

Advanced Backup & Recovery



By:
Francisco Munoz Alvarez 

Unconference

db tech showcase 2012

JPOUG

Japan Oracle User Group

About the Speaker



Francisco Munoz Alvarez 🏆



Oracle ACE Director

CLOUG (Chilean Oracle Users Group) President

LAOUC (Latin American Oracle Users Group Council) President

NZOUG (New Zealand Oracle Users Group) President

IOUC LA Spokesperson

8/9/10g/11g OCP, RAC OCE, AS OCA, E-Business OCP, SQL/PLSQL OCA, Oracle 7 OCM

Oracle 7, 11GR2 and OVM 3.1 Beta Tester

ITIL Certified

2010 ACE Director of the year by Oracle Magazine

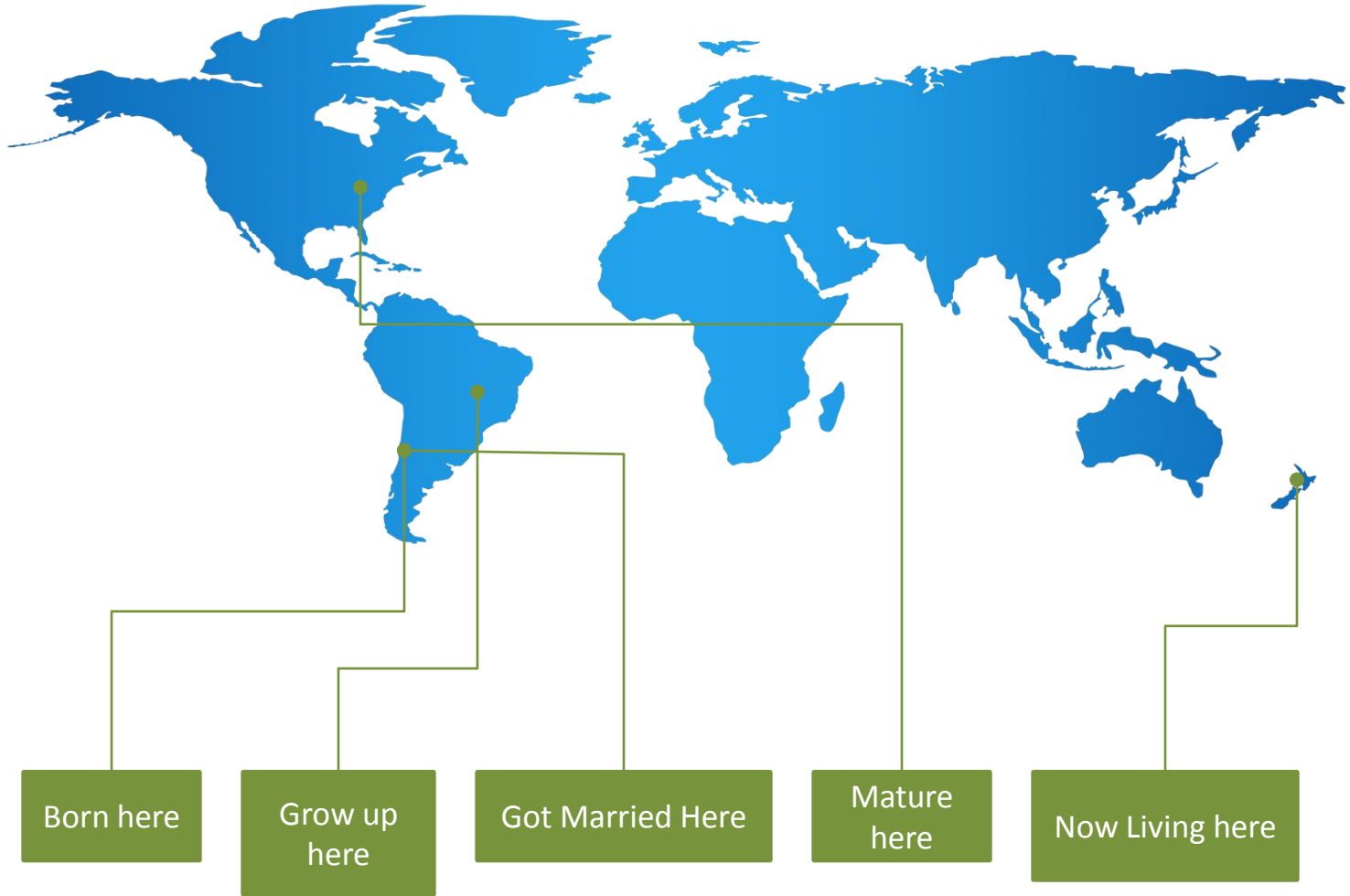
Blog: <http://oraclenz.wordpress.com> - Email: mbatec@hotmail.com - Twitter : [fcomunoz](#)

Revera: Oracle Professional Services Manager

Some Conferences



OK, are you from?



Advanced RMAN

In this session...

- Point-in-time recoveries
- RMAN and Flashback database
- RMAN and encryption
- Performance tuning RMAN operations
- Database duplication
 - Backup based database duplication
 - Active database duplication (11g)
- Tablespace point-in-time recovery
- Reporting from the recovery catalog

RMAN Point-in-Time Recovery

RMAN Point-in-Time Recovery

- You can recover your database to a specific:
 - Point-In-time
 - System Change Number
 - Log Sequence Number
- Useful for
 - Restoring after a major error.
 - Restoring development databases.
 - Restoring from a lost online redo log.

RMAN Recovery

Incomplete Recovery – About Resetlogs

- It can be confusing sometimes, having to figure out when to use resetlogs.
- In general, the database will always tell you when you must use resetlogs.
- All incomplete recoveries will require the use of the resetlogs command.
- If you restore a backup control file, you will need to use the resetlogs command.
- In Oracle9i you should immediately backup the database after a resetlogs operation.

