

Oracle 10g 이기종 Platform 간 Data Migration

작성자 : 신종근
일시 : 2007. 02.

- I. Introduction
- II. Linux ASM → Solaris raw-device
- III. Linux ASM → Solaris ASM
- IV. Linux raw-device → Solaris raw-device
- V. HP F/S → Solaris raw-device
- VI. HP ASM → Solaris ASM
- VII. Move ASM DBF → other Diskgroup
- VIII. 기타내용 및 QnA

I. Introduction

Oracle 10g Support Multi-Platform

```
SQL> COLUMN PLATFORM_NAME FORMAT A32  
SQL> SELECT * FROM V$TRANSPORTABLE_PLATFORM;
```

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 64-bit for AMD	Little
16	Apple Mac OS	Big
12	Microsoft Windows 64-bit for AMD	Little
17	Solaris Operating System (x86)	Little

** 동일 Endian 인 경우 이기종 Platform 간 RMAN Convert 는 필요없음 !!

Limitation on Transportable Tablespace

1. Same Character set and National Character set
2. Cannot transport Same Tablespace name
Rename before Transport Operation.
3. MV, Partitioned Table's Object are All included.
4. 10g R2 can transport XMLType with exp / imp not Data Pump
select distinct p.tablespace_name
from dba_tablespaces p, dba_xml_tables x, dba_users u, all_all_tables t
where t.table_name=x.table_name and t.tablespace_name=p.tablespace_name
and x.owner=u.username;
5. Advanced Queue Transportable Tablespace do not support 8.0-compatible
6. Cannot transport SYSTEM or objects owned by SYS
7. Opaque Type (RAW, BFILE, ...) can be transport only the Same Platform
8. Floating-Point number BINARY_FLOAT and BINARY_DOUBLE types are transportable
using Data Pump not Exp / Imp

Transportable Tablespace란 ?

1. When to use ?

- ← 8i 이후 대용량 Data 의 Data 이관시에 사용가능함.
- ← DB 복구시 Read-Only 로 강제 Open 된 경우, 신규 DB 생성후 TTS 로 복구함.

2. How to ?

- ← Transportable Tablespace 기법을 사용함.

3. Version 별 Limitation !!

- ← 8i :: 동일 OS , 동일 Oracle Version , 동일 DB_BLOCK_SIZE , 동일 Character set
- ← 9i :: 동일 OS , 동일 Oracle Version , 동일 Character set
- ← 10g :: 동일 Character set , 동일 Oracle Version

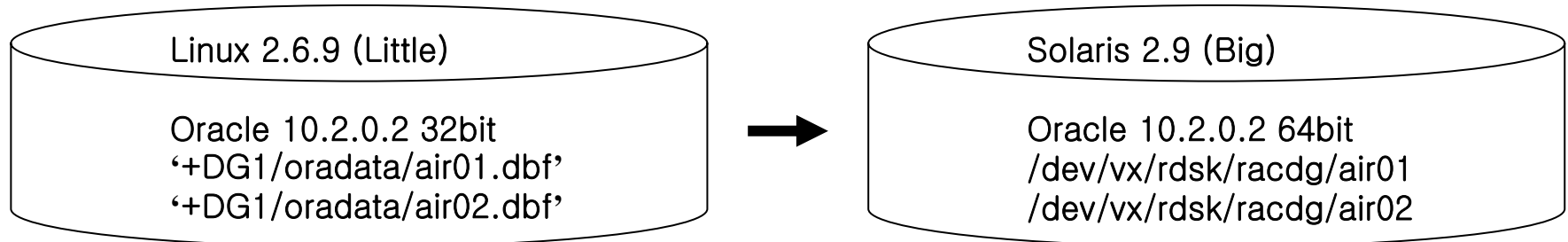
동일 Oracle Version 중 Release Version 은 낮은곳에서 높은곳으로 Imp 는 가능함.

I. Introduction

Transportable Tablespace 정리

구분	방법 정리	비고
1. Little → Big Linux ASM → Solaris raw	1.exp & RMAN Convert to F/S 2.dd or DBMS_FILE_TRANSFER to raw-device 3.Imp	
2. Little → Big Linux ASM → Solaris ASM	1.exp & RMAN Convert 2.DBMS_FILE_TRANSFER to F/S 3.Imp & RMAN 으로 ASM 으로 copy	
3. Little → Big Linux raw → Solaris raw	1.exp & RMAN Convert to F/S 2.dd or DBMS_FILE_TRANSFER to raw-device 3.Imp	
4. Big → Big HP F/S → Solaris raw	1.exp 2.dd or DBMS_FILE_TRANSFER to raw-device 3.Imp	
5. Big → Big HP ASM → Solaris ASM	1.exp 2.DBMS_FILE_TRANSFER to ASM 3.Imp	
6. 기타 Move ASM Diskgroup	1.DBMS_FILE_TRANSFER.COPY_FILE	

II. Linux ASM → Solaris raw-device



Endian 이 다르므로 RMAN으로 Conversion 한 후 DBMS_FILE_TRANSFER 로 이전함.

