

Qt自学笔记1：下载安装&程序示例&Visual Studio配置

原创 Zerg_Wang 于 2020-04-21 00:11:52 发布 338 收藏 1 编辑 版权

分类专栏： Management 文章标签： qt qt5 visual c++ visual studio



Management 专栏收录该内容

0 订阅 10 篇文章

下载

官网下载：<http://download.qt.io/>

国内镜像站：

<http://mirrors.ustc.edu.cn/qtproject/>

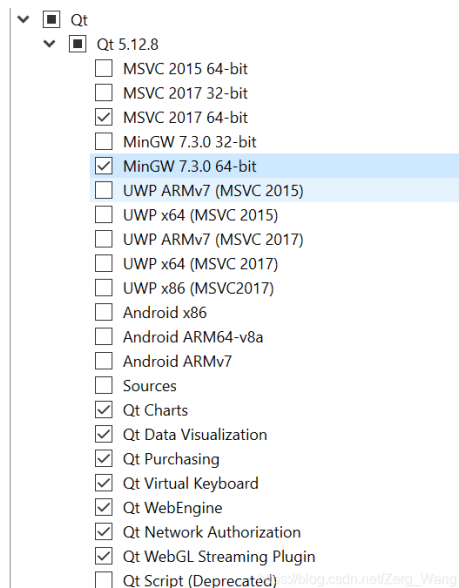
<https://mirrors.tuna.tsinghua.edu.cn/qt/>

建议下载安装长期支持版本（LTS），现在（2020年4月）长期支持版本为5.12。5.12版本下有多个补丁版本（5.12.0到5.12.8），但版本特性是一样的，所以可随意选择。本人安装版本为5.12.8。下载的文件位置在official_releases/qt/5.12/5.12.8目录下，Windows版的安装包为qt-opensource-windows-x86-5.12.8.exe。（没错，只有x86，没带x64.....安装包只有32位的）

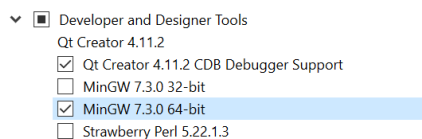
以上下载的为遵循GPL协议的开源免费版本，使用遵循该协议的工具或代码，一般来说最终成果也是要开源的，当然Qt也提供了适用于商业软件开发的收费版本，详情可在官网查看。

安装

注册账号后即可安装：



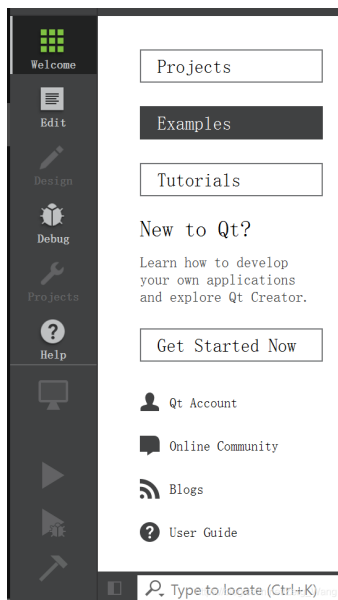
MinGW 为编译器，必选，MSVC 为 Qt 适配 Visual Studio 的插件，如果使用 Qt 自带的 IDE，则可以不选。UWP 为开发 Windows 10 中UWP应用的组件，看需求来选，Android 同理，Sources 为源代码，一般用不到，后面 Qt 开头的为附加模块，建议都选上（除了Qt Script）



