

What was new in 11g for Backup and Recovery?

Contents

Introduction	3
RMAN New Features and Enhancements.....	3
Proactive Health Check.....	3
Database Recovery Advisor	6
Faster Backup Compression.....	11
Multisection Backup	11
Archived Log Deletion Policy.....	13
Long Term Backups	13
Undo Backup Optimization.....	14
SET NEWNAME (11gR2).....	14
Virtual Private Catalog	15
Import Catalog	15
Block Corruption	16
RMAN GUI Interface	16
Tablespace Point in Time Recovery (11gR2).....	17
To Destination Clause (11gR2).....	17
Database Duplication (11gR2)	18
Fast Incremental Backup on Standby	19

Read-Only Transported Tablespace Backup	19
Web-Service Backup (11gR2)	19
Flashback New Features	19
Flashback Transactional Backout	19
Flashback Data Archive	19
Data Pump New Features and Enhancements	20
Reuse Dumpfiles	20
Compression	20
Encryption	20
Remap Table	21
Data Transformation	21
Legacy Mode (11gR2)	21
Exploit Clustering (11gR2)	21
Disable Direct Path	21
Single Partitioning Transportable Tablespace	22
Dynamic Restart of Workers	22
Summary	22

Introduction

In this document, we will talk about some of the main new features and enhancements introduced with Oracle Database 11g in the backup and recovery area. In this version Oracle introduced a great number of new features and enhancements that makes the DBA life more easy and our data more recoverable.

Now, let's first take a look on some of the new features and enhancements on RMAN, than on Flashback and finally with Data Pump.

RMAN New Features and Enhancements

Proactive Health Check

RMAN has now the ability to make a proactive check to detect any corrupt blocks in the database. It can also be used to check corruption at database, tablespace, datafile or datafile block level. Also you can VALIDATE the following extra types of files:

- BACKUPSET
- CONTROLFILECOPY
- CURRENT CONTROLFILE
- DB_RECOVERY_FILE_DEST (Recovery Area)
- SPFILE

Here are some examples of the VALIDATE command:

```
RMAN> validate database;
RMAN> validate tablespace example;
RMAN> validate datafile 1;
RMAN> validate datafile 2 block 1;
```

Now, let's take a look in the output of the command VALIDATE DATABASE as an example:

```
RMAN> validate database;

starting validate at 03-SEP-12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=48 device type=DISK
channel ORA_DISK_1: starting validation of datafile
```

```

channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00001
name=+DATA/orcl/datafile/system.256.748895685
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.257.748895691
input datafile file number=00005
name=+DATA/orcl/datafile/example.269.748896743
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.258.748895693
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.748895693
channel ORA_DISK_1: validation complete, elapsed time: 00:00:55
List of Datafiles
=====

```

```

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
1    OK    0          14438          89604          1540174
   File Name: +DATA/orcl/datafile/system.256.748895685
   Block Type Blocks Failing Blocks Processed
   -----
   Data      0          59267
   Index     0          12506
   Other     0          3389

```

```

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
2    OK    0          19483          74284          1540172
   File Name: +DATA/orcl/datafile/sysaux.257.748895691
   Block Type Blocks Failing Blocks Processed
   -----
   Data      0          15318
   Index     0          11069
   Other     0          28370

```

```

File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
3    OK    0           678          11520          1540174
   File Name: +DATA/orcl/datafile/undotbs1.258.748895693
   Block Type Blocks Failing Blocks Processed
   -----
   Data      0           0
   Index     0           0

```

other 0 10842

```
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
4 OK 0 395 5440 1305916
File Name: +DATA/orcl/datafile/users.259.748895693
Block Type Blocks Failing Blocks Processed
-----
Data 0 4641
Index 0 41
Other 0 363
```

```
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
5 OK 0 14489 25605 1496304
File Name: +DATA/orcl/datafile/example.269.748896743
Block Type Blocks Failing Blocks Processed
-----
Data 0 6604
Index 0 1261
Other 0 3246
```

```
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
including current control file for validation
including current SPFILE in backup set
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
List of Control File and SPFILE
```

```
=====
File Type Status Blocks Failing Blocks Examined
-----
SPFILE OK 0 2
Control File OK 0 602
Finished validate at 03-SEP-12
```

RMAN>

As you can see above, the health check first reports all datafiles, than in the list of datafiles it will detail the total number of blocks, the number of empty blocks, and the number of corrupt blocks per datafile. Followed by a detail of the objects types inside of the datafiles (Data, Index and Other) and in the end it will check the status of the SPFILE and control file.

