

# **AUL(MyDUL)**

## **Oracle 异常数据恢复指南**

---

Ver: 1.0

作者: 楼方鑫

微信: anysql

邮件: anysql(at)126.com,  
anysql(at)live.com

从 2005 年开始, AUL (MyDUL) 已经为全球不同国家及地区的众多客户恢复了数十 TB 计的 Oracle 数据, 从损坏的 Oracle 8, Oracle 8i, Oracle 9i, Oracle 10g, Oracle 11g, Oracle 12c, Oracle 18c, Oracle 19c 及 Oracle ASM 上为客户快速恢复数据. AUL (MyDUL) 可以脱离 Oracle 运行环境, 直接从数据文件中读取记录, 与官方工具 Oracle DUL 具有同等功效并且功能更加丰富. 当你遇到下列极端情况, 并且没有有效备份 (客户有备份动作, 备份不起作用的情况也遇到过) 用来恢复数据时, AUL (MyDUL) 是往往是你最后的机会. 一直坚持 “拯救数据, 帮助客户” 的原则! 在最新版本 AUL 6 中, 可以直接访问 Oracle ASM 来恢复数据, 或从 Oracle ASM 中将数据文件拷贝出来。

针对以下场景, AUL 可以有效地进行数据恢复:

1. 丢失系统表空间。
2. 系统表空间有坏块, 无法启动 Oracle 数据库。
3. 表空间被删除, 但数据文件还在。
4. 表被删除 (Drop) 后, 马上被发现, 释放的空间还没有被其他表重用。
5. 表被截断 (Truncate) 后, 马上被发现, 释放空间未被其他表重用。
6. 一个表空间丢失部份文件或文件中的部份损坏, 导致表无法正常访问。
7. 数据文件头被勒索病毒破坏或加密。
8. Oracle ASM 存储损坏或磁盘损坏。
9. 其他无法正常打开数据库的情况。

AUL (MyDUL) 并不提供免费服务, 没有许可证的情况下最多允许同时打开 10 个数据文件, 并且只能访问文件的前 512MB 内容, 要支持更多的数据文件或更大的数据文件恢复, 你必须获得许可证并在使用前进行注册. 另外一个免费工具 AUL for Oracle ASM (下载) 可以将存放在 Oracle ASM 中的数据文件拷到文件系统, 在 Oracle ASM 损坏或磁盘不可用时, 进行文件级的数据恢复, 在 AUL (MyDUL 6) 中也集成了这个工具的所有功能, 并且免费使用, 最大支持 2028 块盘的 Oracle ASM 存贮。

## 最新版本

AUL 最新版本为 6.6，增强了对 Oracle 19c 版本的正式支持，从 2005 年第一版开始到现在已经有 15 年历史了。AUL 为命令行工具，启动 AUL 后会看到如下信息。

```
Register Code: FHBR-SFEX-TJXK-WPRL-HUHI
AUL : AnySQL UnLoader(MyDUL) for Oracle 19c and ASM, release 6.6.0
(C) Copyright Lou Fangxin 2005-2020 (AnySQL.net), all rights reserved.
Registered version, you are welcome!
AUL>
```

其中第一行为注册码，需要购买许可时将第一行信息给我；第二行和第三行为软件版本信息；第四行为注册信息，在这台机器上已经注册成功，所以显示的是注册的版本，没有“同时打开 10 个数据文件，并且只能访问文件的前 512MB 内容”的限制。

## 购买许可

如果没有许可（请体谅一下，做软件研发真是投入很大的事情，Oracle 数据库也是极其复杂的，研发第一版时差点累到吐血），则启动软件后显示如下：

```
Register Code: FHBR-SFEX-TJXK-WPRL-HUHI
AUL : AnySQL UnLoader(MyDUL) for Oracle 19c and ASM, release 6.6.0
(C) Copyright Lou Fangxin 2005-2020 (AnySQL.net), all rights reserved.
Unregistered version, with 512MB data file size limited!
AUL>
```

