

【DATAGUARD】物理 dg 配置客户端无缝切换 (八.3)--客户端 TAF 配置

1.1 BLOG 文档结构图

└─ 【DATAGUARD】物理 dg 配置客户端无缝切换 (八)--客户 ...
└─ 1.1 BLOG 文档结构图
└─ 1.2 前言部分
└─ 1.2.1 导读
└─ 1.2.2 实验环境介绍
└─ 1.2.3 相关参考文章链接
└─ 1.2.4 本文简介
└─ 1.3 实验部分
└─ 1.3.1 实验目标
└─ 1.3.2 客户端 TAF 配置
└─ 1.3.2.1 首先在主库上配置一个 TAF 的 service
└─ 1.3.2.2 建立一个存储过程，用于调用 service，确 ...
└─ 1.3.2.3 创建 1 个触发器来确保服务可以运行
└─ 1.3.2.4 启动新创建的 service
└─ 1.3.2.5 在备库查询，确认触发器和存过已经应用 ...
└─ 1.3.2.6 客户端 tnsnames 配置
└─ 1.3.3 验证客户端的 TAF
└─ 1.3.4 题外话：Java 中的配置
└─ 1.3.4.1 使用 tnsnames 配置
└─ 1.3.4.2 不使用 tnsname
└─ 1.4 总结
└─ 1.5 About Me

1.2 前言部分

1.2.1 导读

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① Data Guard Broker 的配置

② Fast-Start Failover 的配置

③ Oracle DataGuard 之客户端 TAF 配置

④ 使用 DGMGRL 来管理数据库

⑤ 物理 dg 管理和维护的一些 sql

⑥ DataGuard 客户端特级配置

注意：本篇 BLOG 中代码部分需要特别关注的地方我都用黄色背景和红色字体来表示，比如下边的例子中，thread 1 的最大归档日志号为 33，thread 2 的最大归档日志号为 43

是需要特别关注的地方。

```
List of Archived Logs in backup set 11
Thrd Seq      Low SCN      Low Time          Next SCN      Next Time
-----
1      32          1621589      2015-05-29 11:09:52 1625242      2015-05-29 11:15:48
1      33          1625242      2015-05-29 11:15:48 1625293      2015-05-29 11:15:58
2      42          1613951      2015-05-29 10:41:18 1625245      2015-05-29 11:15:49
2      43          1625245      2015-05-29 11:15:49 1625253      2015-05-29 11:15:53
```

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.2.2 实验环境介绍

项目	主库	dg 库
db 类型	单实例	单实例
db version	11.2.0.3	11.2.0.3
db 存储	FS type	FS type
ORACLE_SID	oradg11g	oradgphy
db_name	oradg11g	oradg11g
主机 IP 地址：	192.168.59.130	192.168.59.130
OS 版本及 kernel 版本	RHEL6.5 64 位，2.6.32-504.16.2.el6.x86_64	RHEL6.5 64 位，2.6.32-504.16.2.el6.x86_64
OS hostname	rhel6_lhr	rhel6_lhr

1.2.3 相关参考文章链接

dg 的系列文章参考：

【DATAGUARD】 基于同一个主机建立物理备库和逻辑备库（一）：<http://blog.itpub.net/26736162/viewspace-1448197/>

【DATAGUARD】 基于同一个主机建立物理备库和逻辑备库（二）：<http://blog.itpub.net/26736162/viewspace-1448207/>

【DATAGUARD】 基于同一个主机建立物理备库和逻辑备库（三）：<http://blog.itpub.net/26736162/viewspace-1481972/>

【DATAGUARD】 基于同一个主机建立物理备库和逻辑备库（四）--添加一个物理 dg 节点：<http://blog.itpub.net/26736162/viewspace-1484878/>

【DATAGUARD】 物理 dg 的 switchover 切换（五）：<http://blog.itpub.net/26736162/viewspace-1753111/>

【DATAGUARD】 物理 dg 的 failover 切换(六)：<http://blog.itpub.net/26736162/viewspace-1753130/>

