Starting RMAN

```
UNIX> rman target /
UNIX> rman target sys/pwd@target_db
UNIX> rman
RMAN> connect target sys/pwd@target_db

UNIX> rman target sys/pwd@target_db nocatalog # Disable catalog (1)
UNIX> rman target sys/pwd@target_db catalog rman/pwd@catalog_db

UNIX> rman target sys/pwd@target_db auxiliary sys/pwd@aux_db

UNIX> rman target sys/pwd@target_db cmdfile script log script_log [append]
UNIX> rman target sys/pwd@target_db cmdfile script checksyntax
```

Comments:

1. **Catalog database** is NOT connected by default – you need to connect to it explicitly. Specifying **nocatalog** in the command prompt prevents further **"connect catalog"** commands from within RMAN.

RMAN Stored Parameters

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 14 DAYS; (1)
CONFIGURE BACKUP OPTIMIZATION ON; (2)
CONFIGURE DEFAULT DEVICE TYPE TO DISK; (3)
CONFIGURE CONTROLFILE AUTOBACKUP ON; (4)
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
(5)
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; #
default
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET; (6)
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 1 BACKUP TYPE TO BACKUPSET; (7)
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'ENV=(TDPO_OPTFILE=/usr/tivoli/tsm/client/oracle/bin64/tdpo_qaten.opt)'; (7)
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY; (8)
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/ora01/app/oracle/product/10.2.0/server_ee_1/dbs/snapcf_qaten.f'; # default
```

Comments:

1. **Retention policy** choices are:

   - "RECOVERY WINDOW OF X DAYS" (meaning: "keep backups for the at least last X days) or
   - "REDUNDANCY Y" (meaning: "keep at least Y copies of backups") .

   - If FRA is used, backups can stay longer in FRA than retention policy requires – depending on available space in FRA

- These *'out of retention policy'* backups are marked as *'reclaimable'* space and will be removed as needed when database needs more space for new backups.

2. **Backup optimization** tells ORACLE to NOT repeat backup or restore operations that have already been completed. I.e. if archived logs have already been backed up – do NOT create *another* backup set.

   - If FRA is used, backup optimization is (almost) mandatory.
   - With *Remove/Create* rotation, using backup optimization is NOT a good idea (it is better to overbackup to make sure fully recoverable backup is available on *this* tape).

3. Set the **default backup device** (DISK or SBT) so that you do NOT need to do it in *every* RMAN command.

4. **Controlfile autobackups MUST be ON**. Be assured, you can never have *too many* CF backups ;-) If set, controlfile autobackups will be made:
   - After every backup operation
   - After every database structure change (i.e. 'adding the tablespace').

5. **Controlfile autobackup format** must have ***default settings*** for the backup to be available in FRA.

6. The main thing to configure here is: **backup type** (COPY, BACKUPSET, COMPRESSED BACKUPSET) and the **# of parallel channels** to run the backup or recovery (PARALLELISM).

7. This is configuration for a **"tape" part of a backup** - *'direct'* connection to external backup server.

8. This setting tells ORACLE to **NOT consider archived logs '*reclaimable*' until they were transferred** to at least one *mandatory* standby destination.

   - Unfortunately, for MAXIMUM PERFORMANCE data guard mode (which is the most popular), ORACLE also recommends configuring ALL standby destinations as *'optional'* (so that *primary* database is NOT affected if *standby* becomes unavailable).
     When this is configured, RMAN will produce *'warning'* message when trying to make this change.
     Check metalink note: **331924.1** for a workaround.

Important Database Parameters

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
control_file_record_keep_time        integer     30

db_recovery_file_dest                string      /ora05/flash_recovery_area
db_recovery_file_dest_size           big integer 120G

log_archive_dest_1                   string    location=use_db_recovery_file_dest
                                                 MANDATORY

db_flashback_retention_target        integer     1440

db_block_checking                    string      LOW
db_block_checksum                    string      TRUE
db_lost_write_protect                string      TYPICAL
```

