

数据泵和db link 联合使用实现数据迁移

参考：<http://blog.itpub.net/28612416/viewspace-764177/>
<http://czmmiao.iteye.com/blog/2041703>

使用impdp/expdp。只是不再将数据导出后导入，而是直接将数据从源库导入到目的库。

```
linux$oracle$> Impdp system/password@目标库 directory=DMPDIR schemas=TESTI network_link=目标库上建 的dblink  
remap_schema=TESTI:TESTA
```

说明：directory定义的路径是在目标库上定义的路径。network_link同样是在目标库上定义的tnsnames.ora中的信息。

上面语句的操作是将源库的TESTI用户的数据，导入到目标库的TESTA用户下。

如果从原库导出schema A，且db_link建立在schema A上，则原库的该schema A用户需具有exp_full_database权限否则会报错：

With the Partitioning, OLAP and Data Mining options

ORA-31631: privileges are required

ORA-39149: cannot link privileged user to non-privileged user

- 1、这个操作是局域网内迁移数据最方便的工具，不过也可能是速度最慢的工具。
- 2、同时还可此方法导表空间，单独的表等等.... tablespaces=xxx_tbs即可。...
- 3、在目标库上建立到源端的db_link的时候，可以针对system用户建立，这样就可以导出导入全库数据或者表空间数据。
- 4、当针对某个用户A创建db_link时，需要给该用户Aexp_full_database的权限才可以导出该schema得数据。
- 5、在导入的过程中注意目标数据库存在表数据的情况，可采用table_exists_action来处理。

ORACLE10G提供了新的导入导出工具，数据泵。

Oracle官方对此的形容是：Oracle DataPump technology enables Very High-Speed movement of data and metadata from one database to another. 其中Very High-Speed是亮点。

先说数据泵提供的主要特性（包括，但不限于）：

1. 支持并行处理导入、导出任务
2. 支持暂停和重启动导入、导出任务
3. 支持通过Database Link的方式导出或导入远端数据库中的对象
4. 支持在导入时通过Remap_schema、Remap_datafile、Remap_tablespace几个参数实现导入过程中自动修改对象属主、数据文件或数据所在表空间。
5. 导入/导出时提供了非常细粒度的对象控制。通过Include、Exclude两个参数，甚至可以详细制定是否包含或不包含某个对象。

Warning:

1. 什么是Directory对象

Directory对象是Oracle10g版本提供的一个新功能。他是一个 指向，指向了操作系统中的一个路径。每个Directory都包含 Read,Write两个权限，可以通过Grant命令授权给指定的用户或角色。拥有读写权限的用户就可以读写该Directory对象指定的操作系统路 径下的文件。

2. 除了使用network_link参数意外，expdp生成的文件都是在服务器上(Directory指定的位置)

如何调用

1. 命令行方式

最简单的调用，但是写的参数有限，建议使用参数文件的方式。

2. 参数文件方式

最常用的方式。通常需要先编写一个参数文件。指定导出时需要的各种参数。然后以如下方式调用。

```
expdp user/pwd parfile=xxx.par
```

这个xxx.par即是我们编辑的参数文件。注意，在这个命令行后面，同样可以再跟别的参数， 甚至是在par参数文件中指定过的参数。如果执行命令中附加 的参数与参数文件中的参数有重复，最终采用哪个参数，会以参数最后出现的位置而定。如：expdp user/pwd parfile=xxx.par logfile=a.log，如果在参数文件中也指定了logfile，这里会以命令行中的logfile为准；如：expdp user/pwd logfile=a.log parfile=xxx.par，而这个，则会以参数文件中的为准，因为parfile=xxx.par写在命令行的后面。

3. 交互方式

Data Pump导入导出任务支持停止，重启等状态操作。如用户执行导入或者导出任务，执行了一半时，使用Ctrl+C中断了任务（或其他原因导致的中断），此时 任务并不是被取消，而是被转移到后台。可以再次使用expdp/impdp命令，附加attach参数的方式重新连接到中断的任务中，并选择后续的操作。 这就是交互方式。

