



OCM 认证培训教材

ORACLE 备份恢复篇



腾科 ORACLE 教学部

修订讲师：唐骏



RMAN 使用详解

一、连接方式

(一)、连接本地数据库

```
[oracle@oracle ~]$ rman target /
```

(二)、连接远程数据库

```
[oracle@oracle ~]$ rman target sys/oracle@orcl
```

二、基本指令

(一)、执行 SQL 语句

```
RMAN>SQL 'ALTER SYSTEM SWITCH LOGFILE';
```

(二)、RMAN 默认配置

1. 显示 RMAN 默认配置

```
RMAN>SHOW ALL;
```

2. RMAN 默认配置详解

```
using target database control file instead of recovery catalog
RMAN configuration parameters 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';
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET;
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 ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/10.2.0/db_1/dbs/snapcf_orcl.f'; # default
```



#default 表示该条配置仍然是初始的默认配置。

如果我们想把某条更改过的配置再置为初始应该怎么办呢，很简单，
用如下命令：**CONFIGURE ... CLEAR;**

例如：

```
RMAN> CONFIGURE RETENTION POLICY CLEAR;
```

旧的RMAN 配置参数：

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 3 DAYS;
```

RMAN 配置参数已成功重置为默认值

1、CONFIGURE RETENTION POLICY 配置备份保留策略

两种保留策略：

基于时间：

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

基于冗余数量：

```
CONFIGURE RETENTION POLICY TO REDUNDANCY n;
```

也可以取消备份保留策略：

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

2、CONFIGURE BACKUP OPTIMIZATION 配置备份优化

如果优化设置打开，将对备份的数据文件、归档日志或备份集运行一个优化算法。

备份优化，包括OFF 和ON 两个状态

打开备份优化：

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

关闭备份优化：

```
CONFIGURE BACKUP OPTIMIZATION OFF;
```

3、CONFIGURE DEFAULT DEVICE TYPE 配置IO 设备类型

RMAN 支持的 IO 设备类型有两种：磁盘(DISK)和磁带(SBT)，默认情况下为磁盘。

使用磁盘设备：

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
```

使用磁带设置：

```
CONFIGURE DEFAULT DEVICE TYPE TO SBT;
```

在这里需要注意的一点是：如果IO 设备发生变化，相关配置项也需要修改。例如：

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



4、CONFIGURE CONTROLFILE AUTOBACKUP 配置控制文件自动备份

是否自动备份，包含两个状态：OFF 和ON

打开自动备份

```
CONFIGURE CONTROLFILE AUTOBACKUP ON
```

禁止自动备份

```
CONFIGURE CONTROLFILE AUTOBACKUP OFF
```

同时可以通过如下配置指定备份的控制格式，路径。例如：

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE  
DISK TO 'd:/backup/%F';
```

在备份期间，将产生一个控制文件的快照，用于控制文件的读一致性，这个快照可以通过如下配置：

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO  
'D:/BACKUP/SNCFJSSWEB.ORA';
```

5、CONFIGURE DEVICE TYPE 设置并行备份

RMAN 支持并行备份与恢复，也可以在配置中指定默认的并行程度。例如：

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
```

指定在以后的备份与恢复中，将采用并行度为2，同时开启2个通道进行备份与恢复，当然也可以在run 中指定通道来决定备份与恢复的并行程度。

默认情况下，自动分配通道的并行度为1，如果你通过设置PARALLELISM 设置了并行通道为2，那么在run 块中，如果你没有单独通过ALLOCATE CHANNEL 命令指定通道，它会默认使用2 条并行通道，如果你在run 命令块中指定了数个ALLOCATE CHANNEL，那么rman 在执行备份命令时会以你设置的channel 为准，而不管configure中配置了多少个并行通道。

需要注意的一点是，在backup 命令中有一个 FILESPERSET 参数该参数是指rman建立的每个备份集中所能包含的备份片段(即磁盘文件)的最大数，该参数默认值为64，如果在执行backup 命令时没有指定该参数值，那么rman 会仅使用第一个通道来执行备份，其它通道将处于空闲状态。filesperset 值不要小于你设定的通道数。

6、CONFIGURE DATAFILE BACKUP COPIES 设置备份文件冗余度

7、CONFIGURE MAXSETSIZE 配置备份集的最大尺寸

该配置限制通道上备份集的最大尺寸。单位支持bytes,K,M,G。默认值是unlimited。

```
RMAN> show channel; // 通道分配  
RMAN> show device type; // IO 设备类型  
RMAN> show retention policy; // 保存策略  
RMAN> show datafile backup copies; // 多个备份的拷贝数目  
RMAN> show maxsetsize; // 备份集大小的最大值  
RMAN> show exclude; // 不必备份的表空间  
RMAN> show backup optimization; // 备份的优化
```



(三)、FORMAT 字符串替代变量

使用FORMAT 参数时可使用的各种替换变量，如下：

%c: 备份片的拷贝数(从1 开始编号)；

%d: 数据库名称；

%D: 位于该月中的天数(DD)；

%M: 位于该年中的月份(MM)；

%F: 一个基于DBID 唯一的名称,这个格式的形式为c-lllllllll-YYYYMMDD-QQ,其中lllllllll 为该数据库的DBID, YYYYMMDD 为日期, QQ 是一个1-256 的序列；

%n: 数据库名称，并且会在右侧用x 字符进行填充，使其保持长度为8；

%u: 是一个由备份集编号和建立时间压缩后组成的8 字符名称。利用%u 可以为每个备份集生成一个唯一的名称；

%p: 表示备份集中备份片段的编号，从1 开始编号；

%U: 是%u_%p_%c 的简写形式，利用它可以为每一个备份片段（即磁盘文件）生成一个唯一名称，这是最常用的命名方式；

%s: 备份集的号；

%t: 备份集时间戳；

%T: 年月日格式(YYYYMMDD)； s

注：如果在BACKUP 命令中没有指定FORMAT 选项，则RMAN 默认使用%U 为备份片段命名。

(四)、RMAN 备份相关的动态性能表

V\$ARCHIVED_LOG: 本视图包含了所有归档重做日志文件的创建情况，备份情况以及其他信息。

V\$BACKUP_CORRUPTION: 这个视图显示了RMAN 在哪些备份集中发现了损坏的数据块。在你使用BACKUP VALIDATE 命令对备份集进行检查时如果发现了损坏的数据块，RMAN 将在这个视图中写入记录。

V\$COPY_CORRUPTIO: 本视图显示了哪些镜像复制备份文件已经被损坏。

V\$BACKUP_DATAFILE: 本视图通常用来获取每个数据文件中非空白数据块的数量，从而帮助你创建出大小基本相等的备份集。另外，在视图中也包含了数据文件中损坏的数据块的信息。

V\$BACKUP_REDOLOG: 本视图显示了在现有的备份集中饮食有哪些归档重做日志文件。

V\$BACKUP_SET: 本视图显示了已经创建的备份集的信息。

V\$BACKUP_PIECE: 本视图显示了已经创建的备份片段的信息。



(五)、备份信息

1. 列出备份信息

列出数据库中所有文件的备份信息：