Database Recovery Advisor

The Data Recovery Advisor (DRA) is a great new feature introduced with Oracle 11g database health check framework and help us to diagnose any persistent data failure with the database, and what is better, it will also help us to fix the issue taking away some of the stress always associated with this type of operation.

In other words, the main benefits of DRA are:

- Simplifies the diagnosis, analysis, and recovery steps;
- Allows the DBA to easily detect and check failures that requires any media recovery;
- Reduce recovery time;
- Reduce human errors;
- Gives repair advice;
- Generate repair scripts;
- Recover using the generated script at your request;

Note: Data Recovery Advisor will only be useful if you are using RMAN for your backups.

DRA can be easily used via OEM or through the RMAN command line interface. At command line, the main commands are:

- **LIST FAILURE:** List all failures that the database encountered.
RMAN> LIST FAILURE [CRITICAL/HIGH/LOW/OPEN/CLOSED];
RMAN> LIST FAILURE <failure_id> DETAIL;
- **ADVISE FAILURE:** Analyzes all backups and provide a recovery script.
RMAN> ADVISE FAILURE;
- **CHANGE FAILURE:** Changes the priority or the status of the failure.
RMAN> CHANGE FAILURE <failure_id> PRIORITY [HIGH/LOW];
- **REPAIR FAILURE:** Repairs the failure with the automatically created script previously generated with the ADVISE FAILURE command)
RMAN> REPAIR FAILURE;
RMAN> REPAIR FAILURE <failure_id>;

To make it easier to understand, let's simulate the lost of a datafile (from the tablespace example) and we will see how the DRA will help us to overcome this situation.

First we will ensure that we have a current backup of our tablespace example:

```
RMAN> backup tablespace example plus archivelog;
```

Now we will need to delete the datafile associated with the tablespace example to simulate a failure, due that I'm using ASM, I will need to first place the tablespace example OFFLINE and then delete the datafile as follows:

```
SQL> alter tablespace example offline;
```

```
Tablespace altered.
```

```
SQL>
```

```
ASMCMD> rm EXAMPLE.269.748896743
```

Now, we can use the DRA to help us to recover the failure.

```
RMAN> list failure;
```

```
List of Database Failures
```

```
=====
```

Failure ID	Priority	Status	Time Detected	Summary
714	HIGH	OPEN	03-SEP-12	Tablespace 6: 'EXAMPLE' is offline
708	HIGH	OPEN	03-SEP-12	One or more non-system datafiles are missing
702	HIGH	OPEN	03-SEP-12	One or more non-system datafiles are offline

```
RMAN>
```

As you can see above, the LIST FAILURE command show us all failures detected by the DRA. We can see the detail of one of the failures, using the command LIST FAILURE <failure_id> DETAIL.

```
RMAN> list failure 708 detail;
```

```
List of Database Failures
```

```
=====
```

Failure ID	Priority	Status	Time Detected	Summary
------------	----------	--------	---------------	---------

```

-----
708          HIGH      OPEN      03-SEP-12      One or more non-system
datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 708
  Failure ID Priority Status    Time Detected Summary
-----
  711          HIGH      OPEN      03-SEP-12      Datafile 5:
'+DATA/orcl/datafile/example.269.748896743' is missing
  Impact: Some objects in tablespace EXAMPLE might be
unavailable

RMAN>

