



DBMaker

New Log System User's Guide

Version: 02.00

Document No: 54/DBM546-T10282024-01-NLSG
Author: DBMaker Support Team

Publication Date: Oct, 28 2024

Table of Content

1. Introduction	1-1
1.1. What is New Log System.....	1-1
1.2. What information will be logged	1-1
1.3. New Log System Constraints	1-1
2. Usage of New Log System	2-1
2.1. General Description.....	2-1
2.2. Keywords	2-1
2.2.1. DB_LGSVR	2-1
2.2.2. DB_LGDAY	2-1
2.2.3. DB_LGDIR	2-2
2.2.4. DB_LGERR.....	2-2
2.2.5. DB_LGSTM.....	2-2
2.2.6. DB_LGPLN	2-2
2.2.7. DB_LGSYS	2-2
2.2.8. DB_LGZIP.....	2-2
2.2.9. DB_LGFSZ	2-3
2.2.10. DB_LGFNO	2-3
2.2.11. DB_LGSQL.....	2-3
2.2.12. DB_LGPAR.....	2-3
2.2.13. DB_LGLCK.....	2-3
2.3. SETSYSTEMOPTION Stored Procedure	2-3
2.4. ADD TRACE command	2-4
2.5. Output Log Files	2-5
2.6. Output Log Information	2-6
3. Samples.....	3-1
3.1. DB_LGSVR.....	3-1
3.1.1. DB_LGSVR=1	3-1
3.1.2. DB_LGSVR=2.....	3-3
3.1.3. DB_LGSVR=3.....	3-4
3.1.4. DB_LGSVR=4.....	3-4
3.1.5. DB_LGSVR=5.....	3-5
3.1.6. DB_LGSVR=6.....	3-6
3.2. DB_LGSYS	3-7
3.2.1. DB_LGSYS=1	3-7

3.3.	DB_LGERR	3-8
3.3.1.	DB_LGERR=1.....	3-8
3.3.2.	DB_LGERR=4.....	3-9
3.4.	DB_LGPAR	3-10
3.4.1.	DB_LGPAR=1.....	3-10
3.5.	DB_LGPLN	3-11
3.5.1.	DB_LGPLN=1	3-11
3.6.	DB_LGLCK	3-14
3.6.1.	DB_LGLCK=1	3-14
4.	Scenario	4-1
4.1.	Slow Operations	4-1
4.2.	Log Connections	4-1
4.2.1.	LOG CONNECTIONS (INCLUDE SLOW OPERATIONS AND ERRORS).....	4-1
4.2.2.	LOG CONNECTIONS (WITHOUT SLOW OPERATIONS AND ERRORS).....	4-2
4.3.	Log Old/New Data When Table is Changed	4-3
4.4.	Log SQL and Input Parameters	4-4
4.5.	Lock Information	4-6
5.	Appendix	5-1
5.1.	SQL Script	5-1

1. Introduction

Welcome to DBMaker New Log System user's guide. In this guide, we will introduce basic information of new log system, setting dmconfig.ini keywords, output log files, and examples to help user to understand and use new log system.

1.1. What is New Log System

New log system is replacing Audit Trail System. Before 5.0 versions, DBMaker has the audit trail system at server side. But it does not record enough information and the format is not easy for us to understand what's going on in the user's environment. Because it is quite hard to maintain or enhance the audit trail system, we will provide a new log system for recording more information and more readable format. The old audit trail feature has been removed from 5.0.

The log system is used for recording the system information for user's operation. The purpose for the log system is to provide a way for DBA or support to understand what operation executed before or when the problem occurs.

1.2. What information will be logged

This information could be used to record more information in the database for DBA or support to check what's going on in the user's environment.

Such as what operation are the database doing, the execution time of a SQL command, what SQL command are doing at user's runtime environment when problem occurs, service function, connection id, user name, statement id, login information, error argument, SYSUSER and SYSINFO,

SYSTEM memory information if the memory could be detected, execution plan, input parameter's value in SQL command, extra locks information when lock time out or deadlock, etc.

1.3. New Log System Constraints

- When the log turns on, the server performance will be affected. Especially when the log level is high.
- User must make sure there is enough HD to store the server log. Or the log information will be lost after the HD is full.
- It's possible that some runtime problem will not be reproduced when the log has turn on because of the timing has been changed.
- Because this is a sever log, client or network error will not be recorded.

2. Usage of New Log System

2.1. General Description

DBMaker supplies two methods to turn new log function on.

One is setting keywords in dmconfig.ini file but should set before start database. User should set LGSVR first, then others log setting can work.

Another is calling SETSYSTEMOPTION() stored procedure, it can help user runtime start log system. The SP can set all functions of new log system.

2.2. Keywords

These keywords are used in server side and must be set before start DB.

DB_LGSVR is primary keyword for new log system. Other keywords are used to assistant setting.

2.2.1. DB_LGSVR

This keyword is used to turn on/off the server log, and specifies the log level. The default value is 0.

0 => do not log at all

1 => Log error, default error level is set in DB_LGERR

2 => Log slow operation, default time is set in DB_LGSTM

3 => Log error and slow operation

4 => Log connect, disconnect, commit rollback, SQL command, error and slow operation, etc.

For whether to log SQL command or not is set in DB_LGSQL.

5 => Log all service functions include set connection option, commit, rollback operation when Function call exit to server.

6 => Log all operation when all service function call enter/exit server.

2.2.2. DB_LGDAY

This keyword specifies the number of days to keep the log files available. The expired log files would be removed by service daemon which is set by DB_STSVR. The default value is 30.

0 => filenames of log system do not include date.

1~365 => the setting of DB_LGFNO will be ignored.

2.2.3. DB_LGDIR

The keyword specifies the server log's directory where output log files reside. The default is 'Database Directory\lgdir' in windows platform and 'Database Directory/lgdir' in Linux platform.

2.2.4. DB_LGERR

The keyword is to set logged error level when only need to log error when LGSVR is 1-4. The default value is 3.

0 => Log when there's core dumped or DB crash error (rc >30000)

1 => Log when there's disconnect error or DB crash error (rc >20000)

2 => Log when there's abort, disconnect or DB crash error (rc >10000)

3 => Log when there's normal, abort, disconnect or DB crash error (rc >100)

4 => Log when there's warning or any error (rc > 0)

2.2.5. DB_LGSTM

The keyword is to set the time for logging when only need to log slow statement when LGSVR is 2, 3 or 4. Unit is second. The default value is 5.

When DB_LGSTM = 6, it means when execution time of a SQL command is longer than 6 seconds, DBMaker will see the SQL command as a slow statement, so the SQL command will be logged when user need to log slow statement. DBMaker will log all slow statements.

2.2.6. DB_LGPLN

The keyword is to set whether to log execution plan for select, update or delete statement. The default value is 0.

0 => do not log execution plan

1 => log execution plan

2.2.7. DB_LGSYS

The keyword is to set what information will be logged when LGSVR is on. The default value is 0. If the LGSYS is turned on, more comma separated value will be logged as in the log file.

