

# 引言

该文件描述的是在 Visual Studio 2012 中，如何创建一个 C++ 解决方案，该解决包含一个 Common 项目，以 DLL 的方式供主程序 ProjectTemplate 调用，Common 项目主要提供一些公共类和公共函数。

另，文档中的截图，有一定的 *PS* 操作，绝对位置会有一些的修改，请认准名称。

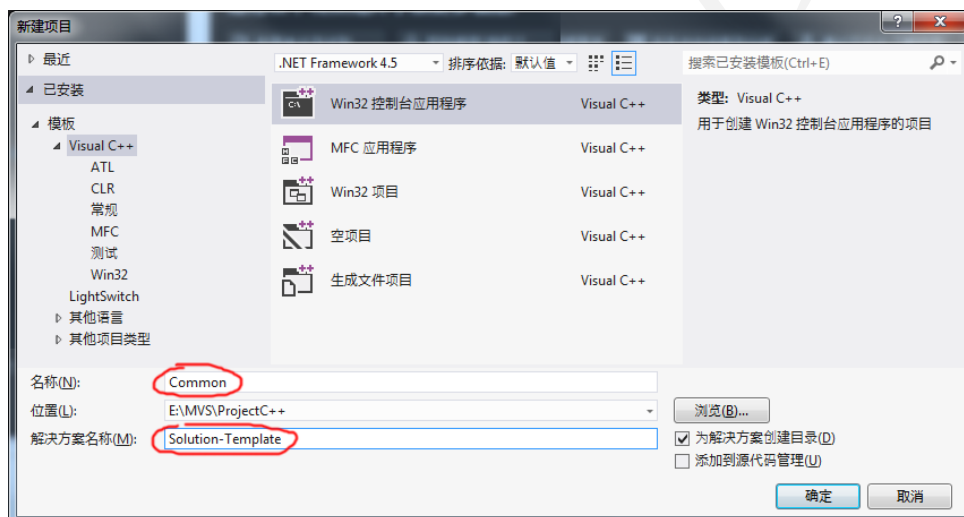
# 目 录

引言 .....	i
第1章 创建工程 .....	1
1.1 Common .....	1
1.2 Project Template .....	2
1.3 设置解决方案属性 .....	3
1.4 配置 32 位程序和 64 位程序 .....	4
1.5 第三方依赖库 .....	4
第2章 Common 项目 .....	6
2.1 Common 属性设置 .....	6
2.1.1 相同设置项 .....	6
2.1.2 win32 .....	8
2.1.3 x64 .....	9
2.2 添加代码 .....	10
第3章 ProjectTemplate .....	11
3.1 Common 属性设置 .....	11
3.2 添加代码 .....	12