```

Now the next step is to ask the DRA for the advice to fix all current issues, for this we use the ADVISE FAILURE command.

```

RMAN> advise failure;

List of Database Failures
=====

Failure ID Priority Status    Time Detected Summary
-----
708          HIGH      OPEN      03-SEP-12      One or more non-system
datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 708
  Failure ID Priority Status    Time Detected Summary
-----
  711          HIGH      OPEN      03-SEP-12      Datafile 5:
'+DATA/orcl/datafile/example.269.748896743' is missing
  Impact: Some objects in tablespace EXAMPLE might be
unavailable

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

Mandatory Manual Actions
=====

```

no manual actions available

Optional Manual Actions

=====

1. If file +DATA/orcl/datafile/example.269.748896743 was unintentionally renamed or moved, restore it

Automated Repair Options

=====

Option Repair Description

1 Restore and recover datafile 5

 Strategy: The repair includes complete media recovery with no data loss

 Repair script:
 /u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_2736176867.hm

RMAN>

The ADVISE FAILURE command will automatically generate a script to repair our database. If you wish, you can use the command REPAIR FAILURE PREVIEW to see the details of the script generated before order the DRA to repair the database.

RMAN> repair failure preview;

Strategy: The repair includes complete media recovery with no data loss

Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_2736176867.hm

contents of repair script:

```
# restore and recover datafile
restore datafile 5;
recover datafile 5;
```

RMAN>

Now, we can order the DRA to repair the database using the REPAIR FAILURE command.

RMAN> repair failure;

Strategy: The repair includes complete media recovery with no data loss

Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_2736176867.hm

contents of repair script:
restore and recover datafile
restore datafile 5;
recover datafile 5;

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

YES

executing repair script

Starting restore at 03-SEP-12
using channel ORA_DISK_1

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

channel ORA_DISK_1: restoring datafile 00005 to
+DATA/orcl/datafile/example.269.748896743

channel ORA_DISK_1: reading from backup piece
/DB/u02/backups/bck_orcl_0vnkabjm_1_1

channel ORA_DISK_1: piece
handle=/DB/u02/backups/bck_orcl_0vnkabjm_1_1
tag=TAG20120903T231710

channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:25
Finished restore at 03-SEP-12

Starting recover at 03-SEP-12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 03-SEP-12
repair failure complete

RMAN>

Note: If you are not using the control file autobackup option when doing your backups and the recovery of the control file is required, the DRA will not be able to do a full automatic recovery process and a combination of automatic and manual recovery process will be necessary to overcome this situation.

Faster Backup Compression

Oracle Database 11g introduced a new compression algorithm called ZLIB in addition to the BZIP2 algorithm that was introduced with Oracle Database 10g. ZLIB is much faster and consume less CPU than BZIP2, but the down side of this new algorithm is that it does not compress as much when compared with BZIP2.

You can easily set the compression algorithm using the following commands:

```
RMAN> configure compression algorithm 'ZLIB';  
RMAN> configure compression algorithm 'BZIP2';
```

Note: You will need to license the "Advance Compression" option to be able to use the new compression algorithm ZLIB. The BZIP2 algorithm is the default compression algorithm in 11g.

Multisection Backup

Multisection (or Parallel backup) is a very attractive new backup command clause that allows large files (datafiles) to be broken into smaller pieces (sections) improving the backup performance and consequently reducing the time it takes to finish.

Note: You must set the initialization parameter compatibility to at least 11.0 to be able to perform multisection backups, due that is impossible to restore multisection backups on early releases than 11.0.

In a multisection backup, RMAN will create a backup set in which each backup piece will contain the blocks of one file section, this will divide the work overload among multiple channels, with each channel backing up a specific section in a file (using parallel mode), also if your backup fails in the middle of the operation, you will only need to back up the sections that weren't backed up prior to the backup failure. In others words, if you

have a datafile with a size of 20GB and you execute a multisection backup of this datafile with 4 channels and specify the section size of 5GB, RMAN will break the 20 GB file on 4 pieces of 5 GB each and will backup all 4 pieces in parallel using the 4 channels you opened.

