**Oracle传输表空间操作手册**

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared by  拟制 |  | Date  日期 |  |
|  |  |  |  |
| Reviewed by  评审人 |  | Date  日期 |  |
| Approved by  批准 |  | Date  日期 |  |
| Authorized by  签发 |  | Date  日期 |  |

HighGO

All rights reserved

版权所有 侵权必究

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 描述 | 作者 |
| 2017-03-21 | V1.0 | 初稿完成 | 刘坤 |
| 2017-12-12 | V2.0 | 跨平台考虑  兼容性考虑  传输表空间-使用增量备份 | 咸士杰 |
|  |  |  |  |
|  |  |  |  |

## **一、Oracle传输表空间介绍**

使用传输表空间（Transportable Tablespaces）特性可以将一组表空间从一个数据库拷贝到另一个数据库

**Note:**

This method for transporting tablespaces requires that you place the tablespaces to be transported in read-only mode until you complete the transporting process. If this is undesirable, you can use the Transportable Tablespaces from Backup feature, described in [*Oracle Database Backup and Recovery User's Guide*](https://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmttbsb.htm#BRADV05141).

传输表空间技术始于oracle9i，不论是数据字典管理的表空间还是本地管理的表空间，都可以使用传输表空间技术；传输表空间不需要在源数据库和目标数据库之间具有同样的DB\_BLOCK\_SIZE块大小；使用传输表空间迁移数据比使用数据导入导出工具迁移数据的速度要快，这是因为传输表空间只是复制包含实际数据的数据文件到目标数据库的指定位置，而使用数据泵（Data Pump）则是传输表空间对象的元数据到新数据库。

## 二、跨平台考虑(XTTS)

从oracle 10g开始，oracle实现了跨平台的表空间传输（XTTS），跨平台的意味着数据库可以从一种类型的平台迁移到另一中类型的平台上，

大多数（但不是全部）的平台都支持传输表空间。首先必须通过查看v$transportable\_platform视图查看oracle支持的平台，并确定每种平台的字节存储次序

以下查询为oracle支持的各种平台及字节存储次序（版本为11.2.0.3），

SQL> col platform\_name for a40

SQL> SELECT \* FROM V$TRANSPORTABLE\_PLATFORM ORDER BY PLATFORM\_ID;

PLATFORM\_ID PLATFORM\_NAME ENDIAN\_FORMAT

----------- ------------------------------------ --------------

6 AIX-Based Systems (64-bit) Big

16 Apple Mac OS Big

21 Apple Mac OS (x86-64) Little

19 HP IA Open VMS Little

15 HP Open VMS Little

5 HP Tru64 UNIX Little

3 HP-UX (64-bit) Big

4 HP-UX IA (64-bit) Big

18 IBM Power Based Linux Big

9 IBM zSeries Based Linux Big

10 Linux IA (32-bit) Little

11 Linux IA (64-bit) Little

13 Linux x86 64-bit Little

7 Microsoft Windows IA (32-bit) Little

8 Microsoft Windows IA (64-bit) Little

12 Microsoft Windows x86 64-bit Little

17 Solaris Operating System (x86) Little

20 Solaris Operating System (x86-64) Little

1 Solaris[tm] OE (32-bit) Big

2 Solaris[tm] OE (64-bit) Big

20 rows selected.

如果源平台和目标平台有不同的字节序，需要一个额外的步骤，在源或目标平台转换表空间，以能够和目标平台格式一致，如果他们属于相同字节序，则不需要表空间转换，表空间可以被传输就像在同一种平台上一样

注意：为了能识别平台字节序，至少将数据文件置于读写状态一次。

**传输表空间限制：**

1 源和目标库必须有兼容的字符集

> 源和目标库字符集相同

> 源数据库字符集是目标数据库字符集的严格（二进制）子集，以下三个条件是真的：

源库版本为10.1.0.3及以上版本

要传输的表空间不包含具有字符长度语义的列, 或者源和目标数据库字符集中的最大字符宽度相同。

要传输的表空间不包含具有 CLOB 数据类型的列, 或者源和目标数据库字符集都是单字节或多字节。

> 源数据库字符集是目标数据库字符集的严格 (二进制) 子集, 以下两个条件为真

源数据库的版本低于10.1.0.3

源和目标数据库字符集中的最大字符宽度相同

2 源和目标数据库必须使用兼容的国家字符集。具体地说, 必须为下列情况之一:

