QT简易入门

1.下载

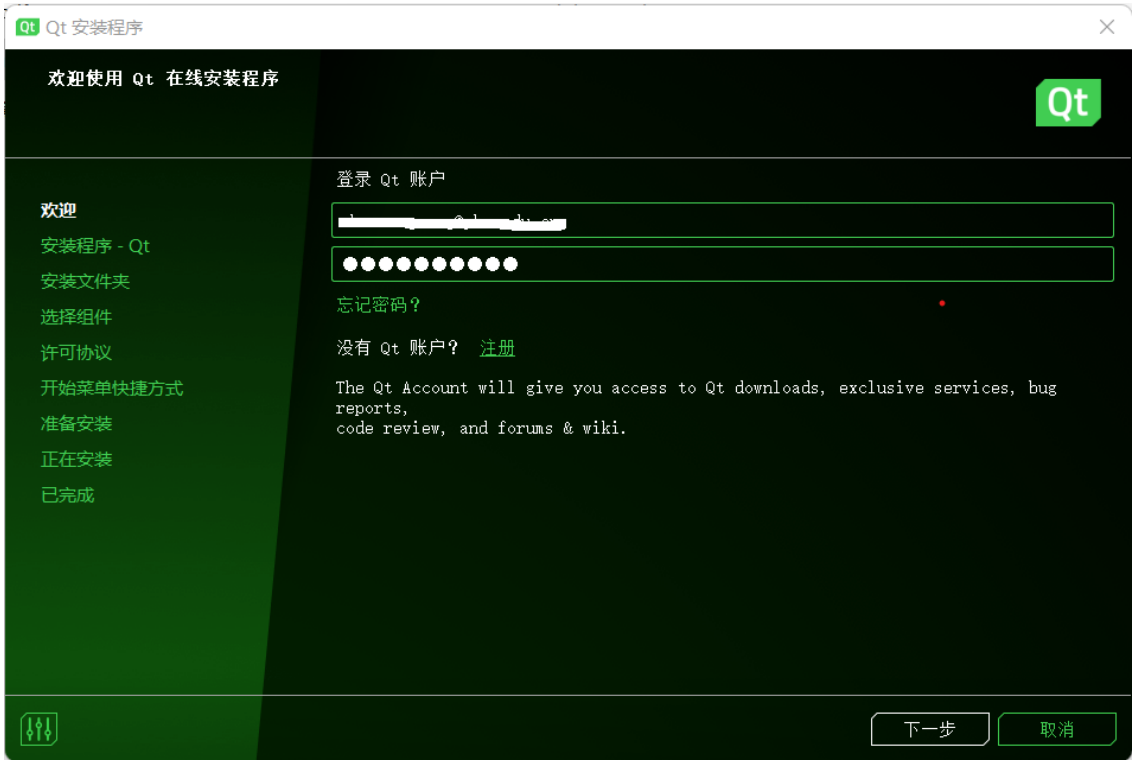
在https://www.qt.io/download，选择Downloads for open source users，Go open source，然后Looking for Qt binaries - Download the Qt Online Installer，Download