https://blog.csdn.net/Zerg_Wang

Qt Creator 和 MinGW 必选，Strawberry Perl 是用于编译 Qt 源代码的 Perl 开发环境，可以不选，以后有需要可以再安装。

Qt Creator 简介

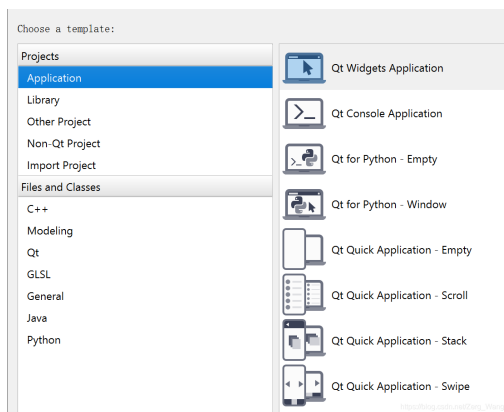


打开后我们处于“Welcome”界面，这里可以管理Projects，有代码示例（Examples）以及教程（Tutorials）。

在“Edit”界面，可以编辑源代码，在“Design”界面可以设计窗体、对话框等可视化界面，“Projects”界面为当前Project进行设置。

新建Project

左上角File—New File or Project，或者直接Ctrl + N，选择Application后：



（以下内容引用自：<http://c.biancheng.net/view/1817.html>）

- Qt Widgets Application，支持桌面平台的有图形用户界面（Graphic User Interface，GUI）界面的应用程序。GUI 的设计完全基于 C++ 语言，采用 Qt 提供的一套 C++ 类库。
- Qt Console Application，控制台应用程序，无 GUI 界面，一般用于学习 C/C++ 语言，只需要简单的输入输出操作时可创建此类项目。
- Qt Quick Application，创建可部署的 Qt Quick 2 应用程序。Qt Quick 是 Qt 支持的一套 GUI 开发架构，其界面设计采用 QML 语言，程序架构采用 C++ 语言。利用 Qt Quick 可以设计非常炫的用户界面，一般用于移动设备或嵌入式设备上无边框的应用程序的设计。这里简单提提 QML 语言，这是一种类似于 HTML 的标记语言，一般只用于编辑用户界面（UI），简单、灵活，跨平台，可以实现非常酷炫的界面效果。

至于其他的for Python 的模版暂时不用管……

选择Qt Widgets Application后，设置路径，后面一切默认即可（qmake + MainWindow，语言翻译那里默认不需要）

Class Information

Specify basic information about the classes for which you want to generate skeleton source code files.

Class name:

Base class:

Header file:

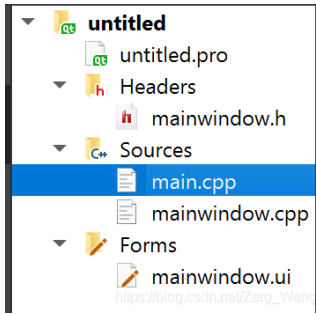
Source file:

☒ Generate form

Form file:

Kits套件建议全选，到时候编译的时候再挑一个就行了。

Project组成部分



.pro文件是Project的管理文件，存储有当前Project的一些设置。在Qt中打开Project时，就是选中 .pro 打开。

Headers下是所有的头文件（.h）。其中mainwindow.h是主窗体类的头文件。其内容为窗体的声明。

Sources下是 C++源代码（.cpp）。mainwindow.cpp 是主窗体类的实现文件，与 mainwindow.h 文件对应。窗体程序的具体定义、函数实现，均在mainwindow.cpp中。main.cpp 是主函数文件，也是应用程序的入口。

Forms下是界面文件（.ui）。其中mainwindow.ui是主窗体的界面文件。界面文件本质上 XML 语言的文本文件，用于描述界面（也就是说该文件设定了该窗体长什么样……）。双击mainwindow.ui即可进入Design对mainwindow窗体进行UI设计。

在编译的时候，编译器会根据mainwindow.ui的内容生成ui_mainwindow.h文件，将窗体内各组件信息转为标准的C++源码的形式与其他源文件进行编译、链接。

main函数

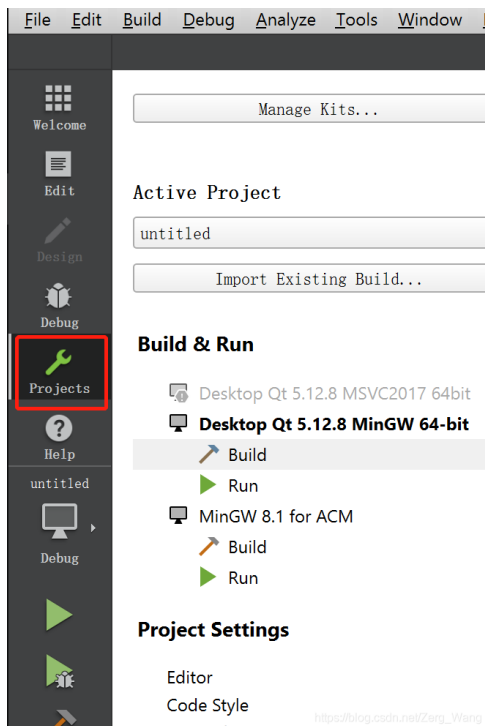
```
1  #include "mainwindow.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      MainWindow w;
8      w.show();
9      return a.exec();
10 }
```

现在我们看回main.cpp。第一行调用头文件mainwindow.h，即前文所述的那个窗体的声明。第二行的include的<QApplication>中有Qt 的标准应用程序类QApplication的声明，在main函数中第 1 行代码中，我们定义了一个 QApplication 类的实例 a，就是应用程序对象。

