

# VS+QT环境配置

## 1、安装vs 2017和qt，

下载下面这个，完成在VS中安装qt插件。

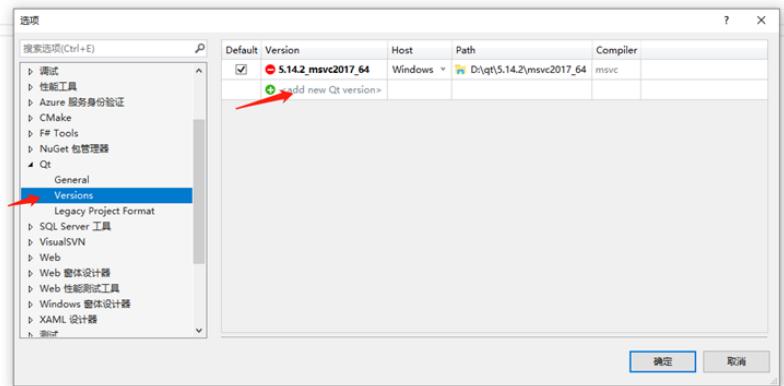
Qt Visual Studio Tools (for vs 2017) 链接：

<https://pan.baidu.com/s/1w15AGfPG5JUWd6jnNd80JQ>, 提取码：8x4z

## 2、在VS中进行qt的配置，如下图



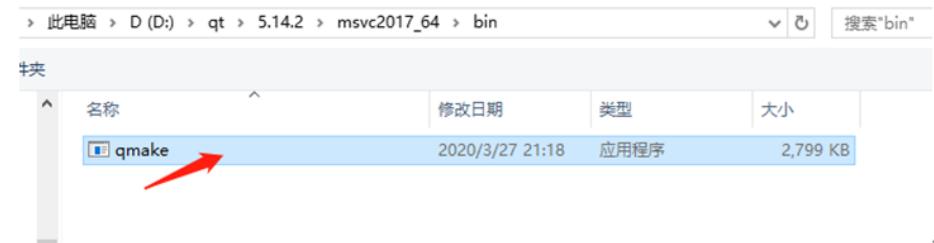
点击后选择 options 或 qt versions,



选择 add new,进行添加。在选择 path 选择添加文件的路径。

Default	Version	Host	Path	Compiler
<input checked="" type="checkbox"/>	5.14.2_msvc2017_64	Windows	D:\qt\5.14.2\msvc2017_64	msvc
<input type="checkbox"/>	5.14.2_msvc2017_64	Windows		msvc
<input type="checkbox"/>	<add new Qt version>			

找到安装 qt 的文件的位置，找到下面这个程序。

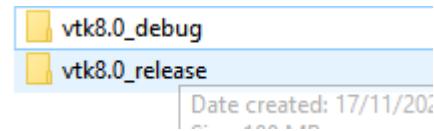


添加成功后，可完成 qt 在 VS 中的配置。

# VTK的库

Debug

release



每个库的结构如下

Bin=dll文件，（此文中该文件夹也用作vs的工作目录）

Include=头文件

Lib=lib文件

---

- bin
- cmake
- include
- lib
- plugins
- share

在lib文件夹下批处理文件输入代码：“DIR \*.\* /B >list.TXT”，并运行，得到lib的名称。后面用。

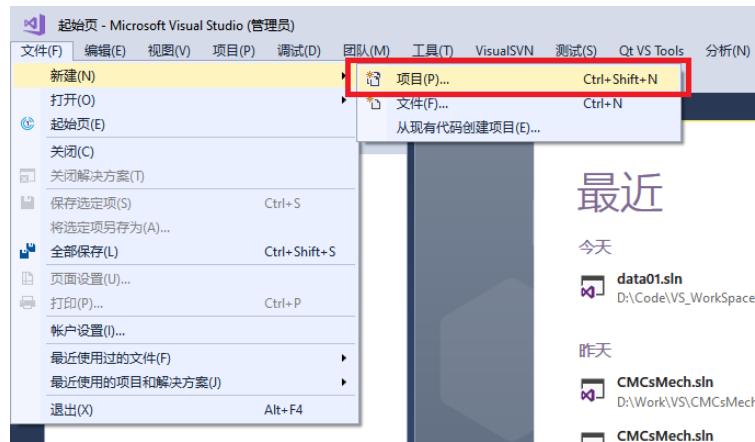
■ cmake	15/07/2021 20:49	File folder	
■ a.bat	18/07/2021 20:02	Windows Batch File	1 KB
■ list.TXT	18/07/2021 20:03	TXT File	3 KB
■■■ vtkalglib-8.0.lib	15/07/2021 20:49	Object File Library	47 KB
■■■ vtkChartsCore-8.0.lib	15/07/2021 20:49	Object File Library	580 KB