Warning：什么是attach参数，每执行一个导入，或者导出，在命令的第一行，会有以下信息：Starting “BAM”. SYS_EXPORT_SCHEMA_01”：bam/***** parfile=expdp_tbs.par，这个SYS_EXPORT_SCHEMA_01就是我们的attach参数。

```
-bash-3.00$ expdp bam/bam parfile=expdp_tbs.par
```

```
Export: Release 10.2.0.4.0 - 64bit Production on Friday, 13 August, 2010 16:35:18
```

```
Copyright (c) 2003, 2007, Oracle. All rights reserved.
```

```
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production
```

```
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

Starting “BAM”. SYS_EXPORT_SCHEMA_01”: bam/***** parfile=expdp_tbs.par

如果想使用交互方式，可以使用如：expdp attach SYS_EXPORT_SCHEMA_01 进入到交互模式

操作模式

1. 全库模式

导入或者导出整个数据库，对应impdp/expdp命令中的full参数，只有拥有dba或者 exp_full_database和imp_full_database权限的用户才能执行。

2. Schema模式

导出或导入Schema下的自有对象，对应impdp/expdp命令中的Schema参数，这是默认的操作模式。 如果拥有dba或者 exp_full_database和 imp_full_database权限的用户执行的话，就可以导出或导入多个Schema中的对象。

3. 表模式

导出指定的表或者表分区（如果有分区的话）以及依赖该表的对象（如该表的索引，约束等，不过前提是这些对象在同一个 Schema中，或者执行的用户有相应的权限）。对应impdp/expdp命令中的Table参数。

4. 表空间模式

导出指定的表空间中的内容。对应impdp/expdp中的Tablespaces参数，这种模式类似于表模式和 Schema模式的补充。

5. 传输表空间模式

对应impdp/expdp中的Transport_tablespaces参数。这种模式与前面几种模式最显著的区 别是生成的Dump文件中并不包含具 体的逻辑数据，而只 导出相关对象的元数据(即对象的定义，可以理解成表的创建语句)，逻辑数据仍然在表空间的数据文件中，导出时需要将元数据和数据文件同 时复 制到目标端服务器。

这种导出方式效率很高，时间开销主要是花在复制数据文件产生的I/O上。expdp执行传输表空间模式的导出，用户必须 拥有 exp_full_database角 色或者DBA角色。而通过传输表空间模式导入时，用户必须拥有imp_full_database角色或者DBA角 色。

过滤数据

过滤数据主要依赖于Query和Sample两个参数。其中Sample参数主要针对expdp导出功能。

1. Query

与exp命令中的Query功能类似，不过Expdp中，该参数功能得到了增强，控制的粒度更细。Expdp中的Query也是指定类 似where语句来限定记录。语 法如下：

Query = [Schema.][Table_name:] Query_clause

默认 情况如果不指定Schema.table_name,则Query_clause针对所有导出的表有效，或者你可以为每一个表指定不同的 Query_clause，如：导出a表 中所有id<5的记录，导出b表中所有name=' a' 的记录，则Query的参数应该如下：

Query=A:" Where id<5",B:" Where name=' a' "

