

DBMaker/DBMaster FAQ Document

Question:

What about JDBC connection support parameters?

Answer:

JDBC connect string not only include connect information but also support some database keyword settings. In DBMaker connect string keyword support **UID, PWD, SVADR, PTNUM, ATCMT, CTIMO, DIFCO, STRSZ, STROP, DSCMT, ERRLCODE, CLILCODE** keyword currently.

LTIMO(5.4.3 #29936, 20200708)

We will introduce how to write JDBC Connection String and give simple sample to illustrate the usage for these keywords:

JDBC Connection String:

```
URL=" jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;uid=sysadm;pwd=abc;
atcmt=1; strsz=1000; strop=1; dscmt=1; errlcode=JA; clilcode=Shift-JIS;"
```

```
Conn = DriverManager.getConnection (URL);
```

For example:

UID, PWD

Write UID and PWD in URL string and getConnection (url) by url one parameter. Program connects to database successfully by this way.

Test Program:

```
public class KeyWords {

    static String driver ="dbmaker.sql.JdbcOdbcDriver";

    static String url =
"jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;uid=sysadm;pwd=";

    public static void main(String[] args) throws SQLException,
    ClassNotFoundException {

        Class.forName (driver);
```

DBMaker/DBMaster FAQ Document

```
Connection conn = DriverManager.getConnection (url);

try{

    System.out.print ("Connect Successful!");

}

catch (Exception e){

    e.printStackTrace ();

}

}

}
```

ATCMT: we can obtain the Auto-commit status by **getAutoCommit ()** function.

Test program:

```
public class KeyWords {

    static String driver ="dbmaker.sql.JdbcOdbcDriver";

    static String url = "jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;ATCMT = 0";

    public static void main(String[] args) throws SQLException,
    ClassNotFoundException {

        Class.forName (driver);

        Connection conn = DriverManager.getConnection (url, "SYSADM", "");

        try{

            Boolean i = conn.getAutoCommit ();

            System.out.print("AUTO-Commit-STAUS□"+i+"\n");

        }

        catch (Exception e){
```

Email: <Support_RD@syscom.com.tw>

DBMaker/DBMaster FAQ Document

```
e.printStackTrace ();  
  
    }  
  
}  
  
}
```

Step (1): Set ATCMT = 0 in url

Result in Console:

AUTO-Commit-STAUS□false

Step (2): Set ATCMT = 1 in url in above program.

Result in console:

AUTO-Commit-STAUS□true

DSCMT: This keyword specifies whether to commit a transaction when an application is disconnecting from the database

Test Program:

```
public class KeyWords {  
  
    static String driver ="dbmaker.sql.JdbcOdbcDriver";  
  
    static String url = "jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;ATCMT =  
0;DSCMT = 0;";  
  
    public static void main(String[] args) throws SQLException,  
ClassNotFoundException {  
  
        Class.forName(driver);  
  
        Connection conn = DriverManager.getConnection (url, "SYSADM", "");  
  
        try{  
  
            Boolean i = conn.setAutoCommit ();
```

Email: <Support_RD@syscom.com.tw>

DBMaker/DBMaster FAQ Document

```
System.out.print("AUTO-Commit-STATUS "+i+"\n");

Statement stmt = conn.createStatement ();

int effctCount = stmt.executeUpdate("UPDATE jobs SET
JOB_DESC='Snow,Modify,DBMaker-test' WHERE JOB_ID = 1");

System.out.println ("The Modified Rows Count is :"+effctCount+"\n");

ResultSet rs = stmt.executeQuery ("select * from jobs");

rs.next ();

String s1= rs.getString (1);

String s = rs.getString ("JOB_DESC");

System.out.print ("The Modified\nJOB_ID:"+s1+"JOB_DESC:"+s);

}

catch (SQLException e){

    e.printStackTrace ();

}

}
```

Step (1): DB_DSCMT keyword act on auto-commit off status, DBMaker decides to whether execute commit command when a client's application issues a SQLDisconnect.so we should set ATCMT = 0 before test keyword DSCMT.

```
URL = "jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;ATCMT = 0; DSCMT = 0;"
```

Step (2): Execute UPDATE statement in AP and Query the result from DBSAMPLE5 to examine the modified result whether affected in database.