```
RMAN>LIST BACKUP OF DATABASE;
```

列出指定表空间的备份信息：

```
RMAN>LIST COPY OF TABLESPACE 'SYSTEM';
```

列出指定数据文件的备份信息：

```
RMAN>LIST BACKUP OF DATAFILE  
'F:\ORAHOME1\ORADATA\JSSWEB\JWEB.ORA';
```

2. 删除备份信息

1. 删除陈旧备份

当使用 RMAN 执行备份操作时，RMAN 会根据备份冗余策略确定陈旧备份。

```
RMAN> delete obsolete;
```

2. 删除 EXPIRED 备份

执行 **crosscheck** 命令核对备份集，如果备份集丢失或者损坏，那么会将该备份集标记为 **EXPIRED** 状态。为了删除相应的备份记录，可以执行 **delete expired backup** 命令。

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

3. 删除 EXPIRED 副本

```
RMAN> delete expired copy;
```

4. 删除特定备份集

```
RMAN> delete backupset 19;
```

5. 删除特定备份片

```
RMAN> delete backuppiece 'd:\backup\DEMO_19.bak';
```

6. 删除所有备份集

```
RMAN> delete backup;
```

7. 删除特定映像副本

```
RMAN> delete datafilecopy 'd:\backup\DEMO_19.bak';
```

8. 删除所有映像副本

```
RMAN> delete copy;
```

9. 在备份后删除输入对象

```
RMAN> delete archivelog all delete input;
```

```
RMAN> delete backupset 22 format = "d:\backup\%u.bak" delete input;
```



(六)、指令运行方式

1. 单个执行

```
RMAN>backup database;
```

2. 批处理

```
RMAN>RUN{  
.....  
.....  
}
```

批处理方式实质是将原来单个执行的命令组合到一起，并放在run{}之间，这种方式最大的好处是所有在run 中的命令被视为一个作业，如果作业中任何一条命令执行失败，则整个命令停止执行，即失败命令后的其它命令都不会再被执行。这种方式是最常使用的方式，特别对于后台执行。

3. 运行脚本(又能细分为几种方式如下：)

```
[oracle@oracle ~]$ rman TARGET / @backup_db.rman  
RMAN> @backup_db.rman  
RMAN> RUN { @backup_db.rman }
```

运行存储在恢复目录中的脚本(需要首先为rman 创建恢复目录)

```
RMAN> RUN { EXECUTE SCRIPT backup_whole_db };
```

4. 操作系统脚本(linux,windows 环境下小有差异，但只是操作系统命令的不同)

```
[oracle@oracle ~]$rman cmdfile=backup_db.rman
```




三、RMAN 备份概念

(一)、RMAN 备份类型

1. 镜像复制

首先大家需要明了这个概念，`rman` 中的镜像复制实质与通过操作系统 `copy` 命令备份相同，甚至连命令的格式都相似，只不过直接应用操作系统的 `copy` 命令复制数据文件时，只是文件拷贝，而 `rman` 的 `copy` 则能够在复制的同时，验证数据的有效性。

2. 备份集

备份集(BACKUPSET)表示进行一次备份所产生的所有备份片集合,是一个逻辑上的概念。备份片(BACKUPPIECE)表示一个由 RMAN 产生备份的文件.用 OS 工具可以实实在在的看到。一个备份集由一个或多个备份片组成。(与表空间、数据文件关系类似)

一个数据文件可以跨备份片存在,而不能跨备份集存在。

示例（数据库 2G，开启归档）：

```
RMAN> run
{
  allocate channel c1 device type disk maxpiecesize 1500m;
  backup database plus archivelog delete all input;
  release channel c1;
}
```

每个备份片的最大为 1500M,超过这个大小就会产生新的备份片,在启动备份时,会归档当前的日志,生成一个备份片,删除已备份的归档日志.然后备份数据文件,生成两个备份片(数据文件 2G),再备份当前的 SPFILE 和 CONTROLFILE,生成一个备份片,最后再做一次 LOG SWITCH,最备份这归档日志,生成一个备份片,因此可以这样描述,这个备份集包含成五个备份片，一个备份片包含文件个数由 FILESPERSET 指定。

```
RMAN> run
{
  allocate channel c1 device type disk maxpiecesize 1500m;
  backup database filesperset 1;
  release channel c1;
}
```

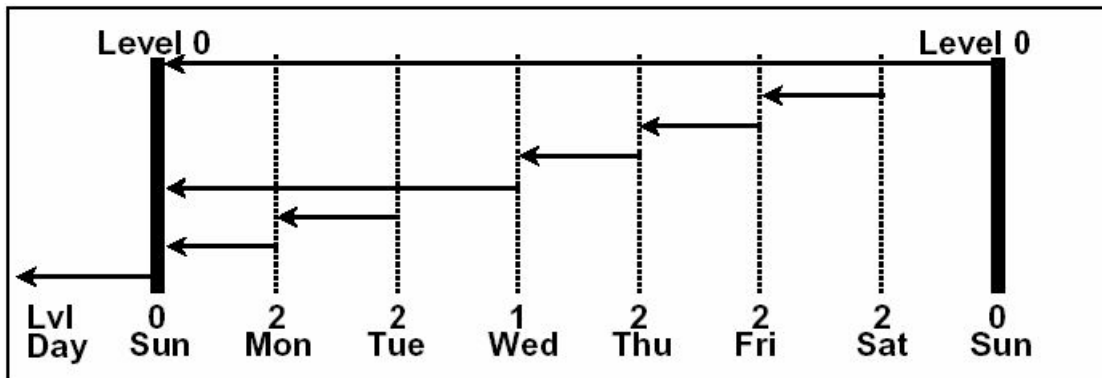
表示一个备份片中包含一个文件,即使没有达到 1500M,也生成新的备份片,如我的有五个数据文件,还有 SPFILE 和 CONTROLFILE 一个备份片,一共生成六个备份片.如果加上 `plus archivelog delete all input` 这个备份选项之后,那 `FILESPERSET` 这个参数就会被 IGNORE 掉。

(二)、增量备份的工作机制

所谓增量备份，顾名思义即是每次备份仅操作那些发生了"变化"的数据块。RMAN 中增量备份有两种：Differential 方式和Cumulative 方式。级别0-4.

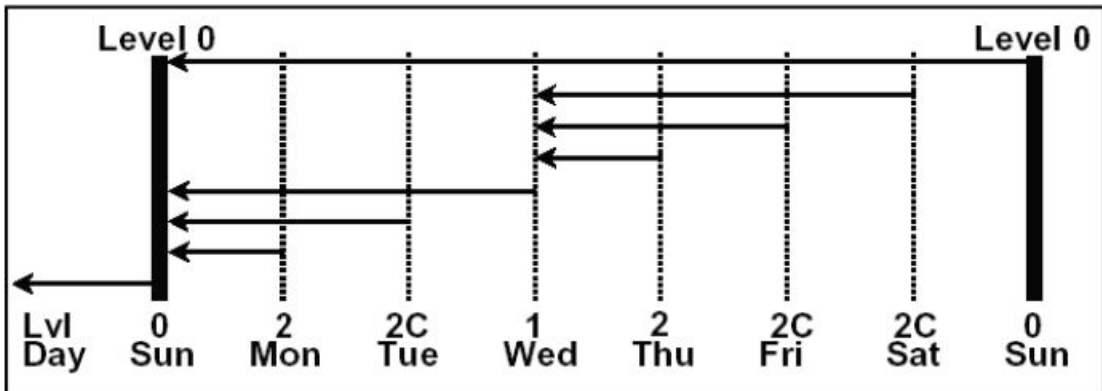
1、差异备份 Differential (DEFAULT)

备份上级及同级以来所有变化的数据块



2、累积增量备份 Cumulative

备份上级以来所有变化的数据块



0 级备份和全库备份的区别：

- 1) 全库备份不能作为 1 级备份的基础
- 2) 全库备份会对 unused 数据库做备份

0 级备份