0 => Log execution time, rc, service function, connection_id, user_name, statement id, login information, error argument, sql statement.

1 => Log all the information as LGSYS = 0 and log the SYSUSER and SYSINFO information.

2 => Log all the information as LGSYS = 1 and log the SYSTEM memory information if the memory could be detected.

2.2.8. DB_LGZIP

This keyword is used to packing/zipping the need of closed log files would be necessary in order to save some storage. The default value is 0.

0 => packing or zipping the need of closed log files.

1 => not packing or zipping the need of closed log files.

2.2.9. DB_LGFSZ

This keyword is to set the file size of the log. The unit is MB. The default value is 100. The valid range is 10~1500.

2.2.10. DB_LGFNO

This keyword is to set the number of file for log. The default value is 20. The valid range is 2~255. When DB_LGFNO = 10 and having generated 10 log files, then the next log file will cover with the 1st log file.

2.2.11. DB_LGSQL

This keyword is to set whether to log SQL command when LGSVR = 4. The default value is 2.

0 => do not log SQL command when LGSVR = 4.

1 => log non-select SQL command when LGSVR = 4.

2 => log all SQL command when LGSVR = 4.

2.2.12. DB_LGPAR

This keyword is to set whether to log input parameter's value. The default value is 0.

0 => do not log input parameter's value.

1 => log input parameter's value.

2 => log input parameter's value and stored procedure's executed SQL command.

3 => log trigger's SQL statement and parameter's value.

4 => log both sp and trigger's SQL statement and parameter's value.

☞Example:

```
insert into sample.employee values(?,?,?);
```

When DB_LGPAR = 0, DBMaker only records the SQL command with question mark, but we do not know input value.

When DB_LGPAR = 1 and log the above SQL command with question mark, DBMaker will use another log file to store input parameter's value.

2.2.13. DB_LGLCK

This keyword is to set whether to log extra lock information when lock time out or deadlock. The default value is 0.

0 => do not log extra lock information when error code is lock time out or deadlock

1 => log extra locks time out information.

2.3. SETSYSTEMOPTION Stored Procedure

Usage of this stored procedure:

```
SETSYSTEMOPTION(VARCHAR(32) option_name INPUT, VARCHAR(256)  
option_value INPUT)
```

option_name: the name of the system option

option_value: the value of the system option

NOTE: Using the stored procedure SETSYSTEMOPTION to set new log, the setting is only available in current session.

☞Example:

```
dmSQL> call SETSYSTEMOPTION('LGSRV', '3');
dmSQL> call SETSYSTEMOPTION('LGERR', '2');
dmSQL> call SETSYSTEMOPTION('LGSTM', '10');
```

2.4.ADD TRACE command

The ADD TRACE command adds trace on a table to log the OLD/NEW data. Actually, it is implemented by 3 internal triggers for insert/update/delete operation, which operation on the traced table would be logged, and the data would be printed in DBNAME_currentdate_###.TXT as extra information. Only users with table owner, DBA, SYSDBA or SYSADM security privileges can execute the ADD TRACE command.

NOTE: DB_LgSvr need to be equal or greater than 4. Otherwise, the detail information would be skipped and nothing would be written to log files.

The syntax of the ADD TRACE command is

```
ADD TRACE ON table_name;
```

☞Example:

Add trace on table tb1, and do insert/update/delete.

```
dmSQL> ADD TRACE ON tb1;
dmSQL> INSERT INTO tb1 VALUES (1, 'abc');
1 rows inserted
dmSQL> UPDATE tb1 SET c2 = 'xyz' WHERE c1=1;
1 rows updated
dmSQL> DELETE FROM tb1;
1 rows deleted
```

The output txt file will be like follow:

```
INFO_01364_1 -----
      [TRACE ON SYSADM.TB1 - INSERT]
      [NEW.C#1] 1
      [NEW.C#2] 'abc '
INFO_01364_2 -----
      [TRACE ON SYSADM.TB1 - UPDATE]
      [OLD.C#1] 1
      [OLD.C#2] 'abc '
```



```
[NEW.C#1] 1
[NEW.C#2] 'xyz '
INFO_01364_3 -----
[TRACE ON SYSADM.TB1 - DELETE]
[OLD.C#1] 1
[OLD.C#2] 'xyz '
```

2.5. Output Log Files

When the log server has been turn on, server's operation according to log option will be record and store the log in the LGDIR. The database server will assign the log name according to DBNAME and the log index number. When first start the log, some system information will be logged at **DBNAME.LOG**. When the log file size reaches default 100 MB byte or the size specified in LGFSZ, the log information will be logged to next log file. Ex: DBSAMPLE5_date_1.LOG, DBSAMPLE5_date_2.LOG ...DBSAMPLE5_date_n.LOG. The log file will be written to next log file until default 20 file or the number specified in the LGFNO, and then rewrite the log from the DBNAME_date_1.LOG.

If there are other additional information to be logged when user turn on LGPLN, LGPAR, LGLCK or there's information in ERROR.LOG, it will be logged at **DBNAME_date_n.TXT**. The extra information's id will be recorded as INFO_connectionid_n at the OTHER_INFO field in the DBNAME_date_n.LOG and a title in DBNAME_date_n.TXT. So user can trace the additional information by this id.

The log format will be text csv format, so user can use excel viewer to check it after rename the .LOG to .csv file.

☞Example:

For the example in [Chapter 3.4. "DB_LGPAR"](#), the extra information's id is '**INFO_21544_1**'.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG TIME,BEG TIME,STATE,RETCODE,EXE TIME,SV FUNC,CONNECT ID,USERNAME,LOGIN TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD
0000004, "2024/10/29 13:08:43","2024/10/29 13:08:43","O", 63,0.003923,"EXECUTE
",21544,"SYSADM","2024/10/29 13:08:40","127.0.0.1",0,1,, "INFO_21544_1", "insert into
test(id,sqlName) values(?,?) "
```

NEWLOGSYSTEM_date_1.TXT:

```
INFO 21544 1 -----
[Parameter]: 4, 'abcdefghijkl'
```

2.6. Output Log Information

The following table lists all of the columns in log file, and a brief description of what is contained in each column.

COLUMN NAME	DESCRIPTION
SERNO	The number of the log
LOG_TIME	The time when this log is written(end of execution)
BEG_TIME	The time when this log begins(begin of execution)
STATE	There are four states: _, O, X, S, according to " _, O, X, S" to judge it's unknown, ok, error, or slow, the check sequence is check rc first, and then check execution time.
RETCODE	Returned code: 0 or error code
EXE_TIME	Execution time
SV_FUNC	Execute which server function at present
CONNECT_ID	Connection ID
USERNAME	User's name
LOGIN_TIME	Login time of the user
LOGIN_ADDR	Login IP address of the user
STMT_ID	The id of the statement in this log, starts from 0; -1 means this log doesn't include a statement handle
NUM_STMT	Number of the allocated statement
ERROR_ARG	Error argument
OTHER_INFO	If turn on other LGXXX setting (ex: LgPln, LgPar), these information will be recorded in LOGNAME.TXT, user can mark [INFO_XXX] to .TXT to check
SQL_CMD	The most recently executed SQL command

3. Samples

This chapter will show users the output files with different keywords setting, most of the samples refer to [5.1. "SQL Script"](#).

Create test database NewLogSystem with the following setting:

```
[NewLogSystem]

DB_DBDIR = D:\NEWLOGSYSTEM

DB_FODIR = D:\NEWLOGSYSTEM\FO

DB_PTNUM = 2455

DB_SVADR = 127.0.0.1

DB_SPLOG = D:\NEWLOGSYSTEM\SP

DB_LBDIR = D:\NEWLOGSYSTEM\UDF

DB_USRFO = 1
```

3.1. DB_LGSR

3.1.1. DB_LGSR=1

3.1.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSR=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSR', '1');
```

3.1.1.2. Function

Log error, default error level is set in DB_LGERR; This sample is to log error code > 100. (Because the DB_LGERR default is 3).

3.1.1.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.1.1.4. Check Output Log Files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM.LOG:

```
Database [NEWLOGSYSTEM] begin trace at 2024/10/29 12:22:36

dbmaker-WIN64: 5.4.7 (#31685, 20240912)

CREATE DB TIME:                2024/10/29 12:21:48

START_DB_TIME:                2024/10/29 12:22:36

START LOG FILE:                NEWLOGSYSTEM 20241029 1.LOG

LCODE:                        10

TDEMD:                        58

FILE_VERSION:                 5.41

NUM PAGE BUF:                 2000

DB_PAGE_SIZE:                 8192

DB_SCA_SIZE:                   1926

NUM JOURNAL BUFFER:           64

DB_SYSCB_SIZE:                1425408

NUM JOURNAL FILE:             1

NUM_JOURNAL_BLOCKS:           16000

NUM MAX HARD CONNECT:         4840

NUM_MAX_SOFT_CONNECT:         310

DDB MODE:                     OFF

BACKUP_MODE:                   NON-BACKUP

USER FO MODE:                 ON

READ_ONLY_MODE:               OFF

FRAME SIZE:                   32768

MAX_ITT_SIZE:                 5306712064

TOTAL_MEMORY:                 17057034240

TOTAL FREE MEMORY:            8835670016

TOTAL_SWAP_MEMORY:            19607171072

TOTAL_FREE_SWAP_MEMORY:       9001287680
```

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG TIME,BEG TIME,STATE,RETCODE,EXE TIME,SV_FUNC,CONNECT ID,USERNAME,LOGIN TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/29 12:23:52","2024/10/29 12:23:52","X", 6018,0.000559,"PREPARE ",
7844,"SYSADM","2024/10/29 12:23:52","127.0.0.1",0,1,": INT [lparser 332], 0, 0,
0",,"create table int(a int)"

0000002, "2024/10/29 12:23:52","2024/10/29 12:23:52","X",20916,0.000522,"CONNECT ",
-1,"","1970/01/01 08:00:00","", -1,0," 5 users connecting now (1) [dbconn.c 356], 5,
5, 20916",,"connect db"

0000003, "2024/10/29 12:23:52","2024/10/29 12:23:52","X", 6521,0.000518,"PREPARE ",
7844,"SYSADM","2024/10/29 12:23:52","127.0.0.1",0,1,": SYSADM.T1 [bitable. 215], 0, 0,
```

```
0",,, "select * from t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1) and a.c2='ABC'"
```

3.1.1.5. Analyze Output Log files

When set DB_LGSRV = 1 to log errors, new log system will refer to DB_LGSYS (the default is 0) and DB_LGERR (the default is 3) to decide log information.

So it only logs errors (rc > 100), error 6018 and error 20916 and error 6521.

3.1.2. DB_LGSRV=2

3.1.2.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=2
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '2');
```

3.1.2.2. Function

This sample is to log slow query when execution time > 5 second.

3.1.2.3. Run SQL Script

Please refer to attached file "slow_sql_data.sql" than run chapter [5.1. "SQL Script"](#).

3.1.2.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN ADDR,STMT ID,NUM STMT,ERROR ARG,OTHER INFO,SQL CMD

0000001, "2024/10/29 12:31:36","2024/10/29 12:31:33","S", 41,5.505168,"EXECUTE
",13252,"SYSADM","2024/10/29 12:28:56","127.0.0.1",0,1,, "select * from t1 a left
outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1) and
a.c2='ABC'"
```

3.1.2.5. Analyze Output Log files

When set DB_LGSRV = 2 only to log slow operation, new log system will refer to DB_LGSYS (the default is 0) and DB_LGERR (the default is 3) and DB_LGSTM (the default is 5 second) to decide log information.

Because execution time of the SQL command is 5.5 second (> 5 second), DBMaker sees the SQL command as a slow operation. In this SQL script, there is only one slow command, so new log system logs the slow command.

3.1.3. DB_LGSRV=3

3.1.3.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=3
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '3');
```

3.1.3.2. Function

This sample is to log error code > 100 and slow query when execution time > 5 second.

3.1.3.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.1.3.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/29 12:34:23","2024/10/29 12:34:23","X", 6018,0.000000,"PREPARE
",21632,"SYSADM","2024/10/29 12:34:10","127.0.0.1",0,1,"": INT [lparser 332], 0, 0,
0",,"create table int(a int)"

0000002, "2024/10/29 12:34:23","2024/10/29 12:34:23","X",20916,0.000000,"CONNECT ",
-1,"", "1970/01/01 08:00:00","", -1,0,"", 5 users connecting now (1) [dbconn.c 356], 5,
5, 20916",,"connect db"

0000003, "2024/10/29 12:34:27","2024/10/29 12:34:23","S", 41, 5.475131,"EXECUTE
",21632,"SYSADM","2024/10/29 12:34:10","127.0.0.1",0,1,,,"select * from t1 a left
outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1) and
a.c2='ABC'"
```

3.1.3.5. Analyze Output Log files

When set DB_LGSRV = 3, new log system not only logs errors, but also logs slow operation. So DB_LGSRV=3 is combination of DB_LGSRV=1 and DB_LGSRV=2.

So it logs error 6018, error 20916, and the slow SQL command.

3.1.4. DB_LGSRV=4

3.1.4.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=4
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '4');
```

3.1.4.2. Function

This sample is to log connect, disconnect, commit, rollback, any operation that is related to a sql statement, error code > 100 and slow query when execution time > 5 second.

3.1.4.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.1.4.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG(part):

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/29 12:36:40","2024/10/29 12:36:40","O",    0,0.266840,"CONNECT ",
17512,"SYSADM","2024/10/29 12:36:40","local",-1,0,,,"start db"

.....

