

1. Qt and MySQL Installation and Setup	2
2. Database Installation Instructions	12
3. Potential Helpful Links	13
4. Math Formula SQL statements	14
5. Math Formulas	15
6. Creating a Standalone Executable	19
7. Bugs	20

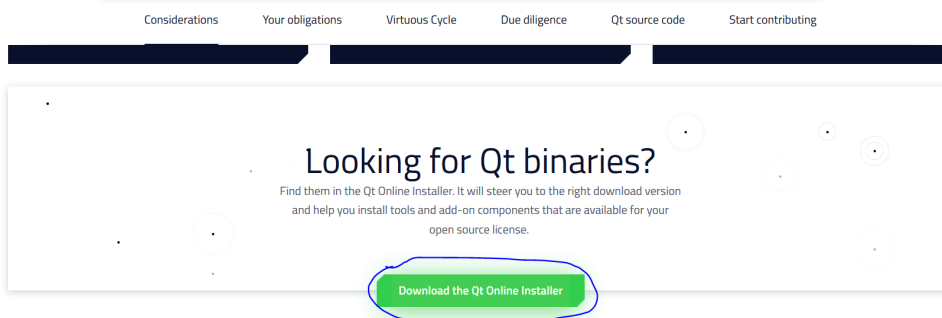
Qt and MySQL Installation and Setup

This page documents the process to install and setup Qt and MySQL to be able to continue the work done by previous teams. This is for Windows 10 operating system.

QT Installation

Steps:

1. Go to <https://www.qt.io/download-open-source>
2. Click on "Download the Qt Online Installer" button at the bottom of the page



3. Scroll down and click the "Download" button

Your download

We detected your operating system as: **Windows**
Recommended download: **Qt Online Installer for Windows**

Not the installer you need? [View other options.](#)

The installer will ask you to optionally sign in using your Qt account credentials. This will ensure you get the right access to the right components, such as those under a commercial license.

Please note:

If you requested a 30-day commercial trial extension or an additional license for embedded components, we will contact you shortly. In the meantime, please get started with Qt.

If you are installing under a Qt open source license, please be sure you are in full compliance with the legal obligations of the (L)GPL v2/3 before installation. For a brief overview visit the [main download page](#) or for more details see the [FAQ](#).

Download



4. Run the downloaded .exe file
5. The installer should pop up. Select "Next".

Qt Setup

Welcome to the Qt online installer

This installer provides you with the option to download either an open source or commercial version of Qt. Please log in with your Qt Account credentials.

If you do not have a Qt Account yet, you can create one free of charge in the next step.

The Qt Account will give you access to Qt downloads, exclusive services, bug reports, code review, and forums & wiki.

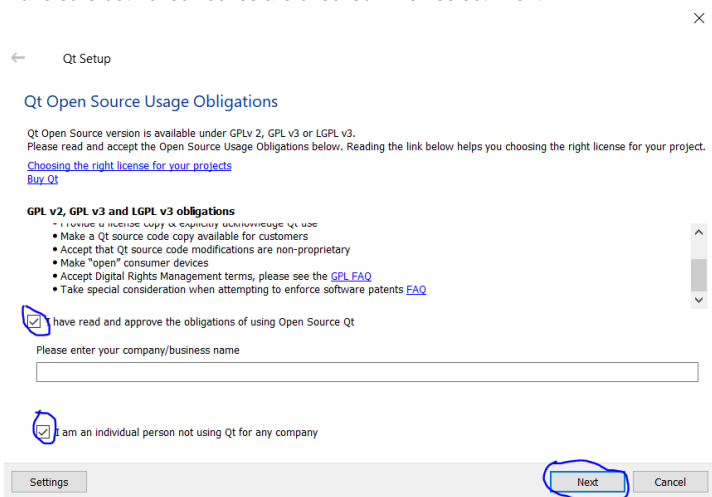
Settings

Next

Cancel

6. Enter your login information or create an account. Select "Next".
 - a. If you created a new account, you will need to check your email for a verification link.
 - b. At the link, you will need to enter your information and select "Individual User"
 - c. After verifying your account, select "Next" again on the installer

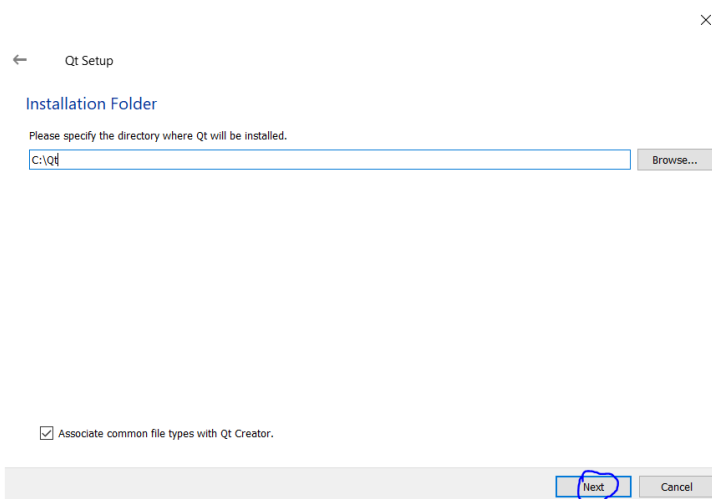
7. Make sure both check boxes are checked. Then select "Next"