```
backup incremental level 0 database;
```

1 级差异增量

```
backup incremental level 1 database;
```

1 级累计增量

```
backup incremental level 1 cumulative database;
```



(三)、备份集复制

RMAN 提供了一种更谨慎的备份策略：**Duplexed**方式备份，其实质即是在生成备份集的同时，向指定位置生成指定份数(最大不超过4 份)的备份集复制，以避免在灾难性事故时数据库损坏和备份丢失的情况下导致完全崩溃，提高备份可用性。

RMAN 中提供了三种方式实现Duplexed 方式备份：

1、在RMAN 中执行BACKUP 命令时显式指定COPIES 参数。例如：

```
RMAN> BACKUP COPIES 3 DATABASE;
```

上述命令将会在全库备份的同时，自动生成当前备份集的2 份拷贝到默认备份目录。

2、在RUN {}命令块中利用SET BACKUP COPIES 命令为该命令块中所有的BACKUP 命令设置Duplexed 方式，例如：

```
RMAN> RUN{  
SET BACKUP COPIES 2;  
BACKUP DEVICE TYPE DISK FORMAT 'D:\BACKUP1\%U','D:\BACKUP2\%U'  
TABLESPACE USERS,SALES;  
}
```

上述命令将生成两份备份集，分别存储到d:\backup1 和d:\backup2 目录。

3、通过CONFIGURE BACKUP COPIES 命令设置预定义的备份Duplexed 方式。

CONFIGURE ... BACKUP COPIES 命令可以为指定的设备类型设置默认的备份拷贝量。这个配置仅适用于数据文件与归档重做日志文件和备份，并且，只有在使用自动分配的通道时才能够使用CONFIGURE ... BACKUP COPIES 命令设置的配置。

例如：

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;  
RMAN> CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE disk TO 2;  
RMAN> CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE disk TO 2;
```

上述命令将disk 设置上数据文件与归档文件的拷贝数量设置为2，当再执行BACKUP DATABASE 备份时，即会自动生成 2 份数据文件的备份集。



(四)、对数据库进行备份

1. 对数据库进行全备

1. 使用backup database 命令执行备份

```
RMAN> BACKUP DATABASE;
```

执行上述命令后将对目标数据库中的所有数据文件进行备份，由于没有显式指定FORMAT 参数，rman 会自动为每个备份片段命名，并保存在ORACLE_HOME/database 目录下。

当然，也可以通过显式指定format 参数来自定义备份片段的命令规则，比如：

```
RMAN> BACKUP DATABASE FORMAT 'D:\BACKUP\%U';
```

2. 通过list 命令查看刚刚创建的备份信息：

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

正在使用目标数据库控制文件替代恢复目录

备份集列表

=====

BS 关键字类型LV 大小设备类型经过时间完成时间

11 Incr 0 827M DISK 00:01:30 26-6 月-07

BP 关键字: 11 状态: AVAILABLE 标记:TAG20070629T150701

段名:D:\BACKUP\0CILGG8L_1_1

备份集11 中的数据文件列表

文件LV 类型Ckp SCN Ckp 时间名称

1 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\SYSTEM01.DBF

2 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\UNDOTBS01.DBF

3 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\DRSYS01.DBF

4 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\EXAMPLE01.DBF

5 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\INDX01.DBF

6 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\ODM01.DBF

7 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\TOOLS01.DBF

8 0 Incr 5233577896 26-6 月-07

F:\ORAHOME1\ORADATA\JSSWEB\USERS01.DBF



2. 备份表空间

只要实例启动并处于加载状态，不论数据库是否打开，都可以在rman 中对表空间进行备份，而且不需要像手动备份那样先'ALTER TABLESPACE ... BEGIN BACKUP'。

例如：

```
RMAN> backup tablespace jweb;
```

启动backup 于26-6 月-07

使用通道ORA_DISK_1

通道ORA_DISK_1: 正在启动full 数据文件备份集

通道ORA_DISK_1: 正在指定备份集中的数据文件

输入数据文件fno=00010 name=F:\ORAHOME1\ORADATA\JSSWEB\JWEB.ORA

通道ORA_DISK_1: 正在启动段1 于26-6 月-07

通道ORA_DISK_1: 已完成段1 于26-6 月-07

段handle=F:\ORAHOME1\DATABASE\0JIM0BBV_1_1 comment=NONE

通道ORA_DISK_1: 备份集已完成，经过时间:00:00:15

完成backup 于26-6 月-07

启动Control File and SPFILE Autobackup 于26-6 月-07

段handle=D:\BACKUP\C-3391142503-20070704-00 comment=NONE

完成Control File and SPFILE Autobackup 于26-6 月-07

同样，我们在使用backup tablespace 也可以通过显式指定format 参数自定义备份片段名称。

我们再通过LIST BACKUP 查看一下备份：

```
RMAN> LIST BACKUP OF TABLESPACE JWEB;
```

发现有两份备份片段，，删除其中一个备份(删除的方法非常多，下例使用删除backupset，有心的朋友不妨自己尝试一下通过其它方式删除指定备份片段)。

```
RMAN> DELETE BACKUPSET 18;
```

