Introduction
News
Download
Contact

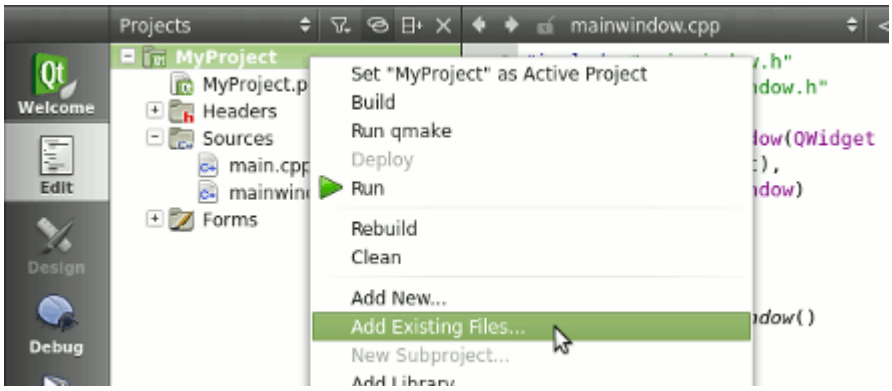
# Setting up QCustomPlot

Getting QCustomPlot to work with your application is very easy:

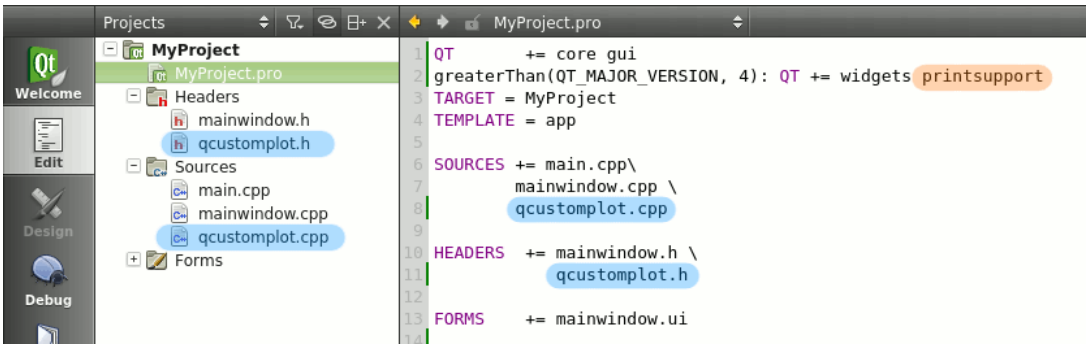
- Get the latest version of QCustomPlot from the download section.
- Use the *qcustomplot.h* and *qcustomplot.cpp* file like any other ordinary class file

## For QtCreator users

Right click on the root entry of your project in the left sidebar and choose *Add Existing Files...*