RMAN Recovery

Incomplete Recovery

- RMAN supports incomplete recovery of your Oracle database.
- RMAN supports
 - Time based recovery
 - SCN based recovery
 - Log sequence based recovery

RMAN Recovery

Incomplete Recovery – Time based recovery

- Use the **restore** and **recover** commands using the until time parameter to do a time based restore.
- Example:

```
startup mount;  
restore database UNTIL TIME  
"TO_DATE('06/28/10 13:00:00','MM/DD/YY HH24:MI:SS')";  
recover database UNTIL TIME  
"TO_DATE('06/28/10 13:00:00','MM/DD/YY HH24:MI:SS')";  
alter database open resetlogs;
```

RMAN Recovery

Incomplete Recovery – Change based recovery

- Use the **restore** and **recover** commands using the until time parameter to do a restore based on a specific log sequence number.
- You can find log sequence numbers in the v\$log_history view.
- Example:

```
startup mount;  
restore database UNTIL sequence 504033;  
recover database UNTIL sequence 504033;  
alter database open resetlogs;
```

RMAN Recovery

Incomplete Recovery – SCN based recovery

- Use the **restore** and **recover** commands using the until time parameter to do a SCN based restore.
- You can find the current SCN in the v\$database column `current_scn`.
- Example:

```
startup mount;  
restore database UNTIL SCN 844039292;  
recover database UNTIL SCN 844039292;  
alter database open resetlogs;
```

RMAN Recovery

Incomplete Recovery – How Far Back Can I Go?

- What is required to complete a database point-in-time recovery?
 - Backup of database datafiles
 - All archived redo logs generated during backup until the desired point of recovery.

RMAN Recovery

Incomplete Recovery – How Far Back Can I Go?

- You can use the `list` command to determine how far back you can restore the database.
- First, set the `nls_date_format` to see the date/time

```
Set nls_date_format=mm/dd/yyyy hh24:mi:ss
```

Or

```
export NLS_DATE_FORMAT="DD-MON-YY HH24:MI:SS"
```


RMAN Recovery

Incomplete Recovery – How Far Back Can I Go?

- Start RMAN and use the `list backup of database summary` command to see how far back you can restore your database.

```
RMAN> list backup of database summary;
```

```
using target database control file instead of recovery catalog
```

```
List of Backups
```

```
=====
```

Key	TY	LV	S	Device	Type	Completion	Time	#Pieces	#Copies
80	B	F	A	DISK		11/05/2008	11:03:08	1	1
88	B	F	A	DISK		11/05/2008	13:36:38	1	1
92	B	F	A	DISK		11/05/2008	13:41:59	1	1

- Oldest backup

Column	Indicates
Key	A unique key identifying this backup set. If you are connected to a recovery catalog, then BS Key is the primary key of the backup set in the catalog. It corresponds to BS_KEY in the RC_BACKUP_SET view. If you are connected in NOCATALOG mode, then BS Key displays the RECID from V\$BACKUP_SET.
TY	The type of backup: backup set (B) or proxy copy (P).
LV	For incremental backups, the incremental backup level (0 or 1). For backup sets containing full backups of datafiles, F. For backup sets containing archived redo logs, A.
S	The status of the backup: A (available), U (unavailable), or X (all backup pieces in set expired). Refer to the CHANGE, CROSSCHECK, and DELETE commands for an explanation of each status.
Device Type	The type of device on which the backup was made, for example, DISK or sbt.
Completion Time	The date and time that the backup set completed. Note that the format of this field depends on the NLS_LANG and NLS_DATE_FORMAT environment settings.
#Pieces	The number of backup pieces in the backup set.
#Copies	The number of copies made of each backup piece in the set. The number is 1 if no duplexing was performed. Otherwise, the value ranges from 2 to 4.
Tag	The tag applied to the backup set; NULL if none. An asterisk (*) indicates multiple copies with different tags. Note that tag names are not case sensitive and display in all uppercase.

RMAN and Oracle Flashback Technologies

RMAN and Flashback Technologies

- RMAN supports the Flashback Database functionality with the flashback database command.

RMAN and Flashback Technologies

- Must have configured the database for flashback database operations
 - Flash recovery area configured
 - `db_recovery_file_dest`
 - `db_recovery_file_dest_size`
 - Set the flashback retention target (1440 default)
 - `db_flashback_retention_target`

RMAN and Flashback Technologies

- Flashback logging configured
 - To enable flashback you should perform a clean shutdown
`shutdown immediate;`
 - Mount the database
`Startup mount`
 - Enable Flashback
`alter database flashback on;`

RMAN and Flashback Technologies

A inconsistent shutdown can lead to the following error:

```
SQL> alter database flashback on;  
alter database flashback on  
*  
ERROR at line 1:  
ORA-38706: Cannot turn on FLASHBACK  
DATABASE logging.  
ORA-38714: Instance recovery required.
```

RMAN and Flashback Technologies

- Open the database

```
Alter database open;
```

- You can determine if the database is using flashback mode:
 - V\$DATABASE column FLASHBACK_ON

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

```
FLASHBACK_ON
```

```
-----
```

```
YES
```

OEM Supports execution of these tasks too.

RMAN and Flashback Technologies

- You can use the `v$flashback_database_stat` view to determine how far you can flashback your database.

```
select begin_time, flashback_data,  
db_data, redo_data, ESTIMATED_FLASHBACK_SIZE  
from v$flashback_database_stat;
```

- You can use the view `v$flashback_database_log` to determine how much space the recovery area requires for flashback logs:

```
select estimated_flashback_size  
from v$flashback_database_log;
```

RMAN and Flashback Technologies

- Using RMAN to flashback your database

```
RMAN> shutdown immediate
```

```
RMAN> startup mount
```

```
RMAN> FLASHBACK DATABASE  
TO TIME="TO_DATE('01/11/2009 12:40:00','MM/DD/YY HH:MI:SS')";
```

```
RMAN SQL 'alter database open resetlogs'
```

- Don't forget, you can open the database read only before you complete the flashback operation.
- Make sure you have flashed back to the correct point in time.

RMAN Performance Tuning

- First, we need to determine, is there a bottleneck?
- Oracle provides database views to help us discover if there are bottlenecks.
 - V\$RMAN_BACKUP_JOB_DETAILS
 - V\$SESSION
 - V\$PROCESS
 - V\$SESSION_LONGOPS
 - V\$BACKUP_ASYNC_IO
 - V\$BACKUP_SYNC_IO

RMAN Performance Tuning

- Where are the possible bottlenecks?
 - Read operations
 - Control file
 - Recovery catalog
 - Reading backup sets during recovery
 - Reading memory buffers
 - Write operations
 - Control file
 - Recovery catalog
 - Writing to tape/disk
 - Writing to memory buffers

RMAN Performance Tuning

- First... what is the history of our backups, are they getting slower, backing up more?
- Sometimes we just think things are getting worse.
- Use V\$RMAN_BACKUP_JOB_DETAILS view to find out for sure.
- Querying the view can be slow. (Bug 6868233) – Throw in a RULE hint to work around this bug.
- View not reset when database is shutdown.

