
TTS 을 이용한 Migration

Author	허종수
Creation Date	2019-07-05
Last Updated	2019-07-05
Version	1.0
Copyright(C) 2004 GoodusData Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2019.07.05	허종수	

Contents

1. 동일 Platform 방식	4
1.1. TEST 환경	4
1.2. As-Is 테스트용 데이터 생성	4
1.3. As-Is Platfrom 확인 및 Violation Check	6
1.4. As-Is Block Change Tracking 설정	7
1.5. As-Is Database Full backup	7
1.6. As-Is Database Full backup 파일 To-Be 전송	8
1.7. To-Be Tablespace 생성 및 유저 생성	8
1.8. To-Be Datafile 명에 맞게 Full backup 본 Copy 수행	9
1.9. As-Is Incremental 백업 테스트를 위한 추가 첫번째 데이터 생성	9
1.10. As-Is 첫번째 추가 데이터를 위한 Incremental 백업	10
1.11. As-Is 첫번째 Incremental 백업 파일 To-Be 전송	11
1.12. To-Be 첫번째 Incremental backup 본 Recovery 수행	11
1.13. As-Is Incremental 백업 테스트를 위한 추가 두번째 데이터 생성	12
1.14. As-Is Tablespace Read Only 모드로 변경(Down Time 시작)	13
1.15. As-Is 두번째 추가 데이터를 위한 최종 Incremental 백업	13
1.16. As-Is 두번째 Incremental 백업 파일 To-Be 전송	14
1.17. To-Be 두번째 Incremental backup 본 최종 Recovery 수행	14
1.18. As-Is TTS Tablespace export	15
1.19. As-Is expdp 파일 To-Be 전송	16
1.20. TTS Tablespace import	16
1.21. 추가작업	17
1.22. To-Be 데이터 확인	18
2. 서로 다른 Platform 방식	18
2.1. TEST 환경	18
2.2. As-Is 테스트용 데이터 생성	18
2.3. As-Is Platfrom 확인 및 Violation Check	20
2.4. As-Is Block Change Tracking 설정	21
2.5. As-Is Database Full backup	21
2.6. As-Is Database Full backup 파일 To-Be 전송	22
2.7. To-Be Tablespace 생성 및 유저 생성	22
2.8. To-Be Endian Forma 에 맞게 Full backup 본 Convert 수행	23
2.9. As-Is Incremental 백업 테스트를 위한 추가 첫번째 데이터 생성	24
2.10. As-Is 첫번째 추가 데이터를 위한 Incremental 백업	25
2.11. As-Is 첫번째 Incremental 백업 파일 To-Be 전송	26
2.12. To-Be Endian Format 에 맞게 Incremental backup 본 Convert 수행	26
2.13. To-Be Incremental backup 본 Recovery 수행	28
2.14. As-Is Incremental 백업 테스트를 위한 추가 두번째 데이터 생성	29
2.15. As-Is Tablespace Read Only 모드로 변경(Down Time 시작)	30
2.16. As-Is 두번째 추가 데이터를 위한 최종 Incremental 백업	30
2.17. As-Is 두번째 Incremental 백업 파일 To-Be 전송	31
2.18. To-Be 두번째 Incremental backup 본 최종 Recovery 수행	31
2.19. To-Be Incremental backup 본 Recovery 수행	32
2.20. As-Is TTS Tablespace export	33
2.21. As-Is expdp 파일 To-Be 전송	34
2.22. TTS Tablespace import	34
2.23. 추가작업	35
2.24. To-Be 데이터 확인	36
3. 제약사항	36

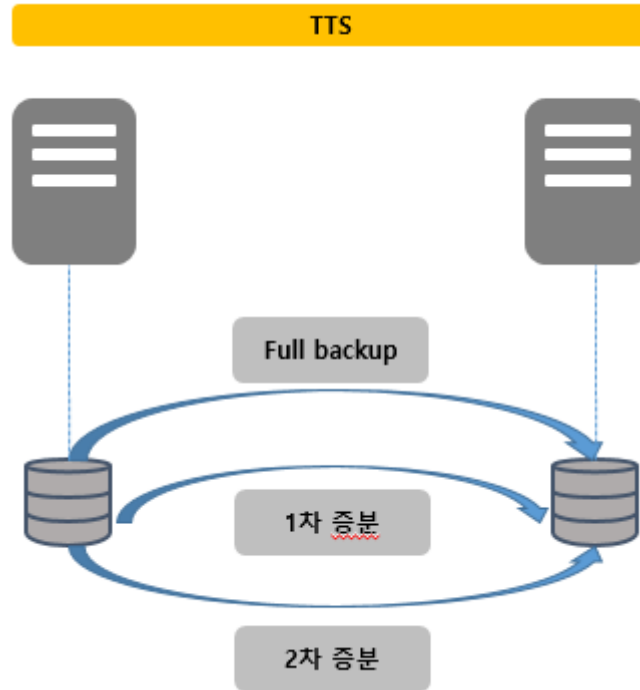
3.1.	expdp 유틸리티 사용 시 TRANSPORT_TABLESPACES 옵션	36
3.2.	Index Organized 테이블은 트랜스포터블 테이블스페이스 사용 시 손상될 수 있습니다.	37
3.3.	SYSTEM 테이블스페이스나 SYS 사용자가 소유한 객체는 트랜스포트하는 것이 불가능합니다.	37
3.4.	Character Set 과 National Character Set 제한사항.....	37
3.5.	같은 이름을 갖는 테이블스페이스가 이미 존재하는 경우.....	37
3.6.	타겟 서버에 owner 가 new 인 객체	37
3.7.	XML 타입을 포함하는 테이블스페이스.....	37
3.8.	11.2에서는 exp / imp 방식은 지원되지 않습니다.....	37
3.9.	암호화된 테이블스페이스는 다음 제약사항을 갖습니다.....	37

대용량 데이터를 TTS(Transportable Tablespace) 증분백업 방식으로 이관하는 방법을 알아보겠습니다.

일반적인 TTS는 Tablespace Read Only 모드로 변경 후 데이터 파일을 To-Be 로 Copy 하는 동안 긴 시간 동안의 Down Time 이 발생하나 증분백업 방식을 이용할 경우 최종 증분 백업 시에만 Down Time 을 필요로 하므로 대용량 / 중요시스템에서 이관하는 방법 중의 하나입니다.

일반적인 TTS 보다 절차는 복잡하지만, 한번 생성한 스크립트는 일부분만 변경하여 계속 증분백업 시 사용가능합니다. 단, 최초 Full 백업이 이루어지므로 백업 받을 충분한 디스크 공간 확보가 되어야합니다.

TTS 를 이용하여 이관하기 위해서는 사전에 To-Be 에 Database 는 생성되어 있어야합니다.



1. 동일 Platform 방식

- 동일 PLATFORM_인 Linux(Linux x86 64-bit)에서 Linux(Linux x86 64-bit)로 이관하는 방법을 알아보겠습니다.