> 源和目标数据库的国家字符集是相同的。

> 源数据库在10.1.0.3 或更高版本中, 要传输的表空间不包含具有 NCHAR、NVARCHAR2 或 NCLOB 数据类型的列。

3 不能将表空间传输到包含同名表空间的目标数据库。但是, 在传输操作之前, 您可以重命名要传输的表空间或目标表空间

4 具有基础对象 (如物化视图) 或包含的对象 (如已分区表) 的对象不可传输, 除非所有的基础对象或所包含的物体都同一个空间集中

5 加密表空间有限制

6 从 Oracle 数据库10g 版本2开始, 您可以传输包含 XMLTypes 的表空间。从 Oracle 数据库11g 版本1开始, 您必须只使用数据泵抽取来导出和导入包含 XMLTypes 的表空间的表空间元数据。

以下语句查询包括XMLTypes 的表空间

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.usernam

## 三、兼容性考虑

当创建最低传输表空间集时，oracle会判断目标数据库必须运行的最低兼容级别，这被称为可移动集的兼容级别。从 Oracle 数据库11g 开始, 无论目标数据库位于相同的平台上还是不同的系统上, 都可以将表空间传输到具有相同或更高兼容性设置的数据库。如果可移动集的兼容级别高于目标数据库的兼容级别, 则数据库会发出错误提示。

| **Transport Scenario** | **Minimum Compatibility Setting** | |
| --- | --- | --- |
| **Source Database** | **Destination Database** |
| Databases on the same platform | 8.0 | 8.0 |
| Tablespace with different database block size than the destination database | 9.0 | 9.0 |
| Databases on different platforms | 10.0 | 10.0 |

## 四、传输表空间-传统

注意：这种方法需要将表空间配置为read-only，如果这种情况不可取的, 可以使用称为可移动表空间从备份的替代方法

### 步骤

1 如果是跨平台的表空间传输，需要检查两个平台支持的字节存储顺序，检查方法见如上文所述，如果可以确定源数据库和目标数据库属于同一平台，可以省略此步骤

2 选择自包含的（self-contained）一组表空间，

在表空间传输的中，要求表空间集为自包含的，自包含表示用于传输的内部表空间集没有引用指向外部表空间集。自包含分为两种：一般自包含表空间集和完全（严格）自包含表空间集。

常见的以下情况是违反自包含原则的：

> 索引在内部表空间集，而表在外部表空间集（相反地，如果表在内部表空间集，而索引在外部表空间集，则不违反自包含原则）。

> 分区表一部分区在内部表空间集，一部分在外部表空间集（对于分区表，要么全部包含在内部表空间集中，要么全不包含）。

> 如果在传输表空间时同时传输约束，则对于引用完整性约束，约束指向的表在外部表空间集，则违反自包含约束；如果不传输约束，则与约束指向无关。

> 表在内部表空间集，而lob列在外部表空间集，则违反自包含约束。

3 将源数据库上的选定表空间修改为read-only状态，生成传输表空间集

表空间集：包括需要传输表空间中的所有数据文件和包含表空间元数据的导出文件（export file）

如果两个平台间的字节存储次序不同，还需完成字节存储次序的转换。可以在此步骤中源端完成转换，也可以在下一步骤中目标端完成转换。

4 传输表空间集

拷贝数据文件及导出文件到目标位置

5 将源数据库的表空间恢复为read-write状态（可选）  
6 在目标数据库，导入表空间集

使用数据泵工具导入元数据

### 示例：

Task 1 ：确认平台支持和字节序

如果跨平台，确认平台是否支持，

如果字节序不同，需要转换表空间集

SELECT d.PLATFORM\_NAME, ENDIAN\_FORMAT

FROM V$TRANSPORTABLE\_PLATFORM tp, V$DATABASE d

WHERE tp.PLATFORM\_NAME = d.PLATFORM\_NAME;

PLATFORM\_NAME ENDIAN\_FORMAT

------------------------------ --------------

Linux x86 64-bit Little

Task 2 ：选择自包含的一组表空间

表空间集之内和之外的对象可能存在逻辑或物理依赖，只能传输自包含的一组表空间。自包含的意思是表空间集内的对象没有引用表空间集外的对象

调用plsql程序包DBMS\_TTS检查违反自包含的情况。（TURE包含了完整性约束检查）

EXECUTE DBMS\_TTS.TRANSPORT\_SET\_CHECK('sales\_1,sales\_2', TRUE);

SQL> SELECT \* FROM TRANSPORT\_SET\_VIOLATIONS;

VIOLATIONS

---------------------------------------------------------------------------

Constraint DEPT\_FK between table JIM.EMP in tablespace SALES\_1 and table

JIM.DEPT in tablespace OTHER

Partitioned table JIM.SALES is partially contained in the transportable set

Task 3 ：生成传输表空间集

将表空间置为read-only模式

SQL> ALTER TABLESPACE sales\_1 READ ONLY;

SQL> ALTER TABLESPACE sales\_2 READ ONLY;