# 步骤1

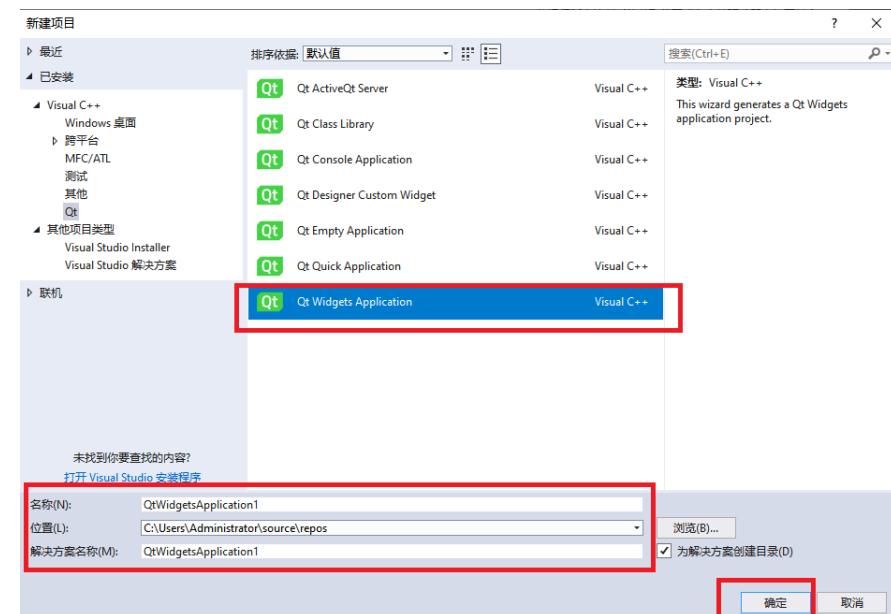
## 1.1 双击打开VS



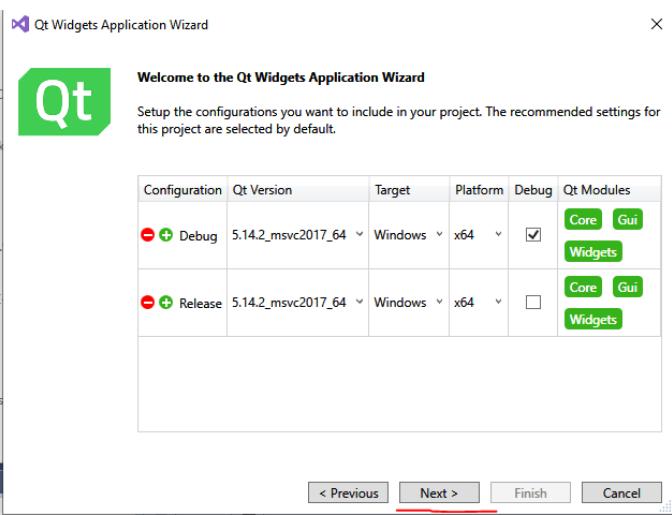
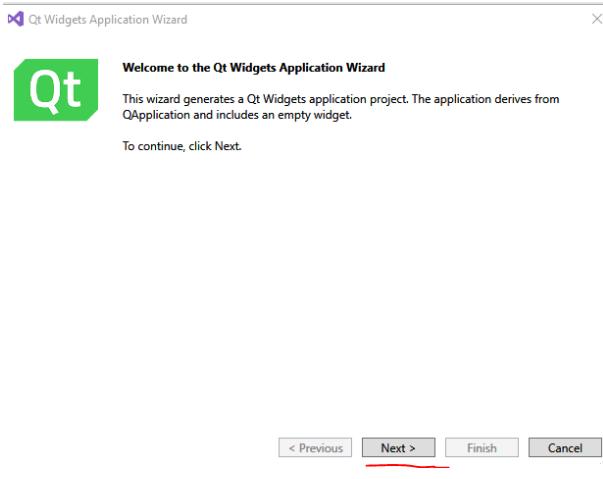
## 1.2 出现空白的软件界面，新建项目



