# 创建静态库并使用

1 创建static lib 项目。

2 创建类。

.h文件

<javascript:void(0);>

// TestStaticLib.h: interface for the TestStaticLib class.

//

//////////////////////////////////////////////////////////////////////

#if !defined(AFX\_TESTSTATICLIB\_H\_\_FBC7FEBA\_1A54\_4DBB\_A54A\_E12FA2266B93\_\_INCLUDED\_)

#define AFX\_TESTSTATICLIB\_H\_\_FBC7FEBA\_1A54\_4DBB\_A54A\_E12FA2266B93\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

int Plus(int x, int y);

int Sub(int x, int y);

int Div (int x, int y);

#endif // !defined(AFX\_TESTSTATICLIB\_H\_\_FBC7FEBA\_1A54\_4DBB\_A54A\_E12FA2266B93\_\_INCLUDED\_)

<javascript:void(0);>

.cpp文件

// TestStaticLib.cpp: implementation of the TestStaticLib class.

//

//////////////////////////////////////////////////////////////////////

#include "TestStaticLib.h"

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

int Plus(int x, int y)

{

return x+y;

}

int Sub(int x, int y)

{

return x-y;

}

int Div (int x, int y)

{

return x/y;

}

编译成功。拿到TestStaticLib.lib和TestStaticLib.h头文件

2 使用。

在控制台程序中，

<javascript:void(0);>

#include "stdafx.h"

#include <windows.h>

// 1

#include "TestStaticLib.h"

// 2

#pragma comment(lib,"TestStaticLib.lib")

int main(int argc, char\* argv[])

{

// 3 调用函数

int x = Plus(2,3);

return 0;

}

还有一种使用方法

上面的第二步不用了，VC6中，project-》setting-〉Link里面，Object/library modules 加上TestStaticLib.lib

# 创建动态链接库并使用

自己创建dll项目，编译后拿到.dll文件和.lib文件。

.h文件如下：（为了测试 没加extern “c” 函数前面也没加调用约定 最好加上\_stdcall）

// MyDll.h: interface for the MyDll class.

//

//////////////////////////////////////////////////////////////////////

#if !defined(AFX\_MYDLL\_H\_\_7FB2193F\_D30D\_434B\_A7DC\_4A3A611D38BB\_\_INCLUDED\_)

#define AFX\_MYDLL\_H\_\_7FB2193F\_D30D\_434B\_A7DC\_4A3A611D38BB\_\_INCLUDED\_

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

\_declspec(dllexport) int Plus(int x, int y);

\_declspec(dllexport) int Sub(int x, int y);

\_declspec(dllexport) int Div (int x, int y);

#endif // !defined(AFX\_MYDLL\_H\_\_7FB2193F\_D30D\_434B\_A7DC\_4A3A611D38BB\_\_INCLUDED\_)

.cpp文件如下：

1 // MyDll.cpp: implementation of the MyDll class.

2 //

3 //////////////////////////////////////////////////////////////////////

4

5 #include "stdafx.h"

6 #include "MyDll.h"

7

8 //////////////////////////////////////////////////////////////////////

9 // Construction/Destruction

10 //////////////////////////////////////////////////////////////////////

11 int Plus(int x, int y)

12 {

13 return x+y;

14 }

15 int Sub(int x, int y)

16 {

17 return x-y;

18 }

19 int Div (int x, int y)

20 {

21 return x/y;

22 }

可用depends查看.dll文件：

有两种方法 **显示链接** 和 **隐式链接**

**1 显示链接**

#include "stdafx.h"

#include <windows.h>

int main(int argc, char\* argv[])

{

// 1 定义函数指针

typedef int(\*lpPlus)(int,int);

// 2 声明函数指针变量

lpPlus myplus;

// 3 动态加载dll到内存中

HINSTANCE hModule = LoadLibrary("TestDll.dll");

// 4 获取函数地址 函数名使用depends可获得

myplus = (lpPlus)GetProcAddress(hModule,"?Plus@@YAHHH@Z");

// 5 调用函数

int x = myplus(2,3);

return 0;

}

**2 隐式链接**

#include "stdafx.h"

#include <windows.h>

//1

#pragma comment(lib,"TestDll.lib")

//2

\_declspec(dllimport) int Plus(int x, int y);

int main(int argc, char\* argv[])

{

// 3 调用函数

int x = Plus(2,3);

return 0;

}

--------------

上面是用dllexport导出，还有一种导出方式.def。**这种方式可以隐藏函数名**

创建dll项目，头文件和app文件和普通程序一样，项目中新增比如叫test.def 定义如下：

EXPORTS

Plus @12

Sub @13

Div @15 NONAME