



Advanced Oracle Document

CONVERT to ASM and Non-ASM

Author: Hyun-Ho, Jung
Site: <http://www.commit.co.kr>
Email: admin@commit.co.kr
cleanto@naver.com
Creation Date: 2011 - 12 - 13

CONVERT to ASM and Non-ASM

1. TEST Information

Items	Description
Test Date	2011 / 12 / 13
CPU	VirtualBox VCPUx4
Main Memory	2GB
O/S version	RHEL 5.5
Host Name	test1, test2
ORACLE_SID	TESTDB1, TESTDB2
Oracle version	10.2.0.5

2. Scenario&Environment

- 1) 10gR2(10.2.0.5) 버전으로 Clusterware 와 S/W는 설치가 되어 있는 상태이다.
- 2) Non-ASM 파일과 ASM 파일간 Convert 를 진행 할 것이다.
- 3) Single Instance에서 RAC 변환 할 때 Target DB(RAC) 의 Storage Type이 ASM 을 사용 할 때 해당 문서를 적용 할 수 있으며, 그 외 다른 요건에 의해서도 convert 를 사용 할 수 있다
- 4) 테스트 환경에서는 Non-ASM 파일은 CFS(Redhat GFS2) 영역에 존재한다.
- 5) ASM의 DiskGroup 명은 DATA 이다.
- 6) 11gR1 이상 사용 중이라면 해당 방법 외 11g NF인 ASM CP 기능을 이용 할 수 있다.

3. Non-ASM File To ASM

■ Convert 대상 확인

온라인 상으로 작업 하기 위해 begin backup 명령을 실행 하고 convert 를 하게 되면 에러가 발생 된다. 그러므로 Instance를 shutdown 후 진행 해야 한다.
shutdown 하기 전 데이터 파일 조회를 한다.

```
SQL> select tablespace_name, file_name from dba_data_files;
```

TABLESPACE_NAME	FILE_NAME
USERS	/oradata/testdb/users01.dbf
UNDOTBS2	/oradata/testdb/undotbs02.dbf
SYSAUX	/oradata/testdb/sysaux01.dbf
UNDOTBS1	/oradata/testdb/undotbs01.dbf
SYSTEM	/oradata/testdb/system01.dbf

위의 파일이 Convert 대상이 된다. temp 는 convert가 안된다. 필요하다면 생성해서 사용한다.

■ shutdown -> mount

shutdown 후 mount 단계로 open 한다.

■ RMAN을 이용해 convert 진행

rman으로 접속한다.

```
$ rman target /
```

```
Recovery Manager: Release 10.2.0.5.0 - Production on Tue Dec 13 21:34:30 2011
```

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

```
connected to target database: TESTDB (DBID=2551595590)
```

커맨드 문법은 아래와 같다

```
convert datafile '/위치/파일명.dbf' format '+DATA';
```

특별히 이름을 지정 하지 않으면 OMF 규칙에 따라 이름이 지정되며, 이름을 지정하고자 한다면 아래와 같이 경로와 파일명까지 지정한다.

```
convert datafile '/oradata/testdb/users01.dbf' format '+DATA/TESTDB/DATAFILE/users01.dbf';
```

여러 파일을 동시에 진행하기 위해서는 아래와 같이 진행 할 수 있다.

```
RMAN> convert datafile
2> '/oradata/testdb/system01.dbf',
3> '/oradata/testdb/sysaux01.dbf',
4> '/oradata/testdb/undotbs01.dbf',
5> '/oradata/testdb/undotbs02.dbf',
6> '/oradata/testdb/users01.dbf'
7> format '+data';
```

위와 같이 작업하면 파일의 이름을 원하는 데로 지정 할 수는 없게 된다.

convert 명령어를 통해 convert하였다면 asmcmd로 들어가서 확인 해본다.
필자는 datafile rename 을 위해 이름을 지정 하여 convert 하였다.