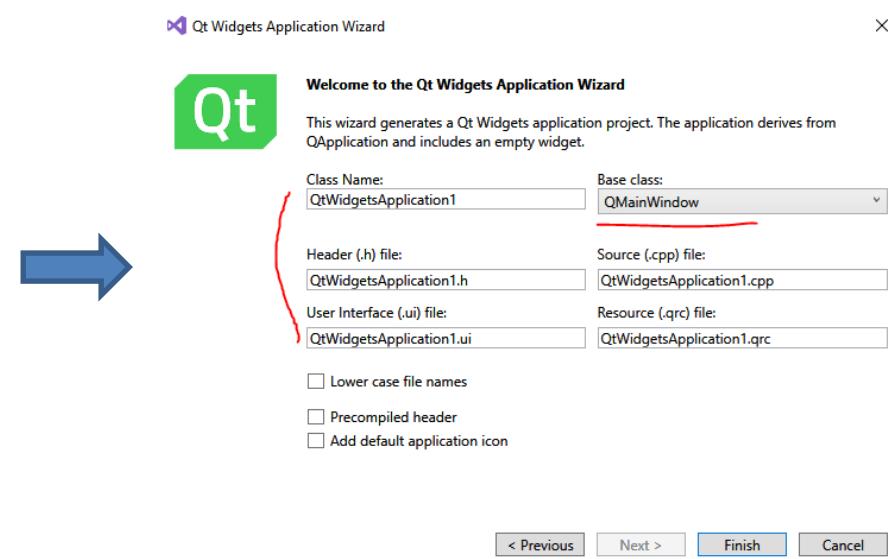
## 1.3 弹出窗口选择QT，并修改名称，路径



# 步骤1.3



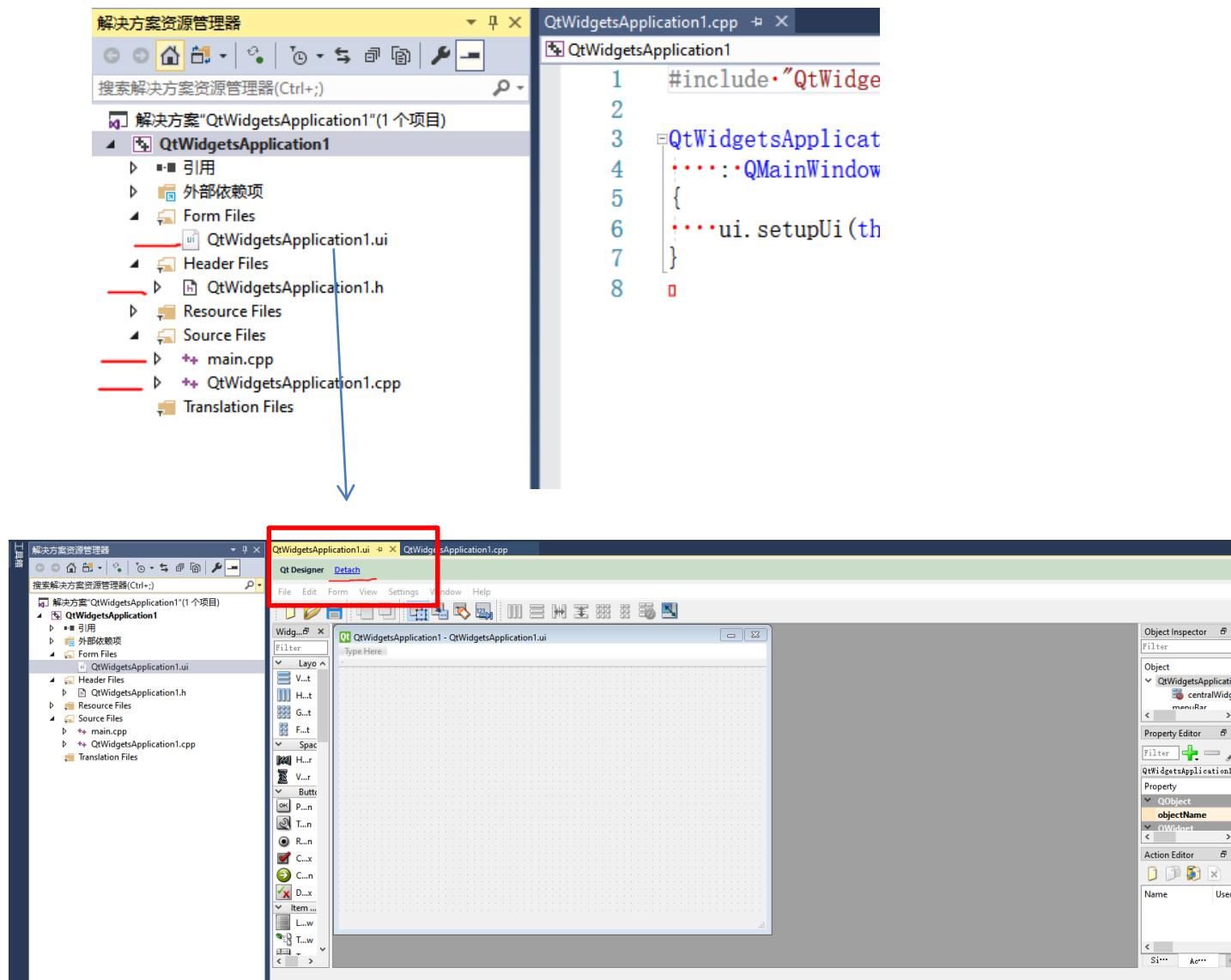
单击Next->next  
有需要就修改类名称  
单击finish



## 步骤1.4

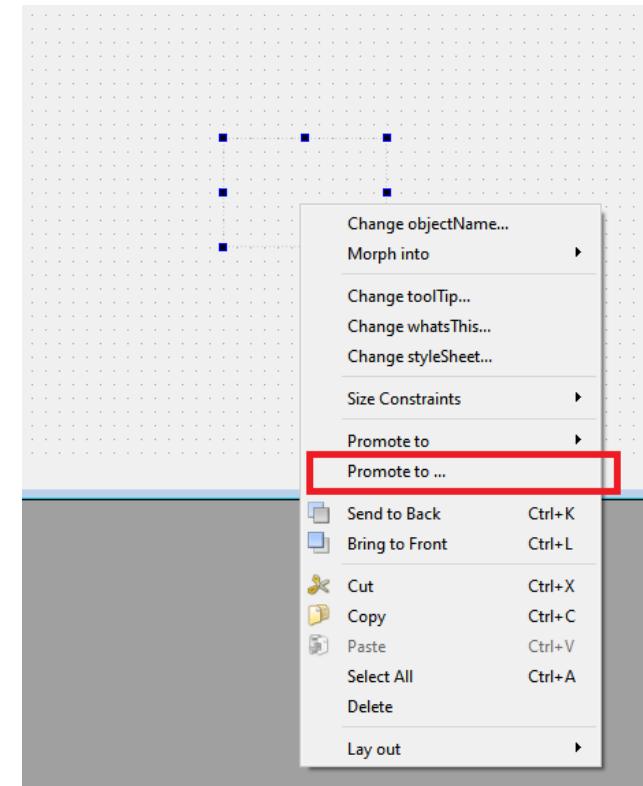
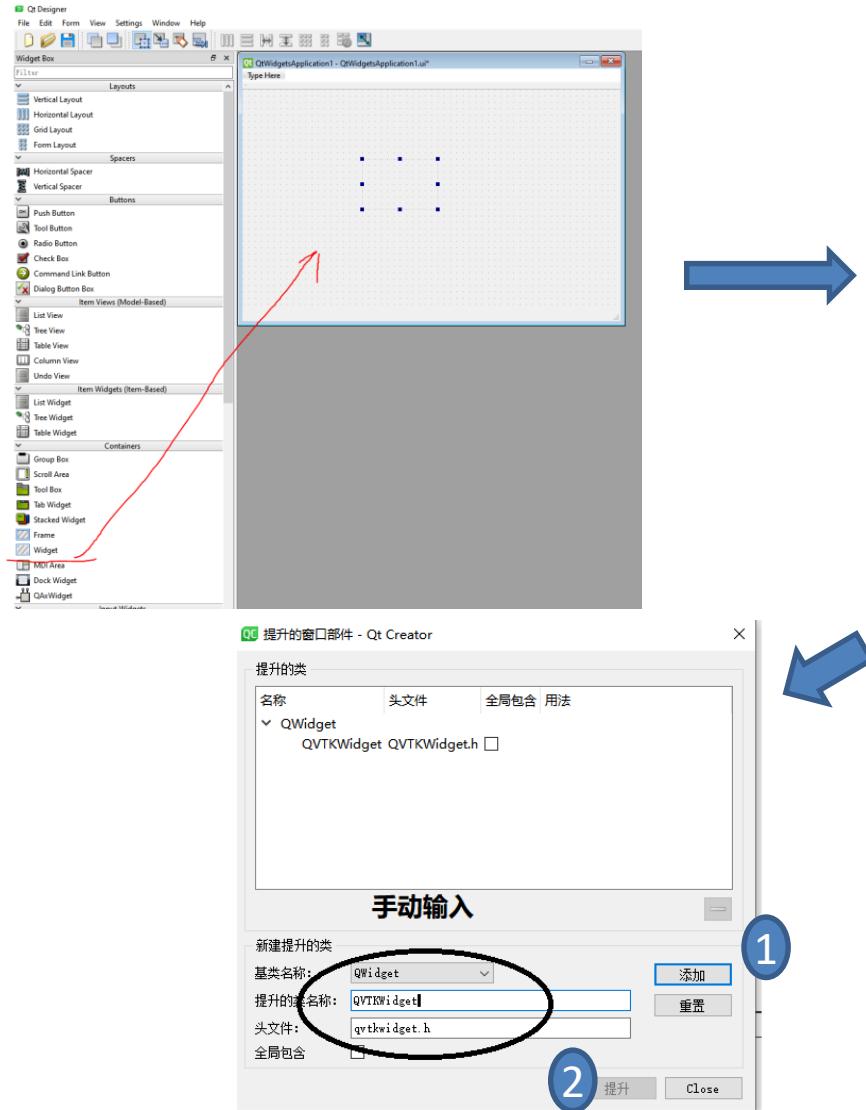
出现如下文件， ui=qt界面（双击进入）， h, cpp文件（单击进入）