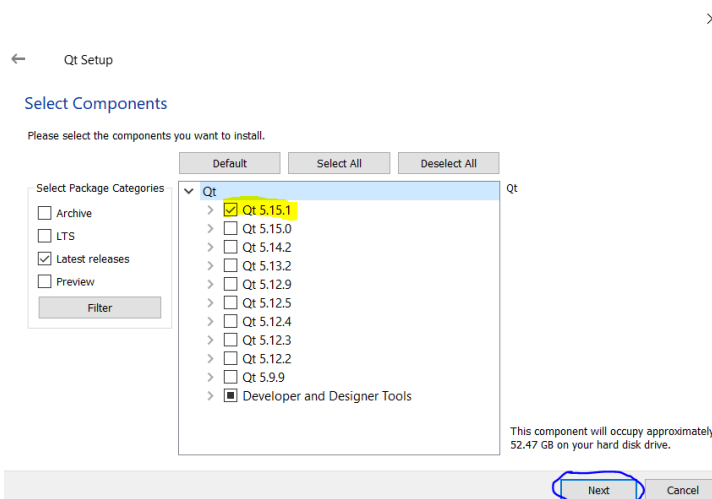
8. A welcome screen should appear. Select "Next".

9. After some initial setup, the installer will prompt you to select whether or not to share usage statistics. Select an option, and click "Next"

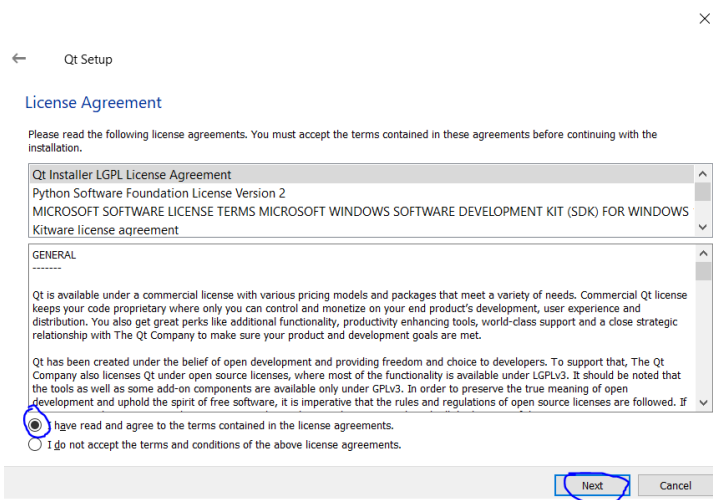
10. Leave the default settings for the installation folder and select "Next"



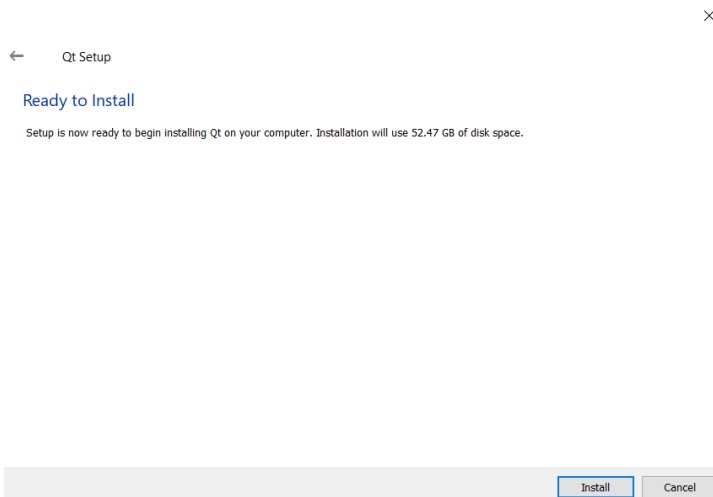
11. Select version 5.15.xx (I used 5.15.1). Click "Next"



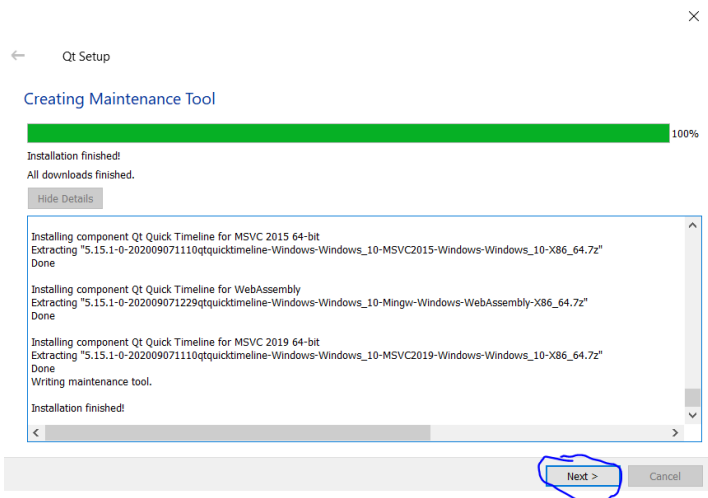
12. Accept the license agreement and click "Next".



