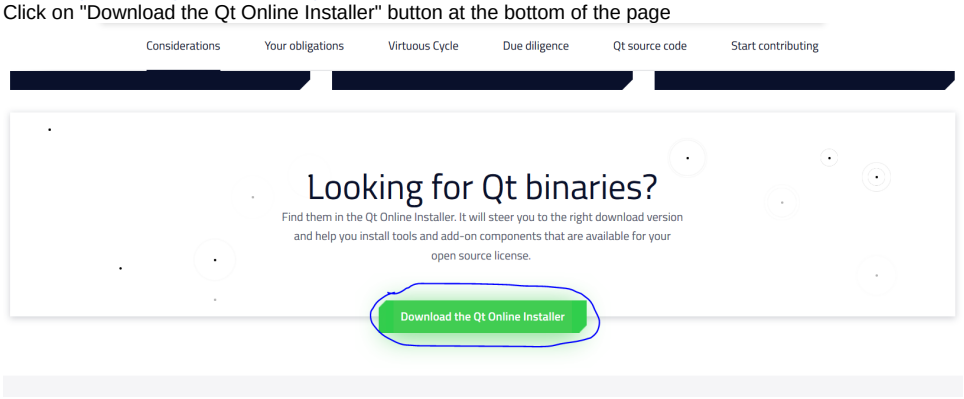
Qt and MySQL Installation and Setup

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.

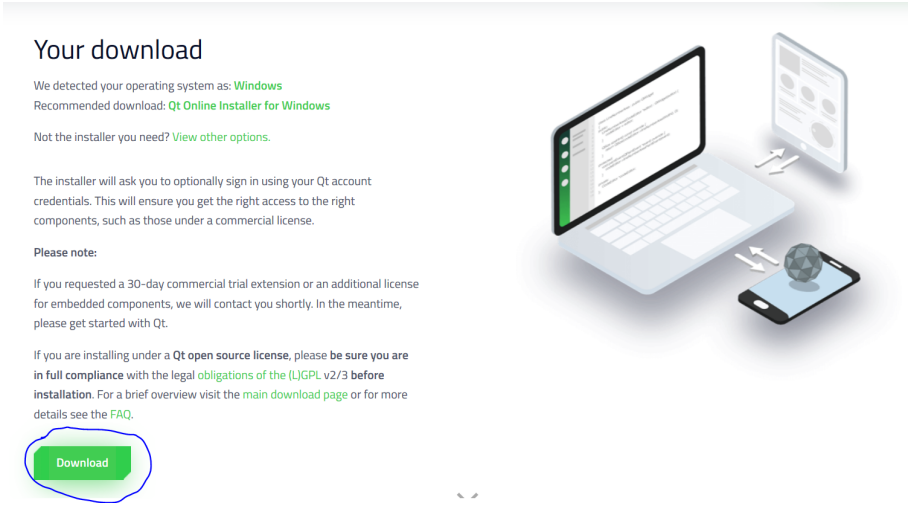
QT Installation

Steps:

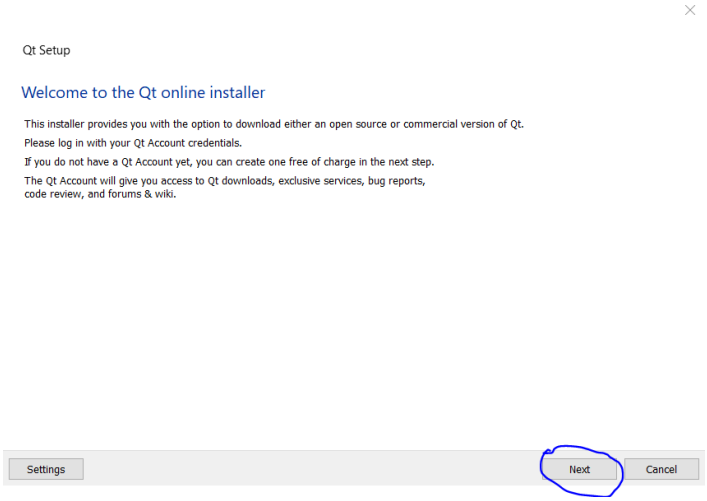
1. Go to https://www.qt.io/download-open-source
2. Click on "Download the Qt Online Installer" button at the bottom of the page



3. Scroll down and click the "Download" button

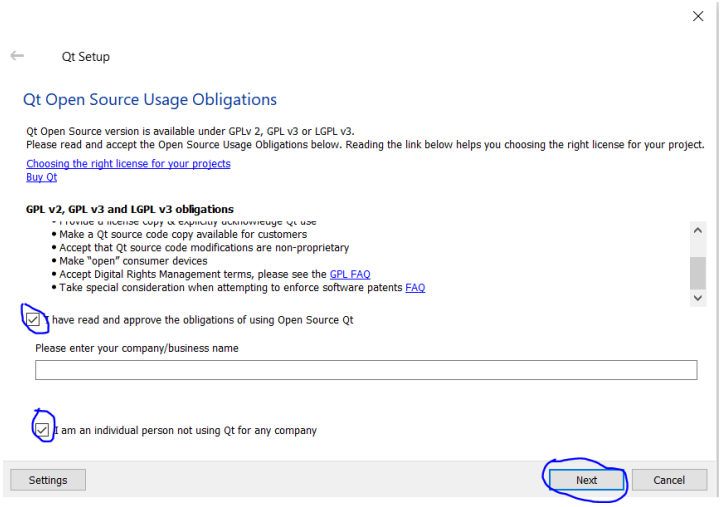


4. Run the downloaded .exe file
5. The installer should pop up. Select "Next".

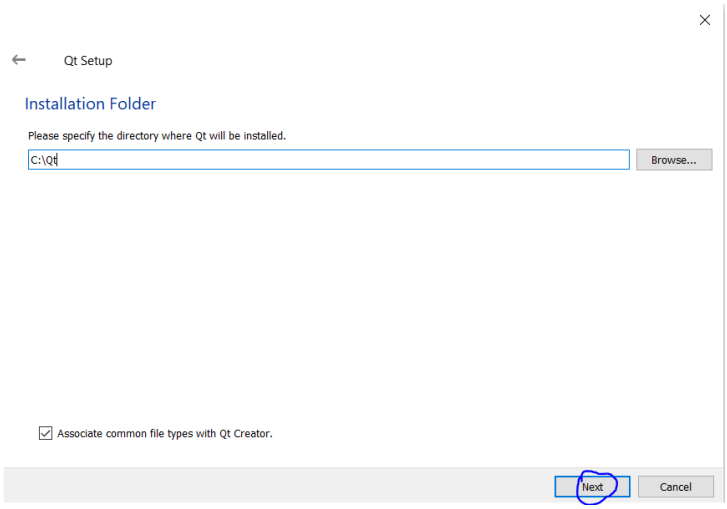


6. Enter your login information or create an account. Select "Next".
    a. If you created a new account, you will need to check your email for a verification link.
    b. At the link, you will need to enter your information and select "Individual User"
    c. After verifying your account, select "Next"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

