

# 达梦技术手册

## DM8\_dmdbchk 使用手册

Service manual of DM8\_dmdbchk



# 前言

## 概述

本文档主要介绍如何使用 DM 的用于检查数据文件完整性的命令行工具 dmdbchk，该工具所提供的功能、及其参数的使用方法。

## 读者对象





本文档主要适用于 DM 数据库的：

- 开发工程师
- 测试工程师
- 技术支持工程师
- 数据库管理员

## 通用约定

在本文档中可能出现下列标志，它们所代表的含义如下：

表 0.1 标志含义

标志	说明
 <b>警告：</b>	表示可能导致系统损坏、数据丢失或不可预知的结果。
 <b>注意：</b>	表示可能导致性能降低、服务不可用。
 <b>小窍门：</b>	可以帮助您解决某个问题或节省您的时间。
 <b>说明：</b>	表示正文的附加信息，是对正文的强调和补充。

在本文档中可能出现下列格式，它们所代表的含义如下：

表 0.2 格式含义

格式	说明
宋体	表示正文。
黑体	标题、警告、注意、小窍门、说明等内容均采用黑体。
Courier new	表示代码或者屏幕显示内容。
粗体	表示命令行中的关键字（命令中保持不变、必须照输的部分）或者正文中强调的内容。
<>	语法符号中，表示一个语法对象。
::=	语法符号中，表示定义符，用来定义一个语法对象。定义符左边为语法对象，右边为相应的语法描述。
	语法符号中，表示或者符，限定的语法选项在实际语句中只能出现一个。
{ }	语法符号中，大括号内的语法选项在实际的语句中可以出现 0...N 次 (N 为大于 0 的自然数)，但是大括号本身不能出现在语句中。
[ ]	语法符号中，中括号内的语法选项在实际的语句中可以出现 0...1 次，但是中括号本身不能出现在语句中。
关键字	关键字在 DM_SQL 语言中具有特殊意义，在 SQL 语法描述中，关键字以大写形式出现。但在实际书写 SQL 语句时，关键字既可以大写也可以小写。

## 访问相关文档

如果您安装了 DM 数据库，可在安装目录的“\doc”子目录中找到 DM 数据库的各种手册与技术丛书。

您也可以通过访问我们的网站 [www.dameng.com](http://www.dameng.com) 阅读或下载 DM 的各种相关文档。

## 联系我们

如果您有任何疑问或是想了解达梦数据库的最新动态消息，请联系我们：

网址：[www.dameng.com](http://www.dameng.com)

技术服务电话：400-991-6599

技术服务邮箱：[dmtech@dameng.com](mailto:dmtech@dameng.com)

# 目录

1 功能简介 .....	1
2 使用 dmdbchk .....	2
3 查看 dmdbchk 参数 .....	2
4 dmdbchk 报告解读 .....	3
5 应用实例 .....	3

# 1 功能简介

dmdbchk 是 DM 提供的用于检查数据库完整性、正确性的命令行工具。在服务器正常关闭后的脱机情况下，用户可以使用 dmdbchk 对数据库进行校验，包括校验 DM 数据库内部的物理存储结构是否正常，对象信息是否合法等。除此之外，dmdbchk 工具还可以检测并删除操作系统中残留的信号量和共享内存，避免达到操作系统上限后数据库服务无法启动。

检验的内容具体包括三方面：

## 1. 数据文件合法性检验。

数据文件校验只校验数据文件大小。如果数据库文件实际大小大于或等于文件头中记录的大小，则合法。

## 2. 索引正确性校验。

索引检验包括：普通表 B 树索引校验、LIST 表扁平 B 树索引校验、列存储表索引校验。校验内容具体包括：B 树的层次、层次关系，每层的内部页、叶子页以及页之间的前后链接关系，每个页的页头信息（如页类型等）等。

## 3. 对象 ID 合法性校验。

对象 ID 校验包括数据库内的所有对象的 ID 检验。对象包括：索引、表、约束、存储过程、模式、同义词、用户等。如果从系统表中查出的对象 ID 小于库的 ID 预留页中记录的该类型对象的下一分配 ID，则合法。因为库的 ID 预留页中记录的是各类型对象下一个新分配对象将使用的 ID，因此若系统表中的对象 ID 大于或等于该 ID，说明库已损坏。



**dmdbchk 并不能检查出用户实际数据的正确性，如果数据文件被手动人为修改，正好修改到的是用户数据部分，是检查不出来的。**

## 4. 残留信号量和共享内存的检测及删除

一般 Linux 下默认可创建的信号量个数为 128 个，共享内存为 4096 个。DM 数据库服务器正常退出的情况下，会将创建的信号量或共享内存删除，但如果服务器异常退出，这些信号量及共享内存会一直留在操作系统中直到通过手工方式删除。如果残留的信号量或共享内存个数达到操作系统上限，服务器再次启动时，会因为创建信号量或共享内存失败而无法启动成功，因此在本工具中提供对 Linux 残留信号量和共享内存的检测及删除命令。