1.1. TEST 환경

항목	As-Is	To-Be
OS Platform	Linux x86 64-bit	Linux x86 64-bit
Database 버전	11.2.0.4	12.2.0.1
Datafile Type	Filesystem	Filesystem

1.2. As-Is 테스트용 데이터 생성

- 이관 할 대상 데이터 생성을 위한 유저 2 개, Tablespace 2 개(JONGSOO, GOODUSDATA), Datafile 3 개 생성합니다.

- 초기 데이터인 "before"을 insert 합니다.

```
SQL> create tablespace jongsoo datafile ' /oracle/oradata/asis/jongsoo01.dbf' size 5m;

Tablespace created.

SQL> alter tablespace jongsoo add datafile ' /oracle/oradata/asis/jongsoo02.dbf' size 5m;

Tablespace altered.

SQL> create tablespace goodusdata datafile ' /oracle/oradata/asis/goodusdata01.dbf' size 10m;

Tablespace created.

SQL> create user jongsoo identified by jongsoo default tablespace JONGSOO;

User created.

SQL> create user goodusdata identified by goodusdata default tablespace GOODUSDATA;

User created.

SQL> grant connect, resource to jongsoo;

Grant succeeded.

SQL> grant connect, resource to goodusdata;

Grant succeeded.

SQL> create table jongsoo.test1 (name varchar(10));

Table created.

SQL> create table goodusdata.test2 (name varchar(10));

Table created.

SQL> insert into jongsoo.test1 values ('before');
```

```

1 row created.

SQL> commit;

Commit complete.

SQL> insert into goodusdata.test2 values ('before');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from jongssoo.test1;

NAME
-----
before

SQL> select * from goodusdata.test2;

NAME
-----
before

```

1.3. As-Is Platform 확인 및 Violation Check

- 이관하려는 Tablespace 에 속해 있는 Table 및 Index 가 다른 영역의 Tablespace 에 저장되었는지를 확인하는 절차입니다.. No rows selected 가 아닌 정보가 나온다면 Table 및 Index 를 이관하려는 Tablespace 로 이동 하거나 해당 Tablespace 도 이관합니다.
- dbms_tts.transport_set_check 에 이관 대상 Tablespace 명 입력합니다.

```

SQL> select PLATFORM_NAME from v$database;

PLATFORM_NAME
-----
Linux x86 64-bit

SQL> exec dbms_tts.transport_set_check('JONGSOO, GOODUSDATA',TRUE);

```

```
PL/SQL procedure successfully completed.
```

```
SQL> select * from transport_set_violations;
```

```
no rows selected
```

1.4. As-Is Block Change Tracking 설정.

- Block 들의 변경사항을 파일에 기록하는 기능입니다.
- 필수 사항은 아니지만 Incremental Performance 향상을 위해 설정합니다.

```
SQL> alter database enable block change tracking using file '/oracle/oradata/asis/bct.ctl';
```

```
Database altered.
```

1.5. As-Is Database Full backup

- "before"가 포함된 초기 데이터백업을 위해 이관 대상 Tablespace 을 Full 백업합니다.

```
[oracle@single83:/oracle/arch/asis]$ rman target /
```

```
Recovery Manager: Release 11.2.0.4.0 - Production on Mon Nov 26 16:19:45 2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ASIS (DBID=743331713)
```

```
RMAN> run
```

```
{
```

```
BACKUP AS COPY INCREMENTAL LEVEL 0 tag 'full_backup' tablespace JONGS00, GOODUSDATA format
```

```
'/oracle/rman/file%f%U.bkp';
```

```
}2> 3> 4>
```

```
Starting backup at 26-NOV-18
```

```
using target database control file instead of recovery catalog
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=131 device type=DISK
```

```
channel ORA_DISK_1: starting datafile copy
```

```
input datafile file number=00007 name=/oracle/oradata/asis/goodusdata01.dbf
```

```
output file name=/oracle/rman/file7data_D-ASIS_I-743331713_TS-GOODUSDATA_FNO-7_04tj6s5b.bkp tag=FULL_BACKUP
```

```
RECID=1 STAMP=993226923
```

```

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00005 name=/oracle/oradata/asis/jongsoo01.dbf
output file name=/oracle/rman/file5data_D-ASIS_I-743331713_TS-JONGS00_FNO-5_05tj6s5c.bkp tag=FULL_BACKUP
RECID=2 STAMP=993226924
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00006 name=/oracle/oradata/asis/jongsoo02.dbf
output file name=/oracle/rman/file6data_D-ASIS_I-743331713_TS-JONGS00_FNO-6_06tj6s5d.bkp tag=FULL_BACKUP
RECID=3 STAMP=993226925
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 26-NOV-18

```

1.6. As-Is Database Full backup 파일 To-Be 전송

- Full backup 파일 To-Be 로 전송합니다.(SCP, FTP 등)

1.7. To-Be Tablespace 생성 및 유저 생성

- To-Be 에 사전에 As-Is 와 같은 유저, default tablespace 는 다르게 하여 생성해놓습니다.

```

SQL> create user jongsoo identified by jongsoo default tablespace USERS;

User created.

SQL> create user goodusdata identified by goodusdata default tablespace USERS;

User created.

SQL> grant connect, resource to jongsoo;

Grant succeeded.

SQL> grant connect, resource to goodusdata;

Grant succeeded.

SQL> select PLATFORM_NAME from v$database;

PLATFORM_NAME

```


1.8. To-Be Datafile 명에 맞게 Full backup 본 Copy 수행

- 파일의 개수 만큼 기존 Datafile Name 또는 변경할 Name 으로 원하는 경로로 Copy 수행합니다.(테스트 환경에선 Datafile 3 개)
- 5 번 절차의 Full Backup 본의 로그에서 output file name 명을 확인합니다.

```
[KIMSYDB:TESTDB:/DATA/rman]cp file5data_D-ASIS_I-743331713_TS-JONGS00_FNO-5_05tj6s5c.bkp
/DATA/TESTDB/jongsoo01.dbf
[KIMSYDB:TESTDB:/DATA/rman]cp file6data_D-ASIS_I-743331713_TS-JONGS00_FNO-6_06tj6s5d.bkp
/DATA/TESTDB/jongsoo02.dbf
[KIMSYDB:TESTDB:/DATA/rman]cp file7data_D-ASIS_I-743331713_TS-GOODUSDATA_FNO-7_04tj6s5b.bkp
/DATA/TESTDB/goodusdata01.dbf
```

1.9. As-Is Incremental 백업 테스트를 위한 추가 첫번째 데이터 생성

- 첫번째 증분백업을 위한 "middle" 데이터를 insert 합니다.

```
SQL> insert into jongsoo.test1 values ('middle');

1 row created.

SQL> commit;

Commit complete.

SQL> insert into goodusdata.test2 values ('middle');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from jongsoo.test1;

NAME
-----
before
middle
```

```
SQL> select * from goodusdata.test2;
```

```
NAME
```

```
-----
```

```
before
```

```
middle
```