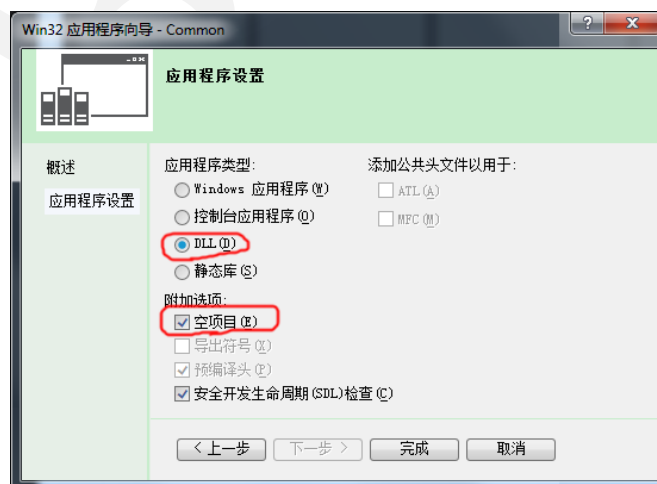
# 第1章 创建工程

## 1.1 Common

点击新建解决方案，新建项目，在名称中输入“Common”，在解决方案名称中输入“Solution-Template”。点击确定进入下一步。



附加选项中选择空项目。由于该项目是以 DLL 形式供其它项目调用，在应用程序类型中选择DLL。

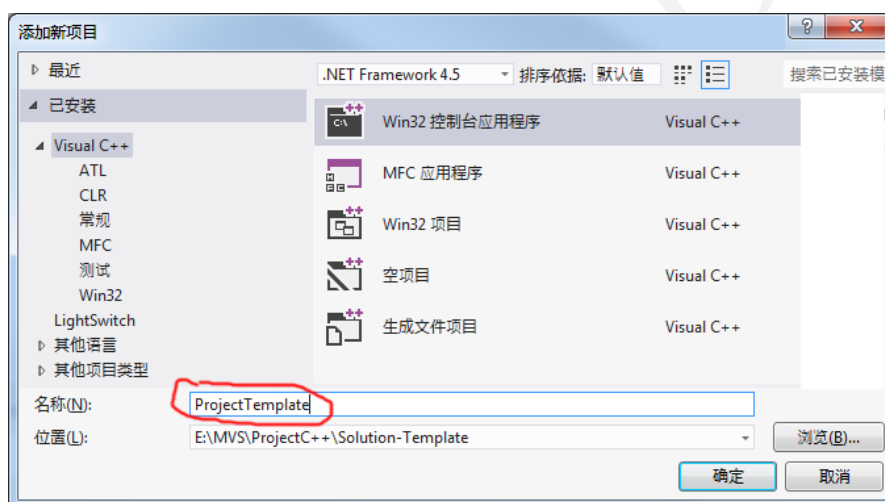


## 1.2 Project Template

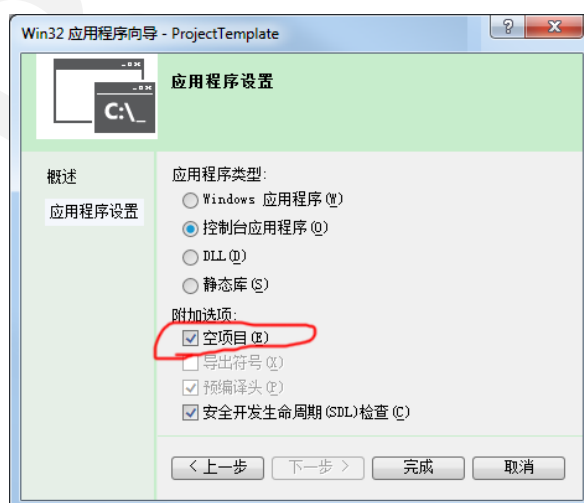
右击解决方案，在**添加**中选择**新建项目**。



在名称中输入“ProjectTemplate”。点击确定进入下一步。

