Home / Database / Oracle Database Online Documentation 11g Release 2 (11.2) / Database Administration

Database Administrator's Guide

3 Starting Up and Shutting Down

[https://docs.oracle.com/cd/E11882\_01/server.112/e25494/start.htm#ADMIN004](https://docs.oracle.com/cd/E11882_01/server.112/e25494/start.htm" \l "ADMIN004)

深入解析Oracle/01.DBStartAndShutdown.pdf

数据库的启动和关闭非常简单，startup和shutdown两个命令即可完成，而命令之后Oracle需要执行一系列复杂的操作，下面来看一下数据库的启动和关闭过程

》Starting Up a Database

connecting as SYSOPER or SYSDBA

you can start up a database instance with SQL\*Plus, Recovery Manager, or Enterprise Manager， If your database is being managed by Oracle Restart, the recommended way to start the database is with SRVCTL

》modes

In either case, you can start an instance in various modes:

NOMOUNT—Start the instance without mounting a database. This does not allow access to the database and usually would be done only for database creation or the re-creation of control files.

MOUNT—Start the instance and mount the database, but leave it closed. This state allows for certain DBA activities, but does not allow general access to the database.

OPEN—Start the instance, and mount and open the database. This can be done in unrestricted mode, allowing access to all users, or in restricted mode, allowing access for database administrators only.

FORCE—Force the instance to start after a startup or shutdown problem.

OPEN RECOVER—Start the instance and have complete media recovery begin immediately.

open的几种子模式

restrict：只允许拥有系统权限ESTRICTED SESSION的用户连接，一般就是dba角色；只允许本地登录无法远程登陆。主要用于以下任务

Perform an export or import of data

Perform a data load (with SQL\*Loader)

Temporarily prevent typical users from using data

Perform certain migration or upgrade operations

Restricting Access to an Open Database--open阶段限制模式的启用和关闭

ALTER SYSTEM disable RESTRICTED SESSION;

ALTER SYSTEM ENABLE RESTRICTED SESSION;

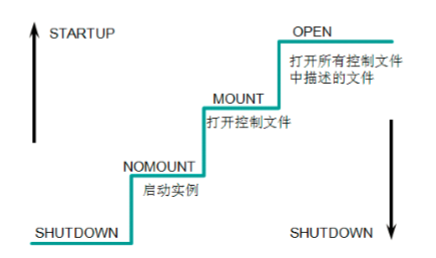
force：

If an instance is running, the force mode shuts it down with mode ABORT before restarting it. In this case, the alert log shows the message "Shutting down instance (abort)" followed by "Starting ORACLE instance (normal)."

readonly：

ALTER DATABASE OPEN READ ONLY;

》启动过程



=====================================

》NOMOUNT阶段

Nomount 的过程**也就是启动数据库实例的过程。** 这个过程在后台是启动 Oracle 可执行程序的过程， Windows上是执行 oracle.exe 文件进行初始化，在 Unix/Linux 上是执行 oracle 可执行文件进行初始化。

1 Oracle 的可执行文件

D:\ dir oracle.exe

2007-10-03  17:42        89,702,400 oracle.exe

$ ls -al $ORACLE\_HOME/bin/oracle

-rwsr-s--x 1 oracle asmadmin 232399123 Dec 31  2014 /u02/app/oracle/product/11.2.0/dbhome\_1/bin/oracle

在 Unix/Linux 上可以通过 file 命令查看 oracle 执行文件来判断 Oracle 是 64 位或是 32 位的，以下是 Linux 平台的一个示范输出，输出显示 Oracle 为 64 位：

$ file $ORACLE\_HOME/bin/oracle

/u02/app/oracle/product/11.2.0/dbhome\_1/bin/oracle: setuid setgid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, not stripped

对于可执行程序oracle可以使用linux 命令strings转储查看

strings $ORACLE\_HOME/bin/oracle >/tmp/oracle

2 实例及进程的创建

在 Nomount 初始化的过程中，需要读取参数文件。Oracle 根据参数文件的内容，创建了 instance,分配了相应的内存区域，启动了相应的后台进程。SGA 的分配信息从以上输出中可以看到。 所有在参数文件中定义的非缺省参数都会记录在告警日志文件中，应用参数创建实例之后，后台进程依次启动。

1 Specifying Initialization Parameters at Startup

从 Oracle 9i 开始，spfile 被引入 Oracle 数据库

In the platform-specific default location（?/dbs/spfile<ORACLE\_SID>.ora）, Oracle Database locates your initialization parameter file by examining file names in the following order:

    1 spfileORACLE\_SID.ora

    2 spfile.ora

    3 initORACLE\_SID.ora

Nondefault Server Parameter Files

pfile：包含SPFILE参数

SRVCTL：srvctl modify database -d db\_unique\_name -p spfile\_path

注意：**你可以在PFILE链接到SPFILE文件,同时在PFILE中定义其他参数， 如果参数重复设置，后读取的参数将取代先前的设置**

**实例启动最小参数需求??  》参考手动建库》实际只需要DB\_NAME**

2 依据参数启动实例

分享：查询隐含参数脚本

3 启动后台进程

V$PROCESS 被认为是从操作系统到数据库的入口

分享：依据spid查询sql

注意：参数文件丢失的报错不会在写入告警日志，因为此时，数据库没有启动。

下面看一下告警日志中有关nonount阶段的记录

Starting ORACLE instance (normal)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Large Pages Information \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

......

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Starting up:

Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production

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

ORACLE\_HOME = /u02/app/oracle/product/11.2.0/dbhome\_1

System name:    Linux

Node name:    db

Release:    2.6.32-358.el6.x86\_64

Version:    #1 SMP Tue Jan 29 11:47:41 EST 2013

Machine:    x86\_64

VM name:    VMWare Version: 6

**Using parameter settings in server-side pfile /u02/app/oracle/product/11.2.0/dbhome\_1/dbs/initorcl.ora**

**System parameters with non-default values:**

  processes                = 150

  spfile                   = "+DATA/orcl/spfileorcl.ora"

  sga\_target               = 296M

  control\_files            = "+DATA/orcl/controlfile/current.260.867775171"

  db\_block\_size            = 8192

  compatible               = "11.2.0.0.0"

  db\_create\_file\_dest      = "+DATA"

  undo\_tablespace          = "UNDOTBS1"

  remote\_login\_passwordfile= "EXCLUSIVE"

  db\_domain                = ""

  dispatchers              = "(PROTOCOL=TCP) (SERVICE=orclXDB)"

  job\_queue\_processes      = 1000

  audit\_file\_dest          = "/u02/app/oracle/admin/orcl/adump"

  audit\_trail              = "DB"

  db\_name                  = "orcl"

  open\_cursors             = 300

  pga\_aggregate\_target     = 98M

  diagnostic\_dest          = "/u02/app/oracle"

Wed Jul 06 13:27:11 2016

**PMON started with pid=2, OS id=18713**

Wed Jul 06 13:27:11 2016

PSP0 started with pid=3, OS id=18715

Wed Jul 06 13:27:12 2016

VKTM started with pid=4, OS id=18717 at elevated priority

VKTM running at (1)millisec precision with DBRM quantum (100)ms

Wed Jul 06 13:27:12 2016

GEN0 started with pid=5, OS id=18721

Wed Jul 06 13:27:12 2016

DIAG started with pid=6, OS id=18723

Wed Jul 06 13:27:12 2016

DBRM started with pid=7, OS id=18725

Wed Jul 06 13:27:12 2016

DIA0 started with pid=8, OS id=18727

Wed Jul 06 13:27:12 2016

MMAN started with pid=9, OS id=18729

Wed Jul 06 13:27:12 2016

DBW0 started with pid=10, OS id=18731

Wed Jul 06 13:27:12 2016

LGWR started with pid=11, OS id=18733

Wed Jul 06 13:27:12 2016

CKPT started with pid=12, OS id=18735

Wed Jul 06 13:27:12 2016

SMON started with pid=13, OS id=18737

Wed Jul 06 13:27:12 2016

RECO started with pid=14, OS id=18739

Wed Jul 06 13:27:12 2016

RBAL started with pid=15, OS id=18741

Wed Jul 06 13:27:12 2016

ASMB started with pid=16, OS id=18743

Wed Jul 06 13:27:12 2016

MMON started with pid=17, OS id=18745

starting up 1 dispatcher(s) for network address '(ADDRESS=(PARTIAL=YES)(PROTOCOL=TCP))'...

NOTE: initiating MARK startup

Wed Jul 06 13:27:12 2016

MMNL started with pid=18, OS id=18749

starting up 1 shared server(s) ...

