

1.简介

QCustomPlot是用于绘图和数据可视化的Qt C ++小部件。它没有进一步的依赖关系，并且有据可查。该绘图库专注于制作美观，出版质量的2D绘图，图形和图表，以及为实时可视化应用程序提供高性能。看一下“ 设置”和“ 基本绘图”教程以开始使用。QCustomPlot可以导出为各种格式，例如量化的PDF文件和光栅化的图像（如PNG，JPG和BMP）。QCustomPlot是用于在应用程序内部显示实时数据以及为其他媒体生成高质量图的解决方案。

2.QCustomPlot下载与安装

参考：<https://www.qcustomplot.com/index.php/tutorials/settingup>

QCustomPlot 2.1.1官方文档：<https://www.qcustomplot.com/documentation/index.html>

QCustomPlot官网地址：<https://www.qcustomplot.com/>

- (1) 下载下来的qcustomplot.h和qcustomplot.cpp加入工程中
- (2) 如果Qt版本在5.0以上，需要在.pro文件中的QT变量加上printsupport

```
1 QT += printsupport
```

- (3) 添加帮助文档

在下载documentation文件夹下有个qcustomplot.qch文件，将它拷贝Qt的文档目录下（一般为C:\Users\用户名\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Qt\6.1.0\MinGW 8.1.0 (64-bit)，会根据你的Qt版本号而做相应变动），然后在QtCreator——>工具——>选项——>帮助——>文档——>添加，选择qcustomplot.qch文件，确定，以后按F1就能跳转到QCustomPlot的帮助文档了。

