

## 一.C++调用 Win32 项目创建的 dll

链接: <http://www.it610.com/article/1954631.htm>

库文件: E:\编写 C 的 Dll 并调用\最简单的使用,普通函数\testDll

调用文件: E:\编写 C 的 Dll 并调用\最简单的使用,普通函数\ImportEdrlibDll

### 一. 创建动态链接库

1.文件->新建->项目->win32 项目, 填写好项目名称, 点击“下一步”,  
应用程序类型选择: “DLL(D)”, 附加选项: 空项目(E), 然后完成。

### 2.编写头文件(edrlib.h)

```
#ifdef __cplusplus
#define EXPORT extern "C" __declspec(dllexport)
#else
#define EXPORT __declspec(dllexport)
#endif
```

```
EXPORT void  EdrCenterTextA();
EXPORT void  EdrCenterTextW();
EXPORT int   IncCounter();
#ifdef UNICODE
#define EdrCenterText EdrCenterTextW
#else
#define EdrCenterText EdrCenterTextA
#endif
```

注解:

i. 定义 \_\_cplusplus 表示是供 C++程序中调用。

ii.\_\_declspec(dllexport)表示函数调用方式,此处表示 VS2008 工程属性调用的默认方式。

更改方法: 右击项目--属性--配置属性--C/C++--高级, 更改右侧“调用约定”。vs2013 暂时不知道怎么更改

### 3.编写 DLL 文件(edrlib.cpp)

```
#include "windows.h"
#include "edrlib.h"
//counter 供调用该 DLL 的所有应用程序共享
#pragma data_seg("shared")
int counter=0;
#pragma comment(linker, "/SECTION:shared,RWS")

int WINAPI DllMain(_In_ HANDLE _HDllHandle, _In_ DWORD _Reason, _In_opt_ LPVOID _Reserved)
{
    return TRUE;
}

EXPORT void  EdrCenterTextA()
{
    MessageBox(NULL,TEXT("调用 DLL 函数! "),TEXT("ASSIC 版本"),MB_OK);
}
```

```

}
EXPORT void  EdrCenterTextW()
{
    MessageBox(NULL,TEXT("调用 DLL 函数! "),TEXT("UNICODE 版本"),MB_OK);

}

EXPORT int  IncCounter()
{
    return ++counter;
}

```

编译生成 DLL.

**注意.h 声明和.cpp 函数名对应!!**

二. 调用 DLL(重点配置 lib,.h 路径)

- 1.新建 Win32 应用程序空项目。
- 2.把生成的 dll、lib 这 2 个文件复制到新建的目录下。
- 3.添加导入库： 调试->项目属性(最后一项)->配置属性->VC++目录  
包含目录:.h 文件全路径  
库目录:.lib 文件全路径

**注意要与文件路径对应,否则会报 LNK1104 找不到\*\*\*.lib 的错误!!!**

调试->项目属性(最后一项)->链接器->输入->附加依赖项中添加导入.lib 的名字

- 4.头文件： edrlib.h（与编写 DLL 中一样）

- 5 源文件： main.cpp

```

/*#define cplusplus*/ //c++调用方式
#include "windows.h"
#include "edrlib.h"

```

**#pragma comment(lib,"testDll.lib") //注意这里是库的名字**

```

int WINAPI WinMain(__in HINSTANCE hInstance, __in_opt HINSTANCE hPrevInstance, __in_opt LPSTR lpCmdLine, __in int nShowCmd )
{

    TCHAR buf[32];
    wprintf(buf,L"now,counter=%i",IncCounter());
    MessageBox(NULL,buf,L"--",MB_OK);
    EdrCenterText();
    return 0;
}