Starting background process MARK

Wed Jul 06 13:27:12 2016

MARK started with pid=20, OS id=18753

NOTE: MARK has subscribed

ORACLE\_BASE from environment = /u02/app/oracle

Wed Jul 06 13:27:12 2016

ALTER SYSTEM SET local\_listener='(DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.6.31)(PORT=1521))))' SCOPE=MEMORY SID='orcl';

自 Oracle Database 11g 开始，这部分信息有了进一步的增强，输出中不仅包含了pid还有 OS ID，而且每个后台进程的启动都有单独的时间标记 （时间标记可以帮助判断每个后台进程启动时所消耗的时间，从而辅助进行问题诊断），在Oracle 9i早期版本中，后台进程启动的日志信息里并不包含OS ID。

########################

提示：从 Oracle 不同版本中的变化来体会 Oracle 的技术进步、甚至借鉴这些变化是学习

Oracle 的方法之一。任何细微的变化都值得注意，**认真、细致、严谨是对 DBA 的基本素质要求**。

################################

》NOMOUNT探索

ORACLE\_SID 的含义

SID 是 System IDentifier 的缩写：是数据库的一个术语

ORACLE\_SID 就是 Oracle System IDentifier 的缩写，是一个环境变量

INSTANCE\_NAME 是数据库初始化参数

这3个是相同的意思：用环境变量ORACLE\_SID来标识SID，INSTANCE\_NAME参数默认取值为ORACLE\_SID变量值

当 Oracle 实例启动时，在操作系 统上 fork 的进程就依据 ORACLE\_SID 这个环境变量来创建，

实例的启动需要一个参数文件，参数文件的名称就是由 ORACLE\_SID 决定的

DB\_NAME 用来定义数据库名称，必须是一个不超过 8 个字符的文本串，在数 据库创建过程中，db\_name 被记录在数据文件，日志文件和控制文件中。如果数 据库实例启动过程中参数文件中的db\_name和控制文件中的数据库名称不一致， 则数据库不能启动。

同一服务器上可存储在多个实例，不同的实例可挂载同一个数据库，但同时只能有一个实例关在数据库！

export ORACLE\_SID=ttt

SQL>conn  / as sysdba

Connected to an idle instance.

SQL> startup nomount

ORA-01078: failure in processing system parameters

LRM-00109: could not open parameter file '/u02/app/oracle/product/11.2.0/dbhome\_1/dbs/initttt.ora'

SQL> !echo "db\_name=test" > /u02/app/oracle/product/11.2.0/dbhome\_1/dbs/initttt.ora

此时可以启动新实例

SQL> !echo "control\_files=/u02/app/oracle/control01.ctl" >>/u02/app/oracle/product/11.2.0/dbhome\_1/dbs/initttt.ora

此时可以启动到挂载数据库并打开使用

RMAN 的缺省实例：指定DB\_NAME=dummy

[oracle@db dbs]$ export ORACLE\_SID=ooo

[oracle@db dbs]$ rman

Recovery Manager: Release 11.2.0.3.0 - Production on Thu Mar 29 11:24:30 2018

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

RMAN> connect target /

connected to target database (not started)

RMAN> startup nomount

startup failed: ORA-01078: failure in processing system parameters

LRM-00109: could not open parameter file '/u02/app/oracle/product/11.2.0/dbhome\_1/dbs/initooo.ora'

starting Oracle instance without parameter file for retrieval of spfile

Oracle instance started

一些特殊情况时可能会用到，如恢复参数文件时

RMAN> restore spfile to '/tmp/spfile.ora' from '/tmp/spfile01sv46kn\_1\_1';

Nomount 状态就出现问题，只和参数文件有关，其他的通常是系统问题，OS 类错误一般说明是系统资源不足

》数据库自启动，依据各平台文章

=========================================

》MOUNT阶段

实例启动到 nomount 状态以后，Oracle 就可以从参数文件中获得控制文件的位置信息，然 后找到控制文件，并且根据控制文件中记录的数据文件位置进行数据文件的存在性判断。

\*.control\_files 定义了控制文件的位置

因为 Oracle 的三个(缺省的)控制文件内容完全相同，如果只是损失了其中 1~2 个，可以复 制完好的控制文件，更改为相应的名称，就可以启动数据库

Mount 数据库 之后，后台进程就可以根据控制文件中记录的数据文件信息来验证数据文件是否存在

控制文件的 HeartBeat

Thu Mar 29 11:47:08 2018

alter database mount

Thu Mar 29 11:47:12 2018

**Successful mount of redo thread 1, with mount id 2270231868**

Database mounted in Exclusive Mode

Lost write protection disabled

Completed: alter database mount

有了mount id说明数据库被特定实例所挂载

在这一步骤中，数据库需要计算 Mount id 并将其记录在控制文件中，然后开始启动心跳 （heartbeat），每 3 秒更新一次控制文件

》Mount 相同 db\_name 的数据库

同一主机上，是否可以启用相同db\_name的数据库呢 》 当然可以 》 db\_unique\_name

1 关闭数据库

a：shutdown immediate

2 复制所有在线日志文件和数据文件

cp -ipr a b

3 重建控制文件

a: alter database backup controlfile to trace;

b：create controlfile

4 使用新实例ttt挂载数据库（修改db\_unique\_name）

lk<db\_unique\_name>

lk文件及作用

命名规则：lk<db\_unique\_name>

用于指示操作系 统对**数据库（mount）**的锁定状态。当数据库启动时获得锁定，数据库关闭时释放。

$ cat lkTEST

DO NOT DELETE THIS FILE!

