

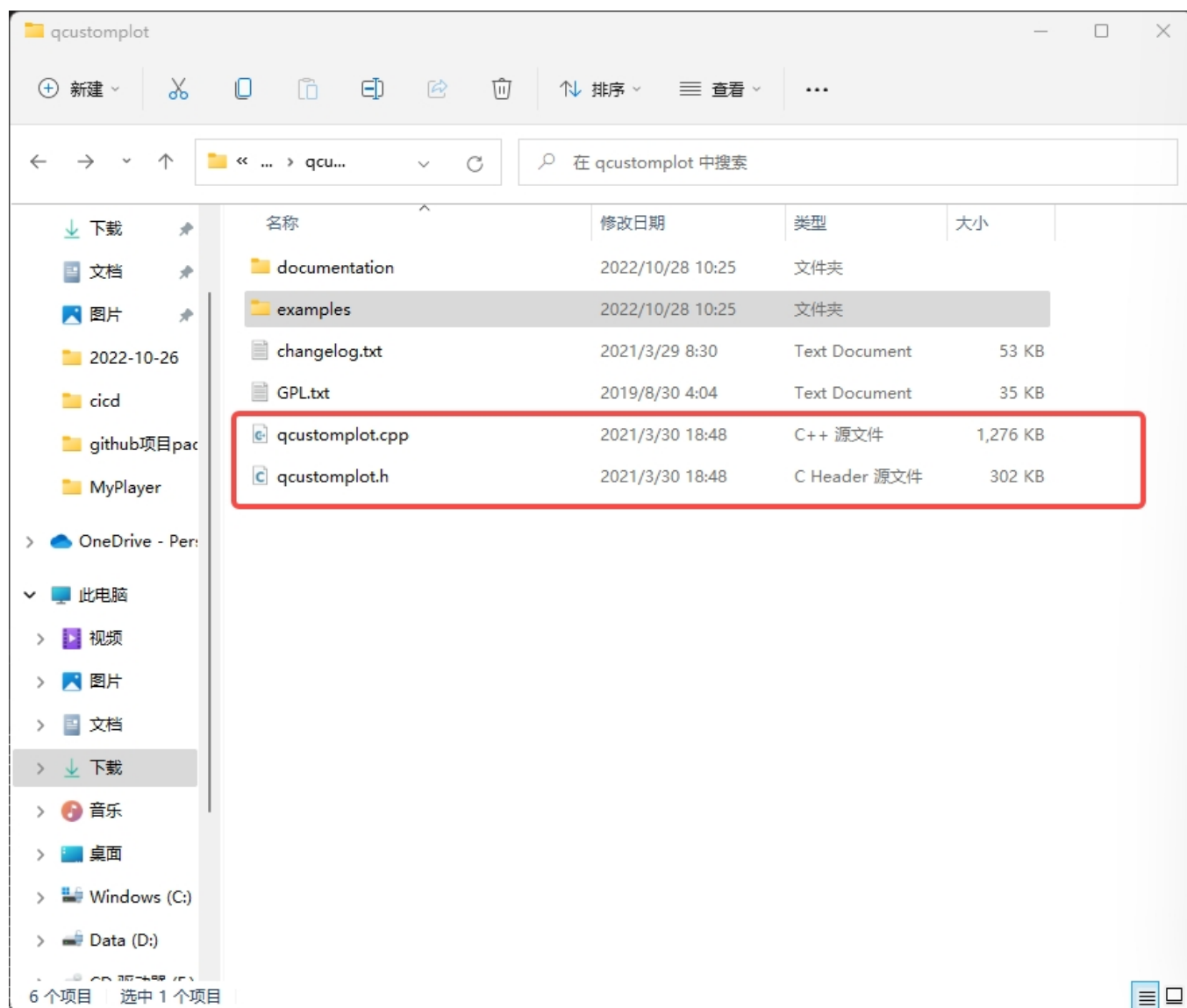
# qcustomplot简介

qcustomplot是一个用于绘图和数据可视化的第三方Qt构件，除qt原版依赖之外不需要其他的第三方库。性能比qchart更优秀，并且基于gpl开源协议对社区开源。

