



2017

第1章 第一个Qt程序

目录

CONTENTE

01 Qt与Qt Creator

02 Qt 下载

03 Qt Creator简介

04 Qt 工具

05 第一个Qt程序

1、Qt与Qt Creator

- ◉ Qt是一个跨平台应用程序和UI开发框架，可以理解为开发包或者库文件。
- ◉ Qt Creator是全新的跨平台 Qt IDE（集成开发环境），可单独使用，也可与 Qt 库和开发工具组成一套完整的SDK（软件开发工具包）。其中包括：高级 C++ 代码编辑器、项目和生成管理工具、集成的上下文相关的帮助系统、图形化调试器、代码管理和浏览工具。

2、Qt、Qt Creator下载与安装

- ◉ Qt官网下载：<https://www.qt.io/download/>
- ◉ 资源下载页面：https://download.qt.io/official_releases/qt/5.8/5.8.0/

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
submodules/	20-Jan-2017 13:19	-	
single/	20-Jan-2017 13:14	-	
qt-opensource-windows-x86-wintr-msvc2015-5.8.0.exe	20-Jan-2017 12:54	1.2G	Details
qt-opensource-windows-x86-wintr-msvc2013-5.8.0.exe	20-Jan-2017 12:53	1.2G	Details
qt-opensource-windows-x86-msvc2015_64-5.8.0.exe	20-Jan-2017 12:52	1.0G	Details
qt-opensource-windows-x86-msvc2015-5.8.0.exe	20-Jan-2017 12:59	1.0G	Details
qt-opensource-windows-x86-msvc2013_64-5.8.0.exe	20-Jan-2017 12:51	958M	Details
qt-opensource-windows-x86-msvc2013-5.8.0.exe	20-Jan-2017 12:50	947M	Details
qt-opensource-windows-x86-mingw530-5.8.0.exe	20-Jan-2017 12:49	1.2G	Details
qt-opensource-windows-x86-android-5.8.0.exe	20-Jan-2017 12:48	1.3G	Details
qt-opensource-mac-x64-clang-5.8.0.dmg	20-Jan-2017 12:45	1.3G	Details
qt-opensource-mac-x64-android-ios-5.8.0.dmg	20-Jan-2017 12:44	3.4G	Details
qt-opensource-mac-x64-android-5.8.0.dmg	20-Jan-2017 12:40	1.4G	Details
qt-opensource-linux-x64-android-5.8.0.run	20-Jan-2017 12:34	817M	Details
qt-opensource-linux-x64-5.8.0.run	20-Jan-2017 12:34	766M	Details
md5sums.txt	20-Jan-2017 13:14	1.0K	Details

2、 Qt、 Qt Creator下载与安装

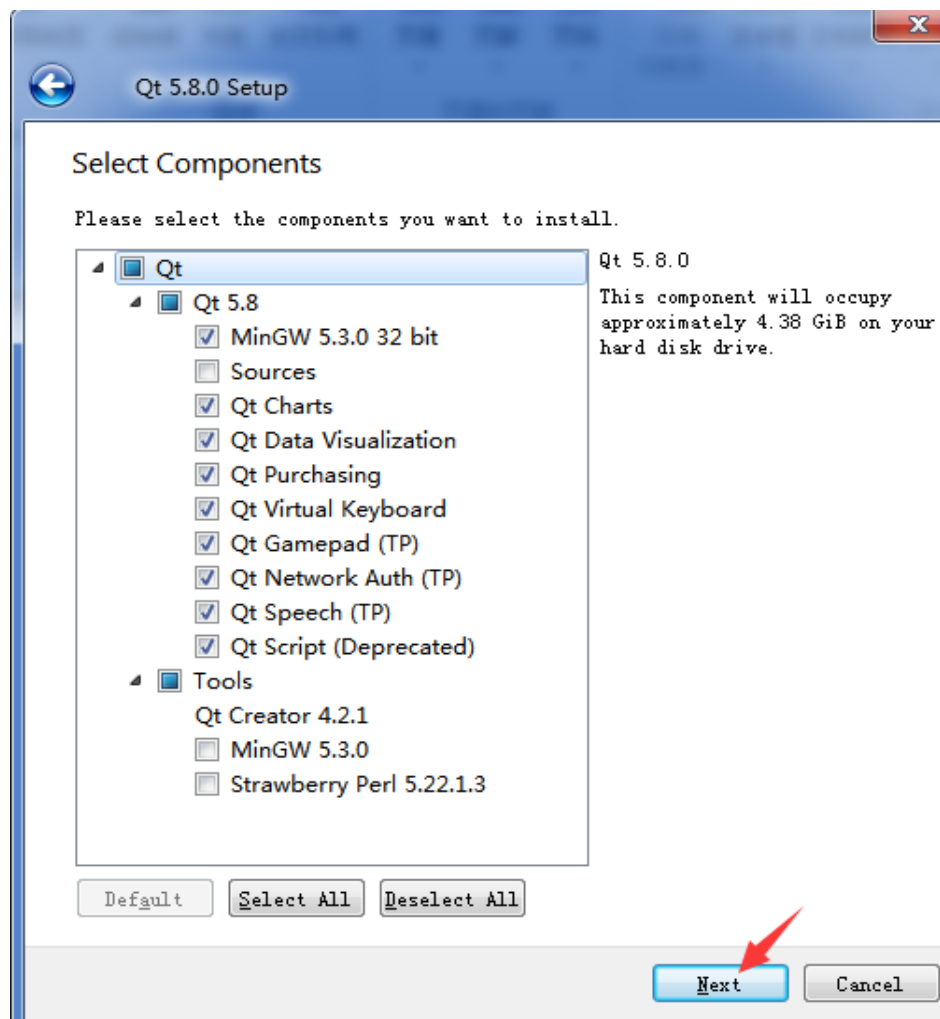
以下下载qt-opensource-windows-x86-mingw530-5.8.0.exe为例：