0000009, "2024/10/29 12:36:59","2024/10/29 12:36:59","X",  6018,0.000000,"PREPARE ",
15252,"SYSADM","2024/10/29 12:36:47","127.0.0.1",0,1,": INT [lparser 332], 0, 0,
0",,"create table int(a int)"

0000010, "2024/10/29 12:36:59","2024/10/29 12:36:59","O",    0,0.000014,"PREPARE ",
15252,"SYSADM","2024/10/29 12:36:47","127.0.0.1",0,1,,,"create table test(num
serial(1),id int,sqlName char(10))"

.....

0000015, "2024/10/29 12:36:59","2024/10/29 12:36:59","O",   63,0.000568,"EXECUTE ",
15252,"SYSADM","2024/10/29 12:36:47","127.0.0.1",0,1,,,"insert into test(id,sqlName)
values(2,'abcdefghijklk')"
```

Please refer to the attached file: "NEWLOGSYSTEM_1(DB_LGSVR=4).LOG" for the complete log.

3.1.4.5. Analyze Output Log files

When set DB_LGSVR = 4, new log system will not only log information when DB_LGSVR = 3, but also log other information, and refer to DB_LGSQL (the default is 2).

So it logs all SQL commands, errors, slow operation, etc.

3.1.5. DB_LGSVR=5

3.1.5.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSVR=5
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started

```
Call SETSYSTEMOPTION('LGSVR', '5');
```

3.1.5.2. Function

This sample is to log all server function of calling server when enter server.

3.1.5.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.1.5.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG(part):

```
0000006, "2024/10/29 12:43:38","2024/10/29 12:43:38","O",    0, 0.000000, "GETCNOPT",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",-1,1,,,"option=108, val=0"

0000007, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    0,0.000000,"PREPARE ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",0,1,,,"grant connect to user1"

0000008, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    0,0.000564,"EXECUTE ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",0,1,,,"grant connect to user1"

0000009, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    76,0.000000,"EXECUTE ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",0,1,,,"grant connect to user1"

0000010, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    0,0.000000,"GET_ERR ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",-1,1,,,"

0000011, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    0,0.000000,"GET_ERR ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",-1,1,,,"

0000012, "2024/10/29 12:44:42","2024/10/29 12:44:42","O",    0,0.000000,"GET_ERR ",
16064,"SYSADM","2024/10/29 12:43:38","127.0.0.1",-1,1,,,"
```

Please refer to the attached file "NEWLOGSYSTEM_1(DB_LGSRV=5).LOG" for the complete log.

3.1.5.5. Analyze output Log files

Compare with DB_LGSRV=4, DB_LGSRV=5 will log information about exit operations. These logs are inner or outer functions, and hard for users to analyze.

3.1.6. DB_LGSRV=6

3.1.6.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=6
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '6');
```

3.1.6.2. Function

This sample is to log all server function of calling server when enter/exit server.

3.1.6.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.1.6.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG(part):

```
00000006, "2024/10/29 12:48:16","2024/10/29 12:48:16","O",    0, 0.005022, "GETCNOPT",
21320,"SYSADM","2024/10/29 12:48:16","127.0.0.1",-1,0,,, "option=28, val=0"

00000007, "2024/10/29 12:48:16","2024/10/29 12:48:16","_",,,, "ALOCCMD ", 21320,
"SYSADM","2024/10/29 12:48:16","127.0.0.1",-1,0,,,

00000008, "2024/10/29 12:48:16","2024/10/29 12:48:16","O",    0,0.006081,"ALOCCMD ",
21320,"SYSADM","2024/10/29 12:48:16","127.0.0.1",-1,1,,,

00000009, "2024/10/29 12:48:16","2024/10/29 12:48:16","_", , , "GETCNOPT", 21320,
"SYSADM","2024/10/29 12:48:16","127.0.0.1",-1,1,,, "option=108, val=0"

00000010, "2024/10/29 12:48:16","2024/10/29 12:48:16","O",    0, 0.005623, "GETCNOPT",
21320,"SYSADM","2024/10/29 12:48:16","127.0.0.1",-1,1,,, "option=108, val=0"

00000011, "2024/10/29 12:48:19","2024/10/29 12:48:19","_",,,, "PREPARE ", 21320,
"SYSADM","2024/10/29 12:48:16","127.0.0.1",0,1,,, "grant connect to user1"

00000012, "2024/10/29 12:48:19","2024/10/29 12:48:19","O",    0,0.006175,"PREPARE ",
21320,"SYSADM","2024/10/29 12:48:16","127.0.0.1",0,1,,, "grant connect to user1"
```

Please refer to the attached file: "NEWLOGSYSTEM_1(DB_LGSRV=6).LOG" for the complete log.

3.1.6.5. Analyze output Log files

Compare with DB_LGSRV=4, DB_LGSRV=6 will log information about enter and exit operations. These logs are inner or outer functions, and hard for users to analyze.

3.2. DB_LGSYS

3.2.1. DB_LGSYS=1

3.2.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=1
DB_LGSYS=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '1');
Call SETSYSTEMOPTION('LGSYS', '1');
```

3.2.1.2. Function

This sample is to log error code > 100. And it logs more information than as LGSYS = 0(default value), it also logs the SYSUSER and SYSINFO information. You can find the first line has more columns in log file.

3.2.1.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.2.1.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,NUM_SCAN,NUM_INSERT,NUM_UPDATE,NUM_DELETE,NUM_TRANX,NUM
IDX_SPLIT,NUM_PAGE_COMPRESS,NUM_PHYSICAL_READ,NUM_PHYSICAL_WRITE,NUM_LOGICAL_READ,NUM
LOGICAL_WRITE,READ_HIT_RATIO,READ_WRITE_RATIO,NUM_JNL_BLK_READ,NUM_JNL_BLK_WRITE,NUM
_JNL_REC_WRITE,NUM_JNL_FRC_WRITE,NUM_JNR_BLOCK_FREE,CURRENT_JOURNAL_FN,CURRENT_JOURNA
L_BN,JOURNAL_FLUSH_RATE,NUM_STARTED_TRANX,NUM_COMMITTED_TRANX,NUM_ABORTED_TRANX,NUM_CH
ECKPOINT,NUM_COMMIT_WAITER,NUM_ROW_LOCK_UPG,NUM_PAGE_LOCK_UPG,NUM_LOCK_TIMEOUT,NUM_LO
CK_WAIT,NUM_LOCK_REQUEST,NUM_DEADLOCK,NUM_CONNECT,NUM_PEAK_CONNECT,NUM_SQL_SELECT,NUM
SQL_INSERT,NUM_SQL_UPDATE,NUM_SQL_DELETE,NUM_SQL_PREPARE,NUM_SQL_EXECUTE,NUM_SQL_EXE
DIRECT,NUM_SQL_FETCH,SYSINFO_RESET_TIME,FREE_DCCA_SIZE,CURRENT_ITT_SIZE,ERROR_ARG,OTH
ER_INFO,SQL_CMD

0000001, "2024/10/29 12:51:44","2024/10/29 12:51:44","X", 6018,0.000505,"PREPARE ",
9636,"SYSADM","2024/10/29 12:51:42", "127.0.0.1", 0, 1, 17, 2, 0, 3, 7, 0, 0, 15, 0,
74,8,"79.73%","79.73%",0,6,10,2,15737,0,9658," 0.00", 8, 7, 1, 0, 0, 0, 0, 0, 206,
0,4,4,0,0,0,3,3,0,0,"2024/10/29 12:51:37",12260432,0,": INT [lxpaser 332], 0, 0,
0", , "create table int(a int)"

0000002, "2024/10/29 12:51:45","2024/10/29 12:51:45","X",20916,0.000000,"CONNECT ",
-1, "", "1970/01/01 08:00:00","", -1, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 190, 29, "88.42%",
"88.42%",0,27,42,8,15716,0,9679," 0.00", 30, 24, 1, 0, 0, 0, 0, 0, 339, 0, 9, 9, 1,
5,0,0,8,10,0,0,"2024/10/29 12:51:37",12258896,0," 5 users connecting now (1)
[dbconn.c 356], 5, 5, 20916",,"connect db"
```

