

《Qt 5.9 C++开发指南》示例源程序

在 Qt 5.15.2 中测试和修改记录

王维波

2023-3-23



说 明

《Qt 5.9 C++开发指南》在 2018 年 5 月出版，随书发布的示例源程序是针对 Qt 5.9 编写和测试的。《Qt 5.9 C++开发指南》出版后，该书的内容没有做大的改版，只是在 2019 年随着 Qt 5.12 LTS(Long Term Supported, 长期支持) 的发布，我们在异步社区发布了针对 Qt 5.12 测试和修改后的示例源程序。

Qt 6.0 在 2020 年底发布。虽然 Qt 6 已经发布两年多了，但是还有不少人在使用 Qt 5，特别是在实际应用中，一些项目或为了向后兼容，或为了稳定性的考虑，依然在使用 Qt 5。Qt 5.15 是 Qt 5 系列最后一个 LTS 版本，Qt 公司依然在维护和更新 Qt 5.15。

在 Qt 5.15 LTS 发布后，本书没有及时发布针对 Qt 5.15 的示例源程序，一是因为我忙其他的事情了，二是觉得有 Qt 5.9 和 Qt 5.12 两个 LTS 版本的示例源程序，新手选择其一就可以，老手自己更新到 Qt 5.15 也没什么大问题。

针对 Qt 6 系列第一个 LTS 版本 Qt 6.2 的新书《Qt 6 C++开发指南》在 2023 年 1 月出版了。《Qt 5.9 C++开发指南》一书可能很快就不再印刷了，但我总觉得像是欠了一份作业似的，于是花了些时间将《Qt 5.9 C++开发指南》一书中的示例程序在 Qt 5.15 中测试和修改，再发布到异步社区里，并编写了本文档。

这样，《Qt 5.9 C++开发指南》一书在异步社区提供 3 个版本的示例源程序：

- (1) 针对 Qt 5.9 的示例程序，与书的内容完全对应；
- (2) 针对 Qt 5.12 的示例程序；
- (3) 针对 Qt 5.15 的示例程序，修改说明文档比较详细（也就是本文档）。

如此，《Qt 5.9 C++开发指南》提供的示例程序就算是比较完美了。如果还有读者使用《Qt 5.9 C++开发指南》学习 Qt 5 的编程，就可以根据自己使用的 Qt 5 版本选择对应的示例程序。如果要学习 Qt 6 的编程，就直接使用《Qt 6 C++开发指南》一书。

第 1 章 认识 Qt

示例 samp1_2MSVC

在 Visual Studio 2019 中，需要安装 Qt VS Tools 扩展后，才可以在 Visual Studio 中创建 Qt 项目。点击 VS 的“扩展→管理扩展”菜单项，在 Marketplace 中搜索“Qt”，找到 Qt Visual Studio Tools（如图 1），安装此扩展即可。



图 1 安装 Qt Visual Studio Tools 扩展

安装 Qt Visual Studio Tools 后，在 Visual Studio 的“扩展”菜单下会新增一个菜单项“Qt VS Tools”，其下面有一些子菜单。要使用 Qt VS Tools，需要先设置 Qt Version。点击菜单项“扩展→Qt VS Tools→Qt Versions”，打开图 2 所示的对话框。在列表中找到安装在本机上的 Qt 的 MSVC 开发套件，例如我的电脑上安装了 MSVC2019-64 位的开发套件，在目录 “D:\Qt\6.2.3\msvc2019_64” 下，设置此 Qt 版本即可。