如果Where条件前没有指定Schema名或者表名的话，默认就是针对当前所有要被 导出的表。如：Query=Where id <5

Warning: 建议把Query参数放入到参数文件中使用，以避免转义符带来的麻烦。

2. Sample

该参数用来指定导出数据的百分比，可指定的值的范围从0.000001到99.999999，语法如下：

Sample=[[Schema_name.]Table_name:]sample_percent

指 定该参数以后，EXPDP导出将自动控制导出的记录量，如导出A表中50%的记录，设置的Sample参数如下：

Sample=A:50

Warning: Sample_percent指定的值只是一个参考值，EXPDP会根据数据量算出一个近似值。

过滤对象

过滤对象主要依赖于Include和Exclude两个参数。这两个参数作用正好相反，在这两个参数中，可以指定你知道的任何对象 类型（如：Package、 Procedure、Table等等）或者对象名称（支持通配符）

1. Exclude 反规则

指定不被包含的对象类型或者对象名称。指定了该参数以后，指定的对象类型对应的所有对象都不会被导入或导出。 如果被排除的对象有依赖的 对象，那么其依赖的 对象也不会被导入或导出。如：通过Exclude参数指定不导出表对象的话，不仅指定的表不会被导出，连这些表关联的Index、 Check等都不会被导出。Warning: 建议把Exclude参数放入到参数文件中使用，以避免转义符带来的麻烦。

2. Include 正规则

与Exclude正好相反。指定包含的对象类型或者对象名称。

Warning: 由于两个参数功能正好相反，因此在执行导入或导出命令时，两个参数不能同时使用，否则Oracle也不知道你想要干什么啦。

exclude/include参数用法：

EXCLUDE=[object_type]:[name_clause],[object_type]:[name_clause] -->排出特定对象

INCLUDE=[object_type]:[name_clause],[object_type]:[name_clause] -->包含特定对象

object_type子句用于指定对象的类型，如table, sequence, view, procedure, package等等

name_clause子句可以为SQL表达式用于过滤特定的对象名字。它由SQL操作符以及对象名(可使用通配符)来过滤指定对象类型中的特定对象。

当未指定name_clause而仅仅指定object_type则所有该类型的对象都将被过滤或筛选。多个[object_type]:[name_clause]中间以逗号分割。

示例：

expdp <other_parameters> SCHEMAS=scott EXCLUDE=SEQUENCE, TABLE:"IN (' EMP', 'DEPT')"

impdp <other_parameters> SCHEMAS=scott INCLUDE=PACKAGE, FUNCTION, PROCEDURE, TABLE:"=' EMP' "

常用的过滤SQL表达式

EXCLUDE=SEQUENCE, VIEW --过滤所有的SEQUENCE, VIEW

EXCLUDE=TABLE:"IN (' EMP', 'DEPT') " --过滤表对象EMP, DEPT

EXCLUDE=SEQUENCE, VIEW, TABLE:"IN ('EMP', 'DEPT')"
--过滤所有的SEQUENCE, VIEW以及表对象EMP, DEPT
EXCLUDE=INDEX:"=' INDX_NAME'"
--过滤指定的索引对象INDX_NAME
INCLUDE=PROCEDURE:"LIKE 'PROC_U%'"
--包含以PROC_U开头的所有存储过程(_ 符号代表任意单个字符)
INCLUDE=TABLE:"> 'E'"
--包含大于字符E的所有表对象

其它常用操作符 NOT IN, NOT LIKE, <, != 等等

直接将过滤操作符封装到参数文件中, 如下面的例子