3.2.1.5. Analyze output Log files

The **Red** part means the differences compare with LGSYS=0, LGSYS=1 will log SYSUSER and SYSINFO information.

3.3. DB_LGERR

3.3.1. DB_LGERR=1

3.3.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=1
DB_LGERR=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '1');
Call SETSYSTEMOPTION('LGERR', '1');
```

3.3.1.2. Function

This sample is to log the error code > 20000.

3.3.1.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.3.1.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG TIME,BEG TIME,STATE,RETCODE,EXE TIME,SV FUNC,CONNECT ID,USERNAME,LOGIN TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/29 12:54:06","2024/10/29 12:54:06","X",20916,0.000000,"CONNECT ",
-1,"","1970/01/01 08:00:00","", -1,0," 5 users connecting now (1) [dbconn.c 356], 5,
5, 20916",,"connect db"
```

3.3.1.5. Analyze Output Log files

When setting DB_LGSRV=1 and DB_LGERR=1, new log system only logs errors (rc >20000).so it logs error 20916.

3.3.2. DB_LGERR=4

3.3.2.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=1
DB_LGERR=4
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '1');
Call SETSYSTEMOPTION('LGERR', '4');
```

3.3.2.2. Function

This sample is to log the error code > 0.

3.3.2.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.3.2.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG and NEWLOGSYSTEM_date_1.LOG will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG TIME,BEG TIME,STATE,RETCODE,EXE TIME,SV FUNC,CONNECT ID,USERNAME,LOGIN TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD
```

```
0000001, "2024/10/29 13:02:14","2024/10/29 13:02:14","O", 76,0.000000,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"grant connect to user1"

0000002, "2024/10/29 13:02:14","2024/10/29 13:02:14","X", 6018,0.000525,"PREPARE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,": INT [lxpaser 332], 0, 0,
0",,"create table int(a int)"

0000003, "2024/10/29 13:02:14","2024/10/29 13:02:14","O", 63,0.000000,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"insert into test(id,sqlName)
values(2,'abcdefghijk')"

0000004, "2024/10/29 13:02:14","2024/10/29 13:02:14","O", 63,0.000503,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"insert into test(id,sqlName)
values(?,?)"

0000005, "2024/10/29 13:02:14","2024/10/29 13:02:14","O", 41,0.000000,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"select * from test"

0000006, "2024/10/29 13:02:14","2024/10/29 13:02:14","X",20916,0.000540,"CONNECT ",
-1,"","1970/01/01 08:00:00","",-1,0," 5 users connecting now (1) [dbconn.c 356], 5,
5, 20916",,"connect db"

0000007, "2024/10/29 13:02:17","2024/10/29 13:02:14","O", 41,5.150399,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"select * from t1 a left
outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1) and
a.c2='ABC'"

0000008, "2024/10/29 13:02:17","2024/10/29 13:02:17","O", 0,0.000000,"EXECUTE
",16380,"SYSADM","2024/10/29 13:02:08","127.0.0.1",0,1,,,"rollback to point1"
```

3.3.2.5. Analyze Output Log files

When setting DB_LGSRV=1 and DB_LGERR=4, new log system logs warning or any error (rc > 0).so it logs warning 63, warning 76, error 6018 and error 20916.

3.4. DB_LGPAR

3.4.1. DB_LGPAR=1

3.4.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=1
DB_LGERR=4
DB_LGPAR=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '1');
Call SETSYSTEMOPTION('LGERR', '4');
Call SETSYSTEMOPTION('LGPAR', '1');
```

3.4.1.2. Function

This sample is to log the error code > 0 and input parameter's value.

3.4.1.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.4.1.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG, NEWLOGSYSTEM_date_1.LOG and NEWLOGSYSTEM_date_1.TXT will be generated.

NEWLOGSYSTEM_date_1.LOG: content same with [chapter 3.3.2.](#)

```
...
00000004, "2024/10/29 13:08:43", "2024/10/29 13:08:43", "O", 63, 0.003923, "EXECUTE
", 21544, "SYSADM", "2024/10/29 13:08:40", "127.0.0.1", 0, 1, "INFO_21544_1", "insert into
test(id,sqlName) values(?,?) "
...
```

NEWLOGSYSTEM_date_1.TXT:

```
INFO_21544_1 -----
[Parameter]: 4, 'abcdefghijk'
```

3.4.1.5. Analyze Output Log files

The error command (insert into test(id,sqlName) values(?,?)) has question mark, we only know the error command, but do not know detailed values. So it generates a new **NEWLOGSYSTEM_date_1.TXT** file to store input parameter's value.

INFO_21544_1: to search **NEWLOGSYSTEM_date_1.LOG** to locate the related command.

[Parameter]: symbol parameter's value.

3.5. DB_LGPLN

3.5.1. DB_LGPLN=1

3.5.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=2
DB_LGPLN=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '2');
Call SETSYSTEMOPTION('LGPAR', '1');
```

3.5.1.2. Function

This sample is to log slow query when execution time > 5 second and execution plan of slow query.

3.5.1.3. Run SQL Script

Please refer to chapter [5.1. "SQL Script"](#).

