

# Généralités



ORACLE

## Généralités

- Outil de sauvegarde/restauration des bases Oracle
- Introduit avec Oracle8
- Langage de commande spécifique au produit
- Interface mode ligne (rman) ou mode graphique
- API d'un Media Management Vendor (MMV) nécessaire pour les sauvegardes sur bande
- Alternative aux sauvegardes "OS"

ORACLE

## Fonctionnalités

- Backups à froid, à chaud, en stratégie noarchivelog ou archivelog, full ou incrémentale
- Sauvegarde des archivelogs, fichier de contrôle
- Détection automatique et sauvegarde de tout nouveau fichier ajouté à la base
- Ne sauvegarde que les blocs utilisés
- Détection des corruptions
- Parallélisation des sauvegardes
- Pas de surproduction de logs si backup à chaud (pas de begin/end backup)

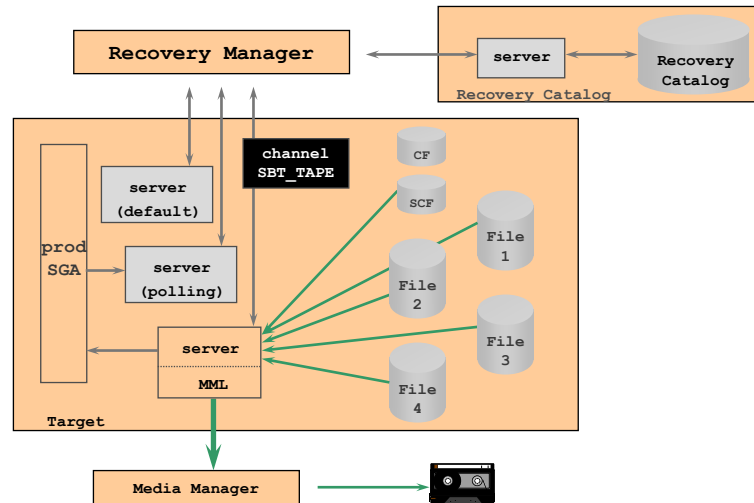
ORACLE

## Fonctionnalités (...)

- Restauration simplifiée
- Historisation des sauvegardes dans le Recovery Catalog
- Scripts de sauvegarde/restauration simples et indépendants de l'Operating System (OS)
- Stockage des scripts dans le RC

ORACLE

## Architecture



ORACLE

## Composants

- **RMAN**
  - Programme PRO\*C
  - Packages dbms\_rcvman et dbms\_rcvcat
  - Librairie recover.bsq
- **Recovery Catalog (RC)**
  - Référentiel géré par rman (ensemble de tables)
  - A stocker dans une autre base que la base cible
  - Toutes les informations contenues dans le catalogue proviennent du fichier de contrôle
  - Evolue avec les versions d'Oracle

ORACLE

## Composants ...

- **Recovery Catalog (...)**
  - N'est pas obligatoire mais conseillé
  - Mise à jour automatique à chaque sauvegarde
  - Nécessaire de le **re-synchroniser** avec le fichier de contrôle
- **Channels**
  - Les sauvegardes / restauration se font à travers des channels (shadow processes)
  - **Nombre de channels = degré de parallélisation**
  - De type «disk» ou de type «SBT\_TAPE» ou «sbt»
  - Le channel de type «SBT\_TAPE» ne peut être utilisé qu'avec un produit de MMV

ORACLE

## Exécution des commandes RMAN

- Individuellement

```
RMAN> backup database;
```

- Dans un RUN bloc

```
RMAN> run {  
  2> allocate channel c1 type disk;  
  3> backup database plus archivelog;  
  4> }
```

- Avec un script

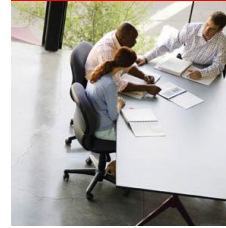
```
RMAN> @backup_full.rcv
```

- En ligne de commande

```
$ rman target=/ nocatalog cmdfile= backup_full.rcv
```

ORACLE

# Installation



ORACLE

## Installation: étapes

- Recovery Manager est installé automatiquement lors de l'installation du noyau (rdbms)
- *Créer (facultatif) une base pour le Recovery Catalog (RC)*
  - *Créer un utilisateur (propriétaire du RC) et lui assigner le rôle `RECOVERY_CATALOG_OWNER`*
  - *Créer le catalogue avec cet utilisateur (méthode différente selon la version d'Oracle)*
- Interfacer avec le produit tiers

ORACLE

## Installation: étapes

- Création du catalogue en version 8.1+
  - Exécuter la commande « create catalog » directement sous RMAN
  - Autres commandes disponibles: drop, upgrade

```
$ rman catalog=rman/rman
```

```
Recovery Manager: Release 11.1.0.6.0 - Production on Tue Jan 29 10:09:30 2008  
Copyright (c) 1982, 2007, Oracle. All rights reserved.  
connected to recovery catalog database
```

```
RMAN> create catalog;
```

```
recovery catalog created
```

ORACLE

## Install : Interfaçage avec une MML – 9i

- Il faut créer un lien \$OH/lib/libobk.so avec la librairie fournie par le produit tiers
- On peut aussi utiliser le paramètre SBT\_LIBRARY

```
run {  
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt  
  PARMS='SBT_LIBRARY=/usr/lib/libnwor.a.so',  
  'ENV=(NSR_SERVER=tape_svr,NSR_CLIENT=oracleclnt,  
  NSR_GROUP=oracle_tapes)';  
}
```

ORACLE

## Install : Connexion / Configuration

- Connexion (8.1+)

```
$ rman target=u1/p1[@target] catalog=u2/p2[@catalog]
```

- Fichier password nécessaire en cas de connexion distante sur la base target
- Enregistrement de la base « target »:  
RMAN> register database;
- Une fois la base enregistrée, il est possible d'exécuter manuellement un script de sauvegarde ou de piloter la sauvegarde à travers Database ou Grid Control

ORACLE

## Install : Connexion ... un exemple

```
$ rman target=/ catalog=rman/rman@rman

Recovery Manager: Release 11.1.0.6.0 - Production on Tue Jan 29 ...

Copyright (c) 1982, 2007, Oracle. All rights reserved.

connected to target database: P1 (DBID=2133968628)
connected to recovery catalog database

RMAN> register database;

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

RMAN>
```

ORACLE



# Configuration



ORACLE

## Catalogue ou fichier de contrôle

- Le catalogue historise les informations, ce que le fichier de contrôle ne peut faire que de façon limitée (`CONTROL_FILE_RECORD_KEEP_TIME`)
- Restrictions
  - Recovery si perte du controlfile (pré-9i)
  - Point-In Time Recovery (en cas de changement de structure de la base)
  - Commandes indisponibles (register database, reset database, ...)
  - Scripts stockés
  - Sauvegarder à partir de la base standby
- Sauvegarde supplémentaire à prendre en compte

ORACLE

## Snapshot Control File

- Pour effectuer une re-synchronisation du catalogue, RMAN utilise une copie du fichier de contrôle (*snapcf\_<sid>.f*)
  - Par défaut dans \$ORACLE\_HOME/dbs
  - Emplacement paramétrable avec la commande SET en 8.x et CONFIGURE à partir de la 9i :

```
set snapshot controlfile to '/oral/snacf_P1.f';
```

```
configure snapshot controlfile to '/oral/snapcf_P1.f';
```

ORACLE

## Paramètres persistants



- Définissables par la commande CONFIGURE
  - Parallélisme, device par défaut, nombre de copies, tablespace à exclure, ...
- Affichables par la commande SHOW
- Sauvegardés dans le fichier de contrôle et le catalogue (RESYNC CATALOG)

```
CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='ENV=(NSR_SERVER=bksvr1)';  
CONFIGURE CHANNEL DEVICE TYPE DISK RATE 5M FORMAT="/backup/%U";  
CONFIGURE CHANNEL DEVICE TYPE DISK clear;
```

ORACLE

## Paramètres persistants

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name P1 are:
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 DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BZIP2'; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/home/ora111/dbs/snapcf_P1.f'; # default
```

```
SQL> SELECT NAME, VALUE FROM V$RMAN_CONFIGURATION;
```

ORACLE

## Politique de rétention des backups

- Méthode **RECOVERY WINDOW**: détermine une fenêtre de temps pendant laquelle la recovery doit être possible.
- Méthode **REDUNDANCY**: fixe le nombre de backups qu'il faut conserver dans le catalogue

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 5 DAYS;
```

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 5;
```

```
CONFIGURE RETENTION POLICY TO NONE;
```

- La commande **DELETE OBSOLETE** supprime les backups inutiles au vu de la rétention utilisée

ORACLE

## Sauvegarde automatique du fichier de contrôle

- `CONFIGURE CONTROLFILE AUTOBACKUP` sauvegarde le fichier de contrôle et le spfile courant:
  - Après toute commande `RMAN BACKUP` ou `COPY`
  - Après tout changement structurel de la base
- Pour activer la sauvegarde:

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

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT  
2> FOR TYPE DISK TO '%F';
```

- Format (%F): `c-YYYYMMDD-QQ`
- Restaurable sans recovery catalog

ORACLE

## Configuration des channels

- Les caractéristiques des channels peuvent être pré-définies avec la commande `CONFIGURE`
- Allocation automatique pour les commandes `BACKUP`, `RESTORE`, `COPY`
- Sauf spécification explicite, RMAN utilisera les paramètres persistants
- Exemples

- Changement du device par défaut

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

- Niveau de parallélisme

```
RMAN> CONFIGURE DEVICE TYPE disk PARALLELISM 2;
```

ORACLE

## Autres commandes de configuration

- Exclusion d'un tablespace

```
CONFIGURE EXCLUDE FOR TABLESPACE <tablespace> [CLEAR];
```

```
RMAN> BACKUP NOEXCLUDE DATABASE;
```