13. You can select a different location in the start menu for the shortcuts or just click "Next".
14. Click "Install".



15. After the installation completes, click "Next".



16. Click "Finish" and launch Qt Creator.

×

Qt Setup

Completing the Qt Wizard

Click Finish to exit the Qt Wizard.

☒ Launch Qt Creator



MySQL Installation

1. Go to <https://dev.mysql.com/downloads/installer/> and Click "Download" for the larger installer.

MySQL Community Downloads

MySQL Installer

The screenshot shows the MySQL Installer 8.0.21 download page. At the top, there are tabs for "General Availability (GA) Releases" and "Archives". Below the tabs, the title "MySQL Installer 8.0.21" is displayed. A dropdown menu for "Select Operating System:" is set to "Microsoft Windows". To the right, there is a link "Looking for previous GA versions?". Below this, there is a table with two rows of download links. The first row is for "Windows (x86, 32-bit), MSI Installer" with a size of 24.5M. The second row is for "Windows (x86, 32-bit), MSI Installer" with a size of 427.6M. The "Download" button for the 427.6M version is circled in blue. Below the table, there is a note: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

2. You can choose to create an account, or just click "No thanks, just start my download."

MySQL Community Downloads

Login Now or Sign Up for a free account.

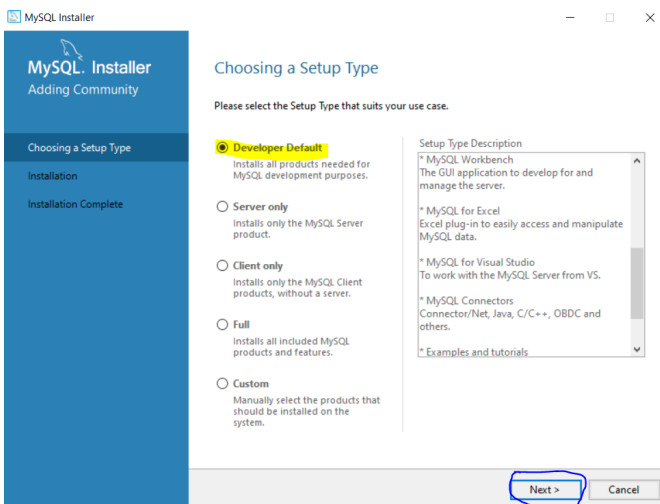
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

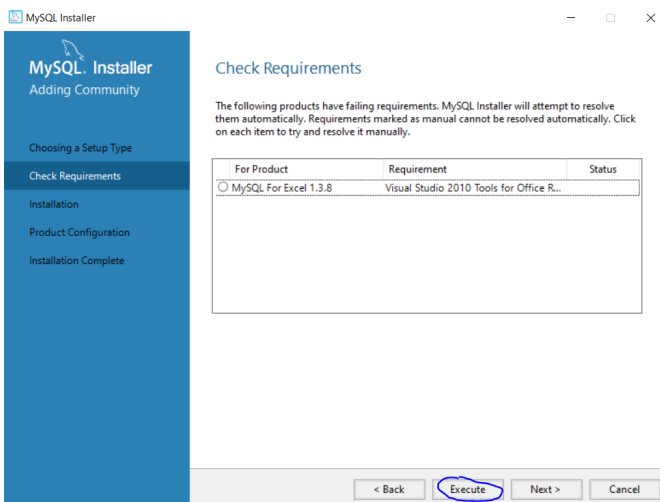
The screenshot shows the MySQL login and sign up section. There are two buttons: "Login »" and "Sign Up »". Below the buttons, there is a paragraph: "MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions."

[No thanks, just start my download.](#)

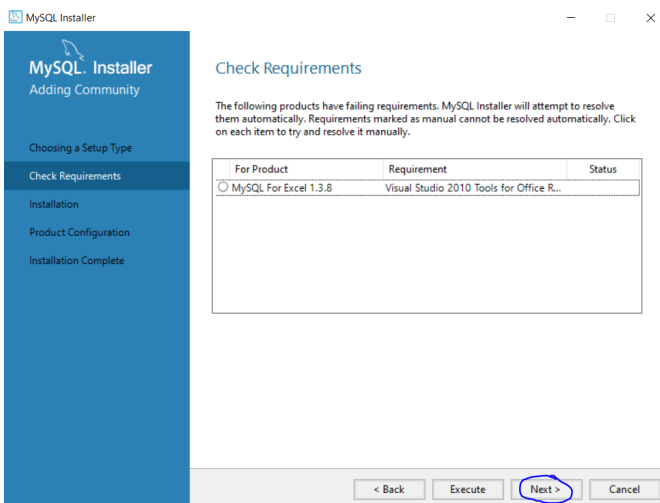
3. After the installer downloads, run the installer.
4. Make sure "Developer Default" is selected, then click "Next".