2.安装

2.1 注册/登录Qt账号

注册：在填写完sign-up之后会给你的邮箱发送一份邮件，需要点开邮件里面的链接进行确认。

登录：如下图



2.2 设置

鼠标点击左下角设置图标进行设置。 网络设置成无代理

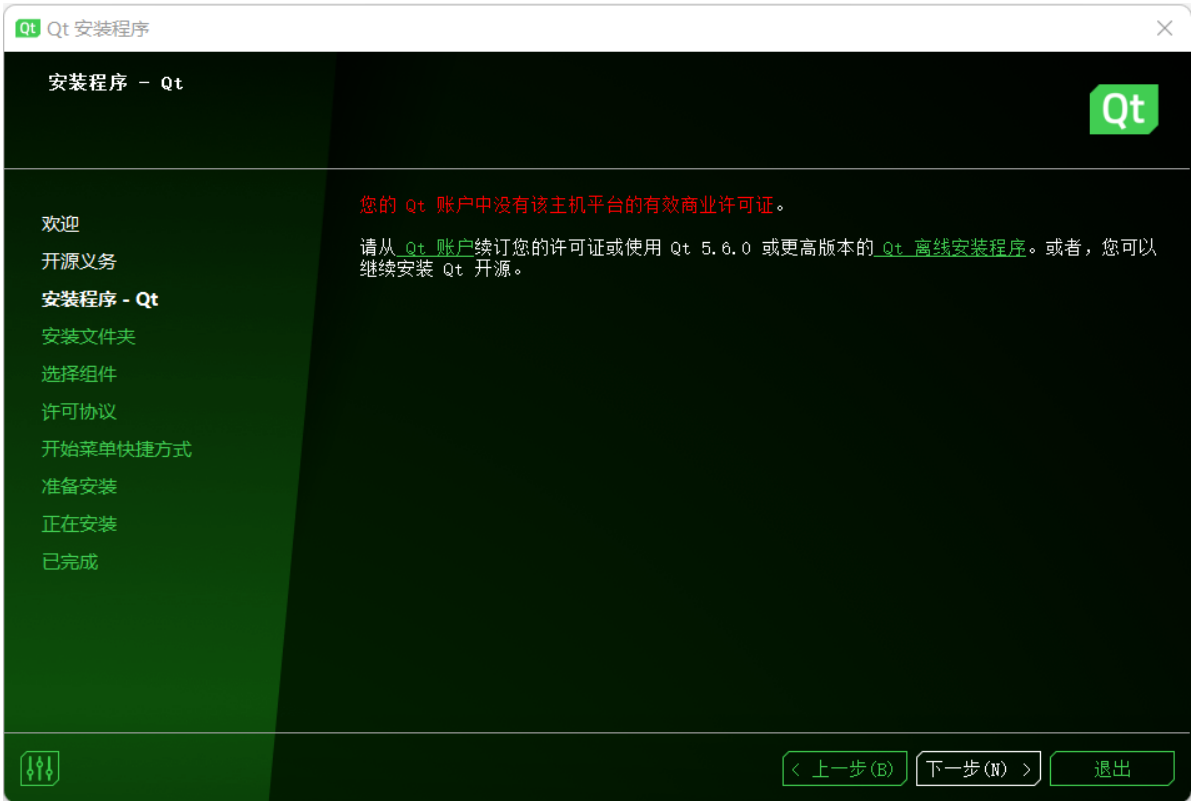

设置
网络
资料档案库
无代理
系统代理设置
手动代理配置
HTTP 代理：
端口：
0
FTP 代理：
端口：
0
确定
取消

2.3 安装

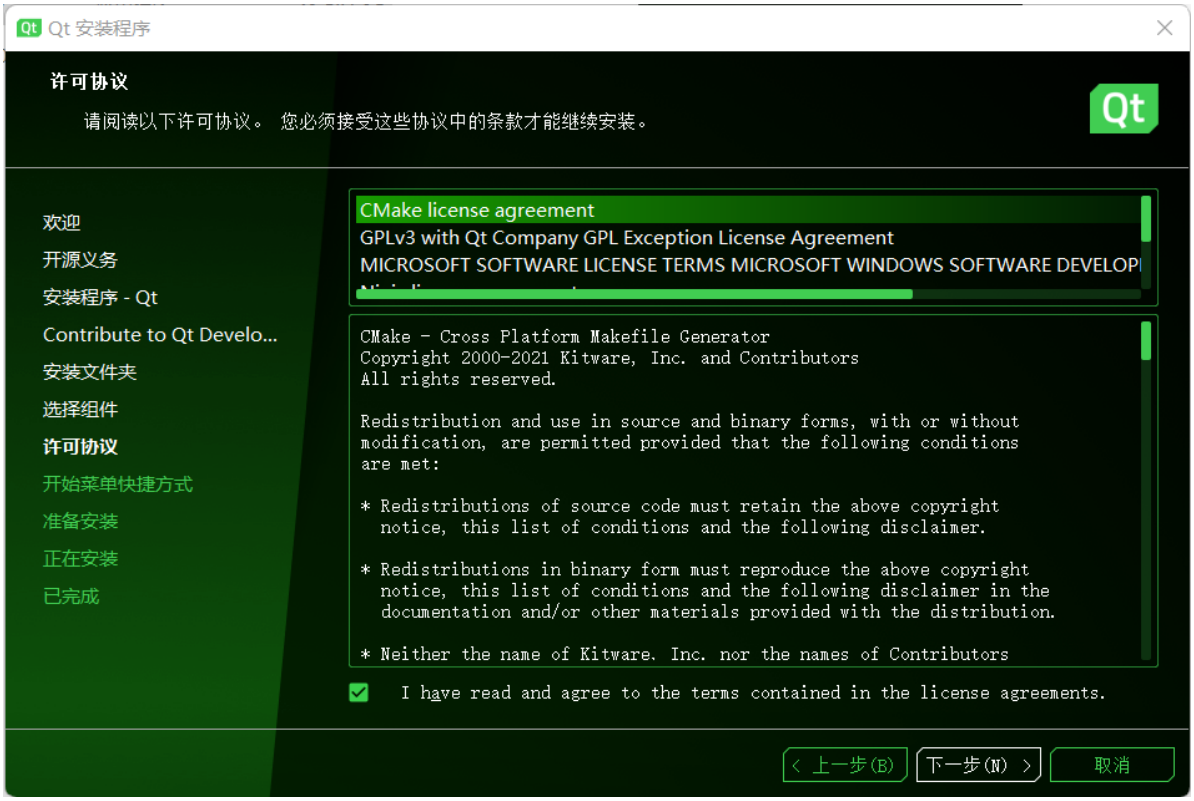
- 开源义务


Qt 安装程序
Qt 开源使用义务
欢迎
开源义务
安装程序 - Qt
安装文件夹
选择组件
许可协议
开始菜单快捷方式
准备安装
正在安装
已完成
一些组件在 GNU 通用公共许可证 v2 下可用。
阅读并接受下面的开源使用义务。阅读下面的链接可以帮助您为项目选择正确的许可证。
为项目选择正确的许可证
购买 Qt
（精简版）GNU 通用公共许可证 v3 义务
在一个项目或产品中，您不得同时使用商业 Qt 许可证开发的代码与 Qt 开源许可证开发的代码
提供许可证副本，并明确承认使用 Qt
为客户提供 Qt 源代码副本
接受 Qt 源代码修改是非专有的
制作消费类设备，允许用户在其中安装和运行修改后的软件版本
接受数字版权管理条款，参阅 GPL 常见问题
尝试强制执行软件专利时，请特别考虑 常问问题
如需更多信息，请参阅GPLv3和LGPLv3
我已阅读并同意使用开源 Qt 的条款和条件
输入您的公司/企业名称
我是个人用户，我不为任何公司使用 Qt
< 上一步(B)
下一步(N) >
取消

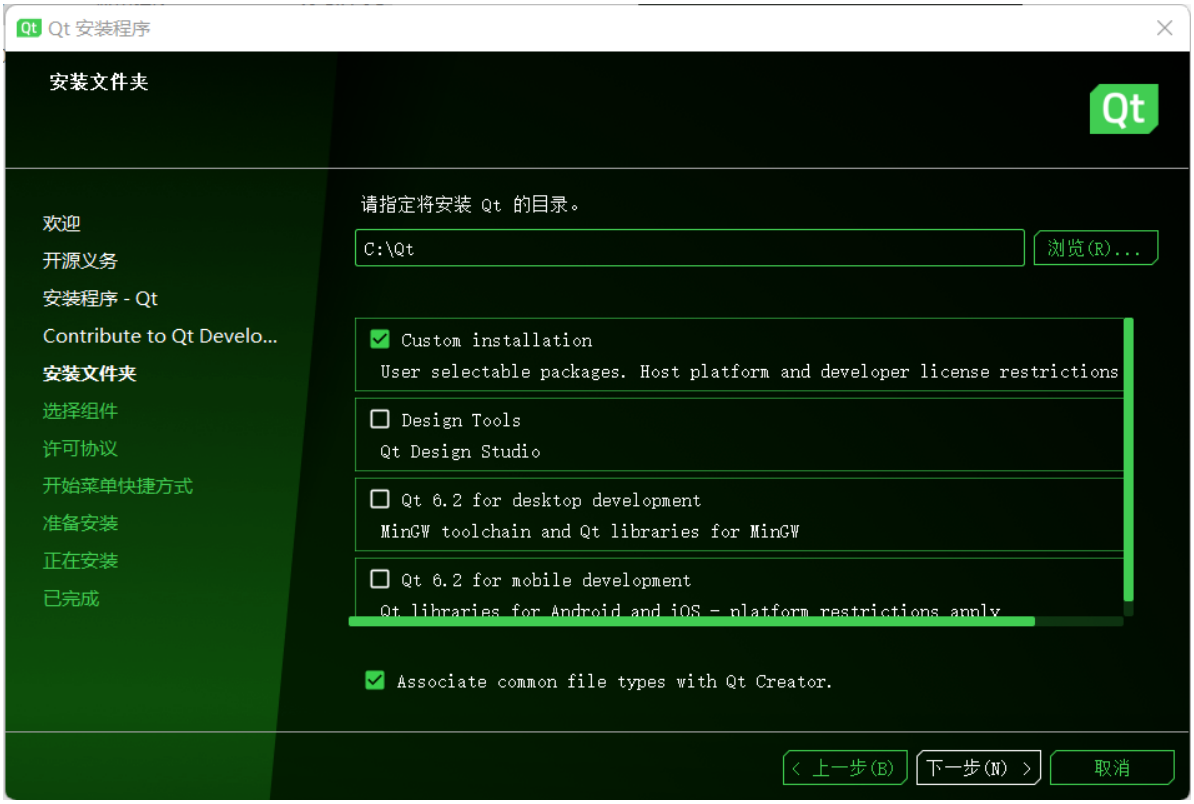
- 许可


Qt 安装程序
安装程序 – Qt
欢迎
开源义务
安装程序 - Qt
安装文件夹
选择组件
许可协议
开始菜单快捷方式
准备安装
正在安装
已完成
您的 Qt 账户中没有该主机平台的有效商业许可证。
请从 Qt 账户续订您的许可证或使用 Qt 5.6.0 或更高版本的 Qt 离线安装程序。或者，您可以继续安装 Qt 开源。
< 上一步(B)
下一步(N) >
退出