调用数据泵导出表空间结构信息

（transport\_tablespaces 确认导出的格式

transport\_full\_check参数严格检查自包含）

expdp system dumpfile=expdat.dmp directory=data\_pump\_dir

transport\_tablespaces=sales\_1,sales\_2 transport\_full\_check=y

logfile=tts\_export.log

如果需要字节序转换，调用rman工具使用CONVERT TABLESPACE命令

CONVERT TABLESPACE sales\_1,sales\_2 TO PLATFORM 'Microsoft Windows IA (32-bit)' FORMAT '/tmp/%U';

Task 4 ：传输表空间集

目标库查询DBA\_DIRECTORIES确认目录对象

传输数据文件到目标位置

如果目标位置位于ASM，建议使用如下工具

The DBMS\_FILE\_TRANSFER package

RMAN

数据文件是否需要转换

RMAN> CONVERT DATAFILE 'C:\Temp\sales\_101.dbf','C:\Temp\sales\_201.dbf' TO PLATFORM="Microsoft Windows IA (32-bit)" FROM PLATFORM="Solaris[tm] OE (32-bit)" DB\_FILE\_NAME\_CONVERT='C:\Temp\', 'C:\app\orauser\oradata\orawin\' PARALLELISM=4;

注意：如果两边路径使用的是ASM，需要使用参数FROM PLATFORM，TO PLATFORM，DB\_FILE\_NAME\_CONVERT

Task 5 ：恢复表空间到读写模式（可选）

ALTER TABLESPACE sales\_1 READ WRITE;

ALTER TABLESPACE sales\_2 READ WRITE;

Task 6 ：导入表空间集

impdp system parfile='par.f'

where the parameter file, par.f contains the following:

DUMPFILE=expdat.dmp

DIRECTORY=data\_pump\_dir

TRANSPORT\_DATAFILES=

C:\app\orauser\oradata\orawin\sales\_101.dbf,

C:\app\orauser\oradata\orawin\sales\_201.dbf

REMAP\_SCHEMA=sales1:crm1 REMAP\_SCHEMA=sales2:crm2

LOGFILE=tts\_import.log

TRANSPORT\_DATAFILES指定所有需要传输的数据文件，如果存在大量的数据文件建议使用parameter file

可以使用传输表空间技术迁移一个数据库到不同的平台。通过创建一个新库，然后迁移所有的用户表空间。你不能传输system表空间，因此比如说序列，plsql程序包或其他依赖系统表空间的对象，你必须手动创建他们，或者使用数据泵将这些对象迁移到目标库。

## 五、传输表空间-使用增量备份

当使用跨平台可移动表空间 (XTTS) 在具有不同字节格式的系统之间迁移数据时, 所需的停机时间可能会很大, 因为它与要移动的数据集的大小直接成正比。 但是, 将 XTTS 与跨平台增量备份相结合可以显著减少在平台之间移动数据所需的停机时间

因为传输表空间的过程开始需要将表空间配置为read-only模式，整个过程数据对与用户来说是不可用的，如果数据量很大，数据的传输和转换时间可能很长，因此停机时间可能很长。

减少宕机时间使用跨平台增量备份，oracle增强了rman使用增量备份前滚数据文件的能力，这可以用在XTTS场景中。通过一系列增量备份，在停机之前，使得源库和目标库数据基本一致，将XTTS与增量备份集合，数据文件传输和转换所需的停机时间与源库的数据库块更改（增量）成正比

### 步骤

1 准备阶段（源库数据保持在线）

> 传输数据文件到目标系统

> 如果需要，做数据文件转换

2 前滚阶段（源库数据保持在线，如果需要，多次前滚，以尽量保持目标库与源库一致）

> 在源库创建增量备份

> 传输增量备份到目标系统

> 转换增量备份为目标系统字节序格式和应用增量备份前滚数据

3 传输阶段（源库数据为只读read-only）

> 将源库表空间置为只读read-only

> 最后执行一次前滚操作

这一步使得目标库与源库数据一致

在处理大型数据时, 此步骤的时间比传统的 XTTS 方法要短得多, 因为增量备份大小较小。

> 使用数据泵导出源库表空间元数据

> 使用数据泵导入源库表空间元数据到目标库

> 将目标库表空间更改为读写read-write

注意：

如果从小字节序（little endian）平台迁移到oracle linux，优先考虑dataguard，参考mos[Note 413484.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1389592.1&id=413484.1)查看dataguard跨平台支持

12c方法有改动，请参考mos

[Note 2005729.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1389592.1&id=2005729.1) 12C - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup.