1.10. As-Is 첫번째 추가 데이터를 위한 Incremental 백업

- "middle" 데이터를 백업하기 위해 증분 백업을 수행합니다.
- 이관 Tablespace 명을 입력합니다.

```
[oracle@single83:/oracle/rman/incre1]$ rman target /
```

```
Recovery Manager: Release 11.2.0.4.0 - Production on Mon Nov 26 18:22:36 2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ASIS (DBID=743331713)
```

```
RMAN> run
```

```
{
```

```
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'full_backup' format  
'/oracle/rman/incre1/INCRE%U.inc' tablespace JONGSOO, GOODUSDATA;
```

```
}2> 3> 4>
```

```
Starting backup at 26-NOV-18
```

```
using target database control file instead of recovery catalog
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=10 device type=DISK
```

```
channel ORA_DISK_1: starting incremental level 1 datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```
input datafile file number=00007 name=/oracle/oradata/asis/goodusdata01.dbf
```

```
input datafile file number=00005 name=/oracle/oradata/asis/jongsoo01.dbf
```

```
input datafile file number=00006 name=/oracle/oradata/asis/jongsoo02.dbf
```

```
channel ORA_DISK_1: starting piece 1 at 26-NOV-18
```

```
channel ORA_DISK_1: finished piece 1 at 26-NOV-18
```

```
piece handle=/oracle/rman/incre1/INCRE07tj7393_1_1.inc tag=FULL_BACKUP comment=NONE
```

```
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
```

```
Finished backup at 26-NOV-18
```

```
RMAN>
```

1.11. As-Is 첫번째 Incremental 백업 파일 To-Be 전송

- Incremental 백업 파일 To-Be 로 전송합니다.(SCP, FTP 등)

1.12. To-Be 첫번째 Incremental backup 본 Recovery 수행

- To-Be 에 첫번째 증분 백업 본을 Recovery 하기 위한 절차입니다.
- 아래 빨간색만 수정한다. dfnumber 는 As-Is file_id 지정, To-Be 파일 경로 지정, Incremental 백업본 파일 위치 지정합니다.

```
SQL> set serveroutput on ;
DECLARE
d varchar2(512);
h varchar2(512) ;
t varchar2(30) ;
b1 boolean ;
b2 boolean ;
DONE boolean ;
FAILOVER boolean ;
BEGIN
d := sys.dbms_backup_restore.deviceAllocate;
DBMS_OUTPUT.PUT_LINE(d);
sys.dbms_backup_restore.applysetdatafile(check_logical=>false, cleanup=>false) ;
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>5 , toname => '/DATA/TESTDB/jongsoo01.dbf' ,
fuzziness_hint=>0,max_corrupt =>0,islevel0=>0,recid=>0,stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>6 , toname => '/DATA/TESTDB/jongsoo02.dbf' ,
fuzziness_hint=>0,max_corrupt =>0,islevel0=>0,recid=>0,stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>7 , toname => '/DATA/TESTDB/goodusdata01.dbf' ,
fuzziness_hint=>0,max_corrupt =>0,islevel0=>0,recid=>0,stamp=>0);
sys.dbms_backup_restore.restoreSetPiece( handle=>'/DATA/rman/incre1/INCRE07tj7393_1_1.inc' ,
tag=>null,fromdisk=>true,recid=>0,stamp=>0) ;
sys.dbms_backup_restore.restoreBackupPiece( done=>DONE, params=>null, outhandle=>h, outtag=>t,
FAILOVER=>FAILOVER);
sys.dbms_backup_restore.restoreCancel(TRUE);
sys.dbms_backup_restore.deviceDeallocate;
END;
/
```

```
DISK
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

1.13. As-Is Incremental 백업 테스트를 위한 추가 두번째 데이터 생성

- 두번째 증분백업을 위한 "after" 데이터를 insert 합니다.
- 테스트 문서에서는 "after" 데이터가 입력 된 시점이 최종 데이터라고 가정합니다.

```
SQL> insert into jongsoo.test1 values ('after');
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> insert into goodusdata.test2 values ('after');
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> select * from jongsoo.test1;
```

```
NAME
```

```
-----
```

```
after
```

```
before
```

```
middle
```

```
SQL> select * from goodusdata.test2;
```

```
NAME
```

```
-----
```

```
after
```

```
before
middle
```

1.14. As-Is Tablespace Read Only 모드로 변경(Down Time 시작)

- 앞단계에서 최종데이터라고 가정하였으므로 Read Only 모드로 변경합니다.
- 이 시점이 Down Time 시작입니다.
- 데이터 변경이 많은 시스템에서는 증분 백업을 여러 번 할수 있습니다.

```
SQL> alter tablespace JONGS00 read only;
```

```
Tablespace altered.
```

```
SQL> alter tablespace GOODUSDATA read only;
```

```
Tablespace altered.
```

1.15. As-Is 두번 째 추가 데이터를 위한 최종 Incremental 백업

- 최종 "after" 데이터를 백업하기 위해 증분 백업을 수행합니다.
- 이관 Tablespace 명을 입력합니다.

```
[oracle@single83:/oracle/rman/last]$ rman target /
```

```
Recovery Manager: Release 11.2.0.4.0 - Production on Mon Nov 26 18:48:12 2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ASIS (DBID=743331713)
```

```
RMAN> run
```

```
{
```

```
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'full_backup' format '/oracle/rman/last/INCRE%U.inc'
tablespace JONGS00, GOODUSDATA;
```

```
}2> 3> 4>
```

```
Starting backup at 26-NOV-18
```

```
using target database control file instead of recovery catalog
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=12 device type=DISK
```

```
channel ORA_DISK_1: starting incremental level 1 datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```

input datafile file number=00007 name=/oracle/oradata/asis/goodusdata01.dbf
input datafile file number=00005 name=/oracle/oradata/asis/jongsoo01.dbf
input datafile file number=00006 name=/oracle/oradata/asis/jongsoo02.dbf
channel ORA_DISK_1: starting piece 1 at 26-NOV-18
channel ORA_DISK_1: finished piece 1 at 26-NOV-18
piece handle=/oracle/rman/last/INCRE08tj74nl_1_1.inc tag=FULL_BACKUP comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 26-NOV-18

RMAN>

```

1.16. As-Is 두번째 Incremental 백업 파일 To-Be 전송

- Incremental 백업 파일 To-Be 로 전송합니다.(SCP, FTP 등)

1.17. To-Be 두번째 Incremental backup 본 최종 Recovery 수행

- To-Be 에 두번째 최종 증분 백업 본을 Recovery 하기 위한 절차입니다.
- 아래 빨간색만 수정합니다. dfnumber 는 As-Is file_id 지정, To-Be 파일 경로 지정, Incremental 백업본 파일명을 지정합니다.
- 첫번째 증분 백업 Recovery 과정에서 변경 점은 Incremental 파일명만 틀립니다.

