

# Qt入门学习——Qt Creator 中 ui 文件和 Qt 代码关系

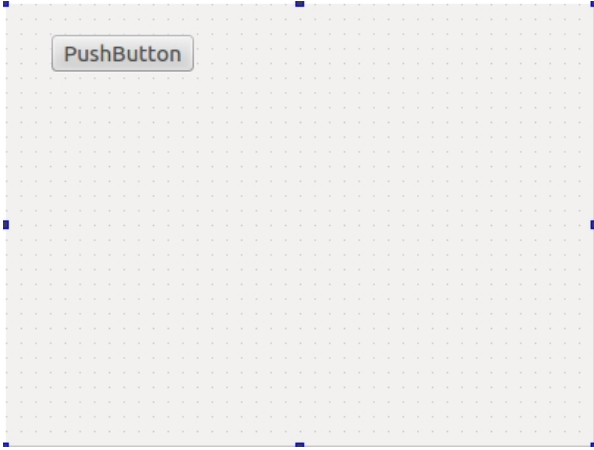
≡

分类: 【图形界面编程】 (53) ▼

版权声明: 本博客文章, 大多是本人整理编写, 或在网络中收集, 转载请注明出处!!

通过《Qt Creator的使用》的学习, 我们可以借助 Designer (界面设计器) 快速设计界面。

此例子 ui 内容如下 (只是简单添加了一个按钮):



工程的代码目录结构如下:



最终在工程所在目录会生成一个 ui 文件:



此 ui 文件实际上是xml 文件:

```
[WidgetWithUi]ls
main.cpp      mywidget.h    WidgetWithUi.pro
mywidget.cpp  mywidget.ui    WidgetWithUi.pro.user
[WidgetWithUi]cat mywidget.ui
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MyWidget</class>
  <widget class="QWidget" name="MyWidget">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>300</height>
      </rect>
    </property>
```

ui 文件内容为xml

当我们编译 Qt 程序代码，Qt Creator 用 uic 工具把 ui 文件的内容转换成 C++ 代码，在工程目录同一级目录的 build- 目录下自动生成 ui\_类名.h 文件，如本例子中的 ui\_mywidget.h，是由 mywidget.ui 利用uic 工具转换而成，只要通过Designer 修改了图形界面里的内容，ui\_mywidget.h 也会跟着自动同步更新内容，ui\_mywidget.h 是自动生成，用户写代码时不用关心其实现过程：

```
[qt_code]ls
build-WidgetWithUi-Desktop_Qt_5_4_2_GCC_32bit-Debug  WidgetWithUi
[qt_code]cd build-WidgetWithUi-Desktop_Qt_5_4_2_GCC_32bit-Debug/
[build-WidgetWithUi-Desktop_Qt_5_4_2_GCC_32bit-Debug]ls
main.o      moc_mywidget.cpp  mywidget.o      WidgetWithUi
Makefile    moc_mywidget.o    ui_mywidget.h
[build-WidgetWithUi-Desktop_Qt_5_4_2_GCC_32bit-Debug]
```

接下来我们一起分析一下此 ui\_mywidget.h 和 Qt 程序代码如何关联起来（即 ui 文件和 Qt 代码关系）。

mywidget.h 中自动多了一个命名空间的声明，类中多了 ui 指针对象成员：

WidgetWithUi

WidgetWithUi.pro

头文件

mywidget.h

源文件

main.cpp

mywidget.cpp

界面文件

mywidget.ui

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

```
#ifndef MYWIDGET_H
#define MYWIDGET_H

#include <QWidget>

namespace Ui {
class MyWidget;
}

class MyWidget : public QWidget
{
    Q_OBJECT

public:
    explicit MyWidget(QWidget *parent = 0);
    ~MyWidget();

private:
    Ui::MyWidget *ui;
};

#endif // MYWIDGET_H
```

此命名空间的作用：提前声明告诉编译器，此命名空间在别处定义。

ui对象指针，所对应类为6行Ui命名空间中MyWidget

mywidget.h 对应的 mywidget.cpp 实现过程如下：



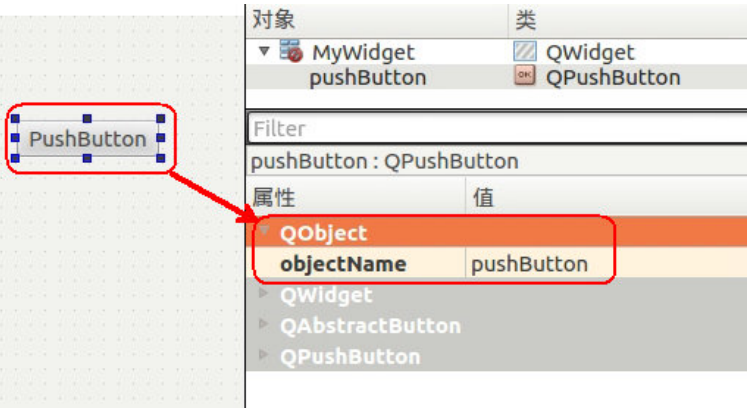
接下来，我们一起看看 ui\_mywidget.h 的实现过程：



通过这样的一步分析，我们发现 Designer（界面设计器）设计 ui 界面最终转换为 C++ 代码（ui\_类名.h），和我们写的 Qt 代码是大同小异的，也就是说代码才是王道，Designer（界面设计器）只是辅助我们快速设计界面，没有它，我们同样可以写 Qt 程序。假如我们对某些部件操作不熟悉，不知道该如何用其相应函数，这时候我们通过 Designer（界面设计器）拖拽此部件，修改其所需属性，接着编译，看其自动转换的 C++ 代码如何实现，这样既可学习其相应函数的用法，所以说，Designer（界面设计器）本身就是学习 Qt 的好老师。