【DATAGUARD】 物理 dg 在主库丢失归档文件的情况下的恢复(七)：<http://blog.itpub.net/26736162/viewspace-1780863/>

【DATAGUARD】 物理 dg 配置客户端无缝切换（八.1）--Data Guard Broker 的配置：<http://blog.itpub.net/26736162/viewspace-1811839/>

【DATAGUARD】 物理 dg 配置客户端无缝切换（八.2）--Fast-Start Failover 的配置：<http://blog.itpub.net/26736162/viewspace-1811936/>

【DATAGUARD】 物理 dg 配置客户端无缝切换（八.3）--客户端 TAF 配置：<http://blog.itpub.net/26736162/viewspace-1811944/>

【DATAGUARD】 物理 dg 配置客户端无缝切换（八.4）--ora-16652 和 ora-16603 错误：<http://blog.itpub.net/26736162/viewspace-1811947/>

1.2.4 本文简介

本篇 blog 是基于 cuug 的公开课内容，我自己进行实践的操作，视频可以参考：<http://blog.itpub.net/26736162/viewspace-1624453/>，简介我就不多写了，把 cuug 的内容直接 copy 过来吧，觉得还是比较有用的。

这个技术如果你不知道，不能算是 ORACLE 高手

这个技术如果你不知道，就不能说你会 DataGuard

这个技术如果你不知道，.....

本次网络课程，**研究当主备库发生切换时，如何在主库启动一个 service，保证客户端的连接能够继续，而且还能够继续 select 查询操作，而不管主备库是在哪台服务器上；同时保证新的客户连接没有任何的问题。**本课程网络上的例子不多，陈老师花了将近一年的时间人肉搜索，最近才找到，急不可待的要分享给大家。

- 1、DataGuard 的配置（快速）
- 2、创建 service
- 3、创建触发器
- 4、主备库切换测试

由于内容较多，我打算分为 4 个章节来共享给大家，贴个图，不要奇怪，还有一个章节是实验过程中配到的问题解决。

第 6 章DataGuard 客户端特级配置
▷ 6.1 Data Guard Broker 的配置
▷ 6.2 Fast-Start Failover 的配置
▷ 6.3 Oracle DataGuard 之客户端 TAF 配置

本篇为第二节，Fast-Start Failover 的配置。

1.3 实验部分

1.3.1 实验目标

完成 Oracle DataGuard 之客户端 TAF 配置，并测试无缝切换。

1.3.2 客户端 TAF 配置

1.3.2.1 首先在主库上配置一个 TAF 的 service

此服务在数据库出现故障时会发送通知给客户端，允许查询语句在故障转移发生后继续运行。

```
在主库端运行：
14:51:45 SQL> begin
14:52:10 2   DBMS_SERVICE.CREATE_SERVICE(service_name      => 'dg_taf_lhr',
14:52:10 3       network_name          => 'dg_taf_lhr',
14:52:10 4       aq_ha_notifications => TRUE,
14:52:10 5       failover_method      => 'BASIC',
14:52:10 6       failover_type        => 'SELECT',
14:52:10 7       failover_retries     => 30,
14:52:10 8       failover_delay       => 5);
14:52:10 9   end;
14:52:10 10  /

PL/SQL 过程已成功完成。

已用时间: 00: 00: 00.07
14:52:10 SQL>
```

1.3.2.2 建立一个存储过程，用于调用 service，确保只在主库运行

我们创建一个存储过程来实现此目的，如果当前数据库是主库它就启动此服务，如果是备库就停止。

主库执行：

```
14:54:58 SQL> create or replace procedure dg_taf_proc_lhr is
14:59:46 2   v_role VARCHAR(30);
14:59:46 3   begin
14:59:46 4       select DATABASE_ROLE into v_role from V$DATABASE;
14:59:46 5       if v_role = 'PRIMARY' then
14:59:46 6           DBMS_SERVICE.START_SERVICE('dg_taf_lhr');
14:59:46 7       else
14:59:46 8           DBMS_SERVICE.STOP_SERVICE('dg_taf_lhr');
14:59:46 9       end if;
```

```
14:59:46 10 end;
14:59:46 11 /

过程已创建。

已用时间: 00: 00: 00.07
```