데이터 파일이 많다면 아래 쿼리를 응용 하여 스크립트를 추출한다.

```
set lines 500
set heading off
col file_name for a40
select 'convert datafile ''' || file_name || ''' format ''' || '+DATA/TESTDB/DATAFILE' ||
SUBSTR(file_name,16,15) || ''';' from dba_data_files;
```

■ rename file 후 open 한다.

```
alter database rename file '/oradata/testdb/users01.dbf' to '+DATA/TESTDB/DATAFILE/users01.dbf';
alter database rename file '/oradata/testdb/undotbs02.dbf' to '+DATA/TESTDB/DATAFILE/undotbs02.dbf';
alter database rename file '/oradata/testdb/sysaux01.dbf' to '+DATA/TESTDB/DATAFILE/sysaux01.dbf';
alter database rename file '/oradata/testdb/undotbs01.dbf' to '+DATA/TESTDB/DATAFILE/undotbs01.dbf';
alter database rename file '/oradata/testdb/system01.dbf' to '+DATA/TESTDB/DATAFILE/system01.dbf';
```

위 커맨드는 아래쿼리를 이용해서 추출 하였다.

```
set lines 500
set heading off
col file_name for a40
select 'alter database rename file ''' || file_name || ''' to ''' || '+DATA/TESTDB/DATAFILE' ||
SUBSTR(file_name,16,15) || ''';' from dba_data_files;
```

open 후 Instance 상태 및 datafile 위치를 조회 해본다.

```
SQL> select status from v$instance;
```

```
STATUS
-----
OPEN
```

```
SQL> select file_name from dba_data_files;
```

```
FILE_NAME
-----
+DATA/testdb/datafile/users01.dbf
+DATA/testdb/datafile/undotbs02.dbf
+DATA/testdb/datafile/sysaux01.dbf
+DATA/testdb/datafile/undotbs01.dbf
+DATA/testdb/datafile/system01.dbf
```

4. ASM To Non-ASM File

■ Users Tablespace offline

위에서 ASM파일로 변환한 users01.dbf 파일을 다시 Non-ASM File 로 변환 하겠다.

테스트의 편의성을 위해 users 테이블스페이스를 offline 시킨 후 진행 하겠다.

```
SQL> alter tablespace users offline;
```

■ RMAN에 접속한다.

rman 으로 접속한다.

```
$ rman target /
```

Recovery Manager: Release 10.2.0.5.0 - Production on Tue Dec 13 23:20:01 2011

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

connected to target database: TESTDB (DBID=2551595590)

■ RMAN을 이용해 convert 진행

```
convert datafile '+DATA/testdb/datafile/users01.dbf' format '/oradata/testdb/convert/users01.dbf';
```

■ rename datafile 후 users Tablespace online

```
alter database rename file '+DATA/testdb/datafile/users01.dbf' to '/oradata/testdb/convert/users01.dbf';
```

rename 후 users Tablespace 를 online 시킨다.

```
SQL> alter tablespace users online;
```

테이블스페이스와 데이터 파일의 상태를 조회한다.

```
SQL> select tablespace_name, file_name, status from dba_data_files;
```

TABLESPACE_NAME	FILE_NAME	STATUS
USERS	/oradata/testdb/convert/users01.dbf	AVAILABLE
UNDOTBS2	+DATA/testdb/datafile/undotbs02.dbf	AVAILABLE
SYSAUX	+DATA/testdb/datafile/sysaux01.dbf	AVAILABLE
UNDOTBS1	+DATA/testdb/datafile/undotbs01.dbf	AVAILABLE
SYSTEM	+DATA/testdb/datafile/system01.dbf	AVAILABLE

위치가 변경 되었으며, 상태도 AVAILABLE 인 것을 확인 할 수 있다.

5. Not Exists Instance

■ Instance 가 존재 하지 않을 때

위의 테스트는 Database Instance 가 존재 할 때를 기반으로 테스트가 진행 되었다.

가령 Single 에서 RAC로 변환 하기 위해서 Clusterware 설치 -> S/W(엔진) 설치 -> ASM인스턴스 생성 순으로 진행하여 Database 가 존재 하지 않고 기존 Single Instance의 Datafile 을 이용해야 할 때는 어떻게 해야 할까?

single 에서 사용하던 datafile 은 /oradata2/ha 위치에 ha 이라는 SID 로 존재 한다. RAC 변환 하면서 COPYDB 로 SID도 변경 하겠다.

