

## wpf调用c++动态链接库

笔记本: C#学习笔记

创建时间: 2019/9/23 14:09

更新时间: 2019/9/23 15:46

作者: 1422084186@qq.com

URL: <https://www.cnblogs.com/zhangweizhong/p/8119340.html>

## 注意事项

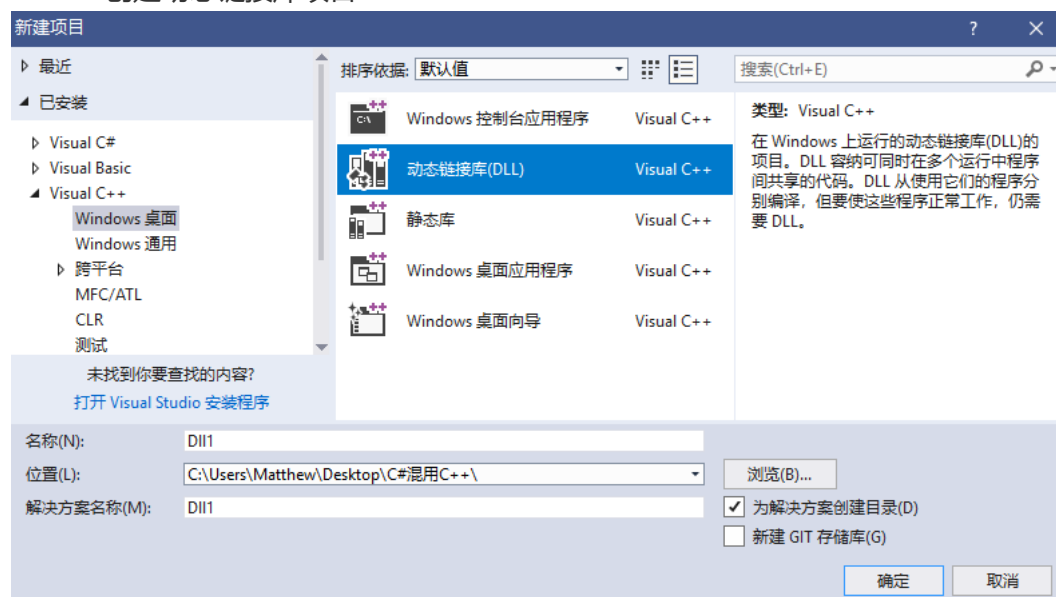
1. c++Dll库编译平台x86还是x64, 需要统一版本

## 动态链接库创建过程

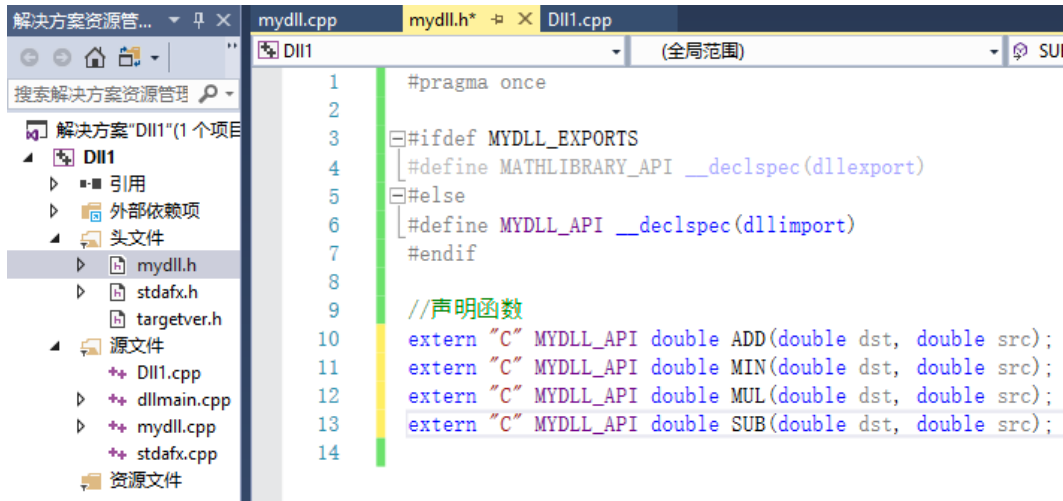
[cnblog章为忠](#)

[微软官方例程](#)