1. 3. 2. 3 创建 1 个触发器来确保服务可以运行

创建两个触发器，让数据库在启动和角色转换时运行此存储过程。用于当数据库 open 时，不需要重启数据库，如果是主库则执行存储过程。当数据库切换后，如果是主库则执行存储过程。

主库执行：

```
14:59:47 SQL> create or replace TRIGGER dg_taf_trg_startup_lhr
14:59:53 2 after startup or db_role_change on database
14:59:53 3 begin
14:59:53 4 dg_taf_proc_lhr;
14:59:53 5 end;
14:59:53 6 /

触发器已创建

已用时间: 00: 00: 00.26
14:59:54 SQL>
```

1. 3. 2. 4 启动新创建的 service

在主库上执行该存储过程(或者重启数据库，在启动数据库时会触发执行 dg_taf_proc 的触发器)，在主库做日志的切换，将变化应用到备库

主库执行：

```
14:59:54 SQL> exec dg_taf_proc_lhr ;

PL/SQL 过程已成功完成。

已用时间: 00: 00: 00.01
15:06:57 SQL> alter system switch logfile;

系统已更改。

已用时间: 00: 00: 01.02
15:20:01 SQL>
[oracle@rhel6_lhr ~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.3.0 - Production on 29-9 月 -2015 15:07:32
```

Copyright (c) 1991, 2011, Oracle. All rights reserved.

正在连接到 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=192.168.59.130) (PORT=1521)))
LISTENER 的 STATUS

```
-----
别名                LISTENER
版本                TNSLSNR for Linux: Version 11.2.0.3.0 - Production
启动日期            29-9 月 -2015 12:14:49
正常运行时间        0 天 2 小时 52 分 43 秒
跟踪级别            off
安全性              ON: Local OS Authentication
SNMP                OFF
监听程序参数文件    /u01/app/grid/11.2.0/network/admin/listener.ora
监听端点概要...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=192.168.59.130) (PORT=1521)))
服务摘要..
服务 "+ASM" 包含 1 个实例。
  实例 "+ASM", 状态 READY, 包含此服务的 1 个处理程序...
服务 "PLSExtProc" 包含 1 个实例。
  实例 "PLSExtProc", 状态 UNKNOWN, 包含此服务的 1 个处理程序...
```

服务 "dg_taf_lhr.lhr.com" 包含 1 个实例。

实例 "oradg11g", 状态 READY, 包含此服务的 1 个处理程序...

服务 "orallg" 包含 1 个实例。

实例 "orallg", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradg11g" 包含 1 个实例。

实例 "oradg11g", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradg11g.lhr.com" 包含 2 个实例。

实例 "oradg11g", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

实例 "oradg11g", 状态 READY, 包含此服务的 1 个处理程序...

服务 "oradg11gXDB.lhr.com" 包含 2 个实例。

实例 "oradg11g", 状态 READY, 包含此服务的 1 个处理程序...

实例 "oradgphy", 状态 READY, 包含此服务的 1 个处理程序...

服务 "oradg11g_DGB.lhr.com" 包含 1 个实例。

实例 "oradg11g", 状态 READY, 包含此服务的 1 个处理程序...

服务 "oradg11g_DGMGRL.lhr.com" 包含 1 个实例。

实例 "oradg11g", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradglg" 包含 1 个实例。

实例 "oradglg", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradgphy" 包含 1 个实例。

实例 "oradgphy", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradgphy.lhr.com" 包含 2 个实例。

实例 "oradgphy", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

实例 "oradgphy", 状态 READY, 包含此服务的 1 个处理程序...

服务 "oradgphy_DGB.lhr.com" 包含 1 个实例。

实例 "oradgphy", 状态 READY, 包含此服务的 1 个处理程序...