3. 备份指定数据文件

首先可以先通过数据字典DBA_DATA_FILES 查询出表空间对应的数据文件及其序号，例如：

```
select file_name,file_id,tablespace_name from dba_data_files
```

然后再通过BACKUP DATAFILE 备份指定序号的数据文件，例如：

```
RMAN> BACKUP DATAFILE n;
```

如果要查看指定数据文件的备份，可以用：

```
RMAN> LIST BACKUP OF DATAFILE n;
```

注：n=指定的的数据文件序号，如果需要备份的数据文件有多个， n=多个序号，中间以逗号分隔即可。



4. 备份控制文件

控制文件在 ORACLE 中的重要性是毋庸置疑的，你看，连 RMAN 都给予特殊的照顾，在 RMAN 中备份控制文件的方式是最多的。

1. 最简单的方式，通过 CONFIGURE 命令将 CONTROLFILE AUTOBACKUP 置为 ON。

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

然后你再通过 rman 做任何备份操作的同时，都会自动对控制文件做备份。

2. 对编号为 1 的数据文件，即 SYSTEM 表空间的数据文件做备份时，RMAN 也会自动对控制文件做备份。

3. 手动执行备份命令。

```
RMAN> BACKUP CURRENT CONTROLFILE;
```

4. 执行 BACKUP 时指定 INCLUDE CURRENT CONTROLFILE 参数，例如：

```
RMAN> BACKUP DATABASE INCLUDE CURRENT CONTROLFILE;
```

如果要查看备份的控制文件，可以通过：

```
RMAN> LIST BACKUP OF CONTROLFILE;
```

5. 备份归档日志文件

归档日志对于数据库介质恢复相当关键，其虽然不像控制文件那样一旦损坏则数据库马上崩溃但重要性丝毫不减。归档日志文件能确保我们将数据库恢复到备份之前的任意时刻。

在RMAN 中备份归档日志有两种方式：

1、利用BACKUP ARCHIVELOG 命令备份

```
RMAN> BACKUP ARCHIVELOG ALL;
```

2、在BACKUP 过程中利用PLUS ARCHIVELOG 参数备份，例如：

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

则在备份数据库的同时自动对所有归档文件进行备份。这种方式与上种有什么区别呢，区别太明显了，BACKUP.....PLUS ARCHIVELOG 命令在备份过程中会依次执行下列步骤：

1>.运行ALTER SYSTEM ARCHIVE LOG CURRENT 语句对当前redolog 进行归档。

2>.执行BACKUP ARCHIVELOG ALL 命令备份所有已归档日志。

3>.执行BACKUP 命令对指定项进行备份。

4>.再次运行ALTER SYSTEM ARCHIVE LOG CURRENT 对当前redolog 归档。

5>.对新生成的尚未备份的归档文件进行备份。

在完成归档日志文件备份后，我们通过需要删除已备份过的归档文件(归档文件记录下了数据库进行过的所有操作，如果您的数据库操作频繁而且量大，那归档文件大小也是相当恐怖，备份后删除释放存储空间是相当有必要地)。RMAN 提供了DELETE ALL INPUT 参数，加在BACKUP 命令后，则会在完成备份后自动删除归档目录中已备份的归档日志。完成备份之后，可以通过下列命令查看已备份的归档日志片段：

```
RMAN> LIST BACKUP OF ARCHIVELOG ALL;
```

综述：上述示例中出现的命令格式较多，但细细看来是否能发现些格式上的规律？比如查看备份基本都是LIST BACKUP OF 备份项名称，如果想自定义片段的话则在备份命令后加上FORMAT 参数。



四、RMAN 备份的保存策略

如果你的数据库非常大,并且备份执行也比较频繁(不大不频繁也得这么干,优秀的dba 一定要拥有对应其身份的良好的工作习惯),有必要对你这些备份文件的保存制订合理的策略。该挪的挪,该搬的搬,该删除的删,合理释放,最大化利用有限的磁盘空间。

在通过RMAN 创建的备份片段中,由于备份文件也是由rman 创建和维护,所以手工删除并不明智,并且RMAN 也提供了备份保留策略,合理制订,由RMAN 自动删除过旧的备份文件更加安全也更加方便。

RMAN 中提供了两种备份保留策略:

基于时间

基于冗余数量

为RMAN 设置了备份保留策略之后, RMAN 会自动判断哪些备份集或镜像复制文件不必再保留。这些备份文件将会被标记为“废弃(Obsolete)”,可以通过REPORT OBSOLETE 命令查看当前处于废弃状态的备份文件,或者通过DELETE OBSOLETE 命令删除这些废弃的备份。

例如:

```
RMAN> report obsolete;
```

正在使用目标数据库控制文件替代恢复目录

RMAN 保留策略将应用于该命令

将RMAN 保留策略设置为3 天的恢复窗口

已废弃的备份和副本报表

类型关键字完成时间文件名/句柄

备份集21 04-7 月-07

备份段21 04-7 月-07 D:\BACKUP\C-3391142503-20070704-01

```
RMAN> delete obsolete;
```

RMAN 保留策略将应用于该命令

将RMAN 保留策略设置为3 天的恢复窗口

分配的通道: ORA_DISK_1

通道ORA_DISK_1: sid=14 devtype=DISK

删除以下已废弃的备份和副本:

类型关键字完成时间文件名/句柄

备份集21 04-7 月-07

备份段21 04-7 月-07 D:\BACKUP\C-3391142503-20070704-01

是否确定要删除以上对象(输入YES 或NO)? y

已删除备份段

备份段handle=D:\BACKUP\C-3391142503-20070704-01 recid=21 stamp=6270616451 对象已删除



在执行删除命令时有两点需要了解：

- 如果被判断为废弃的备份是一个单独数据文件的镜像复制，那么在执行DELETE 命令时将直接删除这个镜像复制文件。
- 如果被判断为废弃的备份是一个备份集中的一部分，则必须等到整个备份集中所有其它文件都被废弃之后，才能删除这个备份集。

1. 基于时间的备份保留策略。

说的简单些，就是你希望数据库最早能恢复到几天前。比如将恢复时间段设置为7，那么RMAN 所保留的备份即是可以保证你将数据库恢复到一周内任何时刻下那些文件。

设置基于时间的备份保留策略可以通过CONFIGURE 命令，例如：

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF n DAYS;
```

注：n=大于0 的正整数

执行该命令后，RMAN 将始终保留那些将数据库恢复到n 天前的状态时需要用到的备份，比如，恢复时间段被设置为7 天，那么各个数据文件的备份必须满足如下条件：

```
SYSDATE-(SELECT CHECKPOINT_TIME FROM V$DATAFILE)>=7
```

任何不满足上述条件的备份都将被RMAN 废弃并可通过DELETE OBSOLETE 命令删除。

2. 基于冗余数量的备份保留策略

基于冗余数量实质即某个数据文件以各种形式（包括备份集和镜像复制）存在的备份的数量。如果某个数据文件的冗余备份数量超出了指定数量，RMAN 将废弃最旧的备份。同样，基于数量的备份保留策略也是通过CONFIGURE 命令设置，例如：

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF n DAYS;
```