- ◉ `opensource`表示开源版本
 - ◉ `windows-x86`表示Windows平台
 - ◉ `mingw530`表示使用MinGW530编译器
 - ◉ `5.8.0`是当前版本号
-
- MinGW即Minimalist GNU For Windows，是将GNU开发工具移植到Win32平台下的产物，是一套Windows上的GNU工具集。用其开发的程序不需要额外的第三方DLL支持就可以直接在Windows下运行。

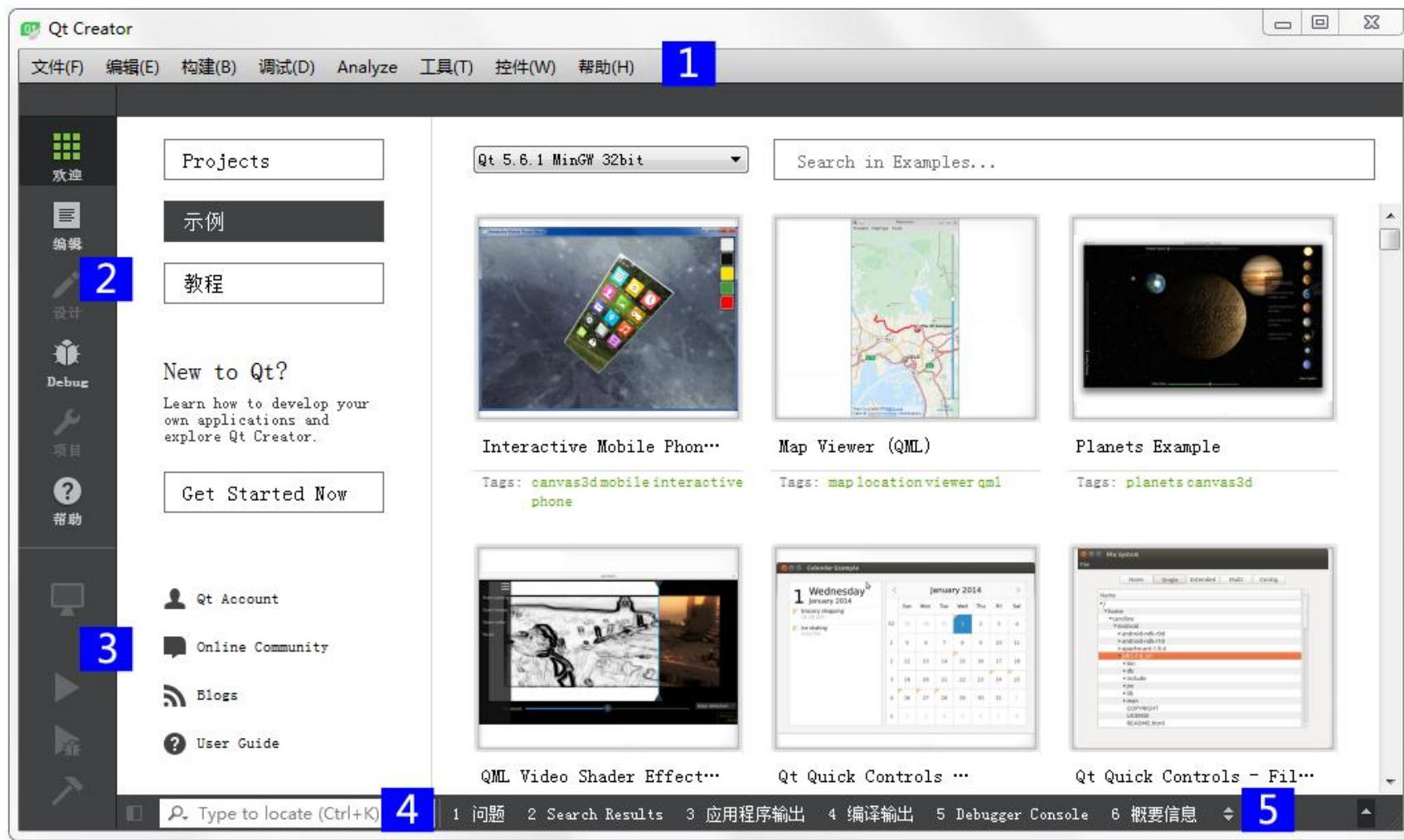
2、Qt、Qt Creator下载与安装

安装注意：

- 安装路径中不能有中文。
- 安装开始是否登陆或者注册Qt账号，不影响程序的安装，可以直接跳过。
- 在选择组件界面，可以选择安装一些模块，单击一个组件，可以在右侧显示该组件的详细介绍，初学者建议保持默认选择。



3、Qt Creator简介



- 1. 菜单栏
- 2. 模式选择
- 3. 构建套件
- 4. 定位器
- 5. 输出窗格

3、Qt Creator简介

1) 菜单栏 (Menu Bar)

- 有8个菜单选项，包含了常用的功能菜单：文件菜单、编辑菜单、构建菜单、调试菜单、analyze分析菜单、工具菜单、工具菜单、帮助菜单。
- 工具菜单提供了快速定位菜单、版本控制工具菜单和外部工具菜单等。里面的选项菜单中包含了Qt Creator各个方面的设置选项：环境设置、文本编辑器设置、帮助设置、构建和运行设置、调试器设置和版本控制设置等。

2) 模式选择器 (Mode Selector)

- 包含欢迎、编辑、设计、调试、项目和帮助6个模式，各个模式完成不同的功能，也可以使用快捷键来更换模式，它们对应的快捷键依次是Ctrl + 数字1~6。

3、 Qt Creator简介

3) 构建套件选择器 (Kit Selector)

- 包含了目标选择器 (Targetselector)、运行按钮 (Run)、调试按钮 (Debug) 和构建按钮 (Building) 4个图标。
- 目标选择器用来选择要构建哪个项目，使用哪个Qt库，这对于多个Qt库的项目很有用。还可以选择编译项目的debug版本或是release版本。
- 运行按钮可以实现项目的构建和运行；
- 调试按钮可以进入调试模式，开始调试程序；
- 构建按钮完成项目的构建。