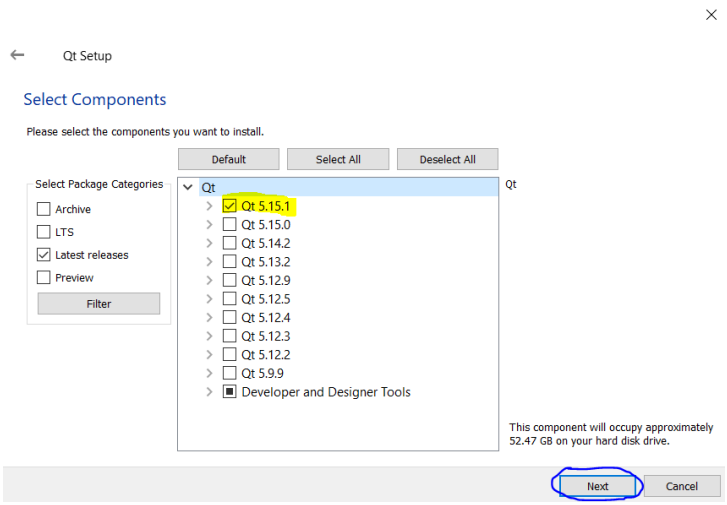
7. Make sure both check boxes are checked. Then select "Next"



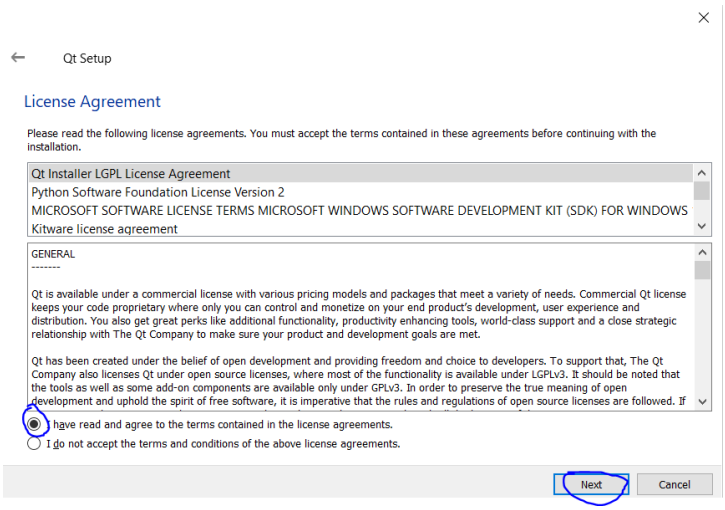
8. A welcome screen should appear. Select "Next".
9. After some initial setup, the installer will prompt you to select whether or not to share usage statistics. Select an option, and click "Next"
10. Leave the default settings for the installation folder and select "Next"



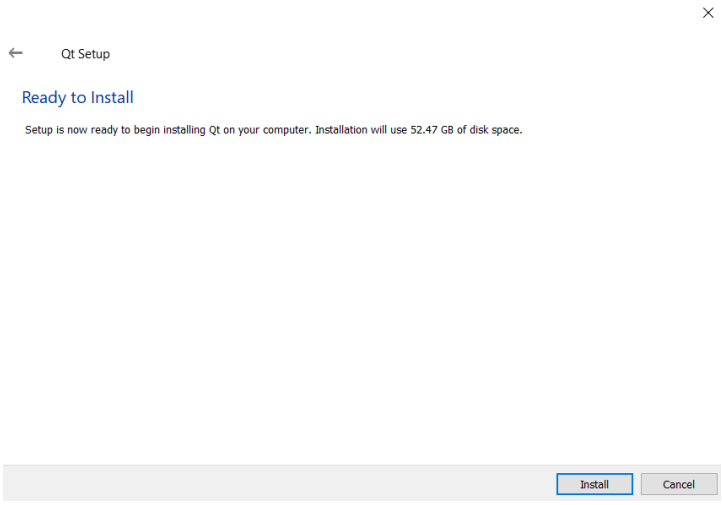
11. Select version 5.15.xx (I used 5.15.1). Click "Next"



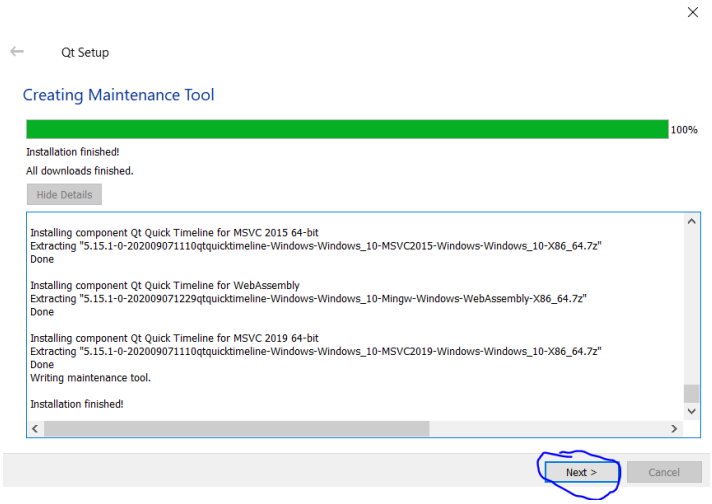
12. Accept the license agreement and click "Next".