ORACLE

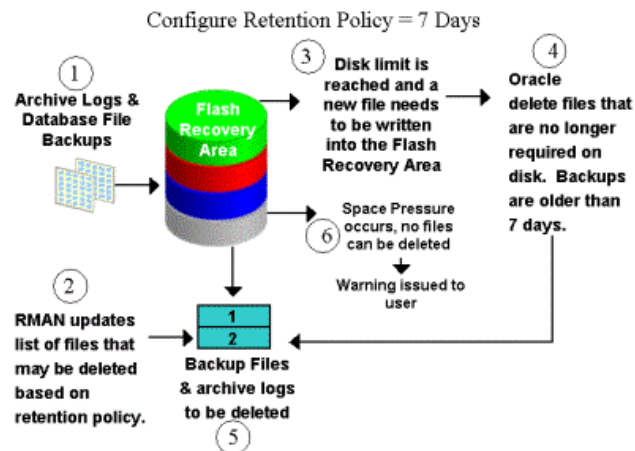
## Fast Recovery Area (FRA)



- Répertoire dédié aux sauvegardes/restaurations
- Tous les backups faits par RMAN (fichiers de données, archivlogs, fichier de contrôle, ...) sont redirigés dans ce répertoire
- Peut aussi contenir une copie du fichier de contrôle, un membre d'un groupe de redologs ou un flashback log
- A partir du moment où une FRA est initialisée, Oracle archive dans ce répertoire (quels que soient les paramètres LOG\_ARCHIVE\_DEST\_n)
- Les fichiers considérés obsolètes, au titre de la politique de rétention, peuvent être supprimés.

ORACLE

## Fast Recovery Area



ORACLE

## Fast Recovery Area

- File System ou ASM
- Les fichiers créés sont de type Oracle Managed File (OMF)
- Selon le taux de remplissage, Oracle émet des alertes (vue `DBA_OUTSTANDING_ALERTS` ou fichier `alert_SID.log`)

ORACLE

## Fast Recovery Area

- Paramètres (dynamiques)
  - `DB_RECOVERY_FILE_DEST` et
  - `DB_RECOVERY_FILE_DEST_SIZE`
- Commandes RMAN pour sauvegarder le contenu de la Fast Recovery Area:
  - `BACKUP RECOVERY AREA;`
  - `BACKUP RECOVERY FILES;`
- Vues
  - `V$RECOVERY_FILE_DEST`
  - `V$FLASH_RECOVERY_AREA_USAGE (R2)`

ORACLE

## Gestion des archivelogs

- Par défaut les archivelogs stockés dans la FRA peuvent être supprimés (en cas de besoin de place):
  - S'ils ont été sauvegardés au moins une fois sur disque ou sur bande
  - S'ils sont obsolètes au regard de la politique de rétention définie (type, durée, restore point, flashback db)
- Différentes possibilités
  - `CONFIGURE ARCHIVELOG DELETION POLICY`
    - `BACKED UP n TIMES TO DEVICE TYPE x`
    - `TO SHIPPED TO [ALL] STANDBY`
    - `TO APPLIED ON [ALL] STANDBY`

11.1  
11.1

} Data Guard

ORACLE

## Flashback Database

- Mode de fonctionnement de la base (doit être activé ainsi qu'une FRA pour stocker les flashback logs)
- Permet de revenir en arrière
- `DB_FLASHBACK_RETENTION_TARGET` permet de fixer la limite du retour en arrière (1440 min par défaut)
- Les flashback logs
  - ne sont pas sauvegardés
  - Peuvent être supprimés ou réutilisés en cas de problème de place

ORACLE

## Restore Points (10gR2)

- Utilisé dans un contexte de `RECOVER` ou de `FLASHBACK`
- `NORMAL` ou `GUARANTEE`
- Dans sa forme `NORMAL` c'est une association d'un libellé avec un SCN:

```
SQL> CREATE RESTORE POINT before_upgrade;
```

- `GUARANTEE` garantit la possibilité de faire un flashback database (même si le flashback logging n'a pas été activé)

```
SQL> CREATE RESTORE POINT myrp GUARANTEE FLASHBACK DATABASE;
```

- Vue `V$RESTORE_POINT`

ORACLE



## Block Compression

- Null block Compression
  - Ne sauvegarde pas les blocs qui n'ont jamais contenu de données
- Unused Block Compression
  - Ne sont sauvegardés que les blocs rattachés à un segment
  - Activé lors des sauvegardes sur disque ou sur bande avec Oracle Secure Backup (OSB)
  - Disponible à partir de la 10.2.0.2

ORACLE

## Compression binaire (10g)

- Backupset compressé (BZIP2). Se définit:
  - Au niveau de la configuration du device

```
CONFIGURE DEVICE TYPE disk BACKUP TYPE TO COMPRESSED  
BACKUPSET;
```

- Ponctuellement dans le script de backup

```
run {  
    allocate channel T1 type disk;  
    backup as compressed backupset tablespace users;  
}
```

ORACLE

## Compression binaire (11gR1)

- Nouvel algorithme: ZLIB
- Amélioration de 40% par rapport à la 10g
- La valeur du paramètre `COMPATIBLE` détermine les algorithmes de compression possibles
- On définit l'algorithme dans les paramètres persistants:

```
CONFIGURE COMPRESSION ALGORITHM '{ZLIB|BZIP2}';
```

- Syntaxe

```
RMAN> backup as compressed backupset database;
```

ORACLE

## Compression binaire (11gR2)

- En 11gR2 on ne parle plus d'algorithme mais de niveau de compression (4):
  - BASIC
  - LOW, MEDIUM, HIGH (nécessite l'option Advanced compression)
- Syntaxe

```
RMAN> backup as compressed backupset database;
```

```
SET COMPRESSION ALGORITHM 'LOW' optimize for load false;
```

```
CONFIGURE COMPRESSION ALGORITHM '<compression algorithm>'  
[AS OF RELEASE '<release>' ] [ OPTIMIZE FOR LOAD {TRUE|FALSE} ] ;
```

ORACLE

## Cryptage (10gR2)

- Trois modes: transparent, password ou dual
  - Transparent et dual nécessite l'utilisation d'Oracle Encryption Wallet
- Le niveau de cryptage (base ou tablespace) est défini avec la commande `CONFIGURE`

```
CONFIGURE ENCRYPTION FOR DATABASE ON;  
CONFIGURE ENCRYPTION FOR TABLESPACE tablespace_name ON;
```

- En mode password le mot de passe est fourni dans le script RMAN:

```
SET ENCRYPTION ON IDENTIFIED BY 'password' ONLY;  
SET DECRYPTION IDENTIFIED BY 'password' ;
```

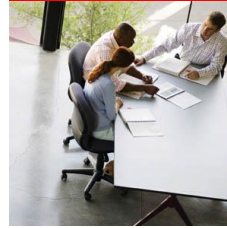
ORACLE

## Sauvegarde / Compression / Cryptage

	Backup with compression	Backup with encryption	Backup with compression+encryption
Data not encrypted	Data compressed for backup.	Data encrypted for backup.	Data compressed first, then encrypted, for backup.
Data encrypted by column (i.e. TDE)	Data compressed only. Encrypted columns gets no special treatment by the backup process, so does not compress very well.	Data encrypted only. Encrypted columns gets no special treatment by the backup process, so they are subjected to double-encryption.	Data compressed first then encrypted. Encrypted columns get no special treatment.
Data encrypted by whole block (i.e. the new 11g tablespace encryption)	Encrypted blocks are decrypted, then compressed, then re-encrypted. Non-encrypted blocks are compressed, and then are also encrypted during the backup, because after compression, it is impossible to distinguish the material that was originally encrypted from that which was not.	Encrypted blocks are passed through to the backup unchanged. They are neither encrypted during the backup and do not need to be decrypted during the restore. Non-encrypted blocks are encrypted for backup as usual.	Encrypted blocks are decrypted, then compressed, then re-encrypted. Non-encrypted blocks are compressed, and then are also encrypted during the backup, because after compression, it is impossible to distinguish the material that was originally encrypted from that which was not.

ORACLE

# Sauvegarde



ORACLE

## Stratégies de sauvegarde

- **Recovery Point Objective (RPO)**
  - Perte de données que l'on s'autorise
    - Données qui peuvent être re-saisies
    - Batches qui peuvent être re-soumis
- **Recovery Time Objective (RTO)**
  - Durée de la période d'indisponibilité
  - Engagement vis à vis des utilisateurs (Service Level Agreement - SLA)
- **Rétention des sauvegardes**
  - Jusqu'à quand veut on pouvoir remonter ?

ORACLE

## Sauvegarde

- Base montée ou ouverte selon le type de backup envisagé (à froid ou à chaud)
- En mode NOARCHIVELOG la base ne peut être que montée
- Portée: database, tablespace ou datafile
- Backup des archivlogs selon la stratégie
- Backup du fichier de contrôle:
  - Automatique
    - Si backup du tablespace system
    - Si « `CONTROLFILE AUTOBACKUP ON` »
  - Explicite

ORACLE

## Sauvegarde

- Le fichier spfile est sauvegardé explicitement ou automatiquement
- RMAN met à jour le catalogue avant exécution de toute commande comme `BACKUP`
- Ne prend pas en compte les blocs non utilisés et qui ne sont plus utilisés (10.2+)
- Pas de bascule des tablespaces en mode «hot backup»
- Le tablespace temporaire n'est jamais sauvegardé par RMAN. Il sera recrée après restauration (10.2+)
- La sauvegarde va automatiquement dans la FRA si celle ci est initialisée

ORACLE

## Type de sauvegarde

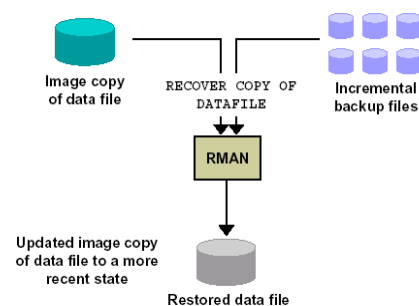
- **Image Copies**
  - Copie identique à l'original (telle que le ferait un outil OS)
  - Enregistrée dans le catalogue
  - Ne nécessite pas RMAN pour être restaurée
  - Uniquement sur disque
- **Backup sets**
  - Contient des données en provenance de plusieurs fichiers, d'archivelogs, de fichier de contrôle
  - Format propriétaire, donc RMAN nécessaire pour la restauration

ORACLE

## Incrementally Updated Backups



- Possibilité d'appliquer des sauvegardes incrémentales sur des copies (image copy)
- L'utilisation de ces copies ne nécessitera qu'un recover minimum
- Il n'est plus nécessaire, non plus, de refaire une copie après application d'une sauvegarde incrémentale.