Block change tracking

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '/oradata/bct01.dbf';
SQL> SELECT * FROM v$block_change_tracking;
```

How to Backup a Database

Common prerequisites for all examples

```
CONFIGURE BACKUP OPTIMIZATION ON;
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # FRA
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'ENV=(TDPO_OPTFILE=/usr/tivoli/tsm/client/oracle/bin64/tdpo_qaten.opt)'; # Tivoli
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
```

When you have lots of disk space for FRA

```
# Requirement: Fast recovery but with the ability to recover data
#              up to 7 days in the past
# Configuration – DISK (FRA) then TAPE (Tivoli), block change tracking

# Prerequisites
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 8 DAYS;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO COPY;

# Solution – daily incrementally updated backup.

run {
   recover copy of database with tag 'incr_updated'
      until time 'sysdate-7';

   backup incremental level 1 for recover of copy with tag 'incr_updated'
      database;
   backup archivelog all;
   backup recovery area;
}

# What happens:
# Day 1 – Level 0 backup is created by 'backup incremental level 1' command
# Days 2+ – Daily level 1 incrementals are created
# Days 8+ - Level 0 copy is recovered to 7 days in the past

# Catches:
# 1. You may need up to apply up to 7 incrementals to recover the database
# 2. Each day after day 7, 'backup recovery area' will send HUGE file to Tivoli
as #    level 0 backup will be always different.
```

When you have a bit less disk space for FRA

```
# Requirement: Fast recovery but limit the amount of backup data on disk
# Configuration – DISK (FRA) then TAPE (Tivoli), block change tracking

# Prerequisites
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET;

# Solution – Compressed backups: Sun - level 0 backup, Mon-Tue, Thu-Sat – level
1,
#            Wed – level 1 cumulative

# Sunday:
run {
   backup incremental level 0 database plus archivelog;
   backup recovery area;
}

# Monday, Tuesday, Thursday, Friday, Saturday
run {
   backup incremental level 1 database plus archivelog;
```

```
   backup recovery area;
}

# Wednesday:

run {
   backup incremental level 1 cumulative database plus archivelog;
   backup recovery area;
}

# Notes:

# 1. Disk to tape synchronization through 'backup recovery area' with level 1
#    backup will be fast as little data is sent
```

When you have little disk space for FRA

```
# Requirement: Absolutely limit the amount of backup data on disk
# Configuration – Level 0 – Tape (Tivoli),
# Arc logs and level 1 - DISK (FRA) then TAPE (Tivoli), block change tracking

# Prerequisites
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO COMPRESSED BACKUPSET;
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 2 BACKUP TYPE TO BACKUPSET;

# Solution – Sun - level 0 backup (tape), Mon-Sat – level 1 disk (FRA)

# Sunday:
backup device type sbt incremental level 0 database plus archivelog;

# Monday-Saturday
run {
   backup incremental level 1 database plus archivelog; # Implied: DISK
   backup recovery area;
}
```

Database Recovery

Disaster Recovery (“empty” system and NO recovery catalog)

```
# First of all, install ORACLE and create database directories

UNIX> export ORACLE_SID=sid
UNIX> rman target /
RMAN> startup force nomount # This will start db instance with default parameters
RMAN> restore spfile from autobackup db_recovery_file_dest='dest' db_name='sid';
RMAN> shutdown immediate
RMAN> startup nomount # This will start db instance with correct parameters
RMAN> restore controlfile from autobackup db_recovery_file_dest='dest'
db_name='sid';
RMAN> alter database mount;
# Register additional archived logs/backups if necessary
# RMAN> catalog start with '…';
# Find the latest scn that you have available:
# I.e.: SQL> SELECT MAX(next_change#) FROM v$archived_log WHERE resetlogs_change#
= (SELECT resetlogs_change# FROM v$database);
RMAN> restore database until scn 1234567;
RMAN> recover database until scn 1234567;
RMAN> alter database open resetlogs;
# (optional) Add TEMPFILES to tablespaces
# Create password file
```

Recovery from Offline Backup