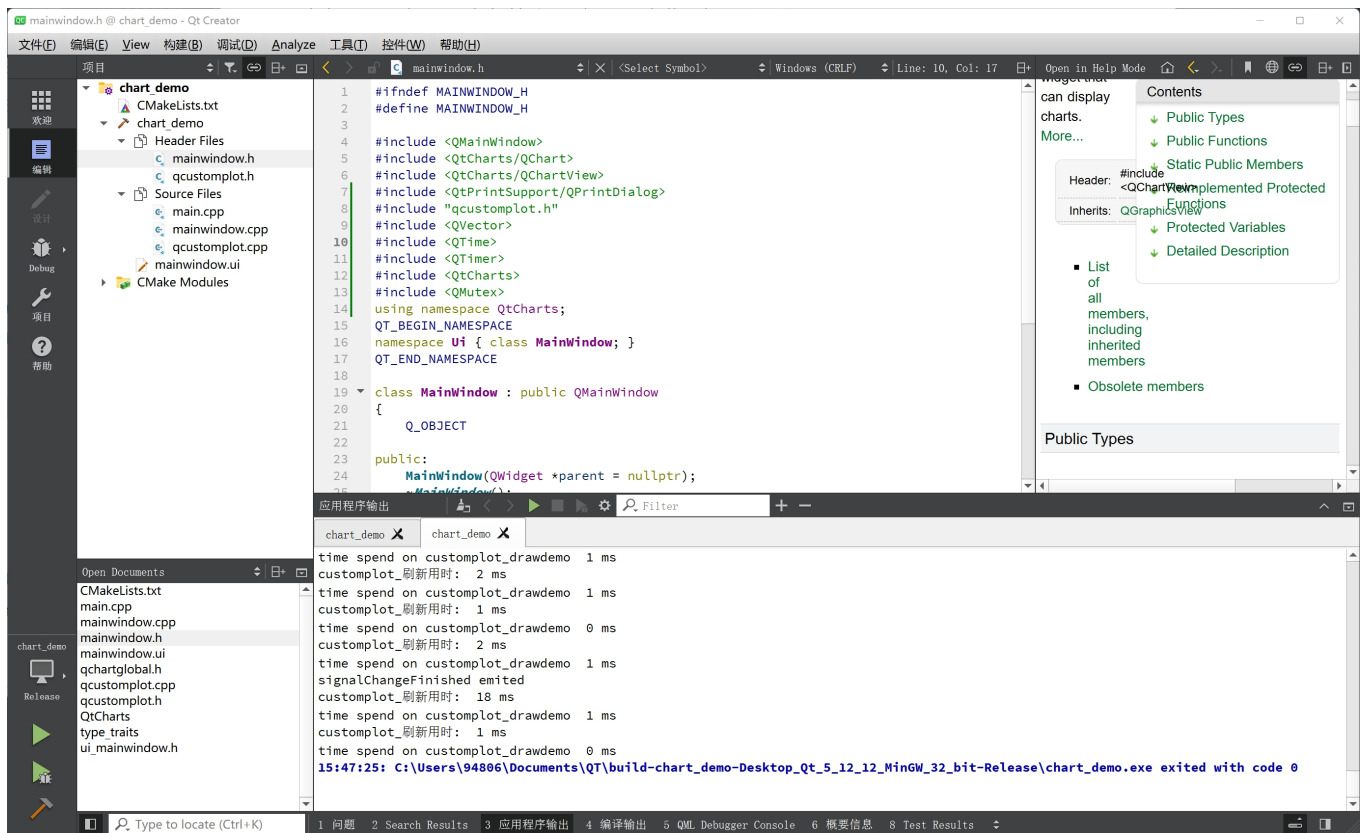
[QCustomPlot官方网站](#)