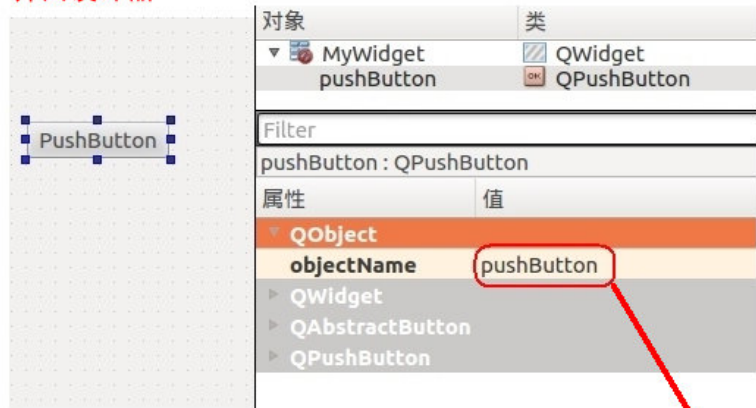
那我们如何通过代码修改 ui 文件部件的属性呢？

在设计模式下，每个部件在“QObject”中都有一个“objectName”（对象名字）的属性：



“objectName”（对象名字）的属性的值（如此例子中的值“pushButton”），此值则为“ui\_类名.h”中自动创建的对象指针名：

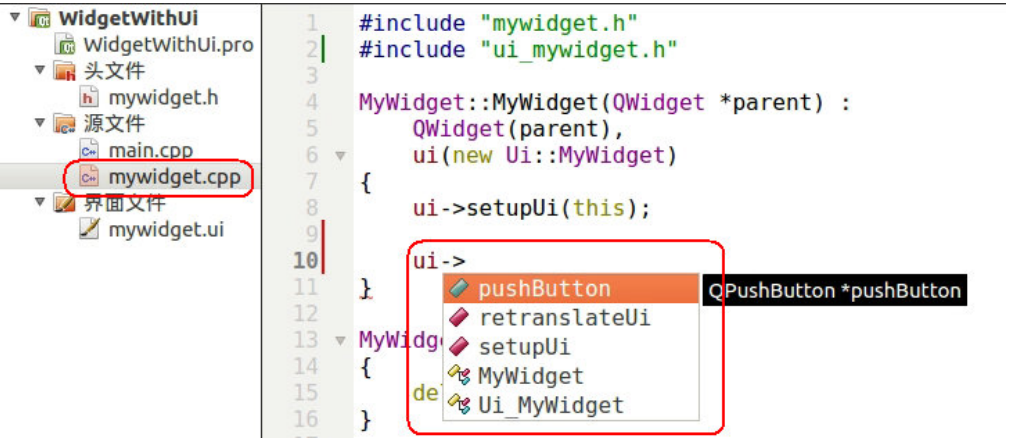
界面设计器



ui\_mywidget.h

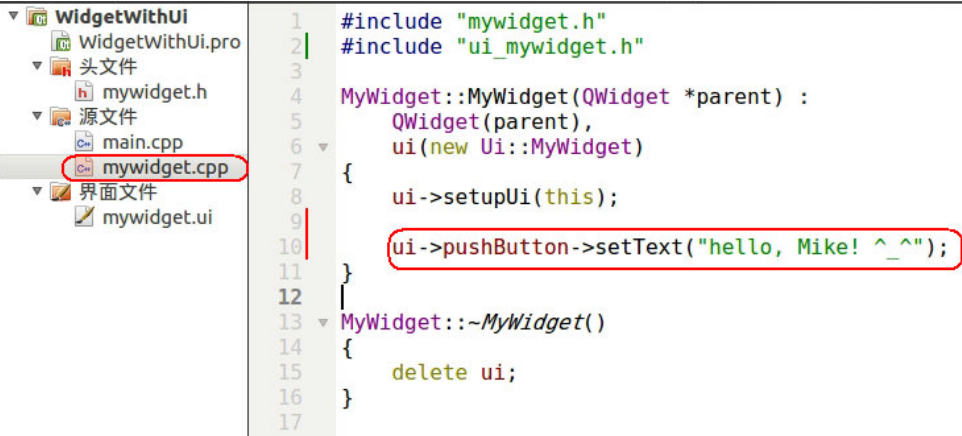
```
22 class Ui_MyWidget
23 {
24 public:
25     QPushButton *pushButton;
26
27 void setupUi(QWidget *MyWidget)
28 {
```

所以，我们在 .cpp 文件某函数中，可以通过“ui->”引用此成员，在 Qt Creator 中，“.”（点）键比较智能，如果操作的是普通对象，则为“.”（点）”，如果是对象指向，自动变为“->”，而且提示可以引用的成员：



这里，通过代码修改 pushButton 的内容：





编译运行程序后，按钮的内容确实被修改：

