

9 维护控制文件

ORACLE®

Copyright © Oracle Corporation, 2001. All rights reserved.

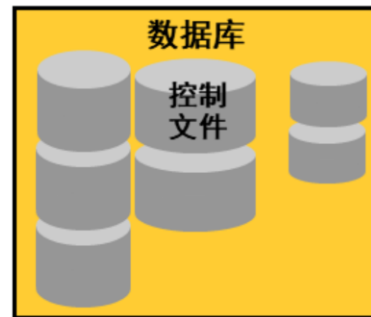
目标

完成这一课的学习后，您应该能达到下列目标：

- 解释控制文件的用途
- 列出控制文件的内容
- 对控制文件进行多元备份和管理
- 使用 “Oracle 管理文件” (Oracle Managed Files, OMF) 管理控制文件
- 获取控制文件信息

控制文件

- 小型二进制文件
- 定义物理数据库的当前状态
- 维护数据库的完整性
- 要求：
 - 在启动数据库时处于 MOUNT 状态
 - 能够操作数据库
- 只链接至一个数据库
- 丢失数据后可能需要恢复
- 最初由 **CREATE DATABASE** 确定大小



ORACLE

9-3

控制文件

控制文件是一个小型二进制文件，是成功启动和操作数据库所必需的。每个控制文件只与一个 Oracle 数据库相关联。在打开一个数据库之前，系统将读取控制文件以确定该数据库是否处于有效状态以供使用。

因为 Oracle 服务器在数据库使用的过程中会不断更新控制文件，所以控制文件必须在数据库打开时随时都可供写入。只有 Oracle 服务器才能修改控制文件中的信息；DBA 或最终用户不能编辑控制文件。

如果由于某些原因控制文件无法访问，则数据库将无法正确运行。如果数据库控制文件的所有副本都丢失，则必须先恢复数据库，然后才能将其打开。

控制文件（续）

确定控制文件的大小：

在创建数据库过程中指定的关键字会影响控制文件的大小。当参数的值较大时，这一点尤其明显。CREATE DATABASE 或 CREATE CONTROLFILE 命令中的以下关键字会影响控制文件的大小：

- MAXLOGFILES
- MAXLOGMEMBERS
- MAXLOGHISTORY
- MAXDATAFILES
- MAXINSTANCES

控制文件的内容

控制文件中包含以下条目：

- 数据库名称和标识符
- 创建数据库的时间戳
- 表空间的名称
- 数据文件和重做日志文件的名称和位置
- 当前重做日志的序列号
- 检查点信息
- 还原段的开始和结尾
- 重做日志归档信息
- 备份信息

ORACLE®

9-5

控制文件的内容

控制文件中的信息包括下列内容：

- 数据库名称取自初始化参数 DB_NAME 所指定的名称或 CREATE DATABASE 语句中所用的名称。
- 当创建数据库时会记录数据库标识符。
- 创建数据库时还会记录创建数据库的时间戳。
- 当在数据库中添加、重命名或删除数据文件或重做日志时，会更新相关数据文件和联机重做日志文件的名称和位置。
- 当添加或删除表空间时会更新表空间信息。
- 在日志切换过程中会记录重做日志历史信息。
- 归档日志的位置和状态会在归档时记录。
- 备份的位置和状态由“恢复管理器”(Recovery Manager) 实用程序记录。
- 在进行日志切换时记录当前日志序列号。
- 在建立检查点时记录检查点信息。

控制文件的内容（续）

控制文件由以下两种类型的部分组成：

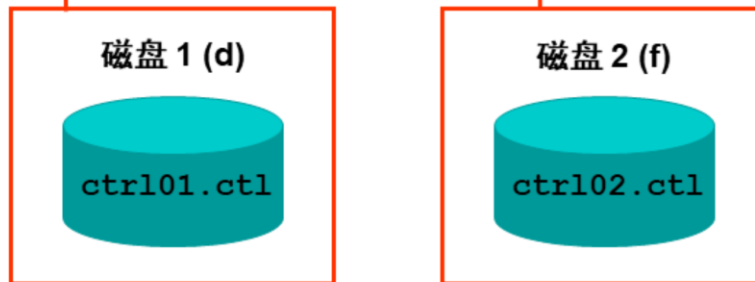
- 可重用
- 不可重用

可重用部分存储“恢复管理器” (Recovery Manager) 的信息，如备份数据文件名和备份重做日志文件名。只有“恢复管理器” (Recovery Manager) 才能以循环方式重新使用这些部分。

对控制文件进行多元备份

```
CONTROL_FILES=
```

```
'D:\u01\ctrl01.ctl', 'f:\u01\ctrl02.ctl'
```



