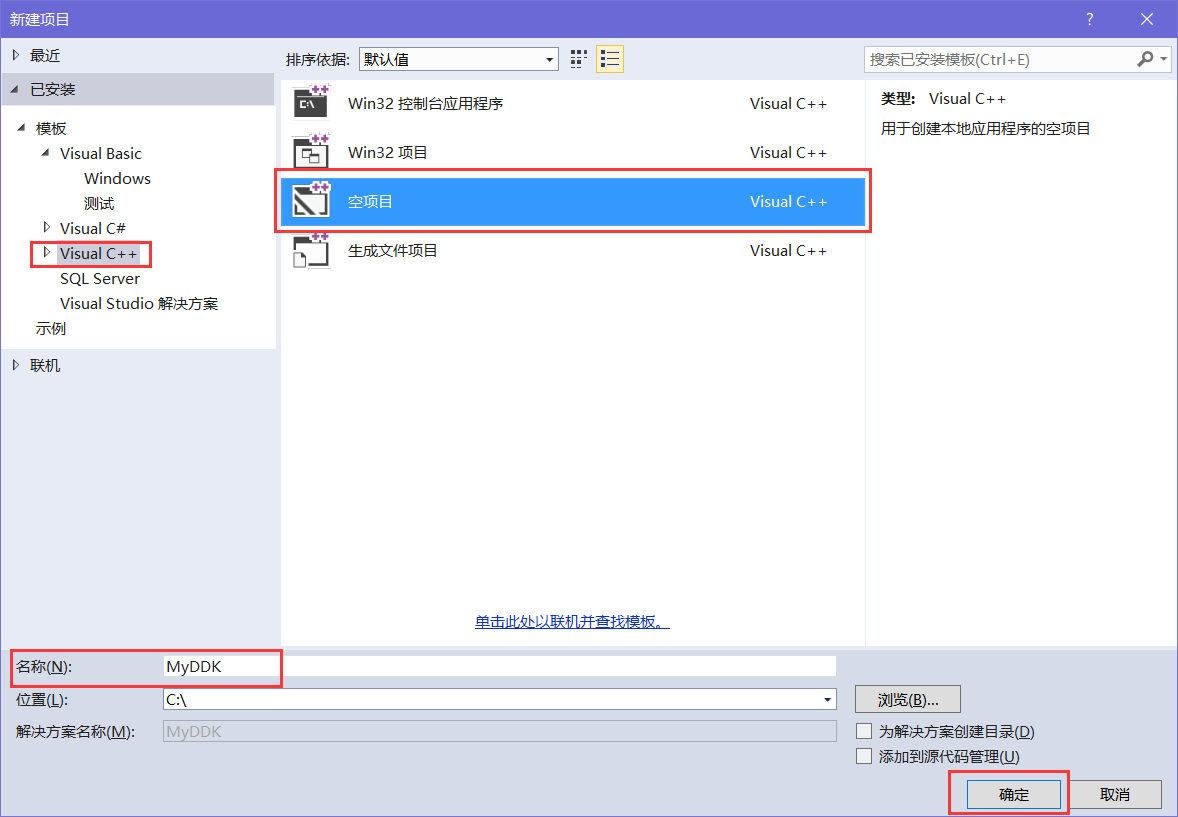
Windows Driver Kit 是一种完全集成的驱动程序开发工具包，它包含 WinDDK 用于测试 Windows 驱动器的可靠性和稳定性，本次实验使用的是 WDK8.1 驱动开发工具包，该工具包支持 Windows 7到Windows 10 系统的驱动开发。

