**一．比较动态链接库、静态库、import库区别（主要补充import库）**

在windows下一般可以看到后缀为dll和后缀为lib的文件，但这两种文件可以分为三种库，分别是动态链接库（Dynamic-Link Library），**静态库(Static Library)**和导入库（Import Library），下面一一说明这三种库。

**1.动态链接库(Dynamic Linked Library)：**  
Windows为应用程序提供了丰富的函数调用，这些函数调用都包含在动态链接库中。

其中有3个最重要的DLL，

* Kernel32.dll

它包含那些用于管理内存、进程和线程的函数，例如CreateThread函数；

* User32.dll

它包含那些用于执行用户界面任务(如窗口的创建和消息的传送)的函数，例如CreateWindow函数；

* GDI32.dll

它包含那些用于画图和显示文本的函数。

**2.静态库(Static Library)：**  
函数和数据被编译进一个二进制文件(通常扩展名为.LIB)。在使用静态库的情况下，在编译链接可执行文件时，链接器从库中复制这些函数和数据并把它们和应用程序的其它模块组合起来创建最终的可执行文件(.EXE文件)。

**3.导入库(Import Library)：**  
在使用动态链接库的时候，往往提供两个文件：一个引入库和一个DLL。引入库包含被DLL导出的函数和变量的符号名，DLL包含实际的函数和数据。在编译链接可执行文件时，只需要链接引入库，DLL中的函数代码和数据并不复制到可执行文件中，在运行的时候，再去加载DLL，访问DLL中导出的函数。

在运行Windows程序时，它通过一个被称作“动态链接”的进程与Windows相接。一个Windows的.EXE文件拥有它使用不同动态链接库的引用，所使用的函数即在那里。当Windows程序被加载到内存中时，程序中的调用被指向DLL函数的入口，如果DLL不在内存中，系统就将其加载到内存中。

当链接Windows程序以产生一个可执行文件时，必须链接由编程环境提供的专门的“导入库(import library)库”。这些导入库包含了动态链接库名称和所有Windows函数调用的引用信息。链接程序使用该信息在.EXE文件中构造一个表，当加载程序时，Windows使用它将调用转换为Windows函数。

**静态库与导入库的区别**：  
导入库和静态库实质是不一样的东西，区别很大。静态库本身就包含了实际执行代码、符号表等等，而对于导入库而言，其实际的执行代码位于**动态库**中，导入库只包含了地址符号表等，确保程序找到对应函数的一些基本地址信息。

# **如何区分windows 导入库与静态库：**

windows中，动态库包括xx.dll和xx.lib两部分。静态库包含yy.lib。  
此处有种lib，一种称为动态库中的导入库，xx.lib。一种称为静态库的yy.lib，区分方法如下：

第一看大小，导入库一般都很小，大的也只有几k，而静态库则很大，小则几M，大则几十M。

第二就要通过命令的方式来查看。（如lib /list Demo.lib）

****对动态链接的一些补充：****

****DLL的部署位置****

DLL文件必须在特定目录下才能被程序所找到，这些特定的路径包括：

1.可执行文件所在目录

2.进程当前目录

3.系统目录

4.Windows目录

5.环境变量PATH标明的目录

应用程序使用DLL可以采用两种方式：一种是载入时动态链接，另一种是运行时动态链接。  
1.载入时动态链接就是在程序开始执行时就将DLL文件加载到应用程序当中。实现载入时动态链接时，将导入函数关键字\_declspec(dllimport)函数名等写到应用程序相应的头文件中就可以了。  
//Dlltest.h    
#pragma comment(lib，"MyDll.lib")    
extern "C"\_declspec(dllimport) int Max(int a,int b);    
extern "C"\_declspec(dllimport) int Min(int a,int b);    
//TestDll.cpp    
#include    
#include"Dlltest.h"    
void main()    
{

int a;    
a=min(8,10)    
printf("比较的结果为%d "，a);    
}    
  
2．运行时动态链接  
　　运行时动态链接是应用程序在执行过程中随时可以加载DLL文件，也可以随时卸载DLL文件，这是载入时动态链接所无法做到的，所以运行时动态链接具有更好的灵活性，对于解释性语言更为合适。不过实现运行时动态链接要麻烦一些。在应用程序中用LoadLibrary或MFC提供的AfxLoadLibrary将自己所做的动态链接库调进来，动态链接库的文件名即是上述两个函数的参数，此后再用GetProcAddress()获取想要引入的函数。在应用程序退出之前，应该用FreeLibrary或MFC提供的AfxFreeLibrary释放动态链接库。  
  