服务 "oradgphy_DGMGRL.lhr.com" 包含 1 个实例。

实例 "oradgphy", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "oradgss" 包含 1 个实例。

实例 "oradgss", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

服务 "orclasm.lhr.com" 包含 1 个实例。

实例 "orclasm", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

命令执行成功

1.3.2.5 在备库查询，确认触发器和存过已经应用到备库

```
15:21:21 SQL> select trigger_name, trigger_name
15:22:09      2      from dba_triggers where trigger_name = 'DG_TAF_TRG_STARTUP_LHR';
```

TRIGGER_NAME	TRIGGER_NAME
--------------	--------------

```
DG_TAF_TRG_STARTUP_LHR      DG_TAF_TRG_STARTUP_LHR

已用时间:   00: 00: 00.09
15:22:09 SQL>
15:24:28 SQL> select d.owner,d.OBJECT_NAME
15:24:54      2      from dba_procedures d
15:24:54      3      where d.OBJECT_NAME = 'DG_TAF_PROC_LHR';

OWNER                                OBJECT_NAME
-----
SYS                                DG_TAF_PROC_LHR

已用时间:   00: 00: 00.05
15:24:54 SQL>
```

1.3.2.6 客户端 tnsnames 配置

```
dg_taf =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = tcp) (HOST = 192.168.59.130) (PORT = 1521))
    (ADDRESS = (PROTOCOL = tcp) (HOST = 192.168.59.130) (PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = dg_taf_lhr.lhr.com)
    (FAILOVER_MODE =
      (TYPE = session)
      (METHOD = basic)
      (RETRIES = 180)
      (DELAY = 5)
    )
  )
)
```

注：我的 dg 环境由于部署在同一台机器，所以 host 一样，生产环境下必然不一样。

1.3.3 验证客户端的 TAF

先说说测试过程，首先我们在 windows 环境下添加 tnsnames，然后 cmd 中连接到 dg 环境，执行一个长久的查询(select * from (select * from sys.dba_objects);)，此时在 dgmgrl 中手动 shutdown abort 掉主库，那么连接的 cmd 中会停顿一会，等待 fast-start failover 切换完成后，则继续返回结果。

主备切换不影响用户的 select 操作，但是如果是 dml 操作，则所有事务回滚：


```
D:\Users\xiaomaimiao>sqlplus lhr/lhr@dg_taf

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 15:37:20 2015

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> set line 9999
SQL> col name format a10
SQL> col FS_FAILOVER_OBSERVER_HOST format a20
SQL> col DB_UNIQUE_NAME format a15
SQL> select dbid,name, DB_UNIQUE_NAME, RESETLOGS_CHANGE#, current_scn, protection_mode, protection_level, database_role, force_logging, open_mode, switchover_status from v$database;
```

DBID	NAME	DB_UNIQUE_NAME	RESETLOGS_CHANGE#	CURRENT_SCN	PROTECTION_MODE	PROTECTION_LEVEL	DATABASE_ROLE	FOR	OPEN_MODE	SWITCHOVER_STATUS
1403587593	ORADG11G	oradg11g	2575356	2604578	MAXIMUM AVAILABILITY	MAXIMUM AVAILABILITY	PRIMARY	YES	READ WRITE	TO STANDBY

```
SQL> SELECT d.DBID,
2      d.DB_UNIQUE_NAME,
3      d.FORCE_LOGGING,
4      d.FLASHBACK_ON,
5      DATAGUARD_BROKER,
6      d.FS_FAILOVER_STATUS,
7      d.FS_FAILOVER_CURRENT_TARGET,
8      d.FS_FAILOVER_THRESHOLD,
9      d.FS_FAILOVER_OBSERVER_PRESENT,
10     d.FS_FAILOVER_OBSERVER_HOST
11  FROM v$database d;
```

DBID	DB_UNIQUE_NAME	FOR	FLASHBACK_ON	DATAGUAR	FS_FAILOVER_STATUS	FS_FAILOVER_CURRENT_TARGET	FS_FAILOVER_THRESHOLD	FS_FAIL	FS_FAILOVER_OBSERVER
1403587593	oradg11g	YES	YES	ENABLED	SYNCHRONIZED	oradgphy	30	YES	rhel6_lhr

```
SQL>
```