双击.ui文件，进入GUI设计界面Qt Designer。单击“detach”，可让窗口弹出，视野更好



## 步骤1.5

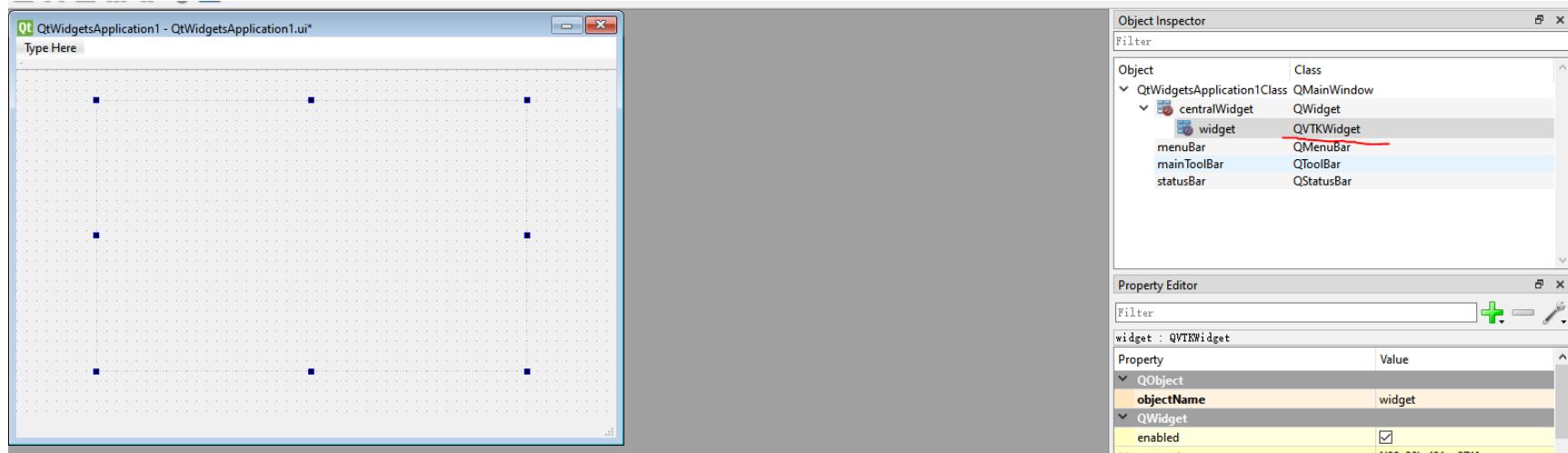
拖动一个“widget”到窗口中；  
在widget上右击，单击“提升为”；  
手动输入QVTKWidget（大小写一致）



输入完毕，先“添加”，后“提升”