安装并设置好 Qt VS Tools 后，就可以在 Visual Studio 里创建和构建 Qt 项目了。

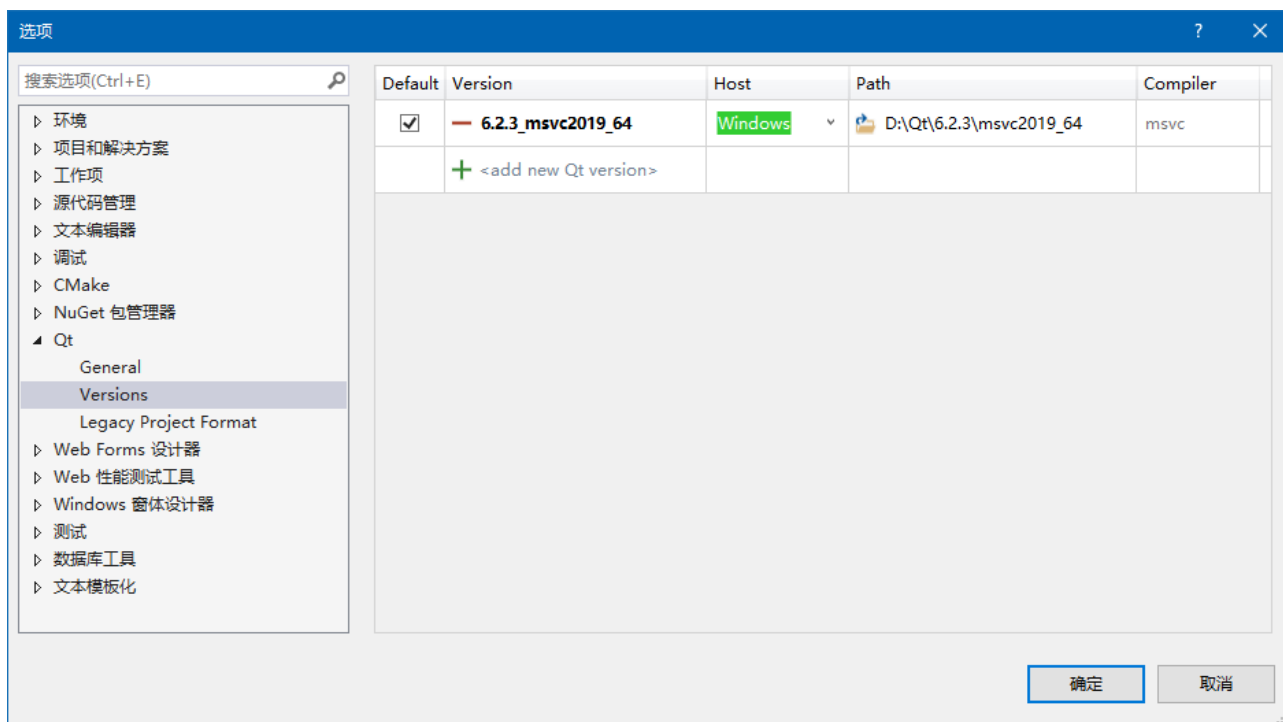


图 2 设置 Qt Versions

第 2 章 GUI 应用程序设计基础

代码无改动

第 3 章 Qt 类库概述

代码无改动

第 4 章 常用界面设计组件

示例 samp4_1Edit

“计算”按钮的代码中用到 QString 的 sprintf()函数,编译时提示这个函数已经被弃用(deprecated),换成用静态函数 QString::asprintf()。“计算”按钮槽函数代码如下:

```
void Widget::on_btnCal_clicked()
{ // “计算”按钮单击响应
    int num=ui->editNum->text().toInt(); //读取字符串为整数
    float price=ui->editPrice->text().toFloat(); //读取字符串为浮点数
```

```

float total=num*price;           //相乘计算
QString str;
str=QString::asprintf("%.2f",total); //格式化输出浮点数
ui->editTotal->setText(str);      //在文本框里显示
}

```

示例 samp4_5DateTime

编译时提示 `QTime` 已经 deprecated。换成使用 `QElapsedTimer`，修改 `Dialog` 类中变量 `fTimeCounter` 的类型为 `QElapsedTimer` 即可。

```

class Dialog : public QDialog
{
    Q_OBJECT
private:
    QElapsedTimer fTimeCounter; //计时器，换成 QElapsedTimer 类型
}