当mount相同db\_name的数据库时，如出现以下报错，说明用的是同一个lk文件

ALTER DATABASE   MOUNT

sculkget: failed to lock /u02/app/oracle/product/11.2.0/dbhome\_1/dbs/lkTEST exclusive

sculkget: lock held by PID: 4492

ORA-09968: unable to lock file

Linux-x86\_64 Error: 11: Resource temporarily unavailable

重启数据库生成不同的lk文件即可

================================

》OPEN 阶段

在数据库的 Open 阶段，Oracle 根据控制文件中记录的信息找到需要的文件，然 后进行检查点及完整性检查。

仅在 open 阶段，Oracle 才尝试打开并锁定数据文件，如果丢失或出现问题，则会给出错 误提示

ORA-01157: cannot identify/lock data file 3 - see DBWR trace file

一致性校验

在数据库 Open 的过程中，Oracle 将会读取数据文件头块和控制文件信息进行校验

1 检查 数据文件头中的 检查点计数（Checkpoint cnt） 和控制文件中的检查点计 数（Checkpoint cnt）是否一致

用来确认数据库文件版本是否一致

overview of control file   》 大神！

2 检查数据文件头的开始 SCN 和控制文件中记录的该文件的结束 SCN 是否一致

数据库打开时，会将结束SCN设置为无穷大

正常关闭时，会将结束scn设置为开始scn，非正常关闭时，结束scn依然是无穷大，两者不一致时就需要恢复

========================================

》Shutting Down a Database

connecting as SYSOPER or SYSDBA

Shutting Down with the Normal Mode

Shutting Down with the Immediate Mode

Shutting Down with the Transactional Mode

Shutting Down with the Abort Mode

Shutdown Timeout

If all events blocking the shutdown do not occur within one hour, the shutdown operation aborts with the following message: ORA-01013: user requested cancel of current operation

手动中断shutdown也会出现此信息，建议不要手动中断关闭过程，这种状态是未知状态，需要再次shutdown，如果继续报错，只能用shutdown abort



》其他方式

Quiescing a Database：

只允许DBA的动作，包括事务执行，查询等

允许管理员完成一些非受限操作，如更改表结构而无须等待锁

相比于重启数据库来说，Quiescing的限制更小一些

ALTER SYSTEM QUIESCE RESTRICTED;

ALTER SYSTEM UNQUIESCE;

Suspending and Resuming a Database

The ALTER SYSTEM SUSPEND statement halts all input and output (I/O) to data files (file header and file data) and control files.

ALTER SYSTEM SUSPEND;

ALTER SYSTEM RESUME;

SQL> select STATUS,DATABASE\_STATUS,ACTIVE\_STATE from v$instance;

STATUS         DATABASE\_STATUS   ACTIVE\_ST

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

OPEN         SUSPENDED           QUIESCED

SQL>

=======================

》数据一致性校验

在数据库启动过程中的检验包含以下两个步骤：

• 第一次检查数据文件头中的 Checkpoint cnt 是否与对应控制文件中的 Checkpoint cnt 一致，如果相等,进行第二次检查;

 • 第二次检查数据文件头的开始SCN和对应控制文件中的结束SCN是否一致如果结束 SCN 等于开始 SCN,则不需要对那个文件进行恢复。

 对每个数据文件都完成检查后,打开数据库.同时将每个数据文件的结束 SCN 设置为无穷 大。

》shudown immediate

控制文件头

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA FILE RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA FILE #5:

  name #9: +DATA/orcl/datafile/example.265.867775185

creation size=12800 block size=8192 status=0xe head=9 tail=9 dup=1

tablespace 6, index=6 krfil=5 prev\_file=0

unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00

Checkpoint cnt:70 scn: 0x0000.00182290 04/12/2018 11:14:00

Stop scn: 0x0000.00182290 04/12/2018 11:14:00

Creation Checkpointed at scn:  0x0000.000fb4e9 12/31/2014 16:41:34

thread:1 rba:(0x1.d397.10)