RMAN Performance Tuning

```
select /*+ RULE */ session_key, session_recid,  
input_type, start_time, end_time, output_bytes,  
elapsed_seconds  
from v$rman_backup_job_details  
where start_time >= sysdate-180  
and status='COMPLETED'  
And input_type='DB FULL';
```

- This view has one little problem you need to be aware of.
 - Running multiple RMAN operations with the same session can cause the view to accumulate totals.
- Oracle also offers a like Recovery catalog view
 - RC_RMAN_BACKUP_JOB_DETAILS

RMAN Performance Tuning

```
select /*+ RULE */ session_key, session_recid,  
start_time, end_time, output_bytes,  
elapsed_seconds  
from v$rman_backup_job_details  
where start_time >= sysdate-180  
and status='COMPLETED'  
And input_type='DB FULL';
```

SESSION_KEY	SESSION_RECID	START_TIM	END_TIME	OUTPUT_BYTES	ELAPSED_SECONDS
456	456	11-JAN-09	11-JAN-09	228353024	397
461	461	12-JAN-09	12-JAN-09	229755904	422

- In this case the operation on the 12th took 25 seconds longer than that on the 11th. Probably not a big deal.

RMAN Performance Tuning

- I might be concerned if it looked like this:

SESSION_KEY	SESSION_RECID	START_TIM	END_TIME	OUTPUT_BYTES	ELAPSED_SECONDS
456	456	1-JAN-09	11-JAN-09	228353024	1397
456	456	2-JAN-09	11-JAN-09	328353024	2355
456	456	3-JAN-09	11-JAN-09	428353024	3390
456	456	4-JAN-09	11-JAN-09	458353024	4505
456	456	5-JAN-09	11-JAN-09	498353024	10044
456	456	6-JAN-09	11-JAN-09	528353024	11597
456	456	7-JAN-09	11-JAN-09	628353024	12497
456	456	11-JAN-09	11-JAN-09	728353024	14497
461	461	12-JAN-09	12-JAN-09	829755904	19422

RMAN Performance Tuning

- Let's look at our most current RMAN backup so we can find session details.

```
select sid, serial#, program
from v$session
where lower(program) like '%rman%';
```

SID	SERIAL#	PROGRAM
125	149	rman.exe
128	130	rman.exe
134	164	rman.exe

RMAN Performance Tuning

- A summary of the active session history might help (make sure you are licensed!)

```
Set lines 132
Column session_id format 999 heading "SESS|ID"
Column session_serial# format 999 heading "SESS|SER|#"
Column event format a40
Column total_waits format 9,999,999,999 heading "TOTAL|TIME|WAITED|MICRO"
Select session_id, session_serial#, Event, sum(time_waited) total_waits
From v$active_session_history
Where session_id||session_serial# in (125149, 128130, 134164)
And sample_time > sysdate - 1 And program like '%rman%'
And session_state='WAITING' And time_waited > 0
Group by session_id, session_serial#, Event
Order by session_id, session_serial#, total_waits desc;
```

RMAN Performance Tuning

- The resulting output

SESS	SESS		TOTAL
ID	SER	EVENT	TIME
	#		WAITED
			MICRO
125	149	control file single write	1,388,961
125	149	control file sequential read	45,964
125	149	control file parallel write	3,789
128	130	RMAN backup & recovery I/O	192,263,005
128	130	control file single write	1,095,253
128	130	control file parallel write	529,012

- Total waits of about 195 seconds. Runtime job details. It took 635 seconds to backup 1.3GB of data. Can we make this faster?

SESSION_KEY	SESSION_RECID	START_TIM	END_TIME	OUTPUT_BYTES	ELAPSED_SECONDS
468	468	11-JAN-09	11-JAN-09	1350017536	635

RMAN Performance Tuning

- Let's see what happens if we use compression.

```
RMAN> CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET;
```

- What do you think will be the result?

RMAN Performance Tuning

- Let's look at the RMAN backup using compression. Here are the session details.

```
select sid, serial#, program
from v$session
where lower(program) like '%rman%';
```

SID	SERIAL#	PROGRAM
128	102	rman.exe
130	116	rman.exe
134	129	rman.exe

RMAN Performance Tuning

- The resulting output from `v$active_session_history`:

SESS ID	SESS SER #	EVENT	TOTAL TIME WAITED MICRO
128	102	control file single write	2,548,592
128	102	control file sequential read	17,916
128	102	recovery area: computing backed up files	14,277
134	129	enq: TC - contention	9,653,878
134	129	RMAN backup & recovery I/O	5,414,782
134	129	control file single write	1,080,975
134	129	control file parallel write	1,014,960

- Total waits of ~20 seconds for a backup that took a total of 397 seconds. Pretty good improvement there.

SESSION_KEY	SESSION_RECID	START_TIME	END_TIME	OUTPUT_BYTES	ELAPSED_SECONDS
456	456	11-JAN-09	11-JAN-09	228353024	397

RMAN Performance Tuning

- How long will it take?
 - Sometimes the question is not how long, but how much longer.
 - You can use `v$session_longops` to get answers to such questions as seen here:
 - First, we have an rman backup running. Let's see which sessions it is connected to:

```
SQL> select sid, serial#, program
       2  from v$session
       3  where lower(program) like '%rman%';
```

SID	SERIAL#	PROGRAM
128	138	rman.exe
134	178	rman.exe
138	267	rman.exe

RMAN Performance Tuning

- How long will it take?
 - Now we query the v\$session_longops view to see how long that backup is expected to take:

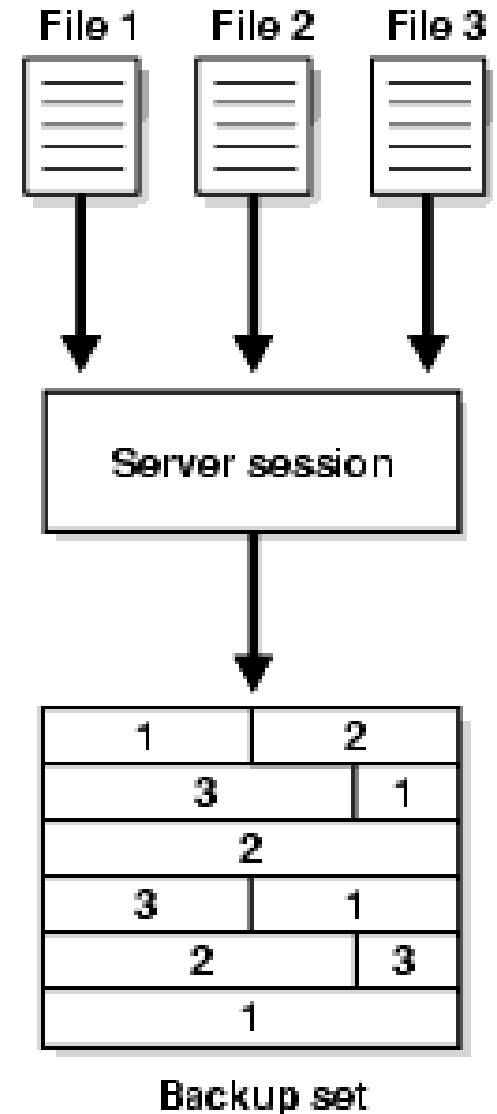
```
Select sid, serial#, opname, time_remaining
From v$session_longops
Where sid||serial# in (128138,134178,138267)
And time_remaining > 0;
```

SID	SERIAL#	OPNAME	TIME_REMAINING
128	138	RMAN: aggregate input	118
138	267	RMAN: full datafile backup	105

- The aggregate input line is the one we are interested in. It seems we have 118 seconds remaining for the running operation.