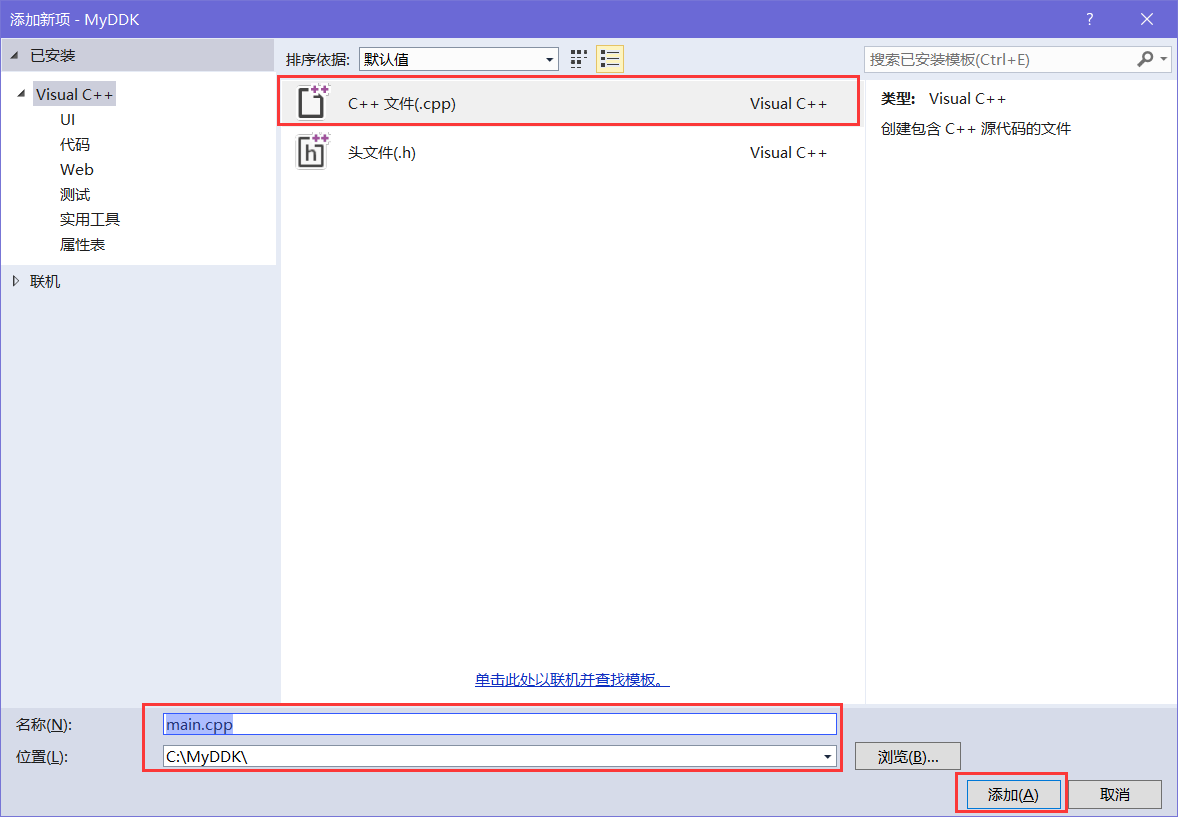
驱动WDK工具包推荐: Windows Driver Kit(WDK) v8.1 离线安装包配置好的案例下载: https://cdn.lyshark.com/code/WinDDK.zip

首先你需要先安装好 Visual Studio 2013 的开发环境，然后再安装 Windows Driver Kit 8.1 的驱动开发工具包，这个工具包安装好以后1.5G左右，不过我已经把这个安装包中的关键库文件提取出来了， 提取出的文件只有 80MB ，直接将提取好的 WinDDK.zip 解压缩到指定文件中，并配置环境即可使用，省去了安装WDK的麻烦。

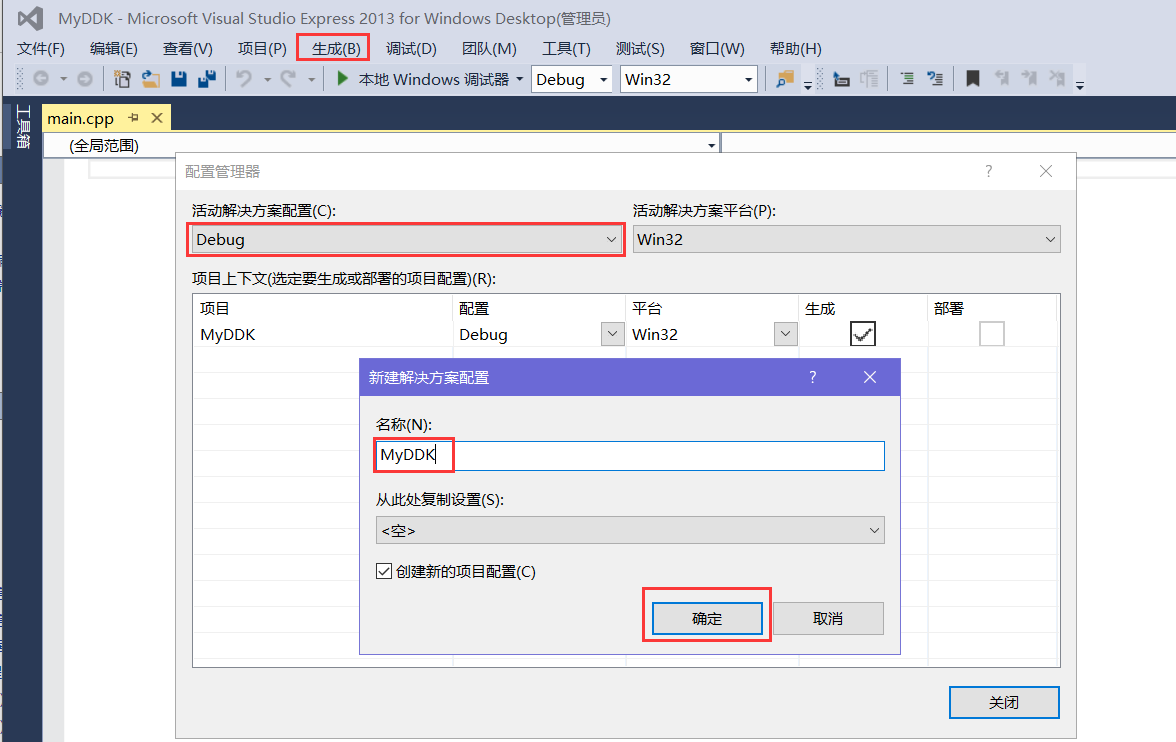
1.这里直接把我提取的 WinDDK.zip 文件解压缩到C盘根目录下，然后打开 Visual Studio 开发工具， 按下 Ctrl+Shift+N 新建空项目并输入项目名称为 MyDDK 即可。