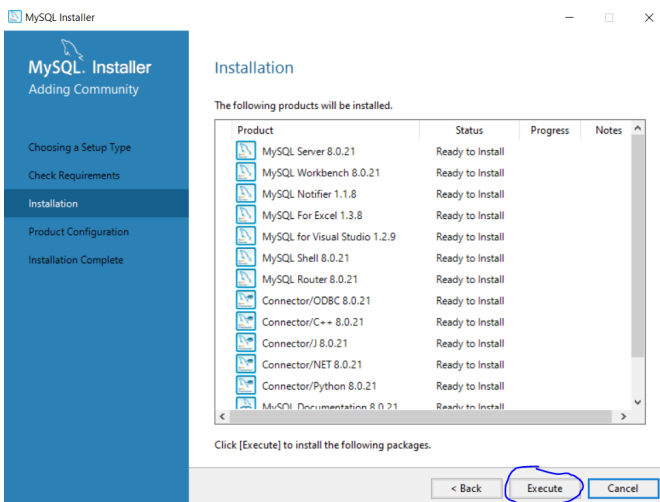
5. Select any issues present and click "Execute" to fix them



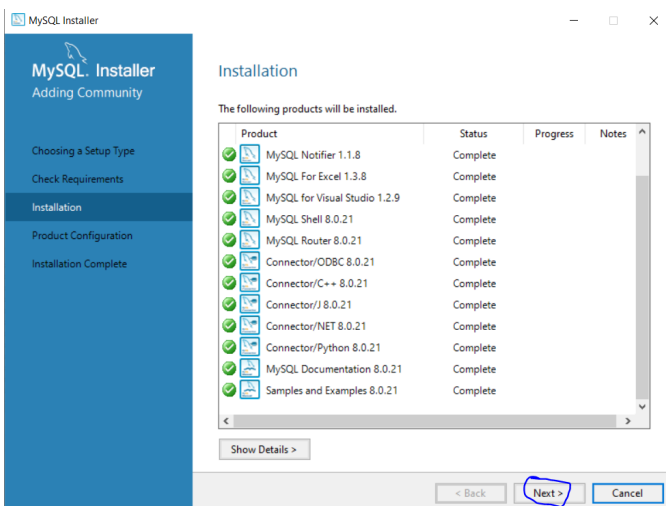
6. After issues are resolved, click "Next".



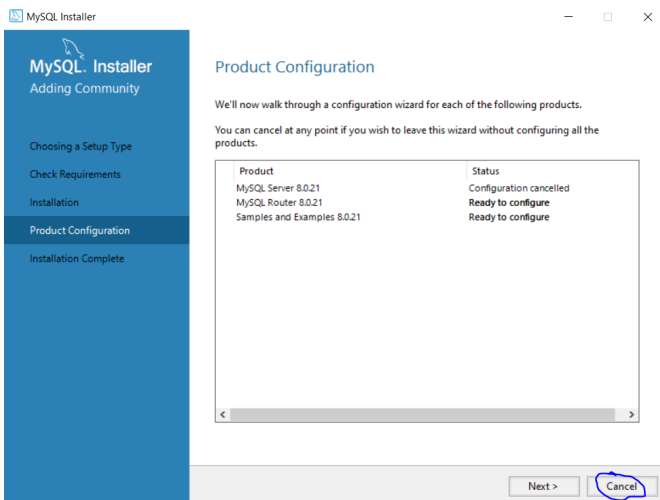
7. Click "Execute".



8. After Installation finishes, click "Next".



9. Click "Cancel" as you do not need to configure these.



Building QMySQL driver

1. After installing MySQL, you need to compile the QMYSQL driver and install it. I did this for the MinGW 8.1 64 bit kit, so this kit will need to be used for building the project. Start by opening mysql.pro with a text editor such as Notepad. Mine was located here: "C:\Qt\5.15.1\Src\qtbase\src\plugins\sql\drivers\mysql\mysql.pro"
2. Remove the following line: QMAKE_USE += mysql
3. Add the following lines to add the paths to the needed MySQL files (these paths may be different depending on where you installed MySQL):

```
INCLUDEPATH+="C:\Program Files\MySQL\MySQL Server 8.0\include"
```

```
LIBS+="C:\Program Files\MySQL\MySQL Server 8.0\lib\libmysql.lib"
```

4. Your mysql.pro file should now look like this:



mysql - Notepad

File Edit Format View Help

```
TARGET = qsqlmysql
```

```
HEADERS += $$PWD/qsql_mysql_p.h
```

```
SOURCES += $$PWD/qsql_mysql.cpp $$PWD/main.cpp
```

```
OTHER_FILES += mysql.json
```

```
PLUGIN_CLASS_NAME = QMYSQLDriverPlugin
```

```
include(../sqldrivabase.pri)
```