ORACLE

## Incrementally Updated Backups

- Syntaxe

```
BACKUP AS COPY format '/tmp/%U' tablespace USERS;
```

```
BACKUP incremental level 1 for recover of copy with  
tag level_0 tablespace USERS;
```

```
RECOVER COPY OF tablespace USERS;
```

- Exemple

```
RECOVER COPY OF DATABASE WITH TAG 'incr_update';  
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY  
WITH TAG 'incr_update' DATABASE;
```

ORACLE

## Backup : exemple

- Script de sauvegarde (base complète sur disque)

```
run {  
    backup format '/bck/df_%d_%t_%s_%p' database;  
    backup format '/bck/al_%d_%t_%s_%p'  
        archivelog all delete input;  
    backup format '/bck/ctl_%d_%t_%s_%p' spfile  
        current controlfile;  
}
```

ORACLE

## Backup : exemple

- Script de sauvegarde (base complète sur bande)

```
run {
  allocate channel t1 type 'SBT_TAPE'
  parms = "ENV=(NB_ORA_CLASS=IT_ORACLE_HOURLY,
NB_ORA_SERV=sun01.prince.jci.com)";
  backup format 'df_%U' database;
  backup format 'al_%U'
    archivelog all delete input;
  backup format 'ctl_%U' current controlfile;
}
```

ORACLE

## Backup

- Tous les backups faits par RMAN (commandes backup, copy) sont enregistrés dans le RC
- Tout backup des archivelogs déclenche une re-synchronisation du catalogue
- Les sauvegardes classiques (begin/end backup) peuvent être cataloguées avec la commande CATALOG

```
RMAN> catalog datafilecopy '/tmp/users.dbf.sav';
```

```
RMAN> catalog archivelog '/tmp/p1_1_156.arc';
```

ORACLE

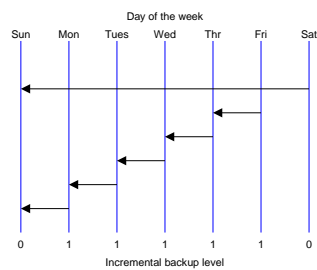


## Backup incrémental

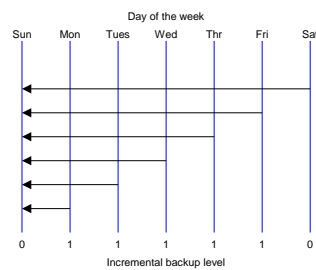
- Ne sauvegarde que les blocs qui ont été modifiés depuis le dernier incrémental
- Non-Cumulatif
  - Un backup incrémental niveau N (avec  $N > 0$ ) sauvegarde tous les blocs modifiés depuis le dernier incrémental de niveau  $\leq N$
- Cumulatif
  - Contient tous les blocs changés depuis le dernier backup de niveau  $< N$
  - Plus long et écrira plus de blocs qu'un backup non cumulatif mais moins de backup de même niveau à appliquer lors du restore.

ORACLE

## Backup incrémental



Incremental Differentiel



Incremental Cumulatif

ORACLE

## Fast Incremental backup



- Historisation des blocs modifiés
- Stockage dans un fichier (OMF par défaut) crée à cet effet
- Utilisation lors des backups incrémentaux
  - Accès direct aux blocs modifiés
  - Accélération des backups incrémentaux
- Activation

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING  
[USING FILE 'file' [REUSE]];
```

```
ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

ORACLE

## Fast Incremental backup



- Emplacement du fichier de « tracking »
  - dans le répertoire pointé par le paramètre `DB_CREATE_FILE_DEST`
  - ou précisé lors de la commande d'activation
  - La taille du fichier est calculée automatiquement en fonction de la taille de la base (10M par défaut avec des incréments de 10M)
- Vue `V$BLOCK_CHANGE_TRACKING`

ORACLE

## Détection des corruptions

- Automatisé lors de la sauvegarde
  - Par défaut aucune corruption n'est tolérée.
  - On peut relâcher cette contrainte avec le paramètre `MAXCORRUPT` (défini au niveau fichier)
  - Vues `V$BACKUP_CORRUPTION`, `V$COPY_CORRUPTION`

```
run {  
  allocate channel t1 type disk;  
  set maxcorrupt for datafile 3 to 2;  
  backup datafile 3;  
}
```

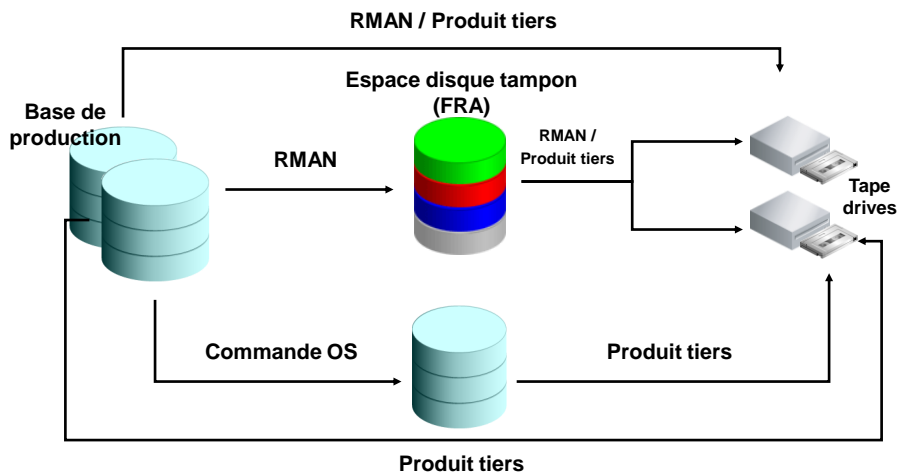
ORACLE

## Backup (divers)

- L'option `SKIP` de la commande `BACKUP` permet de ne pas sauvegarder les fichiers `OFFLINE`, `READONLY` ou `INACCESSIBLE`
- RMAN ne reconnaît pas un tablespace greffé tant qu'il n'a pas été basculé en mode Read Write
- RMAN sauvegardera d'abord à partir de la FRA si elle existe et ensuite les destinations d'archivage dans l'ordre croissant (`dest_1` à `dest_10`).

ORACLE

## Différents chemins de sauvegarde



ORACLE

## Restauration



ORACLE

## Restore / Recover

- Restore au sens reconstruction sur disque des fichiers (RMAN différencie la notion de **restore** de la notion de **recover**)
- Portée : Database, Tablespace, Datafile
- RMAN restaure **automatiquement** les archivelogs qui seront nécessaires à la Recovery.
- Toute ouverture en mode resetlogs devra être suivie d'un « reset database » pour créer une nouvelle *incarnation*

ORACLE

## Recover

- RMAN sélectionne le meilleur backup candidat
  - A partir des informations contenues dans le catalogue
  - Et de l'éventuelle restriction spécifiée par l'utilisateur (`SET UNTIL ...`)
- Tout recover déclenche une re-synchronisation du catalogue
- Toute restauration du fichier de contrôle ou incomplète impliquera la création d'une nouvelle incarnation (`RESET DATABASE`)

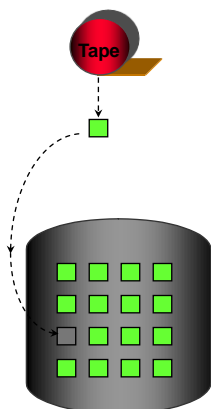
ORACLE

## Recover

- Une restauration / recover ne peut pas être une opération automatique
- Néanmoins avoir sous la main des modèles pour les différents cas de restauration

ORACLE

## Block Media Recovery (BMR)



- Le bloc devient le plus petit élément restaurable
- Avantages BMR
  - Diminue le temps de la recovery
  - Augmente la disponibilité pendant la phase de recovery
- N'est utilisable qu'à travers RMAN

```
RMAN> blockrecover datafile 7 block 3;
```

```
RMAN> blockrecover corruption list;
```

ORACLE

## Restore : exemple (1)

- Script de restauration d'un backup à froid (instance démarrée)

```
Run {  
  restore controlfile;  
  sql 'alter database mount';  
  restore database;  
  recover database noredo;  
  sql 'alter database open resetlogs';  
}
```

```
RMAN> reset database;
```

ORACLE

## Restore : exemple (2)

- Script de restauration et recover complet
- Fichier de contrôle et redologs online présents

```
Run {  
  restore database;  
  recover database;  
  sql 'alter database open;  
}
```

ORACLE

## Restore : exemple (3)

- Script de restauration d'un fichier de la base (base montée)
- Fichier de contrôle et redologs online présents

```
run {
  restore datafile {'filename'|filenumber};
  recover datafile {'filename'|filenumber};
  sql 'alter database open';
}
```