3、 Qt Creator简介

4) 定位器 (Locator)

- 在QtCreator中可以使用定位器来快速定位项目、文件、类、方法、帮助文档以及文件系统。
- 可以使用过滤器来更加准确地定位要查找的结果。
- 可以在“工具→选项”菜单项中设置定位器的相关选项。

3、 Qt Creator简介

5) 输出窗格 (Output panes)

- 这里包含了问题、搜索结果、应用程序输出、编译输出、Debugger Console、概要信息、版本控制7个选项，它们分别对应一个输出窗口，相应的快捷键依次是Alt + 数字1~7。
- 问题窗口显示程序编译时的错误和警告信息；
- 搜索结果窗口显示执行了搜索操作后的结果信息；
- 应用程序输出窗口显示在应用程序运行过程中输出的所有信息；
- 编译输出窗口显示程序编译过程输出的相关信息；
- 版本控制窗口显示版本控制的相关输出信息。

4、Qt 工具

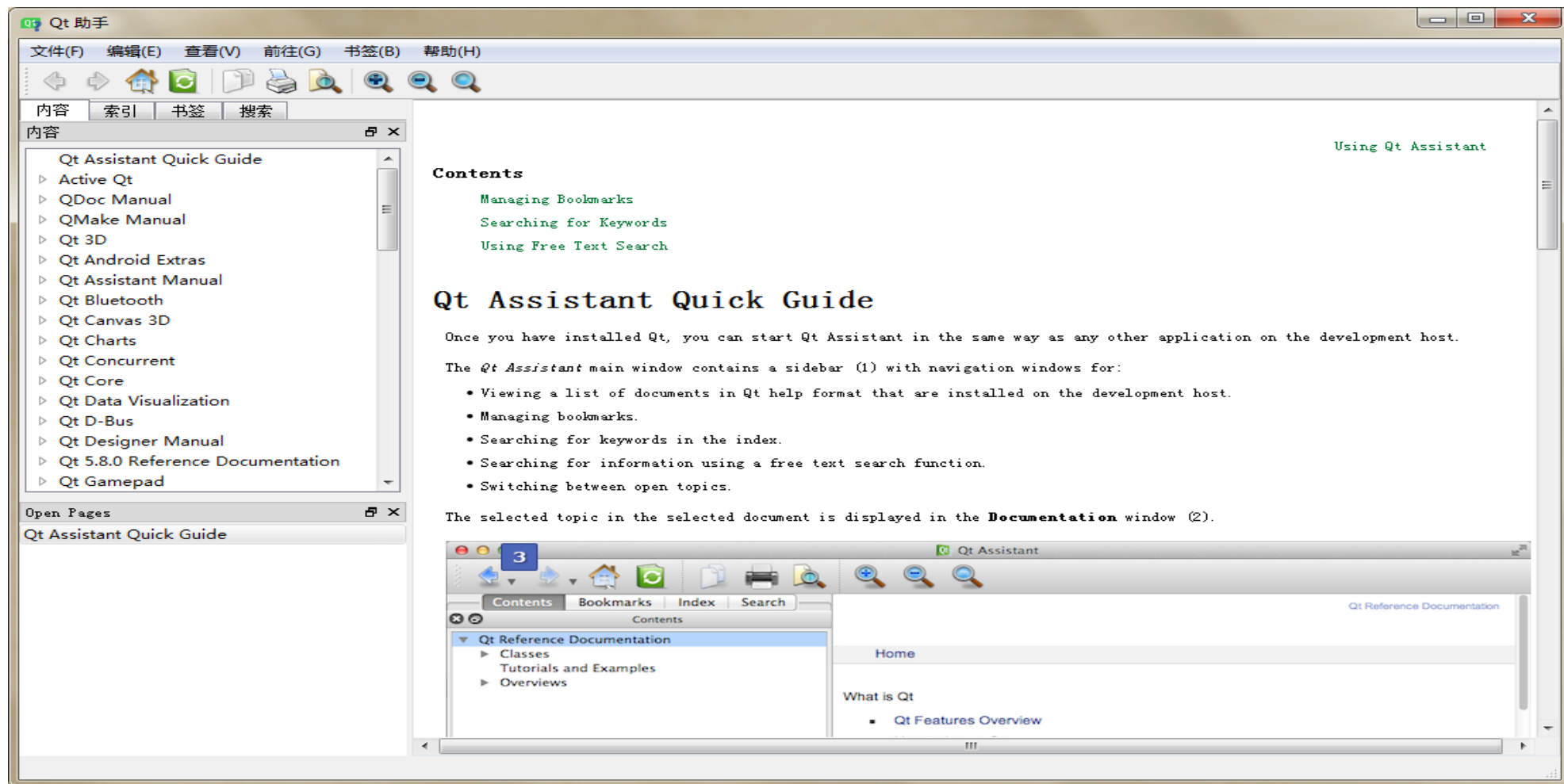
Qt中包含了几个很有用的工具，分别是：

- Qt Assistant (Qt助手)
 - Qt Linguist (Qt语言家)
 - Qt Designer (Qt设计师)
-
- 可以从开始菜单启动它们；也可以在安装目录下找到
 - 例如C:\Qt\Qt5.8.0\5.8\mingw530\bin。

4、Qt 工具

- Qt Assistant是可配置且可重新发布的文档阅读器，可以方便地进行定制并与Qt应用程序一起重新发布。Qt Assistant已经被整合进Qt Creator，就是里面的Qt帮助