需要将第一行的 Register Code 给我，就是“FHBR-SFEX-TJXK-WPRL-HUHI”，

这个 Register Code 在同一台机器上会固定不变（不重装系统），因此只需要注册一次，就可以多次使用，属于非常划算的投资，国内不少做数据恢复的同行用的都是 AUL 软件。

在这里生成的许可码为“ABPHDYJ”，启动 AUL 软件，然后输入“SET LICENCE 许可码”命令就会生成许可文件“AULLIC.DAT”，如下图所示：

```
Register Code: FHBR-SFEX-TJXK-WPRL-HUHI
AUL : AnySQL UnLoader(MyDUL) for Oracle 19c and ASM, release 6.6.0
(C) Copyright Lou Fangxin 2005-2020 (AnySQL.net), all rights reserved.
Unregistered version, with 512MB data file size limited!
AUL> SET LICENCE ABPHDYJ
    Registered, Elapsed: 484
AUL>
```

在启动 AUL 的目录（当前工作目录）中可以找到许可文件“AULLIC.DAT”，将许可文件拷到 Windows 系统目录（Windows 主机）或“/etc”目录（Linux/Unix 主机），就完成了软件注册。再次启动软件就会显示已注册。如下所示：

```
Register Code: FHBR-SFEX-TJXK-WPRL-HUHI
AUL : AnySQL UnLoader(MyDUL) for Oracle 19c and ASM, release 6.6.0
(C) Copyright Lou Fangxin 2005-2020 (AnySQL.net), all rights reserved.
Registered version, you are welcome!
AUL>
```

接下来就可以大显身手来进行数据恢复了。

## 恢复示例

在这里假定 SYSTEM 表空间没有被严重破坏，虽然 Oracle 数据库不能起动，但并不影响 AUL 从 SYSTEM 中读取必要的字典信息。首先需要有一个数据文件列表，需要包含系统表空间和用户表空间，不需要临时表空间文件、UNDO 表空间文件、

控制文件、联机日志和归档日志文件。如下所示：

1. SYSTEM01.DBF
2. PDSCI.DBF

用任何文本编辑工具，依次写入文件列表（包含路径，不能有空格），每一行代表一个数据文件，可以使用“#”开头来表示注释。需要将 SYSTEM 表空间文件写在最第一行，因为系统字典信息是扫描获取的，放在第一个可以加快扫描速度。假配置文件名字为“db.txt”，接下来在 AUL 软件中运行“open db.txt”命令，如下所示：

```
Register Code: FHBR-SFEX-TJXK-WPRL-HUHI
AUL : AnySQL UnLoader(MyDUL) for Oracle 19c and ASM, release 6.6.0
(C) Copyright Lou Fangxin 2005-2020 (AnySQL.net), all rights reserved.
Registered version, you are welcome!

AUL> open db.txt
* ts#  rfn ver bsize      blocks      sizemb hsize filename
-----
Y   0   1 a2  8192      35840       280      0 SYSTEM01.DBF
Y   3  61 a2  8192     2576640     20130      0 PDSCI.DBF
AUL>
```

可以看到数据文件被成功打开了，一些关键信息被自动读出来。每个列的含义如下：

1. \*, 用来表示是否能自动识别到关键信息，“Y”表能，“N”表示不能。
  2. ts#, 表空间编号，如果数据文件文件头破坏，这此值为 0，不影响使用。
  3. rfn, 文件编号，数据文件在 Oracle 中的内部编号，非常关键的信息。
  4. ver, 数据块格式的版本号，可以见到的值一般有“02”和“a2”。
  5. bsize, 数据库大小，这个不是从文件中读取出来的，而是通过“set block\_size 大小”来进行设置的，默认的设置 8192，刚好对上了。
- AUL 只能处理同一种块大小的数据文件，如果不同的表空间用不同的块

大小，则需要按表空间来进行恢复。

6. blocks, 可以访问的数据块的数量, 也表示了数据文件大小。
7. sizemb, 数据文件的大小, 通过 blocks 换算而来的。
8. hsize, 头部保留空间, 仅用于 AIX 等机器的裸设备场景, 可能操作系统保留了前 4K 空间, 一般情况下都为 0。
9. filename, 数据文件名。

