Inhaltsverzeichnis

1. QcustomPlot-

http://www.qcustomplot.com/index.php/introduction

″**QCustomPlot** is a Qt C++ widget for plotting and data visualization.″

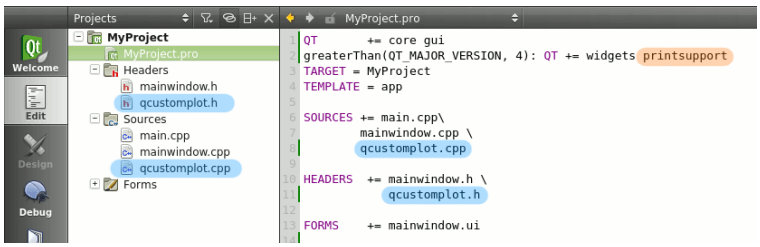
This plotting library focuses on making good looking, publication quality 2D plots, graphs and charts, as well as offering high performance for realtime visualization applications. Have a look at the Setting Up and the Basic Plotting tutorials to get started.

QCustomPlot can export to various formats such as vectorized PDF files and rasterized images like PNG, JPG and BMP. QCustomPlot is the solution for displaying of realtime data inside the application as well as producing high quality plots for other media.

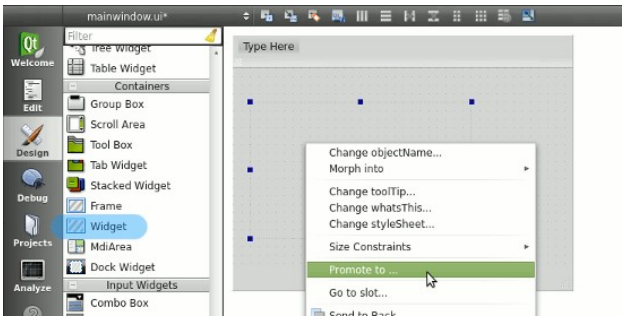
1.1. Projekt: QCustomPlot

* Set-up: http://www.qcustomplot.com/index.php/tutorials/settingup

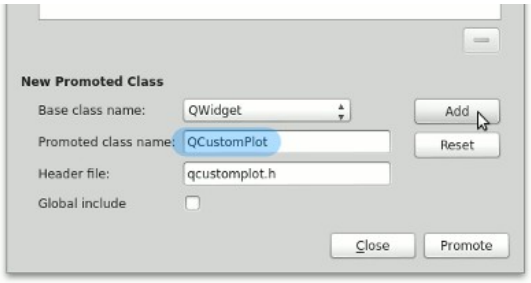
1. Get the latest version of QCustomPlot from the download section.
2. Use the *qcustomplot.h* and *qcustomplot.cpp* file like any other ordinary class file
3. copy *qcustomplot.h* and *qcustomplot.cpp* into your project folder
4. qtcreator: select project → re.Maus→ add existing files



5. .pro file: add printsupport
6. Designer: Neues Widget reinziehen und promote to



7. Dialog-Fenster: QCustomplot eingeben



Wenn customPlot das gerade erzeugte Widget ist, dann kann man eine quadratische Funktion zeichnen, mit:

```
// generate some data:
QVector<double> x(101), y(101); // initialize with entries 0..100

for (int i=0; i<101; ++i)
{
  x[i] = i/50.0 - 1; // x goes from -1 to 1
  y[i] = x[i]*x[i]; // let's plot a quadratic function
}

// create graph and assign data to it:
customPlot->addGraph();
customPlot->graph(0)->setData(x, y);

// give the axes some labels:
customPlot->xAxis->setLabel("x");
customPlot->yAxis->setLabel("y");

// set axes ranges, so we see all data:
customPlot->xAxis->setRange(-1, 1);
customPlot->yAxis->setRange(0, 1);
customPlot->replot();
```