**MFC学习记录001——VS2019MFC创建并调用DLL（初学者详细步骤）**

## 第一节：DLL的创建

### 一、创建"MFC动态链接库"项目 在这里插入图片描述

### 二、项目名称设置为“Test001Dll"——>选择好项目存放的位置——>“使用共享 MFC DLL的常规 DLL” 这时候可以看到自动生成了这些文件： 在这里插入图片描述

### 然后打开资源视图,资源文件夹上面点击右键->添加-资源,在弹出的窗口中选中Dialog点击ok,然后把对话框的id改为Test001Dialog\_IDD

|  |
| --- |
|  |

### 三、项目右击——>“添加”——>“新建项”——>选择“MFC类”——>类名设置为“Test001Dialog” 在这里插入图片描述 基类设置为“CDialog”——>填写对话框的ID（这里一定不能忘！！） 在这里插入图片描述 可以看到这里多了几个文件： 在这里插入图片描述

### 四、项目右击——>“属性” ——>“C/C++”——>“预处理器”——>“预处理器定义”——>点击右边小框——>“编辑” 在这里插入图片描述 如下图所示，拖到最下面一行，添加“DLLEXPORY”在这里插入图片描述

### 五、点开要导出的类的头文件，我这里是后来添加的"CDialog"，添加如下代码，将这个类“设置”为导出类：

//将类作为导出类

#ifndef DLLEXPORY

#define DLL\_EXPORY \_declspec(dllimport)

#else

#define DLL\_EXPORY \_declspec(dllexport)

#endif

#include "resource.h"

#### 并在导出的类的定义前面加上“DLL\_EXPORY”，效果见下图： 在这里插入图片描述

### 六、在类的源文件(即“Test001Dialog.cpp”)中定义一个导出函数，代码如下：

//定义一个导出函数，该函数返回导出界面基类的指针。

//导出接口\*\*加粗样式\*\*

extern "C" DLL\_EXPORY Test001Dialog \* ShowDialog()

{

AFX\_MANAGE\_STATE(AfxGetStaticModuleState());

Test001Dialog\* ph = new Test001Dialog;

ph->DoModal();

return ph;

}

|  |
| --- |
|  |

### 七、点击“生成”——>“生成解决方案”，会发现项目的Debug文件夹底下会生成这样几个文件： 在这里插入图片描述

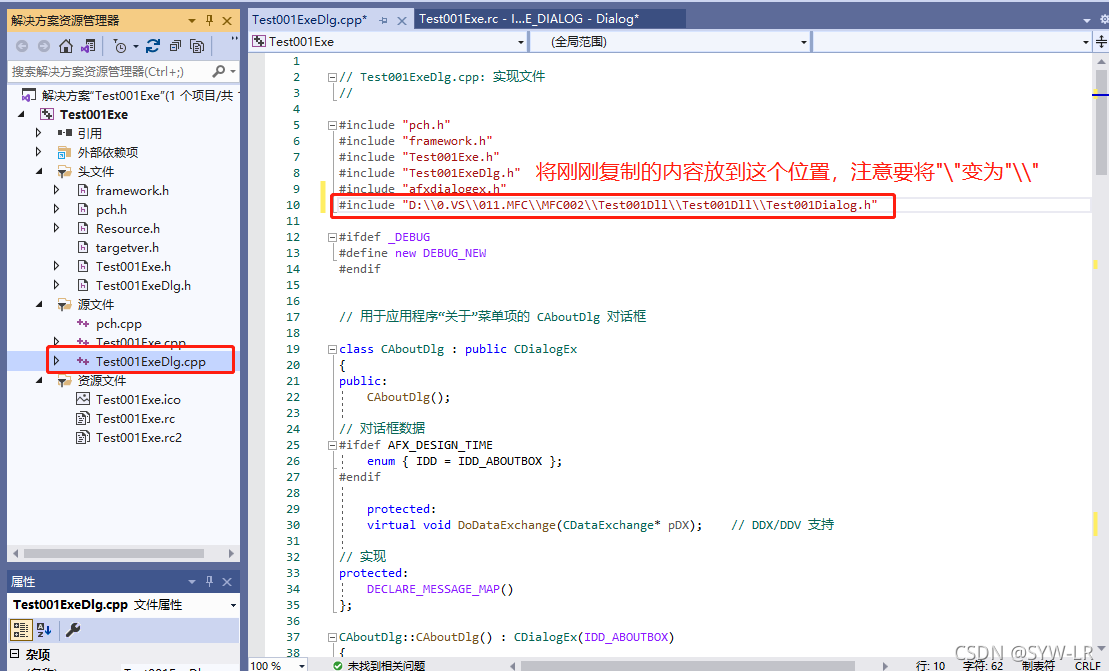
## 第二节：调用DLL

### 一、创建MFC应用程序 在这里插入图片描述 项目名设置为"Test001Exe"——>应用程序类型选择“基于对话框” 在这里插入图片描述 创建成功之后，界面是这个样子的：

### 在这里插入图片描述 二、添加头文件（刚刚创建的DLL的头文件） 位置获取方法如下： 找到刚刚DLL项目的文件夹打开——>找到“Test001Dialog.h”右击——>“属性”——>“安全”——>复制 在这里插入图片描述 在“Test001ExeDlg.cpp”中添加刚刚复制的内容：