■ 파라미터 파일 작성

nomount 단계로 open 하기 위해서는 파라미터 파일이 존재 해야 한다

RAC로 변환 및 ASM convert 하기 위해서 몇가지를 수정 추가 한다.

```
SQL> create pfile='/oracle/init_ha.ora' from spfile;
```

```
*.control_files='+DATA/COPYDB/DATAFILE/control01.ctl','+DATA/COPYDB/DATAFILE/control02.ctl'
```

컨트롤 파일을 재생성을 해야 하며, 재생성을 통해서 ASM 영역에 생성을 한다.

그렇기 때문에 컨트롤위치는 ASM영역으로 지정한다

cluster 관련 파라미터를 추가한다

```
*.cluster_database_instances = 2
```

```
*.undo_management=AUTO
```

```
COPYDB1.undo_tablespace='UNDOTBS1'
```

```
COPYDB1.instance_number=1
```

```
COPYDB1.thread=1
```

```
COPYDB2.instance_number=2
```

```
COPYDB2.thread=2
```

```
COPYDB2.undo_tablespace='UNDOTBS2'
```

```
*.cluster_database = false
```

dump_dest를 변경한다.

```
*.audit_file_dest='/oracle/admin/COPYDB/adump'
```

```
*.background_dump_dest='/oracle/admin/COPYDB/bdump'  
*.core_dump_dest='/oracle/admin/COPYDB/cdump'  
*.user_dump_dest='/oracle/admin/COPYDB/udump'
```

해당 위치에 디렉토리를 생성해 준다.

db_name 변경

```
*.db_name='COPYDB'
```

편집한 컨트롤 파일을 \$ORACLE_HOME/dbs/initCOPYDB1.ora 이름으로 복사 한다.

■ 컨트롤 파일 재생성

rman convert 를 이용 할 수 없는 파일은 컨트롤 파일, redolog file , temp file 이다.
그렇기 때문에 컨트롤 파일을 재생성 하면서 ASM 영역에 생성을 할 것이다.

컨트롤 파일을 TRACE 형태로 생성 한다.

```
SQL> alter database backup controlfile to trace as '/oracle/recon.sql';
```

전)

```
CREATE CONTROLFILE REUSE DATABASE "HA" NORESETLOGS ARCHIVELOG
```

후)

```
CREATE CONTROLFILE SET DATABASE "COPYDB" RESETLOGS NOARCHIVELOG
```

위에서 변경한 이유는 테스트상 SID와 디렉토리 경로를 겹치는 부분이 있어서 SID를 변경한 것이며, 기존 SID를 사용 할 것이라면 REUSE / NORESETLOGS 를 사용 해도 된다.

SINGLE DB는 현재 archivelog Mode 중이며, 테스트의 편의성을 위해서 변환되는 RAC는 No archivelog Mode 로 구성하기 위해서 NOARCHIVELOG 로 기술 한 것이기 때문에 꼭 NOARCHIVELOG 로 해야 하는 것은 아님을 밝혀둔다.

SID 의 변경 유무와 상관없이 ASM영역내로 재생성 해야 하기 때문에 반드시 재생성을 한다.

■ asmcmd 로 접속해 디렉토리를 생성한다.

```
$ export ORACLE_SID=+ASM1
```

```
$ asmcmd
```

```
ASMCMD> cd DATA
```

```
ASMCMD> mkdir COPYDB
```

```
ASMCMD> mkdir COPYDB/DATAFILE
```

■ 기존 Single Database shutdown 후 컨트롤 파일 재생성

single 로 운영중인 ha 인스턴스를 종료한다.

```
SQL> shutdown immediate;
```

sid를 변경 후 컨트롤 파일 재생성을 진행한다

```
$ export ORACLE_SID=COPYDB1
```

```
SQL> @recon.sql
```

■ RMAN 을 이용해서 convert

rman 에 접속 후 아래와 같이 convert 명령을 내리면

```
convert datafile '/oradata2/ha/users01.dbf' format '+DATA/COPYDB/DATAFILE/users01.dbf';
```

아래와 같이 에러가 발생 한다.

```
Starting backup at 14-DEC-11
```

```
using channel ORA_DISK_1
```

```
RMAN-00571: =====
```

```
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
```

```
RMAN-00571: =====
```

```
RMAN-03002: failure of backup command at 12/14/2011 01:56:34
```

```
RMAN-06592: must specify FROM PLATFORM
```

에러 메시지에서 힌트를 얻을 수 있는데 FROM PLATFORM 을 기술해 달라고 한다.