## 环境配置

1. 前往[QCustomPlot 官方下载链接](#)下载最新版本的QCustomPlot并解压



2. 把qcustomplot.cpp、qcustomplot.h两个文件添加到到qt工程源文件。



3. 如果是cmake构建的qt工程，需要添加PrintSupport库，CMakeLists.txt中相关代码字段示例：

```
find_package(QT NAMES Qt6 Qt5 COMPONENTS Widgets PrintSupport REQUIRED)
find_package(Qt${QT_VERSION_MAJOR} COMPONENTS Widgets PrintSupport REQUIRED)
```

REQUIRED表示找不到这些包就报错不进行编译，最主要是添加PrintSupport。

target\_link\_libraries也要进行修改：

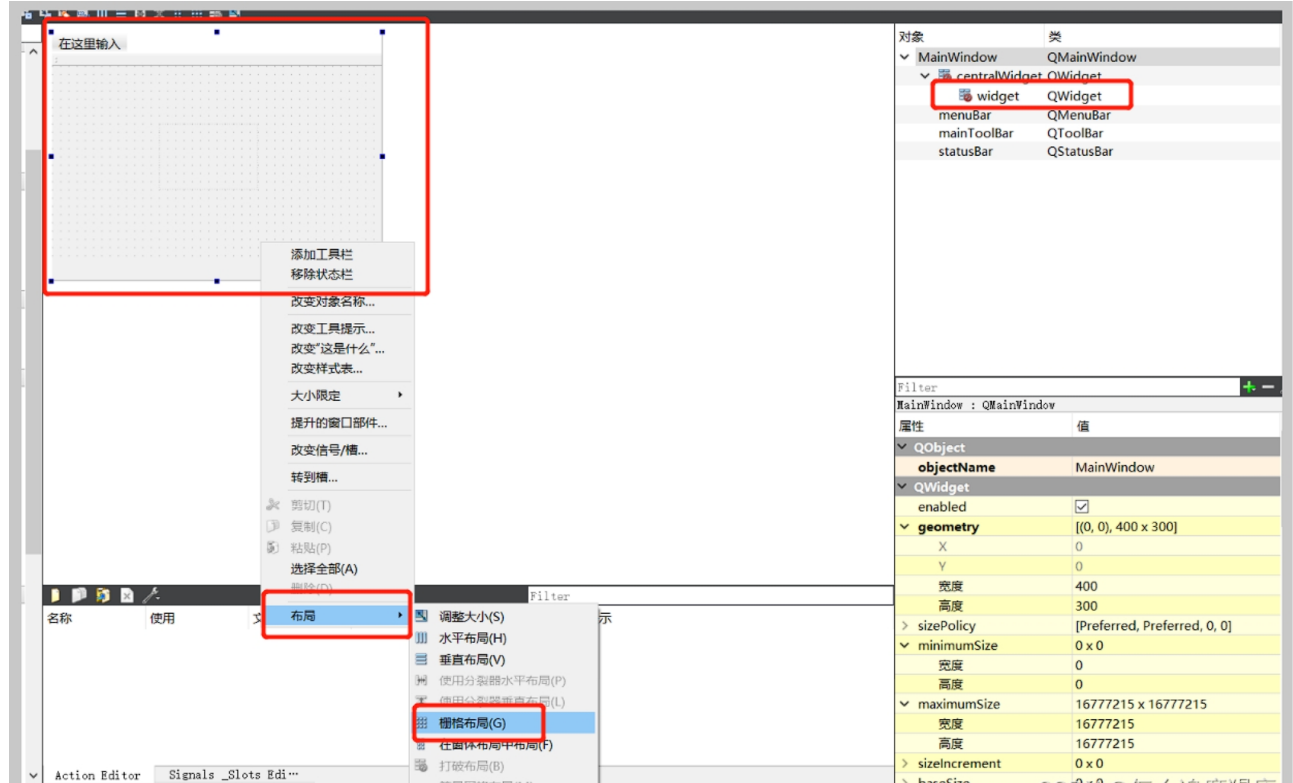
```
target_link_libraries(${你的项目名称} PRIVATE
    Qt${QT_VERSION_MAJOR}::Widgets
    Qt${QT_VERSION_MAJOR}::PrintSupport)
```

若是qmake构建的工程，则只需在.pro中添加 `QT += printsupport`

4. 引用 `#include "qcustomplot.h"` 即可使用

## 使用示例

# 1. 在\*.ui界面中添加一个QWidget，并将其提升为QCustomPlot



在这里输入

改变对象名称...