## 步骤1.6

发现原来的widget已经变成了QVTKWidget

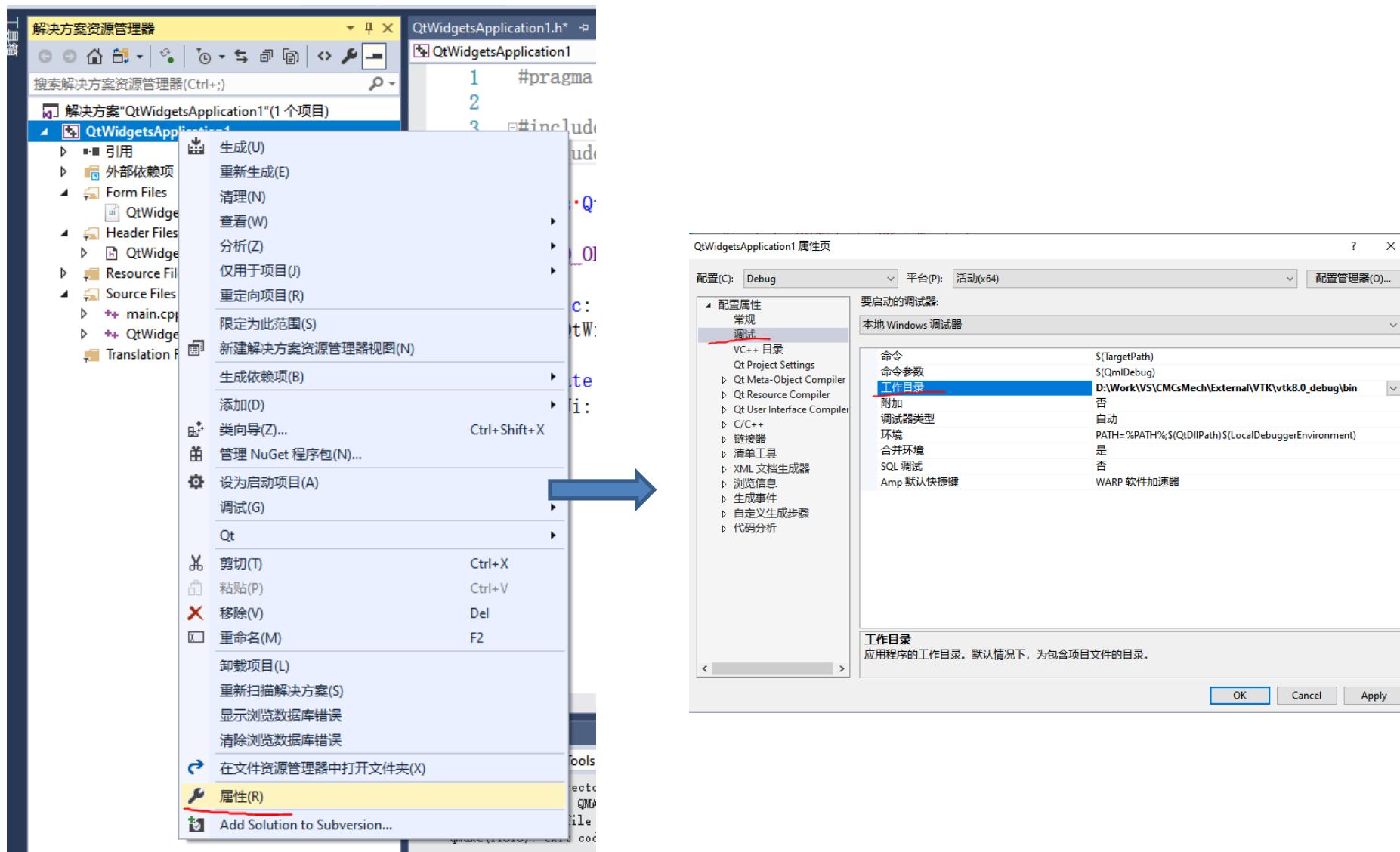


单击保存，数据会传回VS



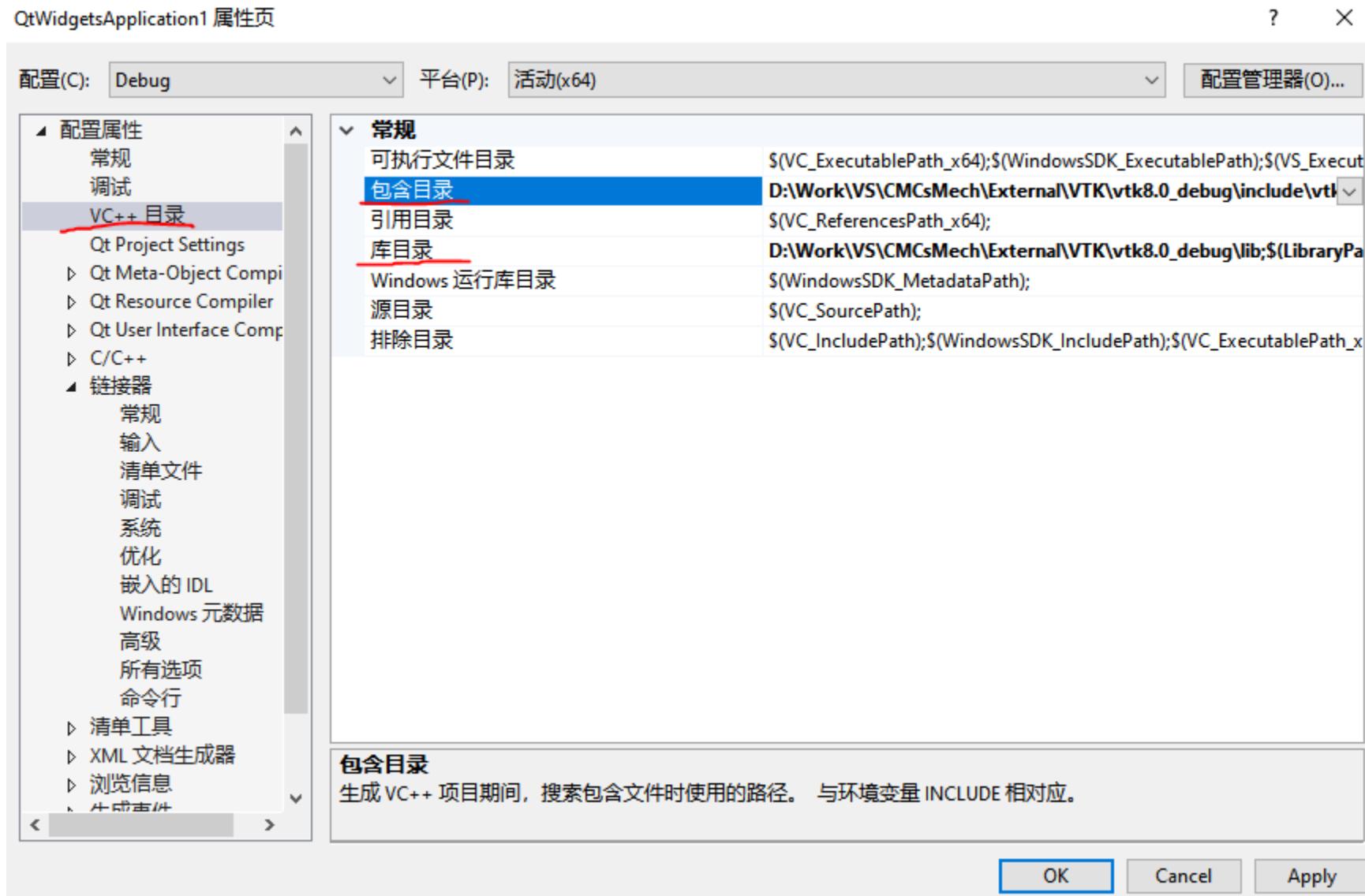
## 步骤1.7

返回vs，在项目上右击->属性。调试->改工作目录为vtk的“bin”