如果看不到这些信息, 则可能不是 Oracle 的数据文件, 或者块大小设错了, 或者数据文件已经完全损坏, 比如从磁盘阵列中恢复出来的文件错位了。

## 字典信息

字典信息只存在于系统表空中, 这一步最好只打开系统表空间数据文件, 而不要包含用户表空间文件, 以加快速度节约时间。在 AUL 软件中运行 “UNLOAD TABLE DICT\$” 命令 (命令后面加分号), 如下所示:

```
AUL> open db.txt
* ts# rfn ver bsize      blocks      sizemb hsize filename
-----
Y    0    1 a2  8192      35840        280      0 SYSTEM01.DBF
AUL> UNLOAD TABLE DICT$;
2020-07-08 20:24:21
2020-07-08 20:24:22
AUL>
```

系统表空间一般不会太大, 这一步会比较快, 这一步会在 “AULDICT” 目录中生成数据字典信息, 供后续的数据恢复使用。在这一步完成后, 就可以在 AUL 中使用 “DESC 用户. 表名” 来查看表结构了。如下所示:

```
AUL> DESC SYS.PROPS$;

Storage(OBJ#=126 OBJD=126 TS=0 FILE=1 BLOCK=1096 CLUSTER=0)
No. SEQ INT Column Name                                     Type
-----
  1   1   1 NAME                                           VARCHAR2(128) NOT NULL
  2   2   2 VALUE$                                           VARCHAR2(4000)
  3   3   3 COMMENT$                                         VARCHAR2(4000)
AUL>
```

如果能看到表结构, 则表示系统字典信息是好的。其中 “Storage” 这一行显示了目标表的存储信息, 各字段信息如下所示:

1. OBJ#, 对象编号
2. OBJD, 数据编号, 在数据文件中只有此编号和真实数据, 没有结构信息。  
因表结构的信息都存放在 SYSTEM 表空间中, 如果没有 SYSTEM 表空间, 能过扫描数据文件, 则只能得到一个编号和真实数据, 无法精确知道数据是属于哪个表, 需要对应用非常熟悉了解的人来进行匹配。
3. TS, 目标表所在的表空间编号。
4. FILE, 目标表第一个数据块 (Segment Header) 所在数据文件编号。
5. BLOCK, 目标表第一个数据块 (Segment Header) 所在数据块位置。
6. CLUSTER, 如果多个表建在 Cluster 上, 则表示在 Cluster 上的位置, 如果为 0 表示此表不是建在 Cluster 上的, Oracle 系统表有很多是基于 Cluster 存放的。

接下来还需要来生成数据分配信息, 假设数据文件是坏的, 需要通过扫描所有数据文件的方式, 来知道不同表的存放信息, 以加速后续的数据恢复。

## 分配信息

这一步需要打开所有的用户数据文件, 如果要恢复系统表空间上的对象, 或有数据表建在系统表空间上, 也需要包含进来。然后运行 “SCAN DATABASE” 命令来扫描生成空间分配信息, 这一步需要访问所有的数据块, 会耗时比较长, 需要耐心等待。还好这一步和前面的字典信息, 针对同一个数据库仅需要执行一次, 而不是每次起动 AUL 都需要执行一次。如下所示:

```
AUL> open db.txt
*  ts#  rfn ver bsize      blocks      sizemb  hsize filename
-----
Y    0    1 a2  8192      35840        280      0 SYSTEM01.DBF
Y    3   61 a2  8192     2576640     20130      0 PDSCI.DBF
AUL> scan database
2020-07-08 20:40:14
2020-07-08 20:45:15
AUL>
```

接下来就可以进行数据恢复了。

## 字符集信息

接下来需要获取数据库字符集信息，如果系统表空间是好的，可以从“SYS.PROPS\$”表中获取，就是前面我们查看过表结构的那张表，如下所示：

```
AUL> DESC SYS.PROPS$;

Storage(Obj#=126 OBJD=126 TS=0 FILE=1 BLOCK=1096 CLUSTER=0)
No. SEQ INT Column Name                                     Type
-----
  1   1   1 NAME                                             VARCHAR2(128) NOT NULL
  2   2   2 VALUE$                                           VARCHAR2(4000)
  3   3   3 COMMENT$                                         VARCHAR2(4000)

AUL>
```