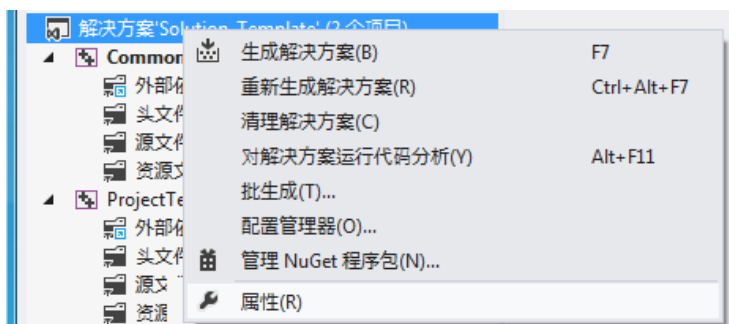


附加选项中选择**空项目**。

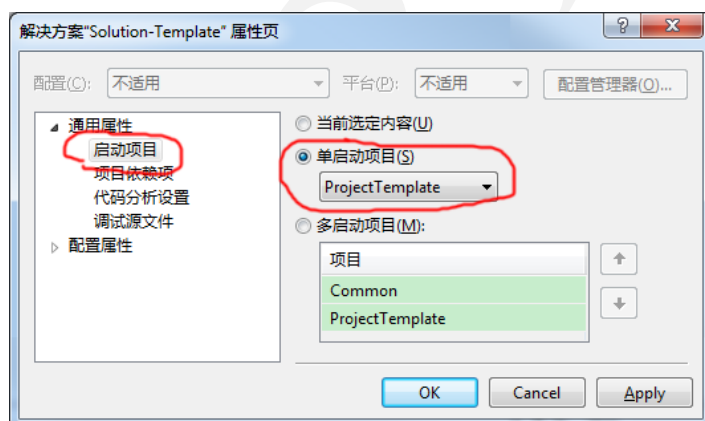


## 1.3 设置解决方案属性

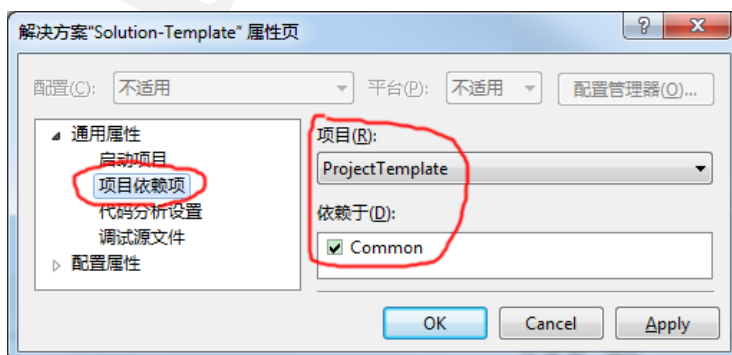
右击解决方案，选择属性。



在通用属性中选择启动，单启动项目选择“ProjectTemplate”，表明该软件只启动“ProjectTemplate”。



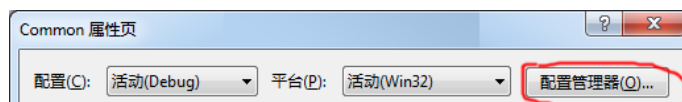
在通用属性中选择项目依赖项，项目选择“ProjectTemplate”，依赖于中，选中“Common”，表明“ProjectTemplate”依赖于“Common”。