- 许可协议


Qt 安装程序
许可协议
请阅读以下许可协议。 您必须接受这些协议中的条款才能继续安装。
欢迎
开源义务
安装程序 - Qt
Contribute to Qt Develo...
安装文件夹
选择组件
许可协议
开始菜单快捷方式
准备安装
正在安装
已完成
CMake license agreement
GPLv3 with Qt Company GPL Exception License Agreement
MICROSOFT SOFTWARE LICENSE TERMS MICROSOFT WINDOWS SOFTWARE DEVELOPI
CMake - Cross Platform Makefile Generator
Copyright 2000-2021 Kitware, Inc. and Contributors
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Kitware, Inc. nor the names of Contributors
I have read and agree to the terms contained in the license agreements.
< 上一步(B)
下一步(N) >
取消

- 选择安装路径



- 选择安装组件

选择LTS版本（此处安装Qt5和Qt6各一个版本）

- 继续安装

3.更新

运行MaintenanceTool.exe进行更新

4. Visual Studio 2019的配置

4.1 Qt Visual Studio Tools插件

Visual Studio 2019 - Extensions - Manage Extensions

Serach: qt 选择Qt Visual Studio Tools下载，关闭Visual Studio 2019，更新插件；打开

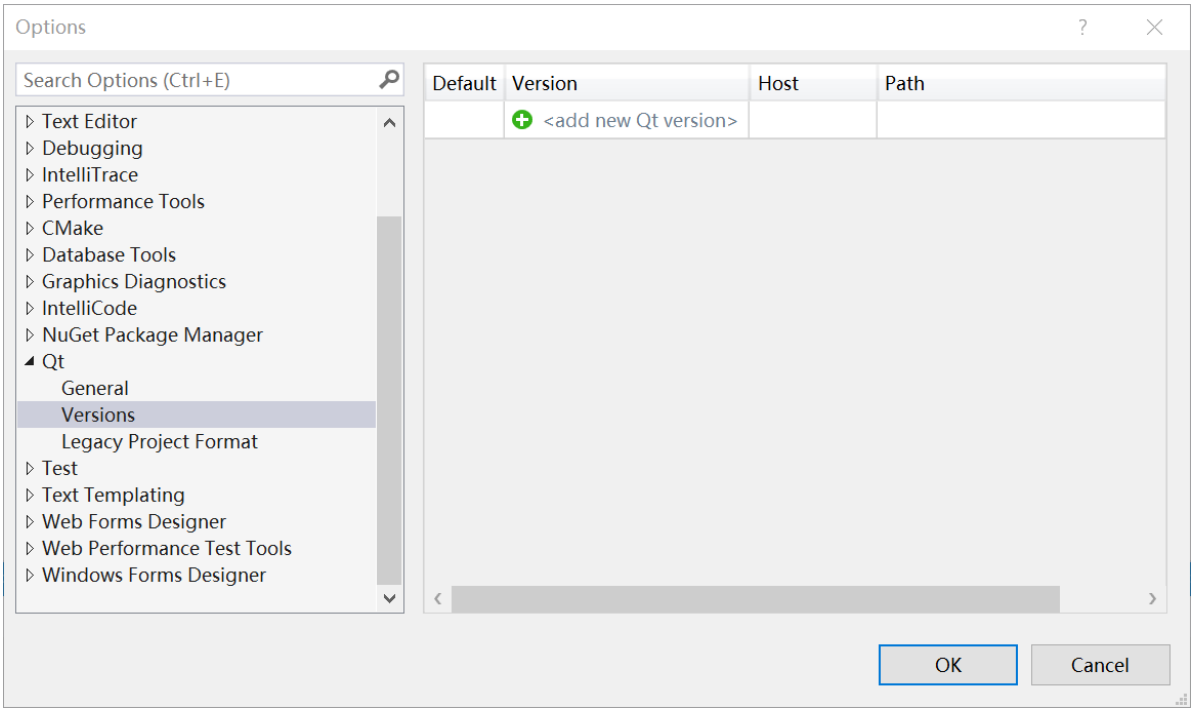
或者，直接https://marketplace.visualstudio.com/搜索qt，下载相应的版本，Maketplace(https://download.qt.io/development_releases/vsaddin/)：如，选择msvc2019，下载vsix文件，安装即可

如果Qt Visual Studio Tools安装有错误（如界面停止程序死机状态），可以试着安装其它的Extension，然后再Qt Visual Studio Tools。