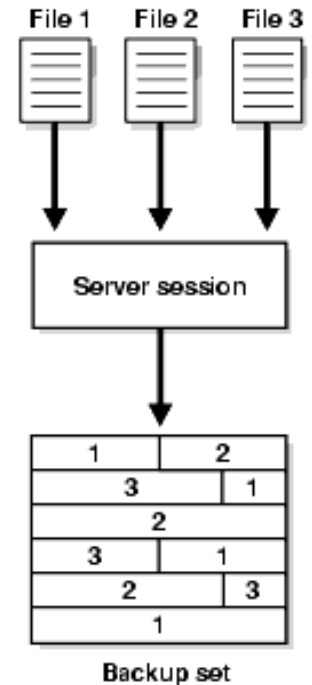
RMAN Performance Tuning

- Ideas to improve performance
 - Channels = available devices
 - Parallelize as best you can.
 - Don't over do it.
 - Datafile multiplexing
 - Allows parallelized datafile backups within a backup set/piece.
 - Impacts the size of the datafile read buffers.



RMAN Performance Tuning

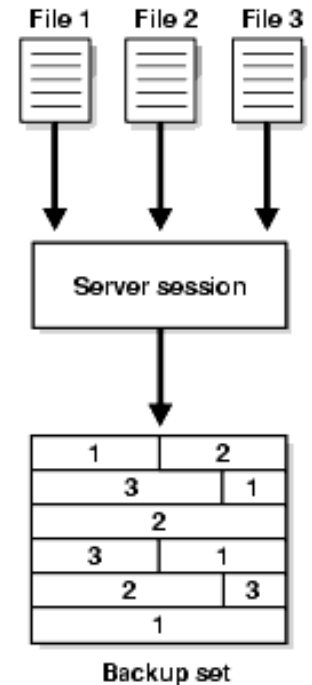
- The default settings for Multiplexing are typically ok.
- Settings that impact Multiplexing:



Setting	Default	Meaning
filesperset	64	How many datafiles can be put in each backup set.
maxopenfile	8	Defines how many datafiles RMAN can read at once.

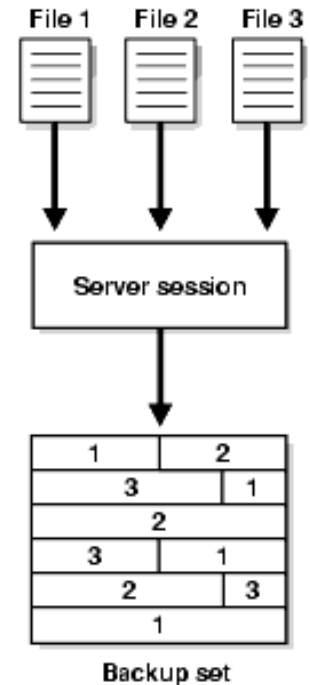
RMAN Performance Tuning

- Multiplexing is dependent on **filesperset** and **maxopenfiles** settings.
- Filesperset controls how many datafile can be in each backup set.
- RMAN does not multiplex a datafile when it is backed up using multi-section backups in 11g.



RMAN Performance Tuning

- The *Level of Multiplexing* is dependent on the setting of maxopenfiles.
- The level of multiplexing is the number of files that will be put in one backup set piece.
- As the level of multiplexing increases the buffer size for input files will decrease.



RMAN Performance Tuning

- Memory allocated to channels based on level of multiplexing:

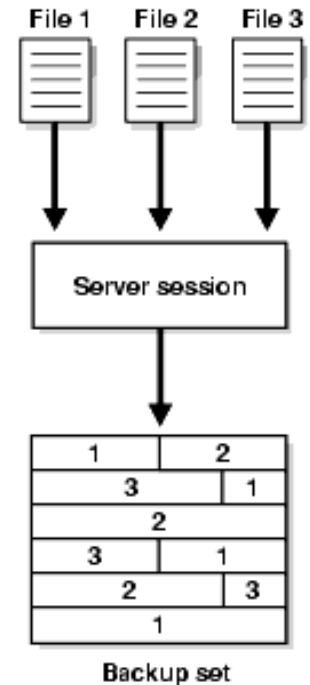
- 4 LOM – 16 buffers @ 1mb per buffer

- >4 but < 8 LOM – Variable # of channels of 512KB.

Total buffer size < 16MB.

- >8 LOB – 4 disk buffers of 128kb for each file.

Total buffer size per input file 512kb.



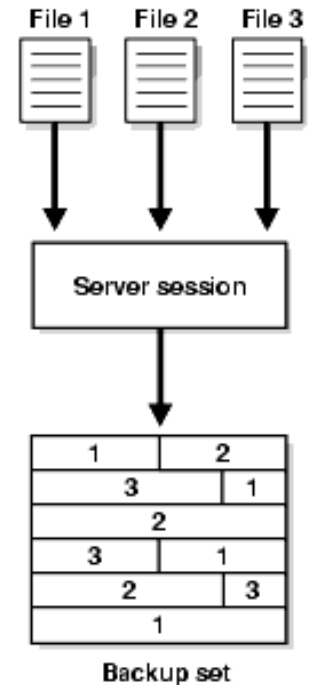
RMAN Performance Tuning

- Changing filesperset
 - **Backup** command

```
Backup database filesperset=4;
```

- Changing maxopenfiles
 - Allocate channel
 - Configure channel

```
configure channel 1 device type disk maxopenfiles 4;
```



RMAN Performance Tuning

- Here are the results of one test where we adjusted maxopenfiles to settings of 4, 8 and 16.
- Note in this test, the larger the setting, the slower the backup.
- Oracle suggests that higher settings may improve performance. Kind of the opposite of this test.
- Note that your results may vary for many reasons.