ORACLE

## Restore : exemple (4)

- Script de restauration incomplète d'une base (suite à un drop table par ex)
- Fichier de contrôle et redologs online présents

```
run {
  set until time '10/10/2007 13:45:00';
  restore database;
  recover database;
  sql 'alter database open resetlogs';
}
```

```
RMAN> reset database;
```

- Autres solutions: TSPITR ou Flashback database

ORACLE



## Restore : exemple (5)

- Script de restauration du controlfile et des fichiers de la base
- Les redologs online sont toujours présents

```
run {  
    restore controlfile;  
    sql 'alter database mount';  
    restore database;  
    recover database;  
    sql 'alter database open resetlogs';  
}
```

```
RMAN> reset database;
```

ORACLE

## Restore : exemple (6)

- Script de restauration d'un tablespace dans un autre emplacement

```
run {  
    sql 'alter tablespace USERS offline immediate';  
    set newname for datafile 12 to '/disk1/userV9.dbf';  
    restore tablespace USERS;  
    switch datafile 12;  
    recover tablespace USERS;  
    sql 'alter tablespace USERS online';  
}
```

ORACLE

## Restore preview

- Simule une restauration en indiquant les backup sets qui vont être sélectionnés.

```
RMAN> restore database preview;

List of Backup Sets
=====
BS Key   Size       Device Type Elapsed Time Completion Time
-----
10       43.70M     DISK        00:00:01     28-MAY-2009 20:25:03
BP Key: 10   Status: AVAILABLE Compressed: NO Tag: TAG20090528T202502
Piece Name: .../backupset/2009_05_28/ol_mf_annnn_TAG20090528T202502_51xwryhh_.bkp

List of Archived Logs in backup set 10
Thrd Seq   Low SCN   Low Time   Next SCN   Next Time
-----
1      402      8517872235198 27-MAY-2009 10:24:22 8517872275720 28-MAY-2009 20:24:59

Media recovery start SCN is 8517871218762
Recovery must be done beyond SCN 8517872274613 to clear datafile fuzziness

Finished restore at 29-MAY-2009 19:41:44
```

ORACLE

## Restore preview

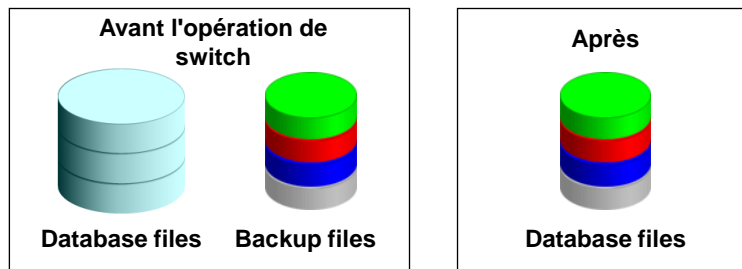
- L'argument SUMMARY limite les informations affichées

```
RESTORE DATABASE PREVIEW SUMMARY;
RESTORE DATAFILE 3 PREVIEW;
RESTORE ARCHIVELOG FROM LOGSEQ 200 PREVIEW;
RESTORE ARCHIVELOG FROM TIME 'SYSDATE-7' PREVIEW;
RESTORE ARCHIVELOG FROM SCN 234546 PREVIEW;
```

ORACLE

## Fast recovery

```
RMAN> SWITCH DATABASE TO COPY;
```



- Les backups doivent être des copies
- Pas de déplacement physique

ORACLE

## Restauration des archivelogs

- RMAN restaure automatiquement les archivelogs dont il a besoin pour la phase de recovery
- On peut néanmoins les restaurer manuellement

```
restore archivelog  
{all|like filename}<archivelog range>|... };  
  
run {  
  allocate channel t1 type disk;  
  set archivelog destination to '/tmp/restore_arch';  
  restore archivelog from logseq 1 until logseq 10;  
}
```

ORACLE

## Restauration des archive logs

- Les archive logs restaurés lors du recover sont conservés sur disque. Ils peuvent être supprimés:
  - Automatiquement avec l'option `DELETE ARCHIVELOG` de la commande `RECOVER`
  - Ou bien avec la commande `DELETE ARCHIVELOG`
- Sinon ils seront re-sauvegardés lors du prochain backup.
- L'option `MAXSIZE` permet de fixer la place utilisée par la restauration des archive logs

ORACLE

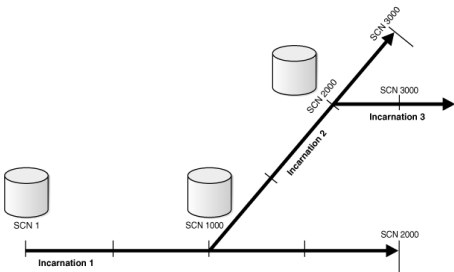
## Restauration (divers)

- Les tablespaces Read Only (RO) ne sont jamais restaurés par défaut.
- L'option `CHECK READONLY` contrôle le fichier (présent, accessible et checkpoint) avant de le restaurer. On peut forcer la restauration en ajoutant l'option `FORCE`.

ORACLE

# Incarnation & backup orphanelin

Current Incarnation	Usable Backups (Nonorphaned)	Orphaned Backups
Incarnation 1	All backups from incarnation 1	All backups from incarnations 2 and 3
Incarnation 2	<ul style="list-style-type: none"><li>• All backups from incarnation 1 prior to SCN 1000</li><li>• All backups from incarnation 2</li></ul>	<ul style="list-style-type: none"><li>• Backups from incarnation 1 after SCN 1000.</li><li>• All backups from incarnation 3</li></ul>
Incarnation 3	<ul style="list-style-type: none"><li>• All backups from incarnation 1 prior to SCN 1000</li><li>• All backups from incarnation 2 prior to SCN 2000</li><li>• All backups from incarnation 3</li></ul>	<ul style="list-style-type: none"><li>• All backups from incarnation 1 after SCN 1000</li><li>• All backups from incarnation 2 after SCN 2000</li></ul>



ORACLE

# Administration & Maintenance



ORACLE

## Interrogation du catalogue RMAN

- LIST: affiche le contenu du catalogue (database, tablespace, datafile, archivelog, ...)

```
RMAN> LIST BACKUP [SUMMARY | BY FILE];
```

```
RMAN> LIST BACKUP OF DATAFILE '/ora1/dbs/x1.dbf';
```

```
RMAN> LIST ARCHIVELOG ALL BACKED UP 2 TIMES TO DEVICE SBT;
```

```
RMAN> LIST INCARNATION;
```

```
RMAN> LIST COPY OF DATAFILE 1 COMPLETED BETWEEN <d1> and <d2>;
```

ORACLE

## Interrogation du catalogue RMAN

- REPORT: sélectionne les éléments en fonction des conditions exprimées
  - REPORT UNRECOVERABLE;
  - REPORT NEED BACKUP;
  - REPORT NEED BACKUP DAYS 5 DATABASE;
  - REPORT OBSOLETE;
  - REPORT SCHEMA AT SCN 45210;

ORACLE

## Interrogation du catalogue RMAN

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

```
...
BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
1        Full 154.48M    DISK        00:00:32      28-NOV-07
BP Key: 1  Status: AVAILABLE Compressed: NO  Tag: TAG20071128T111807
Piece Name: /home/ora102/P1/rman_dest/01j264jf_1_1

List of Datafiles in backup set 1
File LV Type Ckp SCN      Ckp Time  Name
-----
1        Full 185100    28-NOV-07 /home/ora102/P1/syst01.dbf
2        Full 185100    28-NOV-07 /home/ora102/P1/undotbs01.dbf
3        Full 185100    28-NOV-07 /home/ora102/P1/sysaux01.dbf
4        Full 185100    28-NOV-07 /home/ora102/P1/users01.dbf
```

```
RMAN> report unrecoverable;
```

```
RMAN-03022: compiling command: report
Report of files that need backup due to unrecoverable operations
File Type of Backup Required Name
-----
1        full          F:\ORACLE\ORADATA\FOO\SYSTEM01.DBF
```

ORACLE

## Synchronisation du catalogue

- La **resynchronisation** du RC est faite automatiquement avant et après un backup
  - Partielle: ne met pas à jour les modifications du schéma physique (fichier, tablespace, thread, ...)
  - Complète
- Certaines opérations ne sont pas propagées:
  - switch / archivage de redolog
  - modification de la structure de la base (par ex. ajout de fichier)
- Resynchronisation manuelle (complète): RESYNC CATALOG

ORACLE

## Maintenance du catalogue

- Gérer le statut des objets, synchroniser le catalogue de RMAN avec celui du Media Manager
- Les objets du catalogue peuvent avoir le statut `AVAILABLE`, `EXPIRED`, `DELETED` ou `UNAVAILABLE`
- Exemples d'opération
  - Changer le statut d'un objet
    - `UNAVAILABLE` suite à une sortie de bande
    - `EXPIRED` après un `CROSSCHECK`
  - Supprimer les backups obsolètes
  - Mettre à jour le catalogue suite à une suppression externe d'un archivelog

ORACLE

## Maintenance du catalogue

- Trois commandes principales
  - `CROSSCHECK` pour contrôler le statut d'un backup ou d'une copie
  - `DELETE` pour supprimer des objets du catalogue
  - `CHANGE` pour modifier le statut des objets dans le catalogue
- Channel de maintenance nécessaire quand la destination ne correspond pas à la configuration par défaut
- La syntaxe de ces commandes varie selon les versions de RMAN

ORACLE



## Commande CROSSCHECK

- `CROSSCHECK` synchronise les objets du catalogue avec les fichiers correspondants (disque ou bande)
- La commande `CROSSCHECK` n'est pas destructive
- Elle met à jour le statut de l'objet (`AVAILABLE` ou `EXPIRED`)

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';
```

```
RMAN> crosscheck backup;
```

```
RMAN> crosscheck backup of datafile '/t1/userP1.dbf'  
      2 completed after 'sysdate-180';
```