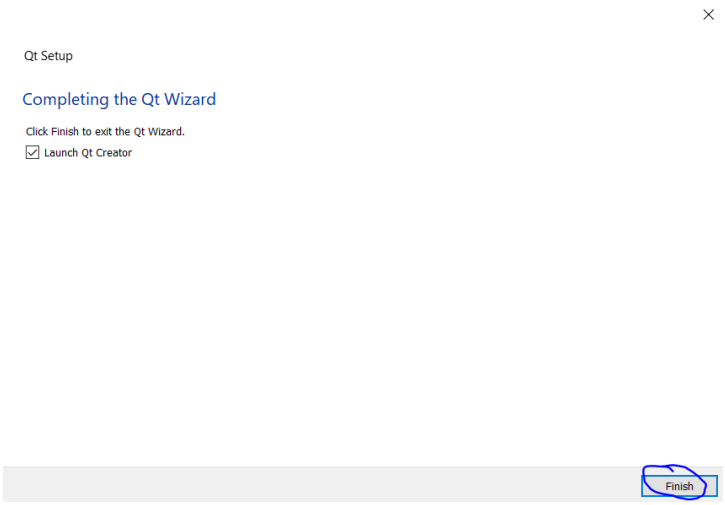
13. You can select a different location in the start menu for the shortcuts or just click "Next".
14. Click "Install".



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

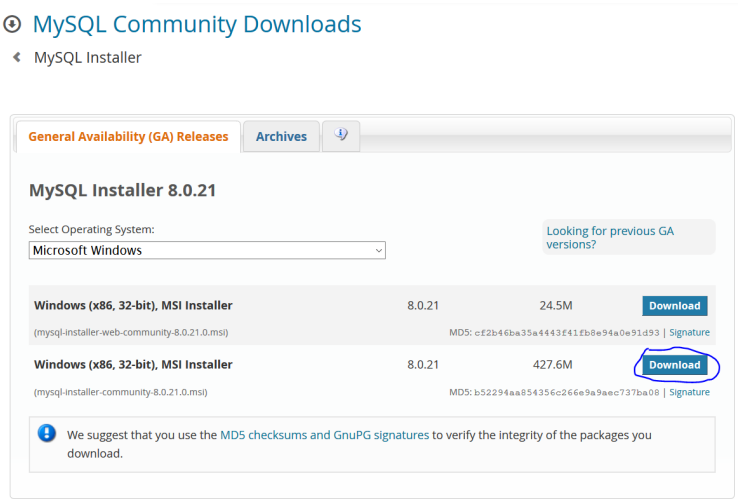


16. Click "Finish" and launch Qt Creator.



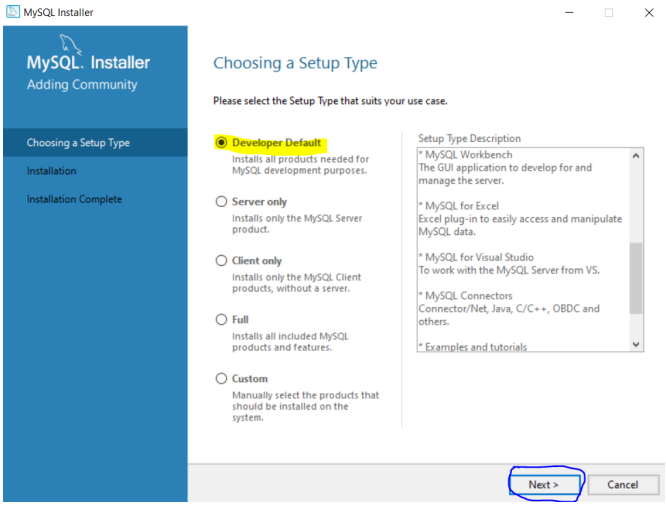
MySQL Installation

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

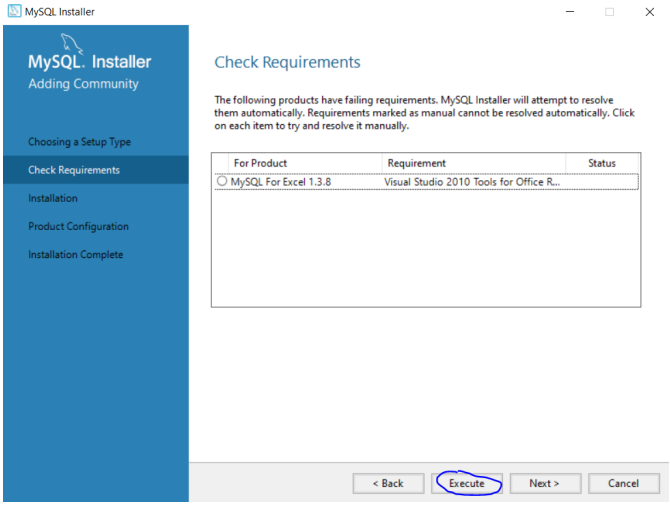


2. You can choose to create an account, or just click "No thanks, just start my download."

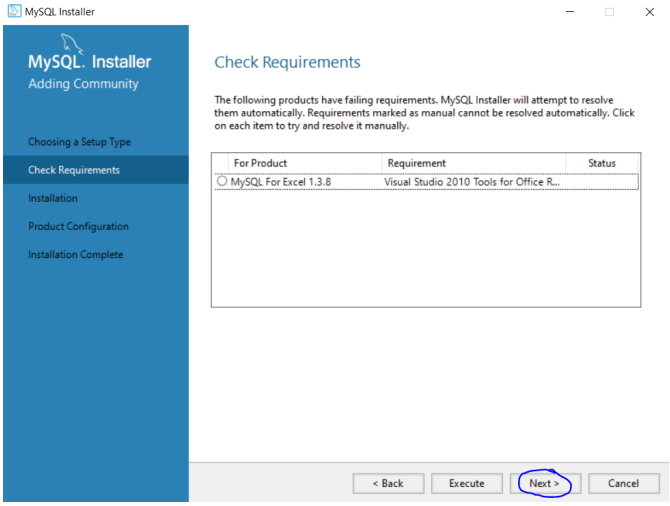
3. After the installer downloads, run the installer.
4. Make sure "Developer Default" is selected, then click "Next".