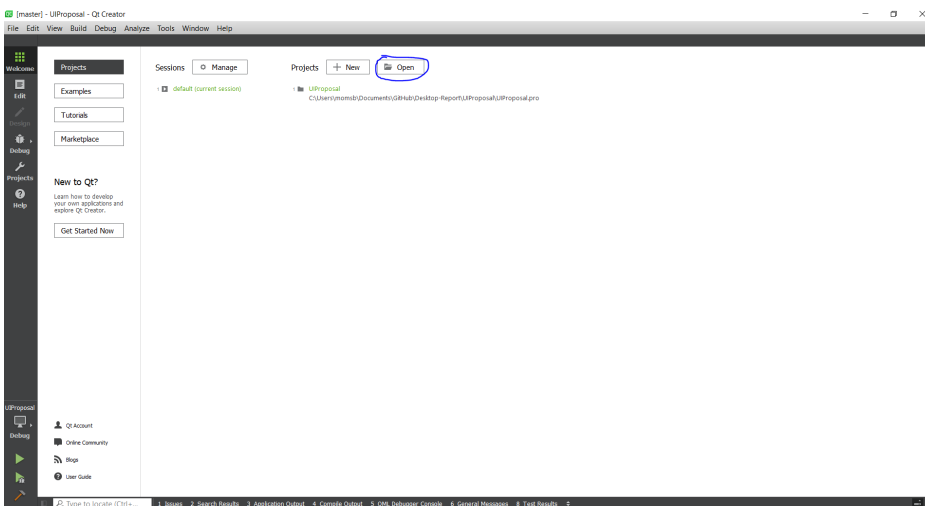
```
INCLUDEPATH+="C:\Program Files\MySQL\MySQL Server 8.0\include"
```

```
LIBS+="C:\Program Files\MySQL\MySQL Server 8.0\lib\libmysql.lib"
```

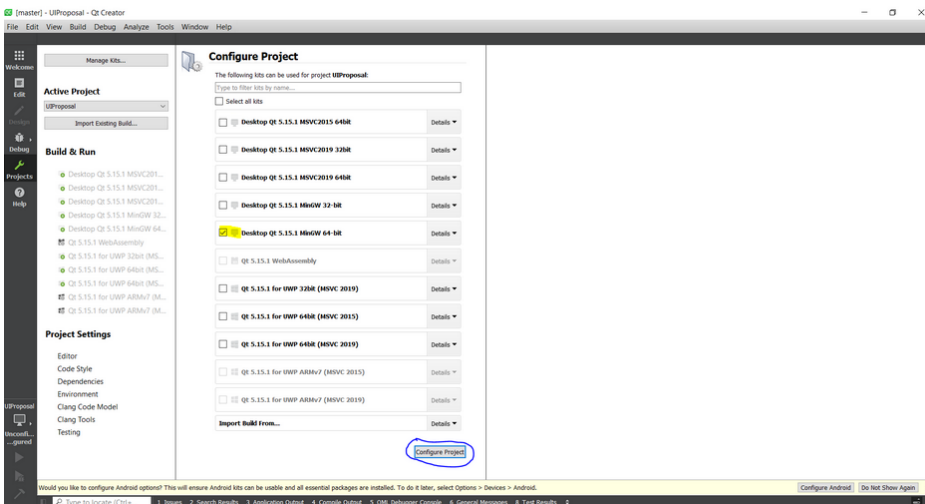
5. Save mysql.pro and close it.
6. Open a command console. (I used Command Prompt)
7. Run the following sequence of commands:
 - a. C:\QT\5.15.1\mingw81_64\bin\qtenv2.bat
 - b. cd C:\QT\5.15.1\Src\qtbase\src\plugins\sqldrivers
 - c. qmake sqldrivers.pro
 - d. cd mysql
 - e. qmake mysql.pro
 - f. mingw32-make
 - g. mingw32-make install
8. Next, copy "C:\Program Files\MySQL\MySQL Server 8.0\lib\libmysql.dll" to "C:\Windows".

Download and Build the Code

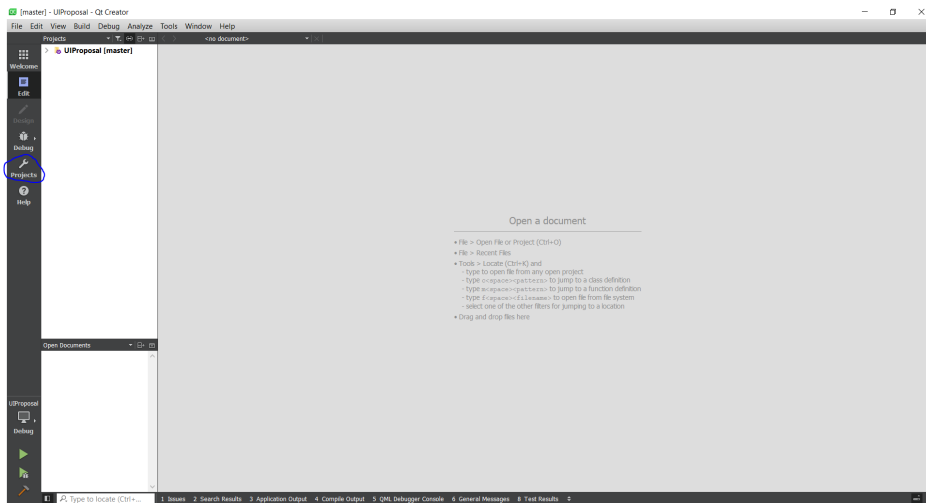
1. Clone the "mfp426/MathCalc" repository to your computer.
2. In Qt Creator, click "Open" to open a project.