可以看到连接到的是主库。

```
C:\WINDOWS\system32\cmd.exe - sqlplus lhr/lhr@dg_taf
```

OWNER	OBJECT_NAME
SYS	V_\$SESSION_BLOCKERS
PUBLIC	V\$SESSION_BLOCKERS
SYS	V_\$WAIT_CHAINS
PUBLIC	V\$WAIT_CHAINS
SYS	V_\$SESSION_EVENT
PUBLIC	V\$SESSION_EVENT
SYS	V_\$SESSION_CONNECT_INFO
PUBLIC	V\$SESSION_CONNECT_INFO
SYS	V_\$SYSTEM_WAIT_CLASS

Cmd 卡住了，等待 observer 切换完成后 cmd 界面继续查询：

15:44:42.75 2015 年 9 月 29 日 星期二

正在为数据库 "oradgphy" 启动快速启动故障转移...

立即执行故障转移, 请稍候...

故障转移成功, 新的主数据库为 "oradgphy"

15:44:49.93 2015 年 9 月 29 日 星期二

```
C:\WINDOWS\system32\cmd.exe - sqlplus lhr/lhr@dg_taf
SYS                                V_$DIAG_IPS_PKG_UNPACK_HIST
OWNER                             OBJECT_NAME
-----
PUBLIC                           V_$DIAG_IPS_PKG_UNPACK_HIST
SYS                               V_$DIAG_IPS_REMOTE_PACKAGE
PUBLIC                           V_$DIAG_IPS_REMOTE_PACKAGE
SYS                               V_$DIAG_IPS_CONFIGURATION
PUBLIC                           V_$DIAG_IPS_CONFIGURATION
SYS                               V_$DIAG_IPS_PROGRESS_LOG
PUBLIC                           V_$DIAG_IPS_PROGRESS_LOG
SYS                               V_$DIAG_INC_METER_SUMMARY
PUBLIC                           V_$DIAG_INC_METER_SUMMARY
SYS                               V_$DIAG_INC_METER_INFO
PUBLIC                           V_$DIAG_INC_METER_INFO
OWNER                             OBJECT_NAME
-----
SYS                               V_$DIAG_INC_METER_CONFIG
PUBLIC                           V_$DIAG_INC_METER_CONFIG
SYS                               V_$DIAG_INC_METER_TMPT_DEF
```

```
SQL> set line 9999
SQL> col name format a10
SQL> col FS_FAILOVER_OBSERVER_HOST format a20
SQL> col DB_UNIQUE_NAME format a15
SQL> select dbid,name, DB_UNIQUE_NAME, RESETLOGS_CHANGE#, current_scn, protection_mode, protection_level, database_role, force_logging, open_mode, switchover_status from v$database;
```

DBID	NAME	DB_UNIQUE_NAME	RESETLOGS_CHANGE#	CURRENT_SCN	PROTECTION_MODE	PROTECTION_LEVEL	DATABASE_ROLE	FOR	OPEN_MODE	SWITCHOVER_STATUS
1403587593	ORADG11G	oradgphy	2605058	2605468	MAXIMUM AVAILABILITY	RESYNCHRONIZATION	PRIMARY	YES	READ WRITE	NOT ALLOWED

```
SQL> SELECT d.DBID,
2      d.DB_UNIQUE_NAME,
3      d.FORCE_LOGGING,
4      d.FLASHBACK_ON,
5      DATAGUARD_BROKER,
6      d.FS_FAILOVER_STATUS,
7      d.FS_FAILOVER_CURRENT_TARGET,
8      d.FS_FAILOVER_THRESHOLD,
9      d.FS_FAILOVER_OBSERVER_PRESENT,
10     d.FS_FAILOVER_OBSERVER_HOST
11 FROM v$database d;
```