ORACLE

9-7

对控制文件进行多元备份

要避免控制文件的单点故障所带来的损失，我们强烈建议您对控制文件进行多元备份，即在不同的物理磁盘上存储一个副本。如果某个控制文件丢失，可以使用控制文件的多元备份副本重新启动例程，而不必恢复数据库。

通过以下方式，最多可以对控制文件进行八次多元备份：

- 创建数据库时，通过在初始化参数文件中指定控制文件的名称和完整路径来创建多个控制文件：

```
CONTROL_FILES='d:\u01\ctrl01.ctl', 'f:\u01\ctrl02.ctl'
```

- 创建数据库后添加控制文件

备份控制文件：

由于控制文件记录数据库的物理结构，所以在更改数据库的物理结构后应当立即备份控制文件。

使用 SPFILE 时对控制文件 进行多元备份

1. 改变 SPFILE:

```
ALTER SYSTEM SET control_files =  
    'D:\u01\ctrl01.ctl', 'f:\u01\ctrl02.ctl' SCOPE=SPFILE;
```

- **Select value from v\$parameter2**
- **Where name='control_files';**

2. 关闭数据库:

```
shutdown immediate
```

3. 创建控制文件副本:

```
copy D:\u01\ctrl01.ctl f:\u01\ctrl02.ctl
```

4. 启动数据库:

```
startup
```

ORACLE

使用 SPFILE 时对控制文件进行多元备份

1. **改变 SPFILE:** 使用 ALTER SYSTEM SET 命令改变 SPFILE, 使其包括要用的所有控制文件的列表: 主控制文件和多元备份副本。
2. **关闭数据库:** 关闭数据库, 以便在操作系统上创建控制文件副本。
3. **创建控制文件副本:** 使用操作系统复制命令, 根据需要创建控制文件副本并验证副本已存在于相应目录中。
4. **启动数据库:** 启动数据库时, 系统将读取 SPFILE, 同时 Oracle 服务器将维护 CONTROL_FILES 参数中列出的所有控制文件。

使用 PFILE 时对控制文件 进行多元备份

1. 关闭数据库：

```
shutdown immediate
```

2. 创建控制文件副本：

```
copy D:\u01\ctrl01.ctl f:\u01\ctrl02.ctl
```

3. 向 PFILE 添加控制文件名：

```
CONTROL_FILES = 'D:\u01\ctrl01.ctl', 'f:\u01\ctrl02.ctl'
```

4. 启动数据库：

```
startup
```

ORACLE

9-9

使用 PFILE 时对控制文件进行多元备份

1. **关闭数据库：**关闭数据库，以便在操作系统上创建控制文件副本。
2. **创建控制文件副本：**使用操作系统复制命令，根据需要创建控制文件副本并验证副本已存在于相应目录中。
3. **向 PFILE 添加控制文件名：**改变 PFILE，使其包括所有控制文件的列表。
4. **启动数据库：**启动数据库时，系统将读取 PFILE，同时 Oracle 服务器将维护 CONTROL_FILES 参数中列出的所有控制文件。

控制文件的语句级备份

alter database backup controlfile to trace;

备份在

%diagnostic_dest%\diag\rdbms\%dbname%\%dbname%\trace下某个DUMP文件。

可查看**alert_<sid>.log**文件，查看存储文件名

ORACLE

9-10

控制文件的语句级备份

为防止控制文件全部被破坏，可进行语句级备份，此备份可生成创建控制文件的语句。执行

alter database backup controlfile to trace

后，可在**%diagnostic_dest%\diag\rdbms\%dbname%\%dbname%\trace**下找到相应的DUMP文件，里面有重建控制文件的步骤。

重建控制文件

- 启动实例到**nomount**状态
- 执行**create controlfile**重建控制文件
- 修复数据库 (**recover**)
- 将实例启动到**open**状态
- 添加临时表空间所包含的文件

ORACLE

9-11

控制文件的重建

如果控制文件全部被破坏，或需要对控制文件里存储的信息做一些修改（如MAXDATAFILES等），可进行重建。

- 启动实例在nomount状态

```
startup nomount
```

- 执行create controlfile重建控制文件

```
CREATE CONTROLFILE REUSE DATABASE "TEST" NORESETLOGS NOARCHIVELOG
```

```
MAXLOGFILES 20
```

```
MAXLOGMEMBERS 5
```

```
MAXDATAFILES 100
```

```
MAXINSTANCES 1
```

```
MAXLOGHISTORY 292
```

```
LOGFILE
```

```
GROUP 1 'D:\APP\111973004\ORADATA\TEST\REDO01.LOG' SIZE 50M BLOCKSIZE 512,
```

```
GROUP 2 'D:\APP\111973004\ORADATA\TEST\REDO02.LOG' SIZE 50M BLOCKSIZE 512
```