해당 정보는 공통적인 정보로써 open 된 인스턴스에서 아래 쿼리를 실행해 조회 해보면 된다.

```
SELECT * FROM V$TRANSPORTABLE_PLATFORM;
```

결과는 아래와 같다.

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
7	Microsoft Windows IA (32-bit)	Little
10	Linux IA (32-bit)	Little
6	AIX-Based Systems (64-bit)	Big
3	HP-UX (64-bit)	Big
5	HP Tru64 UNIX	Little
4	HP-UX IA (64-bit)	Big
11	Linux IA (64-bit)	Little
15	HP Open VMS	Little
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
13	Linux x86 64-bit	Little
16	Apple Mac OS	Big
12	Microsoft Windows x86 64-bit	Little
17	Solaris Operating System (x86)	Little
18	IBM Power Based Linux	Big
19	HP IA Open VMS	Little
20	Solaris Operating System (x86-64)	Little

OS 별 각기 다른 ENDIAN_FORMAT 정보를 담고 있는 뷰를 조회 한 것이다.

RMAN에서 FROM PLATFORM 에서 기술해달라는 플랫폼명은 위에서 참조 해서 기재 하면 된다.

```
convert datafile '/oradata2/ha/users01.dbf' format '+DATA/COPYDB/DATAFILE/users01.dbf';
```

FROM 플랫폼은 위의 커맨드에서 빨간색으로 되어있는 Source 부분을 지칭 하는 것이다.

가령 users01.dbf 파일이 원래 윈도우에서 사용중인 파일이고 해당파일을 Linux의 ASM 영역에 convert 할 것이라면 아래와 같이 source OS의 정보를 입력해 주면 된다.

```
FROM PLATFORM 'Microsoft Windows x86 64-bit'
```

테스트 환경은 리눅스 환경임으로 커맨드는 아래와 같이 된다.

```
convert datafile '/oradata2/ha/users01.dbf'
```

```
FROM PLATFORM 'Linux x86 64-bit'
```

```
format '+DATA/COPYDB/DATAFILE/users01.dbf';
```

FROM절 까지 기술 한 후 실행 하면 아래와 같이 정상적으로 복사 되는 것을 확인 할 수 있다.

같은 방법으로 system, sysaux,undo 등 모든 데이터 파일을 convert 한다.

위에서 언급했듯이 temp,redo 를 제외하고 모든 데이터 파일을 convert 한다.

■ 데이터 파일 경로를 변경 후 open 한다.

datafile 은 아래처럼 변경을 한다.

```
alter database rename file '/oradata2/ha/undotbs02.dbf' to '+DATA/COPYDB/DATAFILE/undotbs02.dbf';
```

```
alter database rename file '/oradata2/ha/sysaux01.dbf' to '+DATA/COPYDB/DATAFILE/sysaux01.dbf';
```

```
alter database rename file '/oradata2/ha/undotbs01.dbf' to '+DATA/COPYDB/DATAFILE/undotbs01.dbf';
```

```
alter database rename file '/oradata2/ha/system01.dbf' to '+DATA/COPYDB/DATAFILE/system01.dbf';
```

```
alter database rename file '/oradata2/ha/users01.dbf' to '+DATA/COPYDB/DATAFILE/users01.dbf';
```

recover 진행

```
SQL> recover database using backup controlfile;
```

```
ORA-00279: change 721438 generated at 12/16/2011 01:36:24 needed for thread 1
```

```
ORA-00289: suggestion : /oracle/product/102/db/dbs/archarc_1_3_769925721.arc
```

```
ORA-00280: change 721438 for thread 1 is in sequence #3
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

위에서 요구하는 변경본이 있는 파일은 redo log 파일 일 것임으로 현재 리두로그 파일을 적용 해본다.

아래와 같이 나온다면 다음 리두를, 그래도 나오면 다른 리두 로그를 입력 한다.