Let's take a look in how it would happen in real life, the command to take a 10GB datafile, open 2 channels and create sections of 5GB will be:

```
RMAN> run {
2>     allocate channel ch1 type disk format
'/DB/u02/backups/bck_orcl_%U';
3>     allocate channel ch2 type disk format
'/DB/u02/backups2/bck_orcl_%U';
4>     backup section size 5G datafile 6;
5> }
```

Note: When making a backup using multisection, the backups show up as section as well, and you are limited to 256 sections per datafile.

And if we want to check if the multisection files were created correctly, we will do the following:

```
RMAN> list backup of datafile 6;
```

```
List of Backup Sets
```

```
=====
```

```
List of Datafiles in backup set 26
```

BP Key	Pc#	Status	Piece Name
31	1	AVAILABLE	/DB/u02/backups/bck_orcl_13nkbm89_1_1
32	2	AVAILABLE	/DB/u02/backups/bck_orcl_13nkbm89_1_2

```
RMAN>
```

Archived Log Deletion Policy

This new Oracle release has enhanced RMAN capability in the management of archived redo logs, improving the archive log deletion policy letting you configure a persistent policy regarding when the archived redo logs are eligible for deletion. This gives the RMAN more flexibility and protection to Data Guard environments.

To understand this improvement let's just compare the deletion policy command we had available for 10G and the new one at 11g.

With Oracle 10g:

```
CONFIGURE ARCHIVELOG DELETION POLICY { CLEAR | TO { APPLIED ON  
STANDBY | NONE } }
```

With Oracle 11g:

```
CONFIGURE ARCHIVELOG DELETION POLICY { CLEAR | TO { APPLIED ON  
[ALL] STANDBY |  
    BACKED UP integer TIMES TO DEVICE TYPE device | NONE | SHIPPED  
TO [ALL]  
    STANDBY} [ { APPLIED ON [ALL] STANDBY | BACKED UP integer  
TIMES TO DEVICE TYPE device | NONE | SHIPPED to [ALL] STANDBY } ]  
...}
```

Long Term Backups

You can now in 11g use the command clause `KEEP` to retain backups for a longer period of time than the one specified by the retention policy for backups. It is very useful to meet regulatory requirements or to restore a database for testing purposes. This type of backup contains all files required to restore and recover a database, in other words, creates a kind of consistent self-contained snapshot of the database.

Some examples of a long term backup are:

```
RMAN> BACKUP DATABASE KEEP UNTIL TIME "TO_DATE('31-DEC-2015','dd-  
mon-yyyy')";  
RMAN> BACKUP DATABASE KEEP FOREVER;
```

Note: You cannot use the option `KEEP` to override the retention policy for files stored in the flash recovery area. Also, the use of a recovery catalog is only required when using the `KEEP FOREVER` command.

Undo Backup Optimization

Before 11g all Undo information was included in the backups, with 11g you can make use of the newly introduced Undo Optimization feature, it will not include in the backups any Undo information regarding already committed transactions, consequently making our backups run faster.

Note: This option cannot be configured, or removed.

SET NEWNAME (11gR2)

The SET NEWNAME command has received some important enhancements. In 10g the SET NEWNAME command allowed us to change the default location of restored datafiles or tempfiles. Now 11gR2 enables you with an easier way to change the names of multiple files in the database and also provides a quick alternative to name each file individually (Except Temp files).

When using the new SET NEWNAME FOR DATABASE and SET NEWNAME FOR TABLESPACE commands, you must specify substitution variables within the filename to avoid any possible name collisions.

The substitution variables that can be used are:

- %b: The filename remains the same as the original (only the name, without including the path);
- %f: Specifies the absolute file number of the datafile for which the new name is generated;
- %I: Specifies the DBID;
- %N: Specifies the Tablespace name;
- %U: Specifies the following format: data-D-%d_id-%I_TS-%N_FNO-%f (The %d specifies the database name).