同上：n=大于0 的正整数

你也可以通过下列命令设置成不采用任何备份保留策略：

```
RMAN> CONFIGURE RETENTION POLICY TO NONE;
```




五、备份优化

RMAN 中的备份优化(Backup Optimization)是指在备份过程中，如果满足特定条件，RMAN 将自动跳过某些文件而不将它们包含在备份集中以节省时间和空间。说的直白些就是能不备的它就不备了，不像原来甭管文件有没有备份过统统再备一遍。

通常必须满足如下几个条件的情况下，才能够启用备份优化的功能：

- CONFIGURE BACKUP OPTIMIZATION 参数置为on;
- 执行的BACKUP DATABASE 或BACKUP ARCHIVELOG 命令中带有ALL 或 LIKE 参数。
- 分配的通道仅使用了一种设备类型，也就是没有同时分配使用sbt 与disk 的多个通道。

打开备份优化设置通过如下命令：

```
RMAN> CONFIGURE BACKUP OPTIMIZATION ON;
```

那么在进行备份优化时，RMAN 是如何判断要备份的文件是否需要被优化呢，这个算法就相当复杂了，而且可能影响优化算法的因素也非常多，假如某库在上午9 点被执行过一次全库备份，等下午3 点再次执行全库备份时，备份的文件没有变动而且也已经被备份过时，才会跳过这部分文件。所以理论上备份优化仅对于只读表空间或offline 表空间起作用。当然对于已经备份过的 archivelog 文件，它也会跳过。

启动备份优化后，如果发现相同的文件已经备份过，在不影响备份策略的情况下，则跳过。



六、RESTORE/RECOVER 恢复命令选项

(一)、相关指令

1. 数据库恢复

```
RMAN> restore/recover database ;
```

2. 表空间恢复

```
RMAN> restore/recover tablespace xx ;
```

3. 只读表空间的恢复

默认情况下，即使丢失了只读的数据文件，RMAN 也不会执行完全数据库还原操作时候还原只读的数据文件。

要在完全恢复期间还原只读的数据文件，就必须在RESTORE命令中使用CHECK READONLY参数：

```
RMAN> RESTORE DATABASE CHECK READONLY;
```

4. 恢复 SPFILE/控制文件

使用自动备份恢复SPFILE/控制文件

```
RMAN> startup nomount;  
RMAN> set dbid=153910023  
RMAN> restore controlfile from autobackup  
RMAN> restore spfile/controlfile to '/xx/xx' from autobackup ;
```

或

```
RMAN> restore controlfile from '/arch/ct_c-2347671489-20060630-00';
```

联机状态：目标数据库MOUNT或OPEN

```
RMAN> restore controlfile to 'd:\temp\control01.ctl';
```

5. 归档重做日志的还原

```
RMAN> RESTORE ARCHIVELOG ALL;  
RMAN> RESTORE ARCHIVELOG FROM LOGSEQ=1 UNTIL LOGSEQ=20;  
RMAN> RESTORE ARCHIVELOG FROM LOGSEQ=1;
```



也可以用SET命令来指定归档日志的还原位置，例如：

```
RMAN> run{
set archivelog destination to "d:\temp";
restore archivelog all;
}
```

6. 数据文件副本还原

```
RMAN> sql "alter datafile 5 offline";
RMAN> restore (datafile 5) from datafilecopy;
RMAN> recover datafile 5;
RMAN> sql "alter datafile 5 online";
```

请注意，上面的圆括号很重要

7. 还原检查与恢复测试

与备份检查一样，还原操作也可以检查是否能正常restore 或者是否该备份集是否有效。

如：

```
RMAN> RESTORE DATABASE VALIDATE;
RMAN> VALIDATE BACKUPSET 218;
RMAN> RESTORE DATABASE VALIDATE CHECK LOGICAL ;
```

8. 从指定的 tag 恢复

```
RMAN> RESTORE FROM tag=' xxxx' ;
```

9. 不完全恢复的还原：

```
RMAN> restore database until scn 1000;
RMAN> restore database "to_date('2005/08/01 13:00:00','yyyy/mm/dd hh24:mi:ss)";
RMAN> restore database until sequence 100 thread 1;
```

10. 块级别的恢复

块恢复Block Media Recovery (BMR)，块是恢复的最小单元，通过块可以减少恢复时间，而且数据文件可以在线。

恢复块的时候，必须指定具体的块号，如：

```
RMAN> blockrecover datafile 6 block 3;
```



(二)、非归档模式下的 BACKUP 备份与恢复

恢复目录： 打开

目标数据库：例程启动，数据库加载，mount 不能open

因为目标数据库不在归档模式下，所以当进行备份/恢复操作的时候数据库无法打开。目标数据库只能在MOUNT状态不能Open，所以属于脱机备份。

非归档模式不备份redo 日志，只有完全备份和readonly/offline 表空间和数据文件备份是有意义的，所以非归档模式最好不用RMAN 进行备份，备份语法与归档模式相同，所以这里只做简单介绍。

1. 全库备份

例： 使用默认的设置脱机全备份的语句

例： 不使用默认的设置执行脱机备份操作，在备份命令中指定备份选项

```
RMAN> shutdown immediate;
RMAN> startup mount;
RMAN> run {
allocate channel c1 type disk format '/xxx/ming_%U';
allocate channel c2 type disk format '/xxx/ming_%U';
backup full tag full_db_backup format "/backups/db_t%t_s%s_p%p" (database);
backup current controlfile;
release channel c1 ;
release channel c2 ;
}
```

在这个例子中，我们分配了两个通道，备份位置是在/xxx。默认情况下，如果备份数据文件1 (SYSTEM 表空间)，控制文件和参数文件也会备份。

可以通过下面的命令显示恢复目录中记载的备份集信息：

```
RMAN> list backupset of database;
```

2. 全库备份的恢复

```
$ rman target /
RMAN> startup mount
RMAN> restore database;
RMAN> recover database;
RMAN> alter database open resetlogs;
```



3. 表空间备份

只有readonly/offline 表空间的备份才有意义。

```
RMAN> run {  
  allocate channel dev1 type disk;  
  backup  
  tag tbs_users_read_only  
  format "/oracle/backups/tbs_users_t%t_s%s"  
  (tablespace users);  
}
```

使用下列命令来显示恢复目录中该表的备份信息：

```
RMAN> list backupset of tablespace users;
```

4. 表空间备份的恢复

```
RMAN> RUN{  
  sql "alter tablespace xx offline immediate;"  
  restore tablespace xx;  
  recover tablespace xx;  
  sql "alter tablespace xx online;"  
}
```