Step (3): Query the result from dmSQL and we will find the new data which modified via AP can't affect in database. This proves DSCMT = 0 is an available setting in URL.

```
dmSQL> select * from jobs where JOB_ID = 1;
```

JOB_ID	JOB_DESC	MIN_LVL	MAX_LVL
--------	----------	---------	---------

Email: <Support_RD@syscom.com.tw>

DBMaker/DBMaster FAQ Document

```
=====
=====
=====

1 New Hire - Job not specified                10    10

Step (4): set DSCMT=1, execute STEP (2) and STEP (3) again. We will find the new
data which modified via AP had been affected in database too. This proves DSCMT
= 1 is an available setting in URL.

dmSQL> select * from jobs;

JOB_ID          JOB_DESC          MIN_LVL MAX_LVL
=====
=====
=====

1 Snow, Modify, DBMaker-test                10    10
```

ERRLCODE: you can set ERRLCODE = en, ja, zh_TW, zh_CN, UTF-8 etc which supported by DBMaker.

Test Program: we can get the ERRLCODE error message character set by function **GETSYSINFO()**.

```
public class KeyWords {

    static String driver ="dbmaker.sql.JdbcOdbcDriver";

    static String url = "jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;ERRLCODE
=zh_CN;";

    public static void main(String[] args) throws SQLException,
    ClassNotFoundException {

        Class.forName (driver);

        Connection conn = DriverManager.getConnection (url, "SYSADM", "");
```

DBMaker/DBMaster FAQ Document

```
try {  
    Statement stmt = conn.createStatement ();  
    ResultSet rs = stmt.executeQuery ("SELECT GETSYSINFO ('errlcode')");  
    rs.next ();  
    String s = rs.getString (1);  
    System.out.print ("ERRLCODE:"+s);  
}
```

Step (1): set ERRLCODE =zh_CN in url.

Result in console:

ERRLCODE:zh_CN

Step (2): set ERRLCODE =en in URL.

Result in console:

ERRLCODE:en

CLILCODE: This keyword specifies the language code in the client side. When use multilingual database and the LCODE of the Database Server is UTF-8, client side can use several local codes to connect to the UTF-8 database server. These codes can be BIG5, EUC-JP, and UTF-8 etc which supported by DBMaker.but it the Server LCODE is not UTF-8, the default value would be same with server LCODE.

Test program: Use the same Test Program with above which tested keyword ERRLCODE and only need do following change:

Replace Test Program of "ResultSet rs = stmt.executeQuery ("SELECT GETSYSINFO ('errlcode')");" with "ResultSet rs = stmt.executeQuery ("SELECT GETSYSINFO ('clilcode')");"

Replace URL of **ERRLCODE** (ERRLCODE=zh_CN) with **CLILCODE** = EUC-JP OR **CLILCODE**=GBK.

```
Step (1) set CLILCODE = EUC-JP in url, url =  
"jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;CLILCODE =EUC-JP;"
```

Email: <Support_RD@syscom.com.tw>

DBMaker/DBMaster FAQ Document

Result in console:

CLILCODE: EUC-JP

Step (2) set CLILCODE = GBK, url =

```
"jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;CLILCODE =GBK";
```

Result in console:

CLILCODE: GBK

STROP: This keyword specifies whether space padding is removed before applying the string concatenation. A value of 0 indicates space padding for fixed length CHAR type data is kept before applying the string concatenation operator. A value of 1 indicates the space padding is removed before applying the string concatenation operator.

Test program: Use the same Test Program with above which tested keyword ERRLCODE and only need do following changes:

Replace Test Program of "ResultSet rs = stmt.executeQuery ("**SELECT GETSYSINFO ('errlcode')**");" with "ResultSet rs = stmt.executeQuery ("**SELECT CONCAT ('aaa ', 'bbbb')**");

Replace URL of **ERRLCODE** (ERRLCODE=zh_CN) with **STROP = 0** or **STROP = 1**

Step (1): set STROP=0 in URL: url =

```
"jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;STROP=0";
```

Result in Console:

aaa bbbb

Step (2): set STROP=1 in URL: url =

```
"jdbc:dbmaker://127.0.0.1:2453/DBSAMPLE5;STROP=1";
```

Result in console:

Aaabbbb

DBMaker/DBMaster FAQ Document

Email: <Support_RD@syscom.com.tw>