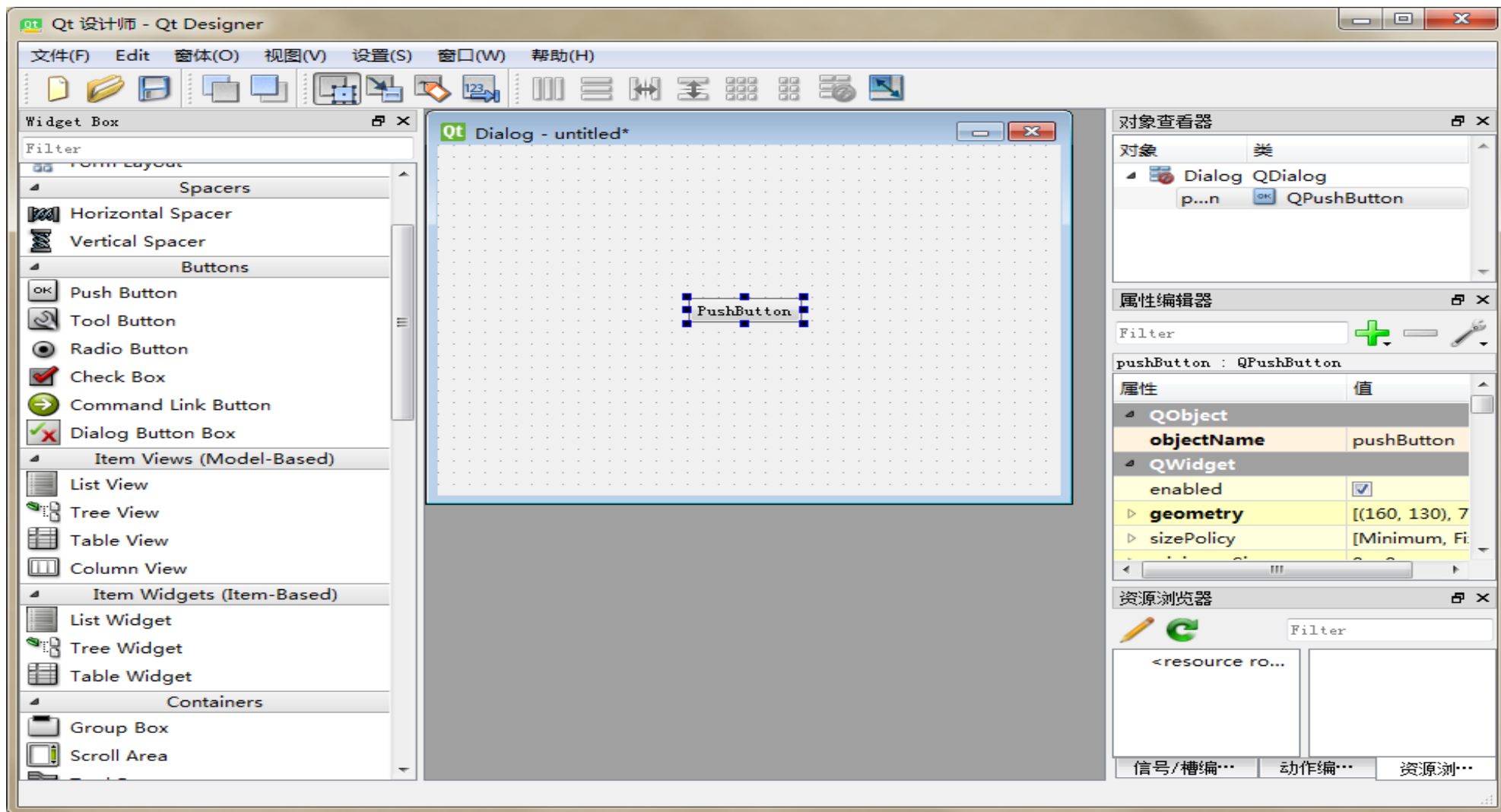
4、Qt 工具



4、Qt 工具

- Qt Designer是强大的跨平台GUI布局和格式构建器。由于使用了与应用程序中将要使用的相同的部件，可以使用屏幕上的格式快速设计、创建部件以及对话框。使用Qt Designer创建的界面样式功能齐全并可以进行预览，这样就可确保其外观完全符合要求 。