With 10g:

```
RMAN> SET NEWNAME FOR DATAFILE 1 TO
'/u01/app/oracle/oradata/system01.dbf';
RMAN> SET NEWNAME FOR TEMPFILE 1 TO
'/u01/app/oracle/oradata/temp1.dbf'
```

With 11gR2:

```
RMAN> SET NEWNAME FOR DATABASE TO '/u01/app/oracle/oradata/%b';
RMAN> SET NEWNAME FOR TABLESPACE example TO
'/u01/app/oracle/oradata/%U';
RMAN> SET NEWNAME FOR DATAFILE 1 TO '+DATA';
```

It's also important to understand that it can be used with DUPLICATE, RESTORE and SWITCH commands.

Note: The SET NEWNAME command support ASM disk groups.
--

Virtual Private Catalog

Before 11g, all DBAs using a recovery catalog used to connect to it using the same main catalog owner credentials. This causes some serious security risks due that if some DBA makes a misuse of this privilege, it will be almost impossible to detect who did it. Also when a company have a centralize catalog in place and not all DBAs manages all databases, it gives access to all registered databases to all DBAs.

With the introduction of Virtual Private Catalog (VPC), we can grant access to the RMAN catalog to some users so they can only access a limited set of databases that are registered within the catalog in use and our security problems are now solved. Now each DBA can have their own subset of the recovery catalog, having access to only the databases each one manages and accessing it using their normal database credentials

Import Catalog

RMAN has improved the manageability on catalogs allowing us now to move and merge schemas of different recovery catalogs of different databases into a centralized repository using the new command IMPORT CATALOG.

To merge two database catalogs data in one centralized repository, all you will need to do is to connect to the destination catalog database that will receive the new data and issue the IMPORT CATALOG command specifying the owner of the source catalog.

```
$ rman catalog=catalog_bck2/rmancatalog
RMAN> import catalog catalog_bck2@orcl;
You can easily use the same logic used to merge catalogs to move a
catalog to another newly created recovery catalog.
```

Note: Each target imported is automatically unregistered from the source catalog. The import command can also be limited to a subset of the catalog by specifying the DBID or DB_NAME of each target to be imported. Also the version of the source catalog schema should match the current version of the RMAN executable.

Block Corruption

At Oracle 10g the V\$DATABASE_BLOCK_CORRUPTION view was populated only when using the VALIDATE command with the CHECK LOGICAL parameter, in other words, if a datafile block is corrupted and you are receiving a block corruption error message in your screen like the ORA-01578 (ORACLE data block corrupted), it will not be recorded to the V\$DATABASE_BLOCK_CORRUPTION view until detected by the RMAN.

Starting from Oracle 11g when any process or a database utility, including RMAN and the Automatic Diagnostic Repository (ADR), encounters a data block corruption it will automatically update the V\$DATABASE_BLOCK_CORRUPTION view, helping this way to speed up the detection process.

Now for block recovery, if the flashback database is currently enabled, starting with 11g the RMAN by default will search the flashback logs (remember that they contain the past images of the block) for a good version of the corrupted block, then it will apply the archived redo logs to roll the block forward. The command RECOVER...BLOCK is used to execute a block recovery operation using the flashback logs.

Note: You can use the EXCLUDE FLASHBACK LOG command to restrict RMAN from searching the flashback logs.

Another new feature available with 11gR2 is the ability to automatically repair corrupted blocks in real time by fetching a good version of the same block from a physical standby database. It is achieved using the new DB_ULTRA_SAFE initialization parameter.

RMAN GUI Interface

RMAN is sometimes a little bit confusing and for the new generation of DBAs that loves to use GUI interfaces and uses Oracle Enterprise Manager (OEM) to do almost everything I have some good news, now you have a GUI interface for RMAN in the Oracle Enterprise Manager to make your life a lot easier.