```
RMAN> crosscheck archivelog all;
```

ORACLE

## Commande DELETE

- Supprime l'objet spécifié, expiré ou obsolète
- Selon le mode (catalogue on non):
  - Supprime l'objet et l'enregistrement dans le Recovery Catalog
  - met à jour le statut de l'objet à `DELETED` dans le fichier de contrôle
- Suite logique d'une commande `CROSSCHECK`

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';
```

```
RMAN> delete expired backup;
```

```
RMAN> delete noprompt archivelog until sequence = 300;
```

```
RMAN> delete obsolete;
```

ORACLE

## Commande CHANGE

- Changer le statut des objets dans le catalogue
  - Disponibilité: AVAILABLE <-> UNAVAILABLE
  - UNCATALOG pour les objets supprimés sans passer par RMAN
  - Changement d'une durée de conservation
- RMAN n'utilisera pas un objet avec le statut UNAVAILABLE lors d'un restore ou un recover

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';
```

```
RMAN> change backupset 100 unavailable;
```

```
RMAN> change datafilecopy '/tmp/dem01.dbf' uncatalog;
```

ORACLE

## Maintenance du catalogue

- Exemples

```
DELETE NOPROMPT COPY BACKED UP 2 TIMES TO DEVICE TYPE sbt;
```

```
DELETE NOPROMPT OBSOLETE RECOVERY WINDOW OF 7 DAYS;
```

```
DELETE NOPROMPT EXPIRED BACKUP OF TABLESPACE users  
DEVICE TYPE sbt COMPLETED BEFORE 'SYSDATE-31';
```

```
CROSSCHECK BACKUP DEVICE TYPE DISK  
COMPLETED BETWEEN '01-JAN-01' AND '01-SEP-01';
```

```
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7'  
BACKED UP 2 TIMES TO DEVICE TYPE DISK;
```

ORACLE

## Maintenance du catalogue

- Supprimer une base du catalogue (10.1+)

```
RMAN> DROP DATABASE [INCLUDING BACKUPS] [NOPROMPT]
```

- La base doit être montée en mode RESTRICTED
- Cette commande:
  - Purge le catalogue (ou le fichier de contrôle) de tous les enregistrements relatifs à la base
  - Supprime les backups (fichiers et archivelogs)
  - Arrête la base et supprime tous les fichiers (datafiles, redologs et controlfiles)

ORACLE

## Maintenance du catalogue

- Dé-enregistrer une base du catalogue

```
RMAN> set DBID <database_dbid>;  
RMAN> UNREGISTER DATABASE <DBNAME> [NOPROMPT]
```

ORACLE

## Scripts stockés

- Peuvent être soit créés sur l'OS et exécutés dans un shell, soit stockés dans le RC puis exécutés comme suit :

```
RMAN> create script backup_full {backup (database);}
```

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

```
$ rman target=/ catalog=u/p@cat SCRIPT 'backup_full'
```

- Local (ne peut être exécuté que dans le contexte de sa base target) vs Global (quelque soit la base)
- Commandes associées: create, replace, print, delete, list, ...

ORACLE

## Sauvegarde/Restauration du catalogue

- Sauvegarde
  - Classiquement avec un backup à froid ou un export quotidien.
  - En utilisant le controlfile de la base catalogue (configurer AUTOBACKUP à ON)
  - Répliquer le catalogue (resync)
- Restauration
  - Avec les backups faits précédemment
  - En recréant et resynchronisant le catalogue avec le fichier de contrôle de la base target.

ORACLE

## Mise à jour du catalogue

- Après un changement de version
- Après l'application d'un patchset

```
RMAN> upgrade catalog;  
recovery catalog owner is rman  
enter UPGRADE CATALOG command again to confirm catalog  
upgrade  
RMAN> upgrade catalog;
```

- Déterminer la version du catalogue courant

```
SQL> select * from RCVER;
```

ORACLE

## Analyse d'un problème RMAN

- Tout problème RMAN est précédé d'une erreur RMAN-00569
- Les erreurs peuvent être liées à RMAN (prefixe RMAN), Oracle (ORA-) ou au media manager (Additional information: 7xxx)
- Les messages se lisent de bas en haut
- Fichiers: alert.log, trc, sbtio.log, log du media manager
- Plus d'infos: Chapitre 12 de la doc 10g (Recovery Manager Troubleshooting)

ORACLE

## Problèmes avec le Media Manager

- L'erreur ORA-27211 indique qu'Oracle ne peut pas charger la librairie libobk.so (ou orasbt.dll sur NT)
- Les erreurs ORA-19511 ou ORA-70xxx indiquent en général un problème de configuration
- En 9i on peut simuler un backup sur bande avec la librairie SBT de test:

```
RMAN> allocate channel c1 type 'sbt_tape'  
'ENV=SBT_LIBRARY=oracle.disksbt,  
(BACKUP_DIR=/your/backup/directory) ';
```

ORACLE

## Debugging

- Option debug de la commande rman

```
$ rman target=xx rcvcat=yy debug trace=/tmp/rman.debug
```

- Option debug dans un script rman

```
run {  
    debug on;  
    ...  
    debug off;  
}
```

- Option trace de la commande allocate (SBT call)

```
allocate channel t1 type 'sbt_tape' trace=1;
```

ORACLE

## Compatibilité

- Quatre composants entrent en ligne de compte:
  - L'exécutable RMAN
  - La base recovery catalog
  - La version du catalogue dans cette base
  - La base target
- Règles générales
  - La version du catalogue doit être  $\geq$  à la version de l'exécutable RMAN
  - Les versions de RMAN et de la base Target doivent être les mêmes

ORACLE

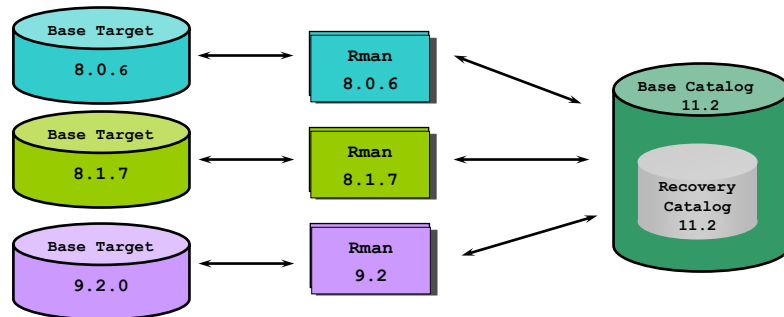
## Compatibilité

- Les erreurs suivantes indiquent en général un problème de compatibilité:
  - RMAN-554: initialization of internal recovery manager package failed
  - RMAN-6186 - "PL/SQL package %s.%s version %s in %s database is too old"
  - RMAN-6429 - "%s database is not compatible with this version of RMAN".

ORACLE

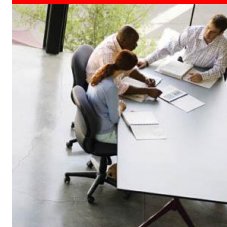
## Compatibilité

- A partir de la 8.0.6, une seule base 10.2 avec un recovery catalog 10.2 peut servir pour toutes les sauvegardes



ORACLE

## Autres Fonctionnalités

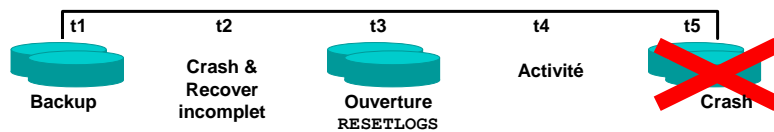


ORACLE



## Recovery through resetlogs

- Il n'est plus nécessaire de faire un backup après une ouverture en mode resetlogs
- Nouveau masque (%t\_%s\_%r.dbf) par défaut pour les archivelogs
- Le %r identifie l'incarnation et permet de différencier les archivelogs
- Nouvelle vue `V$DATABASE_INCARNATION`



ORACLE

## Tablespace Transportable

- Oracle 8i+
- Les restrictions sont levées au fur et à mesure des versions (taille de bloc, plate-forme, characterset)
- Le pré-requis est le passage préalable du tablespace en mode read-only
- A partir de la 11g on peut créer un tablespace transportable (et faire du TSPITR) à partir des sauvegardes

ORACLE

## Tablespace transportable (standard)

1. Mettre le tablespace read-only

```
ALTER TABLESPACE sales READ ONLY;
```

2. Exporter les méta-données à partir de la source

```
$ exp FILE=s980501.dmp  
TRANSPORT_TABLESPACE=y  
TABLESPACES=sales  
TRIGGERS=N CONSTRAINTS=N
```

3. Transfert des fichiers vers le système cible

```
FTP sales01.dbf, sales02.dbf, s980501.dmp
```

4. Import des méta-données

```
imp FILE=s980501.dmp  
TRANSPORT_TABLESPACE=y  
DATAFILES= (/disk1/sales01.dbf)
```

5. Passer le tablespace Read-Write

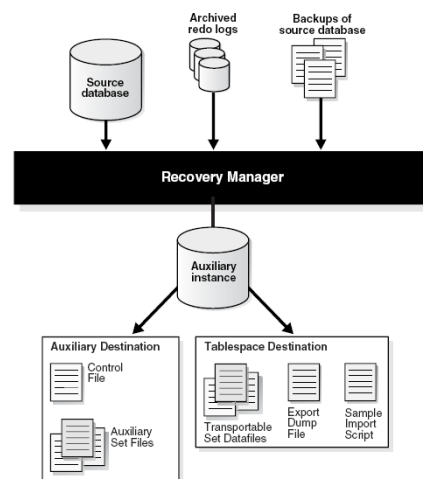
```
ALTER TABLESPACE sales READ WRITE;
```

ORACLE

## Tablespace transportable (RMAN)

- Générer un tablespace transportable à partir d'un backup (évite d'avoir à mettre le tablespace en RO)