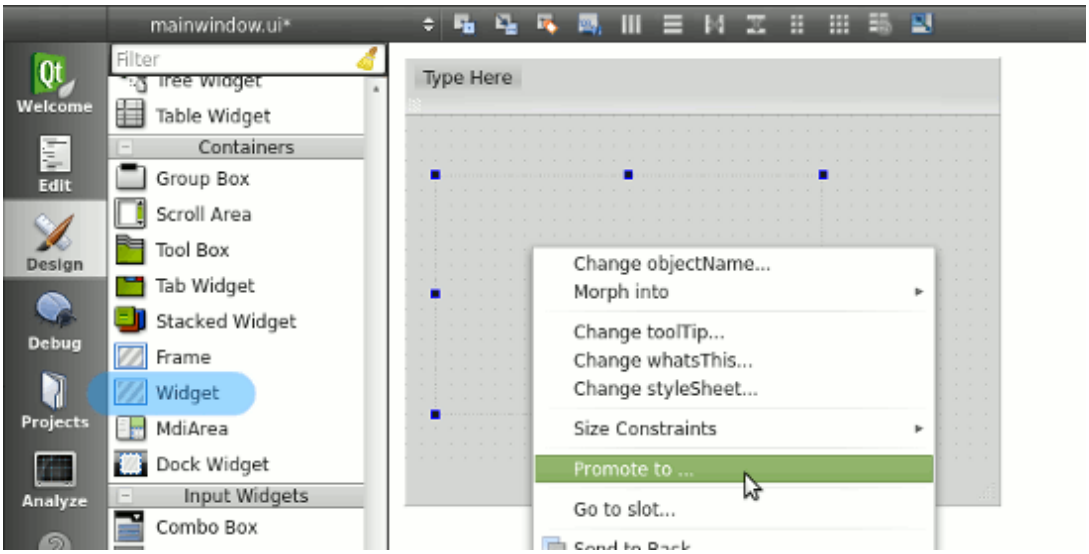

In the appearing file dialog, select the *qcustomplot.h* and *qcustomplot.cpp* files, to add them to your project. If this is done, your project structure and *.pro* file should look something like this:



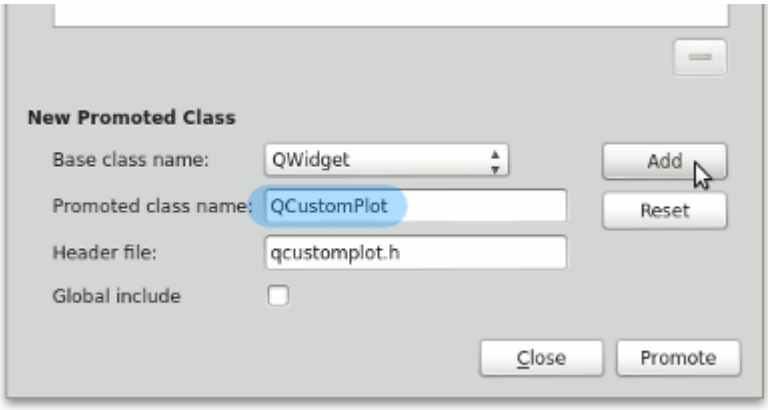

If you are using Qt version 5.0 upwards, you need to add `printsupport` to the `QT` variable in your *.pro* file. In the case shown above, this is done after a `greaterThan(QT_MAJOR_VERSION, 4)` conditional. This makes sure the `printsupport` (and `widgets`) is not added when using older Qt versions.

The project is now ready to use QCustomPlot. Place a regular QWidget on your form in the desired location. Right click on it and hit Promote to...

Introduction
News
Download
Contact




In the appearing dialog, enter `QCustomPlot` in the input field next to *Promoted class name*. The input next to *Header file* should automatically fill with the correct `qcustomplot.h` value. Hit *Add* to add QCustomPlot to the promoted classes list and finally hit *Promote* to turn the QWidget on your form into a QCustomPlot.




You won't see any immediate visual changes in QtCreator (or QtDesigner), but while running the application, you will see an empty plot with axes and grid lines.

## Troubleshooting

**Compilation aborts with "GL/gl.h: No such file or directory" or "cannot find -lGL"**
On a GNU/Linux system, make sure you have the packages *mesa-common-dev*, *libgl1-mesa-dev*, and *libglu1-mesa-dev* installed, e.g. by entering the following into a bash terminal:

```
sudo apt-get install mesa-common-dev libgl1-mesa-dev libglu1-mesa-dev
```

**Linking aborts with "Undefined reference to QPrinter(...)"**
Make sure you have added the module *printsupport* to the `QT` variable in your project file, as described above.

## Using QCustomPlot as shared library .so/.dll

Using a shared library means to not include the .h/.cpp file into your project, but linking with an external *qcustomplot.so* (GNU/Linux) or *qcustomplot.dll* (MSWindows) file. QCustomPlot is ready to be built as a shared library by setting the compiler define `QCUSTOMPLOT_COMPILE_LIBRARY`. To use the shared library in your application, set the define `QCUSTOMPLOT_USE_LIBRARY` before including the QCustomPlot header.

The *sharedlib* package in the download section provides two projects that demonstrate this: one compiles the shared QCustomPlot library and the other uses the shared library. This should quickly get you started using QCustomPlot as a shared library.

Introduction
News
Download
Contact

## Running the examples

The *QCustomPlot.tar.gz* package in the download section contains the example projects ready to be compiled. Just extract the whole package to a new directory, navigate inside the example directories and run `qmake; make` . Alternatively you can open the *.pro* files in QtCreator and work with the examples from there.