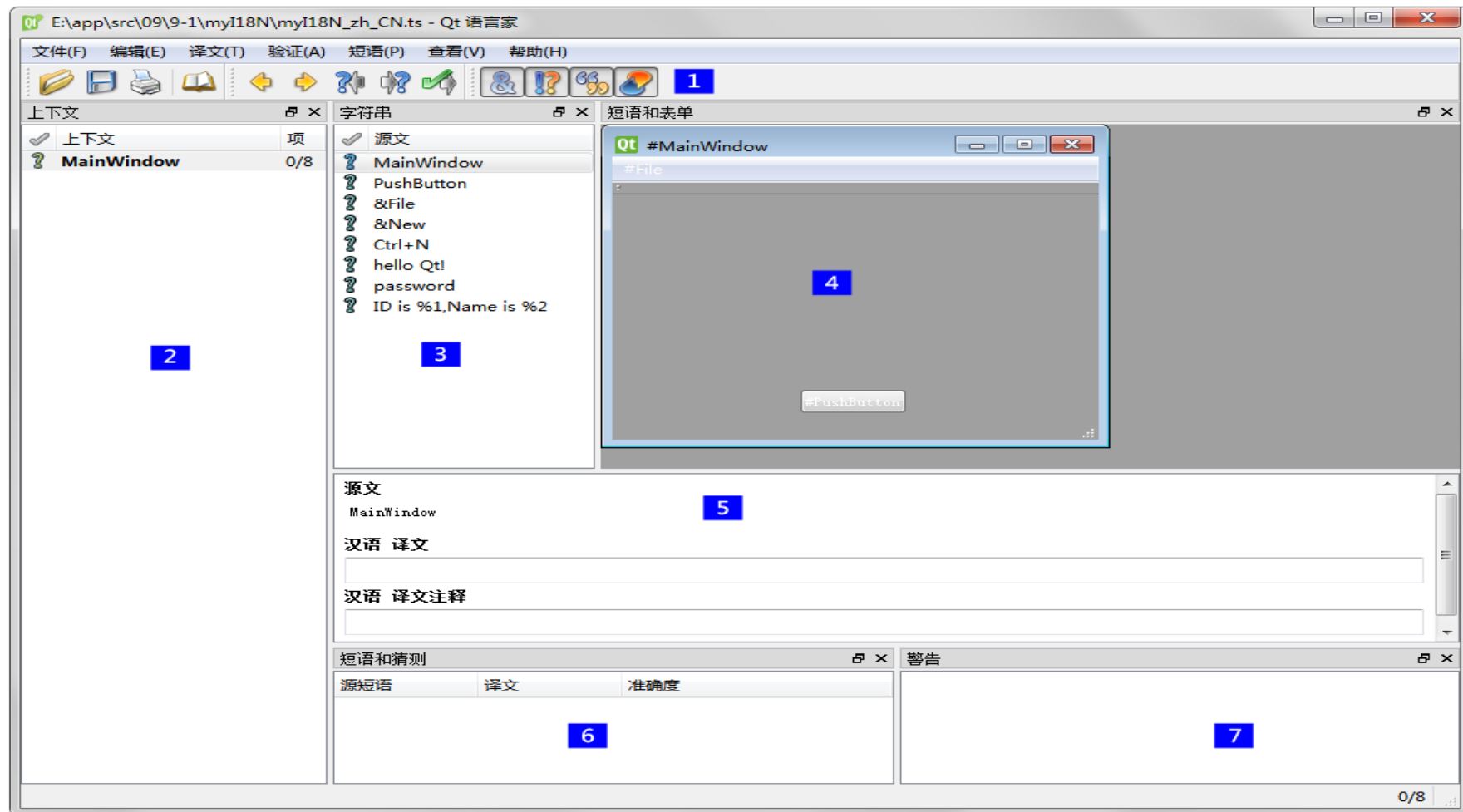
4、Qt 工具



4、Qt 工具

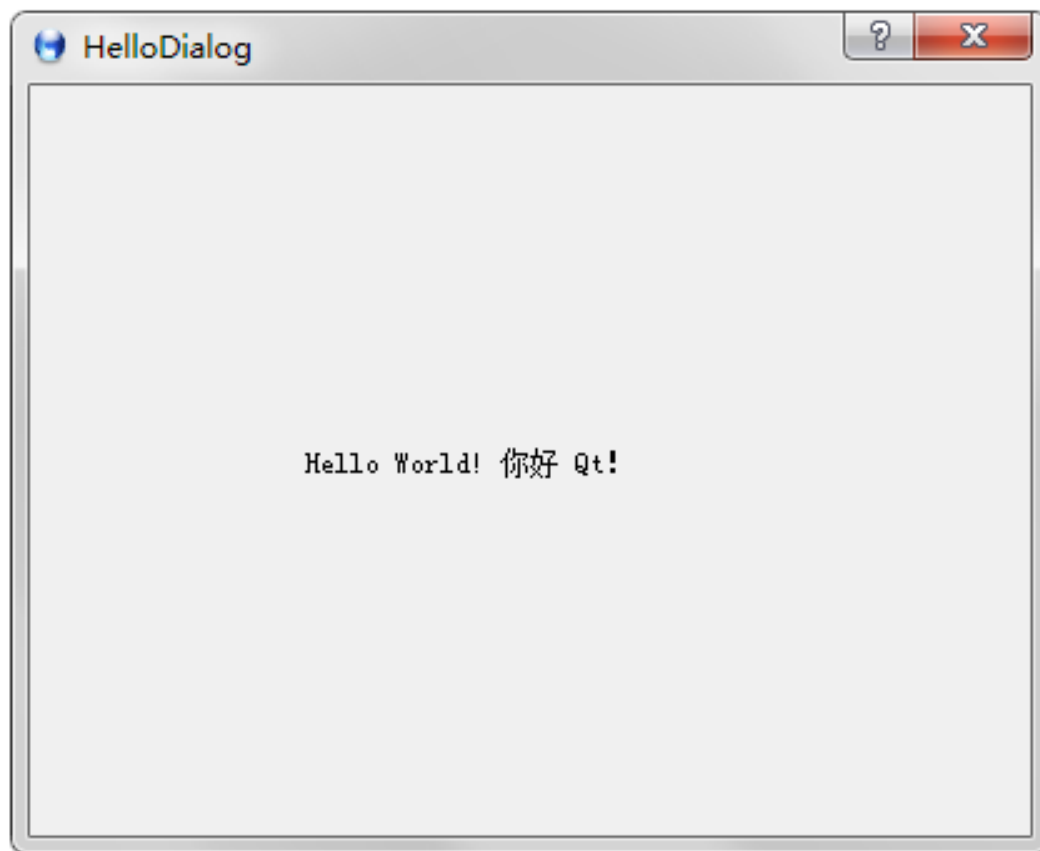
- Qt Linguist 提供了一套加速应用程序翻译和国际化的工具。Qt 使用单一的源码树和单一的应用程序二进制包就可同时支持多个语言和书写系统。