Tablespace Point in Time Recovery (11gR2)

Oracle 10g introduced a great new feature called Tablespace Point in Time Recovery (TSPITR), which introduced the possibility to make a point in time recovery of a tablespace without the need to bring the whole database to a point in time. Now at 11gR2, Oracle enhanced TSPITR to be also able to recover from a case of a dropped tablespace. Also this new version introduced some other enhancements such as:

- Uses the new SET NEWNAME command, SET NEWNAME replaces CONFIGURE AUXNAME for recovery set datafiles.
- DBMS_TTS.TRANSPORT_SET_CHECK is a new procedure that is automatically performed when executing a TSPITR in the fully automated mode (DEFAULT) to determine if the tablespaces in the recovery set are self contained (if the view. TRANSPORT_SET_VIOLATIONS is empty no problems were found, if it has rows, RMAN will stop the TSPITR process and you will need to manually resolve all issues before start the TSPITR process again), also has the improved capability to check all relationships in the transport tablespaces checks. Prior to 11gR2 you would check all the relationships and if the tablespaces are self-contained using SYS.TS_PITR_CHECK.
- Faster TSPITR Performance when using image copies, it directs RMAN to use existing image copies of the recovery set and auxiliary set datafiles for the recovery process. If using it, RMAN will not need to restore the datafiles from the backup.

To Destination Clause (11gR2)

Starting with 11gR2, the backup command has a new clause that allows us to specify a disk location where the backup would be generated.

Note: This clause will create all files in an organized manner, inside newly created directories. First new directory automatically created will have the name of the instance, the second internal directory will use the word backup set, the next level will use the date of creation and finally the backup set generated will have a system generated tag.

```
RMAN> backup database to destination '/DB/u02/backups3';
```

```
starting backup at 06-SEP-12  
allocated channel: ORA_DISK_1
```

```

channel ORA_DISK_1: SID=49 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.256.748895685
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.257.748895691
input datafile file number=00005
name=+DATA/orcl/datafile/example.269.793064071
input datafile file number=00006
name=+DATA/orcl/datafile/example.271.793106443
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.258.748895693
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.748895693
channel ORA_DISK_1: starting piece 1 at 06-SEP-12
channel ORA_DISK_1: finished piece 1 at 06-SEP-12
piece
handle=/DB/u02/backups3/ORCL/backupset/2012_09_06/o1_mf_nnndf_TAG2
0120906T201547_84jpxo6m_.bkp tag=TAG20120906T201547 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:26
Finished backup at 06-SEP-12

```

```

Starting Control File and SPFILE Autobackup at 06-SEP-12
piece
handle=/DB/u02/backups/flash_recovery_area/ORCL/autobackup/2012_09
_06/o1_mf_s_793311434_84jq0f1n_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 06-SEP-12

```

```

RMAN>

```

```

$ ls
o1_mf_nnndf_TAG20120906T201547_84jpxo6m_.bkp
$ pwd
/DB/u02/backups3/ORCL/backupset/2012_09_06
$

```

Database Duplication (11gR2)

The already very useful RMAN feature that allow us to duplicate a database was enhanced at 11gR2 with two new functionalities that makes it way more attractive and useful, they are:

- Duplicate database from backup: Now you can duplicate a database without the need to connect to the target database and recovery catalog, all you will use is the backup copied to the auxiliary database;
- Network-Based Duplication or Active Database Duplication: Duplicate a target database to an auxiliary database without requiring a backup of the target database, making the duplication over the network.

Fast Incremental Backup on Standby

Block change tracking is now supported on physical standby databases, which means that fast incremental backups are now possible to be done on standby databases.

Read-Only Transported Tablespace Backup

Now 11g removes the restriction that transported tablespaces must be made on read-write before backup. You can now backup transported tablespaces in read-only or read-write.

Web-Service Backup (11gR2)

Now with many companies talking and using cloud computing, it was clear that RMAN should not be behind on this new technology and now with Oracle 11gR2 that allows RMAN to backup to Amazon S3 and other web-based storage services, using a specially created Media Management Library (MML).

Flashback New Features

Flashback Transactional Backout

The new Oracle Flashback Transactional Backout feature, allow you to rollback the transactions with all its dependent transactions.

Flashback Data Archive

The new feature called Oracle Flashback Data Archive, lets you to store all transactional changes of any table from its creation, it uses flashback query without relying on any transient storage like the undo, recording the changes in the Flashback Recovery Area (FRA).