```

示例 samp4_9TableWidget

编译时提示 `QTableWidgetItem` 的 `setBackgroundColor()` 函数已经 deprecated，要换成 `setBackground()` 函数，即

```

// item->setBackgroundColor(Qt::yellow); // setBackgroundColor() 函数已经 deprecated
item->setBackground(QBrush(Qt::yellow));

```

编译时提示 `QTableWidgetItem` 的 `setTextColor()` 函数已经 deprecated，要换成 `setForeground()` 函数，即

```

// headerItem->setTextColor(Qt::red); // setTextColor() 函数已经 deprecated
headerItem->setForeground(QBrush(Qt::red)); // 字体颜色

```

第 5 章 Model/View 结构

示例 samp5_3StandardModel

编译时提示 `QString` 的 `split()` 函数参数中使用的 `QRegExp` 过时了 (deprecated)，需要使用 `QRegularExpression`。修改前后的语句如下：

```

// QStringList headerList=header.split(QRegExp("\\s+"),QString::SkipEmptyParts);
// QRegExp 过时了
QStringList headerList=header.split(QRegularExpression("\\s+"));

```

示例 samp5_4Delegate

编译时提示 QString 的 split() 函数参数中使用的 QRegExp 过时了 (deprecated)，需要使用 QRegularExpression。修改前后的语句如下：

```
// headerList=header.split(QRegExp("\\s+"),QString::SkipEmptyParts); //QRegExp 过时了
headerList=header.split(QRegularExpression("\\s+"));
```

第 6 章 对话框与多窗体设计

示例 samp6_5SplashLogin

编译时提示 QByteArray 的 append() 函数过时，改为使用 QString 的 toLocal8Bit() 函数。

```
QByteArray btArray;
// btArray.append(str); //append() 函数过时
btArray = str.toLocal8Bit();
```

第 7 章 文件系统和文件读写

示例 samp7_3DirFiles

编译时提示 QFileInfo 的 created() 函数 deprecated，换用函数 birthTime()，修改为：

```
// QDateTime dt=fileInfo.created(); //created() 函数过时
QDateTime dt=fileInfo.birthTime();
```

第 8 章 绘图

示例 samp8_2PaintPath

编译时提示错误，paintEvent() 的代码中 std::cos() 和 std::sin() 函数不存在。改为使用 <QtMath> 中的函数 qSin() 和 qCos()，修改后可编译了。

```
#include <QtMath>
```

```
QPoint points[5]={ //使用<QtMath>中的函数 qSin(), qCos()
    QPoint(R,0),
    QPoint(R*qCos(deg), -R*qSin(deg)),
```

```

    QPoint(R*qCos(2*deg), -R*qSin(2*deg)),
    QPoint(R*qCos(3*deg), -R*qSin(3*deg)),
    QPoint(R*qCos(4*deg), -R*qSin(4*deg))
};

```

示例 samp8_5GraphicsDraw

这个示例中用到 `qrand()` 和 `rand()` 函数生成随机数，但是这些函数是 deprecated。程序中换成使用 `QRandomGenerator` 类生成随机数。例如，下面的替换

```

// item->setPos(-50+(qrand() % 100), -50+(qrand() % 100));
item->setPos(QRandomGenerator::global()->bounded(-50, 50),
            QRandomGenerator::global()->bounded(-50, 50));

```

第 9 章 Qt Charts

示例 samp9_1SimpleChart

编译时提示 `QChart::setAxisX()`, `QChart::setAxisY()` 函数过时，应改为下面的方式添加坐标轴

```

////下面这些设置坐标轴的代码过时
// chart->setAxisX(axisX, series0); //为序列设置坐标轴
// chart->setAxisY(axisY, series0); //

// chart->setAxisX(axisX, series1); //为序列设置坐标轴
// chart->setAxisY(axisY, series1); //

////使用以下的代码设置坐标轴
chart->addAxis(axisX, Qt::AlignBottom);
chart->addAxis(axisY, Qt::AlignLeft);

series0->attachAxis(axisX);
series0->attachAxis(axisY);

series1->attachAxis(axisX);
series1->attachAxis(axisY);