4、Qt 工具



5、 第一个Qt程序

创建图形界面应用程序，显示“Hello World!”。



5、 第一个Qt程序

5.1 新建Qt程序、构建与运行Qt程序

5.2 发布程序

5.1、新建Qt程序

新建Qt程序两种方式：

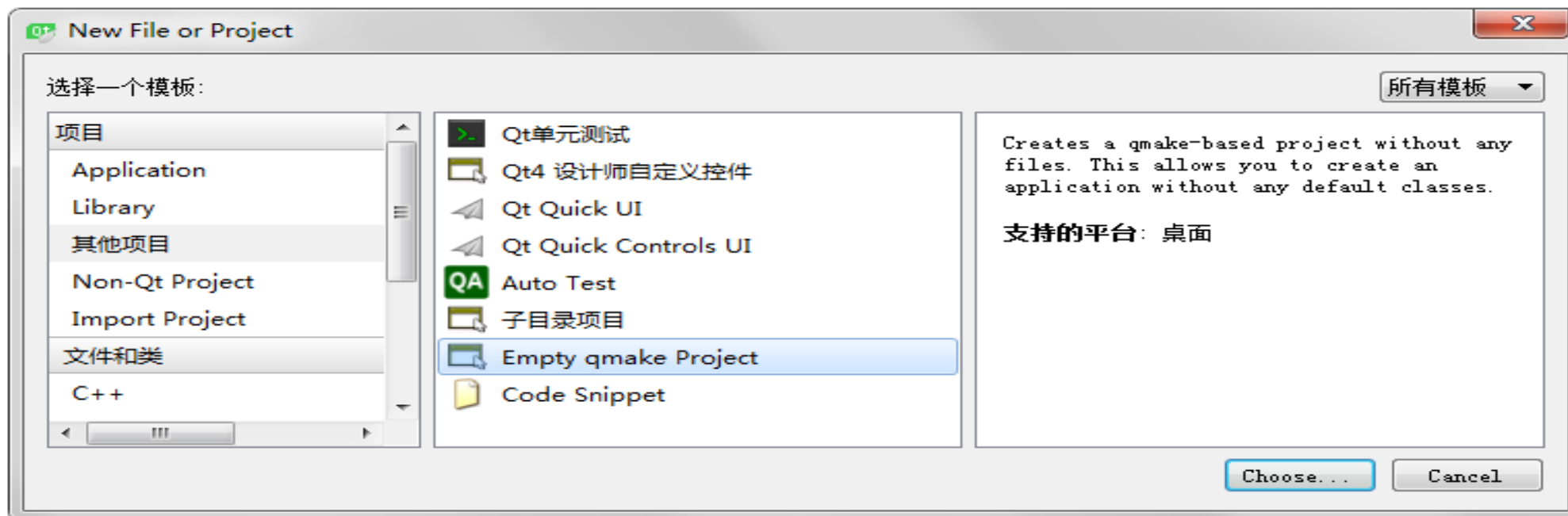
- 一、使用纯代码方式编写
- 二、使用.ui文件

5.1、 新建Qt程序

- 一、使用纯代码方式编写

5.1、新建Qt程序

- 1. 新建空项目。打开Qt Creator，并新建项目，选择“其他项目”中的“Empty qmake Project”。然后将项目命名为helloworld并设置路径，例如E:\app\src\02\2-2。完成后，双击helloworld.pro文件，添加如下一行代码：`greaterThan(QT_MAJOR_VERSION, 4): QT += widgets`



5.1、新建Qt程序

- 2. 往项目中添加 main.cpp 文件。在项目文件列表中的项目文件夹 helloworld 上右击，选择“添加新文件”一项，然后选择 C++ Source File，名称设置为 main.cpp，路径就是默认的项目目录，后面的选项保持默认即可。

```
1  #include <QApplication>
2  #include <QDialog>
3  #include <QLabel>
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      QDialog w;
8      QLabel label(&w);
9      label.setText("Hello World! 你好Qt! ");
10     w.show();
11     return a.exec();
12 }
```


5.1、新建Qt程序

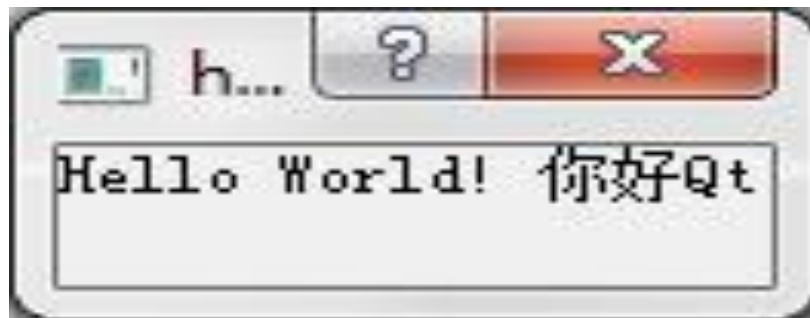
- 前3行是头文件包含。在Qt中每一个类都有一个与其同名的头文件，因为后面用到了QApplication、QDialog和QLabel这3个类，所以这里要包含这些类的定义。
- 第4行就是在C++中最常见到的main()函数，它有两个参数，用来接收命令行参数。
- 在第6行新建了QApplication类对象，用于管理应用程序的资源，任何一个Qt GUI程序都要有一个QApplication对象。因为Qt程序可以接收命令行参数，所以它需要argc和argv两个参数。

5.1、新建Qt程序

- 第7行新建了一个QDialog对象，QDialog类用来实现一个对话框界面。
- 第8行新建了一个QLabel对象，并将QDialog对象作为参数，表明了对话框是它的父窗口，也就是说这个标签放在对话框窗口中。
- 第9行给标签设置要显示的字符。
- 第10行让对话框显示出来。在默认情况下，新建的可视部件对象都是不可见的，要使用show()函数让它们显示出来。
- 第11行让QApplication对象进入事件循环，这样当Qt应用程序在运行时便可以接收产生的事件，例如单击和键盘按下等事件。