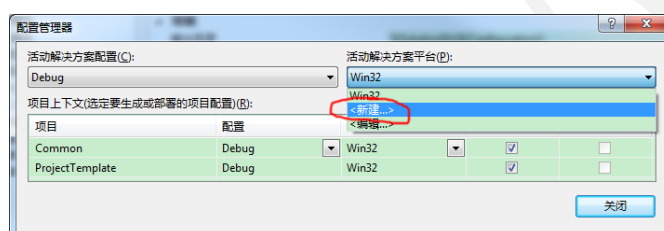
## 1.4 配置 32 位程序和 64 位程序

右击**Common**, 选择属性。

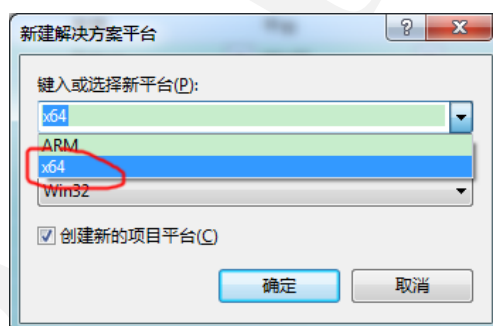
选择**配置管理器**, 配置 64 位程序。



在**活动解决方案平台**的下拉框中选择**新建**。



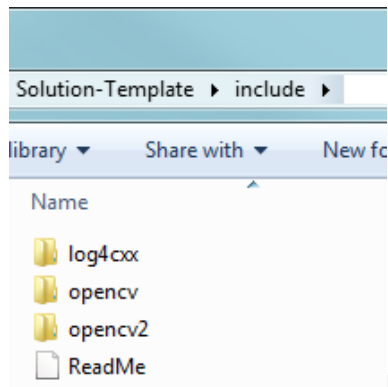
在新建解决方案平台中选择 x64。



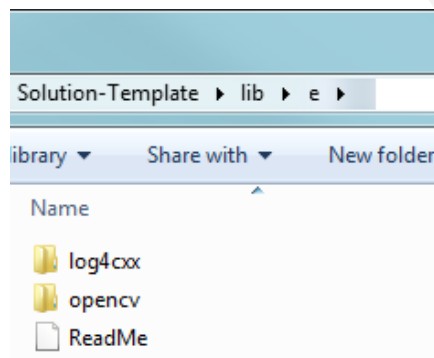
## 1.5 第三方依赖库

将 include 文件存放到解决方案目录下的 include 文件夹。

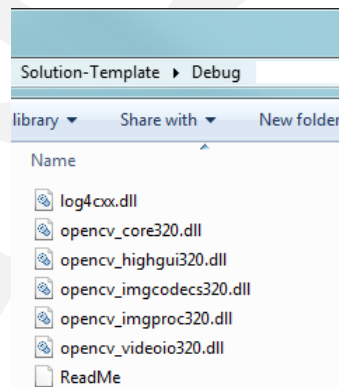
## 1.5 第三方依赖库



将 lib 文件存放到解决方案目录下的 lib 文件夹。



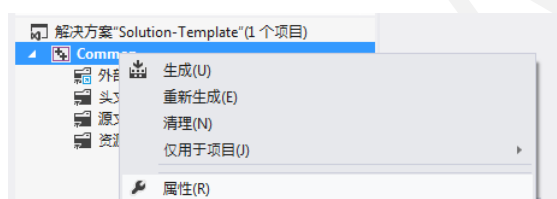
将 dll 文件存放到 Debug、Release、x64/Debug 和 x64/Release 文件夹下。



## 第2章 Common 项目

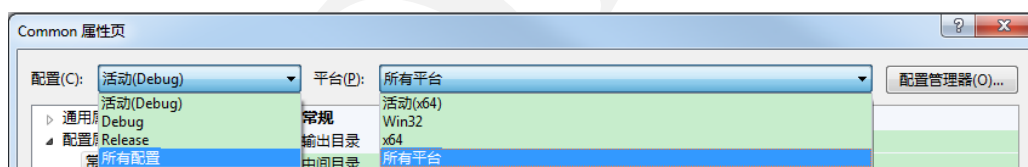
### 2.1 Common 属性设置

右击Common, 选择属性。

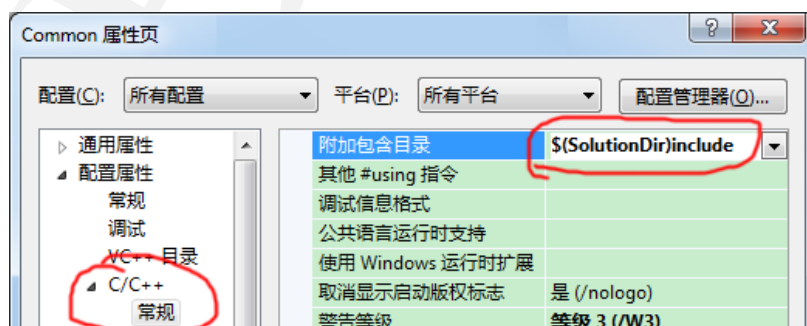


#### 2.1.1 相同设置项

在配置中选择所有配置，在平台中选择所有平台。



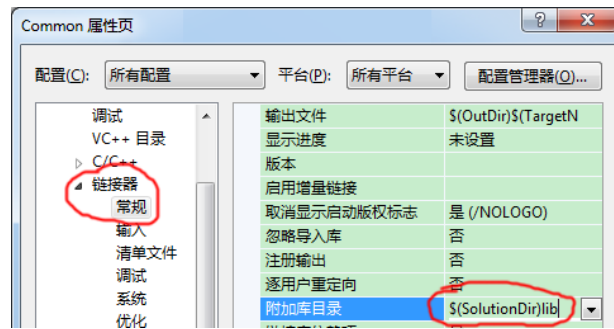
选中配置属性->C/C++->常规，在右侧的附加包含目录中输入：\$(SolutionDir)include。