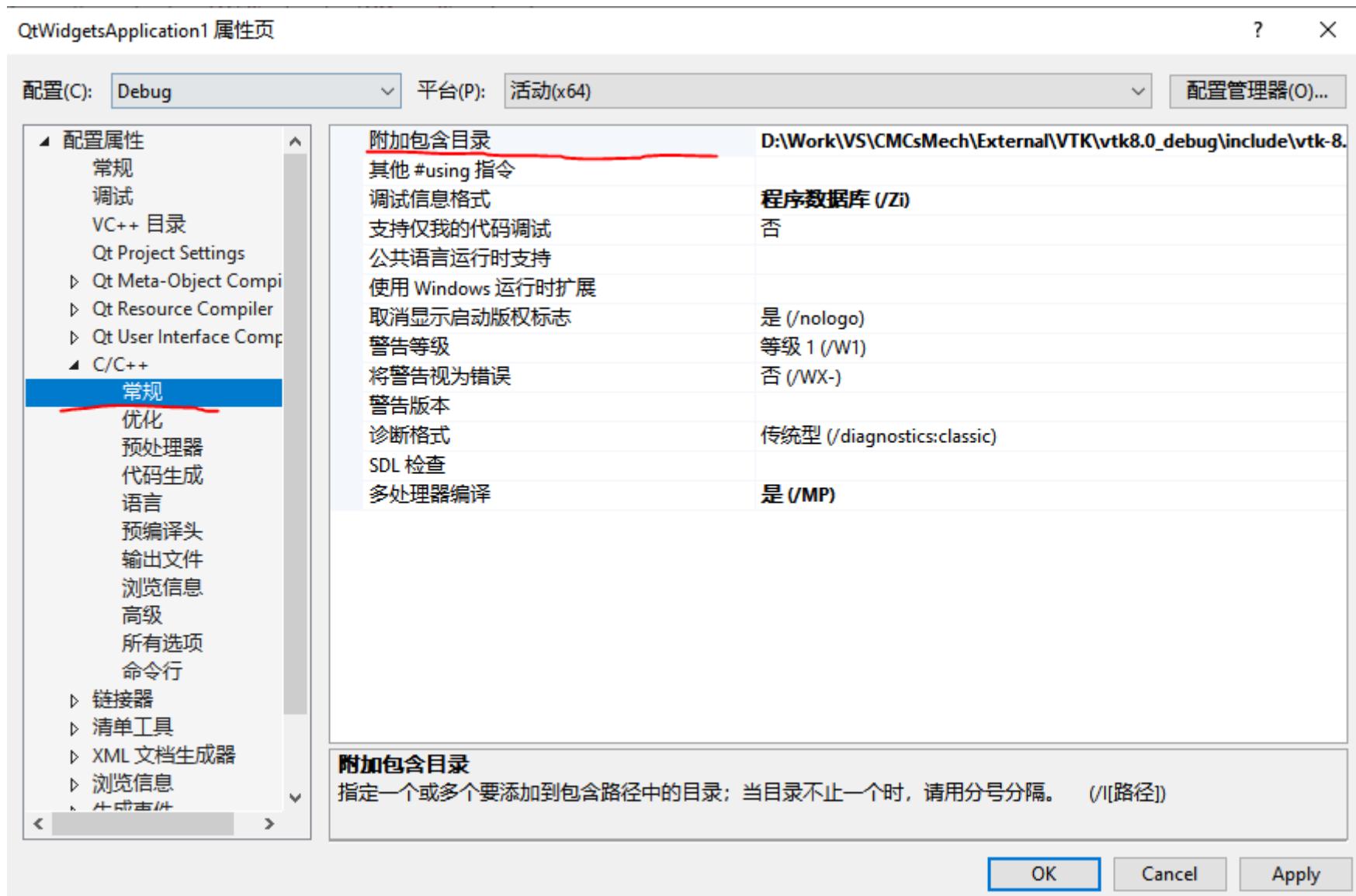
## 步骤1.8

VC++目录-》包含目录=vtk的“include”下面出现h文件的路径”；  
库目录= vtk的“lib”文件夹。



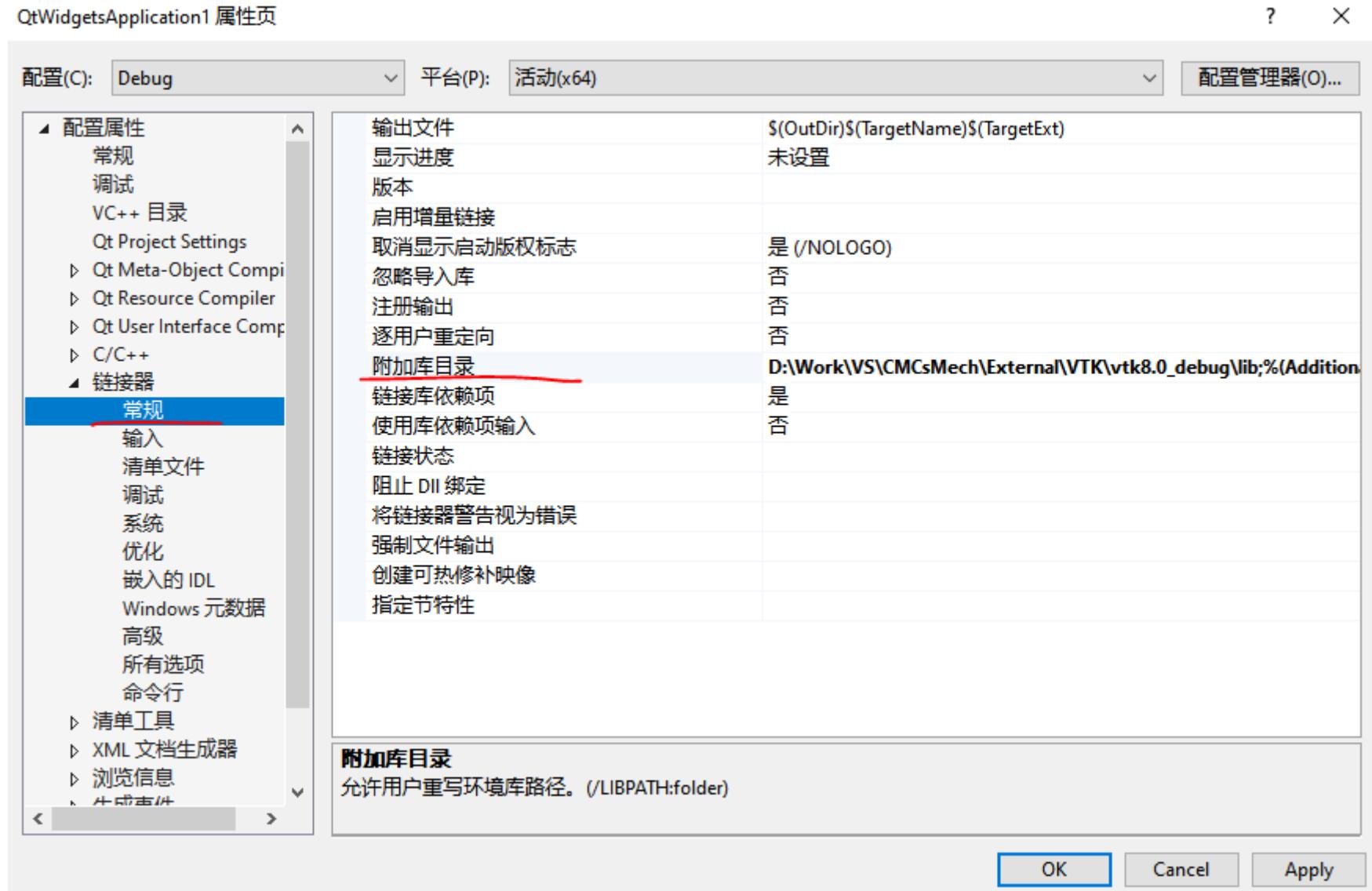
## 步骤1.9

C/C++->常规-> 附加包含目录=vtk的“include下面出现h文件的路径”；



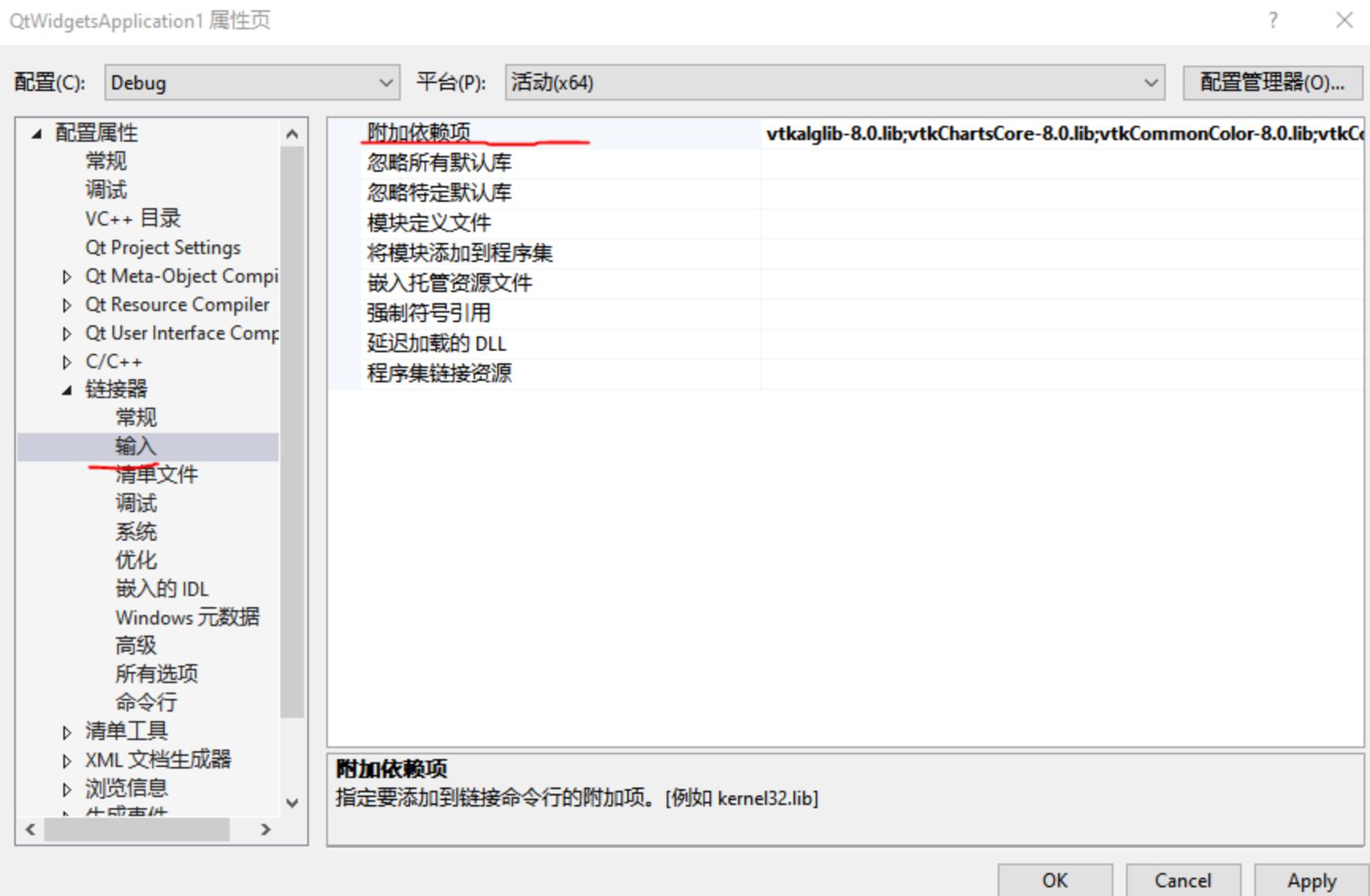
## 步骤1.20

链接器-》常规-》附加库目录=vtk的“lib”文件夹。



## 步骤1.21

链接器-》输入-》附加依赖项=vtk的lib文件夹下所有lib文件名称。第2页ppt得到的。



上述操作冗余了。

## 步骤1.22

头文件增加如下代码。

```
main.cpp QtWidgetsApplication1.h QtWidgetsApplication1.cpp
QtWidgetsApplication1 QtWidgetsApplication1

1 #pragma once
2
3 #include <QtWidgets/QMainWindow>
4 #include "ui_QtWidgetsApplication1.h"
5
6 #include <vtkAutoInit.h>
7 #ifndef INITIAL_OPENGL
8 #define INITIAL_OPENGL
9 VTK_MODULE_INIT(vtkRenderingOpenGL);
10 VTK_MODULE_INIT(vtkInteractionStyle);
11 VTK_MODULE_INIT(vtkRenderingFreeType);
12#endif
13
14 #include <vtkSphereSource.h>
15 #include <vtkPolyData.h>
16 #include <vtkSmartPointer.h>
17 #include <vtkPolyDataMapper.h>
18 #include <vtkActor.h>
19 #include <vtkRenderWindow.h>
20 #include <vtkRenderer.h>
21 #include <vtkRenderWindowInteractor.h>
22
23
24 class QtWidgetsApplication1 : public QMainWindow