void main(void)    
{    
typedef int(\*pMax)(int a,int b);    
typedef int(\*pMin)(int a,int b);    
HINSTANCE hDLL;    
PMax Max    
HDLL=LoadLibrary("MyDll.dll");//加载动态链接库MyDll.dll文件；    
Max=(pMax)GetProcAddress(hDLL,"Max");    
A=Max(5,8);    
Printf("比较的结果为%d "，a);    
FreeLibrary(hDLL);//卸载MyDll.dll文件；    
}

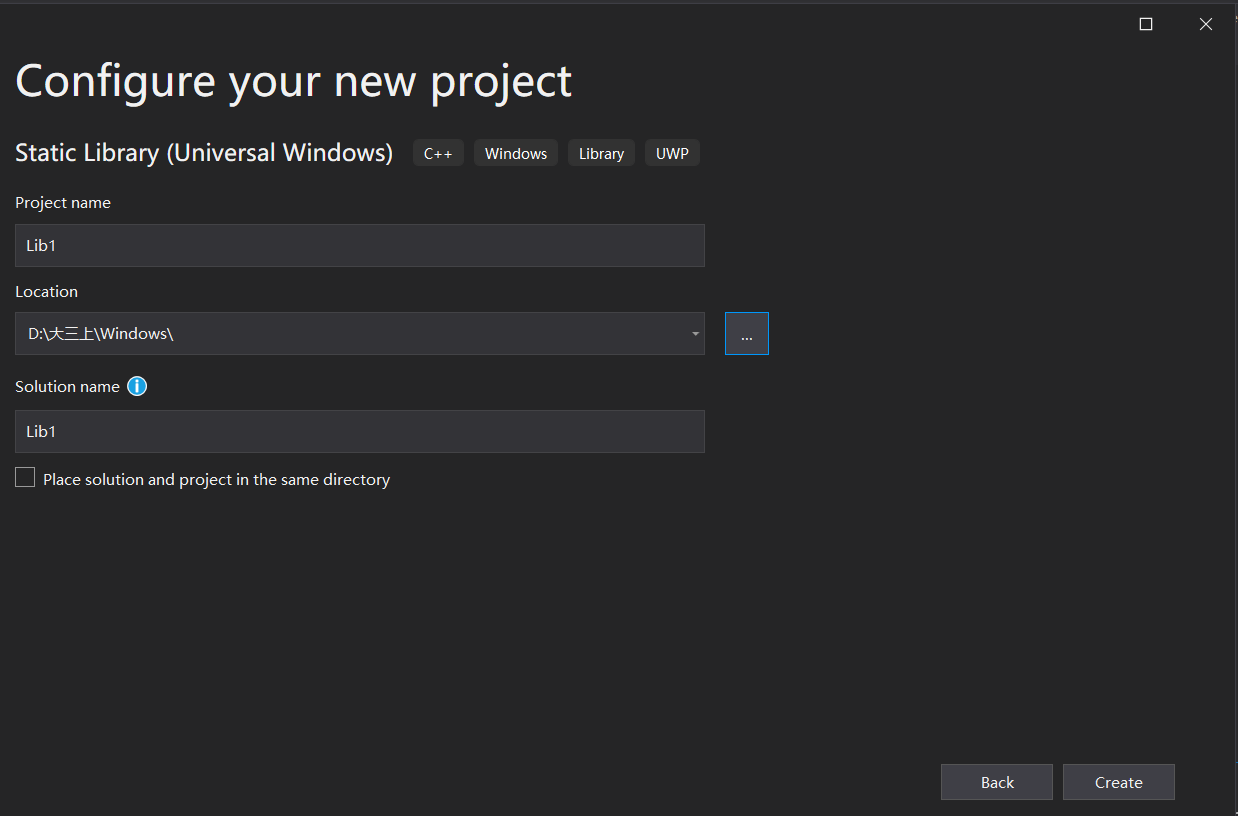
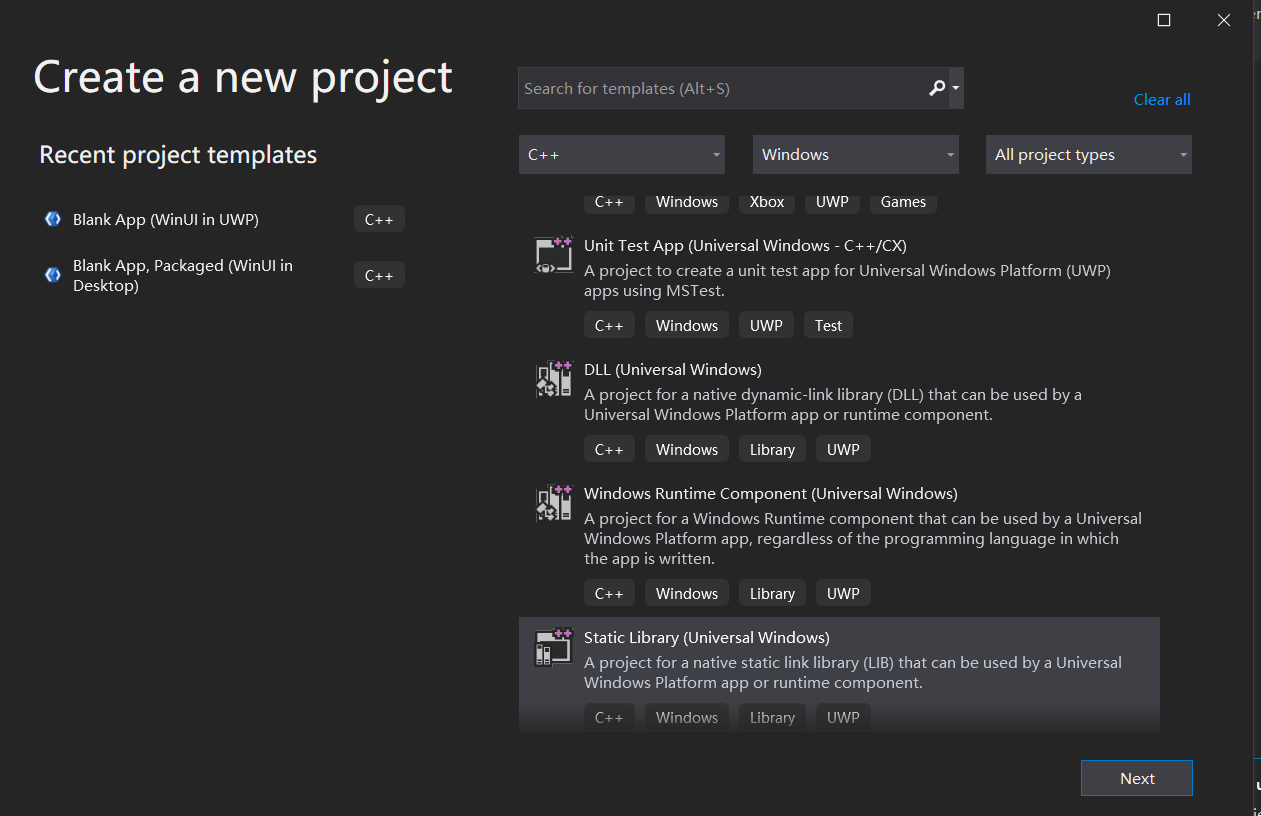
**对静态链接和动态链接的一点归纳：**

在C语言中，要生成可执行文件，必须经历两个阶段，即编译、链接。在编译过程中，只有编译，不会涉及到链接。在链接过程中，静态链接和动态链接就出现了区别。静态链接的过程就已经把要链接的内容已经链接到了生成的可执行文件中，就算现在去把静态库删除也不会影响可执行程序的执行；而动态链接这个过程却没有把内容链接进去，而是在执行的过程中，再去找要链接的内容，生成的可执行文件中并没有要链接的内容，所以删除动态库时，可执行程序就不能运行。