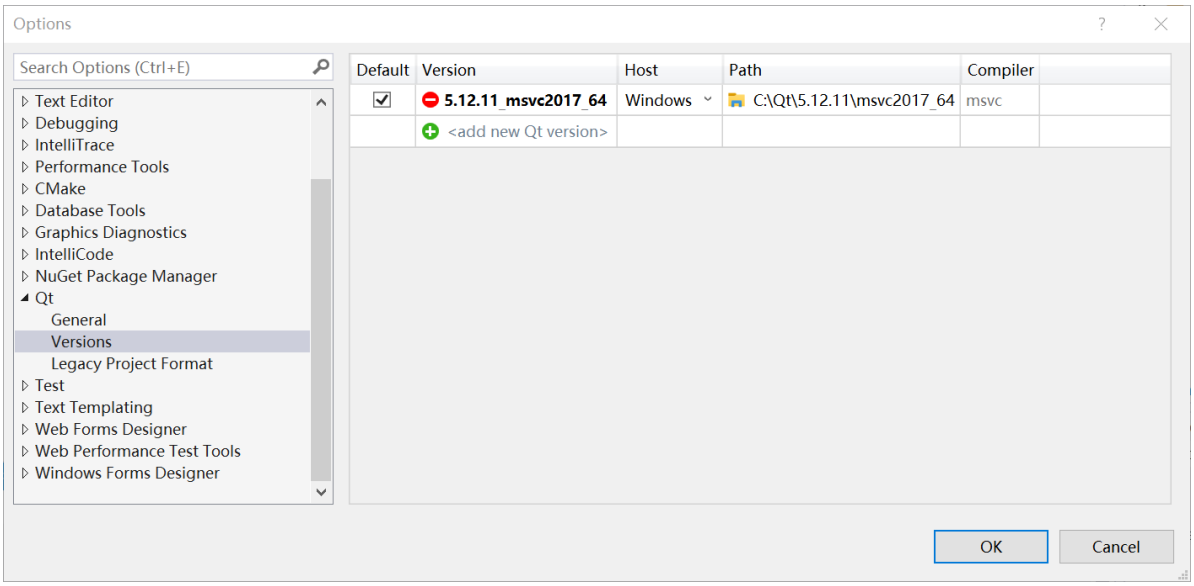
4.2 Qt VS Tools配置

Extensions - Qt VS Tools - Options

Qt - Versions -



Path 选择 qmake.exe文件（C:\Qt\5.12.11\msvc2017_64\bin\ qmake.exe）



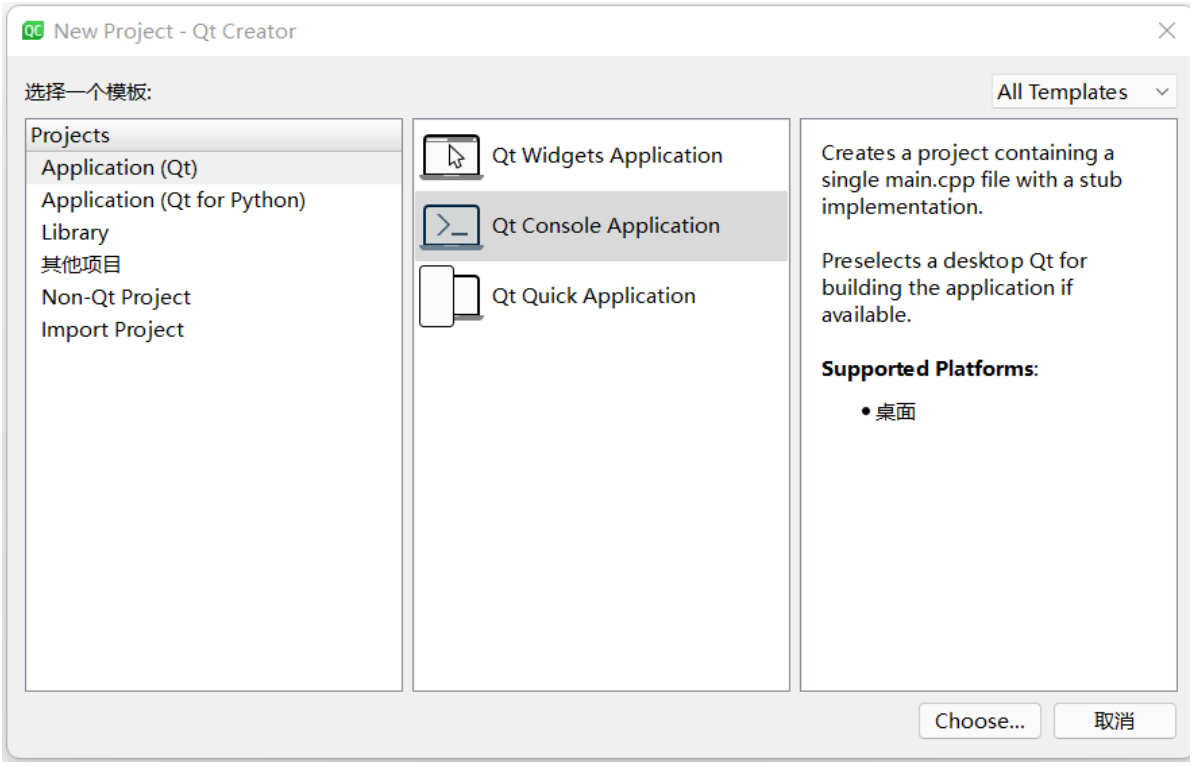
4.3 VS打开qt工程

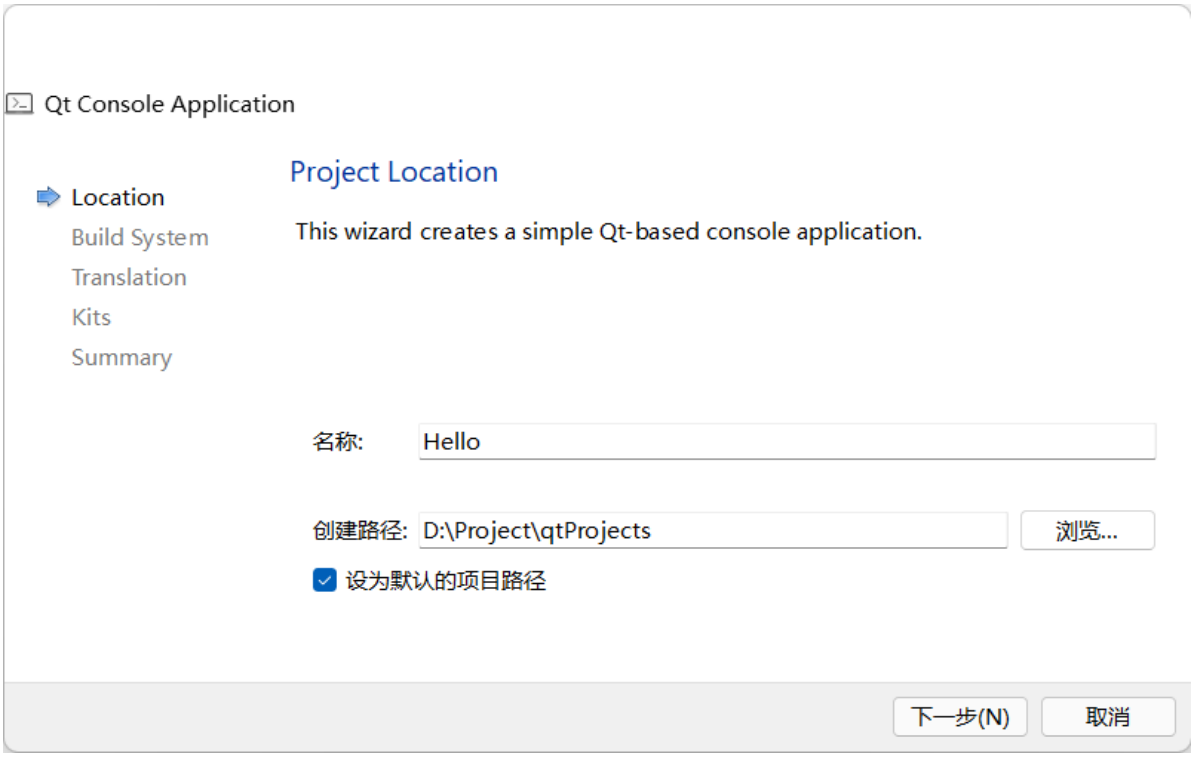
Extensions - Qt VS Tools - Open Qt Project File (.pro)

