ALLOCATE CHANNEL 'dev1' TYPE 'sbt\_tape' PARMS 'ENV=OB2BARTYPE=ORACLE8, OB2APPNAME=ORCL, OB2BARLIST=MACHINENAME\_ORCL\_ARCHLOGS)';

RMAN> ALLOCATE CHANNEL y1 TYPE DISK RATE 70M;

RMAN> ALLOCATE AUXILIARY CHANNEL ac1 TYPE DISK;  
RMAN> ALLOCATE AUXILIARY CHANNEL ac2 DEVICE TYPE sbt;  
  
ALLOCATE CHANNEL FOR MAINTENANCE - allocate a channel in preparation for issuing maintenance commands such as DELETE.  
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK;  
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK FORMAT "/disk2/%U";  
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT '@test1';  
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt;  
RMAN> ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE sbt PARMS 'SBT\_LIBRARY=/usr/local/oracle/backup/lib/libobk.so, ENV=(OB\_DEVICE\_1=tape2)';

**RELEASE CHANNEL command**  
Release a channel that was allocated with an ALLOCATE CHANNEL or ALLOCATE CHANNEL FOR MAINTENANCE command.  
RMAN> RELEASE CHANNEL;  
RMAN> RELEASE CHANNEL c1;

**BLOCKRECOVER command**  
Will recover the corrupted blocks.  
RMAN> BLOCKRECOVER CORRUPTION LIST;  
RMAN> BLOCKRECOVER DATAFILE 8 BLOCK 22;  
RMAN> BLOCKRECOVER DATAFILE 7 BLOCK 233,235 DATAFILE 4 BLOCK 101;  
RMAN> BLOCKRECOVER DATAFILE 2 BLOCK 12,13 DATAFILE 3 BLOCK 5,98,99 DATAFILE 4 BLOCK 19;  
RMAN> BLOCKRECOVER DATAFILE 3 BLOCK 2,4,5 TABLESPACE sales DBA 4194405,4194412 FROM DATAFILECOPY;  
RMAN> BLOCKRECOVER TABLESPACE dwh DBA 4194404,4194405 FROM TAG "weekly";  
RMAN> BLOCKRECOVER TABLESPACE dwh DBA 94404 RESTORE UNTIL TIME 'SYSDATE-2';

**ADVISE FAILURE command** (From [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html))  
Display repair options.  
RMAN> ADVISE FAILURE;  
RMAN> ADVISE FAILURE 555, 242;  
RMAN> ADVISE FAILURE ALL;  
RMAN> ADVISE FAILURE CRITICAL;  
RMAN> ADVISE FAILURE HIGH;  
RMAN> ADVISE FAILURE LOW;  
RMAN> ADVISE FAILURE HIGH EXCLUDE FAILURE 625;

**REPAIR FAILURE command** (From [Oracle 11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html))  
Repair one or more failures recorded in the automated diagnostic repository.  
RMAN> REPAIR FAILURE;  
RMAN> REPAIR FAILURE PREVIEW;  
RMAN> REPAIR FAILURE NOPROMPT;  
RMAN> REPAIR FAILURE USING ADVISE OPTION integer;

**VALIDATE command**  
Examine a backup set and report whether its data is intact. RMAN scans all of the backup pieces in the specified backup sets and looks at the checksums to verify that the contents can be successfully restored.  
RMAN> VALIDATE BACKUPSET 218;  
RMAN> VALIDATE BACKUPSET 3871, 3890;  
RMAN> VALIDATE DATABASE; -- [11g R1](http://satya-dba.blogspot.com/2009/01/whats-new-in-11g.html)

RMAN> VALIDATE CHECK LOGICAL DATABASE;

RMAN> VALIDATE SKIP INACCESSIBLE DATABASE;

RMAN> VALIDATE COPY OF DATABASE;

RMAN> VALIDATE TABLESPACE dwh;  
RMAN> VALIDATE COPY OF TABLESPACE dwh;  
RMAN> VALIDATE DATAFILE 2;  
RMAN> VALIDATE DATAFILE 4,8;  
RMAN> VALIDATE DATAFILE 4 BLOCK 56;

RMAN> VALIDATE DATAFILE 8 SECTION SIZE = 200M;

RMAN> VALIDATE CURRENT CONTROLFILE;  
RMAN> VALIDATE SPFILE;  
RMAN> VALIDATE RECOVERY FILES;  
RMAN> VALIDATE [RECOVERY AREA](http://satya-dba.blogspot.com/2009/02/flash-recovery-area.html);  
RMAN> VALIDATE DB\_RECOVERY\_FILE\_DEST;