变型为

改变工具提示...

改变“这是什么”...


改变样式表...


大小限定


Layout Alignment


提升为...

转到槽...

 放到后面(B)

 放到前面(F)

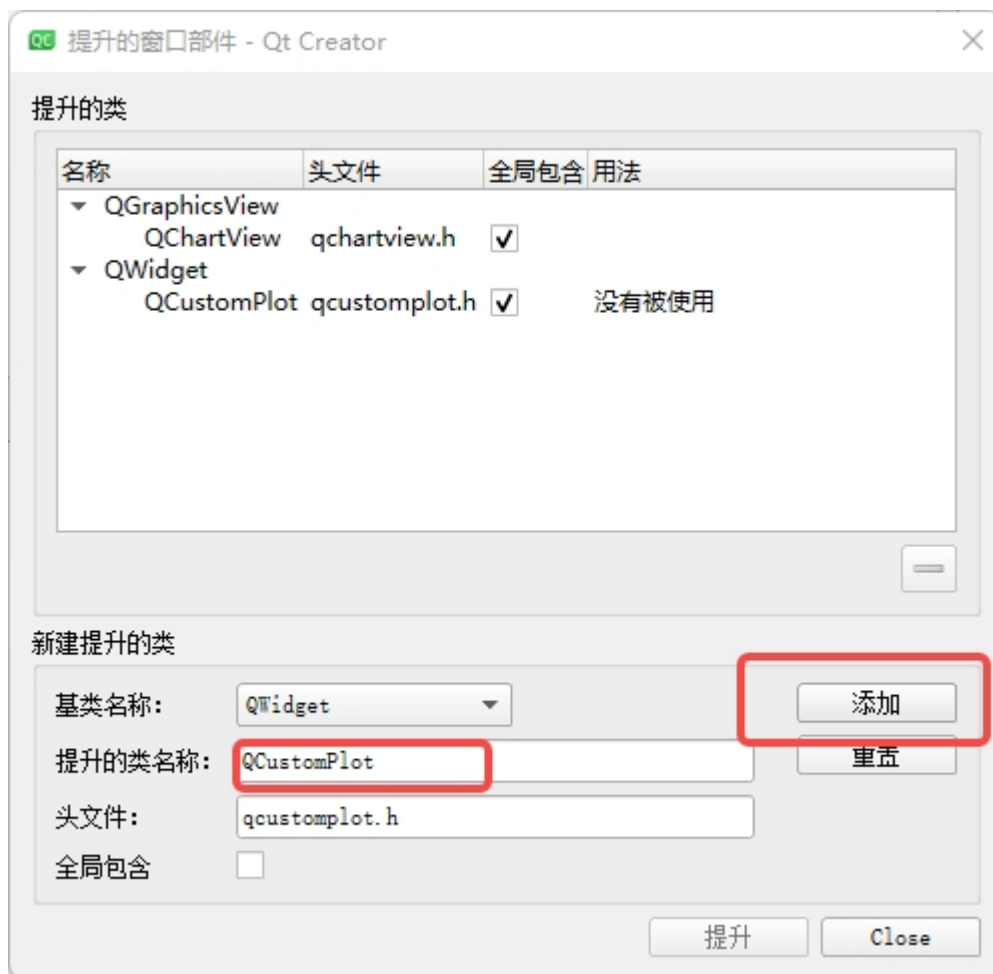
 剪切(T)