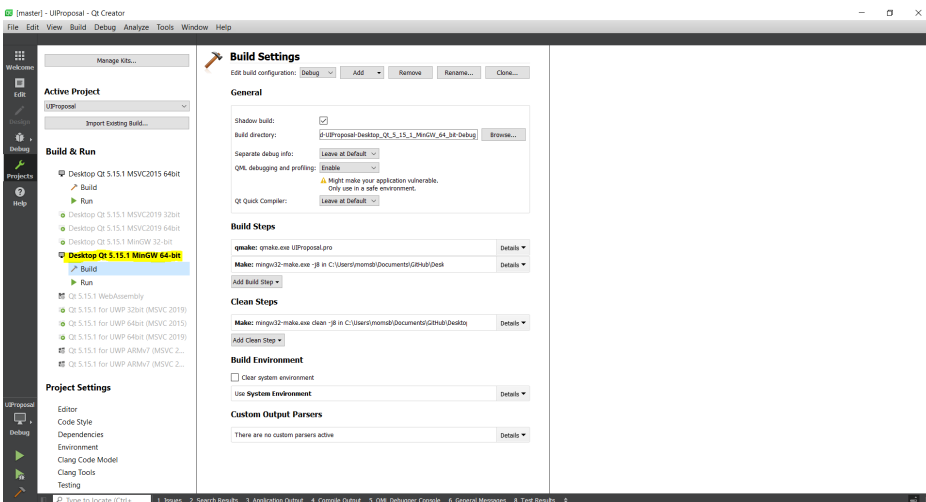
3. In the file browser, navigate to where you cloned the repository and select the Qt project file "FormulasModal".
4. Make sure MinGW 64 bit version 5.15.1 is checked since that is the kit we built MySQL for. Click "Configure Project".



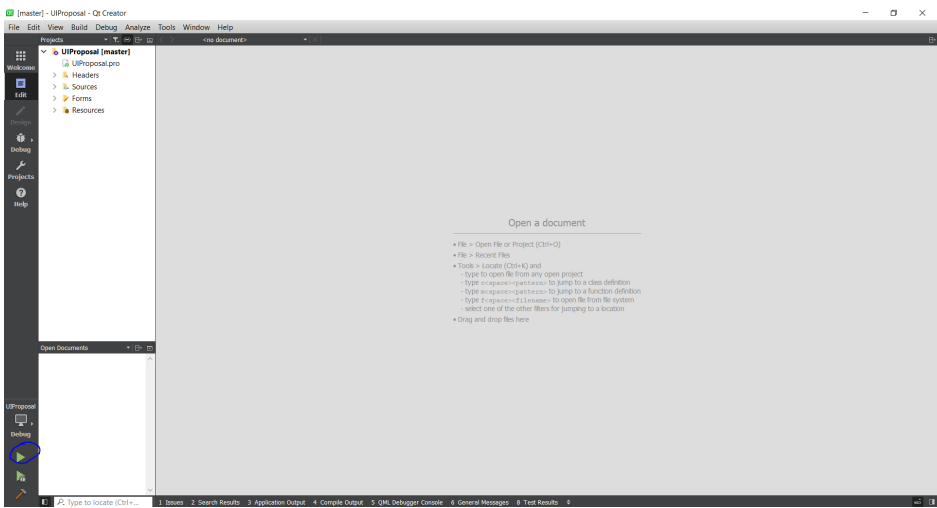
a. If you did not originally select MinGW 64 bit, you can change the kit you are using. Click on "Projects" on the left side.



b. Click on MinGW 64 bit to change to using that kit.



5. Click on the arrow button at the bottom left to build and run the project.

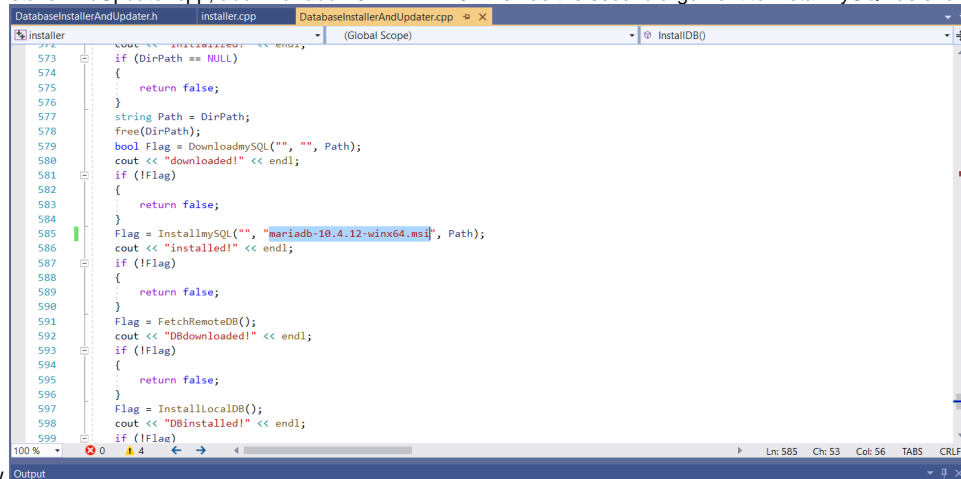


Database Installation Instructions

This documents the steps taken to install the database to a local machine. This is written for a Windows 64 bit OS.

