C++ Qwt - Plotting data from a vector

I'm attempting to plot a graph based on data that I have obtained and stored inside a vector, but, I cannot seem to find any tutorials or references out there and give me any indication to what I need to do. So here is my code:

```
    public:

     Plotter() {

     }
};



int main( int argc, char **argv )
{

   QApplication app(argc, argv);

   //Plotter *d_plot = new Plotter();

    Plotter* d_plot = new Plotter();

   d_plot->setTitle("DEMO");
   d_plot->setCanvasBackground(Qt::white);
   d_plot->setAxisScale( QwtPlot::yLeft, 0.1, 50.0 );
   d_plot->setAxisScale(QwtPlot::yRight, 0.1, 50.00);

   // PLOT THE DATA
   std::vector<double> data;
   data.push_back(1.03);
   data.push_back(13.12);
   //....

   d_plot->resize( 600, 400 );
   d_plot->show();


   return app.exec();
}
```

Could anyone give me any ideas to what function I could use to allow me to plot the data?

Thanks

c++ qt qwt

asked Sep 5 '13 at 15:45

Phorce

878 1 11 36

I am facing this problem now too. Just wish to ask, I noticed your question have only one data vector, and the answers given consist of 2 data vectors. What is the other vector suppose to be? Hope you can help. Thanks (: – rockinfresh Jan 19 at 11:12

add a comment

2 Answers

Check the QwtPlot docs: normally you create a `QwtPlotCurve`, use `QwtPlotCurve::setSamples` to get the data in it then `QwtPlotCurve::attach` to get the data drawn.

Should be something like this:

```
std::vector<double> x;
std::vector<double> y;
//fill x and y vectors
//make sure they're the same size etc
QwtPlotCurve curve( "Foo" );
//or use &x[ 0 ] or &(*x.first()) pre-C++11
cure.setSamples( x.data(), y.data(), (int) x.size() );
curve.attach( &plot );
```

http://qwt.sourceforge.net/class_qwt_plot_curve.html

http://qwt.sourceforge.net/class_qwt_plot.html

answered Sep 5 '13 at 15:59

stijn
14k 4 34 73

Thanks for the reply. Where do you get `&plot` from? Because, if I used `&d_plot` as done in my o/p it gives me an error: `error: cannot initialize a parameter of type 'QwtPlot *' with an rvalue of type 'Plotter **' my_curve.attach(&d_plot);` any ideas? Thanks for your help :) – Phorce Sep 5 '13 at 16:25

for your case, just pass d_plot since it's a pointer already – stijn Sep 5 '13 at 16:28

Thanks for the reply. I did that, all compiled now.. However, when I run the application, there is nothing showing? Do I need to set the Pen colour? Thanks :) – Phorce Sep 5 '13 at 16:30

got it :D thanks so much – Phorce Sep 5 '13 at 16:34

btw see here: stackoverflow.com/questions/5374287/qwt-graph-plot-example you can always check those examples as a reference – stijn Sep 5 '13 at 16:35

add a comment

One way would be to attach a curve to your plot, i.e.:

```
QwtPlotCurve myCurve;
myCurve->attach(&d_plot);
```

You could then use (in a member function, or wherever you need) the function `QwtPlotCurve::setRawSample` which has the following pretty much explanatory signature:

```
void QwtPlotCurve::setRawSample(const double* xData, const double* yData, int
size);
```

Set your data with it and then call `replot()` to refresh the plot. It means you must have also a vector for

the x values.

The code would look like this:

```
int main( int argc, char **argv )
{
   //...
   Plotter* d_plot = new Plotter();

   //Plot config

   // PLOT THE DATA
   std::vector<double> data_y;
   data_y.push_back(1.03);
   data_y.push_back(13.12);
   std::vector<double> data_x;
   data_x.push_back(1.0);
   data_x.push_back(2.0);
   //....
   myCurve->setRawSample(data_x.data(),data_y.data(),data_y.size());
   d_plot->resize( 600, 400 );
   d_plot->replot();
   d_plot->show();
   //...
}
```

I'd suggest you study the Qwt doc about curve

edited Sep 5 '13 at 16:05

answered Sep 5 '13 at 16:00

JBL
5,092 2 14 43

add a comment

Not the answer you're looking for? Browse other questions tagged c++ qt qwt or ask your own question.