接下来运行“UNLOAD TABLE SYS.PROPS\$”命令，就可以看到这个表的内容了，不同字段之间默认使用竖线分隔。如下所示：

```
AUL> UNLOAD TABLE SYS.PROPS$;
2020-07-08 20:49:23
Unload OBJD=126 FILE=1 BLOCK=1096 CLUSTER=0 ...
DICT.BASE|2|dictionary base tables version #
DEFAULT_TEMP_TABLESPACE|TEMP|Name of default temporary tablespace
DEFAULT_PERMANENT_TABLESPACE|SYSTEM|Name of default permanent tablespace
DEFAULT_EDITION|ORA$BASE|Name of the database default edition
Flashback Timestamp TimeZone|GMT|Flashback timestamp created in GMT
TDE_MASTER_KEY_ID
DBTIMEZONE|-05:00|DB time zone
DEFAULT_TBS_TYPE|SMALLFILE|Default tablespace type
GLOBAL_DB_NAME|JSZY|Global database name
NLS_RDBMS_VERSION|12.1.0.2.0|RDBMS version for NLS parameters
NLS_NCHAR_CHARACTERSET|AL16UTF16|NCHAR Character set
NLS_NCHAR_CONV_EXCP|FALSE|NLS conversion exception
NLS_LENGTH_SEMANTICS|BYTE|NLS length semantics
NLS_COMP|BINARY|NLS comparison
NLS_DUAL_CURRENCY|$|Dual currency symbol
NLS_TIMESTAMP_TZ_FORMAT|DD-MON-RR HH.MI.SSXFF AM TZR|Timestamp with time zone
NLS_TIME_TZ_FORMAT|HH.MI.SSXFF AM TZR|Time with time zone format
NLS_TIMESTAMP_FORMAT|DD-MON-RR HH.MI.SSXFF AM|Time stamp format
NLS_TIME_FORMAT|HH.MI.SSXFF AM|Time format
NLS_SORT|BINARY|Linguistic definition
NLS_DATE_LANGUAGE|AMERICAN|Date language
NLS_DATE_FORMAT|DD-MON-RR|Date format
NLS_CALENDAR|GREGORIAN|Calendar system
NLS_CHARACTERSET|ZHS16GBK|Character set
```



找到第一列为“NLS\_CHARACTERSET”的那一行，表示了数据库的字符集，在这里是“ZHS16GBK”；接下来找到“NLS\_NCHAR\_CHARACTERSET”开头的一行，表示了数据库的 NCHAR 字符集，在这里是“AL16UTF16”。这两个信息需要记住，每次重新启动 AUL，都需要重新设置一次，如下所示：

```
AUL> SET CHARSET ZHS16GBK
Current CHARSET is : 0x0354 (852)
AUL> SET NLSCHARSET AL16UTF16
Current NLSCHARSET is : 0x07d0 (2000)
AUL>
```

AUL 会将字符集的名字自动转化为字符集的内部编号，内部集成了 Oracle 19c 所支持的所有字符集，应当是非常齐全了。

## 数据恢复

在前面获取字符集信息时，已经相当于恢复了一个表了，就是使用“UNLOAD TABLE”命令来进行数据恢复。我们来恢复一下“SYSTEM”用户下的“HELP”表看看，只需要运行“UNLOAD TABLE SYSTEM.HELP TO HELP.TXT”命令，如下所示：

```
AUL> UNLOAD TABLE SYSTEM.HELP TO HELP.TXT;
2020-07-08 21:06:57
Unload OBJD=20369 FILE=1 BLOCK=18936 CLUSTER=0 ...
Sucessfully unload 938 rows ...
2020-07-08 21:06:57
AUL>
```

可以看到成功恢复了 938 条记录。默认的恢复格式是文本方式，将数据恢复成格式化文本文件，然后同步生成建表的 SQL 语句文件和用于 SQL\*Loader 工具装载的控制文件，分别为：