25 {
```

```
#include <vtkAutoInit.h>
#ifndef INITIAL_OPENGL
#define INITIAL_OPENGL
VTK_MODULE_INIT(vtkRenderingOpenGL);
VTK_MODULE_INIT(vtkInteractionStyle);
VTK_MODULE_INIT(vtkRenderingFreeType);
#endif

#include <vtkSphereSource.h>
#include <vtkPolyData.h>
#include <vtkSmartPointer.h>
#include <vtkPolyDataMapper.h>
#include <vtkActor.h>
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkRenderWindowInteractor.h>
```

## 步骤1.23

Cpp文件构造函数增加如下代码。  
代码中Ui.widget根据qt designer中的object名称适应性修改。

```
main.cpp QtWidgetsApplication1.h QtWidgetsApplication1.cpp # x
QtWidgetsApplication1
1 #include "QtWidgetsApplication1.h"
2
3 QtWidgetsApplication1::QtWidgetsApplication1(QWidget *parent)
4     : QMainWindow(parent)
5 {
6     ui.setupUi(this);
7
8     vtkSmartPointer<vtkSphereSource> sphereSource = 
9     vtkSmartPointer<vtkSphereSource>::New();
10    sphereSource->SetCenter(0.0, 0.0, 0.0);
11    sphereSource->SetRadius(5.0);
12
13     //mapper
14     vtkSmartPointer<vtkPolyDataMapper> mapper = 
15     vtkSmartPointer<vtkPolyDataMapper>::New();
16     mapper->SetInputConnection(sphereSource->GetOutputPort());