Maxopenfiles setting	Time to backup
4	6:35
8	6:46
16	7:45

- Your mileage may vary. Test before you invest.

RMAN Performance Tuning

- The Duration parameter (11g) can be used to minimize the load of the database backup over a specific time.
 - Use the duration command to indicate the maximum length of the backup (can also use the option partial).
 - Use the minimize load parameter to distribute the IO of the backup over the duration of the backup.
 - Careful with tape backups. This can impact the performance of streaming.
 - The minimize time parameter will cause the backup to utilize as much IO bandwidth as possible.

RMAN Performance Tuning

- Are you incorrectly using RATE?
 - Rate is designed to reduce performance, not improve it!
 - You can sometimes alter the size of the tape buffers by using the PARMs and BLKSIZE options of the configure or allocate commands..
- Large pool does not help unless you are facing synchronous IO issues.
 - Pretty rare these days not to have asynch IO.

RMAN Performance Tuning

- Incremental backups
 - Improve backup performance by instituting an incremental backup strategy.
 - Level 1 backups only write changed blocks.
 - Use a block change tracking file to further speed up the backups.
 - Higher levels of multiplexing with incremental backups might be helpful.
 - Smaller memory buffers get filled quicker
 - Written out to tape quicker.

RMAN Performance Tuning

- Is it a disk bottleneck or a tape bottleneck?
 - Backup validate may provide some insight.
 - No tape IO
 - Same disk IO.
 - If the backup validate is faster, then your problem is tape.
 - If the backup validate is the same, then the problem is disk IO.

RMAN Performance Tuning

- The V\$BACKUP_ASYNC_IO view can provide some insight into specific bottlenecks.
 - Provides throughput information during backup.
 - Total information maintained after backup.
 - Cleared after instance restart.
 - Maintains some historical record, but don't depend on it being there very long.

RMAN Performance Tuning

- The V\$BACKUP_ASYNC_IO view can provide some insight into specific bottlenecks.
 - Provides throughput information during backup.
 - Example:

```
select sid, serial, effective_bytes_per_second, filename
from V$BACKUP_ASYNC_IO;
```

SID	SERIAL	EFF BYTES PER SEC	FILENAME
134	16	86,878,436	
134	16		C:\ORACLE\ORADATA\ORCL\SYSAUX01.DBF
134	16		C:\ORACLE\ORADATA\ORCL\SYSTEM01.DBF
134	16	1,697,564	C:\ORACLE\ORADATA\ORCL\UNDOTBS01.DBF
134	16	1,485,938	C:\ORACLE\ORADATA\ORCL\ORCL_RETENTION_01.DBF
134	16	73,728	C:\ORACLE\ORADATA\ORCL\MY_DATA_01.DBF
134	16	73,728	C:\ORACLE\ORADATA\ORCL\TESTTBS.DBF
134	16	1,581,056	C:\ORACLE\ORADATA\ORCL\NEW_TBS_01.DBF
134	16	1,056,768	C:\ORACLE\ORADATA\ORCL\RMAN_DEMO_01.DBF
134	16	1,587,610	C:\ORACLE\ORADATA\ORCL\USERS01.DBF

RMAN Performance Tuning

- The V\$BACKUP_ASYNC_IO view can provide some insight into specific bottlenecks.
 - We can find the datafile that is causing us the biggest problem:
 - In this example, 2% of the IO's were involved in a long wait for this ASM tablespace :

```
SELECT LONG_WAITS/IO_COUNT, FILENAME
FROM V$BACKUP_ASYNC_IO
WHERE LONG_WAITS/IO_COUNT > 0
ORDER BY LONG_WAITS/IO_COUNT DESC;
```

```
LONG_WAITS/IO_COUNT FILENAME
```

```
-----
.2 +DGROUP1/orcl/datafile/asm_tbs.257.674829161
```

RMAN Performance Tuning

- Large file backups...
- You have a 20GB file.
- What's the fastest way to back that file up?
 - Multiplexing
 - Multi-threaded/process.
 - Many files written in parallel to a single backup set piece.
 - One file limited to one channel/one backup set (but many potential backup set pieces)

RMAN Performance Tuning

- Another option in 11g
- Multi-section backups (also called multisection backups and interfile backup parallelism in the Oracle docs).
 - New in Oracle Database 11g.
 - Spreads IO of datafiles across more than one channel in parallel.
 - Higher I/O
 - Higher CPU
 - Can reduce the time it takes to backup large datafiles.

RMAN Performance Tuning

- If all else fails
 - Create an RMAN trace file
 - Use the trace parameter at the RMAN command prompt.
 - Example

```
rman target=/ trace=c:\rman\trace.log debug
```

RMAN Performance Tuning

- If all else fails
 - Enable RMAN Debugging
 - Enabled at the command line.
 - Shows you in detail what calls RMAN is making.
 - Enabling debug at command line

```
rman target=/ debug
```

- Enabling debug from RMAN prompt

```
Rman>debug on
```

```
Rman>debug on
```

RMAN Performance Tuning

- If all else fails
 - Debug options
 - Io, sql, plsql, rcvman, rpc
 - Enable RMAN Debugging

RMAN Performance Tuning

- If all else fails
 - Enable SQL tracing for RMAN
 - At the end of the day, everything RMAN does happens in the database.
 - Wait events...etc...
 - Use the SQL command to enable event 10046 tracing for the RMAN session
 - Review the trace files for wait events.

RMAN Performance Tuning

Example

```
sql "alter session set events '10046 trace name  
context forever, level 12'";
```

```
RMAN> backup as compressed backupset database plus  
archivelog delete input;
```

- Carefully note the double single quotes.
- Unfortunately this does not trace everything...

RMAN Performance Tuning

Other solutions

- Trace recovery catalog database
- Logon trigger on backup database that checks for RMAN logon and enables tracing.

RMAN Performance Tuning

- Example logon trigger to trace all RMAN sessions.
 - Remember to grant alter session to the schema.

```
Create or replace trigger tr_rman_logon
After logon on database
Declare
    v_program    varchar2(1000);
Begin
    SELECT distinct program into v_program
    FROM v$session
    WHERE audsid = userenv('sessionid')
    and program like '%rman.exe%' and rownum < 2;
    if v_program='rman.exe'
    then
        execute immediate 'alter session set events ''10046 trace
name context forever, level 12''';
    end if;
Exception
    when NO_DATA_FOUND then
        NULL;
End;
/
```

RMAN Performance Tuning

- If all else fails
 - When we use a separate logon trigger we find that there are actually many RMAN related processes that we need to trace.

