# Complete Guide: Building MySQL Database Driver for Qt 6 using MinGW and Ninja

This document provides a complete, step-by-step guide to building the MySQL database driver for Qt 6.10.0 using both MinGW Makefiles and Ninja generators. It includes setup, configuration, build commands, and common troubleshooting steps.

## 1. Requirements

1. Qt 6.10.0 (installed with source code)
2. MinGW 13.1.0 (usually located in C:/Qt/Tools/mingw1310_64/bin)
3. CMake (version 3.26 or newer)
4. MySQL Connector/C 6.1.11 (winx64 version)
5. Environment variables configured for MinGW and CMake

## 2. Verify Environment Setup

1. Open Command Prompt and verify the tools:
2. gcc --version
3. cmake --version
4. ninja --version (if using Ninja build system)
5. Ensure MySQL connector is installed at: C:/Program Files/MySQL/mysql-connector-c-6.1.11-winx64/

## 3. Prepare MySQL Connector Libraries

If 'libmysqlclient.a' is missing, generate it using the following command (run in MySQL connector 'lib' folder):

```
dlltool -D libmysql.dll -d libmysql.def -l libmysqlclient.a
```

## 4. Configure the Build using CMake

Create and open a new build folder:

```
cd C:\Qt\6.10.0\Src\qtbase\src\plugins\sqldrivers mkdir build cd build
```

### *A. Using MinGW Makefiles*

Run the following CMake command:

```
cmake -G "MinGW Makefiles" -DCMAKE_BUILD_TYPE=Release
-DCMAKE_PREFIX_PATH="C:/Qt/6.10.0/mingw_64" -DMySQL_INCLUDE_DIR="C:/Program
Files/MySQL/mysql-connector-c-6.1.11-winx64/include" -DMySQL_LIBRARY="C:/Program
Files/MySQL/mysql-connector-c-6.1.11-winx64/lib/libmysqlclient.a" ..
```

Then build the driver:

```
mingw32-make
```

### *B. Using Ninja*

To use Ninja instead, clear cache and rebuild:

```
rmdir /s /q CMakeFiles del CMakeCache.txt cmake -G Ninja -DCMAKE_BUILD_TYPE=Release
-DCMAKE_PREFIX_PATH="C:/Qt/6.10.0/mingw_64" -DMySQL_INCLUDE_DIR="C:/Program
Files/MySQL/mysql-connector-c-6.1.11-winx64/include" -DMySQL_LIBRARY="C:/Program
Files/MySQL/mysql-connector-c-6.1.11-winx64/lib/libmysqlclient.a" .. ninja
```

## 5. Install the Plugin

After successful build, the plugin (qsqlmysql.dll) will be located in the 'plugins/sqldrivers' folder. Copy it to your Qt build's plugin path, usually:

```
C:/Qt/6.10.0/mingw_64/plugins/sqldrivers/
```

## 6. Verify Installation

To confirm that the MySQL driver is available, run this command in Qt Creator's console or in your app:

```
qDebug() << QSqlDatabase::drivers();
```

You should see 'QMYSQL' in the output list.

## 7. Common Errors and Fixes

1. **Warning:** The officially supported CMake generator is Ninja. – Use the '-G Ninja' generator instead of 'MinGW Makefiles'.
2. **Error:** generator does not match previous one (Ninja vs MinGW). – Delete 'CMakeCache.txt' and 'CMakeFiles' directory, then re-run CMake.
3. **Error:** Can't create .lib file: Permission denied. – Run the command prompt as Administrator or use a writable directory.
4. **Error:** Could not find MySQL include/lib path. – Ensure correct paths are set in CMake command using quotes ("").

## Conclusion

Following these steps will allow you to successfully build and use the MySQL database driver with Qt 6 using either MinGW or Ninja. If you encounter further issues, ensure your compiler and Qt version match exactly, and check that MySQL connector paths are valid.