1. 준비작업

1) 유저 생성 – Target 인 Sun 에...

```
SUN> create user air identified by air;  
SUN> grant connect,resource to air;
```

2) Directory 생성 (Linux, Sun both)

```
Lnx> create or replace directory LNX_DG  
as '+DG1/oradata';
```

```
Lnx> grant write on directory LNX to air;
```

```
SUN> create or replace directory SUN  
as '/dev/vx/rdsk/racdg';
```

```
SUN> grant write on directory SUN to air;
```

3) DataLink 생성 (DBMS_FILE_TRANSFER 용)

```
HP> create public database link DB_LINK  
connect to air identified by air using 'SUN';
```

4) Sun 에 raw-device 생성

```
#vxassist -g racdg make air01 1200m layout=st  
ripe stripeunit=32k disk1 disk2 disk3
```

```
#vxassist -g racdg make air02 2000m layout=st  
ripe stripeunit=32k disk1 disk2 disk3
```

```
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air01
```

```
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air02
```

2. Transportable Check

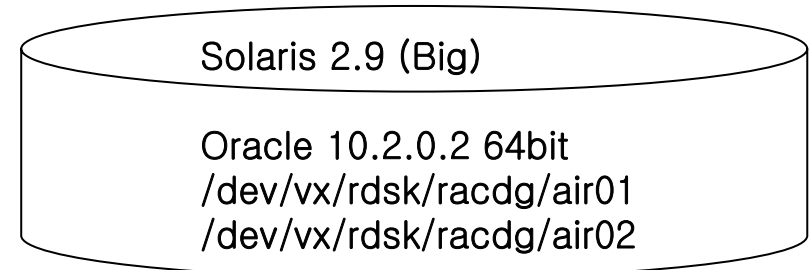
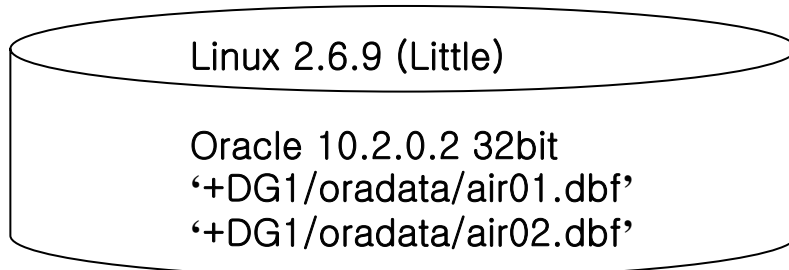
```
HP> exec
```

```
dbms_tts.transport_set_check('AIR',TRUE)
```

```
HP> select * from sys.transport_set_violations;
```

```
HP> alter tablespace air read only;
```

II. Linux ASM → Solaris raw-device



3. Exp only Meta

```
% exp userid=W'sys/welcom as sysdbaW'  
file=air.dmp log=air.log W  
transport_tablespace=y tablespaces=air
```

4. RMAN Convert to Solaris

```
% rman target=/ nocatalog  
RMAN> CONVERT TABLESPACE air  
TO PLATFORM 'Solaris[tm] OE (64-bit)'  
FORMAT '/fw1/oracle10/WORK/%U';
```

5. Data Copy

← 동시 실행 !!

```
Lnx> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX_DG', 'air01.dbf', 'SUN', 'air01', 'DB_LINK' );  
Lnx> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX_DG', 'air02.dbf', 'SUN', 'air02', 'DB_LINK' );
```

ASM 은 dd 가 지원되지 않음!!
ASM ftp 로 OS Down 후 dd 로 전송가능

```
Lnx> ftp sun
```

```
put air.dmp
```

```
Lnx> alter tablespace air read write;
```

6. Plug-In

```
% imp userid=W'sys/welcome as sysdbaW'  
file=air.dmp log=air_i.log W  
transport_tablespace=y W  
datafiles='/dev/vx/rdisk/racdg/air01',  
'/dev/vx/rdisk/racdg/air02'
```

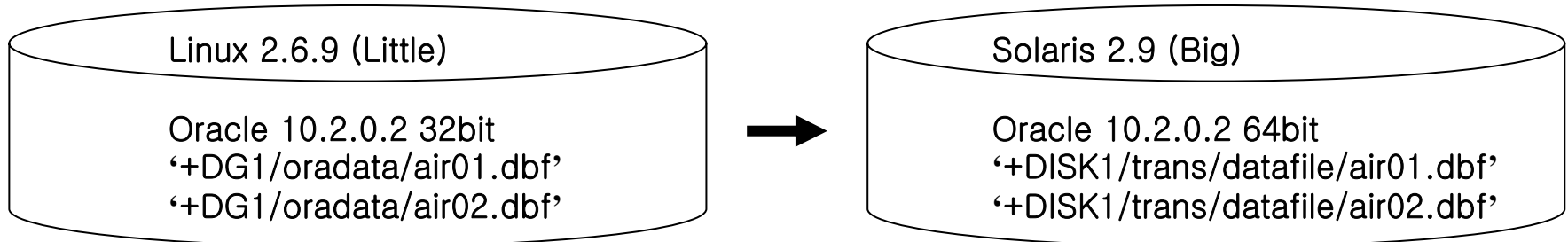
```
SUN> alter tablespace air read write;
```

```
SUN> alter user air default tablespace air;
```