3.5.1.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG, NEWLOGSYSTEM_date_1.LOG and NEWLOGSYSTEM_date_1.TXT will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME,
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/29 13:13:53","2024/10/29 13:13:49","S", 41,5.500195,"EXECUTE
",22144,"SYSADM","2024/10/29 13:13:46","127.0.0.1",0,1,,"INFO_22144_1","select * from
t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and
(a.c1=d.c1) and a.c2='ABC'"
```

NEWLOGSYSTEM_date_1.TXT:

```
INFO_22144_1 -----

[Dump Plan]:

{ON Block 0}

ON Type      : JOIN

[PL Block 0]

Method       : Outer Join

Type        : Nested Join

Factors      : (1) T1.C1 = T2.C1
              (2) T1.C1 = T3.C1
              (3) T1.C1 = T4.C1

I/O Cost     : 0.0

CPU Cost     : 0.0

Sub Cost     : 0.0

Result Rows  : 0.0

Sub Block 1: [PL Block 1]

Sub Block 2: [PL Block 2]

[PL Block 1]

Method       : Scan

Table Name   : T1

Type        : Table Scan

Factors      : (4) T1.C2 = 'ABC'

- Evaluate   : 4

I/O Cost     : 5.0

CPU Cost     : 12.6
```

```
Sub Cost      : 0.0

Result Rows   : 10.0

[PL Block 2]

Method        : Join
Type          : Nested Join
Factors       : <none>
I/O Cost      : 5005005.0
CPU Cost      : 12650150.0
Sub Cost      : 0.0
Result Rows   : 1000000000.0
Sub Block 1: [PL Block 3]
Sub Block 2: [PL Block 4]

[PL Block 3]

Method        : Join
Type          : Nested Join
Factors       : <none>
I/O Cost      : 5005.0
CPU Cost      : 12650.1
Sub Cost      : 0.0
Result Rows   : 1000000.0
Sub Block 1: [PL Block 5]
Sub Block 2: [PL Block 6]

[PL Block 5]

Method        : Scan
Table Name    : T2
Type          : Table Scan
Factors       : <none>
I/O Cost      : 5.0
CPU Cost      : 12.6
Sub Cost      : 0.0
Result Rows   : 1000.0

[PL Block 6]
```

```
Method      : Scan
Table Name  : T3
Type        : Table Scan
Factors     : <none>
I/O Cost    : 5.0
CPU Cost    : 12.6
Sub Cost    : 0.0
Result Rows : 1000.0
```

[PL Block 4]

```
Method      : Scan
Table Name  : T4
Type        : Table Scan
Factors     : <none>
I/O Cost    : 5.0
CPU Cost    : 12.6
Sub Cost    : 0.0
Result Rows : 1000.0
```

3.6. DB_LGLCK

3.6.1. DB_LGLCK=1

3.6.1.1. Usage

Set in dmconfig.ini before database is started:

```
DB_LGSRV=1
DB_LGLCK=1
```

Execute SETSYSTEMOPTION by dmsql32 or dmsqlc after database is started:

```
Call SETSYSTEMOPTION('LGSRV', '1');
Call SETSYSTEMOPTION('LGLCK', '1');
```

3.6.1.2. Function

This sample is to log error code > 100 and extra lock time out information.

3.6.1.3. Run SQL Script

```
connect to newlogsystem sysadm;
drop table test_lock;
create table test_lock(num int,sqlName char(10));
```



```
set autocommit off;

insert into test_lock values(1,'insert');

USE 2;

connect to newlogsystem sysadm;

update test_lock set sqlName = 'update';

disconnect;

use 1;
```

3.6.1.4. Check Output Log files

In 'D:\NEWLOGSYSTEM\lgdir' directory: NEWLOGSYSTEM.LOG, NEWLOGSYSTEM_date_1.LOG and NEWLOGSYSTEM_date_1.TXT will be generated.

NEWLOGSYSTEM_date_1.LOG:

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT ID,NUM STMT,ERROR ARG,OTHER INFO,SQL CMD

0000001, "2024/10/29 13:17:04","2024/10/29 13:17:04","X", 6521,0.000000,"EXECUTE
",10968,"SYSADM","2024/10/29 13:17:04","127.0.0.1",0,1,": TEST LOCK [exddl.c 684], 0,
0, 0",,"drop table test_lock"

0000002, "2024/10/29 13:17:09","2024/10/29 13:17:04","X", 1102,5.016855,"EXECUTE
",16556,"SYSADM","2024/10/29 13:17:04","127.0.0.1",0,1,": Object ID :
2.10.0 ,Connection ID : 10968, LKMode : X .NONE, LKStatus : GRANTED, Command :
[EXIT] insert into test_lock values(1,'insert') [unknown 1123], 501507, 0,
0","INFO_16556_1","update test_lock set sqlName = 'update'"
```

NEWLOGSYSTEM_date_1.TXT:

```
INFO_16556_1 -----

[ERR LK TIMEOUT]: : Object ID : 2.10.0 ,Connection ID : 10968, LKMode : X .NONE,
LKStatus : GRANTED, Command : [EXIT] insert into test_lock values(1,'insert')
```

4. Scenario

This chapter will show users some common situations, these situations are often used in customer's database.

4.1. Slow Operations

If user's sql statement execution is slow or delayed. Users can add the following keywords to database setting in dmconfig.ini. The database with these keywords will log the SQL statement if it's execution time is over 5 seconds.

```
DB_LGSVR=2      //log slow operations
DB_LGSTM=5      //defined the sql statement as slow operation if execution time>5
```

Execute the test sql,

```
connect to DBSAMPLE5 sysadm;

//execute slow sql statement

select * from t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 =
c.c1) and (a.c1=d.c1) and a.c2='ABC';
```

NOTE: Depends on the computer efficiency and DBMaker version, the execution time may be different, users may try their own sql statements.

The output log will be like follows, notice the red part: "S" means this log is a slow operation, because execution time is 5.231787 second

```
0000018, "2024/10/25 16:45:39","2024/10/25 16:45:35","S", 41, 5.231787, "EXECUTE
",15152,"SYSADM","2024/10/25 16:30:09","127.0.0.1",0,1,,,"select * from t1 a left
outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1) and
a.c2='ACC'"
```

4.2. Log Connections

Most of our customers have the requirement to observe users' connections, so the following example will show users how to log users' connections.

4.2.1. LOG CONNECTIONS (INCLUDE SLOW OPERATIONS AND ERRORS)

Add DB_LGSVR=4 can log connect/disconnect information, but include slow operations and errors.

```
DB_LGSVR=4      //log connect, disconnect, error message and slow operations
DB_LGSTM=5      //see as slow operation if execution time>5 second
```

Execute the test sql:

```
connect to DBSAMPLE5 sysadm;

//execute slow sql statement

select * from t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 =
c.c1) and (a.c1=d.c1) and a.c2='ABC';

//execute error sql statement

create table t1(c1 int);

disconnect;
```

The output log will be like follows, SERNO 0000005 and 0000008 respectively represents slow operation and error log.