```
Parameter file:exp_scott.par  
DIRECTORY = dump_scott  
DUMPFILE = exp_scott_%U.dmp  
LOGFILE = exp_scott.log  
SCHEMAS = scott  
PARALLEL= 2  
EXCLUDE = TABLE:"IN ('EMP', 'DEPT')"  
expdp system/manager parfile=exp.par
```

命令行下转义符的处理

Windows平台:

```
D:\> expdp system/manager DIRECTORY=my_dir DUMPFILE=exp_tab.dmp LOGFILE=exp_tab.log SCHEMAS=scott  
INCLUDE=TABLE:"IN ('EMP', 'DEPT')"
```

在Windows平台下, 需要对象双引号进行转义, 使用转义符\

Unix平台:

在未使用parfile文件的情形下, 所有的符号都需要进行转义, 包括括号, 双引号, 单引号等

```
% expdp system/manager DIRECTORY=my_dir DUMPFILE=exp_tab.dmp LOGFILE=exp_tab.log SCHEMAS=scott  
INCLUDE=TABLE:"IN \(\'EMP\'', \'DEPT\'')"
```

exclude/include常见的错误

任意需要转义的字符如果未转义或转义错误, 都会产生ORA错误。下面给出几种常见的ORA错误。

```
ORA-39001: invalid argument value  
ORA-39071: Value for INCLUDE is badly formed.  
ORA-00936: missing expression  
ORA-39001: invalid argument value  
ORA-39071: Value for EXCLUDE is badly formed.  
ORA-00904: "DEPT": invalid identifier  
ORA-39001: invalid argument value  
ORA-39041: Filter "INCLUDE" either identifies all object types or no object types.  
ORA-39001: invalid argument value  
ORA-39041: Filter "EXCLUDE" either identifies all object types or no object types  
ORA-39001: invalid argument value  
ORA-39038: Object path "USER" is not supported for TABLE jobs.
```

高级过滤

在导出/导入的时候, 我们常常有这样的需求, 只想导出/导入表结构, 或者只想导出/导入数据。幸运的是数据泵也提供了该功能。使用 Content 参数。该参数有三个属性

- 1) ALL : 导出/导入对象定义和数据, 该参数的默认值就是ALL
- 2) DATA_ONLY : 只导出/导入数据。
- 3) METADATA_ONLY : 只导出/导入对象定义。

Warning: 有一点值得注意的时, 在执行导出的时候, 如果使用了高级过滤, 如只导出了数据, 那么导入时, 需要确保数据定义已经存在。否则数据都变成没有主子了。如果数据定义已经存在, 导入时最好指定data_only, 否则会触发ORA-39151错误, 因为对象已经存在了。

过滤已经存在的数据

我们知道, 导入的表对象在目标库中已经存在, 并且目标端没有创建数据完整性约束条件(RI)来检验数据的话, 就有可能造成数据被重复导入。数据泵提供了一个新的参数Table_exists_action, 可以一定程度上降低重复数据的产生。该参数用来控制如果要导入的表对象存在, 执行什么操作。有以下几个参数值:

- 1) SKIP : 跳过该表, 继续处理下一个对象。该参数默认就是SKIP。值得注意的是, 如果你同时指定了CONTENT参数为Data_only的话, SKIP参数无效, 默认为APPEND。
- 2) APPEND : 向现有的表中添加数据。
- 3) TRUNCATE : TRUNCATE当前表, 然后再添加记录。使用这个参数需要谨慎, 除非确认当前表中的数据确实无用。否则可能造成数据丢失。
- 4) REPLACE : 删除并重建表对象, 然后再向其中添加数据。值得注意的是, 如果同时指定了CONTENT参数为Data_only的话, REPLACE参数无效。

重定义表的Schema或表空间

我们还可能会遇到这样的需求, 把A用户的对象转移到B用户, 或者更换数据的表空间。数据泵通过 Remap_Schema和Remap_tablespace参数实现了该功能。

- 1) REMAP_SCHEMA : 重定义对象所属Schema

该参数的作用类似IMP中的Fromuser+Touser, 支持多个Schema的转换, 语法如下:

```
REMAP_SCHEMA=Source_schema:Target_schema[, Source_schema:Target_schema]
```

如 把A的对象转换到C用户, 将C转换到D用户。Remap_schema=a:b, c:d

Warning: 不能在同一个IMPDP命令中指定remap_schema=a:b, a:c.

2) REMAP_TABLESPACE : 重定义对象所在的表空间。

该参数用来重映射导入对象存储的表空间, 支持同时对多个表空间进行 转换, 相互间用逗号分割。语法如下:

```
REMAP_TABLESPACE=Source_tablespace:Target_tablespace[, Source_tablespace:Target_tablespace]
```

Warning: 如果使用Remap_tablespace参数, 则要保证导入的用户对目标表空间有读写权限。

优化导入/导出效率

对于大数据量来说, 我们不得不考虑效率问题。数据泵对效率也提出了更高的要求。甚至官方的描述就是Oracle Data Pump technology enables Very High-Speed movement of data and metadata from one database to another. 这里的Very High-Speed依赖我们的parallel参数。

所有的优化操作都会有三种结果: 变得更好、没有变化、变得更差。Parallel参数也是这样, 并不是指定一个大于1的参数, 性能就会有提升。

1) 对于导出的parallel

对于导出来说, 由于dump文件只能由一个线程进行操作(包括I/O处理), 因此如果输出的DUMP文件只有一个, 即使你指定再多的并行, 实际工作仍然是一个, 而且还会触发ORA-39095错误。因此, 建议设置该参数小于或等于生成的DUMP文件数量。那么, 如何控制生成的DUMP文件数量呢?

EXPDP 命令提供了一个FILESIZE参数, 用来指定单个DUMP文件的最大容量, 要有效的利用parallel参数, filesize参数必不可少。

举例: 某用户对象占用了4G左右的空间, 实际导出后的DUMP文件约为3G, 我们尝试在导出该用户时指定并行度为4, 设置单个文件不超过500M, 则语法如下:

```
$ expdp user/pwd directory=dump_file dumpfile=expdp_20100820_%U.dmp logfile=expdp_20100820.log filesize=500M parallel=4
```

2) 对于导入的parallel

对于导入来说, 使用parallel参数则要简单的多, 我认为导入更能体现parallel参数的优势。 参数设置为几, 则认为同时将几张表的内容导入到库中。

举例: 某dmp文件中包含了200张表, 我们尝试在导入该DMP文件时指定并行度为10, 则 语法如下:

```
$ impdp user/pwd directory=dump_file dumpfile=expdp_20100820.dmp logfile=impdp_20100820.log parallel=10
```

如何进入交互模式

在这里, 我正在执行导入, 我想进入交互模式, 查看导入的状态。进入交互模式有两种方式, 操作步骤如下:

i 使用Ctrl+C退出当前模式

ii 在命令行模式下, 执行Expdp/Impdp命令, 同时指定attach参数连接到当前正在制定的导入/导出任务。如:

```
expdp bam/bam attach=SYS_IMPORT_FULL_01
```

Warning: 如果没有指定Attach参数, 则默认进入当前正在运行的 任务。不过如果当前没有正在指定的任务, 而且也没有给Attach赋值, 那么就会报Ora-31626错误。

当命令行进入交互模式后, 会显示如下 界面:

```
Export>
```

7) 交互模式的操作

在交互模式中, 支持下面几种操作。

i 查看JOB的运行状态

```
Export> status
```

ii 回退到命令行

```
Export> continue_client
```

iii 增加并行

```
Export> parallel=4
```

Warning: 在使用导出时, 不能直接指定parallel参数, 否则可能会遇到ORA-39095错误, 因为如果要并行导出, 则必须指定多个导出文件, 这里的并行导出 是指, 多个线程同时工作, 同时从数据库中导出多个dmp文件来。

在Oracle Database Utilities中有如下解释:

Because each active worker process or I/O server process writes exclusively to one file at a time, an insufficient number of files can have adverse effects. Some of the worker processes will be idle while waiting for files, thereby degrading the overall performance of the job. More importantly, if any member of a cooperating group of parallel I/O server processes cannot obtain a file for output, then the export operation will be stopped with an ORA-39095 error. Both situations can be corrected by attaching to the job using the Data Pump Export utility, adding more files using the ADD_FILE command while in interactive mode, and in the case of a stopped job, restarting the job.

You can supply multiple file_name specifications as a comma-delimited list or in separate DUMPFILE parameter specifications. If no extension is given for the filename, then Export uses the default file extension of .dmp. The filenames can contain a substitution variable (%U), which implies that multiple files may be generated. The substitution variable is expanded in the resulting filenames into a 2-digit, fixed-width, incrementing integer starting at 01 and ending at 99. If a file specification contains two substitution variables, both are incremented at the same time. For example, exp%Uaa%U.dmp would resolve to exp01aa01.dmp, exp02aa02.dmp, and so forth.

iv 停止JOB