7. 기타 확인 내용

- 1) 통계정보
- 2) Constraints, Trigger 를 제외한 기타 Procedure, Function 등, Invalid Object 확인
- 3) 기타 실행계획

III. Linux ASM → Solaris ASM



Endian 이 다르므로 RMAN으로 Conversion 한 후 DBMS_FILE_TRANSFER 로 이전함.

1. 준비작업

1) 유저 생성 – Target 인 Sun 에...

```
SUN> create user air identified by air;  
SUN> grant connect,resource to air;
```

2) Directory 생성 (Linux, Sun both)

```
Lnx> create or replace directory LNX  
as '/fw1/oracle10/WORK';  
Lnx> grant write on directory LNX to air;  
SUN> create or replace directory SUN_FS  
as '/backup/trans';  
SUN> grant write on directory SUN to air;
```

3) DataLink 생성 (DBMS_FILE_TRANSFER 용)

```
HP> create public database link DB_LINK  
connect to air identified by air using 'SUN';
```

2. Transportable Check

```
HP> exec  
dbms_tts.transport_set_check('AIR',TRUE)  
HP> select * from sys.transport_set_violations;  
HP> alter tablespace air read only;
```

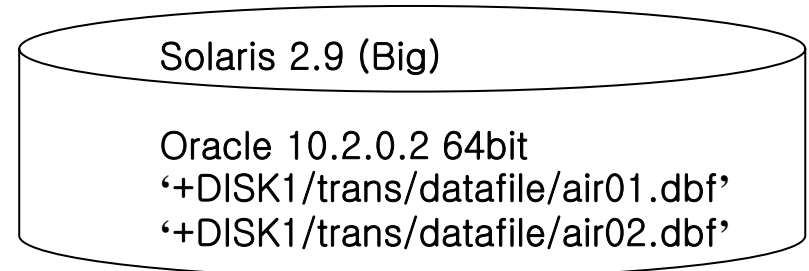
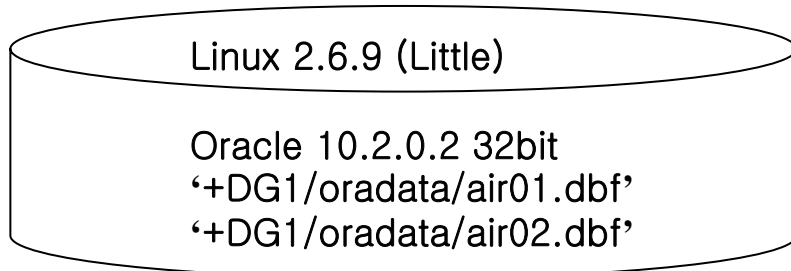
3. Exp only Meta

```
% exp userid=W'sys/welcom as sysdbaW'  
file=air.dmp log=air.log W  
transport_tablespace=y tablespaces=air
```

4. RMAN Convert to Solaris

```
% rman target=/ nocatalog  
RMAN> CONVERT TABLESPACE air  
TO PLATFORM 'Solaris[tm] OE (64-bit)'  
FORMAT '/fw1/oracle10/WORK/%U';
```

III. Linux ASM → Solaris ASM



OS File or raw-device 를 ASM 으로 바로 Copy 할 수 없음. (RMAN 으로 작업)

5. Data Copy

← 동시 실행 !!

```
LnX> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX', '파일이  
름', 'SUN_FS', 'air01.dbf', 'DB_LINK' );  
LnX> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX', '파일이  
름', 'SUN_FS', 'air02.dbf', 'DB_LINK' );
```

-- or --

```
dd if=파일이름1 ibsk=4k | rsh sun "dd  
of=/backup/trans/air01.dbf obs=8k"  
dd if=파일이름2 ibsk=4k | rsh sun "dd  
of=/backup/trans/air02.dbf obs=8k"
```

```
LnX> ftp sun  
put air.dmp  
LnX> alter tablespace air read write;
```