**SPOOL command**  
Write RMAN output to a log file.  
RMAN> SPOOL LOG TO '/tmp/spool.log';  
RMAN> SPOOL LOG TO '/tmp/backup.log' APPEND;  
RMAN> SPOOL LOG OFF;

**run command**  
Execute a sequence of one or more [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) commands, which are one or more statements executed within the braces of RUN.  
RMAN> run {  
ALLOCATE CHANNEL c1 TYPE DISK FORMAT '/orabak/%U';  
BACKUP TABLESPACE users;  
}  
RMAN> run {  
ALLOCATE CHANNEL c1 TYPE DISK FORMAT '&1/%U';  
BACKUP TABLESPACE &2;  
}  
RMAN> run  
{  
ALLOCATE CHANNEL dev1 DEVICE TYPE DISK FORMAT '/fs1/%U';  
ALLOCATE CHANNEL dev2 DEVICE TYPE DISK FORMAT '/fs2/%U';   
BACKUP(TABLESPACE system,fin,mark FILESPERSET 20) (DATAFILE 2,3,6);  
}

**CREATE SCRIPT command**  
Create a stored script and store it in the recovery catalog.  
  
RMAN> CREATE SCRIPT backup\_whole  
COMMENT "backup whole database and archived redo log files"  
{BACKUP INCREMENTAL LEVEL 0 TAG backup\_whole FORMAT "/disk2/backup/%U" DATABASE PLUS ARCHIVELOG;}  
  
RMAN> CREATE SCRIPT backup\_ts\_users  
COMMENT 'tablespace users backup'  
{ALLOCATE CHANNEL c1 TYPE DISK FORMAT 'c:\temp\%U';  
BACKUP TABLESPACE users;}  
  
RMAN> CREATE SCRIPT df {BACKUP DATAFILE &1 TAG &2.1 FORMAT '/disk1/&3\_%U';}  
RMAN> CREATE SCRIPT backup\_ts\_users FROM FILE 'backup\_ts\_users.rman';  
RMAN> CREATE GLOBAL SCRIPT gl\_backup\_db {BACKUP DATABASE PLUS ARCHIVELOG;}  
RMAN> CREATE GLOBAL SCRIPT backup\_db  
COMMENT "back up any database from the recovery catalog, with logs"  
{BACKUP DATABASE;}

**PRINT SCRIPT command**  
Display a stored script.  
RMAN> PRINT SCRIPT backup\_db;  
RMAN> PRINT GLOBAL SCRIPT backup\_db;  
RMAN> PRINT GLOBAL SCRIPT gl\_backup\_db TO FILE "/tmp/gl\_backupdb.rman";

**REPLACE SCRIPT command**  
Replace an existing script stored in the recovery catalog. If the script does not exist, then REPLACE SCRIPT creates it.  
RMAN> REPLACE SCRIPT backup\_db {BACKUP DATABASE PLUS ARCHIVELOG;}  
RMAN> REPLACE SCRIPT df {BACKUP DATAFILE &1 TAG &2.1 FORMAT '&3\_%U';}  
RMAN> REPLACE GLOBAL SCRIPT backup\_db {BACKUP DATABASE PLUS ARCHIVELOG;}  
RMAN> REPLACE GLOBAL SCRIPT gl\_full\_bkp FROM FILE '/tmp/script\_file.txt';

**EXECUTE SCRIPT command**  
Run an [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) stored script.  
RMAN> RUN {EXECUTE SCRIPT backup\_whole;}  
RMAN> RUN {EXECUTE SCRIPT backup\_ts\_any USING 'example';}  
RMAN> RUN {EXECUTE SCRIPT backup\_df USING 3 test\_backup df3;}  
RMAN> RUN {EXECUTE GLOBAL SCRIPT global\_backup\_db;}

**DELETE SCRIPT command**  
Delete a stored script from the recovery catalog.  
RMAN> DELETE SCRIPT backup\_db;  
RMAN> DELETE GLOBAL SCRIPT global\_backup\_db;

**[FLASHBACK DATABASE](http://satya-dba.blogspot.com/2009/02/flashback.html) command**  
Return the database to its state at a previous time or SCN.  
RMAN> FLASHBACK DATABASE TO SCN 411010;  
RMAN> FLASHBACK DATABASE TO RESTORE POINT 'before\_update';