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/28 15:17:12","2024/10/28 15:17:11","O", 0, 0.527990, "CONNECT ",
9064,"SYSADM","2024/10/28 15:17:11","local",-1,0,,, "start db"

0000002, "2024/10/28 15:17:12","2024/10/28 15:17:12","O", 0, 0.012222, "DISCONN ",
9064,"SYSADM","2024/10/28 15:17:11", "local", -1, 0, , , "disconnect"

0000003, "2024/10/28 15:17:24","2024/10/28 15:17:24","O", 0, 0.005595, "CONNECT ",
4472,"SYSADM","2024/10/28 15:17:24", "127.0.0.1", -1, 0, , , "connect db"

0000004, "2024/10/28 15:17:31","2024/10/28 15:17:31","O", 0, 0.002055, "PREPARE ",
4472, "SYSADM", "2024/10/28 15:17:24", "127.0.0.1", 0, 1, , , "select * from t1 a
left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1)
and a.c2='ABC'"

0000005, "2024/10/28 15:17:34","2024/10/28 15:17:31","S", 41, 5.182557, "EXECUTE ",
4472, "SYSADM", "2024/10/28 15:17:24", "127.0.0.1", 0, 1, , , "select * from t1 a
left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 = c.c1) and (a.c1=d.c1)
and a.c2='ABC'"

0000006, "2024/10/28 15:17:34","2024/10/28 15:17:34","O", 0, 0.000000, "COMMIT ",
4472, "SYSADM", "2024/10/28 15:17:24", "127.0.0.1", -1, 1, , ,

0000007, "2024/10/28 15:18:14","2024/10/28 15:18:14","O", 0, 0.000000, "PREPARE ",
4472, "SYSADM", "2024/10/28 15:17:24", "127.0.0.1", 0, 1, , , "create table t1(c1
int)"

0000008, "2024/10/28 15:18:14","2024/10/28 15:18:14","X", 6520, 0.000537, "EXECUTE ",
4472,"SYSADM","2024/10/28 15:17:24", "127.0.0.1", 0, 1, ":", T1 [exddl1.c 621], 0, 0,
0", , "create table t1(c1 int)"

0000009, "2024/10/28 15:18:21","2024/10/28 15:18:21","O", 0, 0.004874, "DISCONN ",
4472,"SYSADM","2024/10/28 15:17:24", "127.0.0.1", -1, 0, , , "disconnect"
```

4.2.2. LOG CONNECTIONS (WITHOUT SLOW OPERATIONS AND ERRORS)

Error information and slow operations stands a higher level in log server, but by setting DB_LGERR and DB_LGSTM, users can filter most of the error message and slow operations information.

```
DB_LGSVR=4

DB_LGERR=0 //only core dump and DB crash error will be logged
```

```
DB LGSTM=300    //Set a bigger value of DB LGSTM, which means only execution time over
5 minutes will be defined as a slow operation
```

Execute following command to generate log:

```
connect to DBSAMPLE5 sysadm;

//execute slow sql statement

select * from t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 =
c.c1) and (a.c1=d.c1) and a.c2='ABC';

//execute error sql statement

create table t1(c1 int);

disconnect;
```

The output file will be like follows: Compare to 4.2.1, the following log file doesn't include error and slow operation.

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT ID,NUM STMT,ERROR ARG,OTHER INFO,SQL CMD

0000001, "2024/10/28 16:09:07","2024/10/28 16:09:06","O", 0, 0.263604, "CONNECT ",
8872,"SYSADM","2024/10/28 16:09:06","local",-1,0,,, "start db"

0000002, "2024/10/28 16:09:07","2024/10/28 16:09:07","O", 0, 0.008788, "DISCONN ",
8872,"SYSADM","2024/10/28 16:09:06", "local", -1, 0, , , "disconnect"

0000003, "2024/10/28 16:09:11","2024/10/28 16:09:11","O", 0, 0.004190, "CONNECT ",
18420,"SYSADM","2024/10/28 16:09:11", "127.0.0.1", -1, 0, , , "connect db"

0000004, "2024/10/28 16:09:16","2024/10/28 16:09:16","O", 0, 0.000000, "COMMIT ",
18420, "SYSADM", "2024/10/28 16:09:11", "127.0.0.1", -1, 1, , ,

0000005, "2024/10/28 16:09:22","2024/10/28 16:09:22","O", 0, 0.004186, "DISCONN ",
18420,"SYSADM","2024/10/28 16:09:11", "127.0.0.1", -1, 0, , , "disconnect"
```

4.3. Log Old/New Data When Table is Changed

Add the following keywords to database setting in dmconfig.ini and execute ADD TRACE command by dmsqlc or dmsql32. The log file (txt) will include OLD/NEW data.

```
DB_LGSVR=4
```

Execute the test sql:

```
connect to DBSAMPLE5 sysadm;

create table t1(c1 int);

add trace on t1;

insert into t1 values(1);

update t1 set c1=2 where c1=1;

delete from t1;
```

The output file (txt) will be like follows:

LOG file: **INFO_06552_n** means the ID to match in the txt file.

```
0000010, "2024/10/23 12:37:07","2024/10/23 12:37:07","O", 0,0.000000,"PREPARE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,,"add trace on t1"

0000011, "2024/10/23 12:37:07","2024/10/23 12:37:07","O", 0,0.022646,"EXECUTE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,"INFO_06552_1","add trace on t1"

0000012, "2024/10/23 12:37:17","2024/10/23 12:37:17","O", 0,0.001089,"PREPARE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,,"insert into t1 values(1)"

0000013, "2024/10/23 12:37:17","2024/10/23 12:37:17","O", 0,0.008700,"EXECUTE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,"INFO_06552_2","insert into t1
values(1)"

0000014, "2024/10/23 12:37:30","2024/10/23 12:37:30","O", 0,0.000000,"PREPARE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,,"update t1 set c1=2 where c1=1"

0000015, "2024/10/23 12:37:30","2024/10/23 12:37:30","O", 0,0.007430,"EXECUTE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,"INFO_06552_3","update t1 set
c1=2 where c1=1"

0000016, "2024/10/23 12:37:40","2024/10/23 12:37:40","O", 0,0.000556,"PREPARE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,,"delete from t1"

0000017, "2024/10/23 12:37:40","2024/10/23 12:37:40","O", 0,0.008467,"EXECUTE ",
6552,"SYSADM","2024/10/23 12:36:21","127.0.0.1",0,1,,"INFO_06552_4","delete from t1"
```

TXT file:

```
INFO 06552 1 -----
[6552]: 2024/10/23 12:37:07 - ENG [add_pages] : [11] FN(1): ADD=20 /43, PE:(1,23,42),
[EXEC] add trace on t1

INFO 06552 2 -----
[TRACE ON SYSADM.T1 - INSERT]

[NEW.C#1] 1

INFO_06552_3 -----
[TRACE ON SYSADM.T1 - UPDATE]

[OLD.C#1] 1

[NEW.C#1] 2

INFO_06552_4 -----
[TRACE ON SYSADM.T1 - DELETE]

[OLD.C#1] 2
```

4.4. Log SQL and Input Parameters

If massive data is updated, prepared statement is often used. However, the parameters will not be stored in the log file. So as triggers, because trigger will activate automatically, log server will not log triggers and their parameters. In this case, set DB_LGPAR to log parameters.