DBID	DB_UNIQUE_NAME	FOR	FLASHBACK_ON	DATAGUAR	FS_FAILOVER_STATUS	FS_FAILOVER_CURRENT_TARGET	FS_FAILOVER_THRESHOLD	FS_FAIL	FS_FAILOVER_OBSERVER
1403587593	oradgphy	YES	YES	ENABLED	REINSTATE REQUIRED	oradg11g	30	YES	rhel6_lhr

```
SQL>
```

切换后，在备库上存储过程启动了 TAF 的 service ，客户端再连接时，自动连接到了当前的主库（原备库）。

@至此，在 DG 环境下客户端的 TAF 配置基本完成。

1.3.4 题外话：Java 中的配置

1.3.4.1 使用 tnsnames 配置

原文链接：<http://aijuans.iteye.com/blog/1488998>

格式一： Oracle JDBC Thin using a ServiceName:

jdbc:oracle:thin:@//<host>:<port>/<service_name>

Example: jdbc:oracle:thin:@//192.168.2.1:1521/XE

注意这里的格式，@后面有//，这是与使用 SID 的主要区别。

这种格式是 Oracle 推荐的格式，因为对于集群来说，每个节点的 SID 是不一样的，但是 SERVICE_NAME 确可以包含所有节点。

格式二: Oracle JDBC Thin using an SID:

jdbc:oracle:thin:@<host>:<port>:<SID>

Example: jdbc:oracle:thin:192.168.2.1:1521:X01A

Note: Support for SID is being phased out. Oracle recommends that users switch over to using service names.

格式三： Oracle JDBC Thin using a TNSName:

jdbc:oracle:thin:@<TNSName>

Example: jdbc:oracle:thin:@GL

Note:

Support for TNSNames was added in the driver release 10.2.0.1

二.测试

2.1 准备工作：

Oracle 是 11gR2

Listener.ora

SID_LIST_LISTENER =

(SID_LIST =

(SID_DESC =

(GLOBAL_DBNAME = dave)

(ORACLE_HOME =D:\app\Administrator\product\11.2.0\dbhome_1)

(SID_NAME = NEWCCS)

)

)

Tnsnames.ora

DVD =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))

)

(CONNECT_DATA =

(SERVICE_NAME = dave)

)

)

2.3 使用 service_name：dave

将 2.2 节的 dbUrl 改成如下：

```
String dbUrl = "jdbc:oracle:thin:@//127.0.0.1:1521/dave";
```

输出结果：

MGMT_VIEW--97

ANQING--94

DVD--93

SYSMAN--95

如果在 11g 里遇到如下错误：

测试运行 Java 类，报错：

java.sql.SQLException: The Network Adapter could not establish the connection

可以尝试更换对应的 jdbc connection driver，官网的说明如下：

JDBC Thin Driver 11g Causes"Java.Sql.Sqlexception: Io Exception: The Network Adapter Could NotEstablish The Connection" While Connecting to Oracle Database 11g [ID947653.1]

Change the JDBC connection driver class inyour application server from:

oracle.jdbc.driver.OracleDriver

to

oracle.jdbc.OracleDriver

2.4 使用 TNS name ： dvd

```
String dbUrl = "jdbc:oracle:thin:@dvd";
```

报错如下：

java.sql.SQLException: Unknown host specified

该问题是因为 JVM 没有 oracle.net.tns_admin 的 system property。解决方法有 2 种：


方法一：在启动 VM 时添加如下参数：

-Doracle.net.tns_admin=D:\app\Administrator\product\11.2.0\dbhome_1\NETWORK\ADMIN

方法二：在 java 代码里添加：

```
System.setProperty("oracle.net.tns_admin","D:\\app\\Administrator\\product\\11.2.0\\dbhome_1\\NETWORK\\ADMIN");
```