之后定义了一个 MainWindow 类的变量 w，MainWindow 就是前文所述窗体的类名。定义此窗体后通过w.show() 显示。

最后一行return a.exec() 启动应用程序的执行，开始应用程序的消息循环和事件处理。

运行Project



点击左侧的Projects，为当前Project指定编译器。每个编译器的 Build 界面上，有一个“Shadow build”复选框。如果勾选此项，编译后将在Project的同级目录下（也可自行指定路径）建立一个编译后的文件目录。由于目录名包含编译器信息，因此该情况适用于使用不同编译器创建不同版本可执行文件的情况。若不勾选，编译后的文件目录将在Project目录下。

点击左侧带“Debug”字样的小电视，选择编译工具以及编译模式（Debug or Release or Profile，按官方说法，Profile是Debug和Release的综合，兼顾性能与调试便利性）

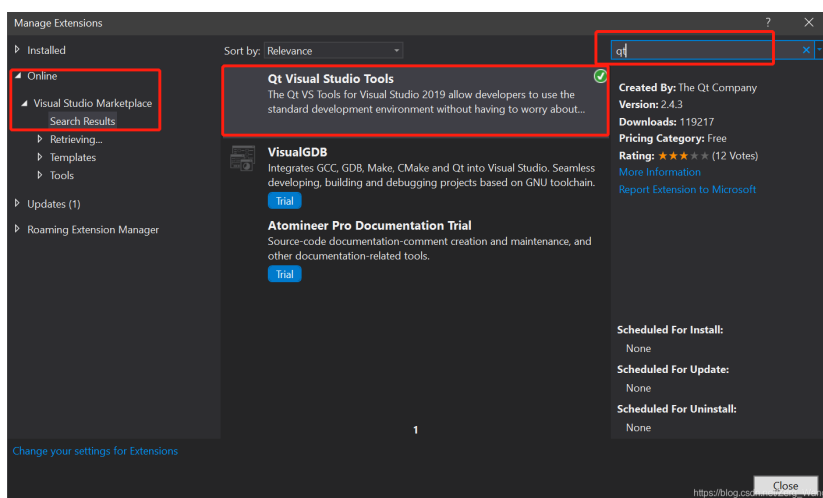
点击左侧的绿色箭头（或者 Ctrl + R），运行。

点击左侧带虫子的绿色箭头（或者 F5），进行调试。

点击最下面的锤子（Ctrl + B），进行编译。

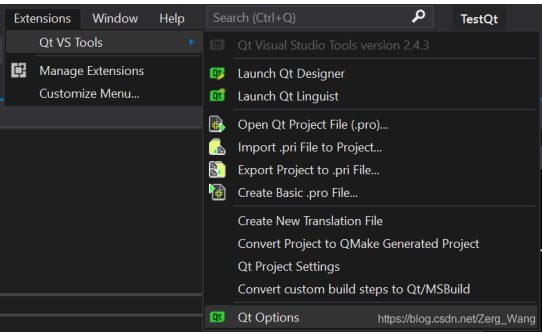
Visual Studio 配置

安装完毕后在 Visual Studio 中联机搜索 qt，安装 Qt Visual Studio Tools 拓展。



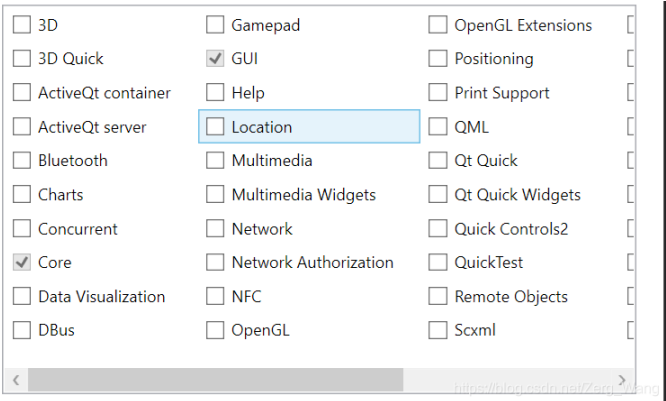
设置：点击Qt Options，添加 Qt 环境，Path填到对应msvc目录，如我的Path为：“C:\Softwares\Qt\Qt5.12.8\5.12.8\msvc2017_64”

即使是Visual Studio 2019，也可以用msvc 2017。



新建Project

打开Visual Studio，新建Project，选择Qt GUI Application，之后全部使用默认设置即可。



Project的编辑、运行就按照Visual Studio来即可。

参考资料

<http://c.biancheng.net/qt/>