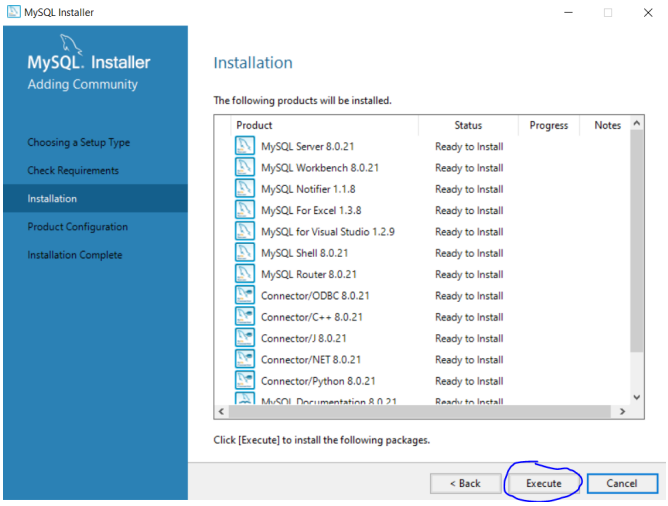
5. Select any issues present and click "Execute" to fix them



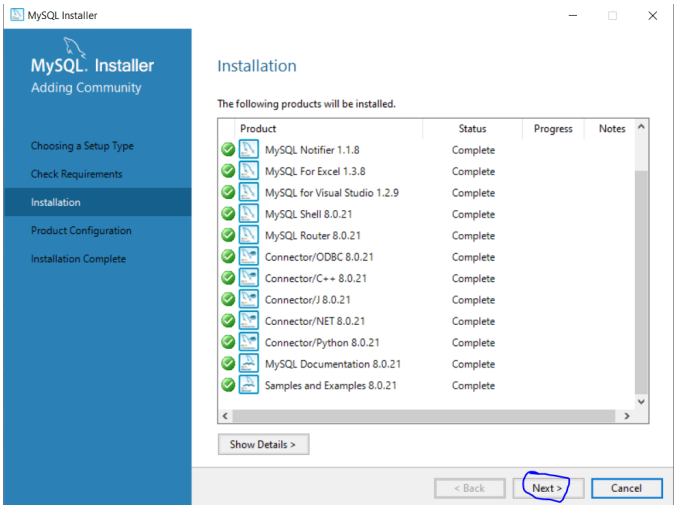
6. After issues are resolved, click "Next".



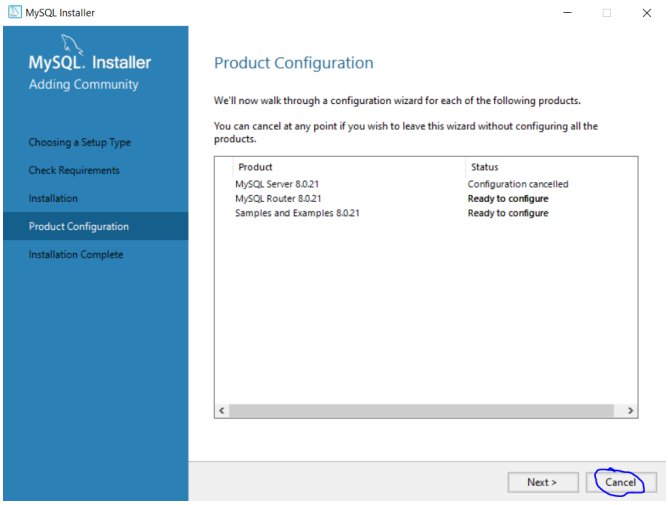
7. Click "Execute".



8. After Installation finishes, click "Next".



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



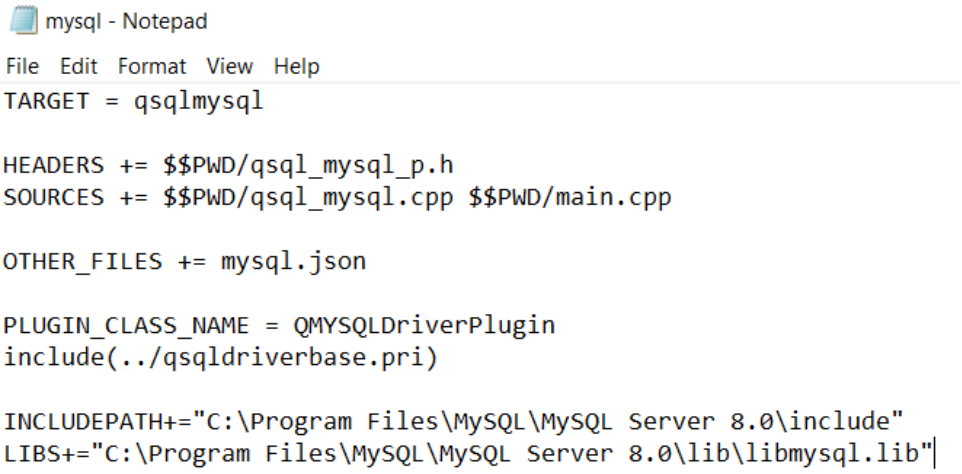
Building QMySQL driver

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\sqldrivers\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"

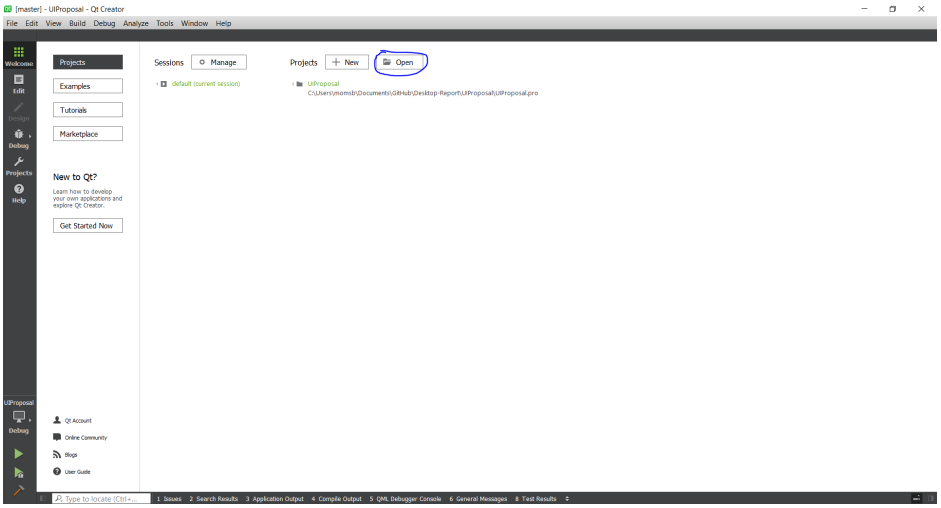
4. Your mysql.pro file should now look like this:



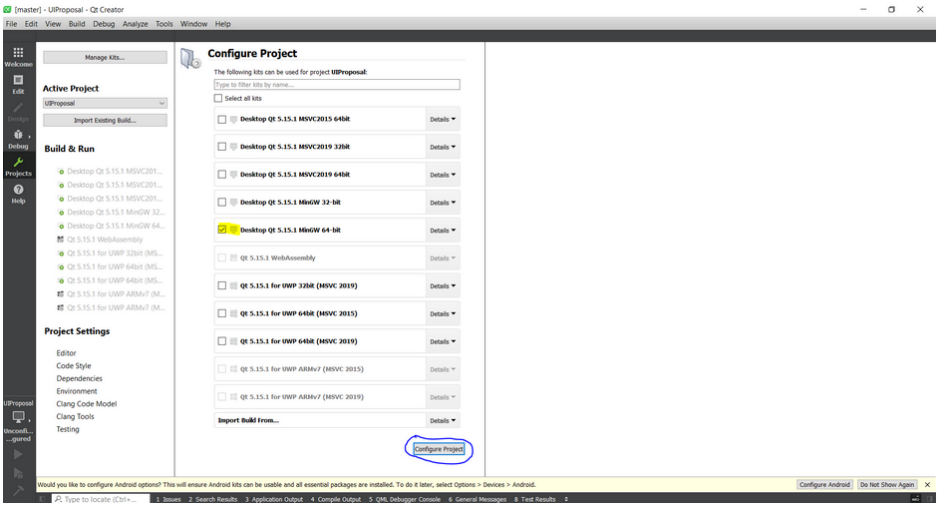
5. Save mysql.pro and close it.
6. Open a command console. (I used Command Prompt)
7. Run the following sequence of commands:
    a. C:\QT\5.15.1\mingw81_64\bin\qtenv2.bat
    b. cd C:\Qt\5.15.1\Src\qtbase\src\plugins\sqldrivers
    c. qmake sqldrivers.pro
    d. cd mysql
    e. qmake mysql.pro
    f. 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

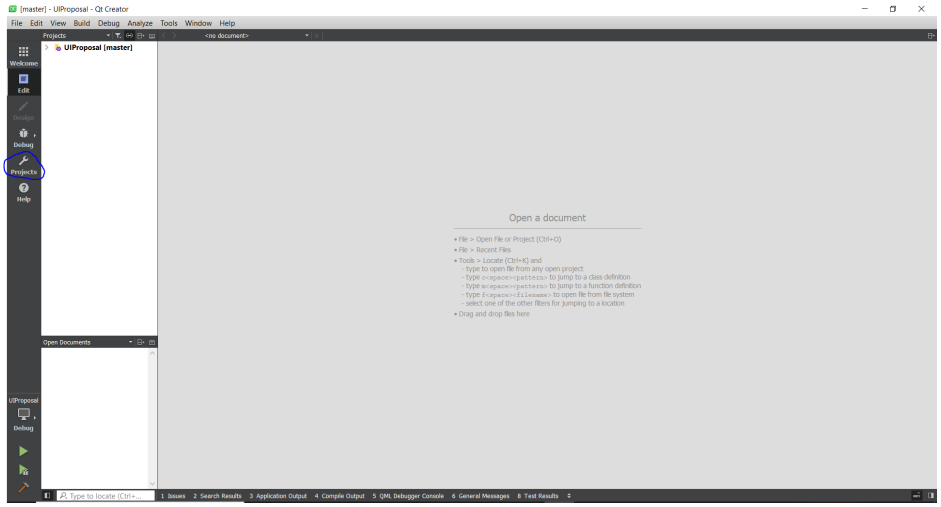
1. Clone the "mfp426/MathCalc" repository to your computer.
2. In Qt Creator, click "Open" to open a project.



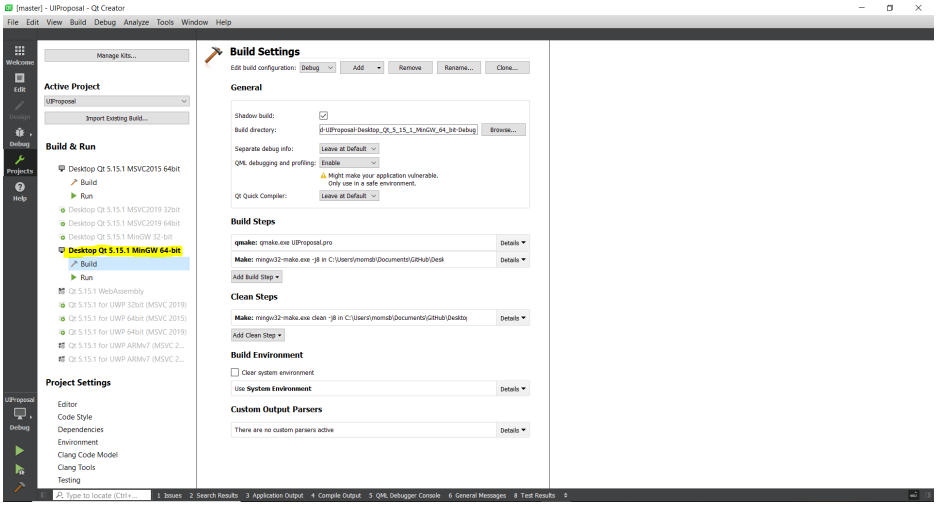
3. In the file browser, navigate to where you cloned the repository and select the Qt project file "FormulasModal".
4. Make sure MinGW 64 bit version 5.15.1 is checked since that is the kit we built MySQL for. Click "Configure Project".