**[TRANSPORT TABLESPACE](http://satya-dba.blogspot.com/2010/01/oracle-transportable-tablespaces-tts.html) command**  
Create transportable tablespace sets from backup for one or more tablespaces.  
RMAN> TRANSPORT TABLESPACE example, tools  
TABLESPACE DESTINATION '/disk1/trans' AUXILIARY DESTINATION '/disk1/aux' UNTIL TIME 'SYSDATE-15/1440';  
RMAN> TRANSPORT TABLESPACE exam  
TABLESPACE DESTINATION '/disk1/trans' AUXILIARY DESTINATION '/disk1/aux' [DATAPUMP](http://satya-dba.blogspot.com/2009/06/datapump-in-oracle.html) DIRECTORY dpdir DUMP FILE 'dmpfile.dmp' IMPORT SCRIPT 'impscript.sql' EXPORT LOG 'explog.log';

**CONVERT command**  
Convert datafile formats for transporting tablespaces and databases across platforms.  
RMAN> CONVERT DATABASE NEW DATABASE 'prodwin' TRANSPORT SCRIPT '/tmp/convertdb/transportscript' TO PLATFORM 'Microsoft Windows IA (32-bit)' DB\_FILE\_NAME\_CONVERT '/disk1/oracle/dbs','/tmp/convertdb';  
RMAN> CONVERT DATABASE ON DESTINATION PLATFORM CONVERT SCRIPT '/tmp/convertdb/convertscript.rman' TRANSPORT SCRIPT '/tmp/convertdb/transportscript.sql' NEW DATABASE 'prodwin' FORMAT '/tmp/convertdb/%U';  
RMAN> CONVERT DATABASE ON DESTINATION PLATFORM CONVERT SCRIPT '/tmp/convert\_newdb.rman' TRANSPORT SCRIPT '/tmp/transport\_newdb.sql' NEW DATABASE 'prodaix' DB\_FILE\_NAME\_CONVERT '/u01/oradata/datafile','+DATA';  
  
RMAN> CONVERT TABLESPACE tbs\_2 FORMAT '/tmp/tbs\_2\_%U.df';  
RMAN> CONVERT TABLESPACE fin, hr TO PLATFORM 'Solaris[tm] OE (32-bit)';  
RMAN> CONVERT TABLESPACE fin, hr TO PLATFORM 'Solaris[tm] OE (32-bit)' FORMAT '/tmp/transport\_to\_solaris/%U';  
  
RMAN> CONVERT DATAFILE '/disk1/oracle/dbs/tbs\_f1.df', '/disk1/oracle/dbs/ax1.f' FORMAT '+DATAFILE';  
RMAN> CONVERT DATAFILE '/u01/oradata/datafile/system.dbf' FROM PLATFORM 'Linux x86 64-bit' FORMAT '+DATA/system.dbf';  
RMAN> CONVERT DATAFILE  
'/tmp/from\_solaris/fin/fin01.dbf', '/tmp/from\_solaris/fin/fin02.dbf',  
'/tmp/from\_solaris/hr/hr01.dbf', '/tmp/from\_solaris/hr/hr02.dbf'  
DB\_FILE\_NAME\_CONVERT '/tmp/from\_solaris/fin','/disk2/orahome/dbs/fin', '/tmp/from\_solaris/hr','/disk2/orahome/dbs/hr'  
FROM PLATFORM 'Solaris[tm] OE (64-bit)';

RMAN> CONVERT DATAFILE '/tmp/PSMN.dbf' TO PLATFORM='Solaris Operating System (x86-64)' FROM PLATFORM='Solaris[tm] OE (64-bit)'

FORMAT '/tmp/test/%N.dbf' DB\_FILE\_NAME\_CONVERT='/ui/prod/oracle/oradata/SEARCHP/data/', '/tmp/test';

**EXIT or QUIT Command**  
Exit the RMAN console.  
  
RMAN> exit;  
RMAN> quit;

**SEND command**  
Send a vendor-specific quoted string to one or more specific channels.  
RMAN> SEND 'OB\_DEVICE tape1';

**HOST command**  
Invoke an operating system command-line subshell from within [RMAN](http://satya-dba.blogspot.com/2009/01/rman-was-first-introduced-in-oracle8.html) or run a specific operating system command.  
RMAN> HOST;  
RMAN> HOST 'ls -lt /disk2/\*';  
RMAN> HOST '/bin/mv $ORACLE\_HOME/dbs/\*.arc /disk2/archlog/';