5. 备份控制文件

```
RMAN> run {  
  allocate channel dev1 type "SBT_TAPE";  
  backup  
  format "cf_t%t_s%s_p%p"  
  tag cf_monday_night  
  (current controlfile);  
  release channel dev1;  
}
```

注：数据库完全备份将自动备份控制文件。(或者备份时加include current controlfile)



(三)、归档模式下的 BACKUP 备份与完全恢复

要用RMAN 进行联机备份操作，数据库就必须位于ARCHIVELOG 模式。恢复目录必须打开，目标数据库例程必须启动，数据库加载或者打开。

1. 整库备份与恢复

1) 备份命令：

只备份数据文件（如果configure controlfile **autobackup on**; 将自动包括控件文件，SPFILE):

```
RMAN> backup database;
```

同时备份归档日志，然后将备份后的归档日志删除

```
RMAN> backup database plus archivelog delete input;
```

明确指定同时备份控件文件：

```
RMAN> run{
allocate channel c1 type disk;
sql 'alter system archive log current';
backup full database include current controlfile tag 'dbfull'
format '/u06/oracle/rmanback/full_%u_%s_%p';
sql 'alter system archive log current';
release channel c1;
```

可以用RMAN 的plus archivelog选项简化数据库备份：

```
RMAN> run {
backup database
format '/xxfull%d_%T_%s'
plus archivelog
format '/xx/arch_%d_%T_%s'
delete all input;
}
```

2) 完全恢复：

目标数据库必须是mount 状态

```
$ rman target /
RMAN> startup mount
RMAN> restore database;
RMAN> recover database;
RMAN> alter database open;
```



2. 表空间的备份与恢复

1) 备份命令:

```
RMAN> backup tablespace users ;  
例:  
RMAN> RUN{  
allocate channel c1 type disk;  
backup tablespace users tag 'ts_users' format '/oracle/rmanback/ts_%u_%s_%p' ;  
release channel c1;  
}
```

2) 恢复:

如果我们只丢失了特定的表空间的数据文件，那么我们可以选择只恢复这个表空间，而不是恢复整个数据库。

表空间恢复可以在不关闭数据库的情况下进行，只需要将需要恢复的表空间offline。

```
RMAN> RUN{  
sql "alter tablespace xx offline immediate;"  
restore tablespace xx;  
recover tablespace xx;  
sql "alter tablespace xx online;"  
}
```

恢复到一个不同的位置:

```
RMAN> RUN{  
sql "alter tablespace xx offline immediate;"  
SET NEWNAME for datafile 1 to '/xx';  
restore tablespace xx;  
switch datafile 1;  
recover tablespace xx;  
sql "alter tablespace tbs1 online;"  
}
```




3. 数据文件的备份与恢复

1) 备份命令:

```
RMAN> backup datafile 3;  
RMAN> backup datafile 'D:\ORACLE\ORADATA\TEST\TEST.DBF';
```

2) 恢复命令:

数据文件恢复与表空间恢复类似。假设数据文件号为5 的文件丢失，文件名是：
'E:\ORACLE\ORADATA\USERS.DBF'， 那么我们恢复的时候可以指定文件号，也可以指定文件名。

```
RMAN> run {  
allocate channel dev1 type disk;  
sql "alter tablespace users offline immediate";  
restore datafile 4; --或者 restore 'E:\ORACLE\ORADATA\USERS.DBF'  
recover datafile 4;  
sql "alter tablespace users online";  
release channel dev1;  
}
```

恢复到一个不同的位置:

```
$ rman target /  
RMAN> startup mount  
RMAN> RUN{  
sql "alter tablespace users offline immediate";  
SET NEWNAME for datafile 9 to '/xx/user01.dbf';  
restore datafile 9;  
switch datafile all;  
recover datafile 9;  
sql "alter tablespace users online";  
}
```



4. 归档重做日志的备份与恢复

1) 备份:

整库备份的同时，备份所有归档 (以及联机日志):

```
RMAN> backup database plus archivelog;
```

备份所有归档:

```
RMAN> backup archivelog all;
```

备份两天来的归档:

```
RMAN> backup archivelog from time='sysdate-2' [to time=' xxx' ];
```

备份从sequence 1 开始的归档:

```
RMAN> backup archivelog from sequence 1 [to sequence = ' n];
```

备份没有三次备份的归档:

```
RMAN> backup archivelog not backed up 3 times;
```

备份所有归档，然后删除归档:

```
RMAN> backup archivelog all delete input;
```

2) 恢复:

显示恢复目录中的归档日志:

```
RMAN> list backupset of archivelog all;
```

一般情况下，在RMAN 的普通恢复过程中，不必恢复归档的重做日志。不过偶尔也需要恢复重做日志，例如我们用Log Miner 来从归档中查找一些东西。

RMAN命令举例:

```
RMAN> RESTORE ARCHIVELOG ALL;  
RMAN> RESTORE ARCHIVELOG FROM LOGSEQ=1 UNTIL LOGSEQ=20;  
RMAN> RESTORE ARCHIVELOG FROM LOGSEQ=1;
```

可以用SET 命令来指定归档日志的还原位置，例如:

```
RMAN> run{  
  set archivelog destination to "d:\temp";  
  restore archivelog all;  
}
```

需要注意的是，即使新的归档日志目录不同于默认的归档日志目录，如果Oracle 判定日志已存在，也不会恢复该归档日志文件。



5. 联机日志的备份

联机日志不能用RMAN 来备份，可以先将其归档，再备份。为了实现这点，必须在RMAN 中执行归档命令语句：

```
RMAN> run {
allocate channel dev1 type disk;
sql "alter system archive log current";
backup (archivelog from time "sysdate-1" all delete input)
"format '/oracle/backups/log_t%t_s%s_p%p";
release channel dev1;
}
```

上面的脚本可以在执行完一个完整的联机数据库备份后执行，确保所有的重做日志可以将数据库恢复到一个一致性的状态。

当然，也可以在全库备份时使用 **plus archivelog** 选项，将自动完成联机日志的备份。

6. 控制文件和服务参数文件的备份与恢复

1) 备份：

// 设置文件名格式

```
RMAN> set controlfile autobackup format for device type disk to 'ctl_%F';
```

1. RMAN> configure controlfile **autobackup on**; // backup database时将自动备份

2. RMAN> backup current controlfile;

3. RMAN> backup **include control file**;

4. RMAN> backup file 1; // system datafile 自动备份

2) 恢复 SPFILE：

SPFILE（PFILE）的丢失/损坏，对数据库不会产生致命影响，可以从其他方式恢复。

不过既然RMAN 的备份计划中包括了SPFILE 的备份，那么就可以使用RMAN 来还原SPFILE 了。

```
$ rman target / catalog "rman/rman@db"
```

```
RMAN> set dbid=153910023 // SET DBID 这个步骤是不能省略的，否则会报错。
```

```
RMAN> restore spfile from autobackup [MAXDAYS 100]; // 或者 restore spfile;
```

```
RMAN> startup force
```