```

SQL> set serveroutput on ;
DECLARE
d varchar2(512);
h varchar2(512) ;
t varchar2(30) ;
b1 boolean ;
b2 boolean ;
DONE boolean ;
FAILOVER boolean ;
BEGIN
d := sys.dbms_backup_restore.deviceAllocate;
DBMS_OUTPUT.PUT_LINE(d);
sys.dbms_backup_restore.applysetdatafile(check_logical=>false, cleanup=>false) ;
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>5 , toname => '/DATA/TESTDB/jongsoo01.dbf' ,
fuzziness_hint=>0,max_corrupt =>0,islevel0=>0,recid=>0,stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>6 , toname => '/DATA/TESTDB/jongsoo02.dbf' ,
fuzziness_hint=>0,max_corrupt =>0,islevel0=>0,recid=>0,stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>7 , toname => '/DATA/TESTDB/goodusdata01.dbf' ,
fuzziness_hint=>0,max_corrupt =>0 , islevel0=>0,recid=>0,stamp=>0);

```

```

sys.dbms_backup_restore.restoreSetPiece( handle=>'/DATA/rman/last/INCRE08tj74nl_1_1.inc',
tag=>null,fromdisk=>true,recid=>0,stamp=>0) ;
sys.dbms_backup_restore.restoreBackupPiece( done=>DONE, params=>null, outhandle=>h, outtag=>t,
FAILOVER=>FAILOVER); -> 빨간색만 수정. dfnumber 는 As-Is file_id 지정, To-Be 파일 경로 지정, Incremental
convert 파일 위치 지정
sys.dbms_backup_restore.restoreCancel(TRUE);
sys.dbms_backup_restore.deviceDeallocate;
END;
/
DISK

PL/SQL procedure successfully completed.

SQL>

```

1.18. As-Is TTS Tablespace export

- Transport_tablespace 옵션을 사용하여 이관하고자 하는 Tablespace 을 export 합니다.
- 본 문서에는 Datapump 유틸리티를 사용하였습니다.

```

SQL> create or replace directory tts as '/oracle/tts';

Directory created.

SQL> grant read, write on directory tts to system;

Grant succeeded.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning option

[oracle@single83:/oracle/tts]$ expdp W'sys/oracle as sysdbaW' dumpfile=exptts.dmp logfile=exptts.log
directory=tts transport_tablespaces=JONGS00,GOODUSDATA

Export: Release 11.2.0.4.0 - Production on Mon Nov 26 18:59:49 2018

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production

```

```

With the Partitioning option
Starting "SYS"."SYS_EXPORT_TRANSPORTABLE_01": "sys/***** AS SYSDBA" dumpfile=exptts.dmp
logfile=exptts.log directory=tts transport_tablespaces=JONGSOO,GOODUSDATA
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Master table "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/unloaded
*****
Dump file set for SYS.SYS_EXPORT_TRANSPORTABLE_01 is:
  /oracle/tts/exptts.dmp
*****
Datafiles required for transportable tablespace GOODUSDATA:
  /oracle/oradata/asis/goodusdata01.dbf
Datafiles required for transportable tablespace JONGSOO:
  /oracle/oradata/asis/jongsoo01.dbf
  /oracle/oradata/asis/jongsoo02.dbf
Job "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Mon Nov 26 19:00:06 2018 elapsed 0
00:00:15

[oracle@single83:/oracle/tts]$

```

1.19. As-Is expdp 파일 To-Be 전송

- expdp 파일 To-Be 로 전송합니다.(SCP, FTP 등)

1.20. TTS Tablespace import

- Transport_datafiles 옵션을 사용하여 이관하고자 하는 Tablespace 을 import 합니다.
- 본 문서에는 Datapump 유틸리티를 사용하였습니다.
- 이관하는 Tablespace 에 object 가 많을 경우 오랜 시간이 소요될 수 있습니다.

```

SQL> create or replace directory tts as '/DATA/tts';

Directory created.

SQL> grant read, write on directory tts to system;

Grant succeeded.

SQL> exit

Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

```



```

[KIMSYDB:TESTDB:/DATA/tts]impdp W'sys/oracle as sysdbaW' dumpfile=exptts.dmp logfile=imptts.log
directory=tts
transport_datafiles='/DATA/TESTDB/jongsoo01.dbf','/DATA/TESTDB/jongsoo02.dbf','/DATA/TESTDB/goodusdata01.dbf
'

Import: Release 12.2.0.1.0 - Production on Mon Nov 26 19:06:04 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
Master table "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/unloaded
Starting "SYS"."SYS_IMPORT_TRANSPORTABLE_01": "sys/***** AS SYSDBA" dumpfile=exptts.dmp
logfile=imptts.log directory=tts
transport_datafiles=/DATA/TESTDB/jongsoo01.dbf,/DATA/TESTDB/jongsoo02.dbf,/DATA/TESTDB/goodusdata01.dbf
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Mon Nov 26 19:06:25 2018 elapsed 0
00:00:12

[KIMSYDB:TESTDB:/DATA/tts]

```

1.21. 추가작업

- 임시로 지정했던 Schema Default tablespace 를 원래 명으로 변경합니다.
- Import 가 끝나면 As-Is 에서 수행했던 Read Only 모드로 되어 있기 때문에 Read Write 모드로 변경합니다.

```

SQL> alter user jongsoo default tablespace JONGSOO;

User altered.

SQL> alter user goodusdata default tablespace GOODUSDATA;

User altered.

SQL> alter tablespace JONGSOO read write;

Tablespace altered.

SQL> alter tablespace GOODUSDATA read write;

```

Tablespace altered.

1.22. To-Be 데이터 확인

- 해당 과정이 끝나면 최종 데이터인 "after" 데이터를 확인 할수 있습니다.

```
SQL> select * from jongsoo.test1;
```

NAME

after

before

middle

```
SQL> select * from goodusdata.test2;
```

NAME

after

before

middle

- 필요 시 기타 object As-Is 에서 export 후 To-Be 에서 import 수행합니다.

2. 서로 다른 Platform 방식

- 이 기존 장비인 AIX-Based Systems (64-bit)에서 Linux(Linux x86 64-bit)로 이관하는 방법을 알아보겠습니다.

2.1. TEST 환경

항목	As-Is	To-Be
OS Platform	AIX-Based Systems (64-bit)	Linux x86 64-bit
Database 버전	11.2.0.4	12.2.0.1
Datafile Type	Filesystem	ASM

2.2. As-Is 테스트용 데이터 생성

- 이관 할 대상 데이터 생성을 위한 유저 2 개, Tablespace 2 개(JONGSOO, GOODUSDATA), Datafile 3 개 생성합니다.

- 초기 데이터인 "before"을 insert 합니다.

```
SQL> create tablespace jongsoo datafile '/data/jongsoo/jongsoo01.dbf' size 5m;
```

Tablespace created.

```
SQL> alter tablespace jongsoo add datafile '/data/jongsoo/jongsoo02.dbf' size 5m;
```

Tablespace altered.

```
SQL> create tablespace goodusdata datafile '/data/jongsoo/goodusdata01.dbf' size 10m;
```

Tablespace created.

```
SQL> create user jongsoo identified by jongsoo default tablespace JONGSOO;
```

User created.