```
RUN  
{  
  SET AUXILIARY INSTANCE PARAMETER FILE TO  
  '/tmp/auxinst.ora';  
  TRANSPORT TABLESPACE data01  
  TABLESPACE DESTINATION '/disk1/transportdest'  
  AUXILIARY DESTINATION '/disk1/auxdest'  
  UNTIL SCN 123456;  
}
```



ORACLE

## TableSpace Point In Time Recovery

- **Automatic Instance Creation for RMAN TSPITR:** la phase de préparation de l'instance AUXILIAIRE n'est plus nécessaire. Il suffit d'exécuter un RECOVER avec une clause UNTIL

```
run {  
  SET NEWNAME FOR DATAFILE 5 to '/tmp/t01.dbf';  
  RECOVER TABLESPACE t1 UNTIL LOGSEQ 10 thread 1  
  AUXILIARY DESTINATION '/tmp/backup';  
}
```

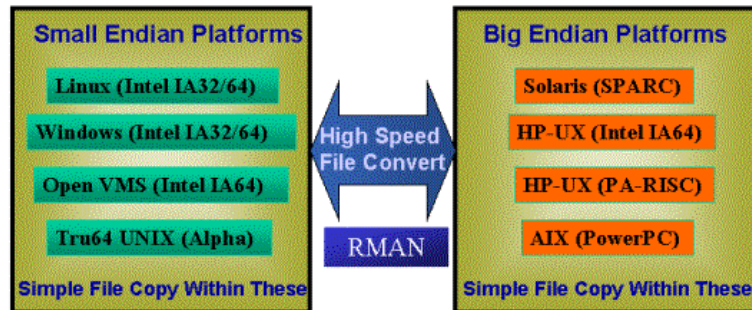
ORACLE

## Cross Platform Tablespace Conversion

- Un tablespace peut être transporté entre plate-forme hétérogène: par exemple de Sun Solaris vers Linux Intel
- En une ou deux étapes selon « l'endianness »
  - Idem versions précédentes si les deux plate-formes sont identiques en terme « d'endianness »
  - Étape supplémentaire de conversion avec RMAN si big endian vers little endian ou vice versa
- Utilisation d'exp/imp ou du nouvel outil (Data Pump) pour l'export et l'import des méta-données

ORACLE

## Cross Platform Tablespace Conversion



- `SELECT PLATFORM_NAME, ENDIAN_FORMAT FROM V$TRANSPORTABLE_PLATFORM;`

ORACLE

## Cross Platform Tablespace Conversion

- Même character set
- Le paramètre `COMPATIBLE` doit être :
  - 8.1+ si le transport se fait entre plate-formes de même type
  - 9.0+ si la taille de bloc est différente
  - 10.0+ si le type est différent
- Attention aux fichiers avec le statut offline ou read-only: il faut les passer au moins une fois RW ou online pour utiliser cette fonctionnalité.

ORACLE

## Durée et impact d'un backup

- Durée
  - Temps disponible pour faire le backup
  - Tout ou rien sauf si on spécifie l'option PARTIAL (les backupsets terminés sont sauvegardés)

```
BACKUP TABLESPACE users DURATION 2:00;  
BACKUP DATABASE DURATION 5:00 PARTIAL FILESPERSET 1;
```

- Impact
  - Maximiser la performance ou l'impact
  - En combinaison avec DURATION

```
BACKUP DATABASE NOT BACKED UP SINCE 'sysdate - 1'  
PARTIAL DURATION 7:00 MINIMIZE TIME;
```

ORACLE

## Automatic datafile creation

- Il n'est plus nécessaire de pré-créeer un fichier qui aurait été oublié lors de la sauvegarde.
- Fichier de contrôle à jour:
  - RESTORE TABLESPACE DATA1;
  - RECOVER TABLESPACE DATA1;
- Ancien fichier de contrôle:

```
run {  
    restore controlfile;  
    sql 'alter database mount';  
    recover database;  
}
```

ORACLE

## Autres fonctionnalités

- **Backup piece failover**
  - Si le backup sélectionné n'est pas utilisable, RMAN est capable de chercher une autre copie ou bien un backup plus ancien.
- **Channel failover**
  - Si un des channels démarré pour la sauvegarde plante, RMAN essaye d'utiliser les autres channels pour terminer le backup
- Enregistrer dans le catalogue des backuppieces qui ont été fait manuellement ou déplacés.
- Proxy Copy élargi aux fichiers archivelogs
- Enhanced Reporting: RESTORE PREVIEW

ORACLE

## Autres fonctionnalités

- Convertir une base vers une plate-forme de même « endianness »

```
CONVERT DATABASE NEW DATABASE 'TOM'  
transport script '/tmp/TOM/tom_script'  
to platform 'Solaris[tm] OE (64-bit)'  
db_file_name_convert '/tmp/TIM' '/tmp/TOM';
```

ORACLE

## Autres fonctionnalités

- Les fichiers du tablespace temporaire sont recréés lors d'un recover.
- La commande `BACKUP` peut produire soit des backupset soit des copies
- On peut fixer le type de backup par défaut avec la commande `CONFIGURE`
- On peut aussi copier une copie (sur disque ou sur bande)
- Cataloguer le contenu d'un répertoire

```
CATALOG START WITH '/d1/arch'
```

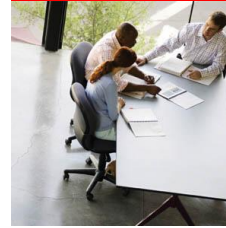
ORACLE

## Nouvelles vues

- `V$RMAN_BACKUP_JOB_DETAILS`
- `V$RMAN_BACKUP_SUBJOB_DETAILS`
- `V$RMAN_BACKUP_TYPE`
- `V$RMAN_ENCRYPTION_ALGORITHMS`
- `V$RMAN_OUTPUT` (jobs RMAN en cours)
- `V$RMAN_STATUS` (jobs en cours et passés)
- `V$BACKUP_FILES`

ORACLE

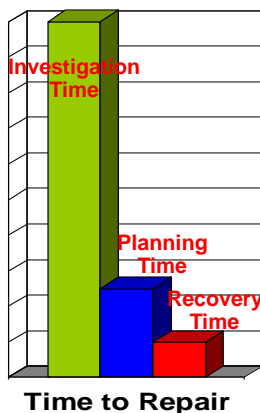
# Nouveautés 11g



ORACLE<sup>®</sup> 11g  
DATABASE

ORACLE

## Data Recovery Advisor



- Data Recovery Advisor (DRA) fait partie de la nouvelle infrastructure "Fault Diagnosability" introduite en 11g
- DRA
  - Détecte, réactivement ou manuellement, les défaillances
  - Regroupe et traite les problèmes similaires
  - Présente les différentes options possibles (sauvegarde, standby, ...) classées par durée et perte de données
  - Exécute l'option choisie par l'utilisateur

ORACLE



## Mode ligne de commande

- Liste des problèmes identifiés

```
RMAN> list failure;
```

- Solution préconisée

```
RMAN> advise failure;
```

- Application de la solution préconisée

```
RMAN> repair failure [preview];
```

ORACLE

## List Failure

```
RMAN> list failure;
```

```
List of Database Failures
```

```
=====
```

```
Failure ID Priority Status      Time Detected Summary
```

```
-----
```

```
3           HIGH      OPEN      05-MAR-07  datafile 4 contains one or more corrupt blocks
```

ORACLE

## Advise Failure

```
RMAN> advise failure 3;
```

### List of Database Failures

=====

Failure ID	Priority	Status	Time Detected	Summary
3	HIGH	OPEN	05-MAR-07	datafile 4 contains one or more corrupt blocks

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

### Automated Repair Options

=====

Option	Strategy	Repair Description
1	no data loss	Perform block media recovery of block 28 in file 4.

Repair script: /home/ora111/diag/rdbms/v111/V111/hm/reco\_3063620186.hm

ORACLE

## Repair Failure

```
RMAN> @/home/ora111/diag/rdbms/v111/V111/hm/reco_3063620186.hm
```

```
RMAN> # block media recovery
```

```
2> recover datafile 4 block 28;
```

```
Starting recover at 05-MAR-07
```

```
using channel ORA_DISK_1
```

```
channel ORA_DISK_1: restoring block(s)
```

```
channel ORA_DISK_1: specifying block(s) to restore from backup set
```

```
restoring blocks of datafile 00004
```

```
channel ORA_DISK_1: reading from backup piece <ol_mf_nnndf_TAG20070305T114043_2yqwx96_.bkp>
```

```
channel ORA_DISK_1: piece handle = <ol_mf_nnndf_TAG20070305T114043_2yqwx96_.bkp>
```

```
channel ORA_DISK_1: restored block(s) from backup piece 1
```

```
channel ORA_DISK_1: block restore complete, elapsed time: 00:00:01
```

```
starting media recovery
```

```
media recovery complete, elapsed time: 00:00:02
```

```
Finished recover at 05-MAR-07
```

ORACLE

## Data Recovery Advisor (EM)

ORACLE Enterprise Manager 11g Database Control

Database Instance: mystar >

Information

1. Database Failures - 1
2. Current Status - MOUNTED

Perform Recovery

Oracle Advised Recovery

The Data Recovery Advisor has detected failures. Click on "Advise and Recover" to have Oracle analyze and produce recovery advice.

Failures Detected Critical: 0 High: 1 Low: 0

Failure Description One or more non-system datafiles are missing

User Directed Recovery

Recovery Scope: Whole Database

Operation Type: ☒ Recover to the current time or a previous point-in-time

Flashback database option is available for faster recovery. You can also recover to a point-in-time prior to the Flashback log retention time. In that case, traditional restore and recovery will be used.

☐ Restore all datafiles

Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.

☐ Recover from previously restored datafiles

Overview

- Recover database failures as advised by Oracle
- Restore and/or recover the entire database or selected objects
- Restore files to a new location
- Recover tablespaces to a point-in-time based on a timestamp, system change number (SCN), or log sequence number
- Recover datafile data blocks that are marked as corrupted, or based on datafile block IDs or tablespace block addresses
- Flashback database or tables to a specific system change number (SCN) or timestamp

ORACLE

## Backup Multi-section

- Possibilité de paralléliser la sauvegarde d'un fichier
- RMAN va découper le fichier en tranches de <n> Mo
- La taille est définie par le paramètre `SECTION SIZE` de la commande `BACKUP`
- On peut aussi utiliser cette fonctionnalité pour paralléliser la vérification (`VALIDATE`) d'un fichier