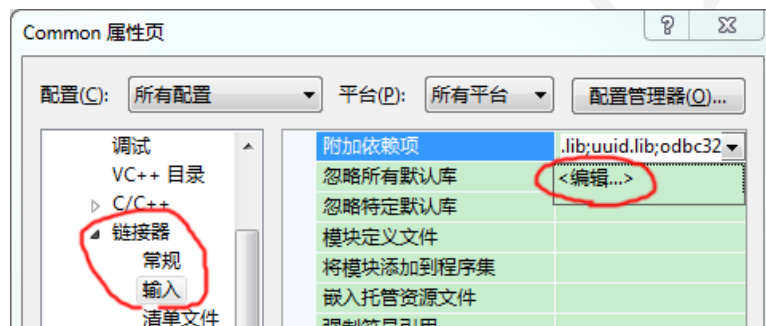
选中配置属性->链接器->常规，在右侧的附加库目录中输入：\$(SolutionDir)lib。



## 2.1 Common 属性设置

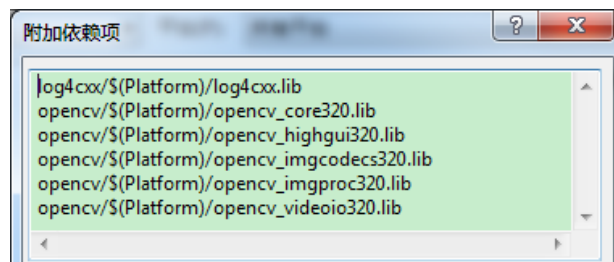


选中配置属性->链接器->输入，右侧附加依赖项的下拉菜单中选择编辑。



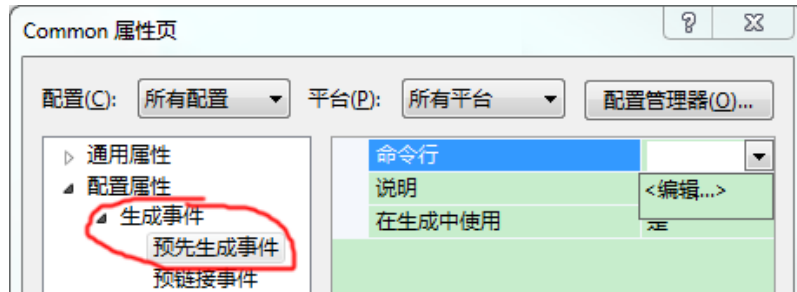
在弹出的窗口中输入 log4cxx 和 opencv（本例中未用到）的依赖项：

- log4cxx/\$(Platform)/log4cxx.lib
- opencv/\$(Platform)/opencv\_core320.lib
- opencv/\$(Platform)/opencv\_highgui320.lib
- opencv/\$(Platform)/opencv\_imgcodecs320.lib
- opencv/\$(Platform)/opencv\_imgproc320.lib
- opencv/\$(Platform)/opencv\_videoio320.lib

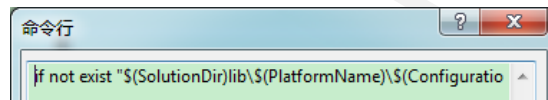


## 2.1 Common 属性设置

选中配置属性->生成事件->预先生成事件，右侧命令行的下拉菜单中选择编辑。

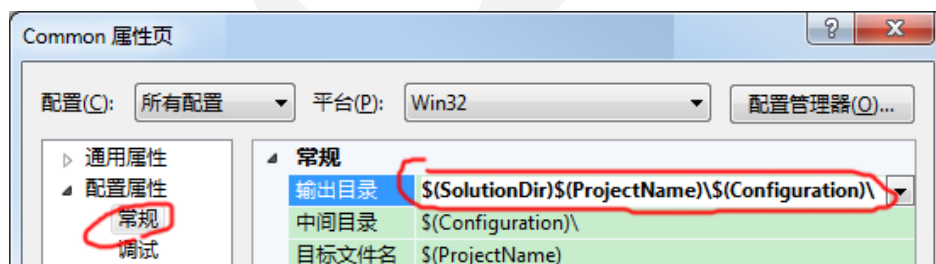


```
1 if not exist "$(SolutionDir)lib\$(PlatformName)\$(Configuration)\"  
   mkdir "$(SolutionDir)lib\$(PlatformName)\$(Configuration)\  
   ")
```