```

示例 samp9_2ChartConfig

这个示例中用到 `qrand()` 和 `rand()` 函数生成随机数，但是这些函数是 deprecated。程序中换成使用 `QRandomGenerator` 类生成随机数。例如

```

// rd=(qrand() % 10)-5; //随机数, -5~+5
rd= QRandomGenerator::global()->bounded(-5, 6); //

```

示例 samp9_3AllCharts

1. 随机数问题

这个示例中用到 `qrand()`和 `qrand()`函数生成随机数，但是这些函数是 `deprecated`。程序中换成使用 `QRandomGenerator` 类生成随机数。

2. QChart 的过时函数 `axisX()`和 `axisY()`

编译时提示 `QChart` 的 `axisX()`和 `axisY()`函数过时，即下面的两行语句是过时的写法

```
chart->removeAxis(chart->axisX()); //删除坐标轴
chart->removeAxis(chart->axisY()); //删除坐标轴
```

为替换这两行代码，自定义了一个函数 `removeAllAxis()`，代码如下：

```
void MainWindow::removeAllAxis(QChart *chart)
{ //删除一个 chart 的所有坐标轴
    QList<QAbstractAxis *> axisList=chart->axes(); //获取坐标轴列表
    int count=axisList.count();
    for(int i=0; i<count; i++)
    {
        QAbstractAxis *one=axisList.at(0);
        chart->removeAxis(one); //从图表中移除坐标轴，
        axisList.removeFirst(); //从列表中移除坐标轴
        delete one; //删除坐标轴对象，释放内存
    }
}
```

程序中替代后的代码如下：

```
// chart->removeAxis(chart->axisX()); //删除坐标轴
// chart->removeAxis(chart->axisY()); //删除坐标轴
removeAllAxis(chart); //
```

3. QChart 的过时函数 `setAxisX()`和 `setAxisY()`

按示例 `samp9_1SimpleChart` 中的修改方法进行修改

示例 samp9_4RectZoom

这个示例中用到 `qrand()`和 `qrand()`函数生成随机数，但是这些函数是 `deprecated`。程序中换成使用 `QRandomGenerator` 类生成随机数。

第 10 章 Data Visualization

代码无改动

第 11 章 数据库

代码无改动

第 12 章 自定义插件和库

示例 samp12_2: 编写并使用 Widget 插件

由于测试电脑上只安装了 Qt 5.15.2 MinGW 64-bit 的开发套件，所以编写的插件无法在 Qt Creator 的组件面板中显示。但是示例项目 QwBatteryPlugin 编译没有问题，生成的插件相关的文件 qwbatteryplugin.lib 和 qwbatteryplugin.dll 给示例项目 BatteryUser 使用也没有问题。

示例 samp12_4: 创建和使用动态库

samp12_4 调用 Delphi 编写的实例无法运行，因为 Delphi 是 32 位的，编写的 DLL 是 32 位的。测试电脑上只安装了 Qt 5.15.2 MinGW 64-bit 的开发套件，编译的程序是 64 位的，无法调用 32 位的 DLL 文件。

第 13 章 多线程

示例 samp13_1ThreadSignal

本章多个示例中都需要生成[1,6]之间的随机整数，原来程序中用到 `qsrand()`和 `qrand()`函数生成随机数，但是这些函数是 deprecated。程序中换成使用 `QRandomGenerator` 类生成随机数。

```
//      m_diceValue=qrand(); //获取随机数，qrand()是过时函数
//      m_diceValue=(m_diceValue % 6)+1;
m_diceValue= QRandomGenerator::global()->bounded(0,7); //随机数[0,6]
```

第 14 章 网络编程

代码无改动

第 15 章 多媒体

示例 samp15_4

替换了过时的函数 `QChart::setAxisX()`和 `QChart::setAxisY()`。

第 16 章 应用程序设计辅助功能

代码无改动