添加之后，就可以正常在 JDBC 中使用 tnsname 了。

Java 代码 

1. 2.2 测试 1，使用 SID:newccs
- 2.
3. [java] view plaincopy

```
4.
5. import java.sql.*;
6.
7. public class jdbc {
8.     String dbUrl = "jdbc:oracle:thin:@127.0.0.1:1521:newccs";
9.     String theUser = "dave";
10.    String thePw = "dave";
11.    Connection c = null;
12.    Statement conn;
13.    ResultSet rs = null;
14.
15.    public jdbc() {
16.        try {
17.            Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();
18.            c = DriverManager.getConnection(dbUrl, theUser, thePw);
19.            conn = c.createStatement();
20.        } catch (Exception e) {
21.            e.printStackTrace();
22.        }
23.    }
24.
25.    public boolean executeUpdate(String sql) {
26.        try {
```

```
27.     conn.executeUpdate(sql);
28.     return true;
29. } catch (SQLException e) {
30.     e.printStackTrace();
31.     return false;
32. }
33. }
34.
35. public ResultSet executeQuery(String sql) {
36.     rs = null;
37.     try {
38.         rs = conn.executeQuery(sql);
39.     } catch (SQLException e) {
40.         e.printStackTrace();
41.     }
42.     return rs;
43. }
44.
45. public void close() {
46.     try {
47.         conn.close();
48.         c.close();
49.     } catch (Exception e) {
```



```
50.     e.printStackTrace();
51. }
52. }
53.
54. public static void main(String[] args) {
55.     ResultSet rs;
56.     jdbc conn = new jdbc();
57.     rs = conn.executeQuery("select * from dave where rownum<5");
58.     try {
59.         while (rs.next()) {
60.             System.out.println(rs.getString("username")+"--"+rs.getString("user_id"));
61.         }
62.     } catch (Exception e) {
63.         e.printStackTrace();
64.     }
65. }
66. }
67.
68. ---输出正常:
69. MGMT_VIEW--97
70. ANQING--94
71. DVD--93
72. SYSMAN--95
```

1.3.4.2 不使用 tnsname

```
jdbc:oracle:thin:@
(description=
(ADDRESS_LIST =
(address=(protocol=tcp)(host=192.168.1.44)(port=1521))
(address=(protocol=tcp)(host=192.168.1.45)(port=1521))
(address=(protocol=tcp)(host=192.168.1.46)(port=1521))
(load_balance=yes)
)
(connect_data =
(service_name=ORACMS)
(failover_mode =
(type=session)
(method=basic)
(retries=5)
(delay=15)
)
)
)
```

-- 加上注释后的 TNS 连接串

```
jdbc:oracle:thin:@
(description=
(ADDRESS_LIST =
(address=(protocol=tcp)(host=192.168.1.44)(port=1521))
(address=(protocol=tcp)(host=192.168.1.45)(port=1521))
(address=(protocol=tcp)(host=192.168.1.46)(port=1521))

(load_balance=yes)//表示是否负载均衡

)
(connect_data =

//(server = dedicated)//该参数表示专用服务器模式

(service_name=ORACMS)//要操作数据库的服务名

(failover_mode =//连接失败后处理的方式
```

(type=session)//TYPE =SESSION 表示当一个连接好的会话的实例发生故障，系统会自动将会话切换到其他可用的实例，前台应用无须再度发起连接，但会话正在执行的 SQL 需要重新执行。

```
(method=basic)//表示初始连接就连接一个接点

(retries=5)//连接失败后重试连接的次数

(delay=15)//连接失败后重试的延迟时间(以秒为单位)

)

)

)
```

1.4 总结

本篇为第三节，请查看第四节： dgmgrl 维护中的常见错误。

1.5 About Me

.....

本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用

ITPUB BLOG：<http://blog.itpub.net/26736162>

本文地址：<http://blog.itpub.net/26736162/viewspace-1811944/>

本文pdf版：<http://yunpan.cn/cdEQedhCs2kFz>（提取码：ed9b）

QQ：642808185 若加 QQ 请注明你所正在读的文章标题

创作时间地点：2015-09-28 09:00~ 2015-09-29 23:00 于外汇交易中心完成

<版权所有，文章允许转载，但须以链接方式注明源地址，否则追究法律责任!>

.....