```
SQL> create user goodusdata identified by goodusdata default tablespace GOODUSDATA;
```

User created.

```
SQL> grant connect, resource to jongsoo;
```

Grant succeeded.

```
SQL> grant connect, resource to goodusdata;
```

Grant succeeded.

```
SQL> create table jongsoo.test1 (name varchar(10));
```

Table created.

```
SQL> create table goodusdata.test2 (name varchar(10));
```

Table created.

```
SQL> insert into jongsoo.test1 values ('before');
```

```

1 row created.

SQL> commit;

Commit complete.

SQL> insert into goodusdata.test2 values ('before');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from jongsoo.test1;

NAME
-----
before

SQL> select * from goodusdata.test2;

NAME
-----
before

```

2.3. As-Is Platform 확인 및 Violation Check

- 이관하려는 Tablespace 에 속해 있는 Table 및 Index 가 다른 영역의 Tablespace 에 저장되었는지를 확인하는 절차입니다.. No rows selected 가 아닌 정보가 나온다면 Table 및 Index 를 이관하려는 Tablespace 로 이동 하거나 해당 Tablespace 도 이관합니다.
- dbms_tts.transport_set_check 에 이관 대상 Tablespace 명을 입력합니다.

```

SQL> select PLATFORM_NAME from v$database;

PLATFORM_NAME
-----
AIX-Based Systems (64-bit)

SQL> exec dbms_tts.transport_set_check('JONGSOO, GOODUSDATA',TRUE);

```

```
PL/SQL procedure successfully completed.
```

```
SQL> select * from transport_set_violations;
```

```
no rows selected
```

2.4. As-Is Block Change Tracking 설정.

- Block 들의 변경사항을 파일에 기록하는 기능입니다.
- 필수 사항은 아니지만 Incremental Performance 향상을 위해 설정합니다.

```
SQL> alter database enable block change tracking using file '/data/jongsoo/bct.ctl';
```

```
Database altered.
```

2.5. As-Is Database Full backup

- "before"가 포함된 초기 데이터백업을 위해 이관 대상 Tablespace 을 Full 백업합니다.

```
/data/jongsoo/file $rman target /
```

```
Recovery Manager: Release 11.2.0.4.0 - Production on Tue Dec 11 10:25:23 2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: JONGSOO (DBID=88982687)
```

```
RMAN> run
```

```
{
```

```
BACKUP AS COPY INCREMENTAL LEVEL 0 tag 'full_backup' tablespace JONGSOO, GOODUSDATA format  
'/data/jongsoo/file/file%f%U.bkp';
```

```
}
```

```
Starting backup at 2018-12-11:10:25:30
```

```
using target database control file instead of recovery catalog
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=21 device type=DISK
```

```
channel ORA_DISK_1: starting datafile copy
```

```
input datafile file number=00007 name=/data/jongsoo/goodusdata01.dbf
```

```
output file name=/data/jongsoo/file/file7data_D-JONGSOO_I-88982687_TS-GOODUSDATA_FNO-7_08tkgd8s.bkp
```

```
tag=FULL_BACKUP RECID=4 STAMP=994587934
```

```

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting datafile copy
input datafile file number=00005 name=/data/jongsoo/jongsoo01.dbf
output file name=/data/jongsoo/file/file5data_D-JONGSOO_I-88982687_TS-JONGSOO_FNO-5_09tkgd8v.bkp
tag=FULL_BACKUP RECID=5 STAMP=994587936
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00006 name=/data/jongsoo/jongsoo02.dbf
output file name=/data/jongsoo/file/file6data_D-JONGSOO_I-88982687_TS-JONGSOO_FNO-6_0atkgd90.bkp
tag=FULL_BACKUP RECID=6 STAMP=994587937
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 2018-12-11:10:25:37

RMAN>

```

2.6. As-Is Database Full backup 파일 To-Be 전송

- Full backup 파일 To-Be 로 전송합니다.(SCP, FTP 등)

2.7. To-Be Tablespace 생성 및 유저 생성

- To-Be 에 사전에 As-Is 와 같은 유저, default tablespace 는 다르게 하여 생성해놓습니다.

```

SQL> create user jongsoo identified by jongsoo default tablespace USERS;

User created.

SQL> create user goodusdata identified by goodusdata default tablespace USERS;

User created.

SQL> grant connect, resource to jongsoo;

Grant succeeded.

SQL> grant connect, resource to goodusdata;

Grant succeeded.

SQL> select PLATFORM_NAME from v$database;

```

PLATFORM_NAME

Linux x86 64-bit

2.8. To-Be Endian Forma에 맞게 Full backup 본 Convert 수행

- 파일의 개수 만큼 convert 수행합니다.(테스트 환경에선 Datafile 3 개)
- 5 번 절차의 Full Backup 본의 로그에서 output file name 명을 확인합니다.

```
[oracle@rac01:/oracle/xtts]$ rman target /

Recovery Manager: Release 12.2.0.1.0 - Production on Tue Mar 27 21:41:08 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

connected to target database: GOODUS (DBID=3587895952)

RMAN> run
{
convert from platform 'AIX-Based Systems (64-bit)' datafile '/oracle/xtts/file5data_D-JONGS00_I-88982687_TS-
JONGS00_FNO-5_09tkgd8v.bkp' format '+DATA';
convert from platform 'AIX-Based Systems (64-bit)' datafile '/oracle/xtts/file6data_D-JONGS00_I-882>
982687_TS-JONGS00_FNO-6_0atkgd90.bkp' format '+DATA';
convert from platform 'AIX-Based Systems (64-bit)' datafile '/oracle/xtts/file7data_D-JONGS00_I-88982687_TS-
GOODUSDATA_FNO-7_08tkgd8s.bkp' format '+DATA';
}
3> 4> 5> 6>

Starting conversion at target at 27-MAR-18
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=36 instance=goodus1 device type=DISK
channel ORA_DISK_1: starting datafile conversion
input file name=/oracle/xtts/file5data_D-JONGS00_I-88982687_TS-JONGS00_FNO-5_09tkgd8v.bkp
converted datafile=+DATA/GOODUS/DATAFILE/jongsoo.269.971905275
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
Finished conversion at target at 27-MAR-18

Starting conversion at target at 27-MAR-18
using channel ORA_DISK_1
```

```

channel ORA_DISK_1: starting datafile conversion
input file name=/oracle/xtts/file6data_D-JONGS00_I-88982687_TS-JONGS00_FNO-6_0atkgd90.bkp
converted datafile=+DATA/GOODUS/DATAFILE/jongsoo.270.971905277
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
Finished conversion at target at 27-MAR-18

Starting conversion at target at 27-MAR-18
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile conversion
input file name=/oracle/xtts/file7data_D-JONGS00_I-88982687_TS-GOODUSDATA_FNO-7_08tkgd8s.bkp
converted datafile=+DATA/GOODUS/DATAFILE/goodusdata.271.971905279
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:01
Finished conversion at target at 27-MAR-18

RMAN>

```

2.9. As-Is Incremental 백업 테스트를 위한 추가 첫번째 데이터 생성

- 첫번째 증분백업을 위한 "middle" 데이터를 insert 합니다.