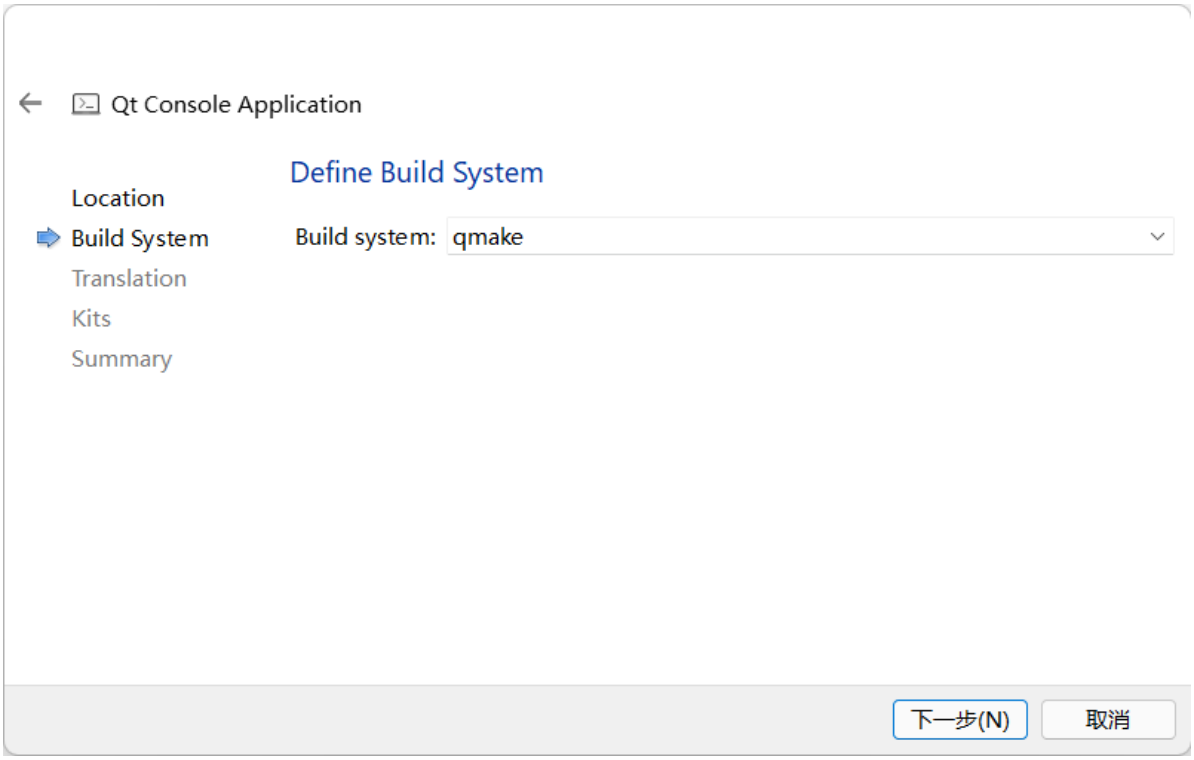
5. 第一个程序

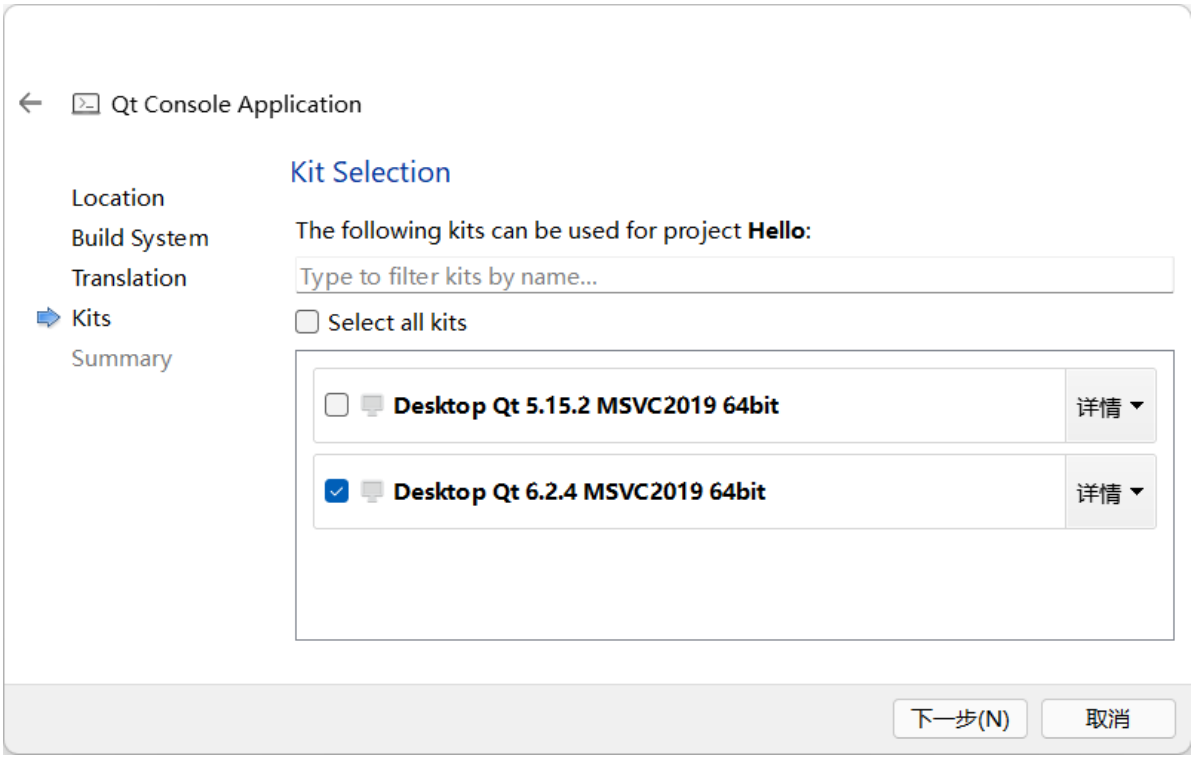
Qt Creator创建

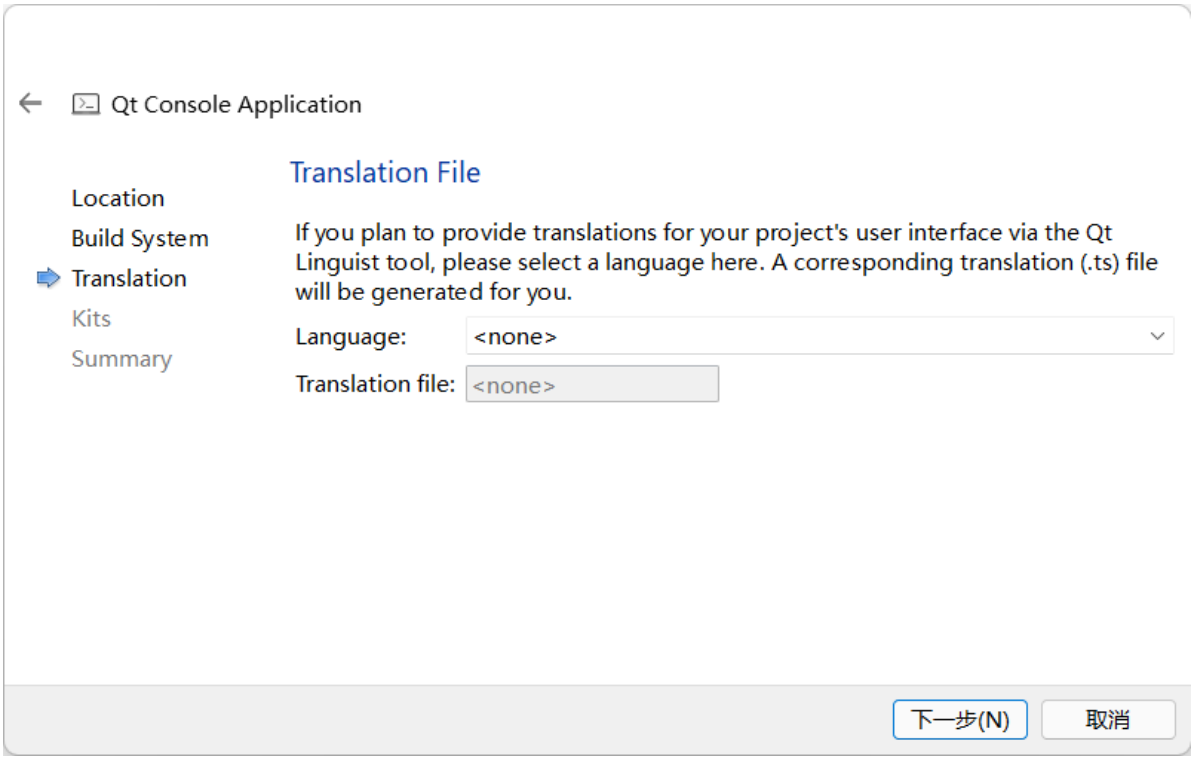
- Console Application