6. Plug-In

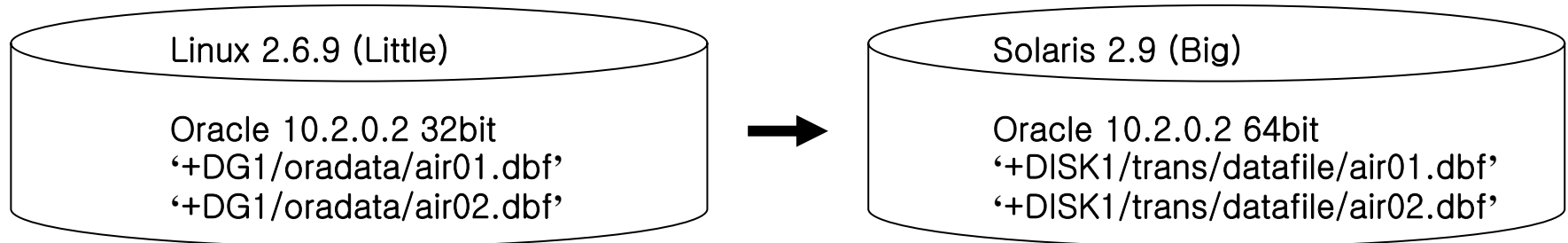
```
% imp userid=W'sys/welcome as sysdbaW'  
file=air.dmp log=air_i.log W  
transport_tablespace=y W  
datafiles='/backup/trans/air01.dbf',  
          '/backup/trans/air02.dbf'
```

```
SUN> alter tablespace air read write;  
SUN> alter user air default tablespace air;
```

7. Move F/S DBF to ASM DBF

```
SQL> alter tablespace TEST offline;  
% rman target=/  
RMAN> copy datafile 9  
to '+DISK1/trans/datafile/air01.dbf';  
    <<-- ORA-19602: cannot backup or copy  
active file in NOARCHIVELOG mode
```

III. Linux ASM → Solaris ASM



```
RMAN> shutdown immediate
RMAN> startup mount
RMAN> copy datafile 9 to '+DISK1/trans/datafile/air01.dbf';
RMAN> copy datafile 10 to '+DISK1/trans/datafile/air02.dbf';
input datafile fno=00007 name=/backup/trans/air01.dbf
output filename='+DISK1/trans/datafile/air01.dbf tag=TAG20070211T224451 recid=1 stamp=614299496
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
Finished backup at 11-FEB-07
```

```
% sqlplus '/as sysdba'
```

```
SQL> alter database rename file '/backup/trans/air01.dbf' to '+DISK1/trans/datafile/air01.dbf';
```

```
SQL> alter database rename file '/backup/trans/air02.dbf' to '+DISK1/trans/datafile/air02.dbf';
```

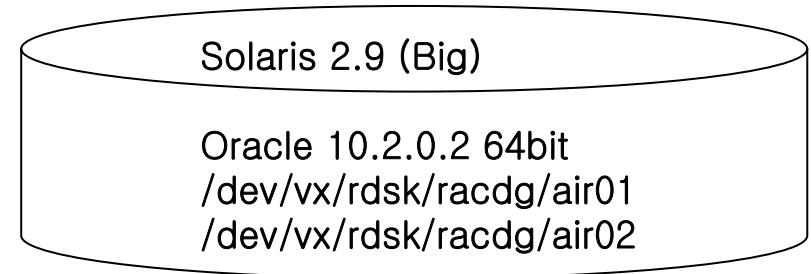
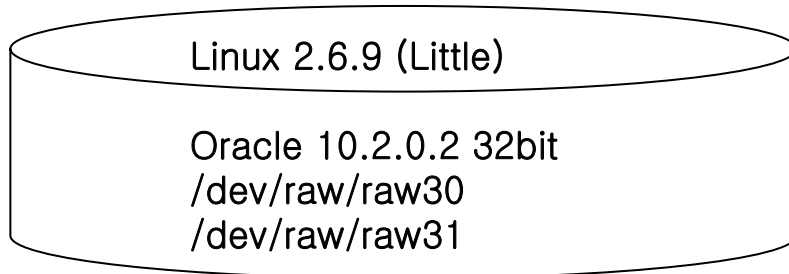
```
SQL> alter database open;
```

```
SQL> alter tablespace air online;
```

7. 기타 확인 내용

- 1) 통계정보
- 2) Constraints, Trigger 를 제외한 기타 Procedure, Function 등, Invalid Object 확인
- 3) 기타 실행계획

IV. Linux raw-device → Solaris raw-device



Endian 이 다르므로 RMAN으로 Conversion 한 후 dd or DBMS_FILE_TRANSFER 로 이전함.