```

SQL> insert into jongsoo.test1 values ('middle');

1 row created.

SQL> commit;

Commit complete.

SQL> insert into goodusdata.test2 values ('middle');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from jongsoo.test1;

NAME
-----

```



```
before
```

```
middle
```

```
SQL> select * from goodusdata.test2;
```

```
NAME
```

```
-----
```

```
before
```

```
middle
```

2.10. As-Is 첫번째 추가 데이터를 위한 Incremental 백업

- "middle" 데이터를 백업하기 위해 증분 백업을 수행합니다.
- 이관 Tablespace 명을 입력합니다.

```
/data/jongsoo/file/incre1 $rman target /
```

```
Recovery Manager: Release 11.2.0.4.0 - Production on Tue Apr 9 15:22:38 2019
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: JONGSOO (DBID=88982687)
```

```
RMAN> run
```

```
{
```

```
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'full_backup' format  
'/data/jongsoo/file/incre1/INCRE%U.inc' tablespace JONGSOO, GOODUSDATA;
```

```
}2> 3> 4>
```

```
Starting backup at 2019-04-09:15:23:09
```

```
using target database control file instead of recovery catalog
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=21 device type=DISK
```

```
channel ORA_DISK_1: starting incremental level 1 datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```
input datafile file number=00007 name=/data/jongsoo/goodusdata01.dbf
```

```
input datafile file number=00005 name=/data/jongsoo/jongsoo01.dbf
```

```
input datafile file number=00006 name=/data/jongsoo/jongsoo02.dbf
```

```
channel ORA_DISK_1: starting piece 1 at 2019-04-09:15:23:13
```

```

channel ORA_DISK_1: finished piece 1 at 2019-04-09:15:23:14
piece handle=/data/jongsoo/file/incre1/INCRE0btuikf1_1_1.inc tag=FULL_BACKUP comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 2019-04-09:15:23:14

RMAN>

```

2.11. As-Is 첫번째 Incremental 백업 파일 To-Be 전송

- Incremental 백업 파일 To-Be 로 전송합니다.(SCP, FTP 등)

2.12. To-Be Endian Format에 맞게 Incremental backup 본 Convert 수행

- As-Is Platform ID 확인합니다.
- AIX-Based Systems (64-bit)는 6 번입니다.
- Incremental backup 본 파일 위치 지정합니다.
- Incremental convert 파일 위치 지정(새로 생성되는 파일) 합니다.
- As-Is Platform ID 를 지정합니다.

```
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 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
21 Apple Mac OS (x86-64)	Little

20 rows selected.

```

SQL> set serveroutput on;
set termout      on;
set verify      off;
DECLARE
  handle varchar2(512) ;
  comment varchar2(80) ;
  media varchar2(80) ;
  concur boolean ;
  recid number ;
  stamp number ;
  platfrmto number;
  same_endian number := 1;
  devtype VARCHAR2(512);
BEGIN
  BEGIN
    sys.dbms_backup_restore.restoreCancel(TRUE);
    devtype := sys.dbms_backup_restore.deviceAllocate;
    sys.dbms_backup_restore.backupBackupPiece(
      bpname => '/oracle/xtts/incre1/INCRE0btuikf1_1_1.inc',      -> Incremental backup 본 파일 위치 지정
      fname  => '/oracle/xtts/incre1/INCRE0btuikf1_1_1.inc.conv', -> Incremental convert 파일 위치 지정
      (새로 생성되는 파일)
      handle => handle, media => media, comment => comment,
      concur => concur, recid => recid, stamp => stamp, check_logical => FALSE,
      copyno => 1, deffmt => 0, copy_recid => 0, copy_stamp => 0,
      npieces => 1, dest => 0,
      pltfmfr => 6);      -> Platform ID 입력
  EXCEPTION
    WHEN OTHERS
    THEN
      DBMS_OUTPUT.put_line ('ERROR IN CONVERSION ' || SQLERRM);
  END ;
  sys.dbms_backup_restore.deviceDeallocate;
  DBMS_OUTPUT.put_line('CONVERTED BACKUP PIECE');

```

```

END;

/

CONVERTED BACKUP PIECE

PL/SQL procedure successfully completed.

```

2.13. To-Be Incremental backup 본 Recovery 수행

- 2.8 번 절차에서 To-Be 장비에 Convert 된 datafile 이름을 확인하고 dfnumber 는 As-Is File_id 지정, To-Be 파일 경로 지정, Incremental convert 파일 위치를 지정합니다.

```

SQL> set serveroutput on ;
DECLARE
d varchar2(512);
h varchar2(512) ;
t varchar2(30) ;
b1 boolean ;
b2 boolean ;
DONE boolean ;
FAILOVER boolean ;
BEGIN
d := sys.dbms_backup_restore.deviceAllocate;
DBMS_OUTPUT.PUT_LINE(d);
sys.dbms_backup_restore.applysetdatafile(check_logical=>false, cleanup=>false) ;
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>5 , toname =>
'+DATA/GOODUS/DATAFILE/jongsoo.269.971905275' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>6 , toname =>
'+DATA/GOODUS/DATAFILE/jongsoo.270.971905277' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>7 , toname =>
'+DATA/GOODUS/DATAFILE/goodusdata.271.971905279' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.restoreSetPiece( handle=>'/oracle/xtts/incre1/INCREObtuikf1_1_1.inc.conv',
tag=>null, fromdisk=>true, recid=>0, stamp=>0) ;
sys.dbms_backup_restore.restoreBackupPiece( done=>DONE, params=>null, outhandle=>h, outtag=>t,
FAILOVER=>FAILOVER); -> 빨간색만 수정. dfnumber 는 As-Is file_id 지정, To-Be 파일 경로 지정, Incremental
convert 파일 위치 지정
sys.dbms_backup_restore.restoreCancel(TRUE);

```

```
sys.dbms_backup_restore.deviceDeallocate;
END;
/

DISK

PL/SQL procedure successfully completed.
```

2.14. As-Is Incremental 백업 테스트를 위한 추가 두번째 데이터 생성

- 최종 "after" 데이터를 백업하기 위해 증분 백업을 수행합니다.
- 이관 Tablespace 명을 입력합니다.

```
SQL> insert into jongsoo.test1 values ('after');

1 row created.

SQL> commit;

Commit complete.

SQL> insert into goodusdata.test2 values ('after');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from jongsoo.test1;

NAME
-----
after
before
middle

SQL> select * from goodusdata.test2;

NAME
```

after

before

middle

2.15. As-Is Tablespace Read Only 모드로 변경(Down Time 시작)

- 앞단계에서 최종데이터라고 가정하였으므로 Read Only 모드로 변경합니다.
- 이 시점이 Down Time 시작입니다.
- 데이터 변경이 많은 시스템에서는 증분 백업을 여러 번 할수 있습니다.

```
SQL> alter tablespace JONGSOO read only;
```

Tablespace altered.

```
SQL> alter tablespace GOODUSDATA read only;
```

Tablespace altered.