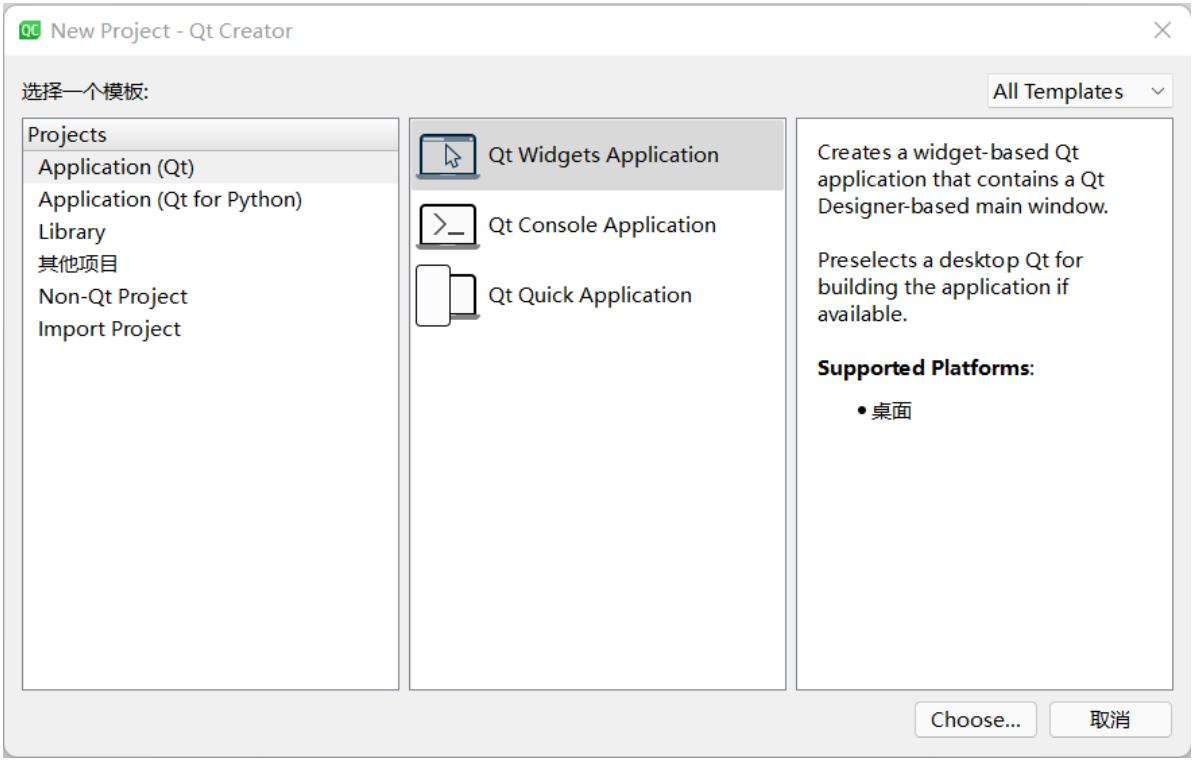


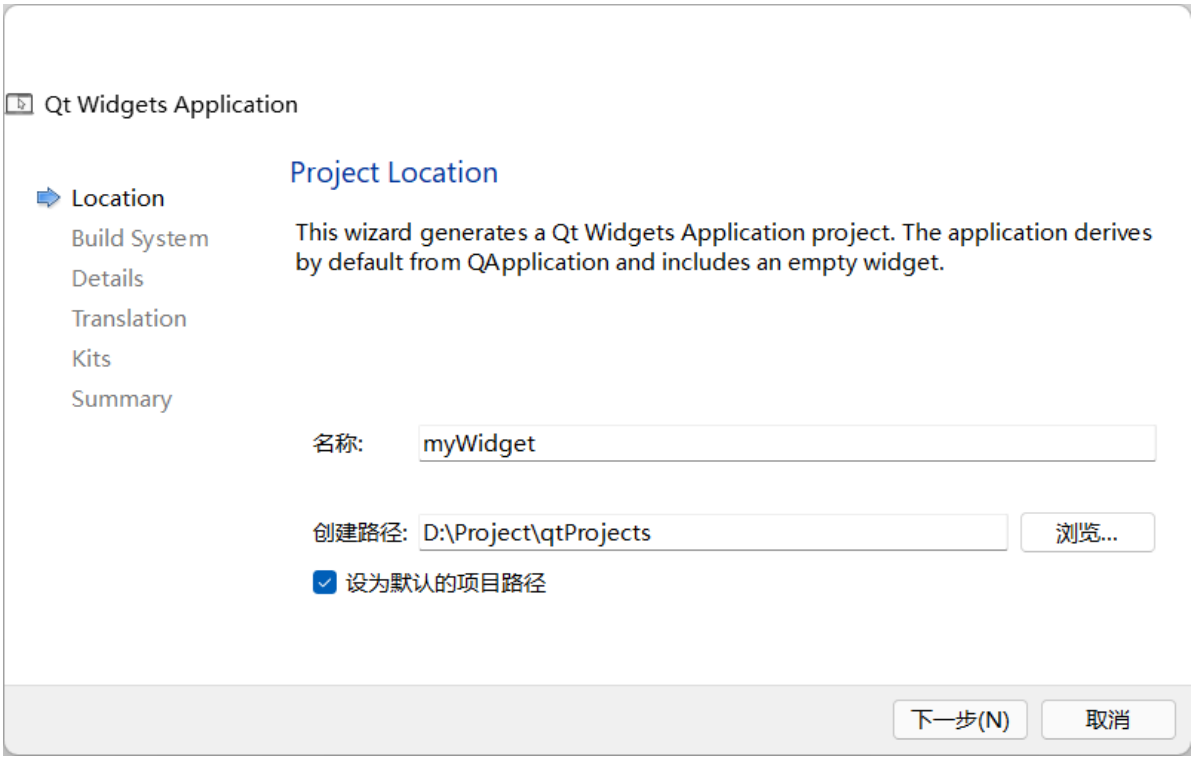

Qt Console Application
Define Build System
Location
Build System
Translation
Kits
Summary
Build system:
qmake
下一步(N)
取消