### 前提条件：

仅列出一些重要条件

> 当前版本不支持windows

> 传输的表空间不能包括offline的数据文件

> 源库oracle版本需要低于或等于目标库版本

> rman设备类型属性必须配置为DISK

> 准备阶段选择dbms\_file\_tranfer方法，目标库必须为11.2.0.4

> 准备阶段选择rman备份方法，在源和目标系统需要有临时区域

> If the destination database version is 11.2.0.3 or lower, then a separate database home containing 11.2.0.4 running an 11.2.0.4 instance on the destination system is required to perform the incremental backup conversion.  See the Destination Database 11.2.0.3 and Earlier Requires a Separate Incremental Convert Home and Instance section for details. If using ASM for 11.2.0.4 Convert Home, then ASM needs to be on 11.2.0.4, else error ORA-15295 (e.g. ORA-15295: ASM instance software version 11.2.0.3.0 less than client version 11.2.0.4.0) is raised.

### 准备阶段的方法

**1** dbms\_file\_transfer (DFT)

通过dbms\_file\_transfer.get\_file()程序包，使用database link传输数据文件，优点为1 不需要额外的空间 2 不需要单独的转换步骤，传输过程中自动换。但是目标库必须为11.2.0.4。

**2** Recovery Manager (RMAN) RMAN backup

需要额外的空间存放rman备份

### 目标数据库为11.2.0.3 或更早版本，需要单独的增量转换 Home 和实例

跨平台增量备份核心功能 (即增量备份转换) 在 Oracle 数据库11.2.0.4 和更高版本中提供。

如果目标库为11.2.0.4或更新版本，可以实现增量备份转换；如果目标库为11.2.0.3或更早版本，为了能够实现增量备份转换，需要一个单独的11.2.0.4软件home，并且有一个oracle实例需要启动到nomout阶段。

### 示例：

### Phase 1 初始化阶段

***Step 1.1 安装目标数据库软件和创建目标数据库***

强烈建议使用oracle database 11.2.0.4或更新版本

***Step 1.2 如果需要配置增量转换home和实例***

安装一个11.2.0.4的软件在目标系统

使用11.2.0.4软件目录启动实例到nomount阶段（不需要创建数据库）

[oracle@dest]$ export ORACLE\_HOME=/u01/app/oracle/product/11.2.0.4/xtt\_home

[oracle@dest]$ export ORACLE\_SID=xtt

[oracle@dest]$ cat << EOF > $ORACLE\_HOME/dbs/init$ORACLE\_SID.ora

db\_name=xtt

compatible=11.2.0.4.0

EOF

[oracle@dest]$ sqlplus / as sysdba

SQL> startup nomount

***Step 1.3 确认需要被传输的表空间***

***Step 1.4 如果使用dbms\_file\_transfer的方法，确认目录对象和dblinks***

1 源库目录对象是要传输（copy）数据文件所在目录

2 目标库目录对象为要传输（placed）数据文件所在目录

3 目标库中dbliks访问源库

SQL@source> create directory sourcedir as '+DATA/prod/datafile';

SQL@dest> create directory destdir as '+DATA/prod/datafile';

SQL@dest> create public database link ttslink connect to system identified by <password> using '<tns\_to\_source>';

SQL@dest> select \* from dual@ttslink;

***Step 1.5 创建临时区域***

使用rman备份方式，创建临时区域存储rman备份

***Step 1.6 安装xttconvert脚本在源库***

在源库，下载解压rman-xttconvert\_2.0.zip

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=1389592.1&_adf.ctrl-state=lwxk1uik5_241&_afrLoop=349937799466560>

[oracle@source]$ unzip rman\_xttconvert\_v3.zip

Archive: rman\_xttconvert\_v3.zip

inflating: xtt.properties

inflating: xttcnvrtbkupdest.sql

inflating: xttdbopen.sql

inflating: xttdriver.pl

inflating: xttprep.tmpl

extracting: xttstartupnomount.sql

***Step 1.7 配置脚本xtt.properties在源库***

更改文件xtt.properties，配置需要传输的表空间

***Step 1.8 拷贝xttconvert脚本和xtt.properties到目标系统***

[oracle@source]$ scp -r /home/oracle/xtt dest:/home/oracle/xtt

***Step 1.9 – 设置TMPDIR***

在shell环境，在源和目标系统设置TMPDIR环境变量为支持脚本所在的路径

[oracle@source]$ export TMPDIR=/home/oracle/xtt

[oracle@dest]$ export TMPDIR=/home/oracle/xtt

注：如果没有这个环境变量，运行脚本xttdriver.pl生成的文件会写入到/tmp

### Phase 2 准备阶段

使用脚本xttdriver.pl传输和转换表空间集，有两个方法

Phase 2A - dbms\_file\_transfer Method

Phase 2B - RMAN Backup Method

注意：更大数据量使用Phase 2A更快

***Phase 2A 准备阶段针对dbms\_file\_transfer方法***

将数据文件传输到目标系统最终位置，如果需要转换，自动做单独的转换操作，

***Step 2A.1 在源系统运行准备步骤***

[oracle@source]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -S

准备脚本实现以下动作

> 确认表空间是online的，是读写read-write状态的，不包括offline的数据文件

> 创建以下的脚本文件

xttnewdatafiles.txt

getfile.sql

***Step 2A.2 传输数据文件到目标系统***

在目标系统用oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)，如果没有配置的话，不会尝试使用增量转换实例。从源库拷贝xttnewdatafiles.txt and getfile.sql文件。并使用-G 参数运行命令