2.16. As-Is 두번 째 추가 데이터를 위한 최종 Incremental 백업

- 최종 "after" 데이터를 백업하기 위해 증분 백업을 수행합니다.
- 이관 Tablespace 명을 입력합니다.

```
/data/jongsoo/file/last $rman target /
```

Recovery Manager: Release 11.2.0.4.0 - Production on Tue Apr 9 15:25:23 2019

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

connected to target database: JONGSOO (DBID=88982687)

```
RMAN> run
```

```
{
```

```
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'full_backup' format
```

```
'/data/jongsoo/file/last/INCRE%U.inc' tablespace JONGSOO, GOODUSDATA;
```

```
}2> 3> 4>
```

Starting backup at 2019-04-09:15:25:42

using target database control file instead of recovery catalog

allocated channel: ORA_DISK_1

channel ORA_DISK_1: SID=29 device type=DISK

```

channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00007 name=/data/jongsoo/goodusdata01.dbf
input datafile file number=00005 name=/data/jongsoo/jongsoo01.dbf
input datafile file number=00006 name=/data/jongsoo/jongsoo02.dbf
channel ORA_DISK_1: starting piece 1 at 2019-04-09:15:25:45
channel ORA_DISK_1: finished piece 1 at 2019-04-09:15:25:46
piece handle=/data/jongsoo/file/last/INCRE0ctuikjp_1_1.inc tag=FULL_BACKUP comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 2019-04-09:15:25:46

RMAN>

```

2.17. As-Is 두번째 Incremental 백업 파일 To-Be 전송

- Incremental 백업 파일 To-Be 로 전송합니다.(SCP, FTP 등)

2.18. To-Be 두번째 Incremental backup 본 최종 Recovery 수행

- Incremental backup 본 파일 위치 지정합니다.
- Incremental convert 파일 위치 지정(새로 생성되는 파일) 합니다.
- As-Is Platform ID 를 지정합니다.

```

SQL> set serveroutput on;
set termout on;
set verify off;
DECLARE
  handle varchar2(512) ;
  comment varchar2(80) ;
  media varchar2(80) ;
  concur boolean ;
  recid number ;
  stamp number ;
  platfrmt number;
  same_endian number := 1;
  devtype VARCHAR2(512);
BEGIN
  BEGIN
    sys.dbms_backup_restore.restoreCancel(TRUE);
    devtype := sys.dbms_backup_restore.deviceAllocate;
    sys.dbms_backup_restore.backupBackupPiece(

```

```

    bpname => '/oracle/xtts/last/INCRE0ctuikjp_1_1.inc',      -> Incremental backup 본 파일 위치 지정
    fname  => '/oracle/xtts/last/INCRE0ctuikjp_1_1.inc.conv', -> Incremental convert 파일 위치 지정(새
로 생성되는 파일)

    handle => handle, media => media, comment => comment,
    concur => concur, recid => recid, stamp => stamp, check_logical => FALSE,
    copyno => 1, deffmt => 0, copy_recid => 0, copy_stamp => 0,
    npieces => 1, dest => 0,
    pltfmfr => 6); -> Platform ID 입력

EXCEPTION
  WHEN OTHERS
  THEN
    DBMS_OUTPUT.put_line ('ERROR IN CONVERSION ' || SQLERRM);
END ;
sys.dbms_backup_restore.deviceDeallocate;
    DBMS_OUTPUT.put_line('CONVERTED BACKUP PIECE');
END;
/

CONVERTED BACKUP PIECE

PL/SQL procedure successfully completed.

```

2.19. To-Be Incremental backup 본 Recovery 수행

- 2.8 번 절차에서 To-Be 장비에 Convert 된 datafile 이름을 확인하고 dfnumber 는 As-Is File_id 지정, To-Be 파일 경로 지정, Incremental convert 파일 위치를 지정합니다.

```

SQL> set serveroutput on ;
DECLARE
d varchar2(512);
h varchar2(512) ;
t varchar2(30) ;
b1 boolean ;
b2 boolean ;
DONE boolean ;
FAILOVER boolean ;
BEGIN
d := sys.dbms_backup_restore.deviceAllocate;
DBMS_OUTPUT.PUT_LINE(d);
sys.dbms_backup_restore.applysetdatafile(check_logical=>false, cleanup=>false) ;

```



```

sys.dbms_backup_restore.applyDatafileTo( dfnumber=>5      , toname =>
'+DATA/GOODUS/DATAFILE/jongsoo.269.971905275' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>6      , toname =>
'+DATA/GOODUS/DATAFILE/jongsoo.270.971905277' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.applyDatafileTo( dfnumber=>7      , toname =>
'+DATA/GOODUS/DATAFILE/goodusdata.271.971905279' , fuzziness_hint=>0,max_corrupt
=>0, islevel0=>0, recid=>0, stamp=>0);
sys.dbms_backup_restore.restoreSetPiece( handle=>'/oracle/xtts/last/INCRE0ctuikjp_1_1.inc.conv' ,
tag=>null, fromdisk=>true, recid=>0, stamp=>0) ;
sys.dbms_backup_restore.restoreBackupPiece( done=>DONE, params=>null, outhandle=>h, outtag=>t,
FAILOVER=>FAILOVER); -> 빨간색만 수정. dfnumber 는 As-Is file_id 지정, To-Be 파일 경로 지정, Incremental
convert 파일 위치 지정
sys.dbms_backup_restore.restoreCancel(TRUE);
sys.dbms_backup_restore.deviceDeallocate;
END;
/

DISK

PL/SQL procedure successfully completed.

```

2.20. As-Is TTS Tablespace export

- Transport_tablespace 옵션을 사용하여 이관하고자 하는 Tablespace 을 export 합니다.
- 본 문서에는 Datapump 유틸리티를 사용하였습니다.

```

SQL> create or replace directory tts as '/data/jongsoo/tts';

Directory created.

SQL> grant read, write on directory data_dir to system;

Grant succeeded.

/data/jongsoo/tts $ expdp W'sys/oracle as sysdbaW' dumpfile=exptts.dmp logfile=exptts.log directory=tts
transport_tablespace=JONGSOO,GOODUSDATA

Export: Release 11.2.0.4.0 - Production on Tue Apr 9 15:27:31 2019

```

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

```
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Starting "SYS"."SYS_EXPORT_TRANSPORTABLE_01": "sys/***** AS SYSDBA" dumpfile=exptts.dmp
logfile=exptts.log directory=tts transport_tablespaces=JONGSOO,GOODUSDATA
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Master table "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/unloaded
*****
Dump file set for SYS.SYS_EXPORT_TRANSPORTABLE_01 is:
  /data/jongsoo/tts/exptts.dmp
*****
Datafiles required for transportable tablespace GOODUSDATA:
  /data/jongsoo/goodusdata01.dbf
Datafiles required for transportable tablespace JONGSOO:
  /data/jongsoo/jongsoo01.dbf
  /data/jongsoo/jongsoo02.dbf
Job "SYS"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Tue Apr 9 15:29:10 2019 elapsed 0
00:01:27

/data/jongsoo/tts $
```