1. 준비작업

1) 유저 생성 – Target 인 Sun 에...

```
SUN> create user air identified by air;  
SUN> grant connect,resource to air;
```

2) Directory 생성 (Linux, Sun both)

```
Lnx> create or replace directory LNX  
as '/fw1/oracle10/WORK';  
Lnx> grant write on directory LNX to air;  
SUN> create or replace directory SUN  
as '/dev/vx/rdisk/racdg';  
SUN> grant write on directory SUN to air;
```

3) DataLink 생성 (DBMS_FILE_TRANSFER 용)

```
HP> create public database link DB_LINK  
connect to air identified by air using 'SUN';
```

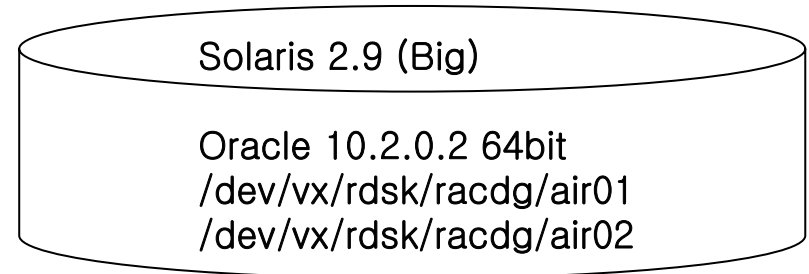
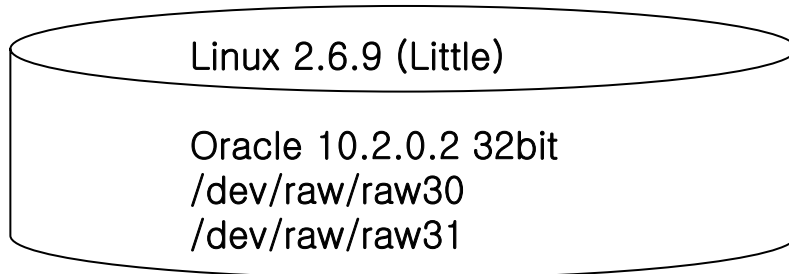
4) Sun 에 raw-device 생성

```
#vxassist -g racdg make air01 1200m layout=st  
ripe stripeunit=32k disk1 disk2 disk3  
#vxassist -g racdg make air02 2000m layout=st  
ripe stripeunit=32k disk1 disk2 disk3  
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air01  
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air02
```

2. Transportable Check

```
HP> exec  
dbms_tts.transport_set_check('AIR',TRUE)  
HP> select * from sys.transport_set_violations;  
HP> alter tablespace air read only;
```

IV. Linux raw-device → Solaris raw-device



3. Exp only Meta

```
% exp userid=W'sys/welcom as sysdbaW'  
file=air.dmp log=air.log W  
transport_tablespace=y tablespaces=air
```

4. RMAN Convert to Solaris

```
% rman target=/ nocatalog  
RMAN> CONVERT TABLESPACE air  
TO PLATFORM 'Solaris[tm] OE (64-bit)'  
FORMAT '/fw1/oracle10/WORK/%U';
```

5. Data Copy

← 동시 실행 !!

```
Lnx> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX', '파일이름1', 'SUN', 'air01', 'DB_LINK' );
```

```
Lnx> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'LNX', '파일이름2', 'SUN', 'air02', 'DB_LINK' );
```

-- or --

```
dd if=파일이름 ibsk=4k | rsh sun "dd  
of=/dev/vx/rdisk/racdg/air01 obs=8k"
```

```
Lnx> ftp sun
```

```
put air.dmp
```

```
Lnx> alter tablespace air read write;
```

6. Plug-In

```
% imp userid=W'sys/welcome as sysdbaW'  
file=air.dmp log=air_i.log W  
transport_tablespace=y W  
datafiles='/dev/vx/rdisk/racdg/air01',  
'/dev/vx/rdisk/racdg/air02'
```

```
SUN> alter tablespace air read write;
```

```
SUN> alter user air default tablespace air;
```

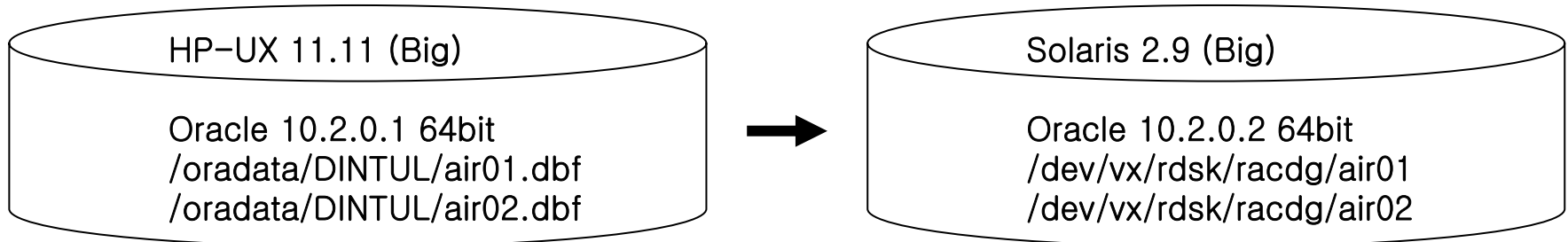
7. 기타 확인 내용

1) 통계정보

2) Constraints, Trigger 를 제외한 기타 Procedure, Function 등, Invalid Object 확인

3) 기타 실행계획

V. HP F/S → Solaris raw-device



Endian 이 같으므로 RMAN으로 Conversion 할 필요없이 dd or DBMS_FILE_TRANSFER 로 이전함.