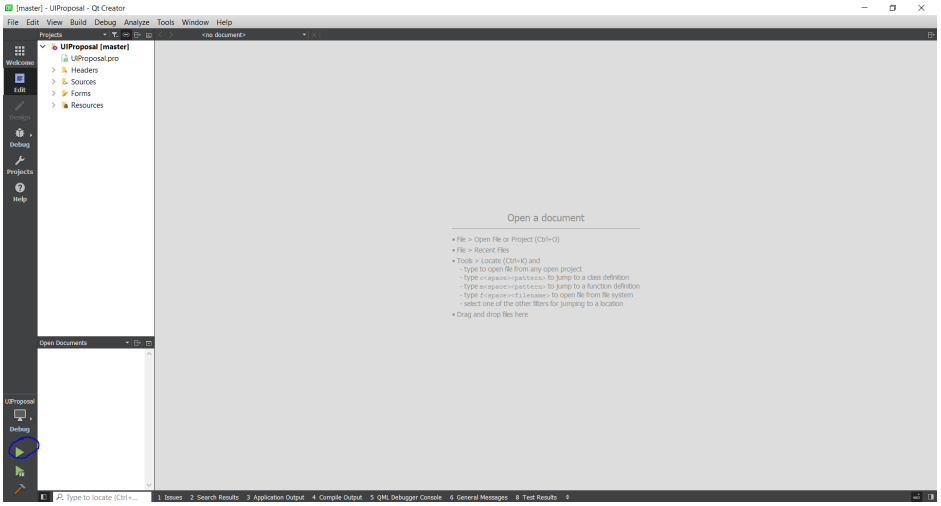
a. If you did not originally select MinGW 64 bit, you can change the kit you are using. Click on "Projects" on the left side.



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.

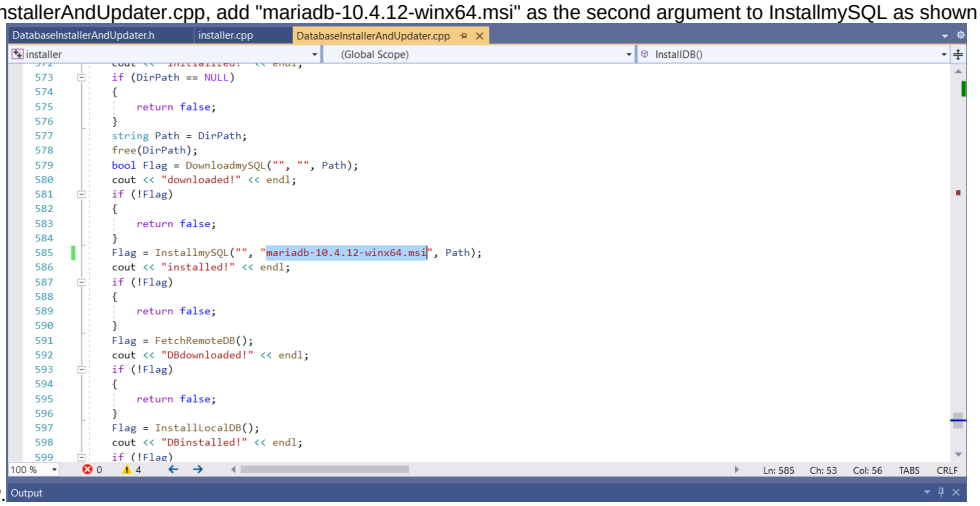

[master] - UIProposal - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Projects
<no document>
UIProposal [master]
UIProposal.pro
Headers
Sources
Forms
Resources
Welcome
Edit
Design
Debug
Projects
Help
Open a document
File > Open File or Project (Ctrl+O)
File > Recent Files
Tools > Locate (Ctrl+K) and
- type to open file from any open project
- type c:<space><pattern> to jump to a class definition
- type m:<space><pattern> to jump to a function definition
- type f:<space><filename> to open file from file system
- select one of the other filters for jumping to a location
Drag and drop files here
Open Documents
UIProposal
Debug
Type to locate (Ctrl+...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

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\x64\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 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*/

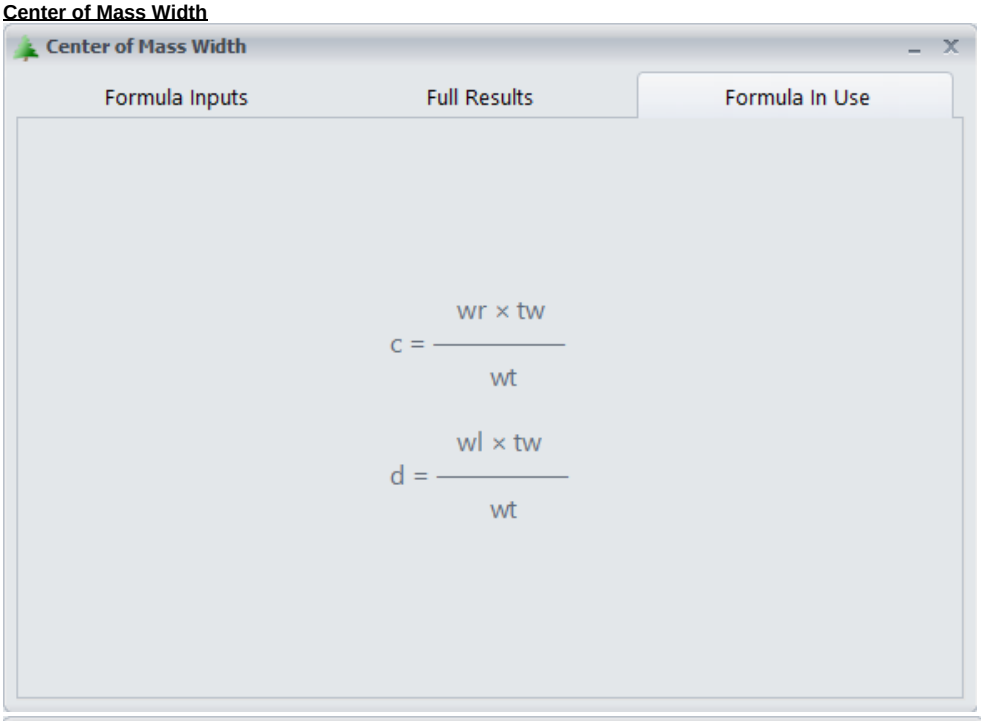
SELECT weightDistribution FROM vehicle_specs_additional WHERE Make='<INSERT MAKE ID>' AND Model='<INSERT NAME>' AND **model_trim='<INSERT TRIM>'** AND Year='<INSERT YEAR>';