或者从某个备份集恢复：

```
RMAN> restore spfile from backupset bs_num 命令。
```



使用 `dbms_backup_restore` 包恢复服务器参数文件:

在一些不常见的情况下, 我们可能需要直接使用 `dbms_backup_restore` 包来恢复 `spfile`。

当然这个包也可以用来恢复其它数据, 是常规办法都没有用的时候的一个利器。

这个包可以在数据库 `NOMOUNT` 状态下使用。假设我们有一个自动备份文件

`C-2600315304-20060829-02`, 我们需要从这里恢复数据, 那么可以通过执行下面的脚本来完成:

```
DECLARE
DEVTYPE VARCHAR2(256);
DONE BOOLEAN;
BEGIN
DEVTYPE:=DBMS_BACKUP_RESTORE.deviceallocate(NULL);
DBMS_BACKUP_RESTORE.restoresetdatafile;
DBMS_BACKUP_RESTORE.restorespfileto('/back/SPFILE.ORA');
DBMS_BACKUP_RESTORE.restorebackuppiece(
'/back/C-2600315304-20060829-02',DONE=>done);
DBMS_BACKUP_RESTORE.devicedeallocate(NULL);
END;
/
```

3) 恢复控制文件:

```
RMAN> startup nomount;
RMAN> set dbid=153910023
RMAN> restore controlfile from autobackup
```

或

```
RMAN> restore controlfile from '/arch/ct_c-2347671489-20060630-00';
```

联机状态: 目标数据库 `MOUNT` 或 `OPEN`

```
RMAN> restore controlfile to 'd:\temp\control01.ctl';
```

然后再执行恢复数据库的其他步骤:

```
RMAN> restore database;
RMAN> recover database;
RMAN> alter database open resetlogs;
```



使用 `dbms_backup_restore`包恢复控制文件：

```
DECLARE
DEVTYPE VARCHAR2(256);
DONE BOOLEAN;
BEGIN
DEVTYPE:=DBMS_BACKUP_RESTORE.deviceallocate(NULL);
DBMS_BACKUP_RESTORE.resetdatafile;
DBMS_BACKUP_RESTORE.restorecontrolfileto('/back/CONTROL01.CTL');
DBMS_BACKUP_RESTORE.restorebackuppiece
('/back/C-2600315304-20060829-00',DONE=>done);
DBMS_BACKUP_RESTORE.deviceallocate(NULL);
END;
/
```

7. 备份集的备份的备份与恢复

1) 备份：

备份所有备份集：

```
RMAN> backup backupset all;
```

备份指定备份集：

```
RMAN> backup backupset bs_num;
```

2) 恢复：（这种备份只是增加一个镜像，不用恢复）

主要用于改变备份集的位置，或者创建多个镜像备份，比如将备份集从硬盘备份到磁带。



(四)、归档模式下的不完全恢复

不完全恢复就意味着有数据的丢失。引起不完全恢复的原因有很多，如丢失了联机日志或某个归档日志。

另外如果出现了严重损害数据库的用户错误，比如某用户错误的删除了某个重要的数据，那么数据库也要恢复到这个错误操作之前。

不完全恢复会影响整个数据库，需要在MOUNT状态下进行。在不完全恢复完成之后，通常需要使用resetlogs 选项来打开数据库。resetlogs 表示一个数据库逻辑生存期的结束和另一个数据库逻辑生存期的开始。数据库的逻辑生存期也被称为一个对应物(incarnation)。

每次使用resetlogs 选项来打开数据库后都会创建一个新的数据库对应物，这对于恢复操作来说非常重要。每次使用resetlogs 后，SCN 计数器不会被重置，但是Oracle会重置联机日志序列号，同时还会重置联机重做日志内容。因此执行了resetlogs 之后，应该立即重新备份整个数据库，否则恢复起来相当麻烦。（注:Oracle 10g中已经可以在resetlogs 之后不备份数据库，恢复的时候能够穿越resetlogs）

使用 RMAN 执行不完全恢复操作时需要完成的一个工作是建立**恢复目标**。恢复目标是要终止恢复进程的点，可以是时间点、指定的SCN 或者一个日志序列号。我们可以在run代码中使用**set 命令**和until time、until scn、until sequence参数。

也可以选择在RESTORE和RECOVER命令中直接使用UNTILTIME、UNTIL SCN、或者UNTIL SEQUENCE参数，这样就可以避免使用run 代码。例如：

```
startup mount;
restore database "to_date('2005/08/01 13:00:00','yyyy/mm/dd hh24:mi:ss)";
recover database "to_date('2005/08/01 13:00:00','yyyy/mm/dd hh24:mi:ss)";
alter database open resetlogs;
```

1. 基于 SCN 的恢复

如果知道数据库出错前的SCN，可以将数据库还原到指定SCN 状态。

```
$ startup mount;
RMAN> run{
allocate channel d1 type disk;
restore database until scn 1317011; --或者 set until scn 1317011
recover database until scn 1317011;
sql 'alter database open resetlogs';
release channel d1;
}
```



2. 基于时间的恢复

下面使用set until time 命令为2005 年8 月1 日下午 1点的恢复目标:

```
$ startup mount;
RMAN> run{
set until time "to_date('2005/08/01 13:00:00','yyyy/mm/dd hh24:mi:ss')";
restore database;
recover database;
alter database open resetlogs;
}
```

执行上面的命令时， RMAN 会查找与恢复目标时间最近、但是不包含恢复目标时间及以后时间的备份集，并且从这个备份中还原数据库。如果数据库非归档模式，那么恢复操作会在备份集的时间点停止；否则RECOVER 命令会应用恢复目标之前的归档重做日志或需要的增量备份。

3. 基于日志序列的恢复

RMAN 允许用户将数据库恢复到指定的归档重做日志序列号。如果归档的重做日志中有间隙（某个归档日志文件或备份损坏或丢失），使用这种方法很方便。间隙通常意味着我们只能将数据库还原到间隙开始的地方。

```
SQL> startup mount;
RMAN> restore database until sequence 100 thread 1; --not include 100
RMAN> recover database until sequence 100 thread 1;
SQL> alter database open resetlogs;
RMAN> RUN {
SET UNTIL SEQUENCE 120 THREAD 1;
RESTORE DATABASE;
RECOVER DATABASE; --recovers through log 119 not include 120
ALTER DATABASE OPEN RESESTLOGS;
}
```