5.1、 新建Qt程序

- 4. 编译运行。



5.1、新建Qt程序

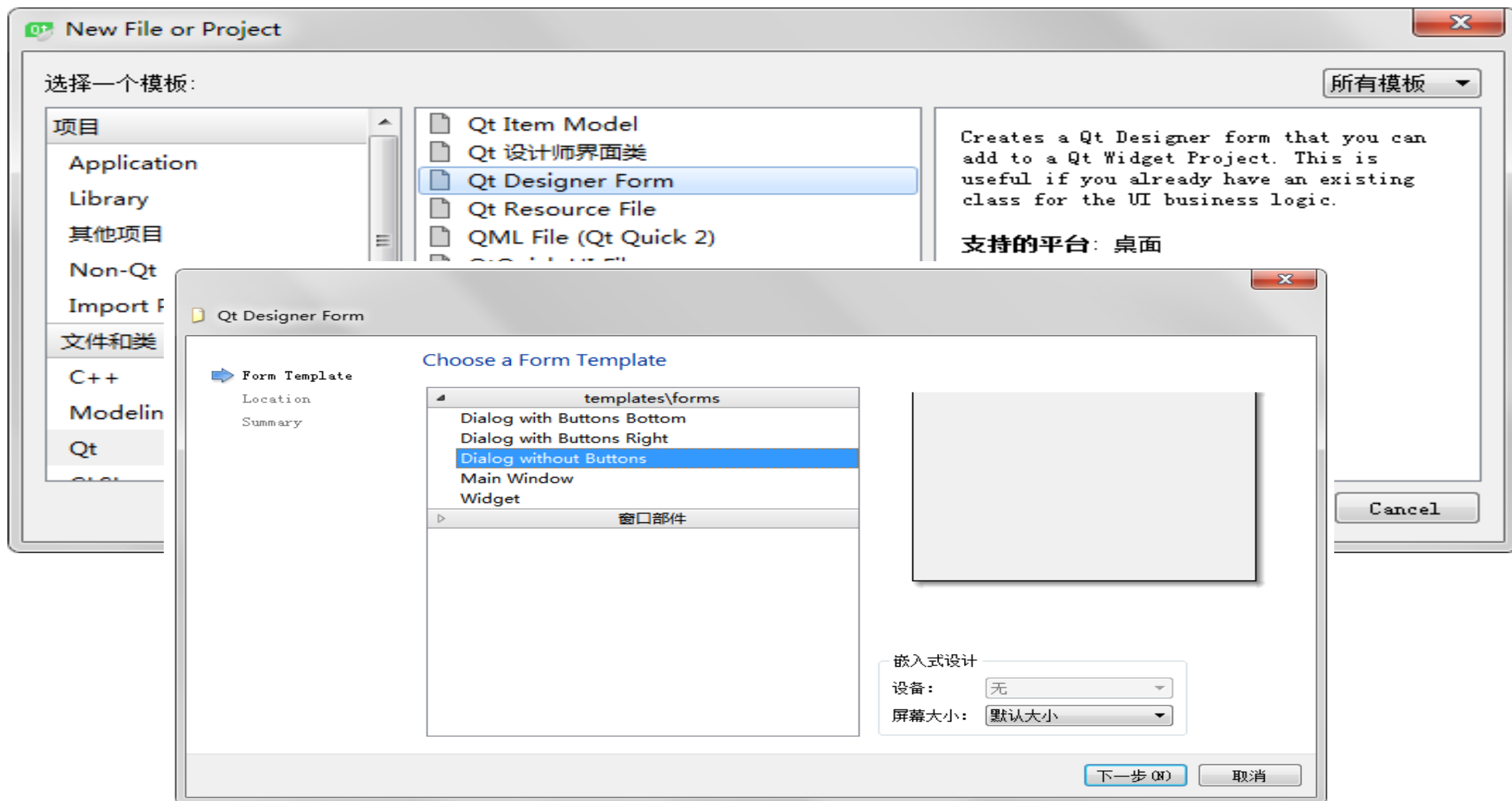
5. 设置窗口大小。

```
1  #include <QApplication>
2  #include <QDialog>
3  #include <QLabel>
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      QDialog w;
8      w.resize(400, 300);
9      QLabel label(&w);
10     label.move(120, 120);
11     label.setText(QObject::tr("Hello World! 你好Qt! "));
12     w.show();
13     return a.exec();
14 }
```

5.1、新建Qt程序

- 二、使用ui方式
- 1. 添加.ui文件。向工程中继续添加文件。在模板中选择Qt中的“Qt Designer Form”。在选择界面模板时选择Dialog without Buttons项。再单击“下一步”，将文件名称改为hellodialog.ui。

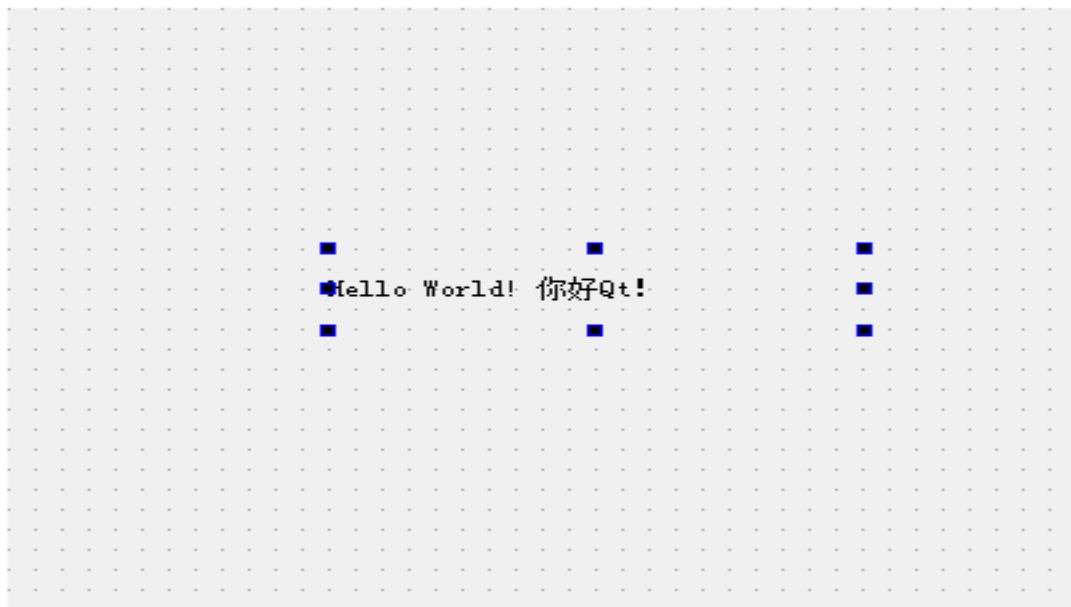
5.1、新建Qt程序



5.1、新建Qt程序

- 2. 设计界面。生成好文件后便进入了设计模式，在界面上添加一个Label部件，并且更改其显示内容为“Hello World! 你好Qt! ”。
- 在右侧的属性栏的geometry属性中更改其坐标位置为“X: 120, Y: 120”。这样就与那行代码“`label.move(120, 120);`”起到了相同的作用。
- 在右上角的类列表中选择QDialog类对象，并且在下面的属性中更改它的对象名objectName为“HelloDialog”。