### 1. vs2017创建动态链接库项目



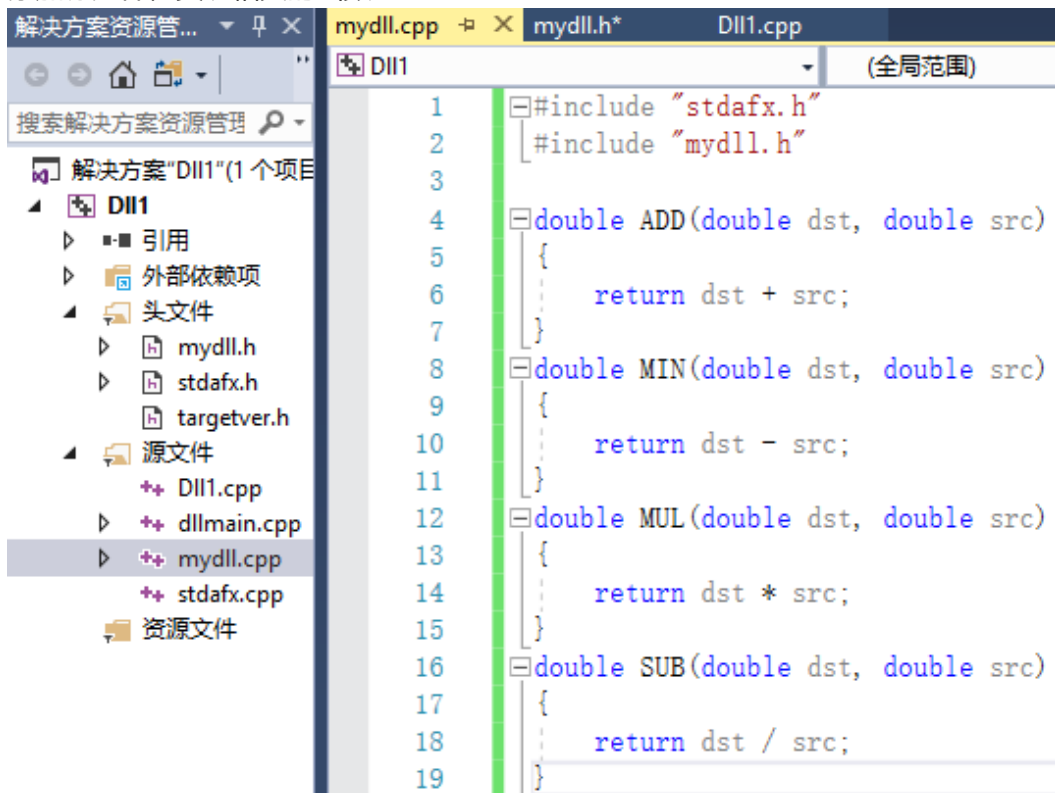
## 2. 添加头文件，声明函数接口



The screenshot shows the Visual Studio IDE with the 'mydll.h' file open. The left sidebar displays the 'Solution Explorer' with the project 'DII1' and its files: 'mydll.h', 'stdafx.h', 'targetver.h', 'DII1.cpp', 'dllmain.cpp', 'mydll.cpp', 'stdafx.cpp', and '资源文件'. The main editor window shows the content of 'mydll.h' with the following code:

```
1  #pragma once
2
3  #ifdef MYDLL_EXPORTS
4      #define MATHLIBRARY_API __declspec(dllexport)
5  #else
6      #define MYDLL_API __declspec(dllimport)
7  #endif
8
9  //声明函数
10 extern "C" MYDLL_API double ADD(double dst, double src);
11 extern "C" MYDLL_API double MIN(double dst, double src);
12 extern "C" MYDLL_API double MUL(double dst, double src);
13 extern "C" MYDLL_API double SUB(double dst, double src);
14
```