```
Export> stop_job
```

v 启动JOB

Export> start_job

vi 杀掉JOB

Export> kill_job

vii 退出交互模式

Export> exit_client

viii 指定文件大小

Export> filesize=1G

ix 帮助

Export> Help

JOB_NAME

指定要导出作用的名称, 默认为SYS_XXX

JOB_NAME=jobname_string

SELECT * FROM DBA_DATAPUMP_JOBS;一查看存在的job

NETWORK_LINK导出文件到本地

远程:

IP:192.168.10.55, OS:RHEL4.8 64bit, Oracle:10.2.0.5 64bit

被导出用户:monitor

本地:

ip: 172.*, WINDOWS XP, oracle 10.2.0.4.

本地tns :cheniwo

本地配置的联系到远程的tns:lobom55

1) 确定本地可访问远程

F:\oracle\product\10.2.0\db_2\network\admin\sqlnet.ora

已使用 TNSNAMES 适配器来解析别名

Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP
(HOST = 192.168.10.55) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = lobom55)))

OK (40 毫秒)

2) 在本地建立dblink到远程

F:\Documents and Settings\Administrator>sqlplus / as sysdba

SQL*Plus: Release 10.2.0.4.0 - Production on 星期四 9月 6 14:25:44 2012

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

连接到:

Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create database link dl_monitor connect to monitor identified by monitor using 'lobom55';

数据库链接已创建。

SQL> select sysdate from dual@dl_monitor;

SYSDATE

06-9月 -12

3) 数据泵导出

F:\Documents and Settings\Administrator>expdp network_link=dl_monitor dumpfile=network_monitor.dmp

/*-----*/

说明: network_link是你建立的到远程的database link

下边输入的用户名是你本地数据库的认证

dumpfile 将放在data_pump_dir目录定义的本地目录下

-----*/

Export: Release 10.2.0.4.0 - Production on 星期四, 06 9月, 2012 14:40:17

Copyright (c) 2003, 2007, Oracle. All rights reserved.

用户名: / as sysdba

连接到: Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

启动 "SYS"."SYS_EXPORT_SCHEMA_01": /***** AS SYSDBA network_link=dl_monitor

dumpfile=network_monitor.dmp

正在使用 BLOCKS 方法进行估计...

处理对象类型 SCHEMA_EXPORT/TABLE/TABLE_DATA

使用 BLOCKS 方法的总估计: 81.68 MB

处理对象类型 SCHEMA_EXPORT/USER

.....省略若干行.....

处理对象类型 SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/FUNCTIONAL_AND_BITMAP/INDEX_STATISTICS

已成功加载/卸载了主表 "SYS"."SYS_EXPORT_SCHEMA_01"

SYS.SYS_EXPORT_SCHEMA_01 的转储文件集为:

F:\ORACLE\PRODUCT\10.2.0\ADMIN\CHENIWO\DPDUMP\NETWORK_MONITOR.DMP

作业 "SYS"."SYS_EXPORT_SCHEMA_01" 已于 14:51:41 成功完成

4) 检查结果:

SQL> set head off

SQL> select * from dba_directories where directory_name='DATA_PUMP_DIR';

SYS DATA_PUMP_DIR F:\oracle\product\10.2.0\admin\cheniwo\dpdump\

F:\Documents and Settings\Administrator>cd F:\oracle\product\10.2.0\admin\cheniwo\dpdump\

F:\oracle\product\10.2.0\admin\cheniwo\dpdump>dir /w

驱动器 F 中的卷是 新加卷

卷的序列号是 88F0-A421

F:\oracle\product\10.2.0\admin\cheniwo\dpdump 的目录

[.] [..] cheniwo.sql

CHENIWO_DP.DMP dp.log EXPDAT.DMP

export.log FULL_CHENIWO.DMP import.log

NETWORK_MONITOR.DMP TABLES_CHENIWO.DMP TABLES_CHENIWO2.DMP

TS_CHENIWO.DMP USER_CHENIWO.DMP

12 个文件 128,977,860 字节

2 个目录 155,032,698,880 可用字节

参考至:<http://loveboth.iteye.com/blog/1745801>

<http://bbs.chinaunix.net/thread-3643706-1-1.html>

<http://www.cnblogs.com/iImax/p/oracle-11g-expdp-interactive.html>

<http://blog.csdn.net/liqfyiyi/article/details/7248911>

1. 创建 DB LINK

