


# Dmlog小工具使用简要

Dmlog小工具可以用作分析已经形成的logcommit，直接形成直观的excel,判断哪些是执行时间较长的SQL，频率较高的SQL，从而优化整个事务，使应用效率更高。

 Dmlog\_DM7\_v5.1.jar

以此版本为例，使用过程纪要如下：

1 在windows环境中，开启一个dos窗口，执行以下内容

```
java -jar E:\javahome\jdk1.6.0_4564\bin\Dmlog_DM7_v5.1.jar
```

执行后即打开工具，有以下内容需要填写，按照顺序为：

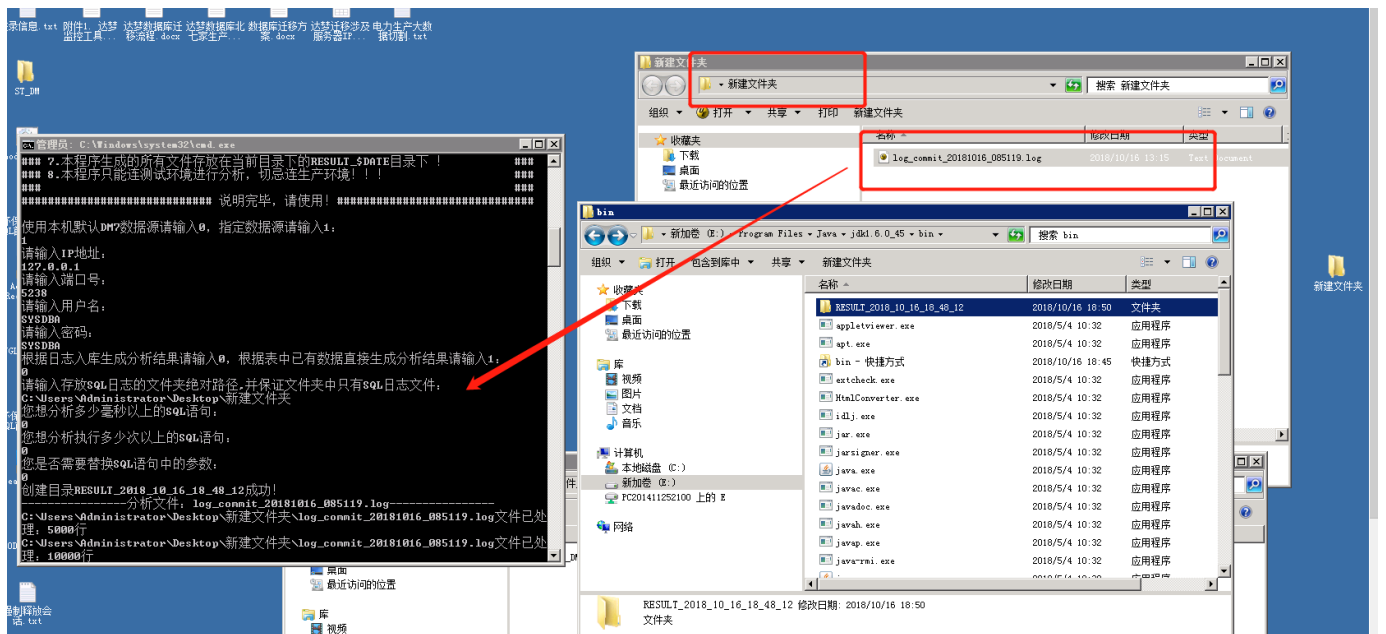
IP地址：本电脑数据库地址

端口号：本地库端口号

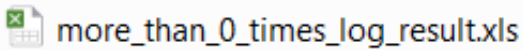
用户名，密码：本地数据库信息

```
E:\Program Files\Java\jdk1.6.0_45\bin>java -jar E:\达梦数据库相关软件_20161129\log_commit日志分析工具\Dmlog_DM7_v5.1.jar
##### dm7日志分析程序使用说明 #####
###
### 1.请确认sql trace参数，确保每条语句后紧跟sql语句时间：1:25 ###
### 2.本程序建表log_commit进行分析 ###
### 3.本程序建表前会删除同名表，请做好备份！ ###
### 4.请使用页大小为32k的DM7进行分析！ ###
### 5.结果中sql语句背景为黄色的表示sql长度超过30000,已截断！ ###
### 6.截断的语句会保存到文本文件中，如第一条截断会生成Q1.txt！ ###
### 7.本程序生成的所有文件存放在当前目录下的RESULT_$DATE目录下！ ###
### 8.本程序只能连测试环境进行分析，切忌连生产环境！！ ###
###
##### 说明完毕，请使用！ #####
使用本机默认DM7数据源请输入0，指定数据源请输入1：
1
请输入IP地址：
127.0.0.1
请输入端口号：
5238
请输入用户名：
SYSDBA
请输入密码：
SYSDBA
```

2 准备一个新的路径地址，存放一个单独的logcommit文件



按顺序填写自动跳出的内容并回车，会在开启这个小软件的路径下生成对应的Excel文件：



可以按照需求进行排序，查看SQL具体情况，示例如下：

1	序号	SQL语句	最大执行时间	最小执行时间	90%平均执行时间	平均执行时间	执行次数	优化方法	优化后时间
2	251	select to_char(count(*) m from ( select distinct	13907.0	13907.0	13907.0	13907.0	1		
3	208	select distinct aa.*, decode(trunc(aa.cxs	7738.0	7738.0	7738.0	7738.0	1		
4	204	select distinct aa.*, decode(trunc(aa.cxs	7616.0	7616.0	7616.0	7616.0	1		
5	234	select distinct aa.*, decode(trunc(aa.cxs	7480.0	7480.0	7480.0	7480.0	1		
6	207	select distinct aa.*, decode(trunc(aa.cxs	7313.0	7313.0	7313.0	7313.0	1		
7	193	select to_char(count(*) m from ( select distinct	6371.0	6371.0	6371.0	6371.0	1		
8	211	select to_char(count(*) m from ( select distinct	6260.0	6260.0	6260.0	6260.0	1		
9	217	select distinct aa.*, decode(trunc(aa.cxs	4726.0	4726.0	4726.0	4726.0	1		
10	135	select '0' as orgid, as text,0 as coms	27985.0	1015.0	4163.0	5662.0	36		
11	227	select distinct aa.*, decode(trunc(aa.cxs	4127.0	4127.0	4127.0	4127.0	1		
12	243	select to_char(count(*) m from ( select distinct	3951.0	3951.0	3951.0	3951.0	1		
13	238	select to_char(count(*) m from ( select distinct	3950.0	3950.0	3950.0	3950.0	1		
14	245	select to_char(count(*) m from ( select distinct	3933.0	3933.0	3933.0	3933.0	1		
15	197	select to_char(count(*) m from ( select distinct	3779.0	3779.0	3779.0	3779.0	1		
16	194	select to_char(count(*) m from ( select distinct	3742.0	3742.0	3742.0	3742.0	1		
17	190	select to_char(count(*) m from ( select distinct	3719.0	3719.0	3719.0	3719.0	1		
18	184	select to_char(count(*) m from ( select distinct	3713.0	3713.0	3713.0	3713.0	1		
19	229	select distinct aa.*, decode(trunc(aa.cxs	3669.0	3669.0	3669.0	3669.0	1		
20	240	select distinct aa.*, decode(trunc(aa.cxs	3590.0	3590.0	3590.0	3590.0	1		