5.1、新建Qt程序



label : QLabel	
属性	值
▾ QObject	
objectName	label
▾ QWidget	
enabled	<input checked="" type="checkbox"/>
▾ geometry	[(120, 120), 201 x 41]
X	120
Y	120

对象	类
▾ HelloDialog	
label	QLabel
Filter	
HelloDialog : QDialog	
属性	值
▾ QObject	
objectName	HelloDialog

5.1、新建Qt程序

- 3. 生成ui头文件。这时按下Ctrl+S快捷键保存修改，然后按下Ctrl+2快捷键回到编辑模式，那么就会看到.ui文件的内容了，它是一个XML文件，里面是界面部件的相关信息。使用Ctrl+Shift+B快捷键或者左下角的  图标来构建工程。然后到本地磁盘的项目目录的 build-helloworld-Desktop_Qt_5_6_1_MinGW_32bit-Debug目录中，就可以看到由.ui文件生成的ui_hellodialog.h头文件了。

5.1、新建Qt程序

4. 更改main.cpp文件。

```
#include "ui_hellodialog.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QDialog w;
    Ui::HelloDialog ui;
    ui.setupUi(&w);
    w.show();
    return a.exec();
}
```

- 使用命名空间Ui中的HelloDialog类定义了一个ui对象。
- setupUi()函数，并将对话框类对象作为参数，这样就可以将设计好的界面应用到对象w所表示的对话框上了。

5.1、 新建Qt程序

5. 构建运行

5.1、新建Qt程序

文件	说明
helloworld.pro	该文件是项目文件，其中包含了项目相关信息
helloworld.pro.user	该文件中包含了与用户有关的项目信息
hellodialog.h	该文件是新建的 HelloDialog 类的头文件
hellodialog.cpp	该文件是新建的 HelloDialog 类的源文件
main.cpp	该文件中包含了 main() 主函数
hellodialog.ui	该文件是设计师设计的界面对应的界面文件

5.1、新建Qt程序

helloworld.pro

```
1 #-----
2 #
3 # Project created by QtCreator 2016-07-16T21:45:42
4 #
5 #-----
6
7 QT      += core gui
8
9 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
10
11 TARGET = helloworld
12 TEMPLATE = app
13
14
15 SOURCES += main.cpp\
16          hellodialog.cpp
17
18 HEADERS  += hellodialog.h
19
20 FORMS    += hellodialog.ui
21
22 RC_ICONS = myico.ico
```

第7行表明了这个项目使用的模块。core模块包含了Qt的核心功能，其他所有模块都依赖于这个模块；而gui模块提供了窗口系统集成、事件处理、OpenGL和OpenGL ES集成、2D图形、基本图像、字体和文本等功能。

第9行添加了widgets模块，在Qt Widgets模块中提供了经典的桌面用户界面的UI元素集合，简单来说，所有C++程序用户界面部件都在该模块中。

第11行是生成的目标文件的名称，就是生成的exe文件的名称，默认的是项目的名称，当然也可以在这里改为别的名称。第12行使用app模板，表明这是个应用程序。

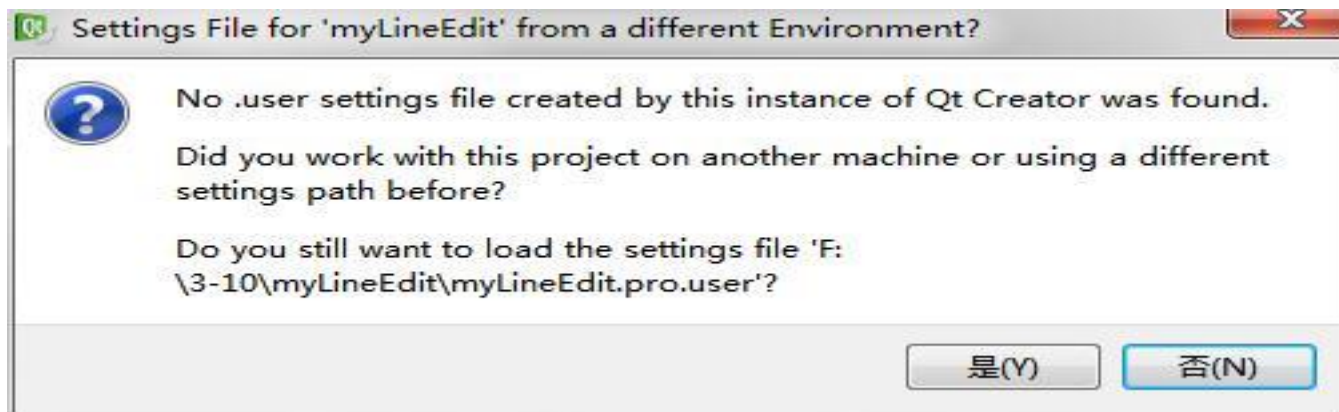
第15，18和20行分别是工程中包含的源文件、头文件和界面文件。

第22行就是添加的应用程序图标。这里这些文件都使用了相对路径，因为都在项目目录中，所以只写了文件名。

5.1、新建Qt程序

pro.user文件

包含了本地构建信息，包含Qt版本和构建目录等。可以用记事本或者写字板将这个文件打开查看其内容。当使用Qt Creator打开一个.pro文件时会自动生成一个.pro.user文件。因为读者的系统环境都不太一样，Qt的安装于设置也不尽相同，所以如果要将自己的源码公开，一般不需要包含这个user文件。如果要打开别人的项目文件，但里面包含了user文件，Qt Creator则会弹出提示窗口，询问是否载入特定的环境设置，这时应该选择“否”，然后选择自己的Qt版本即可。



5.2、发布程序

自学

作业

1. 下载安装Qt
2. 运行示例程序，并发布