数据文件头

DATA FILE #5:

  name #9: +DATA/orcl/datafile/example.265.867775185

creation size=12800 block size=8192 status=0xe head=9 tail=9 dup=1

tablespace 6, index=6 krfil=5 prev\_file=0

unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00

Checkpoint cnt:70 scn: 0x0000.00182290 04/12/2018 11:14:00

Stop scn: 0x0000.00182290 04/12/2018 11:14:00

Creation Checkpointed at scn:  0x0000.000fb4e9 12/31/2014 16:41:34

thread:1 rba:(0x1.d397.10)

1 检查点计数

70：70

scn: 0x0000.00182290：Stop scn: 0x0000.00182290

》shutdown abort

控制文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA FILE RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA FILE #5:

  name #9: +DATA/orcl/datafile/example.265.867775185

creation size=12800 block size=8192 status=0xe head=9 tail=9 dup=1

tablespace 6, index=6 krfil=5 prev\_file=0

unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00

Checkpoint cnt:76 scn: 0x0000.00187651 04/12/2018 11:40:02

Stop scn: 0xffff.ffffffff 04/12/2018 11:39:41

Creation Checkpointed at scn:  0x0000.000fb4e9 12/31/2014 16:41:34

thread:1 rba:(0x1.d397.10)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATABASE ENTRY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Database checkpoint: Thread=1 scn: 0x0000.00187651

Controlfile Checkpointed at scn:  0x0000.00187710 04/12/2018 11:40:03

thread:0 rba:(0x0.0.0)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CHECKPOINT PROGRESS RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THREAD #1 - status:0x2 flags:0x0 dirty:38

low cache rba:(0x26.480.0) on disk rba:(0x26.512.0)

on disk scn: 0x0000.0018788b 04/12/2018 11:45:08

resetlogs scn: 0x0000.000f30dc 12/31/2014 16:39:34

heartbeat: 973234775 mount id: 1501033676

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

REDO THREAD RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Checkpointed at scn:  0x0000.00187651 04/12/2018 11:40:02

thread:1 rba:(0x26.402.10)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LOG FILE RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LOG FILE #1:

  name #3: +DATA/orcl/onlinelog/group\_1.261.867775175

Thread 1 redo log links: forward: 2 backward: 0

siz: 0x19000 seq: 0x00000025 hws: 0x8 bsz: 512 nab: 0x4d5 flg: 0x1 dup: 1

Archive links: fwrd: 0 back: 0 Prev scn: 0x0000.00181fa4

Low scn: 0x0000.00181fa7 04/12/2018 11:00:44

Next scn: 0x0000.001872ca 04/12/2018 11:34:46

LOG FILE #2:

  name #2: +DATA/orcl/onlinelog/group\_2.262.867775175

Thread 1 redo log links: forward: 3 backward: 1

siz: 0x19000 seq: 0x00000026 hws: 0x5 bsz: 512 nab: 0xffffffff flg: 0x8 dup: 1

Archive links: fwrd: 0 back: 0 Prev scn: 0x0000.00181fa7

Low scn: 0x0000.001872ca 04/12/2018 11:34:46

Next scn: 0xffff.ffffffff 01/01/1988 00:00:00

LOG FILE #3:

  name #1: +DATA/orcl/onlinelog/group\_3.263.867775175

Thread 1 redo log links: forward: 0 backward: 2

siz: 0x19000 seq: 0x00000024 hws: 0x5 bsz: 512 nab: 0x6 flg: 0x1 dup: 1

Archive links: fwrd: 0 back: 0 Prev scn: 0x0000.00181e7b

Low scn: 0x0000.00181fa4 04/12/2018 11:00:43

Next scn: 0x0000.00181fa7 04/12/2018 11:00:44

数据文件头

DATA FILE #5:

  name #9: +DATA/orcl/datafile/example.265.867775185

creation size=12800 block size=8192 status=0xe head=9 tail=9 dup=1

tablespace 6, index=6 krfil=5 prev\_file=0

unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00

Checkpoint cnt:76 scn: 0x0000.00187651 04/12/2018 11:40:02

Stop scn: 0xffff.ffffffff 04/12/2018 11:39:41

Creation Checkpointed at scn:  0x0000.000fb4e9 12/31/2014 16:41:34

thread:1 rba:(0x1.d397.10)

76：76

scn: 0x0000.00187651：Stop scn: 0xffff.ffffffff

需要恢复，如何进行的呢？

文档原创，转载请注明出处-------------------