1. 준비작업

1) 유저 생성 – Target 인 Sun 에...

```
SUN> create user air identified by air;  
SUN> grant connect,resource to air;
```

2) Directory 생성 (HP, Sun both)

```
HP> create or replace directory HP  
as '/oradata/DINTUL';
```

```
HP> grant write on directory HP to air;
```

```
SUN> create or replace directory SUN  
as '/dev/vx/rdsk/racdg';
```

```
SUN> grant write on directory SUN to air;
```

3) DataLink 생성 (DBMS_FILE_TRANSFER 용)

```
HP> create public database link DB_LINK  
connect to air identified by air using 'SUN';
```

4) Sun 에 raw-device 생성

```
#vxassist -g racdg make air01 1200m layout=st  
ripe stripeunit=32k disk1 disk2 disk3
```

```
#vxassist -g racdg make air02 2000m layout=st  
ripe stripeunit=32k disk1 disk2 disk3
```

```
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air01
```

```
#vxdedit -g racdg set user=oracle10 group=dba  
mode=660 air02
```

2. Transportable Check

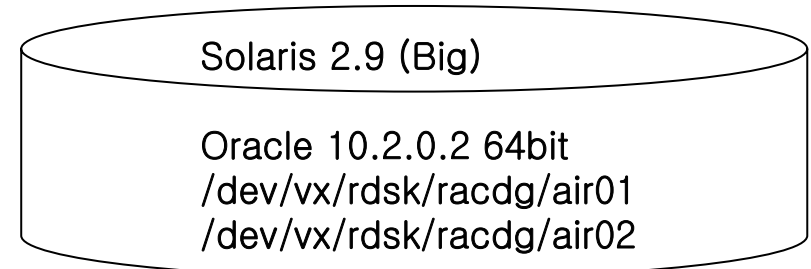
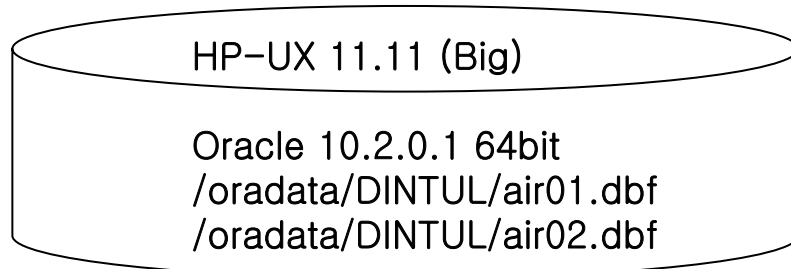
```
HP> exec
```

```
dbms_tts.transport_set_check('AIR',TRUE)
```

```
HP> select * from sys.transport_set_violations;
```

```
HP> alter tablespace air read only;
```

V. HP F/S → Solaris raw-device



3. Exp only Meta

```
% exp userid=W'sys/welcom as sysdbaW'  
file=air.dmp log=air.log W  
transport_tablespace=y tablespaces=air
```

4. Data Copy

← 동시 실행 !!

```
HP> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'HP', 'air01.dbf', 'SUN', 'air01', 'DB_LINK' );  
HP> exec DBMS\_FILE\_TRANSFER.PUT\_FILE  
( 'HP', 'air02.dbf', 'SUN', 'air02', 'DB_LINK' );
```

```
HP> ftp sun  
put air.dmp
```

```
HP> alter tablespace air read write;
```

5. Plug-In

```
% imp userid=W'sys/welcome as sysdbaW'  
file=air.dmp log=air_i.log W  
transport_tablespace=y W  
datafiles=' /dev/vx/rdisk/racdg/air01',  
          ' /dev/vx/rdisk/racdg/air02'
```

```
SUN> alter tablespace air read write;  
SUN> alter user air default tablespace air;
```

6. 기타 확인 내용

- 1) 통계정보
- 2) Constraints, Trigger 를 제외한 기타 Procedure, Function 등, Invalid Object 확인
- 3) 기타 실행계획

V. HP F/S → Solaris raw-device

** TTS 의 Target 에 Plug-In 시에 10.2.0.2 → 10.2.0.1 로 하는 경우
즉 높은 Version 의 Dump 를 낮은 Version 으로 Imp 할때는 Error 발생 !!

```
. imp userid=W'sys/welcome as sysdbaW' file=test.dmp log=test_i.log W  
transport_tablespace=y datafiles='/oracle10/app/oracle/oradata/DINTUL/test01.dbf'
```

Export file created by EXPORT:V10.02.01 via conventional path

About to import transportable tablespace(s) metadata...

import done in KO16MSWIN949 character set and AL16UTF16 NCHAR character set

export client uses KO16KSC5601 character set (possible charset conversion)