1. HELP.TXT，数据文件
2. HELP\_syntax.sql，基本建表语句，无分区、索引、约束等信息。

3. HELP\_sqlldrctl, 用于 SQL\*Loader 工具装载的控制文件。

有了这三个文件，可以很方便地进行数据恢复。可以在数据文件中找到以下信息：

```
ARCHIVE LOG|1
ARCHIVE LOG|2| ARCHIVE LOG
ARCHIVE LOG|3| -----
ARCHIVE LOG|4
ARCHIVE LOG|5| Displays information about redo log files.
ARCHIVE LOG|6
ARCHIVE LOG|7| ARCHIVE LOG LIST
ARCHIVE LOG|8
```

可以在建表语句文件中看到以下内容：

```
CREATE TABLE "HELP" (
  "TOPIC" VARCHAR2(50) NOT NULL ,
  "SEQ" NUMBER NOT NULL ,
  "INFO" VARCHAR2(80)
);
exit;
```

可以在 SQL\*Loader 工具装载的控制文件看到以下内容：

```
--
-- Generated by AUL/MyDUL, for table SYSTEM.HELP
--
OPTIONS(BINDSIZE=8388608, READSIZE=8388608, ERRORS=2147483647, ROWS=50000)
LOAD DATA
INFILE 'HELP.TXT' "STR X'0d0a'"
APPEND INTO TABLE HELP
FIELDS TERMINATED BY X'7c' TRAILING NULLCOLS
(
  TOPIC      CHAR(50) ,
  SEQ        CHAR ,
  INFO       CHAR(80)
)
```

接下来只需要运行 SQL\*Loader 工具来进行数据恢复（请自行在目标用户下创建表结构），在操作系统下（需要在机器上安装 Oracle 客户端及工具，或者将

文件拷到 Oracle 服务器上运行，需要注意设置准确的字符集环境变量）运行命令“sqlldr user/password control=HELP\_sqlldr.ctl”。先在“SCOTT”用户下创建“HELP”表，如下所示：

```
SQL> CREATE TABLE "HELP" (  
2  "TOPIC"  VARCHAR2(50) NOT NULL ,  
3  "SEQ"    NUMBER NOT NULL ,  
4  "INFO"   VARCHAR2(80)  
5  );  
表已创建。  
SQL> DESC HELP  
名称                                是否为空? 类型  
-----  
TOPIC                                NOT NULL  VARCHAR2(50)  
SEQ                                  NOT NULL  NUMBER  
INFO                                VARCHAR2(80)  
SQL>
```

然后使用 SQL\*Loader 工具来装载数据，如下所示：

```
D:\BaiduNetdiskDownload\jszy>sqlldr scott/tiger control=HELP_sqlldr.ctl  
SQL*Loader: Release 19.0.0.0.0 - Production on 星期三 7月 8 22:12:27 2020  
Version 19.3.0.0.0  
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.  
所用路径: 常规  
达到提交点 - 逻辑记录计数 938  
表 HELP:  
已成功载入 938 行。  
查看日志文件:  
HELP_sqlldr.log  
了解有关加载的详细信息。
```

可以看到全部 938 条记录装载成功，可以到 SQL\*Plus 中去查一下记录数，如下所示：

```
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.3.0.0.0  
SQL> select count(*) from help;  
  
COUNT(*)  
-----  
938
```

可以在数据库中查到一致的记录数，说明数据恢复完全成功了。

将数据恢复成文本格式时，由于真实数据千变成化，需要注意设置准确的字

段与字段之间的分隔符号（“FIELD\_TAG”）以及记录与记录之间的分隔符号（“RECORD\_TAG”）。字段分隔符默认为竖线（“|”），记录分隔符默认为换行符，当表中数据本身会包含竖线或换行符时，就需要显式设置分隔符号了。可以设置任何不会出现在数据库的字符或字符串，或使有用“\xXX”格式来设置任何字符，如下所示：

```
AUL> set field_tag <field>
  Current FIELD_TAG :<field>
AUL> set record_tag <row>\x0a
  Current RECORD_TAG :<row>\n
AUL> set field_tag \x07
  Current FIELD_TAG : \x07
AUL> set record_tag \x06
  Current RECORD_TAG : \x06
AUL>
```