```

## 二. C++调用 C dll 使用.def 文件,只调试不运行,运行出错

链接: <http://blog.csdn.net/u010273652/article/details/25514577>

库文件: E:\编写 C 的 Dll 并调用\最简单的使用,普通函数\CDll

调用文件: E:\编写 C 的 Dll 并调用\最简单的使用,普通函数\ImportCDlltest

### 一.创建动态库方法:

创建动态库是生成 .dll .lib 两个文件

1.文件 -> 新建 -> 项目 -> win32 控制台应用程序 项目名称: CDll

应用程序类型: DLL 附加选项: 空项目

2. 建立源文件 dllgenerator.cpp (不需要主函数, 只写你需要动态调用的函数)

3. 建立头文 dllgenerator.h (一定要把函数声明写好)

4. 建立源文件 dllmain.cpp (一定要包含#include<windows.h>)

- 下面的函数是 VS2013 自动生成, 没生成复制即可。其实是在创建工程项目时不选空项目时候 vs 自动生成的, 但不选空项目会产生很多其他文件和依赖项, 本文教你建立最纯净的工程, 注意包含 window.h

```
1. #include <windows.h>
2. BOOL APIENTRY DllMain(HMODULE hModule,
3. DWORD ul_reason_for_call,
4. LPVOID lpReserved
5. )
6. {
7. switch (ul_reason_for_call)
8. {
9. case DLL_PROCESS_ATTACH:
10. case DLL_THREAD_ATTACH:
11. case DLL_THREAD_DETACH:
12. case DLL_PROCESS_DETACH:
13. break;
14. }
15. return TRUE;
16. }
```

建立源文件 Source.def

- 建立方式如下图, 选择 VisualC++ -> 代码 -> 模块定义文件(.def)

- 在此文件第一行引号中填你的项目名称即可, EXPORTS 下面列出要生成的函数名称 格式: “名称 @序号” )

[cpp] view plain copy



```
1. LIBRARY "DLLGenerator"
```

```

2.  EXPORTS
3.      Add @1
4.      Mul @2

```

点击菜单栏 生成 -> 生成解决方案 或者 按 Ctrl+F5

输出栏显示 创建库 DLLGenerator.lib 和 生成 DLLGenerator.dll

创建成功

项目调用动态库方法如下:

调用动态库是调用 .dll .lib .h 三个文件

1.文件 -> 新建 -> 项目 -> win32 控制台应用程序 项目名称: ImportCDlltest

应用程序类型: 控制台应用程序 附加选项: 空项目

2.配置.h,.lib 路径

打开项目 -> 项目属性 -> 配置属性 -> VC++目录 -> 包含目录 加入 DLLGenerator.h 所在目录的绝对路径 (E:\dlltest\DLL), 以分号间隔

打开项目 -> 项目属性 -> 配置属性 -> VC++目录 -> 库目录 加入 DLLGenerator.lib 所在目录的绝对路径 (E:\dlltest\DLL)

打开项目 -> 项目属性 -> 配置属性 -> 链接器 -> 输入 -> 附加依赖项 中填上 DLLGenerator.lib (以分号间隔), 点击右下角 应用 确定

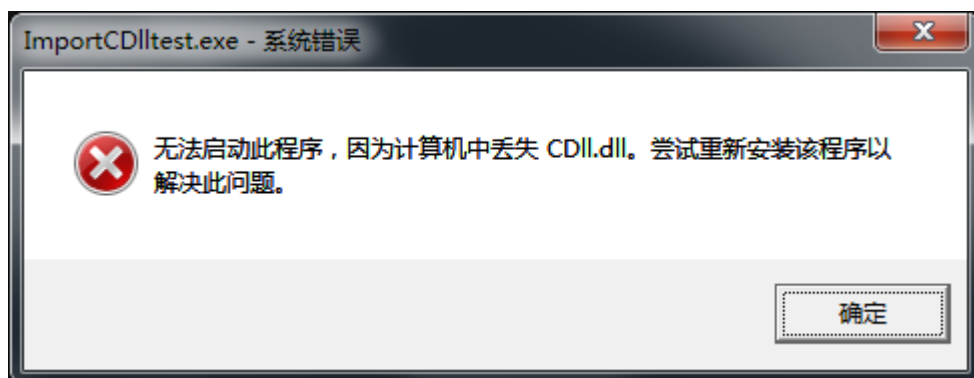
3. 添加已有项\*.h,新建源文件 main.cpp,其他写法与 C++一样

```

1. #include<iostream>
2. #include"dllgenerator.h"
3. using namespace std;
4. int main(){
5.     int a = 2;
6.     int b = 3;
7.     cout << Add(a, b) << endl;
8.     cout << Mul(a, b) << endl;
9. }

```

5. 点击菜单栏,调试->开始执行(不调试) 或者按 F5,不能直接运行,会报错



6.