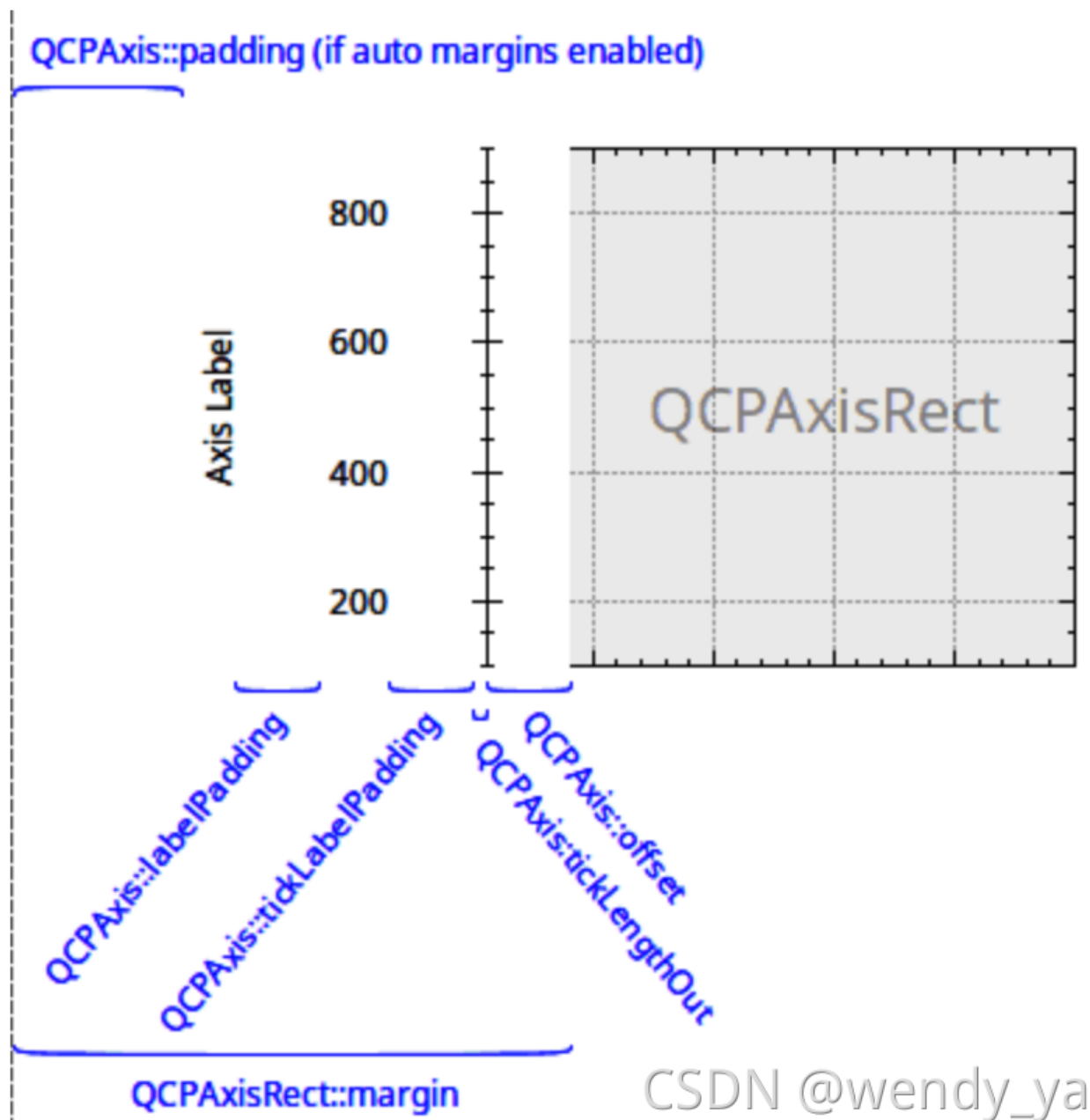
3.QCustomPlot图层介绍

图层介绍：

QCustomPlot类管理着所有的图层，它默认自带了六个图层，分别是：

1. 背景层background
2. 网格层grid
3. 绘图层main
4. 坐标轴层axes
5. 图例层legend
6. overlay层overlay

依据层的顺序的不同，绘制的顺序也不同，越在底下的层越早绘制，当前层默认为绘图层main。下图中就是QCPAxisRect。



层有两种刷新模式：

- ImLogical：只用于渲染的顺序，并且与相邻的ImLogical层共享绘图缓存（注意这里的相邻，如果前一个层是ImBuffered模式，则会新建一个绘图缓存用于绘图）
- ImBuffered：拥有自己的绘图缓存，并且可以单独的刷新（调用QCPLayer::replot函数）

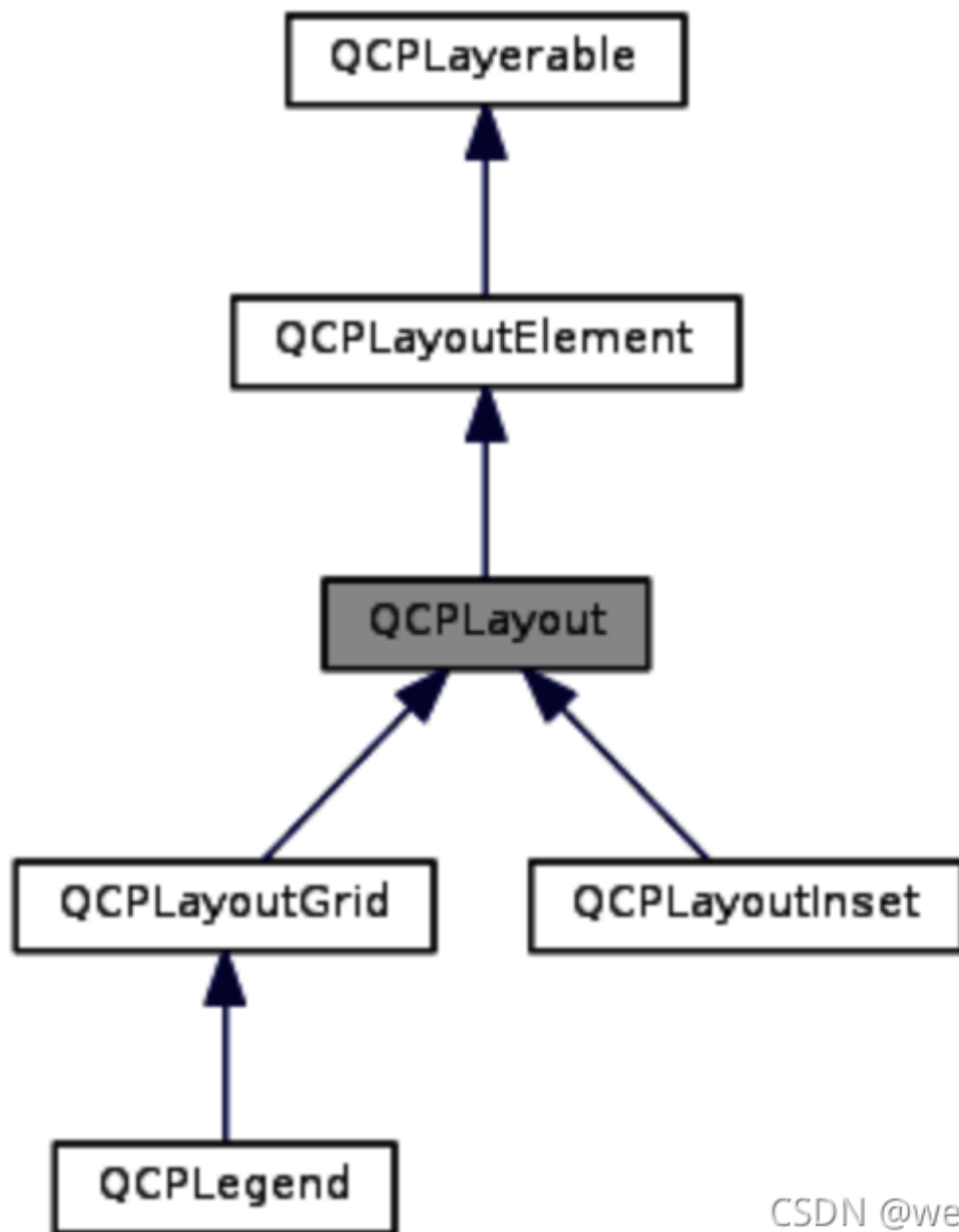
默认只有overlay层启用了ImBuffered单独绘制机制，其它层都共享一个绘图缓存，因为overlay层主要是用于存放悬浮与图表上的item项，需要频繁的刷新，启用了ImBuffered的层会多占用内存。