例如：

```
AUL> open db.txt
* ts# rfn ver bsize blocks sizemb hsize filename
-----
Y 0 1 a2 8192 35840 280 0 SYSTEM01.DBF
Y 3 61 a2 8192 2576640 20130 0 PDSCI.DBF
AUL> set field_tag <field>
  Current FIELD_TAG :<field>
AUL> set record_tag <row>\x0a
  Current RECORD_TAG :<row>\n
AUL> unload table sys.props$ limit 10;
2020-07-09 05:55:58
Unload OBJD=126 FILE=1 BLOCK=1096 CLUSTER=0 ...
DICT.BASE<field>2<field>dictionary base tables version #<row>
DEFAULT_TEMP_TABLESPACE<field>TEMP<field>Name of default temporary tablespace<row>
DEFAULT_PERMANENT_TABLESPACE<field>SYSTEM<field>Name of default permanent tables
DEFAULT_EDITION<field>ORA$BASE<field>Name of the database default edition<row>
Flashback Timestamp TimeZone<field>GMT<field>Flashback timestamp created in GMT<
TDE_MASTER_KEY_ID<row>
DBTIMEZONE<field>-05:00<field>DB time zone<row>
DEFAULT_TBS_TYPE<field>SMALLFILE<field>Default tablespace type<row>
GLOBAL_DB_NAME<field>JSZY<field>Global database name<row>
NLS_RDBMS_VERSION<field>12.1.0.2.0<field>RDBMS version for NLS parameters<row>
Sucessfully unload 10 rows ...
2020-07-09 05:55:58
```

可以看到恢复出的数据中，字段及记录分隔符不再是默认的竖线和换行了，对应生成的 SQL\*Loader 控制文件也会相应变化，不影响数据载装。当然也可以将数据恢复成 Oracle Dump 格式（简称“DMP 格式”），就不需要这么小心地设置

字段与记录分隔符了。可以在“UNLOAD”命令之前执行“SET OUTPUT\_STYLE {DMP | TXT}”来切换文格格式或 DMP 格式。如下所示：

```
AUL> open db.txt
*  ts#  rfn ver bsize      blocks    sizemb  hsize filename
-----
Y    0    1 a2  8192      35840      280    0 SYSTEM01.DBF
Y    3   61 a2  8192     257664     20130    0 PDSCI.DBF
AUL> SET CHARSET ZHS16GBK
Current CHARSET is : 0x0354 (852)
AUL> SET NLSCHARSET AL16UTF16
Current NLSCHARSET is : 0x07d0 (2000)
AUL> SET OUTPUT_STYLE DMP
Current OUTPUT_STYLE is : DMP
AUL> UNLOAD TABLE SYSTEM.HELP TO HELP.dmp;
2020-07-09 06:04:08
Unload OBJD=20369 FILE=1 BLOCK=18936 CLUSTER=0 ...
Sucessfully unload 938 rows ...
2020-07-09 06:04:08
AUL>
```

接下来使用 Import 工具进行导入，如下所示产：

```
D:\BaiduNetdiskDownload\jszy>imp system/oracle file=HELP.dmp fromuser=system touser=scott ignore=y
Import: Release 19.0.0.0.0 - Production on 星期四 7月 9 06:08:21 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

连接到: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

经由常规路径由 EXPORT:V08.01.07 创建的导出文件
已经完成 ZHS16GBK 字符集和 AL16UTF16 NCHAR 字符集中的导入
IMP-00403:

警告: 此导入生成了单独的 SQL 文件 "import_sys", 其中包含了由于权限问题而失败的 DDL。

. 正在将 SYSTEM 的对象导入到 SCOTT
. . 正在导入表 "HELP"导入了 938 行
成功终止导入, 但出现警告。
```

可以看到已经成功导入 938 行记录,和前面文本方式的导入记录数完全一致。DMP 格式的生成需要完整的数据字典信息,即要求 SYSTEM 表空间相对 AUL 是完好的,比如一个月之前的系统表空间文件备份,对 Oracle 来讲可能时间差距过大,但对 AUL 来讲信息也是完整的(不包含最近一个月内创建或重建的表)。

(未完待续)