### 三.C++调用 C dll,只调试不运行,否则报错

链接: <http://developer.51cto.com/art/201105/262715.htm>

动态连接库的创建步骤:

#### 一、创建 Non-MFC DLL 动态链接库

1、打开文件—> 新建 —> 项目选项, 选择 Win32 控制台 勾选 Dll —>sample project —>工程名: DllDemo

2、新建一个。h 文件 DllDemo.h

```
1. #ifdef DllDemo_EXPORTS
2. #define DllAPI __declspec(dllexport)
3. #else
4. #define DllAPI __declspec(dllimport)
5. extern "C" //原样编译
6. {
7. DllAPI int __stdcall Max(int a,int b); //__stdcall 使非 C/C++语言内能够调用 API
8. }
9. #endif
```

3、在 DllDemo.cpp 文件中导入 DllDemo.h 文件, 并实现 Max (int, int) 函数

#include "DllDemo.h"

```
1. DllAPI int __stdcall Max(int a,int b)
2. {
3. if(a==b)
4. return NULL;
5. else if(a>b)
6. return a;
7. else
8. return b;
9. }
```

4、编译程序生成动态连接库

动态链接的调用步骤:

#### 一、隐式调用

1、建立 DllCns1Test 工程

2、将文件 DllDemo.dll、DllDemo.lib 拷贝到 DllCns1Test 工程所在的目录

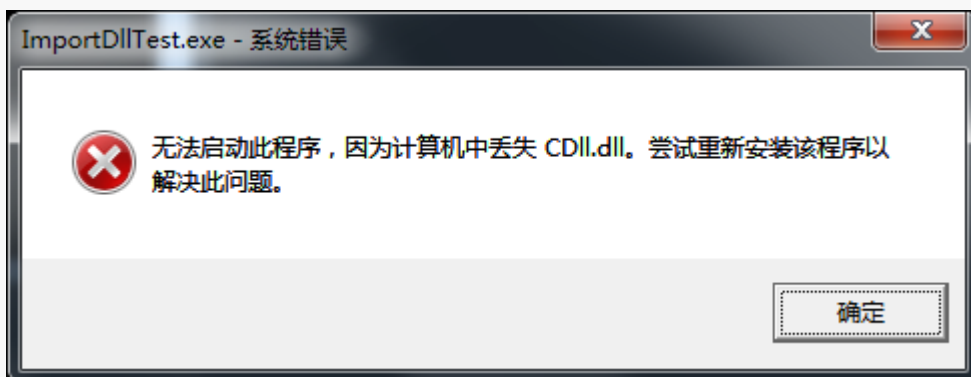
3、在 DllCns1Test.h 中添加如下语句：

```
1. #define DllAPI __declspec(dllimport)
2. #pragma comment(lib, "DllDemo.lib") //在编辑器 link 时，链接到 DllDemo.lib 文件
3. extern "C"
4. {
5. DllAPI int __stdcall Max(int a,int b);
6. }
```

4、在 DllCns1Test.cpp 文件中添加如下语句： #include "DllCns1Test.h"//或者  
#include "DllDemo.h"

```
1. void main()
2. {
3. int value;
4. value = Max(2,9);
5. printf("The Max value is %d\n",value);
6. }
```

5、编译并生成应用程序 DllCns1Test.exe, 调试->开始执行(不调试), 直接运行报错



不用配置.h 以及 lib 的路径, 为什么?

