**Mycat for SQL Server**

**10 minutes quick start**

**1. Download JDK**

Mycat is developed based on JAVA, so you have to install JRE, and the version of JRE should be JDK7 or higher as Mycat uses some features of it.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

for more information, please see Mycat project homepage [https://github.com/MycatApache](https://github.com/MyCATApache)

Note: All Mycat source codes and docs can be downloaded from this website.

**2. Download Mycat**

With the Windows environment, you can download Mycat-server-xxxxx-win.tar.gz from <https://github.com/MyCATApache/Mycat-download> and unzip it. We suggest that you deposit it to the root directory of one disk partition.



directory explanation is shown as following.

**bin\:**

deposits executable files both for Windows and Linux, the version can be either service encapsulation or shell command nowrap.

You can run Mycat under Windows by Mycat.bat. If it does not work, you can run it by startup\_nowrap.bat too, this ensure that java commands can be executed on the command line.

Under Windows, Mycat startup is wrapped to system service to start or stop Mycat.

Enter the bin directory, input ./Mycat start to start Mycat service.

**conf\:**

deposits the configuration files.

Server.xml is for Mycat server parameters tuning and users authentication.

Schema.xml is for logical databases, logical tables and datanodes definition.

Rule.xml is for rule of sharding.

Some of the parameter information files of the sharding rule are also deposited in this directory standalone. You should reboot Mycat or run reload by port 8066 to bring the modification into effect.

**lib\:**

deposits jar files that Mycat relied on.

**logs\:**

Mycat log is deposited in logs/Mycat.log, one file per day. The setting of logs is in conf\log4j.xml, you can alter the level of output to debug to show more information for troubleshooting.

1. **Configuration**

**3.1.Add the dataHost information into config\schema.xml**

<?xml version="1.0"?>

<!DOCTYPE Mycat:schema SYSTEM "schema.dtd">

<Mycat:schema xmlns:Mycat="http://org.opencloudb/">

 <schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100">

 <!-- random sharding using mod sharind rule -->

 <table name="test\_table" primaryKey="ID" dataNode="dn1,dn2,dn3" rule="mod-long" />

 </schema>

 <dataNode name="dn1" dataHost="sqlserver1" database="db1" />

 <dataNode name="dn2" dataHost="sqlserver1" database="db2" />

 <dataNode name="dn3" dataHost="sqlserver1" database="db3" />

 <dataHost name="sqlserver1" maxCon="1000" minCon="1" balance="0" writeType="0" dbType="sqlserver" dbDriver="jdbc">

 <heartbeat></heartbeat>

 <connectionInitSql></connectionInitSql>

 <writeHost host="hostM1" url="jdbc:sqlserver://localhost:1433" user="sa" password="sa" >

 </writeHost>

 </dataHost>

</Mycat:schema>

Note:

\* the value of dbDriver must be jdbc.

\* dbType means the type of database.

\* url is the address of JDBC connection, similar to the normal Java web application jdbc url. But the port is not always 1433,it depends on which port SQL Server listens to.

\* user and password are the user account and password of SQL Server for Mycat.

\* heartbeat is the check statement of heartbeat, it can be null.

\* connectionInitSql is the initialiation statement connecting to the SQL Server.

\* It's need to put the sqljdbc4.jar library to /lib directory manually.

\* If you want to support multiple databases, you can specify them within the dataNode tag.

**3.2.Create databases**

You should create databases in the dataHosts defined in config\schema.xml manually.

**3.3.Start Mycat**

In D:\Mycat-server-1.5.1-win\bin directory, you execute startup\_nowrap.bat to start the service of Mycat.

**3.4.Login in Mycat**

Using the Mysql command to login Mycat:

mysql -utest -P8066 -ptest

\* the account and the password are defined in config/server.xml

**3.5.Create tables**

Creating tables in Mycat, and they will be created in the dataNodes which are defined in config\schema.xml. shown as following:

<table name="test\_table" primaryKey="ID" dataNode="dn1,dn2,dn3" rule="mod-long" />

then you insert value as id into the table test\_table.



You can see that:

1. The table test\_table is created in db1, db2 and db3.
2. The value is just inserted into the test\_table in db2, for the rule of the table we defined is mod-long.

