

How To Embed Mysql Database Into Java Application

Embedding mysql into java application is **not straightforward**. Therefore, I will try to explain necessary operations step by step.

1. Download **this library** file then extract jar files into a folder.
2. Create a Java application and add downloaded jar files into the application lib folder

If you use maven application add following dependencies into pom.xml file

```
1 <dependency>
2   <groupId>mysql</groupId>
3   <artifactId>mysql-connector-mxj-gpl</artifactId>
4   <version>5.0.12</version>
5 </dependency>
6 <dependency>
7   <groupId>mysql</groupId>
8   <artifactId>mysql-connector-mxj-gpl-db-files</artifactId>
9   <version>5.0.12</version>
10 </dependency>
11 <dependency>
12   <groupId>mysql</groupId>
13   <artifactId>mysql-connector-java</artifactId>
14   <version>5.1.32</version>
15 </dependency>
```

Note: Above dependencies can be found in maven central repository, but I strongly **suggest** that use these downloaded jar files by installing them locally by using following mvn commands:

```
1 mvn install:install-file -Dfile=mysql-connector-mxj-gpl-db-files-5.0.12.jar ^
2 -DgroupId=mysql -DartifactId=mysql-connector-mxj-gpl-db-files -Dversion=5.0.12 -Dpackaging=jar
3
4 mvn install:install-file -Dfile=mysql-connector-java-5.1.32.jar ^
5 -DgroupId=mysql -DartifactId=mysql-connector-java -Dversion=5.1.32 -Dpackaging=jar
6
7 mvn install:install-file -Dfile=mysql-connector-mxj-gpl-5.0.12.jar ^
8 -DgroupId=mysql -DartifactId=mysql-connector-mxj-gpl -Dversion=5.0.12 -Dpackaging=jar
```

3. After step 1 and step create a class named as **ConnectorMXJObjectTestExample.java** and add following codes:

```
1 import java.io.File;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.util.HashMap;
5 import java.util.Map;
6 import com.mysql.management.MysqldResource;
7 import com.mysql.management.MysqldResourceI;
8 import com.mysql.management.util.QueryUtil;
9 public class ConnectorMXJObjectTestExample {
10     public static final String DRIVER = "com.mysql.jdbc.Driver"
11     public static final String JAVA_IO_TMPDIR = "java.io.tmpdir"
12     public static void main(String[] args) throws Exception {
13         File ourAppDir = new File(System.getProperty(JAVA_IO_TMPDIR));
14         File databaseDir = new File(ourAppDir, "mysql-mxj");
15         int portNumber = Integer.parseInt(System.getProperty("c-mxj_test_port",
16             "3336"));
17         String userName = "olyanren";
18         String password = "1987";
19         MysqldResource mysqlResource = startDatabase(databaseDir, portNumber,
20             userName, password);
21         Class.forName(DRIVER);
22         Connection conn = null;
23         try {
24             String dbName = "our_test_app";
```

```

25         String url = "jdbc:mysql://localhost:" + portNumber + "/" + dbName
26             + "?" + "createDatabaseIfNotExist=true"
27             ;
28         conn = DriverManager.getConnection(url, userName, password);
29         String sql = "SELECT VERSION()";
30         String queryForString = new QueryUtil(conn).queryForString(sql);
31
32         System.out.println("-----");
33         System.out.println(sql);
34         System.out.println("-----");
35         System.out.println(queryForString);
36         System.out.println("-----");
37         System.out.flush();
38         Thread.sleep(100); // wait for System.out to finish flush
39     } finally {
40         try {
41             if (conn != null) {
42                 conn.close();
43             }
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47         try {
48             mysqlDResource.shutdown();
49         } catch (Exception e) {
50             e.printStackTrace();
51         }
52     }
53 }
54 public static MysqlDResource startDatabase(File databaseDir, int port, String userName,
55 String password) {
56     MysqlDResource mysqlDResource = new MysqlDResource(databaseDir);
57     Map database_options = new HashMap();
58     database_options.put(MysqlDResourceI.PORT, Integer.toString(port));
59     database_options.put(MysqlDResourceI.INITIALIZE_USER, "true");
60     database_options.put(MysqlDResourceI.INITIALIZE_USER_NAME, userName);
61     database_options.put(MysqlDResourceI.INITIALIZE_PASSWORD, password);
62
63     mysqlDResource.start("test-mysqlD-thread", database_options);
64     if (!mysqlDResource.isRunning()) {
65         throw new RuntimeException("MySQL did not start.");
66     }
67     System.out.println("MySQL is running.");
68     return mysqlDResource;
69 }
70 }

```

If you see the following error, then you have to **override platform-map.properties** file located in **mysql-connector-mxj-gpl-db-files-5.0.12.jar**

```

1 Exception in thread "main" java.util.MissingResourceException: Resource '5-5-9/Windows_8-
2 x86/mysqlD.exe' not found

```

New platform-map.properties file:

```

1 #String key = System.getProperty("os.name") + "-" + System.getProperty("os.arch");
2 #key = key.replace(' ', '_').replace('/', '_').replace('\\', '_');
3
4 Linux-i386=Linux-i386
5 Linux-x86=Linux-i386
6 Linux-i686=Linux-i386
7 Linux-x86_64=Linux-i386
8 Linux-amd64=Linux-i386
9 Linux-ia64=Linux-i386
10
11 Mac_OS_X-i386=Mac_OS_X-i386
12 Mac_OS_X-x86_64=Mac_OS_X-i386

```

```
13
14 SunOS-sparc=SunOS-sparc
15 Solaris-sparc=SunOS-sparc
16 SunOS-sparcv9=SunOS-sparc
17 SunOS-x86=SunOS-x86
18 Solaris-x86=SunOS-x86
19 SunOS-amd64=SunOS-x86
20
21 FreeBSD-x86=FreeBSD-x86
22 FreeBSD-i386=FreeBSD-x86
23
24 Windows_7-x86=Win-x86
25 Windows_Vista-x86=Win-x86
26 Windows_2003-x86=Win-x86
27 Windows_XP-x86=Win-x86
28 Windows_2000-x86=Win-x86
29 Windows_NT-x86=Win-x86
30 Windows_NT_(unknown)-x86=Win-x86
31 Windows_Server_2008-x86=Win-x86
32 Windows_Vista-amd64=Win-x86
33 Windows_2003-amd64=Win-x86
34 Windows_2000-amd64=Win-x86
35 Windows_8-x86=Win-x86
```

Note: Only Windows_8-x86=Win-x86 line is added.

To **override** this file, you have to create a new file in the classpath with same name(**platform-map.properties**) and copy above content into this file. If you use maven, copy this new file into **resources** folder.

Result

When you run **ConnectorMXJObjectTestExample.java** file, automatically a folder named as **mysql-mxj** is created in **java.io.tmpdir** location. In Windows, this location is **C:\Users\{UserName}\AppData\Local\Temp**

All necessary files for an embedded mysql database is located in the mysql-mxj folder. You can change this folder name if you want.

Database Name: our_test_app

Database Port: **3336**

User Name: olyanren

Password: **1987**

Some Important Notes

1. I created mysql-connector-mxj-gpl-db-files-5.0.12.jar file from **this file** by removing platforms except Windows platform.
2. After version **5.0.12**, development of **MySQL Connector/MXJ** has been discontinued.
3. For more information, please **click**

To download our sample application, please **click**