. importing SYS's objects into SYS

. importing SYS's objects into SYS

IMP-00017: following statement failed with ORACLE error 721:

```
"BEGIN sys.dbms_plugts.checkCompType('COMPATSG','10.2.0.2.0'); END;"
```

IMP-00003: ORACLE error 721 encountered

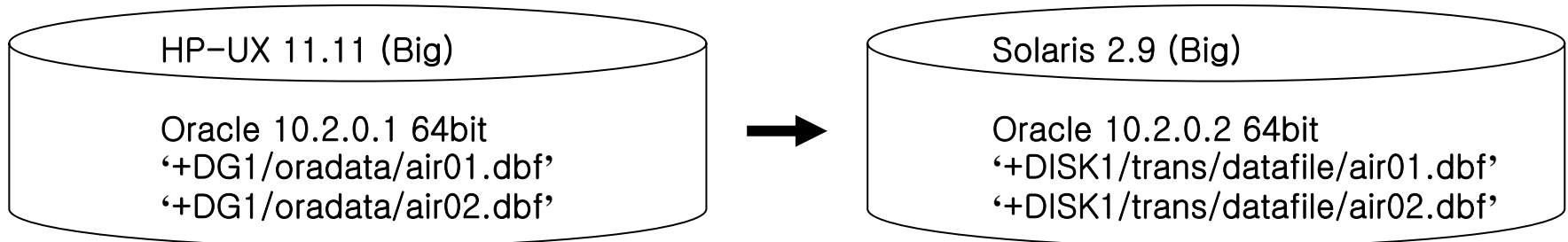
ORA-00721: changes by release 10.2.0.2.0 cannot be used by release 10.2.0.1.0

ORA-06512: at "SYS.DBMS_PLUGTS", line 2004

ORA-06512: at line 1

IMP-00000: Import terminated unsuccessfully

VI. HP ASM → Solaris ASM



Endian 이 같으므로 RMAN으로 Conversion 할 필요없이 DBMS_FILE_TRANSFER 로 이전함.

1. 준비작업

1) 유저 생성 – Target 인 Sun 에...

```
SUN> create user air identified by air;  
SUN> grant connect,resource to air;
```

2) Directory 생성 (HP, Sun both)

```
HP> create or replace directory HP_DG  
as '+DG1/oradata;  
HP> grant write on directory HP to air;  
SUN> create or replace directory SUN_DG  
as '+DISK2/trans/datafile';  
SUN> grant write on directory SUN to air;
```

3) DataLink 생성 (DBMS_FILE_TRANSFER 용)

```
HP> create public database link DB_LINK  
connect to air identified by air using 'SUN';
```

2. Transportable Check

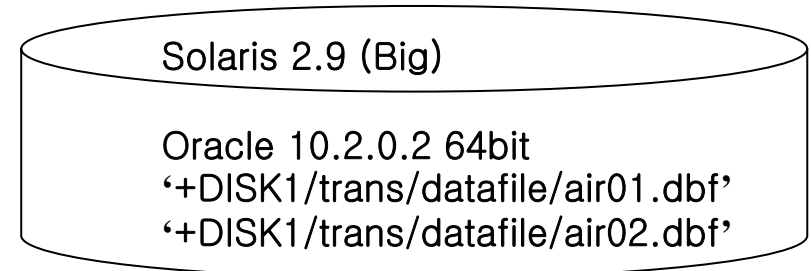
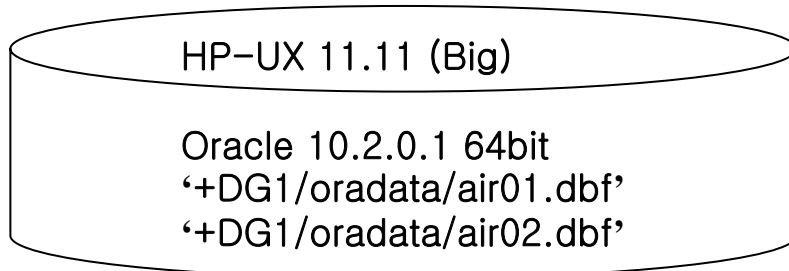
```
HP> exec  
dbms_tts.transport_set_check('AIR',TRUE)  
HP> select * from sys.transport_set_violations;  
HP> alter tablespace air read only;
```

3. Exp only Meta

```
% exp userid=W'sys/welcom as sysdbaW'  
file=air.dmp log=air.log W  
transport_tablespace=y tablespaces=air
```

ASM 은 dd 가 지원되지 않음!!
DBMS_FILE_TRANSFER 를 통해 ASM 간에
DBF 를 옮길수 있음!!