布局方式：

在QCustomPlot中，布局方式只有两种：

- QCPLLayoutGrid：网格布局
- QCPLLayoutInset：内嵌式布局，这种布局的作用是把子元素与布局的边界对齐或者将其放置在布局内的任意位置(图例QCPLegend就被放置在此布局内)，严格来说这个并不算布局

继承关系图如下所示：



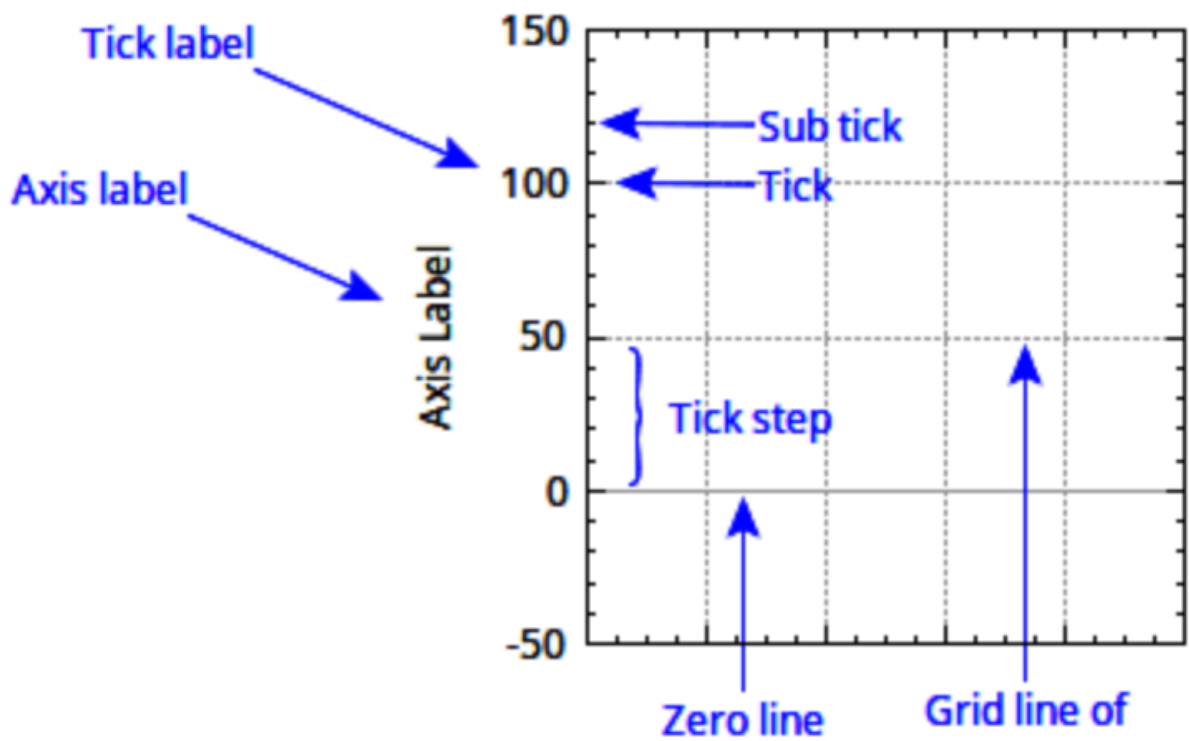
CSDN @wendy_ya

这两种布局都继承自QCPLayout，而QCPLayout继承自QCPLayoutElement布局元素，所以一个布局可以包含另一个布局，布局内的元素都要继承自QCPLayoutElement。

4.QCustomPlot坐标轴的相关属性和方法

(1) QCustomPlot中的轴介绍：

QCustomPlot中由QCPAxis类管理QCustomPlot内的单个轴。通常不需要外部实例化。通过QCustomPlot:: xAxis（底部）、QCustomPlot:: yAxis（左侧）、QCustomPlot:: xAxis2（顶部）和QCustomPlot:: yAxis2（右侧）访问QCustomPlot的默认四个轴。其位置如下图所示：



CSDN @wendy_ya

(2) 设置轴标签与清空轴标签：

设置轴标签利用setLabel函数：

```
1 void QCPAxis::setLabel(const QString &str)
```