```
create database link data_migration connect to dws identified by qwer1234 using '(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP) (HOST = 172.16.91.105) (PORT = 1521)) (CONNECT_DATA =(SERVER = DEDICATED) (SERVICE_NAME = dws)))';
```

2.

源端库的tnsnames.ora :

```
oracle@dwsdata1[/u01/app/oracle/dbhome_1/network/admin]$cat tnsnames.ora
```

```
# tnsnames.ora Network Configuration File: /u01/app/oracle/dbhome_1/network/admin/tnsnames.ora
```

```
# Generated by Oracle configuration tools.
```

DWS =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP) (HOST = dwsdata1-vip) (PORT = 1521))

(ADDRESS = (PROTOCOL = TCP) (HOST = dwsdata2-vip) (PORT = 1521))

(CONNECT_DATA =

(SERVER = DEDICATED)

(SERVICE_NAME = dws)

)

)

DWS1 =

```

(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = dwsdata1) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = dws)
  )
)

DWS2 =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = dwsdata2) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = dws)
  )
)

SJKPT =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = 193.168.103.25) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = sjkpt)
  )
)

oracle@dwsdata1[/u01/app/oracle/dbhome_1/network/admin]$
目标库的tnsnames.ora :
[oracle@rac01 admin]$ cat tnsnames.ora
# tnsnames.ora Network Configuration File: /u01/app/oracle/product/11.2/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

DWS =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = dws-scan) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = dws)
  )
)

[oracle@rac01 admin]$

oracle@dwsdata1[/u01/app/oracle/dbhome_1/network/admin]$cat /etc/hosts
# @(#)47      1.2  src/bos/usr/sbin/netstart/hosts, cmdnet, bos61D, d2007_49A2 10/1/07 13:57:52
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos61D src/bos/usr/sbin/netstart/hosts 1.2
#
# Licensed Materials - Property of IBM
#
# COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#

```

```

# IBM_PROLOG_END_TAG
#
# COMPONENT_NAME: TCPIP hosts
#
# FUNCTIONS: loopback
#
# ORIGINS: 26 27
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# /etc/hosts
#
# This file contains the hostnames and their address for hosts in the
# network. This file is used to resolve a hostname into an Internet
# address.
#
# At minimum, this file must contain the name and address for each
# device defined for TCP in your /etc/net file. It may also contain
# entries for well-known (reserved) names such as timeserver
# and printserver as well as any other host name and address.
#
# The format of this file is:
# Internet Address      Hostname      # Comments
# Internet Address can be either IPv4 or IPv6 address.
# Items are separated by any number of blanks and/or tabs. A '#'
# indicates the beginning of a comment; characters up to the end of the
# line are not interpreted by routines which search this file. Blank
# lines are allowed.

# Internet Address      Hostname      # Comments
# 192.9.200.1           net0sample    # ethernet name/address
# 128.100.0.1           token0sample  # token ring name/address
# 10.2.0.2              x25sample    # x.25 name/address
# 2000:1:1:1:209:6bff:feee:2b7f      ipv6sample    # ipv6 name/address
127.0.0.1              loopback localhost    # loopback (lo0) name/address
172.16.91.1            dwsdata1
172.16.91.2            dwsdata2

10.0.0.1               dwsdata1-priv
10.0.0.2               dwsdata2-priv

172.16.91.11           dwsdata1-vip
172.16.91.12           dwsdata2-vip

172.16.91.20           dwsdata-cluster-scan
172.16.5.105           dwsbak.jkxx.local    dwsbak

172.16.91.3            dwsdata3
172.16.91.4            dwsdata4
oracle@dwsdata1[/u01/app/oracle/dbhome_1/network/admin]$

```