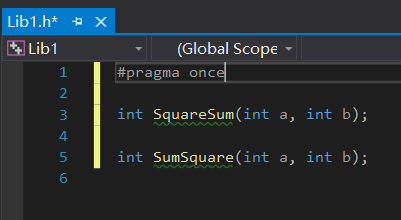
静态链接和动态链接的区别（按点描述）：  
1.时期：（最大区别）  
静态库在编译时连接，在链接时拷贝  
动态库在运行时连接  
2.资源  
静态库在每次使用时会全部连接进可执行程序，浪费资源。  
动态库在使用时访问动态库中函数，节省资源。  
3.更新升级  
静态库更新，则每个使用该静态库的程序都需要更新，不易于更新升级  
动态库仅更新自身，易于更新升级  
4.包含其他库  
静态链接库不能再包含其他动态链接库  
动态链接库可以包含其他动态链接库

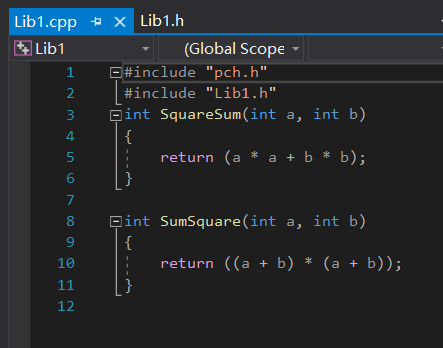
1. **创立静态库的步骤**

**（1）创立静态库项目**

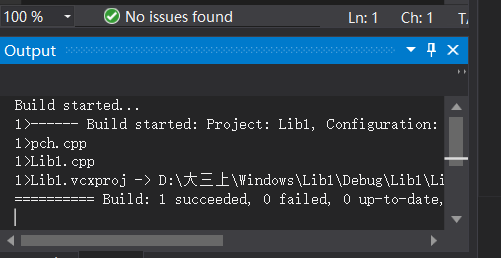


**（2）添加.h和.cpp文件**





**（3）生成.lib文件（点击生成解决方案，得到Lib.lib文件）**



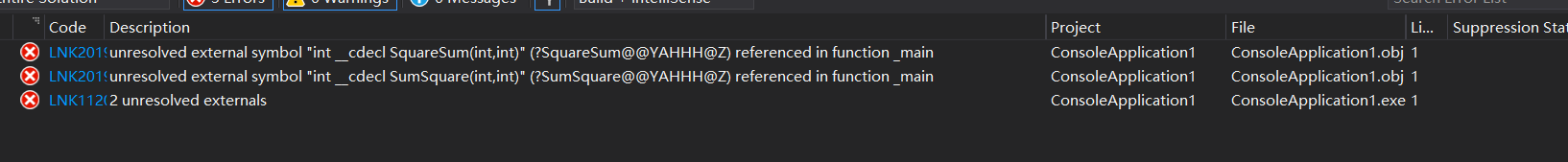


## ****二．使用静态库****

**（1）正常一样创建一个控制台应用程序**

**（2）在文件中添加头文件：Lib1.h（注意这个头文件的位置，建议拷贝到当前项目目录下）**

**如果在这里直接运行，会报错：**

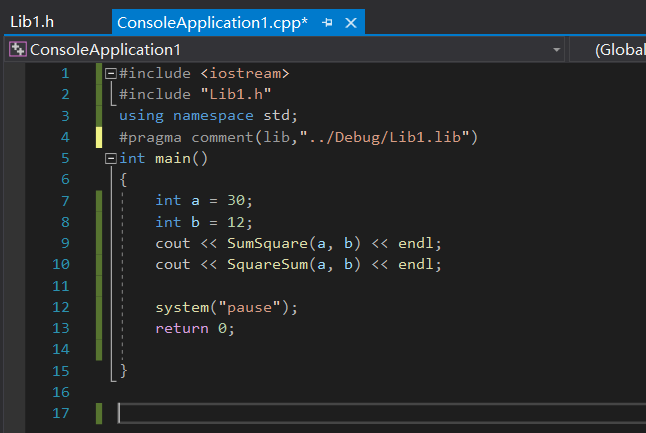


**（3）在项目中添加lib文件（选择在文件中增加语句）**



**（注意lib文件的位置，这里可以添加相对路径，也可以添加绝对路径，建议也把lib文件放在当前项目下）**

**实例如下：**



**运行结果：**



**（如果删除Lib1.lib文件，双击.exe文件也能得到这样的结果）**