[oracle@dest]$ scp oracle@source:/home/oracle/xtt/xttnewdatafiles.txt /home/oracle/xtt

[oracle@dest]$ scp oracle@source:/home/oracle/xtt/getfile.sql /home/oracle/xtt

# MUST set environment to destination database

[oracle@dest]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -G

当这一不完成，所有数据文件都被传输到目标端，如果需要，会自动转换字节序

***Phase 2B 准备阶段针对rman备份方法***

在源库创建表空间备份，传输到目标系统，字节序转换，恢复数据文件到目标位置

***Step 2B.1 在源库运行准备步骤***

用oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)

[oracle@source]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -p

准备脚本实现以下动作

> 创建要传输表空间中 数据文件映像副本 备份到指定位置，位置由文件xtt.properties参数dfcopydir指定

> 确认表空间是online的，是读写read-write状态的，不包括offline的数据文件

> 创建以下的文件

xttplan.txt

rmanconvert.cmd

***Step 2B.2 传输数据文件映像副本到目标系统***

用oracle用户登陆源库数据文件映像副本会被创建在指定位置（xtt.properties参数dfcopydir指定），映像副本需要被转换到目标位置（xtt.properties参数stageondest指定）

使用以下命令将源库映像副本传输到目标系统 目标位置

[oracle@dest]$ scp oracle@source:/stage\_source/\* /stage\_dest

***Step 2B.3 转换数据文件字节序在目标系统***

用oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)在目标系统。拷贝 rmanconvert.cmd从源库。运行以下命令转换数据文件字节序

[oracle@dest]$ scp oracle@source:/home/oracle/xtt/rmanconvert.cmd /home/oracle/xtt

[oracle@dest]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -c

Convert数据文件的步骤转换数据文件映像副本到目标位置（xtt.properties参数stageondest指定）

### Phase 3 前滚阶段

在这个阶段，增量备份从源库创建，传输到目标系统，转换目标系统字节序格式，应用增量备份前滚

***Step 3.1 生成需要传输表空间的增量备份在源库***

在源库oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)，运行以下命令

[oracle@source]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -i

这个脚本调用rman命令生成增量备份（all tablespaces listed in xtt.properties）并生成以下的文件

tsbkupmap.txt

ncrbackups.txt

***Step 3.2 传输增量备份到目标系统***

增量备份列表信息在源库的incrbackups.txt文件中，传输文件及备份目标系统

[oracle@source]$ scp `cat incrbackups.txt` oracle@dest:/stage\_dest

***Step 3.3 转换增量备份并应用增量备份在目标库***

在目标库oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)，从源库拷贝xttplan.txt and tsbkupmap.txt文件，并运行命令前滚数据文件

[oracle@dest]$ scp oracle@source:/home/oracle/xtt/xttplan.txt /home/oracle/xtt

[oracle@dest]$ scp oracle@source:/home/oracle/xtt/tsbkupmap.txt /home/oracle/xtt

[oracle@dest]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -r

以上应用命令包括转换步骤，前滚数据文件步骤，以sys用户连接到增量恢复实例，转换增量备份，然后连接至目标库并应用增量备份。

注意：

1 每次执行增量恢复都需要拷贝文件xttplan.txt and tsbkupmap.txt，因为每次内容都会变化

2 注意不要更改、拷贝、修改脚本生成的文件xttplan.txt.new

3 这个过程中，目标实例会重启

***Step 3.4 确认FROM\_SCN为下一次增量备份***

在源库oracle用户登陆并设置环境变量(ORACLE\_HOME and ORACLE\_SID environment variables)。运行以下命令确认最新FROM\_SCN

[oracle@source]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -s

新FROM\_SCN记录在文件xttplan.txt中，然后在下一次增量备份时使用

***Step 3.5 重复前滚操作或跳至下一阶段***

重复应用前滚操作，尽可能保持目标库与源库数据一致

注意：如果操作过程期间有新的数据文件被添加，执行以下的操作

NOTE: If a datafile is added to one a tablespace since last incremental backup and/or a new tablespace name is added to the xtt.properties, the following will appear:

Error:  
------  
The incremental backup was not taken as a datafile has been added to the tablespace:

Please Do the following:  
--------------------------  
1. Copy fixnewdf.txt from source to destination temp dir

2. Copy backups:  
<backup list>  
from <source location> to the <stage\_dest> in destination

3. On Destination, run $ORACLE\_HOME/perl/bin/perl xttdriver.pl --fixnewdf

4. Re-execute the incremental backup in source:  
$ORACLE\_HOME/perl/bin/perl xttdriver.pl --bkpincr

NOTE: Before running incremental backup, delete FAILED in source temp dir or  
run xttdriver.pl with -L option:

$ORACLE\_HOME/perl/bin/perl xttdriver.pl -L --bkpincr

These instructions must be followed exactly as listed. The next incremental backup will include the new datafile.

### Phase 4 传输阶段

这个阶段需要将源库数据置于只读read-only模式。通过应用最终的增量备份使目标库与源库数据保持一致，使数据一致后，传统传输表空间的步骤，用数据泵导出导入元数据的操作需要被操作。只读read-only模式直到整个阶段完成

***Step 4.1 将源库表空间配置为只读 read-only***

system@source/prod SQL> alter tablespace TS1 read only;

system@source/prod SQL> alter tablespace TS2 read only;

***Step 4.2 创建最终的增量备份，传输，转换，应用到目标库***

最后一次执行3.1-3.3步骤

[oracle@source]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -i

[oracle@source]$ scp `cat incrbackups.txt` oracle@dest:/stage\_dest

[oracle@source]$ scp xttplan.txt oracle@dest:/home/oracle/xtt

[oracle@source]$ scp tsbkupmap.txt oracle@dest:/home/oracle/xtt

[oracle@dest]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -r

***Step 4.3 导入元数据到目标库***

运行以下TTS命令生成 data pump 文件

[oracle@dest]$ $ORACLE\_HOME/perl/bin/perl xttdriver.pl -e

The generate Data Pump TTS command step creates a sample Data Pump network\_link transportable import command in the file xttplugin.txt with the transportable tablespaces parameters TRANSPORT\_TABLESPACES and TRANSPORT\_DATAFILES correctly set.  Note that network\_link mode initiates an import over a database link that refers to the source database.  A separate export or dump file is not required.  If you choose to perform the tablespace transport with this command, then you must edit the import command to replace import parameters DIRECTORY, LOGFILE, and NETWORK\_LINK with site-specific values.

The following is an example network mode transportable import command:

[oracle@dest]$ impdp directory=DATA\_PUMP\_DIR logfile=tts\_imp.log network\_link=ttslink \

transport\_full\_check=no \

transport\_tablespaces=TS1,TS2 \

transport\_datafiles='+DATA/prod/datafile/ts1.285.771686721', \

'+DATA/prod/datafile/ts2.286.771686723', \

'+DATA/prod/datafile/ts2.287.771686743'

注意：传输表空间中对象所属数据库用户必须存在于目标库中

***Step 4.4 将源库表空间置于读写模式***

system@dest/prod SQL> alter tablespace TS1 read write;

system@dest/prod SQL> alter tablespace TS2 read write;

***Step 4.5 验证传输数据***

在这一步，目标库的传输数据是read-only状态，应用开发确认传输数据的正确性

同时，通过以下rman命令确认是否存在物理和逻辑块损坏

RMAN> validate tablespace TS1, TS2 check logical;

### Phase 5 清理

如果有配置单独的增量转换home和实例，实例需要被关闭，软件需要被卸载

以下被创建的文件不再需要，可删除

* dfcopydir location on the source system
* backupformat location on the source system
* stageondest location on the destination system
* backupondest location on the destination system
* $TMPDIR location in both destination and source systems

## 附录1 部分脚本参数参考

#### *Description of Perl Script xttdriver.pl Options*

 The following table describes the options available for the main supporting script xttdriver.pl.