Add the following keywords to database setting in dmconfig.ini.

```
DB_LGSVR=4
```

```
DB LGPAR=4      //log parameters (include trigger and stored procedure)
DB_LGSQL=1      //log sql to check which command match which parameters
```

Execute the test sql: SAMPLE.Employee in DBSAMPLE5 has trigger InsertSP and TriggerInsert, InsertSP will call stored procedure sp_trigger before insert into SAMPLE.Employee; TriggerInsert will insert the same values to SAMPLE.Employee_bak after insert into SAMPLE.Employee.

```
connect to DBSAMPLE5 sysadm;

insert into sample.employee values(?,?,?);

456,'test user1','test address1';

end;
```

The output file will be like follows:

LOG file: **INFO_19000_1** means the ID in the txt file, auto executed trigger and stored procedure will be logged in txt file.

```
SERNO,LOG TIME,BEG TIME,STATE,RETCODE,EXE TIME,SV FUNC,CONNECT ID,USERNAME,LOGIN TIME
,LOGIN_ADDR,STMT_ID,NUM_STMT,ERROR_ARG,OTHER_INFO,SQL_CMD

0000001, "2024/10/25 12:18:59","2024/10/25 12:18:59","O",    0,0.408577,"CONNECT ",
16360,"SYSADM","2024/10/25 12:18:59","local",-1,0,,, "start db"

0000002, "2024/10/25 12:18:59","2024/10/25 12:18:59","O",    0,0.009187,"DISCONN ",
16360,"SYSADM","2024/10/25 12:18:59","local",-1,0,,, "disconnect"

0000003, "2024/10/25 12:19:03","2024/10/25 12:19:03","O",    0,0.006413,"CONNECT ",
19000,"SYSADM","2024/10/25 12:19:03","127.0.0.1",-1,0,,, "connect db"

0000004, "2024/10/25 12:19:11","2024/10/25 12:19:11","O",    0,0.002229,"PREPARE ",
19000,"SYSADM","2024/10/25 12:19:03","127.0.0.1",0,1,,, "insert into sample.employee
values(?,?,?)"

0000005, "2024/10/25 12:19:25","2024/10/25 12:19:25","O",    0,0.057285,"EXECUTE ",
19000,"SYSADM","2024/10/25 12:19:03","127.0.0.1",0,1,, "INFO_19000_1", "insert into
sample.employee values(?,?,?)"
```

TXT file: **INFO_19000_1** means the parameters from the SQL command, the following two parts mean the command and parameters of auto executed trigger and stored procedure executed by trigger.

```
INFO 19000 1 -----

[Parameter]: 456, 'test_user1', 'test_address1'

[ExecSQL] CREATE TRIGGER InsertSP BEFORE INSERT ON SAMPLE.Employee FOR EACH ROW
(call sample.sp_Trigger(new.id,new.name,new.address) )

[NEW.C#1] 456

[NEW.C#2] 'test_user1'

[NEW.C#3] 'test address1'

[Parameter]: 456, 'test_user1', 'test_address1'
```

```
[ExecSQL] Insert into SAMPLE.Employee bak
values (:id,:id,:name,'From_SP',:address,'From_SP',CURDATE())

[Parameter]: 456, 456, 'test user1', 'test address1'

[ExecSQL] CREATE TRIGGER TriggerInsert AFTER INSERT ON SAMPLE.Employee FOR EACH
ROW ( INSERT INTO SAMPLE.Employee_bak VALUES ( new.id, new.id, new.name, new.name,
new.address, new.address, CURDATE() ) )

[NEW.C#1] 456

[NEW.C#2] 'test user1'

[NEW.C#3] 'test_address1'

[Parameter]: 456, 'test_user1', 'test_address1'
```

4.5. Lock Information

DBMaker log server has no direct keywords related to lock. However, users can check the error message about lock information.

```
DB_LGSRV=2      //set to 2 or higher to log error message
DB_LGERR=3      //set to 3 or 4 so the error message can include error message about
lock
```

Execute the following sql by dmsql32:

```
connect to DBSAMPLE5 sysadm;

set autocommit off;

create table t1(c1 int);

insert into t1 values(1);

use 2;

connect to DBSAMPLE5 sysadm;

delete from t1 where c1=1;
```

The output file will be like follows: represents error log, 1102 is the error number

```
SERNO,LOG_TIME,BEG_TIME,STATE,RETCODE,EXE_TIME,SV_FUNC,CONNECT_ID,USERNAME,LOGIN_TIME
,LOGIN_ADDR,STMT ID,NUM STMT,ERROR ARG,OTHER INFO,SQL CMD

0000001, "2024/10/22 11:08:10","2024/10/22 11:08:05","X", 1102, 5.000989, "PREPARE ",
17396, "SYSADM", "2024/10/22 11:08:00", "127.0.0.1", 0, 1, ": Object ID :
0.402.9 ,Connection ID : 19300, LKMode : X .NONE, LKStatus : GRANTED, Command :
[EXIT] insert into t1 values(1) [unknown 1123], 501507, 0, 0,,,"delete from t1
where c1=1"
```

5. Appendix

5.1. SQL Script

The following script is used in:

chapter [3.1.1. "DB_LGSVR = 1"](#),

chapter [3.1.3. "DB_LGSVR = 3"](#),

chapter [3.1.4. "DB_LGSVR = 4"](#),

chapter [3.1.5. "DB_LGSVR = 5"](#),

chapter [3.1.6. "DB_LGSVR = 6"](#),

chapter [3.2.1. "DB_LGSYS = 1"](#),

chapter [3.3.1. "DB_LGERR = 1"](#),

chapter [3.3.2. "DB_LGERR = 4"](#),

chapter [3.4.1. "DB_LGPAR = 1"](#),

chapter [3.5.1. "DB_LGPLN = 1"](#).

```
connect to newlogsystem sysadm;

//Warning(76)
grant connect to user1;
grant connect to user1;
revoke connect from user1;

//Error(6018)
create table int(a int);

create table test(num serial(1),id int,sqlName char(10));
insert into test(id,sqlName) values(1,'abcde');

//WARNING(63)
insert into test(id,sqlName) values(2,'abcdefghijkl');
insert into test(id,sqlName) values(?,?);
3,'abc';
```



```
4,'abcdefghijk';

5,'abc';

end;

select * from test;

//Error20916

use 2;

connect to newlogsystem sysadm;

use 3;

connect to newlogsystem sysadm;

use 4;

connect to newlogsystem sysadm;

use 5;

connect to newlogsystem sysadm;

use 6;

connect to newlogsystem sysadm;

use 7;

connect to newlogsystem sysadm;

use 2;

disconnect;

use 3;

disconnect;

use 4;

disconnect;

use 5;

disconnect;

use 6;

disconnect;

use 1;

//To test slow SQL, run slow sql data.sql to prepare data

//please refer to the script in Appendix

run 'D:\slow sql data.sql';

select * from t1 a left outer join (t2 b,t3 c, t4 d) where (a.c1=b.c1) and (a.c1 =
c.c1) and (a.c1=d.c1) and a.c2='ABC';
```

```
//To log commit,rollback command  
  
set autocommit off;  
  
insert into test(id,sqlName) values(6,'commit');  
  
savepoint point1;  
  
insert into test(id,sqlName) values(7,'rollback');  
  
rollback to point1;  
  
commit;  
  
drop table test;  
  
commit;
```