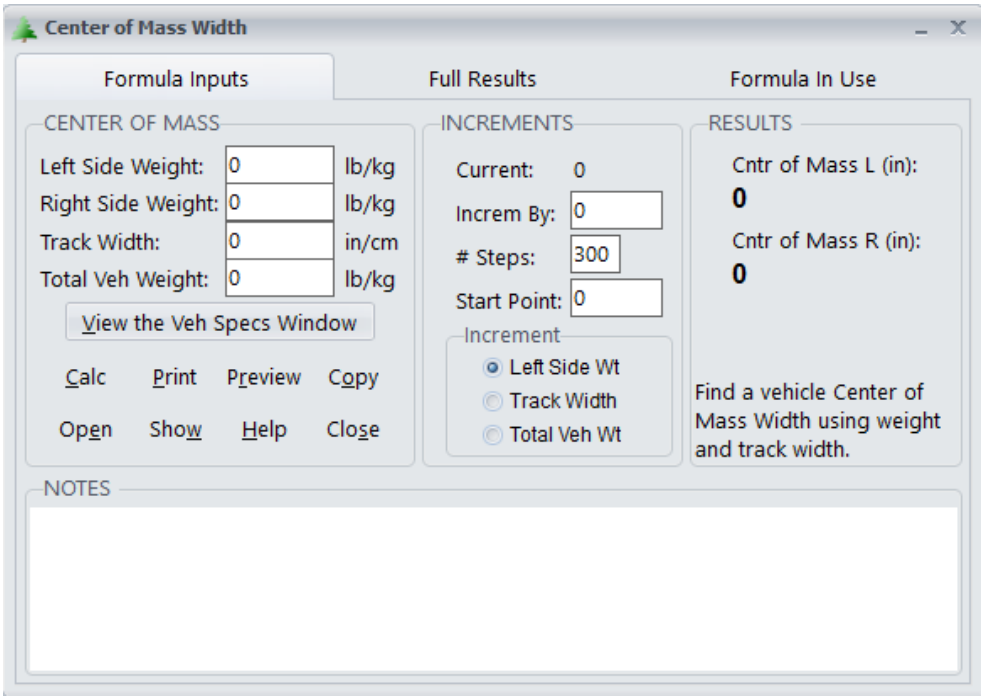
/***Wheelbase** */

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

Math Formulas

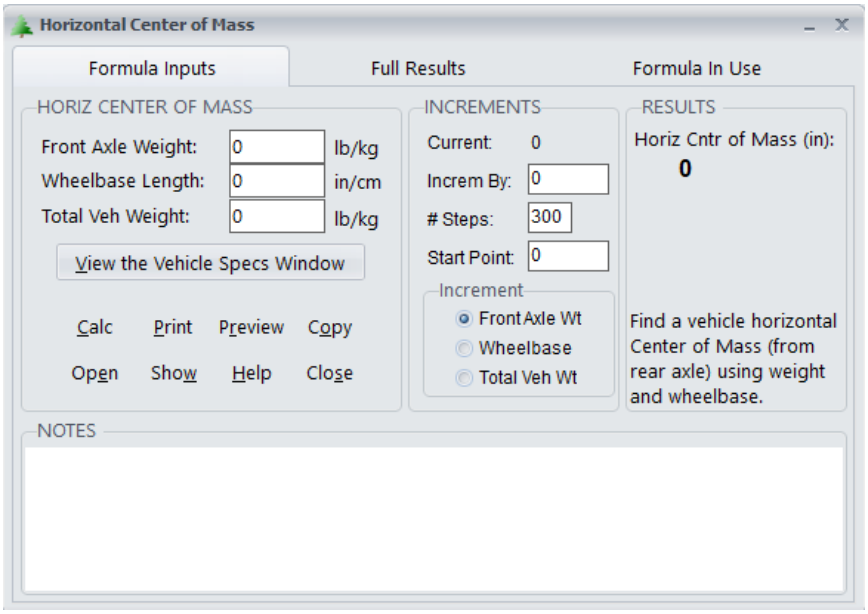
Center of Mass Width



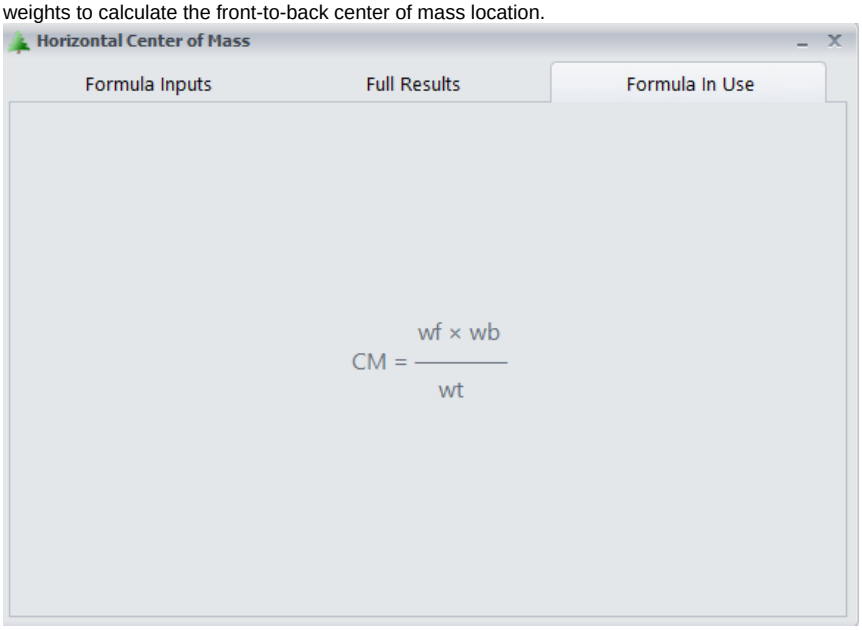


This formula is used on the initial screen with the vehicle weights to calculate the side-to-side center of mass location.