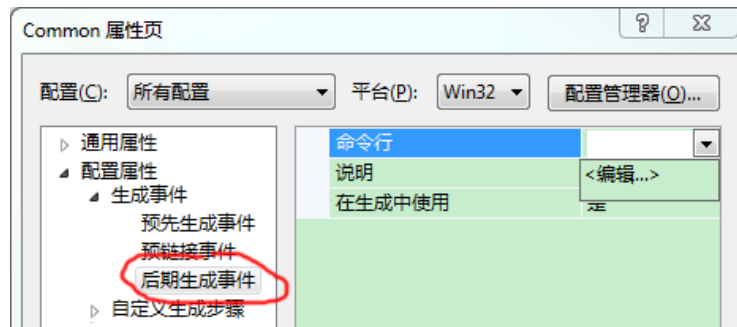
### 2.1.2 win32

选中配置属性->常规，在右侧的输出目录中输入：  
\$(SolutionDir)\\$(ProjectName)\\$(Configuration)\。

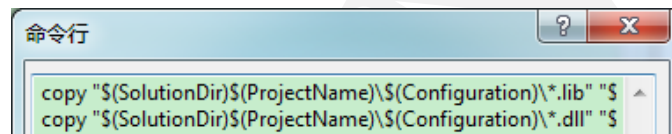


选中配置属性->生成事件->后期生成事件，右侧命令行的下拉菜单中选择编辑。

## 2.1 Common 属性设置

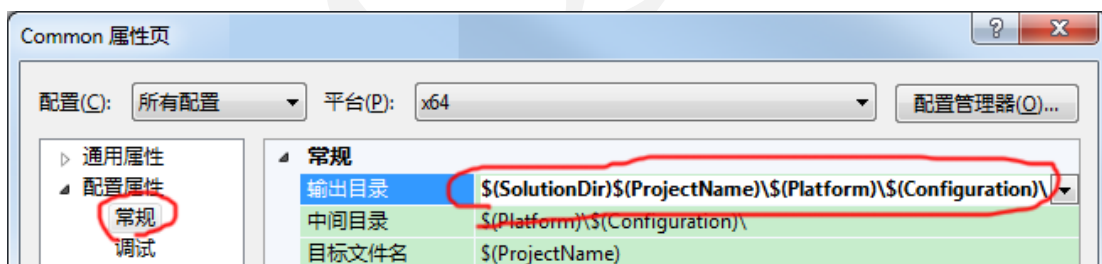


```
1 copy "$(SolutionDir)$(ProjectName)\$(Configuration)\*.lib" "$(SolutionDir)lib\$(Platform)\$(Configuration)"
2 copy "$(SolutionDir)$(ProjectName)\$(Configuration)\*.dll" "$(SolutionDir)$(Configuration)"
```