2.21. As-Is expdp 파일 To-Be 전송

- expdp 파일 To-Be 로 전송합니다.(SCP, FTP 등)

2.22. TTS Tablespace import

- Transport_datafiles 옵션을 사용하여 이관하고자 하는 Tablespace 을 import 합니다.
- 본 문서에는 Datapump 유틸리티를 사용하였습니다.
- 이관하는 Tablespace 에 object 가 많을 경우 오랜 시간이 소요될 수 있습니다.

```
SQL> create or replace directory tts as '/oracle/xtts/exp';
```

```
Directory created.
```

```
SQL> grant read, write on directory tts to system;
```

```

Grant succeeded.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[oracle@rac01:/oracle/xtts/exp]$ impdp W'sys/oracle as sysdbaW' dumpfile=exptts.dmp logfile=imptts.log
directory=tts
transport_datafiles='+DATA/GOODUS/DATAFILE/jongsoo.269.971905275','+DATA/GOODUS/DATAFILE/jongsoo.270.971905277',
'+DATA/GOODUS/DATAFILE/goodusdata.271.971905279'

Import: Release 12.2.0.1.0 - Production on Tue Mar 27 22:20:46 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
Master table "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/unloaded
Starting "SYS"."SYS_IMPORT_TRANSPORTABLE_01": "sys/***** AS SYSDBA" dumpfile=exptts.dmp
logfile=imptts.log directory=tts
transport_datafiles=+DATA/GOODUS/DATAFILE/jongsoo.269.971905275,+DATA/GOODUS/DATAFILE/jongsoo.270.971905277,
+DATA/GOODUS/DATAFILE/goodusdata.271.971905279
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYS"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Tue Mar 27 22:21:20 2018 elapsed 0
00:00:16

[oracle@rac01:/oracle/xtts/exp]$

```

2.23. 추가작업

- 임시로 지정했던 Schema Default tablespace 를 원래 명으로 변경합니다.
- Import 가 끝나면 As-Is 에서 수행했던 Read Only 모드로 되어 있기 때문에 Read Write 모드로 변경합니다.

```

SQL> alter user jongsoo default tablespace JONGSOO;

User altered.

SQL> alter user goodusdata default tablespace GOODUSDATA;

User altered.

```

```
SQL> alter tablespace JONGSOO read write;
```

```
Tablespace altered.
```

```
SQL> alter tablespace GOODUSDATA read write;
```

```
Tablespace altered.
```

2.24. To-Be 데이터 확인

- 해당 과정이 끝나면 최종 데이터인 "after" 데이터를 확인 할수 있습니다.

```
SQL> select * from jongsoo.test1;
```

```
NAME
```

```
-----
```

```
after
```

```
before
```

```
middle
```

```
SQL> select * from goodusdata.test2;
```

```
NAME
```

```
-----
```

```
after
```

```
before
```

```
middle
```

- 필요 시 기타 object As-Is 에서 export 후 To-Be 에서 import 수행합니다.

3. 제약사항

3.1. expdp 유틸리티 사용 시 TRANSPORT_TABLESPACES 옵션

- 트랜스포터블 job 은 재기동 가능하지 않습니다.
- 트랜스포터블 job 은 parallelism degree 1 에 제한됩니다.
- 트랜스포터블 모드는 암호화된 컬럼을 지원하지 않습니다.
- SYSTEM 과 SYSAUX 테이블스페이스는 트랜스포터블 되지 않습니다.
- 트랜스포터블 집합 내에 있는 모든 테이블스페이스들은 read-only(읽기 전용) 모드로 셋팅되어야 합니다.
- Data Pump Export VERSION 파라미터가 TRANSPORT_TABLESPACES 파라미터와 같이 기술되면, 버전은 Oracle 데이터베이스 COMPATIBLE 초기화 파라미터보다 크거나 같아야 합니다.

3.2. Index Organized 테이블은 트랜스포터블 테이블스페이스 사용 시 손상될 수 있습니다.

- Index Organized 테이블은 Solaris, Linux 또는 AIX 에서 HP/UX 로 트랜스포터블 테이블스페이스 사용 시 손상될 수 있습니다. 현재 이 이슈에 대한 패치는 존재하지 않습니다. 이 문제를 만나면 Index Organized 테이블은 TTS 이후에 재생성해야 합니다.

3.3. SYSTEM 테이블스페이스나 SYS 사용자가 소유한 객체는 트랜스포트하는 것이 불가능합니다.

- SYSTEM Tablespace 객체. SYSTEM 테이블스페이스나 SYS 사용자가 소유한 객체는 트랜스포트 불가능합니다.

3.4. Character Set 과 National Character Set 제한사항

- 소스와 타겟 데이터베이스는 반드시 호환성이 있는 character set 과 national character set 을 사용해야 합니다.

3.5. 같은 이름을 갖는 테이블스페이스가 이미 존재하는 경우

- 같은 이름을 갖는 테이블스페이스가 이미 존재하는 타겟 데이터베이스에 테이블스페이스를 이전할 수 없습니다. 그러나, 트랜스포트 수행 전에 트랜스포트될 테이블스페이스나 destination 테이블스페이스를 rename 할 수 있습니다.

3.6. 타겟 서버에 owner가 new인 객체

- 만일 테이블스페이스 객체의 owner 가 타겟 데이터베이스에 존재하지 않으면, 트랜스포터블 테이블스페이스 import 전에 username 은 수동으로 생성해야 합니다.

3.7. XML 타입을 포함하는 테이블스페이스

- Oracle 데이터베이스 10g Release 2 부터 XML 타입을 갖는 테이블스페이스를 트랜스포트할 수 있습니다. 그러나, data pump 가 아니라, exp / imp 유틸리티를 사용해야 합니다.

3.8. 11.2에서는 exp / imp 방식은 지원되지 않습니다.

- TTS 프로시저를 수행하기 전에 11.2에서는 exp / imp 방식은 지원되지 않습니다.

3.9. 암호화된 테이블스페이스는 다음 제약사항을 갖습니다.

- 암호화된 테이블스페이스를 트랜스포트하기 이전에 만일 마스터 암호화 키가 하드웨어 보안 모듈 안에 저장되어 있지 않으면 오라클 wallet 을 타겟 데이터베이스에 수동으로 복사해야 합니다.

-
- 암호화된 테이블스페이스를 다른 endian 을 갖는 플랫폼으로 이전할 수 없습니다.
 - 암호화된 컬럼을 갖는 테이블을 갖는 테이블스페이스는 트랜스포트될 수 없습니다.

- 참고문서

- 트랜스포터블 테이블스페이스에 대한 마스터 노트 - 일반적인 질문 및 이슈들 (문서 ID 1532661.1)
- 트랜스포터블 테이블스페이스 (TTS) 제약사항, 제한: 세부사항, 참조, 그리고 적용 가능한 버전 (문서 ID 1541838.1)