RMAN Performance Tuning

- Identify all the RMAN trace files being generated:

```
select a.sid, a.serial#, a.program, b.tracefile
from v$session a, v$process b
where a.paddr=b.addr
and a.program like '%rman%';
```

```

  SID      SERIAL#  PROGRAM
-----
TRACEFILE
-----
          136          453 rman.exe
c:\oracle\diag\rdbms\orcl\orcl\trace\orcl_ora_12724.trc

          137          500 rman.exe
c:\oracle\diag\rdbms\orcl\orcl\trace\orcl_ora_13928.trc

          170          304 rman.exe
c:\oracle\diag\rdbms\orcl\orcl\trace\orcl_ora_13076.trc
```

RMAN Performance Tuning

- Use tkprof to process the trace files...

```
tkprof orcl_ora_12724.trc orcl_ora_12724.trc.tkprof  
sort=prsela,exeela,fchela print=3 aggregate=no  
explain=rman/rman
```

```
tkprof orcl_ora_13928.trc orcl_ora_13928.trc.tkprof  
sort=prsela,exeela,fchela print=3 aggregate=no  
explain=rman/rman
```

```
tkprof orcl_ora_13076.trc orcl_ora_13076.trc.tkprof  
sort=prsela,exeela,fchela print=3 aggregate=no  
explain=rman/rman
```

RMAN Performance Tuning

- Review the TKPROF output to determine the main performance culprits.
 - Sort the output by elapsed time.
 - Some waits do not surface easily (RMAN backup & recovery I/O for example).
 - Typical problems
 - Control file access
 - Recovery catalog access
 - Backup device I/O waits

RMAN Performance Tuning

- Problems we have seen in the past...
 - Check Metalink for bugs
 - Bugs in the recovery catalog.
 - Indexes missing.
 - May need to split into two catalogs.
 - Control file issues
 - Lots of RMAN records can cause slowing response.
 - Conflicts with different RMAN operations on the same database running at the same time.
(Snapshot control file – Enqueues)

RMAN Performance Tuning

- Problems we have seen in the past...
 - MML layer settings need tweaked
 - Block size
 - Network frame size
 - Network throughput
 - Do you have enough devices that you are backing up too?
 - Scheduling? Are to many backups happening at the same time?
 - Backup during the day?
 - Are you validating to death?

RMAN Performance Tuning

- Summary of performance tuning tips
 - Check your network settings.
 - Check your MML vendor settings.
 - How many devices do you have.
 - Device contention – Scheduling.
 - RMAN bugs
 - Upgrade to 11g
 - Duration
 - Multi-section backups
- If backing up to tape, is the problem with your disks or your tape backup?
 - Use backup validate to determine.

RMAN Performance Tuning

- Summary of performance tuning tips
 - Review Metalink documents
 - 247611.1 – Known RMAN Performance Problems
 - 463227.1 – Oracle 10g RMAN Recovery catalog known performance issues.

RMAN Performance Tuning

- Summary of performance tuning tips
 - Commonly run into bugs..
 - [Bug 6476935](#) - RMAN Resyncs may take a long time to complete
 - Version affected : 10.2.0.3
 - Version fixed : 11.2
 - Affects : Catalog Schema
 - Workaround : create index rsr_i_stamp on rsr(rsr_sstamp, rsr_srecid);

RMAN Performance Tuning

- Summary of performance tuning tips
 - [Bug 5219484](#) - CATALOG RESYNCS ARE VERY SLOW - ROUT TABLE HAS 6 MILLION ROWS +
 - Version affected : 10.1
 - Version fixed : 10.2.0.4
 - Affects : Catalog Schema
 - Diagnosis : Many rows in ROUT-table.
 - Workaround : See [Note 378234.1](#) Rman Catalog Resync Operation is Very slow at 10G

RMAN Trouble Shooting

- So what happens if you get RMAN errors and you can't figure out what's up?
 - Metalink (bug lookup)
 - Use slaxdebug
 - Useful for syntax errors
 - Useful for strings that are too long for RMAN

RMAN Trouble Shooting

- Slaxdebug Example

```
Rman target=/ slaxdebug
RMAN> backup database;
debug: LOOKAHEAD token is "BACKUP_".
debug: SHIFT token "BACKUP_", enter state "5"
debug: (ACT STK) [5, SHFT]
debug: (PAR STK) [0] [5]
debug: LOOKAHEAD token is "DATABASE_".
debug: SHIFT token "DATABASE_", enter state
"154"
debug: (ACT STK) [154, SHFT]
debug: (PAR STK) [0] [5] [154]
```

RMAN Trouble Shooting

- Commonly used views
 - V\$RMAN_BACKUP_JOB_DETAILS
 - Information on RMAN backup jobs.
 - V\$BACKUP_ASYNC_IO
 - Information on ASYNC performance of RMAN operations
 - V\$BACKUP_SYNC_IO
 - Information on SYNC performance of RMAN operations

RMAN Trouble Shooting

- Commonly used views
 - V\$PROCESS
 - Information on processes in the database
 - V\$SESSION
 - Information on sessions in the database.
 - V\$SESSION_LONGOPS
 - Information on long running operations in the database.

RMAN Trouble Shooting

- Commonly used views
 - V\$RECOVERY_PROGRESS
 - Show progress of RMAN user managed recovery operations.
 - V\$SESSION_WAIT
 - Displays wait events for sessions, including RMAN sessions.

RMAN Trouble Shooting

- Commonly used views
 - V\$RMAN_OUTPUT
 - Provides the output from RMAN in the form of a table.
 - V\$RMAN_STATUS
 - Displays the status of finished and on-going jobs.

RMAN Database Duplication

- RMAN provides an easy (sorta) way of duplicating your database to another database.
 - Backup based duplication
 - Requires that a backup of the database being duplication be available.
 - Active database duplication (11g only)
 - Backup the target database online duplicating it to the auxiliary instance over the network. No backup required.

RMAN Database Duplication

- Duplicate from
 - Source host
 - Target database
- Duplicate to
 - Destination host
 - Auxiliary Instance

RMAN Database Duplication

- Basic steps to prepare to duplicate a database
 - Back up the target database (Backup based database duplication only)
 - Make backup images available to the destination host. (Backup based database duplication only)

RMAN Database Duplication

- Basic steps to prepare to duplicate a database
 - Deciding where to put the duplicate database related files.
 - The destination host may have different file systems.
 - The duplicate command can make many of the needed changes for you on the fly.
 - May be duplicating to the same host too.