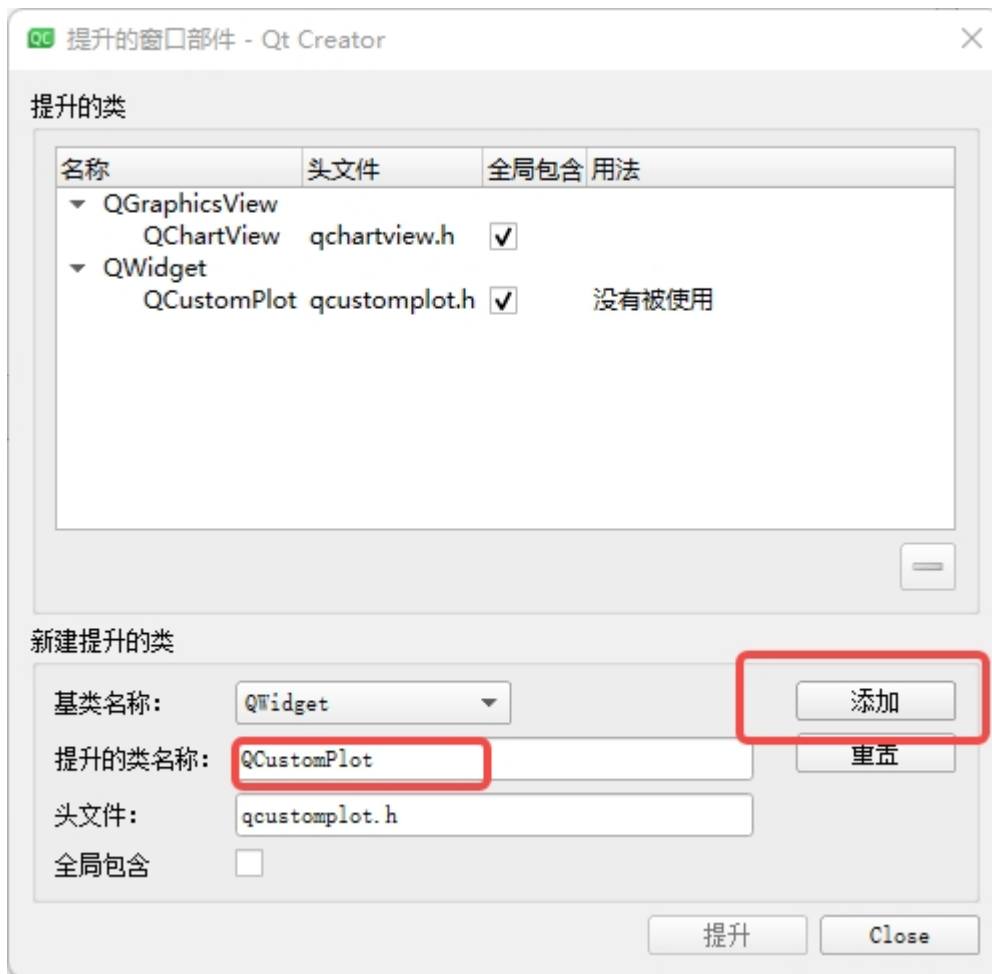
 复制(C)

 粘贴(P)

选择全部(A)

删除(D)





在这里我将QCustomPlot实例窗口命名为customplot，绘图代码如下：

```
void MainWindow::customplot_drawdemo(){
    QVector<double> x(20000),y(20000);
    for(int i=0;i<20000;i++){
        x[i]=i/1000.0-10;
        y[i]=qSin(x[i]);
    }
    if(ui->customplot->graphCount()==0)
        ui->customplot->addGraph();
    //ui->customplot->graph(0)->
    ui->customplot->graph(0)->setData(x,y);
    ui->customplot->xAxis->setLabel("x");
    ui->customplot->yAxis->setLabel("y");
    ui->customplot->yAxis->setRange(-1,1);
    ui->customplot->xAxis->setRange(-11,11);
    ui->customplot->legend->setVisible(true); // 显示图例
    ui->customplot->replot();
}
```

效果如下：



## 与qchart相比性能比较

比qchart性能好