七、关于 CROSSCHECK 交叉校验命令

该命令用于核对磁盘和磁带上的备份文件，以确保RMAN 资料库与备份文件保持同步。

该命令只会检查RMAN 资料库所记载的备份文件。

当执行CROSSCHECK命令时，如果资料库记录不匹配于备份文件的物理状态，那么该命令会更新资料库记录的状态信息。

当使用CROSSCHECK 命令核对备份文件之后，备份文件的状态会包括AVAILABLE、UNAVAILABLE 和EXPIRED 三种，如果备份文件处于EXPIRED 状态，则说明物理文件已经被手工删除或者损坏。

注意，如果备份文件处于EXPIRED 状态，应该使用DELETE 命令删除该备份文件。

1. 核对所有备份集

```
RMAN>CROSSCHECK BACKUP;
```

2. 核对所有数据文件的备份集

```
RMAN> CROSSCHECK BACKUP OF DATABASE;
```

3. 核对特定表空间的备份集

```
RMAN>CROSSCHECK BACKUP OF TABLESPACE SYSTEM;
```

4. 核对特定数据文件的备份集

```
RMAN>CROSSCHECK BACKUP OF DATAFILE 4;
```

5. 核对控制文件的备份集

```
RMAN>CROSSCHECK BACKUP OF CONTROLFILE;
```

6. 核对 SPFILE 的备份集

```
RMAN> CROSSCHECK BACKUP OF SPFILE;
```

7. 核对归档日志的备份集

```
RMAN> CROSSCHECK BACKUP OFARCHIVELOG SEQUENCE 3;
```

8. 核对所有映像副本

```
RMAN> CROSSCHECK COPY;
```



9. 核对所有数据文件的映像副本

```
RMAN> CROSSCHECK COPY OF DATABASE;
```

10. 核对特定表空间的映像副本

```
RMAN> CROSSCHECK COPY OF TABLESPACE USERS;
```

11. 核对特定数据文件的映像副本

```
RMAN> CROSSCHECK COPY OF DATAFILE 4;
```

12. 核对控制文件的映像副本

```
RMAN> CROSSCHECK COPY OF CONTROLFILE;
```

13. 核对归档日志的映像副本

```
RMAN> CROSSCHECK COPY OF ARCHIVELOG SEQUENCE 4;
```

八、修改备份的可用状态、保存策略 Change 命令

Change 命令可以修改备份的状态是可用（available）还是不可用（unavailable）。

对于不可用的备份，还原与恢复期间不会被考虑到，不过执行delete expired 命令期间也不会删除这些记录。

例如：

```
RMAN> change backup of database tag='GOLD' unavailable;  
RMAN> change copy of database like '%GOLD%' available;  
RMAN> change archivelog 'd:\arc\arch_001.arc' unavailable;  
RMAN> change backupset 4981 available;  
RMAN> change backup of database available;  
RMAN> change archivelog all available;  
RMAN> change archivelog all backed up 5 times unavailable;
```

当一个给定的备份或者副本根据备份的保存策略的标准而被废弃的时候，RMAN并不会自动删除这个备份或者副本，而只是标记这个备份为废弃。我们可以用reportobsolete 命令来查看标记为废弃的备份。可以使用change 命令来将一个备份修改为永久保留的备份，也可以修改为要保存多少天的备份。还可以使用change ... nokeep 来手工丢弃一个备份。



举例：

将4421 备份集标记为废弃

```
RMAN> change backupset 4421 nokeep;
```

将4421 备份集标记为7 天内有效

```
RMAN> change backupset 4421 keep until time 'sysdate+7' logs;
```

将4421 备份集标记为永久有效

```
RMAN> change backupset 4421 keep forever logs;
```

废弃的备份集并不真正的删除，如果需要物理删除，则可以用下面的命令：

```
RMAN> delete obsolete;
```

九、查看与删除过时的备份信息

列出已经过时的备份：

```
RMAN> report obsolete;
```

定义delete通道：

```
RMAN> allocate channel for delete/maintenance type disk;
```

删除过时的备份信息

```
RMAN> allocate channel for maintenance type disk;  
RMAN> change backupset id delete;  
RMAN> release channel;
```

当手工删除了数据库的RMAN备份文件后，要执行以下脚本进行同步：

```
RMAN> allocate channel for maintenance type disk;  
RMAN> crosscheck backup;  
RMAN> delete expired backup; -- 删除过期的备份  
RMAN> delete obsolete; -- 删除废弃的备份  
RMAN> release channel;
```

在8i 和8i 之前的版本只能用change...delete 命令来删除物理备份。

```
RMAN> change archivelog until logseq=500 delete;
```



十、RMAN 脚本

1. 创建脚本

```
RMAN>create script b
{
  backup incremental level 0
  format 'c:\temp\10_%U.bak' database;
  sql 'alter system archive log current';
}
```

2. 执行脚本

```
run{execute script b;}
```

3. Print 脚本

```
print script b;
```

4. 修改脚本

```
RMAN>replace script b
{
  backup database format 'c:\temp\11_%U.bak';
  sql 'alter system archive log current';
}
```

5. 删除脚本

```
delete script b;
```

6. 列出脚本

```
list script names;
```

7. 脚本导出到文件

```
print script b to file 'c:\a.txt';
```

8. 文件导入成脚本

```
create script a from file 'c:\a.txt';
```

9. 通过文件修改脚本

```
replace script b from file 'c:\a.txt';
```



十一、常用脚本

1. 全备脚本

备份策略：fullback.sh

```
export ORACLE_BASE=/home/oracle
export ORACLE_HOME=/home/oracle/product/10.2.0/db_1
export ORACLE_SID=orcl
rq=`date +"20%y%m%d%H%M%S"`

//抓屏，抓取日志文件，文件名
rman target / log=/home/oracle/bak/backupall_${rq}.log<<EOF
run
{
//定义通道
allocate channel ch1 device type disk;
allocate channel ch2 device type disk;
//切换日志并全备数据库
sql 'alter system archive log current';
backup database format '/opt/oracle/bak/db_%d_%T_%U';
sql 'alter system archive log current';
//备份所有日志文件并删除输入
backup archivelog all format '/opt/oracle/bak/arch_%t_%s' delete all input;
//备份当前控制文件
backup format '/opt/oracle/bak/con_%s_%p' current controlfile;
//效验是否过期、失效（损坏、物理丢失）
crosscheck backup;
crosscheck archivelog all;
//删除失效的（逻辑删除）
delete noprompt expired backup;
//删除过期的,noprompt 为不提示是否删除
delete noprompt obsolete;
//物理删除过期的
delete noprompt backup of database completed before 'sysdate -15';
delete noprompt archivelog all;
delete noprompt backup of archivelog all completed before 'sysdate -15';
release channel ch1;
release channel ch2;
}
EOF
```

