# 动态链接库

**动态链接库**（**Dynamic Link Library**或者 **Dynamic-link Library**，缩写为 **DLL**），是微软公司在微软Windows操作系统中，实现共享函数库概念的一种方式。这些库函数的扩展名是 ”.dll"、".ocx"（包含ActiveX控制的库）或者 ".drv"（旧式的系统驱动程序）。

[动态链接](http://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E9%93%BE%E6%8E%A5)提供了一种方法，使进程可以调用不属于其[可执行代码](http://baike.baidu.com/item/%E5%8F%AF%E6%89%A7%E8%A1%8C%E4%BB%A3%E7%A0%81)的函数。函数的可执行代码位于一个 DLL 文件中，该 DLL 包含一个或多个已被编译、链接并与使用它们的进程分开存储的函数。DLL 还有助于共享数据和资源。多个应用程序可同时访问内存中单个 DLL 副本的内容。

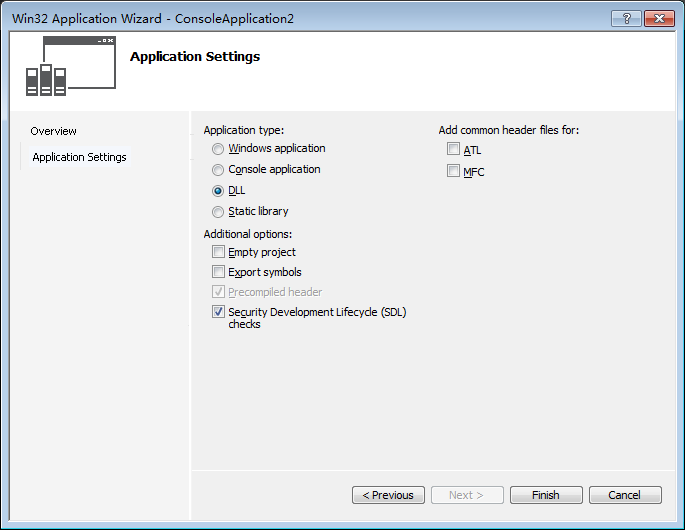
使用动态链接库可以更为容易地将更新应用于各个[模块](http://baike.baidu.com/item/%E6%A8%A1%E5%9D%97)，而不会影响该程序的其他部分。例如，您有一个大型网络游戏，如果把整个数百MB甚至数GB的游戏的代码都放在一个应用程序里，日后的修改工作将会十分费时，而如果把不同功能的代码分别放在数个[动态链接](http://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E9%93%BE%E6%8E%A5)库中，您无需重新生成或安装整个程序就可以应用更新。

详情：

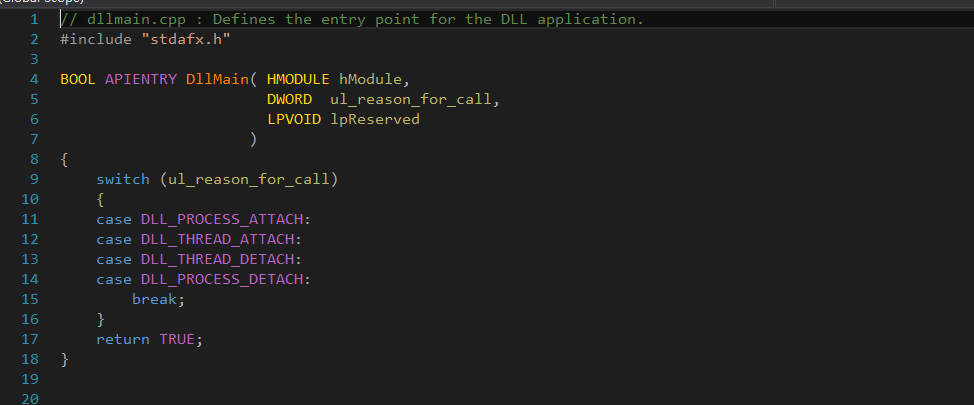
<http://baike.baidu.com/link?url=GS9VKpL7Fgqc9vQzydwtOh8bHZATTVJlxkzrZIEuQbmsfuOeFcrBrJ12gw2qsOKMxrkNR4cxukaEFW2VyKX1pPixk6kS8pl0x1yyG1jnra-5Gm2pPuPUVj1hOZRk7te11eSqJ5GuuUYg3sm-aGxYZq>

实现步骤：

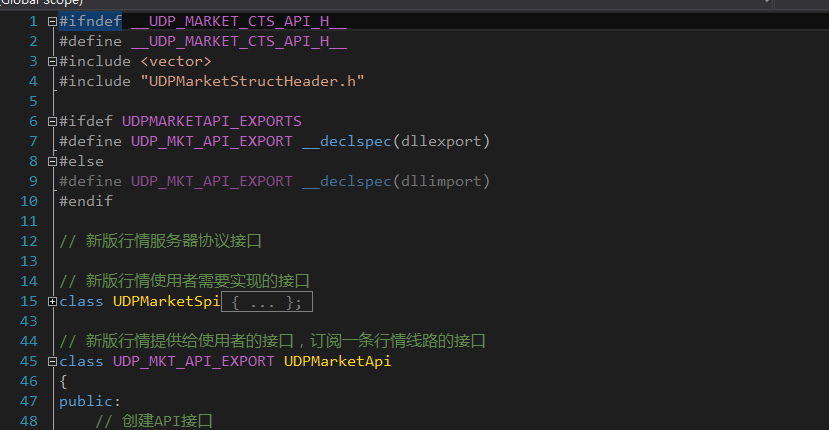
1：定义一个DLL项目的入口函数



2:自动生成dll的出入口函数



3：给dll中加入一些可以调用的函数或者类



其中，我们看到的

#ifndef \_\_UDP\_MARKET\_CTS\_API\_H\_\_

#define \_\_UDP\_MARKET\_CTS\_API\_H\_\_

#endif

这三段代码，是为了防止头文件的重定义，虽然#progma once也可以防止重定义，但是其只适用于c++环境之中，而我们的dll不一定是给c++开发使用，使用#ifdef可以避免该问题。