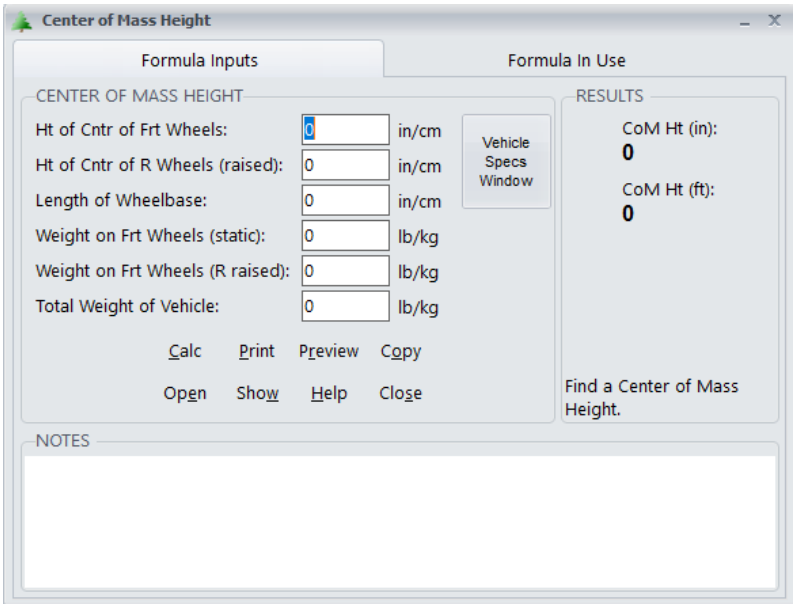
Horizontal Center of Mass



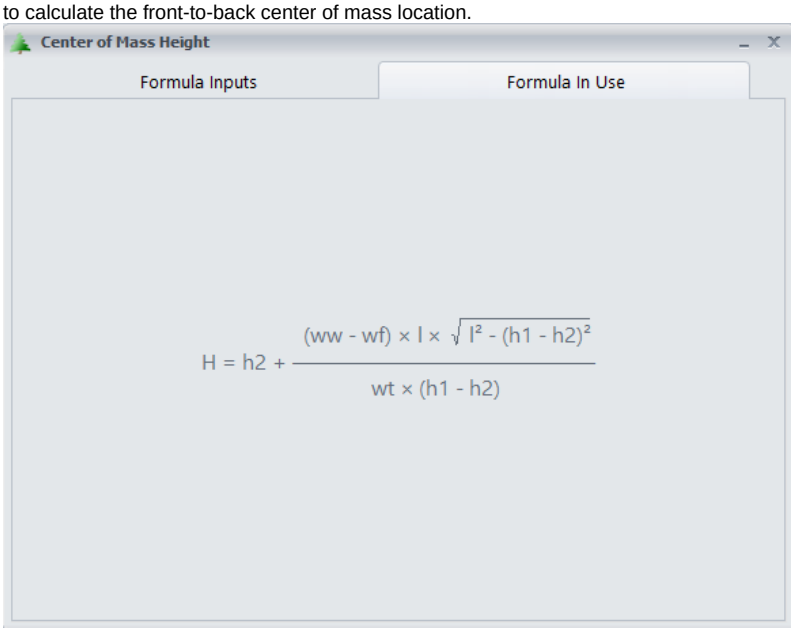
This formula is used on the initial screen with the vehicle weights to calculate the front-to-back center of mass location.



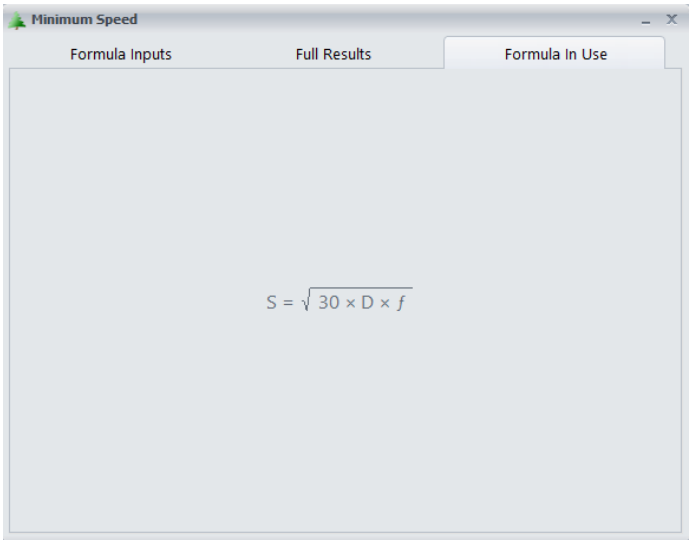
Center of Mass Height

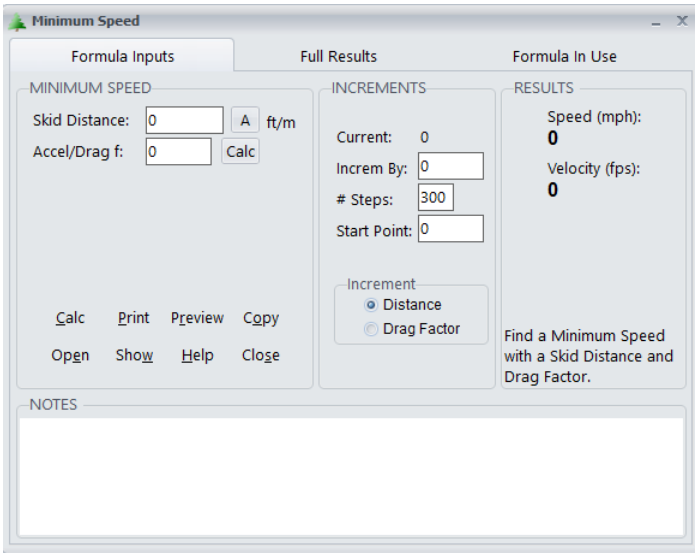


This formula is used on the initial screen with the vehicle weights to calculate the front-to-back center of mass location.



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% × 4000 = 2400lbs on the front axle, and 40% × 4000lbs = 1600lbs 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: 2400lbs ÷ 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 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 Speed × 1.466 = 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().