|  |  |
| --- | --- |
| **Option** | **Description** |
| -S prepare source for transfer | -S option is used only when Prepare phase method is dbms\_file\_transfer.  Prepare step is run once on the source system during Phase 2A with the environment (ORACLE\_HOME and ORACLE\_SID) set to the source database.  This step creates files xttnewdatafiles.txt and getfile.sql. |
| -G get datafiles from source | -G option is used only when Prepare phase method is dbms\_file\_transfer.  Get datafiles step is run once on the destination system during Phase 2A with the environment (ORACLE\_HOME and ORACLE\_SID) set to the destination database.  The -S option must be run beforehand and files xttnewdatafiles.txt and getfile.sql transferred to the destination system.  This option connects to the destination database and runs script getfile.sql.  getfile.sql invokes dbms\_file\_transfer.get\_file() subprogram for each datafile to transfer it from the source database directory object (defined by parameter srcdir) to the destination database directory object (defined by parameter dstdir) over a database link (defined by parameter srclink). |
| -p prepare source for backup | -p option is used only when Prepare phase method is RMAN backup.  Prepare step is run once on the source system during Phase 2B with the environment (ORACLE\_HOME and ORACLE\_SID) set to the source database.  This step connects to the source database and runs the xttpreparesrc.sql script once for each tablespace to be transported, as configured in xtt.properties.  xttpreparesrc.sql does the following:   1. Verifies the tablespace is online, in READ WRITE mode, and contains no offline datafiles. 2. Identifies the SCN that will be used for the first iteration of the incremental backup step and writes it into file $TMPDIR/xttplan.txt. 3. Creates the initial datafile copies on the destination system in the location specified by the parameter dfcopydir set in xtt.properties.  These datafile copies must be transferred manually to the destination system. 4. Creates RMAN script $TMPDIR/rmanconvert.cmd that will be used to convert the datafile copies to the required endian format on the destination system. |
| -c convert datafiles | -c option is used only when Prepare phase method is RMAN backup.  Convert datafiles step is run once on the destination system during Phase 2B with the environment (ORACLE\_HOME and ORACLE\_SID) set to the destination database.  This step uses the rmanconvert.cmd file created in the Prepare step to convert the datafile copies to the proper endian format.  Converted datafile copies are written on the destination system to the location specified by the parameter storageondest set in xtt.properties. |
| -i create incremental | Create incremental step is run one or more times on the source system with the environment (ORACLE\_HOME and ORACLE\_SID) set to the source database.  This step reads the SCNs listed in $TMPDIR/xttplan.txt and generates an incremental backup that will be used to roll forward the datafile copies on the destination system. |
| -r rollforward datafiles | Rollforward datafiles step is run once for every incremental backup created with the environment (ORACLE\_HOME and ORACLE\_SID) set to the destination database.  This step connects to the incremental convert instance using the parameters cnvinst\_home and cnvinst\_sid, converts the incremental backup pieces created by the Create Incremental step, then connects to the destination database and rolls forward the datafile copies by applying the incremental for each tablespace being transported. |
| -s determine new FROM\_SCN | Determine new FROM\_SCN step is run one or more times with the environment (ORACLE\_HOME and ORACLE\_SID) set to the source database. This step calculates the next FROM\_SCN, records it in the file xttplan.txt, then uses that SCN when the next incremental backup is created in step 3.1. It reports the mapping of the new FROM\_SCN to wall clock time to indicate how far behind the changes in the next incremental backup will be. |
| -e generate Data Pump TTS command | Generate Data Pump TTS command step is run once on the destination system with the environment (ORACLE\_HOME and ORACLE\_SID) set to the destination database.  This step creates the template of a Data Pump Import command that uses a network\_link to import metadata of objects that are in the tablespaces being transported. |
| -d debug | -d option enables debug mode for xttdriver.pl and RMAN commands it executes.  Debug mode can also be enabled by setting environment variable XTTDEBUG=1. |
|  |  |