VI. HP ASM → Solaris ASM



4. Data Copy

← 동시 실행 !!

```
HP> exec DBMS_FILE_TRANSFER.PUT_FILE  
( 'HP_DG', 'air01.dbf', 'SUN_DG', 'air01.dbf', 'DB_L  
INK' );
```

```
HP> exec DBMS_FILE_TRANSFER.PUT_FILE  
( 'HP_DG', 'air02.dbf', 'SUN_DG', 'air02.dbf', 'DB_L  
INK' );
```

```
HP> ftp sun  
put air.dmp
```

```
HP> alter tablespace air read write;
```

5. Plug-In

```
% imp userid=W'sys/welcome as sysdbaW'  
file=air.dmp log=air_i.log W  
transport_tablespace=y W  
datafiles='+DISK1/trans/datafile/air01.dbf',  
          '+DISK1/trans/datafile/air02.dbf'
```

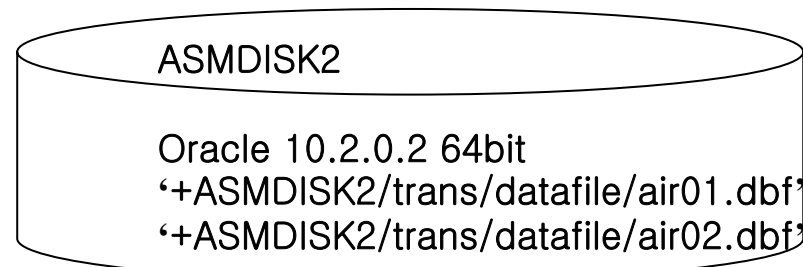
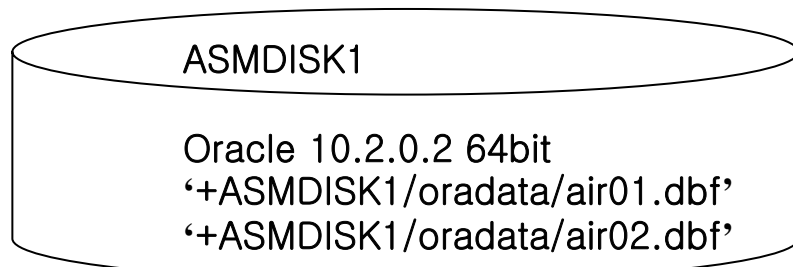
```
SUN> alter tablespace air read write;
```

```
SUN> alter user air default tablespace air;
```

6. 기타 확인 내용

- 1) 통계정보
- 2) Constraints, Trigger 를 제외한 기타 Procedure, Function 등, Invalid Object 확인
- 3) 기타 실행계획

VII. Move ASM DBF → other Diskgroup



1. 준비작업

1) Directory 생성 (ASMDISK1, 2 both)

```
SQL> create or replace directory ASMDISK1  
as '+ASMDISK1/oradata';
```

```
ASM> alter diskgroup ASMDISK2 add directory  
'+ASMDISK2/trans/datafile';
```

```
SUN> create or replace directory ASMDISK2  
as '+DISK2/trans/datafile';
```

2. TBS Offline

```
SQL> alter tablespace air offline;  
or alter database datafile 9 offline;  
alter database datafile 10 offline;
```

3. Data Copy

← 동시 실행 !!

```
SQL> exec DBMS_FILE_TRANSFER.COPY_FILE  
( 'ASMDISK1', 'air01.dbf', 'ASMDISK2', 'air01.dbf' );
```

```
SQL> exec DBMS_FILE_TRANSFER.COPY_FILE  
( 'ASMDISK1', 'air02.dbf', 'ASMDISK2', 'air02.dbf' );
```

4. Rename , Recover & Online

```
SQL> alter database rename file  
'+ASMDISK1/oradata/air01.dbf' to  
'+ASMDISK2/trans/datafile/air01.dbf';
```

```
SQL> alter database rename file  
'+ASMDISK1/oradata/air02.dbf' to  
'+ASMDISK2/trans/datafile/air02.dbf';
```

```
SQL> recover datafile 9;
```

```
SQL> recover datafile 10;
```

```
SQL> alter database datafile 9 online;
```

```
SQL> alter database datafile 10 online;
```

```
SQL> select name from v$dbfile;
```

5. File Drop Original location

```
SQL> alter diskgroup ASMDISK1  
drop file air01.dbf;
```

```
SQL> alter diskgroup ASMDISK1  
drop file air02.dbf;
```

** 일반 OS 의 File-System DBF or raw-device 를 DBMS_FILE_TRANSFER 을 이용해서 ASM 으로 Copy 할 경우 !!

```
SQL> exec dbms_file_transfer.copy_file('TEST','test','DIR','test01.dbf');  
ERROR at line 1:  
ORA-29285: file write error  
ORA-19664: file type: Non-Oracle file transfer: OS->ASM, file name:  
+DISK1/trans/datafile/test01.dbf  
ORA-06512: at "SYS.DBMS_FILE_TRANSFER", line 84  
ORA-06512: at "SYS.DBMS_FILE_TRANSFER", line 193  
ORA-06512: at line 1
```

1. 참고자료

- Metalink Note :: 371556.1
- Metalink Note :: 283991.1
- Metalink Note :: 144212.1

2. QnA

감 사 합 니 다