```
ORA-00310: archived log contains sequence 2; sequence 3 required
```

```
ORA-00334: archived log: '/oradata2/ha/redo02.log'
```

맞는 리두로그 파일을 입력하면 아래와 같이 복구가 완료 되었다는 메시지를 볼수있다.

```
Log applied.
```

```
Media recovery complete.
```

resetlogs 로 open 한다.

```
SQL> alter database open resetlogs;
```

■ tempfile 추가

```
ALTER TABLESPACE TEMP ADD TEMPFILE '+DATA/COPYDB/DATAFILE/temp01.dbf'  
    SIZE 1024M REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;
```

■ redo로그 파일 추가 삭제

group1 과 2에 멤버를 각각 하나씩 추가한다.

```
alter database add logfile member '+DATA/COPYDB/DATAFILE/redo01a.log' to group 1;  
alter database add logfile member '+DATA/COPYDB/DATAFILE/redo02a.log' to group 2;
```

redolog file 은 아래쿼리로 조회 해서 active, inactive 상태를 조회 한다.

```
SQL>set linesize 500  
SQL> col member format a35  
SQL> col IS_RECOVERY_DEST_FILE format a25  
SQL> select b.THREAD# ,a.group#,b.FIRST_CHANGE#, a.member, b.bytes/1024/1024 MB,b.status,  
2 a.type TYPE  
3 from v$logfile a, v$log b  
4 where a.group# = b.group#  
5 order by a.group#, a.member;
```

THREAD#	GROUP#	FIRST_CHANGE#	MEMBER	MB	STATUS	TYPE
1	1	721440	+DATA/copydb/datafile/redo01a.log	200	CURRENT	ONLINE
1	1	721440	/oradata2/ha/redo01.log	200	CURRENT	ONLINE
1	2	0	+DATA/copydb/datafile/redo02a.log	200	UNUSED	ONLINE
1	2	0	/oradata2/ha/redo02.log	200	UNUSED	ONLINE
1	3	0	/oradata2/ha/redo03.log	200	UNUSED	ONLINE

현재 group3 이 unused 중이라 바로 삭제가 가능하기 때문에 삭제를 한다.

```
alter database drop logfile group 3;
```

2번 인스턴스용(Thread2) Redolog 를 생성한다.

```
SQL> alter database add logfile thread 2 group 3 ('+DATA/COPYDB/DATAFILE/redo03a.dbf') size 100M,  
group 4 ('+DATA/COPYDB/DATAFILE/redo04a.dbf') size 100M;
```

활성화 한다.

```
SQL> alter database enable public thread 2;
```

다시 위의 쿼리로 조회하면서 alter system switch logfile 과 alter system checkpoint 를 이용하여 group 1 과 group 2를 inactive 로 만든다. ASM영역이 아닌 기존의 filesystem 에 있는 redo log 파일을 삭제 해야 한다.

redo 로그 상태를 조회해서 inactive 가 된 그룹의 멤버를 삭제 한다.

```
alter database drop logfile member '/oradata2/ha/redo01.log';
alter database drop logfile member '/oradata2/ha/redo02.log';
```

이렇게 추가 삭제를 반복하면 아래와 같이 4개의 그룹 파일이 ASM에 위치 하게 된다.

THREAD#	GROUP#	FIRST_CHANGE#	MEMBER	MB STATUS	TYPE
1	1	721716	+DATA/copydb/datafile/redo01a.log	200 INACTIVE	ONLINE
1	2	721766	+DATA/copydb/datafile/redo02a.log	200 CURRENT	ONLINE
2	3	721695	+DATA/copydb/datafile/redo03a.dbf	100 CURRENT	ONLINE
2	4	0	+DATA/copydb/datafile/redo04a.dbf	100 UNUSED	ONLINE

■ undotbs 를 생성

2번 인스턴스(THREAD2) 의 undotbs 를 생성(존재 한다면 생성하지 않아도 된다)