#include <vector>

#include "UDPMarketStructHeader.h"

头文件中包含了一些自己所需要的头文件，主要是为了解决该头文件的自依赖性，避免给使用者造成不便。

#ifdef UDPMARKETAPI\_EXPORTS

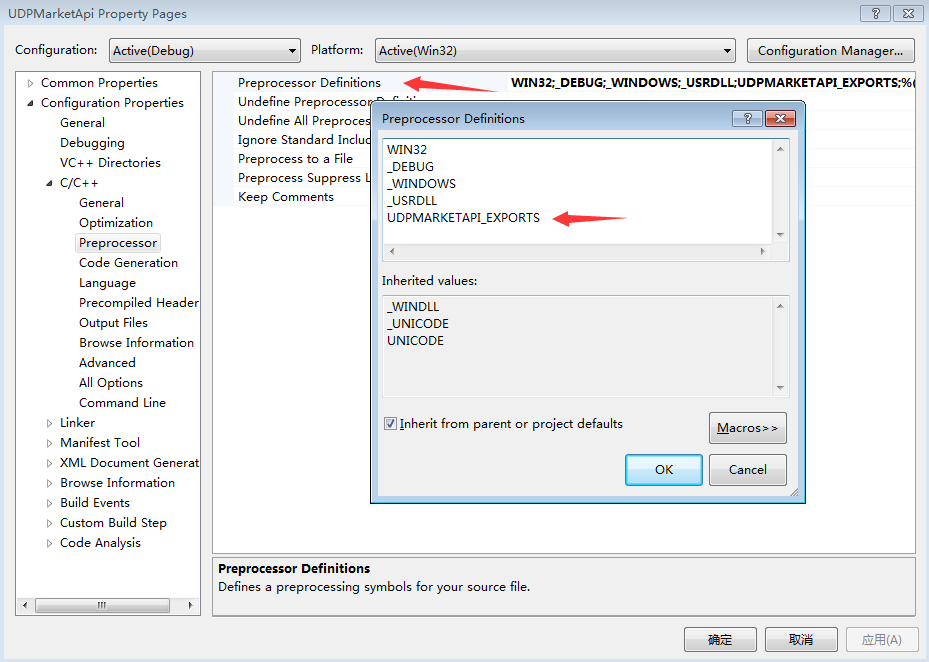
#define UDP\_MKT\_API\_EXPORT \_\_declspec(dllexport)

#else

#define UDP\_MKT\_API\_EXPORT \_\_declspec(dllimport)

#endif

预编译宏，可以通过图中的方式



将UDPMARKETAPI\_EXPORTS加入dll工程的宏定义之中，当其他工程没有加入时，就可以进入#define UDP\_MKT\_API\_EXPORT \_\_declspec(dllimport)到处宏定义，实现dll导出、导入接口的智能化。

之后就可以定义class UDP\_MKT\_API\_EXPORT UDPMarketApi这样类型的导出类，而使用的时候只需初始化该类的接口函数static UDPMarketApi \*CreateUDPMarketApi(char \*pcLogPath)，就可以使用该导出类了

3：编译完成

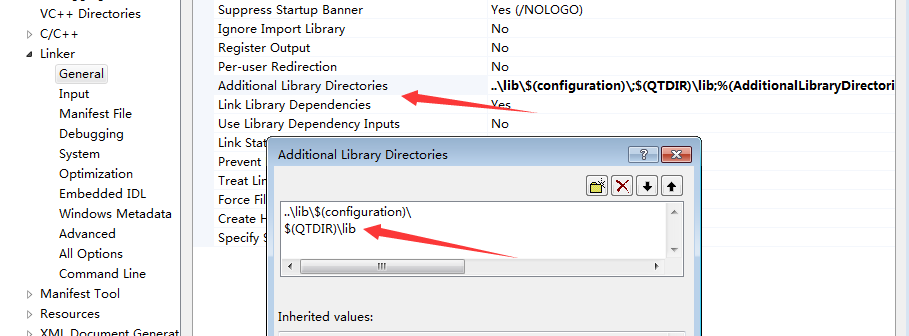
生成以下几个文件：.dll,.lib,.ilk,.exp,.pdb等几个文件，而我们需要用的就是.dll和.lib文件。

如果没用这些调用的函数或者类的\_\_declspec(dllexport) 声明，那么就不能生成lib文件。这个文件是工程隐式调用dll项目的一个静态链接库，代码都在dll中，调用完成，lib文件的使命就完成了。

调用dll分为显示调用和隐式调用，由于显示调用的LoadLibraryA（）加载一个dll之后，还要手动导出每一个接口函数，使用比较麻烦，所以工程之中使用的是隐式调用。

4：隐式调用

1. 先创建一个测试或者使用dll的工程，然后将.lib和.dll文件导入工程，导入的之前要先将这两个文件拷贝到.exe的目录之下，也可以通过project的setting中设置linker中的general 的外部链接库的地址：



1. 设置linker中的input中的外部依赖项