Steps:

1. Clone the mfp426/2020CS499DB2 repository to your computer.
2. Install Visual Studios 2019 from <https://azure.microsoft.com/en-us/products/visual-studio/>. Make sure it installs MSVC Build tools 142 x86/64.
3. Open the installer.sln project from 2020CS499DB2 folder in Visual Studios
4. The download link for the MariaDB installer did not work. Manually download the installer and update the code to use the installer.
 - a. Download mariadb-10.4.12-winx64.msi from <https://downloads.mariadb.org/interstitial/mariadb-10.4.12/winx64-packages/mariadb-10.4.12-winx64.msi/from/http%3A//ftp.hosteurope.de/mirror/archive.mariadb.org/>
 - b. Copy the mariadb-10.4.12-winx64.msi to C:\Users\momsb\Documents\GitHub\2020CS499DB2\64\Debug\installer.exDBM folder
 - c. In DatabaseInstallerAndUpdater.cpp, add "mariadb-10.4.12-winx64.msi" as the second argument to InstallMySQL as shown in the



The screenshot shows a Visual Studio code editor window with the file DatabaseInstallerAndUpdater.cpp open. The code is in C++ and shows a function InstallMySQL. Line 585 is highlighted, showing the second argument to the InstallMySQL function: "mariadb-10.4.12-winx64.msi". The code also includes comments and other function calls like DownloadMySQL, FetchRemoteDB, and InstallLocalDB.

```
573     if (DirPath == NULL)
574     {
575         return false;
576     }
577     string Path = DirPath;
578     free(DirPath);
579     bool Flag = DownloadMySQL("", "", Path);
580     cout << "downloaded!" << endl;
581     if (!Flag)
582     {
583         return false;
584     }
585     Flag = InstallMySQL("", "mariadb-10.4.12-winx64.msi", Path);
586     cout << "installed!" << endl;
587     if (!Flag)
588     {
589         return false;
590     }
591     Flag = FetchRemoteDB();
592     cout << "DBdownloaded!" << endl;
593     if (!Flag)
594     {
595         return false;
596     }
597     Flag = InstallLocalDB();
598     cout << "DBinstalled!" << endl;
599     if (!Flag)
```

picture below.

5. Compile and run the program.

Potential Helpful Links

Links related to SQL and QT:

- connect QT to SQL: <https://doc.qt.io/qt-5/sql-connecting.html>
- example on executing SQL statements: <https://doc.qt.io/qt-5/sql-sqlstatements.html>
- SQL classes offered by QT: <https://doc.qt.io/qt-5/sql-model.html>

Math Formula SQL statements

meeting: 9-29-2020

1. Ask Austin/Patrick for understanding what is in db
2. Install db onto personal computer (should be smooth)
 - a. Take note to get it installed
3. Use ui code in google drive
4. Figure out how to pull data from local db
5. Use 090420
6. Note wt = weight
7. Tw = track width (middle of tire to other tire with middle)
8. Wheel base = front axel to back axel
9. D = distance in skid mark
10. F = coefficient of friction

/* I think this is the correct **weight** */

left and right weight depends on weight distribution

left side weight = driver left + rear left

right side weight = passenger right + rear right

changes when you add new weight to the car

- dont have to worry about the distribution

wt = SELECT model_weight_kg FROM vehicle_specs_merge WHERE model_make_id='<INSERT MAKE ID>' AND model_name='<INSERT NAME>' AND model_trim='<INSERT TRIM>' AND model_year='<INSERT YEAR>';

/* not sure which track width he wants for **track width** so here are both */

front track width

fTrackWidth: SELECT fTrackWidth FROM vehicle_specs WHERE model_make_id='<INSERT MAKE ID>' AND model_name='<INSERT NAME>' AND model_trim='<INSERT TRIM>' AND model_year='<INSERT YEAR>';

rear track width WONT NEED

~~rTrackWidth: SELECT rTrackWidth FROM vehicle_specs WHERE model_make_id='<INSERT MAKE ID>' AND model_name='<INSERT NAME>' AND model_trim='<INSERT TRIM>' AND model_year='<INSERT YEAR>';~~


/* **weight ratio** I HAVE IN RED WHAT I'M NOT SURE HOW TO GET*/

SELECT weightDistribution FROM vehicle_specs_additional WHERE Make='<INSERT MAKE ID>' AND Model='<INSERT NAME>' AND **model_trim='<INSERT TRIM>'** AND Year='<INSERT YEAR>';