#### *Description of Parameters in Configuration File xtt.properties*

The following table describes the parameters defined in the xtt.properties file that is used by xttdriver.pl.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Example Setting** |
| tablespaces | Comma-separated list of tablespaces to transport from source database to destination database. Must be a single line, any subsequent lines will not be read. | tablespaces=TS1,TS2 |
| platformid | Source database platform id, obtained from V$DATABASE.PLATFORM\_ID. | platformid=2 |
| srcdir | Directory object in the source database that defines where the source datafiles currently reside. Multiple locations can be used separated by ",". The srcdir to dstdir mapping can either be N:1 or N:N. i.e. there can be multiple source directories and the files will be written to a single destination directory, or files from a particular source directory can be written to a particular destination directory.  This parameter is used only when Prepare phase method is dbms\_file\_transfer. | srcdir=SOURCEDIR  srcdir=SRC1,SRC2 |
| dstdir | Directory object in the destination database that defines where the destination datafiles will be created.  If multiple source directories are used (srcdir), then multiple destinations can be defined so a particular source directory is written to a particular destination directory.  This parameter is used only when Prepare phase method is dbms\_file\_transfer. | dstdir=DESTDIR  dstdir=DST1,DST2 |
| srclink | Database link in the destination database that refers to the source database.  Datafiles will be transferred over this database link using dbms\_file\_transfer.  This parameter is used only when Prepare phase method is dbms\_file\_transfer. | srclink=TTSLINK |
| dfcopydir | Location on the source system where datafile copies are created during the "-p prepare" step.   This location must have sufficient free space to hold copies of all datafiles being transported.  This location may be an NFS-mounted filesystem that is shared with the destination system, in which case it should reference the same NFS location as the stageondest parameter for the destination system.  See [Note 359515.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1389592.1&id=359515.1) for mount option guidelines.  This parameter is used only when Prepare phase method is RMAN backup. | dfcopydir=/stage\_source |
| backupformat | Location on the source system where incremental backups are created.    This location must have sufficient free space to hold the incremental backups created for one iteration through the process documented above.  This location may be an NFS-mounted filesystem that is shared with the destination system, in which case it should reference the same NFS location as the stageondest parameter for the destination system. | backupformat=/stage\_source |
| stageondest | Location on the destination system where datafile copies are placed by the user when they are transferred manually from the source system.   This location must have sufficient free space to hold copies of all datafiles being transported.  This is also the location from where datafiles copies and incremental backups are read when they are converted in the "-c conversion of datafiles" and "-r roll forward datafiles" steps.  This location may be a DBFS-mounted filesystem.  This location may be an NFS-mounted filesystem that is shared with the source system, in which case it should reference the same NFS location as the dfcopydir and backupformat parameters for the source system.  See [Note 359515.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1389592.1&id=359515.1) for mount option guidelines. | stageondest=/stage\_dest |
| storageondest | Location on the destination system where the converted datafile copies will be written during the "-c conversion of datafiles" step.   This location must have sufficient free space to permanently hold the datafiles that are transported.  This is the final location of the datafiles where they will be used by the destination database.  This parameter is used only when Prepare phase method is RMAN backup. | storageondest=+DATA - or - storageondest=/oradata/prod/%U |
| backupondest | Location on the destination system where converted incremental backups on the destination system will be written during the "-r roll forward datafiles" step.   This location must have sufficient free space to hold the incremental backups created for one iteration through the process documented above.  NOTE: If this is set to an ASM location then define properties asm\_home and asm\_sid below. If this is set to a file system location, then comment out asm\_home and asm\_sid parameters below. | backupondest=+RECO |
| cnvinst\_home | Only set this parameter if a separate incremental convert home is in use.  ORACLE\_HOME of the incremental convert instance that runs on the destination system. | cnvinst\_home=/u01/app/oracle/product/11.2.0.4/xtt\_home |
| cnvinst\_sid | Only set this parameter if a separate incremental convert home is in use.  ORACLE\_SID of the incremental convert instance that runs on the destination system. | cnvinst\_sid=xtt |
| asm\_home | ORACLE\_HOME for the ASM instance that runs on the destination system.  NOTE: If backupondest is set to a file system location, then comment out both asm\_home and asm\_sid. | asm\_home=/u01/app/11.2.0.4/grid |
| asm\_sid | ORACLE\_SID for the ASM instance that runs on the destination system. | asm\_sid=+ASM1 |
| parallel | Defines the degree of parallelism set in the RMAN CONVERT command file rmanconvert.cmd. This file is created during the prepare step and used by RMAN in the convert datafiles step to convert the datafile copies on the destination system.  If this parameter is unset, xttdriver.pl uses parallel=8.  NOTE: RMAN parallelism used for the datafile copies created in the RMAN Backup prepare phase and the incremental backup created in the rollforward phase is controlled by the RMAN configuration on the source system. It is not controlled by this parameter. | parallel=3 |
| rollparallel | Defines the level of parallelism for the -r roll forward operation. | rollparallel=2 |
| getfileparallel | Defines the level of parallelism for the -G operation  Default value is 1. Maximum supported value is 8. | getfileparallel=4 |

## 附录2 已知问题

### Known Issue Known Issues for Cross Platform Transportable Tablespaces XTTS [Document 2311677.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1389592.1&id=2311677.1)

#### *Change History*

|  |  |
| --- | --- |
| **Change** | **Date** |
| rman\_xttconvert\_v3.zip released - adds support for added datafiles | 2017-Jun-06 |
| rman-xttconvert\_2.0.zip released - add support for multiple source and destination directories | 2015-Apr-20 |
| rman-xttconvert\_1.4.2.zip released - add parallelism support for -G get file from source operation | 2014-Nov-14 |
| rman-xttconvert\_1.4.zip released - remove staging area requirement, add parallel rollforward, eliminate conversion instance requirements when using 11.2.0.4. | 2014-Feb-21 |
| rman-xttconvert\_1.3.zip released - improves handling of large databases with large number of datafiles. | 2013-Apr-10 |

## 附录3 参考：

https://docs.oracle.com/cd/E11882\_01/server.112/e25494/tspaces.htm#ADMIN10140

Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable(1454872.1)

Known Issues for Cross Platform Transportable Tablespaces XTTS (Doc ID 2311677.1)

11G - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup (Doc ID 1389592.1)