-- STANDBY LOGFILE

DATAFILE

'D:\APP\111973004\ORADATA\TEST\SYSTEM01.DBF',

'D:\APP\111973004\ORADATA\TEST\SYSAUX01.DBF',

'D:\APP\111973004\ORADATA\TEST\UNDO01.DBF'

CHARACTER SET ZHS16GBK

;

- 修复数据库

RECOVER DATABASE;

- 将实例启动到open状态

ALTER DATABASE OPEN;

- 添加临时表空间所包含的文件。

ALTER TABLESPACE TEMP ADD TEMPFILE

'D:\APP\111973004\ORADATA\TEST\TEMP01.DBF'

SIZE 52428800 REUSE AUTOEXTEND ON NEXT 8192 MAXSIZE 32767M;

使用 OMF 管理控制文件

- 如果未指定 `CONTROL_FILES` 参数，则创建 OMF
- 位置由 `DB_CREATE_ONLINE_LOG_DEST_n` 指定
- 生成的名称是唯一的并在 `alertSID.log` 中显示
- 在手工创建数据库中使用

使用 OMF 管理控制文件

如果在初始化参数文件中未指定 `CONTROL_FILES` 参数，则在创建数据库时将自动把控制文件创建为 OMF。

如果使用 `init.ora` 文件，必须将 `CONTROL_FILES` 参数设置为 OMF 生成的名称，这些名称可在 `V$CONTROLFILE` 或 `alert_SID.log` 中找到。如果使用 `SPFILE`，创建数据库时将自动设置和保存 `CONTROL_FILES` 参数。

控制文件的位置由 `DB_CREATE_ONLINE_LOG_DEST_n` 参数确定。如果未设置这个参数，控制文件将位于 `DB_CREATE_FILE_DEST` 参数定义的位置。如果未设置这两个参数，控制文件将不属于 OMF。如果控制文件不属于 OMF，将需要在初始化参数文件中设置 `CONTROL_FILES` 参数，否则将会收到错误消息。

创建文件时，将生成唯一的控制文件名称 (`ora_cmr7t30p.ctl`)，并在 `alert_SID.log` 中显示。

获取控制文件信息

有关控制文件状态和位置的信息，可以通过查询以下视图来检索。

- **V\$CONTROLFILE**: 列出与该例程相关联的所有控制文件的名称和状态
- **V\$PARAMETER**: 列出所有参数的状态和位置
- **V\$CONTROLFILE_RECORD_SECTION**: 提供有关控制文件记录部分的信息
- **SHOW PARAMETER CONTROL_FILES**: 列出控制文件的名称、状态和位置

ORACLE

9-13

获取控制文件信息

要获取控制文件的位置和名称，请查询 V\$CONTROLFILE 视图。

```
SELECT name FROM V$CONTROLFILE;

NAME
-----
D:\APP\111973004\ORADATA\TEST\CONTROL01.CTL
D:\APP\111973004\ORADATA\TEST\CONTROL02.CTL
2 rows selected.
```

也可以使用 V\$PARAMETER 视图。

```
SELECT name, value from V$PARAMETER
WHERE name = 'control_files';

NAME                                VALUE
-----                                -
----control_files
D:\APP\111973004\ORADATA\TEST\CONTROL01.CTL,
```

D:\APP\111973004\ORADATA\TEST\CONTROL02.CTL

获取控制文件信息（续）

要获取有关控制文件不同部分的信息，请查询 V\$CONTROLFILE_RECORD_SECTION 视图。

```
SQL> SELECT type, record_size, records_total, records_used
2  FROM v$controlfile_record_section
3  WHERE TYPE='DATAFILE';
```

TYPE	RECORD_SIZE	RECORDS_TOTAL	RECORDS_USED
DATAFILE	180	40	10

1 row selected.

RECORDS_TOTAL 列指定分配给某特定部分的记录数。例如，在以上示例中，您可以看到数据文件的最大数为 30，这个数字由 CREATE DATABASE 命令中的 AXDATAFILES 参数确定。

也可以使用 SHOW PARAMETER 命令找到控制文件的位置。

```
SQL> SHOW PARAMETER control_files;
```

NAME	TYPE	VALUE
control_files	string	D:\APP\111973004\ORADATA\TEST\CONTROL01.CTL, D:\APP\111973004\ORADATA\TEST\CONTROL02.CTL

几个动态性能视图中的信息可以从控制文件中获得。下面是一些示例：

- V\$BACKUP
- V\$DATAFILE
- V\$TEMPFILE
- V\$TABLESPACE
- V\$ARCHIVE
- V\$LOG
- V\$LOGFILE

- V\$LOGHIST
- V\$ARCHIVED_LOG
- V\$DATABASE

小结

在这一课中，您应该能够掌握：

- 使用 **SPFILE** 对控制文件进行多元备份
- 使用 **init.ora** 对控制文件进行多元备份
- 使用 **OMF** 管理控制文件