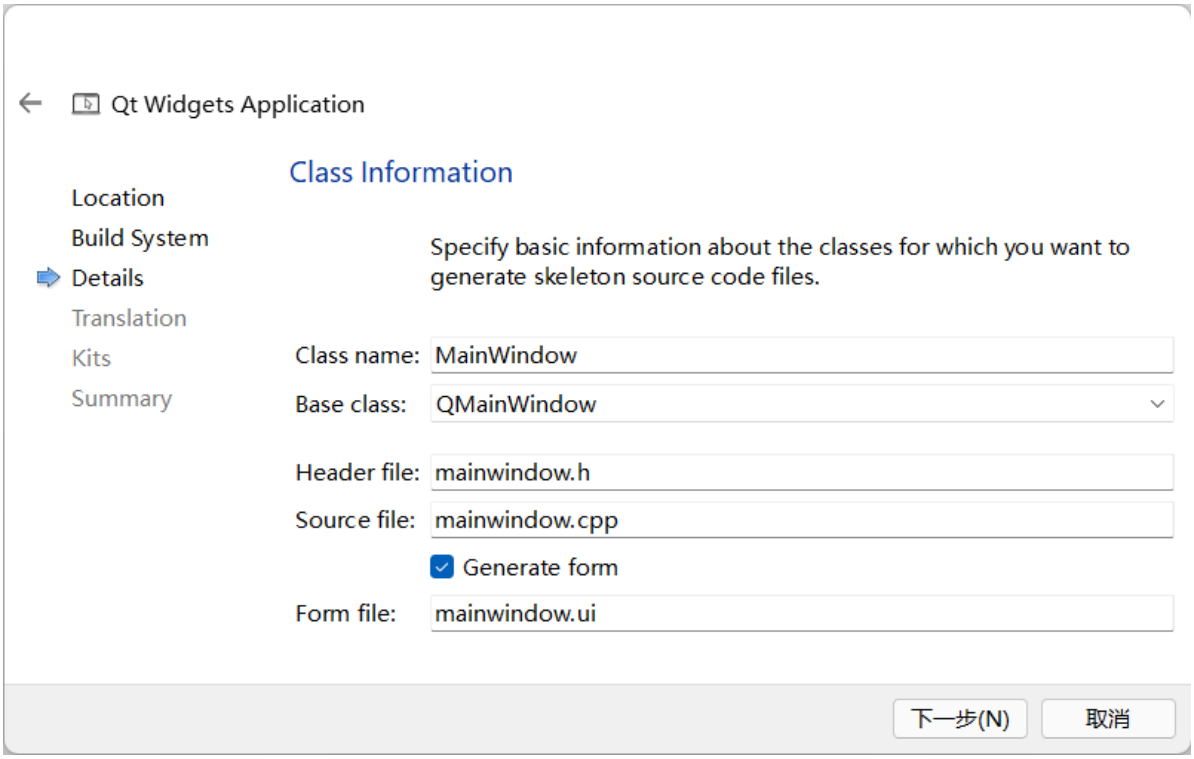

Qt Console Application
Kit Selection
Location
Build System
Translation
Kits
Summary
The following kits can be used for project Hello:
Type to filter kits by name...
Select all kits
Desktop Qt 5.15.2 MSVC2019 64bit
详情
Desktop Qt 6.2.4 MSVC2019 64bit
详情
下一步(N)
取消

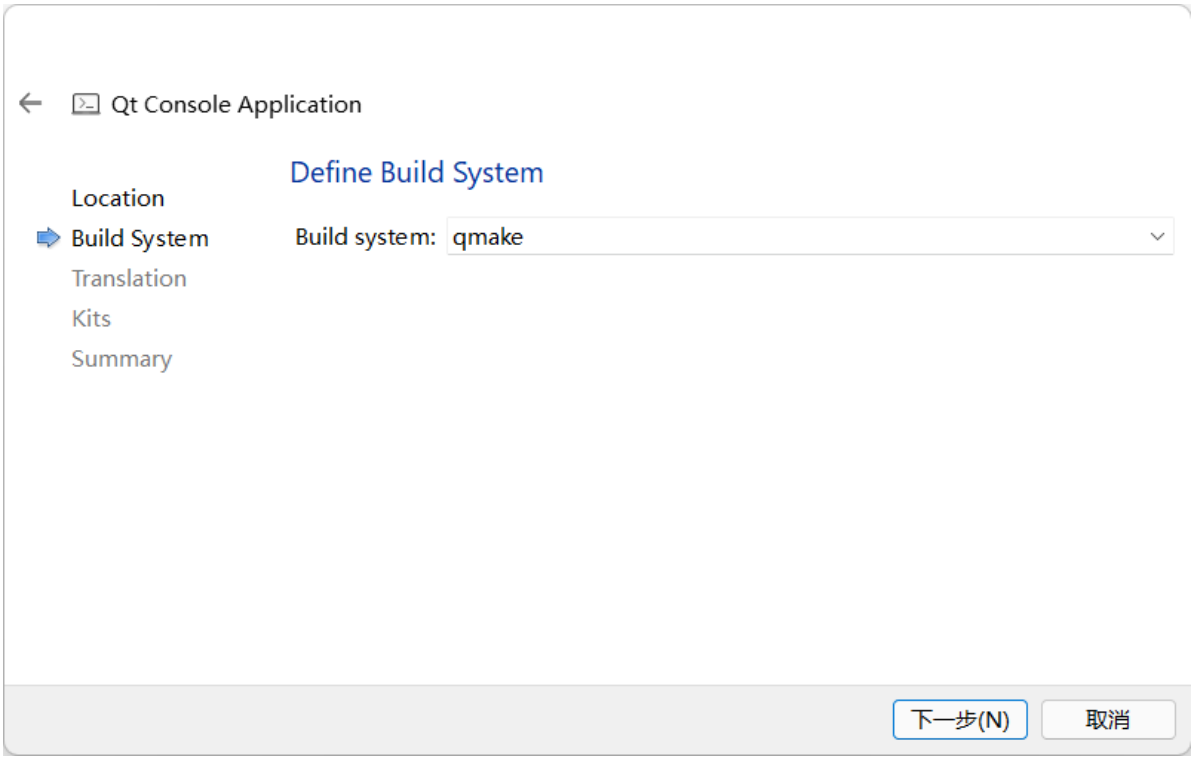


- Widgets Application

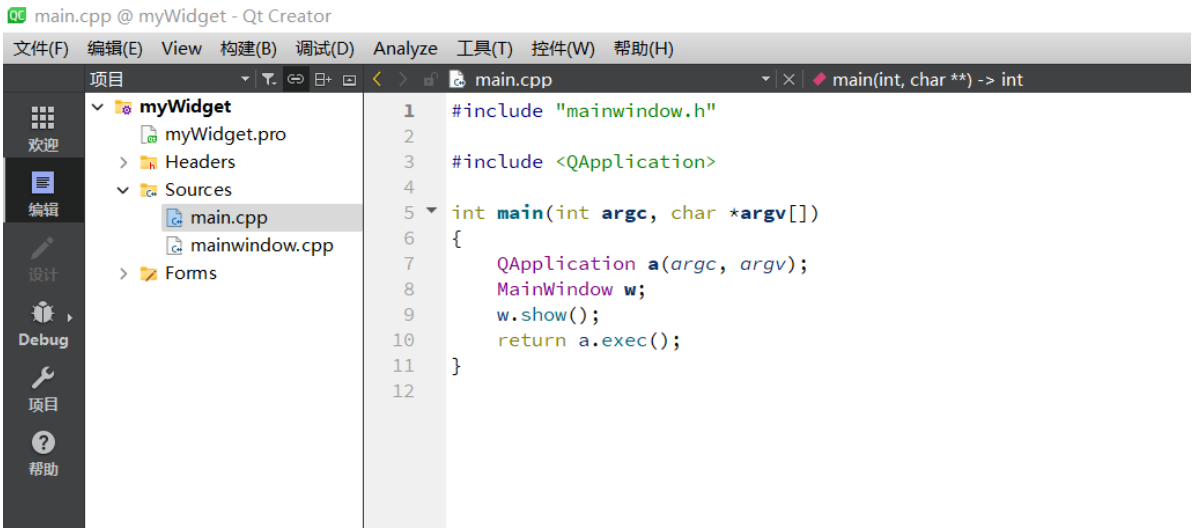



Qt Widgets Application
Location
Build System
Details
Translation
Kits
Summary
Project Location
This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.
名称:
myWidget
创建路径:
D:\Project\qtProjects
浏览...
设为默认的项目路径
下一步(N)
取消