RMAN Database Duplication

- Basic steps to prepare to duplicate a database
 - Preparing the auxiliary instance for the duplication.
 - Configure OS Specific settings
 - Configure the auxiliary instance parameters.
 - Typically it's a basic parameter file.
 - Use the target database SPFILE as the final copy.
 - Create the password file.
 - Configure networking
 - Startup nomount the auxiliary instance

RMAN Database Duplication

- Connecting to RMAN for duplication

```
Set oracle_sid=orcl
```

```
Rman target=/ auxiliary=sys/password@mydb
```

RMAN Database Duplication

- The duplicate database command
 - Used to execute the duplication
 - Many different options
 - RMAN (10g and later) will create the needed tempfiles for you.

RMAN Database Duplication

- The duplicate database command
 - Duplicate to a specific point in time.
 - Duplicate to the point of the duplicate command.
 - Duplicate to a restore point.
 - Active or backup based database duplication.

RMAN Database Duplication

- The duplicate database command
 - Examples – Backup based database duplication

```
Duplicate target database to neworcl  
nofilenamecheck spfile;
```

```
Duplicate target database to neworcl  
nofilenamecheck spfile skip tablespace users;
```

```
Duplicate target database to neworcl  
nofilenamecheck spfile to restore point 'Test';
```

```
Duplicate target database to neworcl  
nofilenamecheck spfile open restricted;
```


RMAN Database Duplication

- The duplicate database command
 - Active database duplication
 - Does not support **until** clause
 - Does not support **to restore point** clause.

RMAN Database Duplication

- The duplicate database command
 - Example - Active database duplication

```
duplicate target database to neworcl from active  
database nofilenamecheck  
spfile set control_files  
'c:\oracle\oradata\neworcl\control01.ctl',  
'c:\oracle\oradata\neworcl\control02.ctl'  
set db_file_name_convert  
'c:\oracle\oradata\orcl',  
'c:\oracle\oradata\neworcl'  
set log_file_name_convert  
'c:\oracle\oradata\orcl',  
'c:\oracle\oradata\neworcl';
```

RMAN Tablespace Point Time Recovery (TSPITR)

- TSPITR provides the ability to restore a specific tablespace to a specific point in time in the database.
- RMAN provides this functionality.
- Minimal configuration
 - Figure out where you want the temporary instance files to reside.
 - Figure out which tablespaces you want to recover.

RMAN Tablespace Point Time Recovery (TSPITR)

- Use the RMAN recover command to start the restore:

```
recover tablespace users
until time '10/06/2008:22:42:00'
auxiliary destination 'c:\oracle\auxiliary';
```

- Tablespaces recovered are left offline
 - You should back them up after the TSPITR.
 - Bring them online with the **alter tablespace** command using the online keyword.

RMAN Tablespace Point Time Recovery (TSPITR)

- You should make sure the tablespace is fully self-contained.
 - Tablespace in the transport set can not have external object references.
 - Tables in Tablespace A
 - Indexes in Tablespace B
 - Must move both tablespaces A and B.
- Use the TS_PITR_CHECK view to determine if you are moving self-contained tablespaces.

RMAN Tablespace Point Time Recovery (TSPITR)

- Example of checking the transport set

```
SELECT ts1_name, ts2_name, reason
FROM SYS.TS_PITR_CHECK
WHERE ( TS1_NAME IN ('USERS') AND TS2_NAME NOT IN ('USERS') )
OR     ( TS1_NAME NOT IN ('USERS') AND TS2_NAME IN ('USERS') );
```

```
TS1_NAME    TS2_NAME
-----
```

```
REASON
-----
```

```
USERS      INDEX_TBS
```

```
Tables and associated indexes not fully contained in the
recovery set
```

RMAN Tablespace Point Time Recovery (TSPITR)

- It is possible that you will lose objects in the tablespace that were created after the point in time to which you restore the tablespace.
- Determine which objects will be lost by querying the view `TS_PITR_OBJECTS_TO_BE_DROPPED`
- Export the objects before the TSPITR and then re-import them afterwards.

RMAN Tablespace Point Time Recovery (TSPITR)

- Example of looking for objects that will be dropped.

```
SQL> SELECT OWNER, NAME, TABLESPACE_NAME,  
2          TO_CHAR(CREATION_TIME, 'YYYY-MM-DD:HH24:MI:SS')  
3          FROM TS_PITR_OBJECTS_TO_BE_DROPPED  
4 WHERE TABLESPACE_NAME IN ('USERS')  
5 AND CREATION_TIME >  
6 TO_DATE('07-OCT-08:22:35:30', 'YY-MON-DD:HH24:MI:SS')  
7 ORDER BY TABLESPACE_NAME, CREATION_TIME;
```

OWNER	NAME
-----	-----
TABLESPACE_NAME	TO_CHAR(CREATION_TI
-----	-----
SCOTT	TESTTABLE
USERS	2008-10-07:19:34:39

RMAN Tablespace Point Time Recovery (TSPITR)

- There are a number of rules related to TSPITR. Some highlights:
 - The database must be in ARCHIVELOG mode.
 - You must have at least one backup take before the point in time to which you wish to restore the tablespace. All archived redo logs required after that backup must also be available.
 - If you rename a tablespace, you cannot perform TSPITR to any point in time before the rename operation.

RMAN Tablespace Point Time Recovery (TSPITR)

- The normal no no's continued...
 - No replicated master tables.
 - Complete tables must be transported (e.g. all partitions).
 - Tablespaces with tables that contain VARRAY or nested tables are not allowed.
 - Tablespaces with external tables are not allowed.
 - UNDO tablespaces are not allowed.
 - Any tablespace with SYS owned objects are not allowed.

RMAN Tablespace Point Time Recovery (TSPITR)

- After the recovery
 - Datafiles added to the tablespace on the target database after the point in time for the recovery will be restored but empty.
 - All backups associated with tablespaces in the transport set taken before the point in time that you restored the tablespaces to are no longer valid. **You should run a backup after the TSPITR.**

RMAN Tablespace Point Time Recovery (TSPITR)

- After the recovery
 - You will not be able to run another TSPITR on that tablespace to any time before the point to which you restored the tablespace.
 - Once a TSPITR is complete, you will not be able to use the control file to restore any part of the database to any point in time before the time that you restored the tablespaces to during the TSPITR.

Recovery Catalog Shenanigans

- About the recovery catalog views
- Reporting against the recovery catalog
- Recovery catalog performance issues

Recovery Catalog Views

- You may already be familiar with the RMAN related V\$ views.
 - Exposes RMAN related information contained in the control file.
 - Helpful in determining certain RMAN information.