校验完毕后，dmdbchk 会在当前目录下（dmdbchk 所在目录）生成一个名为 dbchk\_err.txt 的检查报告，供用户查看。

该工具位于安装目录的/dmdbms/bin 目录下。

## 2 使用 dmdbchk

dmdbchk 工具需要从命令行启动。找到 dmdbchk 所在安装目录/dmdbms/bin，输入 dmdbchk 和参数后回车。参数在下一节详细介绍。



注意：

dmdbchk 工具不同的参数用法也不同，如果是校验数据库的参数，则要求必须是经过初始化，且正常关闭的数据库，才能被 dmdbchk 工具校验，否则 dmdbchk 会报错退出。

例如，正常关闭/home/test/dmdbms 目录下数据库后，使用 dmdbchk 对数据库进行校验。

```
./dmdbchk PATH=/home/test/dmdbms/dm.ini
```

如果 dm.ini 在 dmdbchk 当前目录，也可以写成这样：

```
./dmdbchk PATH=dm.ini
```

## 3 查看 dmdbchk 参数

dmdbchk 使用简单灵活。用户可使用“dmdbchk HELP”快速查看参数用法。

```
./dmdbchk HELP
```

格式：dmdbchk.exe KEYWORD=value

例程：dmdbchk.exe path=d:\dmdbms\bin\dm.ini

关键字说明（默认）

PATH	dm.ini 绝对路径或者当前目录的 dm.ini
DCR_INI	dmdcr.ini 的路径
HELP	打印帮助信息
START_INDEXID	最小检查索引号
END_INDEXID	最大检查索引号

CHECK_SEMA	检查当前系统信号量使用情况（只适用于 LINUX，1：只做检查 2：检查并删除残留信号量）
CHECK_SHM	检查当前系统共享内存使用情况（只适用于 LINUX，1：只做检查 2：检查并删除残留共享内存）

## 4 dmdbchk 报告解读

dmdbchk 校验的过程对用户不可见，校验的结果以报告的形式呈现给用户。

dmdbchk 校验数据库的报告内容分为七部分：一 dmdbchk 版本信息；二开始标志；三数据文件校验结果；四索引校验结果；五对象 ID 校验结果；六结束标志；七错误总数。



图 4.1 dmdbchk 报告内容

数据文件校验结果、索引校验结果、对象 ID 校验结果三部分内容伸缩性较大，内容多少由数据库大小决定。对于数据文件校验结果和索引校验结果，如果校验成功，直接打印出具体校验的对象；如果校验失败，则打印出具体错误信息。对于对象 ID 校验结果，如果校验成功，不打印出具体校验的对象；如果校验失败，则打印出具体错误信息。

dmdbchk 检测信号量或共享内存的报告则比较简单，分为三部分：一 dmdbchk 版本信息；二系统中所有信号量或共享内存的检测结果；三各项结果总数归类。

## 5 应用实例

**例 1**：例如，初始化一个数据库，并创建 DM 的示例库 BOOKSHOP，正常关闭后，使用

dmdbchk 工具对该库的数据文件进行校验。

```
./dmdbchk PATH=/home/test/dmdbms/dm.ini START_INDEXID=33554433
END_INDEXID=33555531
```