```
SQL> CREATE UNDO TABLESPACE UNDOTBS2 DATAFILE '+DATA/COPYDB/DATAFILE/undotbs02.dbf' size 1024M;
```

여기 까지 진행 했다면 모든 데이터파일(undo, temp포함) 이 ASM영역으로 위치해 있을 것이다.

아래 쿼리로 조회해보자

```
select file_name from dba_data_files
union all
select file_name from dba_temp_files;
```

```
+DATA/copydb/datafile/undotbs02.dbf
+DATA/copydb/datafile/users01.dbf
+DATA/copydb/datafile/sysaux01.dbf
+DATA/copydb/datafile/undotbs01.dbf
```

```
+DATA/copydb/datafile/system01.dbf
+DATA/copydb/datafile/temp01.dbf
```

컨트롤 파일을 조회 해본다.

```
SQL> show parameter control
```

```
control_files                string                +DATA/copydb/datafile/control0
                             1.ctl, +DATA/copydb/datafile/c
                             ontrol02.ctl
```

2 개의 controlfile 이 ASM 영역에 위치해 있다.

■ 파라미터 수정 후 RAC convert

다시 파라미터 파일을 열어서 아래 파라미터를 수정한다.

전)

```
*.cluster_database = false
```

후)

```
*.cluster_database = true
```

파라미터 파일을 수정 후 데이터 베이스를 재시작 한다. 인스턴스가 open 된 후 아래 sql을 수행한다.

```
SQL> @?/rdbms/admin/catclust.sql
```

■ spfile 생성

```
create spfile='+DATA/COPYDB/DATAFILE/spfileCOPYDB.ora' from
pfile='/oracle/product/102/db/dbs/initCOPYDB1.ora';
```

(한줄로 붙여서 실행)

spfile 을 만든 후 1번 서버에서는 initCOPYDB1.ora 을 rename(백업) 하고
1번에서 initCOPYDB1.ora , 2번서버에서 initCOPYDB2.ora 를 만든다

내용은 아래와 같이 SPFILE을 가리키도록 한다.

```
SPFILE='+DATA/COPYDB/DATAFILE/spfileCOPYDB.ora'
```

위에서 생성한 dump_dest 디렉토리를 2번 서버에서도 동일 하게 생성한다.

```
*.audit_file_dest='/oracle/admin/COPYDB/adump'  
*.background_dump_dest='/oracle/admin/COPYDB/bdump'  
*.core_dump_dest='/oracle/admin/COPYDB/cdump'  
*.user_dump_dest='/oracle/admin/COPYDB/udump'
```

■ clusterware의 Resource 로 등록한다.

srvctl 명령어를 이용하여 database를 등록한다. 형식은 아래와 같다.

```
$ srvctl add database -d COPYDB -o /oracle/product/102/db -p +DATA/COPYDB/DATAFILE/spfileCOPYDB.ora
```

각 옵션의 의미는 아래와 같다

-d : database의 이름
-o : oracle 홈 디렉토리
-p : 파라미터 파일 경로

srvctl 명령어를 이용하여 instance를 등록한다. 형식은 아래와 같다.

```
$ srvctl add instance -d COPYDB -i COPYDB1 -n test1  
$ srvctl add instance -d COPYDB -i COPYDB2 -n test2
```

-i : 인스턴스 명
-n : 각 노드의 호스트명

■ resource 확인

리소스 등록 까지 모두 마친 후 crs_stat -t 를 통해 리소스 상태를 확인 한다.

아래와 같이 database 및 instance 리소스가 OFFLINE 이라면 리소스 시작 커맨드를 실행한다.

```
ora...B1.inst application      OFFLINE  OFFLINE  
ora...B2.inst application      OFFLINE  OFFLINE  
ora.COPYDB.db application      OFFLINE  OFFLINE
```

```
$ crs_start -all
```

커맨드 실행 후 다시 확인 해보면 모두 ONLINE 상태로 변경 된것을 확인 할 수 있다.

```
ora...B1.inst application      ONLINE   ONLINE   test1  
ora...B2.inst application      ONLINE   ONLINE   test2  
ora.COPYDB.db application      ONLINE   ONLINE   test2
```