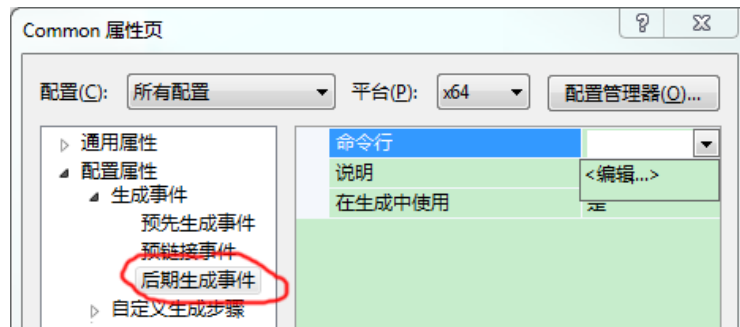
### 2.1.3 x64

选中配置属性->常规，在右侧的输出目录中输入：  
\$(SolutionDir)\\$(ProjectName)\\$(Configuration)\。

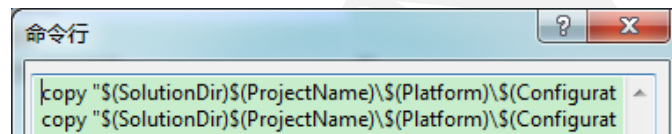


选中配置属性->生成事件->后期生成事件，右侧命令行的下拉菜单中选择编辑。

## 2.2 添加代码



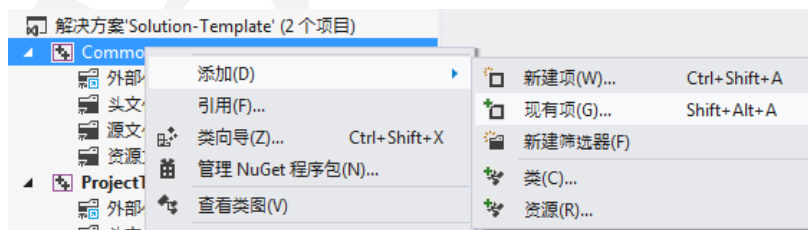
```
1 copy "$(SolutionDir)$(ProjectName)\$(Platform)\$(Configuration) \
  *.lib" "$(SolutionDir)lib\$(Platform)\$(Configuration) "
2 copy "$(SolutionDir)$(ProjectName)\$(Platform)\$(Configuration) \
  *.dll" "$(SolutionDir)$(Platform)\$(Configuration) "
```



## 2.2 添加代码

添加如下类文件：

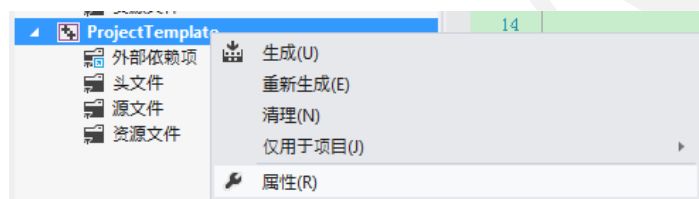
- CommonMethod.cpp
- CommonMethod.h
- Logger.h