```
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 3;  
RMAN> backup section size 100M tablespace users;
```

```
RMAN> validate tablespace sysaux section size 100M ;
```

ORACLE

## Backup Multi-section

```
RMAN> backup section size 80M tablespace sysaux;

Starting backup at 13-SEP-07
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=31 device type=DISK
allocated channel: ORA_DISK_2
...
input datafile file number=00002 name=/home/oral11/P1/sysaux01.dbf
backing up blocks 1 through 10240
...
input datafile file number=00002 name=/home/oral11/P1/sysaux01.dbf
backing up blocks 10241 through 15360
channel ORA_DISK_2: starting piece 2 at 13-SEP-07
channel ORA_DISK_1: finished piece 1 at 13-SEP-07
piece handle=/P1/rman_dest/0hrrmdc_1_1 tag=TAG20070913T163700 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
channel ORA_DISK_2: finished piece 2 at 13-SEP-07
...
```

ORACLE

## Validate Multi-section

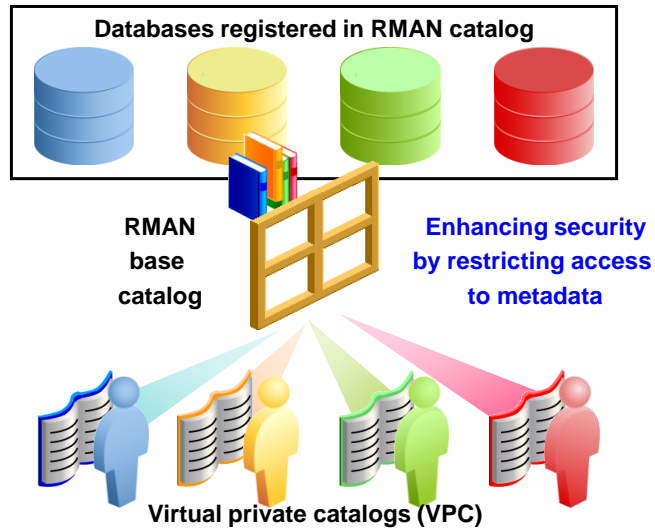
```
RMAN> validate tablespace sysaux section size 100M ;

...
List of Datafiles
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
2    OK      0              10799          15359          210137
File Name: /home/oral11/tmp/prichard/P1/sysaux01.dbf
Block Type Blocks Failing Blocks Processed
-----
Data          0              770
Index         0              847
Other         0              2943

Finished validate at 13-SEP-07
```

ORACLE

## Catalogue virtuel



ORACLE

## Catalogue virtuel

- Accorder des droits (administration, sauvegarde, restauration, enregistrement) à un utilisateur sur des bases du catalogue
- Pré-requis:
  - Privilège DATABASE RECOVERY\_CATALOG\_OWNER
  - Privilèges RMAN
    - CATALOG FOR DATABASE <db> FOR <user>
    - REGISTER DATABASE (optionnel)

```
SQL> grant recovery_catalog_owner to user_red;  
RMAN> grant catalog for database RED to user_red;  
RMAN> grant register database to user_red;
```

```
RMAN> connect user_red/***  
RMAN> create virtual catalog;
```

ORACLE

## Catalogue virtuel (...)

- Annuler des droits

```
RMAN> REVOKE REGISTER DATABASE FROM <user>;  
RMAN> REVOKE CATALOG FOR DATABASE <db> FROM <user>;  
RMAN> REVOKE ALL PRIVILEGES FROM <user>;
```

- Supprimer le catalogue virtuel

```
RMAN> connect user_red/***  
RMAN> DROP CATALOG;
```

ORACLE

## Fusion de catalogues

- La nouvelle commande `IMPORT CATALOG` permet de fusionner plusieurs catalogues existants.
- Ne peut pas importer des catalogues de version inférieure
- Se connecter sur le catalogue cible

```
RMAN> connect catalog rman/rman@rcvcat111
```

- Importer tout ou partie des catalogues sources

```
RMAN> import catalog rman/rman@rcvcat  
RMAN> import catalog rman/rman@rcvcat dbid=142341  
RMAN> import catalog rman/rman@rcvcat DB_NAME=P1
```

ORACLE

## Archivelog deletion policies

- Par défaut `NONE`: la suppression est basée sur la politique de rétention des sauvegardes
- Ne concerne que le contenu de la FRA
- Deux nouvelles possibilités pour gérer les archivelogs:
  - `BACKED UP <n> TIMES TO DEVICE TYPE {disk |sbt}`
  - `SHIPPED TO [ALL] STANDBY`

ORACLE

## Archival backup

- L'archivage « longue durée » existe depuis la 9iR1 mais il impliquait de conserver tous les archivelogs générés après la création de ce backup

```
RMAN> BACKUP DATABASE KEEP FOREVER LOGS;
```

- Suppression de cette contrainte en 11g. Le backup « longue durée » réalisé en 11g est auto-restaurable.

```
RMAN> BACKUP DATABASE KEEP UNTIL TIME 'sysdate + 365;
```

```
RMAN> BACKUP DATABASE KEEP UNTIL TIME 'sysdate + 365  
RESTORE POINT Q4FY07;
```

ORACLE

## Validation des fichiers

- La commande `VALIDATE` (précédemment limitée aux backupset) a été étendue aux fichiers de la base.
- Niveau base, tablespace, fichier, bloc, ...
- Plus simple à mettre en œuvre que DBVerify
- Tout problème détecté sera visible dans le Data Recovery Advisor

```
RMAN> validate database;  
RMAN> validate datafile 4 block 56;  
RMAN> validate datafile 1 block 10 to 100;  
RMAN> validate tablespace sysaux section size 100M ;
```

ORACLE

## Output d'un VALIDATE

```
RMAN> validate tablespace users;  
  
Starting validate at 15-OCT-07  
...  
input datafile file number=00004 name=/home/P1/users01.dbf  
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01  
List of Datafiles  
=====
```

File	Status	Marked	Corrupt	Empty	Blocks	Blocks	Examined	High	SCN
4	FAILED	0			597	640			183199

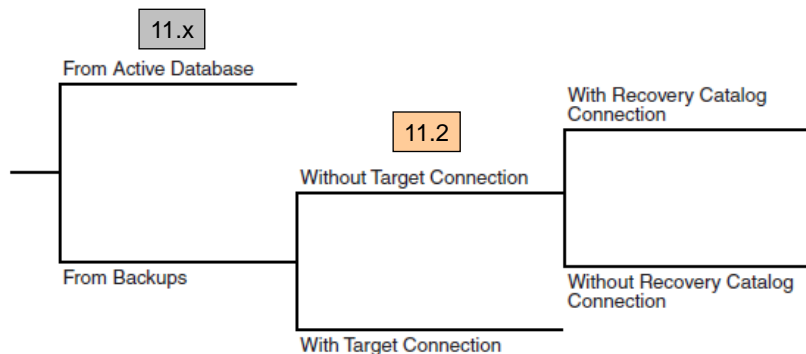
```
File Name: /home/ora111/tmp/prichard/P1/users01.dbf  
Block Type Blocks Failing Blocks Processed  
-----  
Data      0          14  
Index     0           2  
Other     1          27
```

validate found one or more corrupt blocks  
See trace file /home/P1/diag/rdbms/p1/P1/trace/P1\_ora\_6355.trc for details

ORACLE



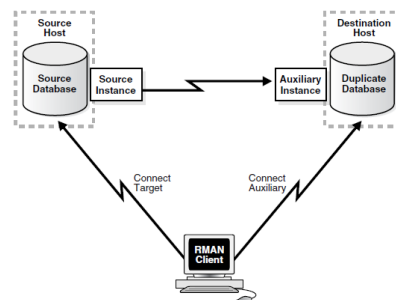
## Duplication: les différentes possibilités



ORACLE

## Duplication à travers le réseau

- Jusqu'à présent la duplication (copie ou standby) s'appuyait sur une sauvegarde de la base (*Backup-based duplication*)
- En 11g il est possible de dupliquer directement à partir de la base « target » (*Active database duplication*)



```
$ rman target=/ nocatalog auxiliary=u/p@alias
RMAN> duplicate target database to S1 from active database;
```

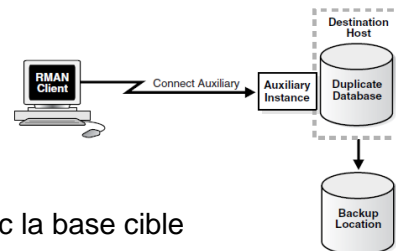
ORACLE

## Création d'un standby

```
run {
  duplicate target database for standby from active database
  spfile
    parameter_value_convert 'chicago','boston'
    set db_unique_name='boston'
    set db_file_name_convert='/chicago/', '/boston/'
    set log_file_name_convert='/chicago/', '/boston/'
    set control_files='/oracle/oradata/boston/control.ctl'
    set fal_client='boston'
    set fal_server='chicago'
    set standby_file_management='AUTO'
    set log_archive_config='dg_config=(chicago,boston)'
    set log_archive_dest_1='service=chicago LGWR ASYNC
      valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=chicago';
}
```

ORACLE

## Duplicate



- DUPLICATE sans connexion avec la base cible
- Sur le site cible:
  - Créer un fichier password
  - Créer un fichier init.ora
  - Démarrer l'instance