17
18     //actor
19 }
```

```
vtkSmartPointer<vtkSphereSource> sphereSource =
vtkSmartPointer<vtkSphereSource>::New();
sphereSource->SetCenter(0.0, 0.0, 0.0);
sphereSource->SetRadius(5.0);

//mapper
vtkSmartPointer<vtkPolyDataMapper> mapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
mapper->SetInputConnection(sphereSource->GetOutputPort());

//actor
vtkSmartPointer<vtkActor> actor =
vtkSmartPointer<vtkActor>::New();
actor->SetMapper(mapper);

//renderer ,renderWindow, renderWindowInteractor.
vtkSmartPointer<vtkRenderer> renderer =
vtkSmartPointer<vtkRenderer>::New();
vtkSmartPointer<vtkRenderWindow> renderWindow =
vtkSmartPointer<vtkRenderWindow>::New();

ui.widget->GetRenderWindow()->AddRenderer(renderer); //加入renderer显示

vtkSmartPointer<vtkRenderWindowInteractor> renderWindowInteractor =
vtkSmartPointer<vtkRenderWindowInteractor>::New();
renderWindowInteractor = ui.widget->GetInteractor();

renderWindow = ui.widget->GetRenderWindow();

renderer->AddActor(actor);
renderer->SetBackground(1, 1, 1); // Background color green

ui.widget->update();
```

步骤1.24  
运行。