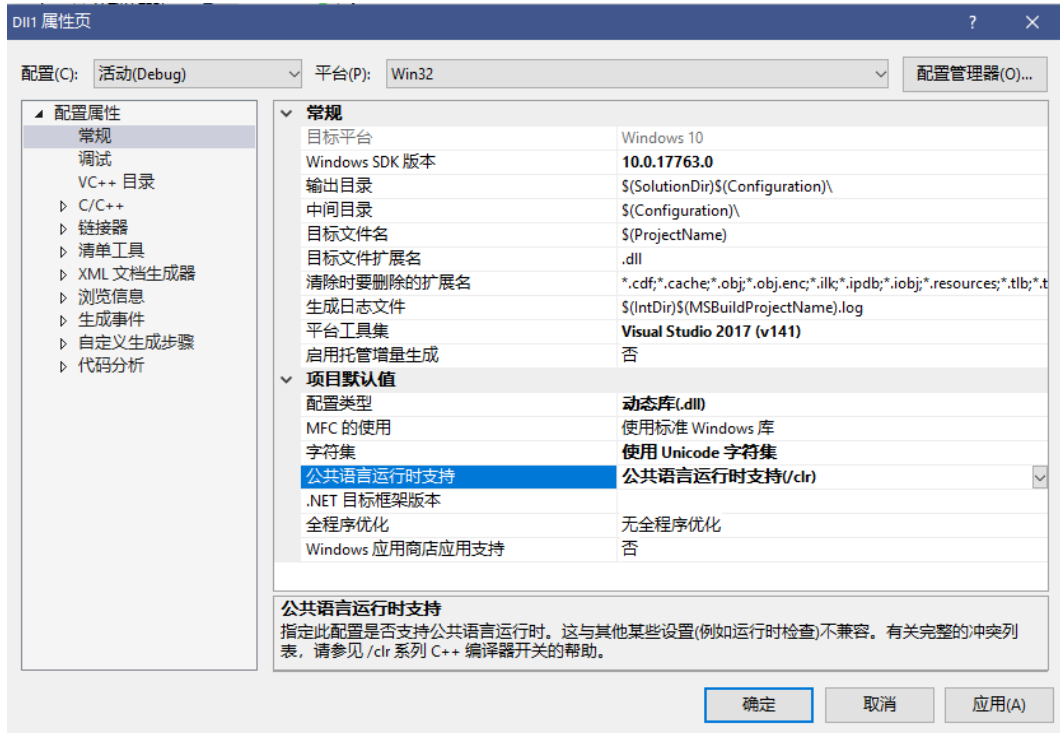
## 3. 添加源文件，实现相关的函数



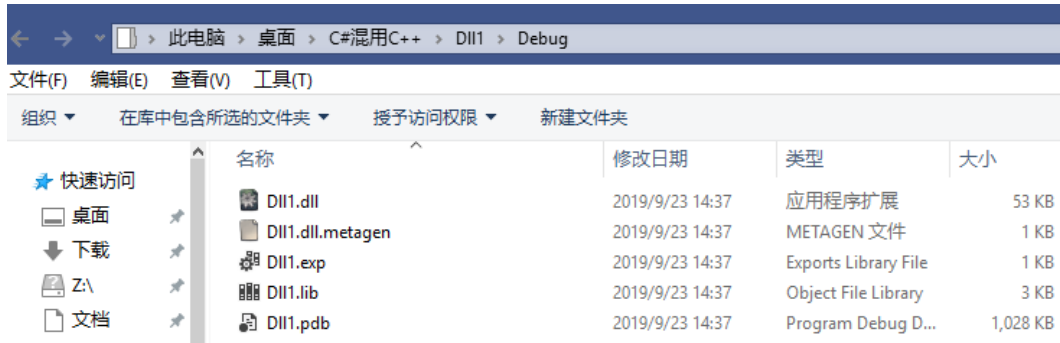
The screenshot shows the Visual Studio IDE with the 'mydll.cpp' file open. The left sidebar displays the 'Solution Explorer' with the project 'DII1' and its files: 'mydll.h', 'stdafx.h', 'targetver.h', 'DII1.cpp', 'dllmain.cpp', 'mydll.cpp', 'stdafx.cpp', and '资源文件'. The main editor window shows the content of 'mydll.cpp' with the following code:

```
1  #include "stdafx.h"
2  #include "mydll.h"
3
4  double ADD(double dst, double src)
5  {
6      return dst + src;
7  }
8  double MIN(double dst, double src)
9  {
10     return dst - src;
11 }
12 double MUL(double dst, double src)
13 {
14     return dst * src;
15 }
16 double SUB(double dst, double src)
17 {
18     return dst / src;
19 }
```

#### 4. 添加公共语言运行时支持



#### 5. 编译生成相关的.dll文件



## wpf对c++Dll的静态调用

### 1. 导入c++动态链接库

## 2. 在C#程序中声明相关DLL函数

```
MainWindow.xaml MainWindow.xaml.cs WpfApp.MainWindow Btn_Click(object sender, Ro
1 using System;
2 using System.Runtime.InteropServices;
3 using System.Windows;
4 using System.Windows.Controls;
5
6 namespace WpfApp
7 {
8     /// <summary>
9     /// MainWindow.xaml 的交互逻辑
10    /// </summary>
11    2 个引用
12    public partial class MainWindow : Window
13    {
14        [DllImport("Dll1.dll", EntryPoint = "ADD", CallingConvention = CallingConvention.Cdecl)]
15        1 个引用
16        public static extern double ADD(double a, double b);
17
18        [DllImport("Dll1.dll", EntryPoint = "MIN", CallingConvention = CallingConvention.Cdecl)]
19        1 个引用
20        public static extern double MIN(double a, double b);
21
22        [DllImport("Dll1.dll", EntryPoint = "MUL", CallingConvention = CallingConvention.Cdecl)]
23        1 个引用
24        public static extern double MUL(double a, double b);
25
26        [DllImport("Dll1.dll", EntryPoint = "SUB", CallingConvention = CallingConvention.Cdecl)]
27        1 个引用
28        public static extern double SUB(double a, double b);
29    }
30 }
```

## 3. LoaderLock异常

```
MainWindow.xaml MainWindow.xaml.cs WpfApp.MainWindow Btn_Click(object :
34 case 1:
35     ret = ADD(lef, rig);
36     break;
37 case 2:
38     ret = MIN(lef, rig);
39     break;
```

已引发异常

托管调试助手 "LoaderLock": "DLL 'C:\Users\Matthew\Desktop\C#混用C++\WpfApp\WpfApp\bin\x86\Debug\Dll1.dll' 正尝试在 OS 加载程序锁内执行托管代码。不要尝试在 DllMain 或映像初始化函数内运行托管代码。这样做会导致应用程序挂起。"

复制详细信息

异常设置

☒ 引发此异常类型时中断

从以下位置引发时除外:

☐ WpfApp.exe

打开异常设置 | 编辑条件

[illegible]