```
Same as in the above example except for:

RMAN> recover database noredo;
```

How to Corrupt and Recover Database Block

```
# This exercise was created by a trained professional
# Please, DO try it (only) at home ;-)

# Here is how you can corrupt the block. Make sure db is DOWN, then (i.e.):
dd of=db_file_to_corrupt(N) if=another_file bs=8192 seek=15 count=1 conv=notrunc

# Start up db, then check for corrupted blocks
RMAN> backup validate datafile N;

# Find out if you have corrupted blocks – alert log OR:
SQL> SELECT * FROM v$database_block_corruption;

# Then recover the corruption
RMAN> blockrecover datafile X block Y;
or
RMAN> blockrecover corruption list; # Recover ALL known block corruptions
```

Backup and Recovery validation

```
# First of all, ALTER SYSTEM SET db_block_checksum = TRUE;

# Check block for physical and logical corruptions during backup
RMAN> backup check logical full database;
SQL> SELECT * FROM v$database_block_corruption;

# Do NOT backup. Just scan database files for corruptions
RMAN> backup validate check logical full database;
SQL> SELECT * FROM v$database_block_corruption;

# Do NOT restore. Just scan database backup for corruptions
RMAN> restore database validate;

# Validate existing backupset (you cannot validate datafile copy!)
RMAN> validate backupset 4;
```

Review Backups in Repository

```
RMAN> list backup of database; # Backupsets only
RMAN> list copy of database;   # Copies only
RMAN> list backup of archivelog all completed after "to_date('01/15/2009',
'MM/DD/YYYY')";
RMAN> list backup summary; # Short report, listing all backupsets
RMAN> list backup of database by file; # Backups sorted by database file
RMAN> list incarnation; # List of resetlog operations
```

Backup Advice

```
RMAN> report need backup database; # Based on RMAN retention policy
RMAN> report need backup days 1 database; # Need more than 1 day of arcs to apply
RMAN> report unrecoverable database; # NOLOGGING changes
```

Recovery Advise (11g)

```
RMAN> list failure;
RMAN> advise failure;
RMAN> repair failure [using advise option N];
```

Trial Restore and Recovery

```
RMAN> restore database preview;
RMAN> recover database test; # Data files need to be restored
RMAN> repair failure preview; # 11g
```

Manage Backup Repository

```
RMAN> report obsolete; # Find 'reclaimable' Backups
RMAN> delete [noprompt] obsolete; # Delete 'reclaimable' backups

# Catalog backups that CF does NOT know about (i.e. made after backup of CF)
RMAN> catalog start with '…';

# Check that backups registered in CF are still available on disk/tape
RMAN> crosscheck backup;
RMAN> crosscheck copy;
# Catch: TAPE backups will be only checked against 'tape' (i.e. Tivoli) database

# Unregister backups, copies or archived logs that are no longer available
RMAN> delete [noprompt] expired backup;
RMAN> delete [noprompt] expired copy;
RMAN> delete [noprompt] expired archivelog like '/ora04/oradbarc/%';
```

Managing Binary Metadata

Create Physical Standby Database on Separate System

```
# On primary database
# Make sure full backup exists
RMAN> backup current controlfile for standby;
RMAN> sql 'alter system archive log current';

# On standby database
# Create standby db directories, spfile and password file
# Copy or link primary database backups
# Start standby instance NOMOUNT

UNIX> rman target primary_db auxiliary standby_db
RMAN> duplicate target database for standby dorecover;
```

Transport Tablespace (online)

```
RMAN> transport tablespace tbs
   tablespace destination '/tbs_dest' # TTS export dump will be located here
   auxiliary destination '/aux_dest'  # Automatic auxiliary instance will be here
   until time 'sysdate-1';
```

Major Views

```
v$recovery_file_dest
v$flash_recovery_area_usage
v$rman_configuration
v$rman_status
v$rman_output
v$rman_job_details
```