Data Pump New Features and Enhancements

Reuse Dumpfiles

The new parameter `REUSE_DUMPFILE` allows us to tell the expdp to overwrite any created dumpfile in a specific folder.

```
REUSE_DUMPFILE={Y|N}
```

Compression

For a long time we had the issue with Data Pump that it was not able to compress the dumpfile while getting created. With Oracle 11g, Data Pump can now compress the dumpfiles while being created by using the parameter `COMPRESSION` in the expdp command line and what is better, it will automatically decompress the dumpfile during the Impdp process.

This new parameter has three options:

- `METADATA_ONLY`: Only the metadata is compressed;
- `DATA_ONLY`: Only the data is compressed;
- `ALL`: Both the metadata and data are compressed;
- `NONE`: No compression is performed (DEFAULT).

Note: The compression parameter is part of the Advance Compression Option (ACO) and needs to be licensed.

Encryption

The already existent encryption parameter has being enhanced allowing the dumpfile to be encrypted while being created, supplying also new encryption options and adding more flexibility and security. Now you are able to encrypt the dumpfile using:

- Transparent Data Encryption (TDE) that uses Oracle Wallet
- Use a password-based encryption;
- Or use both encryption methods: TDE and password-based.

Note: Encryption is part of the Advance Security Options (ASO) and needs to be licensed.

Remap Table

The newly created parameter `REMAP_TABLE` allows you to rename tables during an `impdp` operation.

Data Transformation

The powerful parameter `REMAP_DATA`, allows the Data Pump to use functions stored within the database to transform data during the `expdp` or `impdp`. A very specific use of this new parameter is to mask sensitive data when moving it outside a production database.

Legacy Mode (11gR2)

Data Pump on 11gR2 automatically detects any legacy parameter used with the old `exp` and `imp` utilities and convert them to the correspondent parameter on Data Pump. In other words, you can now use your old scripts you used with `exp/imp` and use them with Data Pump without the need to modify the parameters in the script.

Exploit Clustering (11gR2)

The new `CLUSTER` parameter allows Data Pump to make use of a set of instances or all instances of a database when using Real Applications Cluster (RAC) instead of use one instance for all slaves.

```
CLUSTER={Y|N} [SERVICE_NAME={service_name}]
```

Disable Direct Path

Data Pump uses by default direct path inserts to load the data, making the load process faster. You can now disable the default direct path mode when using `impdp` with the new parameter `DATA_OPTIONS=DISABLE_APPEND_HINT`. This is very useful if you want to load some data that will be subject of DML operations.

Single Partitioning Transportable Tablespace

You can now export or import one or more partitions of a table without having to move the entire table with Data Pump. When using impdp you can choose to:

- Load partitions as they are
- Merge the partitions into a single table
- Or promote each into a separate table

`PARTITION_OPTIONS={NONE | DEPARTITION | MERGE }`

Note: The `PARTITION_OPTIONS` parameter default is `DEPARTITION` when partition names are specified on the `TABLES` parameter and `TRANSPORTABLE=ALWAYS` is set during an import or export operation, otherwise the default will be `NONE`.

Dynamic Restart of Workers

If a Data Pump worker process hits a fatal error, such as a process being manually shut down, then the worker will be restarted a number of times in an attempt to continue processing. Direct Path and External Tables (11gR2)

Direct path and external tables are the two main data access methods provided by Oracle 11gR2. The direct path access method is the faster of the two, but does not support intra-partition parallelism, by the other side, the external tables access method does support this function, and therefore may be chosen to load or unload a very large table or partition. Data loading with the conventional access method is much slower than the direct path and external table's methods. So, the Data Pump uses this method only when it has no other choice.

Summary

In this document, we learned about a great number of new features and enhancements introduced with Oracle Database 11g for RMAN, Flashback Recovery and Data pump. These were very high level overviews of each feature, and you will be able to learn more about them and how to use reading my book: "12c Backup and Recovery Survival Guide" that soon will be published by Packt Publishing.