/***Wheelbase** */

SELECT Wheelbase FROM vehicle_specs_additional WHERE Make='<INSERT MAKE ID>' AND Model='<INSERT NAME>' AND **model_trim='<INSERT TRIM>'** AND Year='<INSERT YEAR>';

Center of Mass Width



Center of Mass Width

Formula Inputs Full Results **Formula In Use**

$$c = \frac{wr \times tw}{wt}$$
$$d = \frac{wl \times tw}{wt}$$

Center of Mass Width

Formula Inputs

CENTER OF MASS

Left Side Weight: lb/kg

Right Side Weight: lb/kg

Track Width: in/cm

Total Veh Weight: lb/kg

[View the Veh Specs Window](#)

[Calc](#) [Print](#) [Preview](#) [Copy](#)

[Open](#) [Show](#) [Help](#) [Close](#)

Full Results

INCREMENTS

Current:

Increment By:

Steps:

Start Point:

Increment

☒ Left Side Wt

☐ Track Width

☐ Total Veh Wt

Formula In Use

RESULTS

Cntr of Mass L (in): **0**

Cntr of Mass R (in): **0**

Find a vehicle Center of Mass Width using weight and track width.

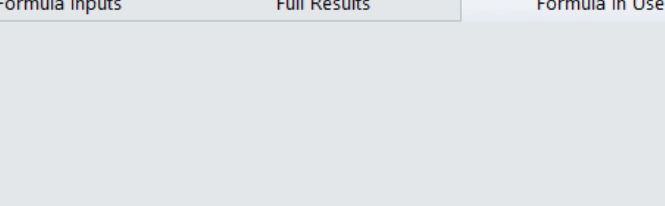
NOTES

with the vehicle weights to calculate the side-to-side center of mass location.

This formula is used on the initial screen

Horizontal Center of Mass

This formula is used on the initial screen with the vehicle



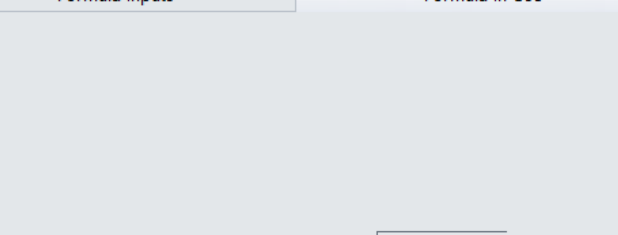
Horizontal Center of Mass

Formula Inputs Full Results **Formula In Use**

$$CM = \frac{wf \times wb}{wt}$$

Center of Mass Height

This formula is used on the initial screen with the vehicle weights



Center of Mass Height

Formula Inputs

Formula In Use

$$H = h2 + \frac{(ww - wf) \times l \times \sqrt{l^2 - (h1 - h2)^2}}{wt \times (h1 - h2)}$$

In order to calculate the specific weights for the vehicle, we start by importing the overall vehicle weight and display it on the UI. The front and rear axle weights are calculated by using the weight ratio (found in the vehicle specs) and multiplying that by the overall weight. For example: if the weight ratio is 60 /40, this would mean that 60% of the weight is on the front axle and 40% on the rear. For a 4000lbs car, $60\% \times 4000 = 2400\text{lbs}$ on the front axle, and $40\% \times 4000\text{lbs} = 1600\text{lbs}$ on the rear. These values will then be displayed in the boxes for "Axle Weight." Now using the calculated axle weight, we will divide by 2 to acquire the individual wheel weight for each axle. IE: $2400\text{lbs} \div 2 = 1200$ on each front wheel. This would be displayed in the front tire boxes. Following the same logic, 600lbs would be assigned to the two rear axle tire boxes.



Minimum Speed

Formula Inputs Full Results Formula In Use

MINIMUM SPEED

Skid Distance: ft/m

Accel/Drag f:

Calc Print Preview Copy

Open Show Help Close

INCREMENTS

Current:

Increment By:

Steps:

Start Point:

Increment

☒ Distance

☐ Drag Factor

RESULTS

Speed (mph): **0**

Velocity (fps): **0**

Find a Minimum Speed with a Skid Distance and Drag Factor.

NOTES

Minimum Speed Calculations

The only additional calculation with this Minimum Speed calculation is to include the Velocity (fps) in addition to the Speed that we calculate. This is a simple conversion of $\text{Speed} \times 1.466 = \text{Velocity (fps)}$.

Creating a Standalone Executable

I followed the steps given here: https://github.com/jlre249/CS499_Project/wiki/Creating-a-Standalone-Executable

Bugs

This documents bugs found during reviews of the application.

- If the user clicks either "Get vehicle Info" button without entering the vehicle make, model, and year, the application crashes
- If the user selects a year, but then goes back to "Select a year," the model list is still populated and the user can still select a model
- The secondary search adds an extra 0 to some results making them 10 times larger. This was to fix inconsistencies with the vehicle_specs_merged table.
- Height of front wheels and height of rear wheels raised were in the wrong order in the CoM Ht calculation causing miscalculations. Fixed 11-4-2020.
- Negative numbers are always rounded towards 0 instead of toward .5 or a more negative number. Fixed 11-5-2020 by adding abs()).