```
$ rman auxiliary=/
...
RMAN> duplicate database to P1
backup location '/home/ora112/tmp' nofilenamecheck;
```

ORACLE

## SET NEWNAME

- Extension au niveau tablespace ou database

```
RMAN> SET NEWNAME FOR TABLESPACE users TO NEW;  
RMAN> SET NEWNAME FOR TABLESPACE tools TO '/oradata%f/%b'  
RMAN> SET NEWNAME FOR DATABASE TO NEW;  
RMAN> SET NEWNAME FOR DATABASE TO '/oradata/newdir/%b';
```

- Masques de substitution:
  - %b basenome
  - %f absolute file number

ORACLE

## TableSpace Point In Time Recovery

- TSPITR permet de restaurer un tablespace supprimé.

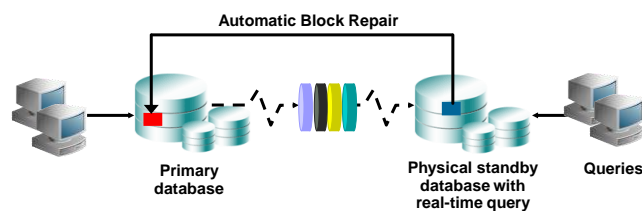
```
RMAN> recover tablespace users until scn 152391  
auxiliary destination '/tmp';
```

- Auparavant il fallait faire un DBPITR
  - Fichier de contrôle antérieur à la suppression
  - Mode nocatalog
  - Restauration
  - Recover incomplet

ORACLE

## Automatic Block Repair

- Fonctionnalité mixte RMAN / Data Guard
- Un bloc corrompu sur la primaire peut être récupéré à la volée sur la standby database et supprimer l'ORA-1578 synonyme de bloc corrompu (et vice-versa)
- Transparent à moins d'un échec
- Nécessite le mode real time query (Active Data Guard)



ORACLE

## Automatic Block Media Repair

```
Fractured block found during multiblock buffer read
Data in bad block:
  type: 6 format: 2 rdba: 0x01000095
  last change scn: 0x0000.0003149d seq: 0x10 flg: 0x04
  spare1: 0x0 spare2: 0x0 spare3: 0x0
  consistency value in tail: 0x00000000
  check value in block header: 0x248b
  computed block checksum: 0x128d
Reading datafile '/home/ora112/tmp/prichard/P1/users01.dbf' for corruption at rdba:
0x01000095 (file 4, block 149)
Reread (file 4, block 149) found same corrupt data
Starting background process ABMR
Thu Oct 01 11:19:54 2009
ABMR started with pid=26, OS id=5514
Auto BMR service is active.
Requesting Auto BMR for (file# 4, block# 149)
Waiting Auto BMR response for (file# 4, block# 149)
Auto BMR successful
```

ORACLE

## Recover BLOCK

- Pour réparer un bloc corrompu, la commande RECOVER DATAFILE x BLOCK y cherche:
  - Physical standby database (11gR2)
  - Flashback logs
  - Blocks in full or level 0 incremental backup
- On peut exclure la standby en précisant:

```
RMAN> recover datafile 4 block 128 exclude standby;
```

ORACLE

## Divers

- **UNDO optimization** (par défaut)
  - Les blocs UNDO des transactions validées ne sont pas sauvegardés.
- **Variables de substitution (&n) dans les scripts stockés**

```
CREATE SCRIPT quarterly
{
  ALLOCATE CHANNEL c1 DEVICE TYPE sbt
  PARS 'ENV=(OB_MEDIA_FAMILY=&1)';
  BACKUP TAG &2 FORMAT '&1%U.bck' DATABASE;
}
```

```
% rman TARGET=/ CATALOG=rman/rman@inst2 USING \
archival_backup bck0906
```

ORACLE

## Divers

- Nouvelle syntaxe « AS OF {SCN|Timestamp} » pour la création d'un RESTORE POINT
- Utilisation des flashback logs (s'ils existent) lors d'un Block Media Recovery
- Sauvegarde des tablespaces Read Only (greffés)
- Block Change Tracking disponible sur la standby physique et par conséquent les sauvegardes incrémentales rapides.
- Flash Recovery Area devient Fast Recovery Area
- **Archivelog failover**: fonctionnalité existante depuis la 9i étendue à la commande `BACKUP RECOVERY AREA`
- Duplication avec exclusion / inclusion de tablespaces:

```
RMAN> duplicate database ... tablespace USERS ...
```

ORACLE

## Divers

- Nouvelle option `SKIP UNNECESSARY DATAFILES` de la commande `CONVERT DATABASE` pour ne convertir que le nécessaire.
- Sauvegarde de la FRA sur disque

```
RMAN> backup recovery area to destination '/tmp';
```

ORACLE

# Deprecated RMAN commands

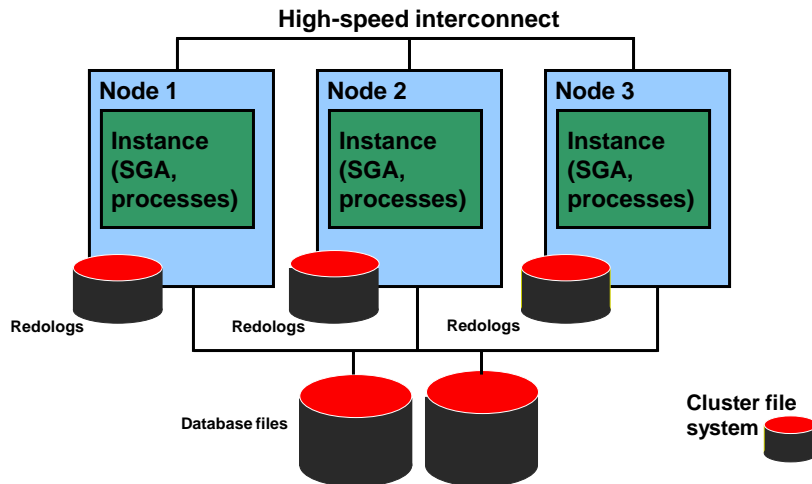
Table A-1    *Deprecated RMAN Syntax*

Deprecated in Release	Deprecated Syntax	Preferred Current Syntax
11.2.0	CONFIGURE or SET COMPRESSION ALGORITHM 'ZLIB'	CONFIGURE or SET COMPRESSION ALGORITHM 'MEDIUM'
11.2.0	CONFIGURE or SET COMPRESSION ALGORITHM 'BZIP2'	CONFIGURE or SET COMPRESSION ALGORITHM 'BASIC'
11.2.0	V\$FLASH_RECOVERY_AREA_USAGE	V\$RECOVERY_AREA_USAGE
11.1.0	CONVERT ON TARGET PLATFORM	CONVERT ON DESTINATION PLATFORM
11.1.0	UNTIL RESTORE POINT	TO RESTORE POINT
11.1.0	BACKUP ... AS STANDEY	n/a
11.1.0	... KEEP [LOGS   NOLOGS]	... KEEP
11.1.0	BLOCKRECOVER	RECOVER

# Annexes



## RMAN & RAC



ORACLE

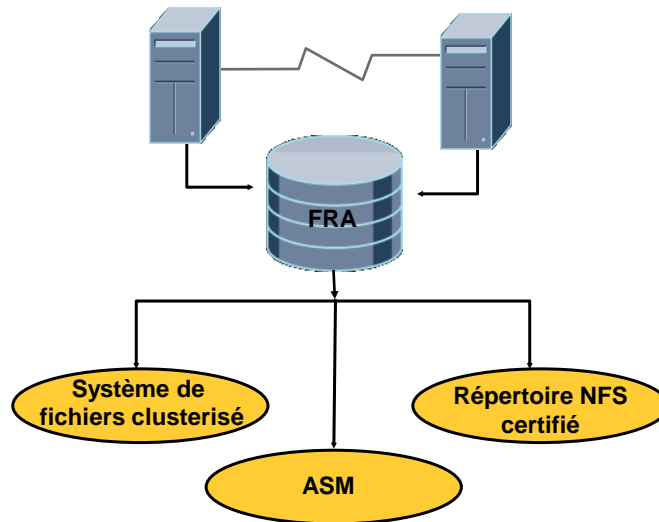
## RMAN & RAC

- La sauvegarde de la base peut s'effectuer indifféremment du nœud puisque les fichiers sont sur un FS partagé.
- En revanche chaque nœud a son propre thread de redologs:
  - Archivage local: une connexion sur chaque nœud sera nécessaire pour sauvegarder ses archivelogs
  - Un montage NFS rend les archivelogs visibles de tous les nœuds: rien de plus à faire
  - Utilisation d'un Cluster File System ou la FRA (en ASM): rien de plus à faire

ORACLE



## RMAN et Flash Recovery Area



ORACLE

## RMAN et RAC (sauvegarde)

- Il est possible de démarrer des processus sur chaque instance RAC existante ou d'utiliser l'auto balancing
- Conséquence: plusieurs noeuds de backup peuvent être actifs pour une même sauvegarde

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;  
CONFIGURE CHANNEL DEVICE TYPE sbt CONNECT='sys/rac@BR';
```

OU

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT='sys/rac@RACDB1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT='sys/rac@RACDB2';
```

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

ORACLE

## **RMAN et RAC (Restauration)**

- Un seul nœud assure la restauration (restore + recover) quelque soit le nombre de nœuds du cluster.
- Il doit « voir » toutes les sauvegardes plus éventuellement ce qui n'a pas été sauvegardé
- Vérifier que les nœuds sont interchangeables: ce qui a été sauvegardé par le nœud 1 peut être restauré par le nœud 2 et vice-versa

ORACLE