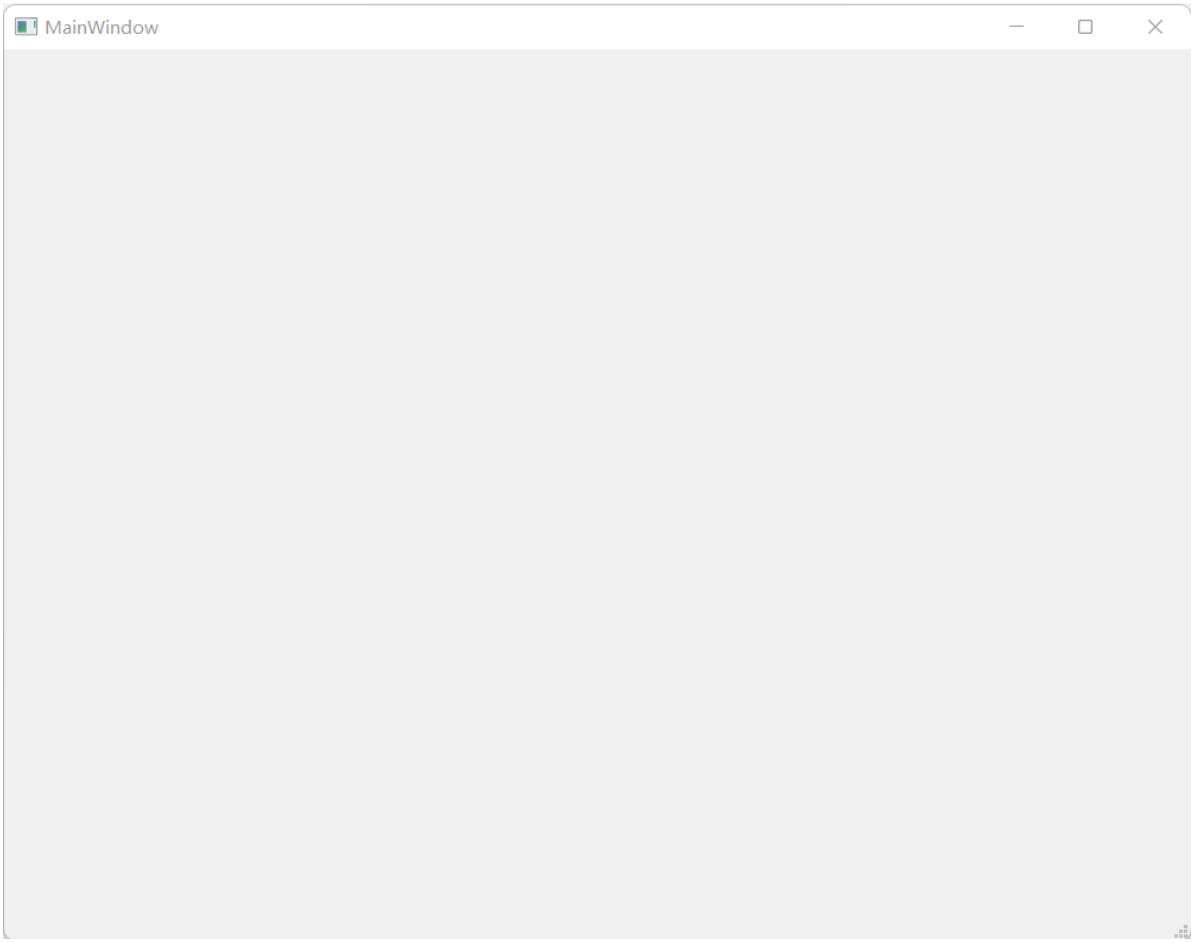

Qt Widgets Application
Location
Build System
Details
Translation
Kits
Summary
Class Information
Specify basic information about the classes for which you want to generate skeleton source code files.
Class name:
MainWindow
Base class:
QMainWindow
Header file:
mainwindow.h
Source file:
mainwindow.cpp
Generate form
Form file:
mainwindow.ui
下一步(N)
取消


Qt Console Application
Define Build System
Location
Build System
Translation
Kits
Summary
Build system:
qmake
下一步(N)
取消

代码


main.cpp @ myWidget - Qt Creator
文件(F) 编辑(E) View 构建(B) 调试(D) Analyze 工具(T) 控件(W) 帮助(H)
项目
main.cpp
main(int, char **) -> int
欢迎
编辑
设计
Debug
项目
帮助
myWidget
myWidget.pro
Headers
Sources
main.cpp
mainwindow.cpp
Forms
#include "mainwindow.h"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.show();
return a.exec();
}

运行结果



6 QMainWindow类型

main函数如下

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

MainWindow为QMainWindow派生的子类，GUI通过该子类对象来实现

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
private:
    void paintEvent(QPaintEvent *);
    void timerEvent(QTimerEvent *);
    void resizeEvent(QResizeEvent *event);
    void mousePressEvent(QMouseEvent *event);
private:
    Ui::MainWindow *ui;

```

```
    QSize   window_size;

};
```

6.1 绘制

绘制在void paintEvent(QPaintEvent*)之中，由专门的QPainter对象来实现

```
void MainWindow::paintEvent(QPaintEvent*)
{
    QPainter painter(this);
    //通过painter来实现绘制
}
```

画圆用QPainter::drawEllipse

```
void CCircle::Render(QPainter& painter)
{
    painter.drawEllipse(pos.x - radius, pos.y - radius, radius * 2, radius * 2);
}
```

画矩形用QPainter::drawRect

```
void CRectangle::Render(QPainter& painter)
{
    painter.drawRect(pos.x - width / 2, pos.y - height / 2, width, height);
}
```

绘制图像使用QImage类对象QPainter::drawImage

```
void ImageShape::Render(QPainter& painter)
{
    QRectF target(pos.x - width / 2, pos.y - height / 2, width, height); //建立目标矩形
    QRectF source(0.0, 0.0, width, height); //建立源矩形，用来框定源图像文件中要显示的部分
    painter.drawImage(target, image, source); //image是QImage对象
}
```

6.2 事件响应

继承自QWidget Class，

QWidget Class提供了事件响应机制https://doc.qt.io/qt-6/qwidget.html#events，提供了了键盘、鼠标、窗口改变等事件处理函数

```
QWidget::paintEvent() //绘制函数
QWidget::resizeEvent()//窗口大小改变
QWidget::mousePressEvent() //鼠标某个键按下
QWidget::mouseReleaseEvent()//鼠标某个键松开下
QWidget::mouseDoubleClickEvent() //双击
QWidget::mouseMoveEvent() //鼠标移动
QWidget::keyPressEvent()//键盘某个键按下
```

如处理鼠标左键按下，event->pos()得到鼠标当前位置

```
void MainWindow::mousePressEvent(QMouseEvent* event)
{
    QPoint pt = event->pos();
    g_interface.UserWndProc(0, pt.x(), pt.y());
}
```

7 其它注意事项

7.1 Qt Creator打开.pro工程

直接打开，编辑模式，选中项目，右键 - 执行qmake

然后可以正常使用。

- 右键 - 构建
- Build All Projects

7.2 Visual Studio打开.pro工程

如果有sln文件，直接打开。如果没有:

- 打开Visual Studio 2019， Continue without code
- Extension - Qt VS Tools - Open Qt Project File (.pro)