Recovery Catalog Views

- You may already be familiar with the RMAN related V\$ views.
 - The **list** and **report** command often are sufficient, so the use of these views is probably rare.
- Examples:
 - V\$backupset
 - V\$backup_piece

Recovery Catalog Views

- The recovery catalog has a number of views that are roughly equivalent to the V\$ views.
- All recovery catalog views start with RC_.
- Contain records for **all** databases registered with the catalog.

Recovery Catalog Views

- You can correlate most RMAN control file views with recovery catalog views. The Oracle Database Backup and Recovery Reference provides a table that associates the V\$ views with the RC_ views.
- Example
 - V\$BACKUPSET – RC_BACKUPSET
 - V\$BACKUP_PIECE – RC_BACKUP_PIECE

Recovery Catalog Views

- Hmm... This seems like this could be useful...
 - Enterprise reporting on backup status
 - Tracking length and size of backups
 - Looking for possible contention issues
 - Databases not being backed up

Recovery Catalog Views

- Hmm... This seems like this could be useful...
 - Combined with a configuration management tool, the recovery catalog can be even more powerful!

Recovery Catalog Views

- Here is a script that displays databases that are registered in the catalog and the last date they were backed up.

```
Select a.db_key, a.dbid, a.name db_name,  
b.backup_type, b.incremental_level,  
b.completion_time, max(b.completion_time)  
over (partition by a.name, a.dbid)  
max_completion_time  
from rman.rc_database a, rman.rc_backup_set b  
where b.status = 'A'  
and b.incremental_level = 0  
and b.backup_type = 'D'  
and b.db_key = a.db_key;
```

RMAN Encryption (11g)

- RMAN Supports 3 modes of encrypting backups.
 - Transparent mode
 - Requires configuration of the Oracle encryption wallet.
 - Password mode
 - Requires that you define the password before you start the backup.
 - Dual mode
 - Uses both transparent mode and password mode encryption.

RMAN Encryption (11g)

- To use encryption, you must first enable it.
- This will use transparent encryption by default.

```
Configure encryption for database on;
```

- Example of enabling password encryption

```
Set encryption on identified by backup only;
```

RMAN has GUTS!

The RMAN Internal Packages

- At the end of the day, pretty much everything RMAN does boils down to the use of packages built into the database kernel.
- These packages are largely undocumented.
- The most common packages used are:
 - DBMS_BACKUP_RESTORE
 - DBMS_RCVMAN
 - DBMS_RCVCAT

RMAN has GUTS!

The RMAN Internal Packages

- Another example.
- <10g, how to unregister a database?
 - Use `dbms_rcvcat.unregisterdatabase`
 - Need the DBID, the DBKEY from the recovery catalog:

```
SELECT db_key, db_id  
FROM db WHERE db_id = 1231209694;
```

```
DB_KEY DB_ID  
-----  
1 1231209694
```


RMAN has GUTS!

The RMAN Internal Packages

- Then simply unregister the database:

```
begin
  dbms_rcvcat.unregisterdatabase(1 , 1237603294);
End;
/
```

RMAN has GUTS!

The RMAN Internal Packages

- We can see an example of the use of these packages in `rmanRestoreDatafiles.sql`.
 - This script is created when you use DBCA to create a database using a template with datafiles.
 - Typically located in
 - `ORACLE_BASE\cfgtools\dbca\<dbname>` (11g)
 - `ORACLE_HOME\admin\<dbname>` (10g)
 - Also found in the scripts directory when DBCA creates a script to create the database.
 - Let's look at an example of this script for a minute...

RMAN has GUTS!

The RMAN Internal Packages

- Recover.bsq is the heart of RMAN.
- All RMAN operations are scripted in recover.bsq in some way.
- Located in `$ORACLE_HOME/rdbms/admin`
- Let's look in recover.bsq real quick....

RMAN has GUTS!

The RMAN Internal Packages

- If you are having frustrating problems with RMAN the debug option may be of help.
- You can enable debug from
 - When starting RMAN from the OS prompt
 - RMAN Command line
 - When allocating a channel
 - When configuring a channel

RMAN has GUTS!

The RMAN Internal Packages

- You can enable debug in different modes
 - IO
 - SQL
 - PL/SQL
 - RCVMAN
 - RPC
 - ALL

- Example

```
Rman target=/ debug=io
```

```
Rman target=/ debug=all
```

Migration using RMAN

How to migrate a 32 bit DB to 64 bit

This procedure can be used for single instance or RAC (9i to 11g):

In the 32 bit source server :

Start making a full backup of the database including all archivelogs:

```
RMAN> run {  
allocate channel c1 type disk;  
allocate channel c2 type disk;  
backup database plus archivelog;  
backup current controlfile; }
```

In the target 64 bit server:

- 1) Copy backup files from source server
- 2) Create the pfile or copy the pfile from source server
- 3) Startup nomount the new instance.
- 4) Connect to rman and set the dbid:
\$ rman target / nocatalog RMAN> set dbid=<dbid from the 32 server>;

Migration using RMAN

How to migrate a 32 bit DB to 64 bit

5) Restore the controlfile:

```
RMAN> restore controlfile from '/disk1/backup_32/backup_piece_name';  
RMAN> alter database mount;  
RMAN> restore database;
```

6) Identify the archivelog till you want to do the recovery and check if you have all the backups you need to do it and run:

```
RMAN> run {  
set until sequence xxxx;  
recover database;  
}  
RMAN> exit
```

Migration using RMAN

How to migrate a 32 bit DB to 64 bit

7) Now Migrating to 64 bit:

```
SQL> recover database until cancel using backup controlfile;  
cancel  
SQL> alter database open resetlogs migrate;  
SQL> @$ORACLE_HOME/rdbms/admin/utlirp.sql  
SQL> @$ORACLE_HOME/rdbms/admin/utlirp.sql  
SQL> shutdown immediate;  
SQL> startup
```

Note: Also please take a look in other methods in the following Metalink Note:
Migration of Oracle Instances Across OS Platforms - Doc ID: Note:733205.1

What else is new on 11G

RMAN Enhancements

The RMAN in 11g includes several interesting new features, including the following:

- Active (network-based) database duplication
- Parallel backup and restore of large files (multisection backups)
- Archival backups (long term)
- Improved lock media recovery performance
- New persistent configuration Parameters
- Backup failover to non-flash recovery areas

Reading recommendations:

Oracle RMAN 11g – Backup and Recovery
By Robert G. Freeman and Matthew Hart

OCP Oracle Database 11g – New Features for Administrators
By Sam Alapati

Soon:

Pro Backup & Recovery – Apress
By Francisco Munoz Alvarez

Thank you !