#include "D:\\0.VS\\011.MFC\\MFC002\\Test001Dll\\Test001Dll\\Test001Dialog.h"

* 1



### 三、添加链接器 项目右击——>“属性”——>“链接器”——>“输入”——>“附加依赖项”——>“Test001Dll.lib” 在这里插入图片描述

### 四、打开界面，从工具箱拖一个按钮到界面中，双击按钮添加按钮动作（调用DLL）

// TODO: 在此添加控件通知处理程序代码

// 动态调用DLL

HINSTANCE m\_hDLL = LoadLibrary(TEXT("Test001Dll.dll"));

if (NULL == m\_hDLL)

{

MessageBox(TEXT("加载DLL失败"));

}

typedef Test001Dialog\* (\*hpDllFun)();

hpDllFun pShowDlg = (hpDllFun)GetProcAddress(m\_hDLL, "ShowDialog");

if (NULL == pShowDlg)

{

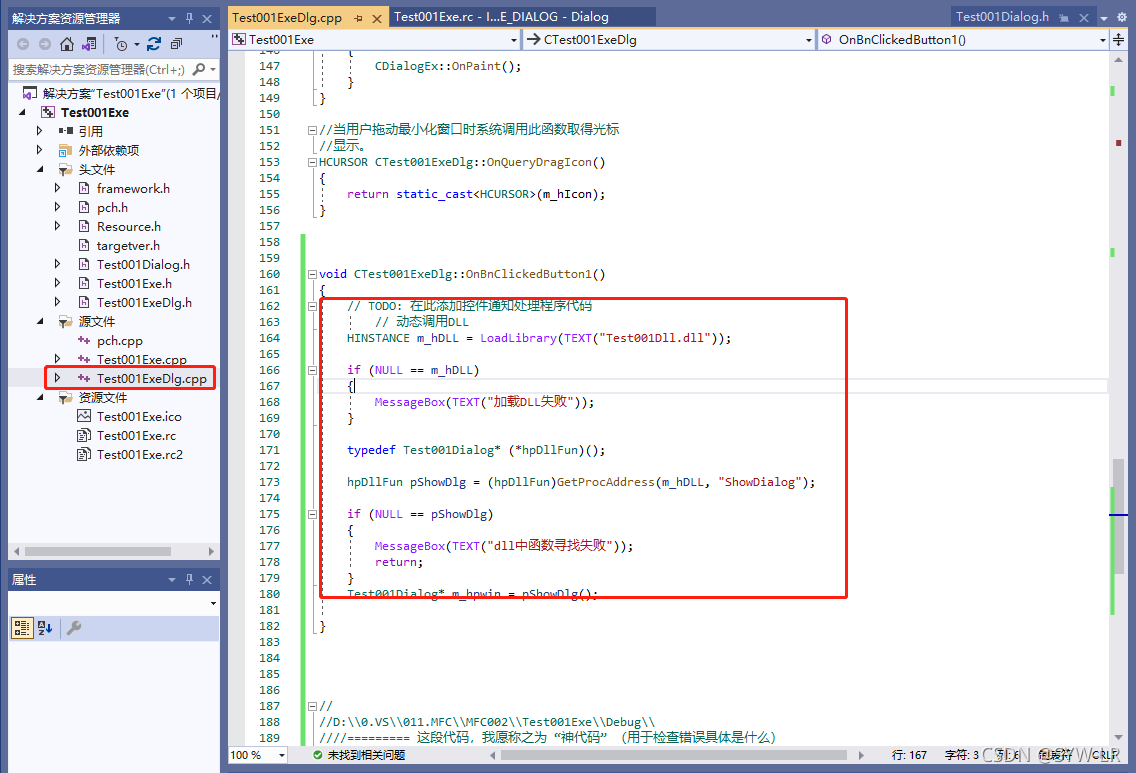
MessageBox(TEXT("dll中函数寻找失败"));

return;

}

Test001Dialog\* m\_hpwin = pShowDlg();





### 五、移动DLL的文件 将DLL项目底下的.dll和.lib文件移动到应用程序的如图位置： 在这里插入图片描述

### 六、运行 在这里插入图片描述

### 七.在使用完之后，调用下面函数卸载动态库：

#### FreeLibrary(m\_hDll);