校验后的报告存在 dmdbchk 工具所在的目录里，名称为 dbchk\_err.txt。报告内容如下：

```
/**一 dmdbchk 版本信息**/
[2015-12-31 16:57:29] dmdbchk V7.1.5.42-Build(2015.12.30-64335trunc)
/**二开始标志**/
[2015-12-31 16:57:30] DM DB CHECK START.....
/**三数据文件校验结果**/
[2015-12-31 16:57:30] -----check dbf file size start-----
[2015-12-31 16:57:30] FILE=(ts_id=0, fil_id=0, path=D:\xx\DAMENG\SYSTEM.DBF)
[2015-12-31 16:57:30] FILE=(ts_id=1, fil_id=0, path=D:\xx\DAMENG\ROLL.DBF)
[2015-12-31 16:57:30] FILE=(ts_id=4, fil_id=0, path=D:\xx\DAMENG\MAIN.DBF)
[2015-12-31 16:57:30] FILE=(ts_id=5, fil_id=0, path=D:\xx\DAMENG\BOOKSHOP.DBF)
[2015-12-31 16:57:30] -----check dbf file size end-----
/**四索引校验结果**/
[2015-12-31 16:57:30] -----check indexes start-----
[2015-12-31 16:57:30] INDEX=(id=33554433, name=SYSINDEXCOLUMNS,
table_name=SYSCOLUMNS)
[2015-12-31 16:57:30] INDEX=(id=33554434, name=SYSINDEXINDEXES,
table_name=SYSINDEXES)
[2015-12-31 16:57:30] INDEX=(id=33554440, name=SYSINDEXTUSERS,
table_name=SYSUSER$)
[2015-12-31 16:57:30] INDEX=(id=33554442, name=SYSINDEXSYSGRANTS,
table_name=SYSGRANTS)
[2015-12-31 16:57:30] INDEX=(id=33554452, name=SYSINDEXCONSTRAINTS,
table_name=SYSCONS)
[2015-12-31 16:57:30] INDEX=(id=33554458, name=SYSINDEXSYSAUDIT,
table_name=SYSAUDIT)
[2015-12-31 16:57:30] INDEX=(id=33554459, name=SYSINDEXSYSAUDITSQLSEQ,
table_name=SYSAUDITSQLSEQ)
[2015-12-31 16:57:30] INDEX=(id=33554464, name=SYSINDEXCONTEXTINDEXES,
table_name=SYSCONTEXTINDEXES)
.....
省略一部分索引校验结果.....
.....
[2015-12-31 16:57:30] INDEX=(id=33555528, name=INDEX33555528, table_name=EMPTAB)
[2015-12-31 16:57:30] INDEX=(id=33555529, name=INDEX33555529, table_name=EMPTAB)
[2015-12-31 16:57:30] INDEX=(id=33555530, name=INDEX33555530,
table_name=SALGRADE)
[2015-12-31 16:57:30] INDEX=(id=33555531, name=INDEX33555531,
```



```

table_name=COMPANYHOLIDAYS)
[2015-12-31 16:57:30] -----check indexes end-----
/**五对象 ID 校验结果**/
[2015-12-31 16:57:30] -----check iid start-----
[2015-12-31 16:57:30] check cons id ...
[2015-12-31 16:57:30] check index id ...
[2015-12-31 16:57:30] check table id ...
[2015-12-31 16:57:30] check proc id ...
[2015-12-31 16:57:30] check schema id ...
[2015-12-31 16:57:30] check synonym id ...
[2015-12-31 16:57:30] check user id ...
[2015-12-31 16:57:30] -----check iid end-----
/**六结束标志**/
[2015-12-31 16:57:30] DM DB CHECK END.....
/**七错误总数**/
[2015-12-31 16:57:30] error count is 0

```

**例 2：**例如，检测某台 linux 机器上信号量的使用情况。

```
./dmdbchk CHECK_SEMA=1
```

检测后的报告存在 dmdbchk 工具所在的目录里，名称为 dbchk\_err.txt。报告内容如下：

```

/**一 dmdbchk 版本信息**/
[2018-11-05 14:54:44] dmdbchk V8.0.0.105-Build(2018.10.31-98635-debug)ENT
/**二所有信号量检测结果**/
[2018-11-05 14:54:44] get semid 40992768(key:0xcc020ab4) current value:0,
cur_time:1541400884, last op time:1540894468, need delete it!
[2018-11-05 14:54:44] get semid 41025537(key:0xcd020ab4) current value:0,
cur_time:1541400884, last op time:1540894468, need delete it!
[2018-11-05 14:54:44] get semid 53116930(key:0xcd02111d) current value:0,
cur_time:1541400884, last op time:1541318198, need delete it!
/**三总数归类**/
[2018-11-05 14:54:44] check os semaphore finished:
[2018-11-05 14:54:44] total_cnt:3, active_cnt:0, check_err_cnt:0, need_del:3,
real_del:0, del_err_cnt:0!

```

**例 3：**例如，删除某台 linux 机器上的残留信号量。

```
./dmdbchk CHECK_SEMA=2
```

检测后的报告存在 dmdbchk 工具所在的目录里，名称为 dbchk\_err.txt。报告内容如下：

```
/**一 dmdbchk 版本信息**/
```

```
[2018-11-05 14:54:47] dmdbchk V8.0.0.105-Build(2018.10.31-98635-debug) ENT
/**二所有信号量检测结果**/
[2018-11-05 14:54:47] get semid 40992768(key:0xcc020ab4) current value:0,
cur_time:1541400887, otime:1540894468, delete it success!
[2018-11-05 14:54:47] get semid 41025537(key:0xcd020ab4) current value:0,
cur_time:1541400887, otime:1540894468, delete it success!
[2018-11-05 14:54:47] get semid 53116930(key:0xcd02111d) current value:0,
cur_time:1541400887, otime:1541318198, delete it success!
/**三总数归类**/
[2018-11-05 14:54:47] check os semaphore finished:
[2018-11-05 14:54:47] total_cnt:3, active_cnt:0, check_err_cnt:0, need_del:3,
real_del:3, del_err_cnt:0!
```

咨询热线：400-991-6599

技术支持：dmtech@dameng.com

官网网址：www.dameng.com



**武汉达梦数据库有限公司**  
**Wuhan Dameng Database Co.,Ltd.**

地址：武汉市东湖新技术开发区高新大道999号未来科技大厦C3栋16—19层

16th-19th Floor, Future Tech Building C3, No.999 Gaoxin Road, Donghu New Tech Development Zone,Wuhan,Hubei Province,China

电话：(+86) 027-87588000 传真：(+86) 027-87588810