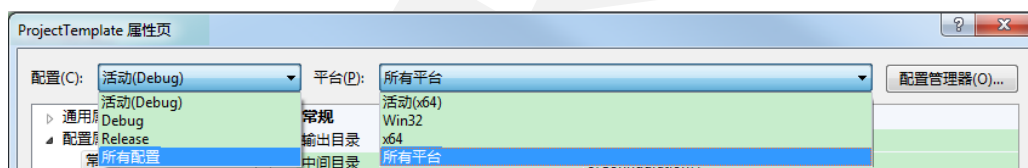
## 第3章 ProjectTemplate

### 3.1 Common 属性设置

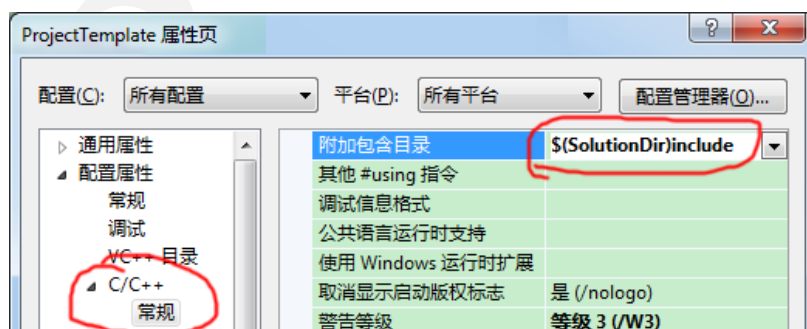
右击ProjectTemplate, 选择属性。



在配置中选择所有配置，在平台中选择所有平台。

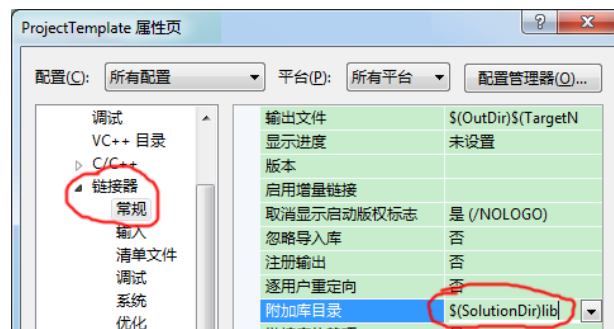


选中配置属性->C/C++->常规，在右侧的附加包含目录中输入：\$(SolutionDir)include;\$(SolutionDir)Common。

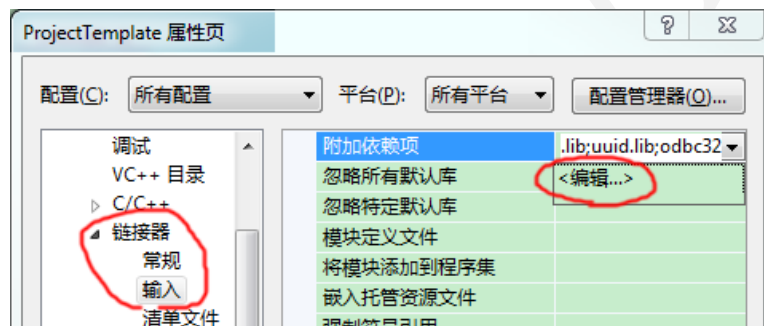


选中配置属性->链接器->常规，在右侧的附加库目录中输入：\$(SolutionDir)lib。

## 3.2 添加代码

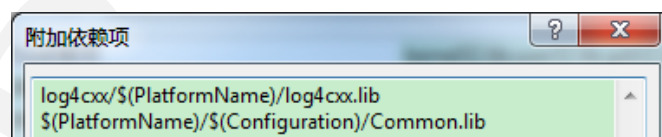


选中配置属性->链接器->输入，右侧附加依赖项的下拉菜单中选择编辑。



在弹出的窗口中输入 log4cxx 和 Common 的依赖项：

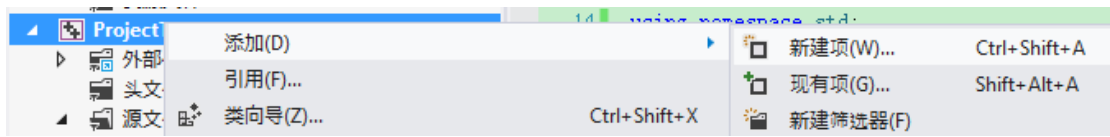
- log4cxx/\$(Platform)/log4cxx.lib
- \$(PlatformName)/\$(Configuration)/Common.lib



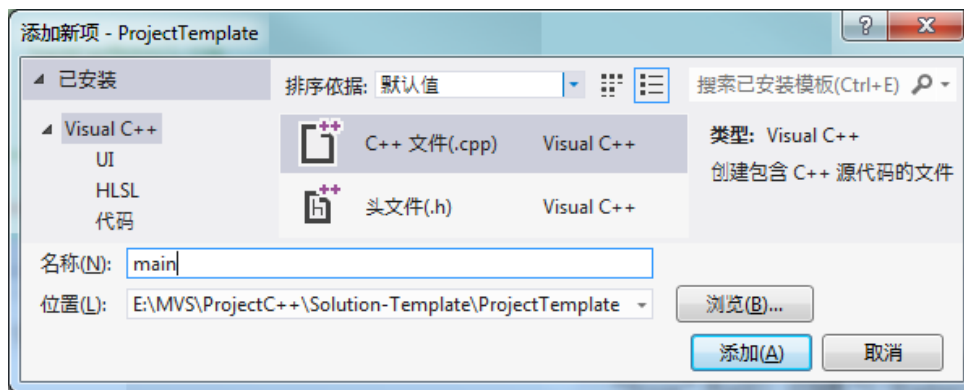
## 3.2 添加代码

右击“ProjectTemplate”，选择添加->新建：

### 3.2 添加代码



新建一个“main.cpp”的文件：



添加代码：

```
1 #include <string>
2
3 #include "CommonMethod.h"
4 #include "Logger.h"
5
6 using namespace std;
7
8 int main(int argc, char**argv)
9 {
10     log4cxx::PropertyConfigurator::configure("logcfg.properties");
11
12     float fVal = 1.5;
13     string strVal = ConvertValToString(fVal, 2);
14
15     printf("%s\n", strVal.c_str());
16     __LOG_DEBUG__(strVal);
17
18     printf("press Enter to exit...\n");
19     getchar();
20 }
```