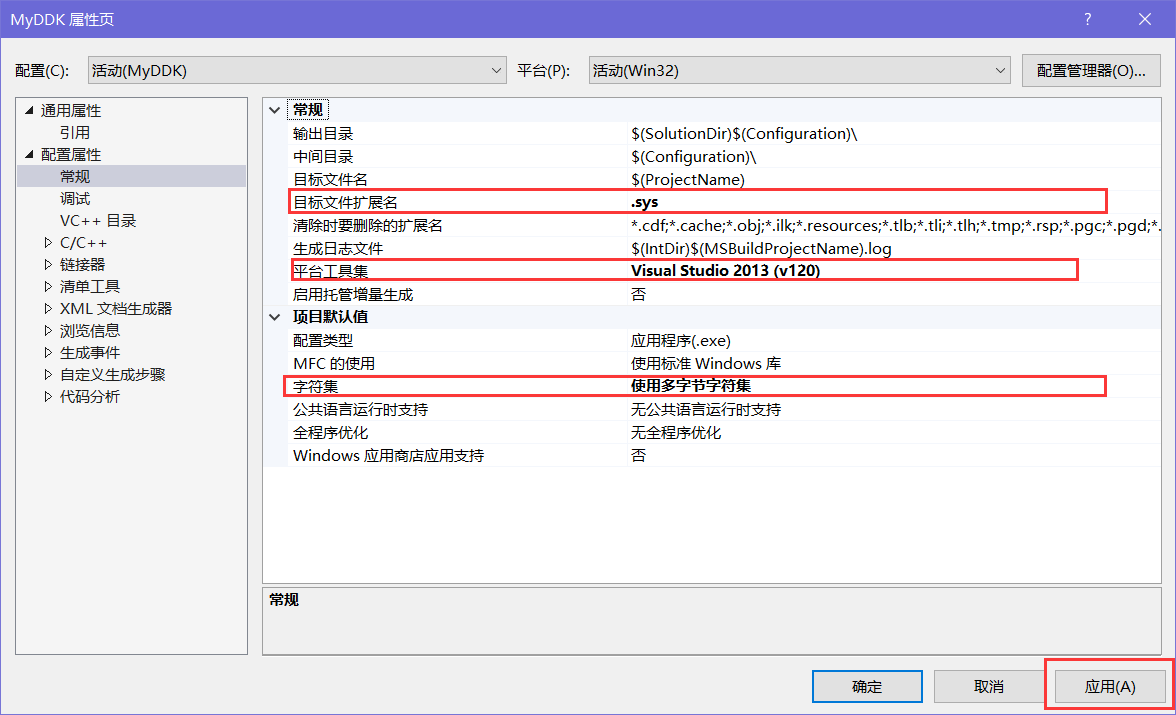
2.依次选择解决方案视图 -> 源文件 -> 添加新建项，或者直接按下 Ctrl + Shift + A 快捷打开菜单， 并创建 main.cpp 文件。



3.接着我们需要修改一下配置管理器，添加自定义配置管理，选择 生成 -> 配置管理器 -> 新建 ，此处我们命名为 MyDDK 即可。



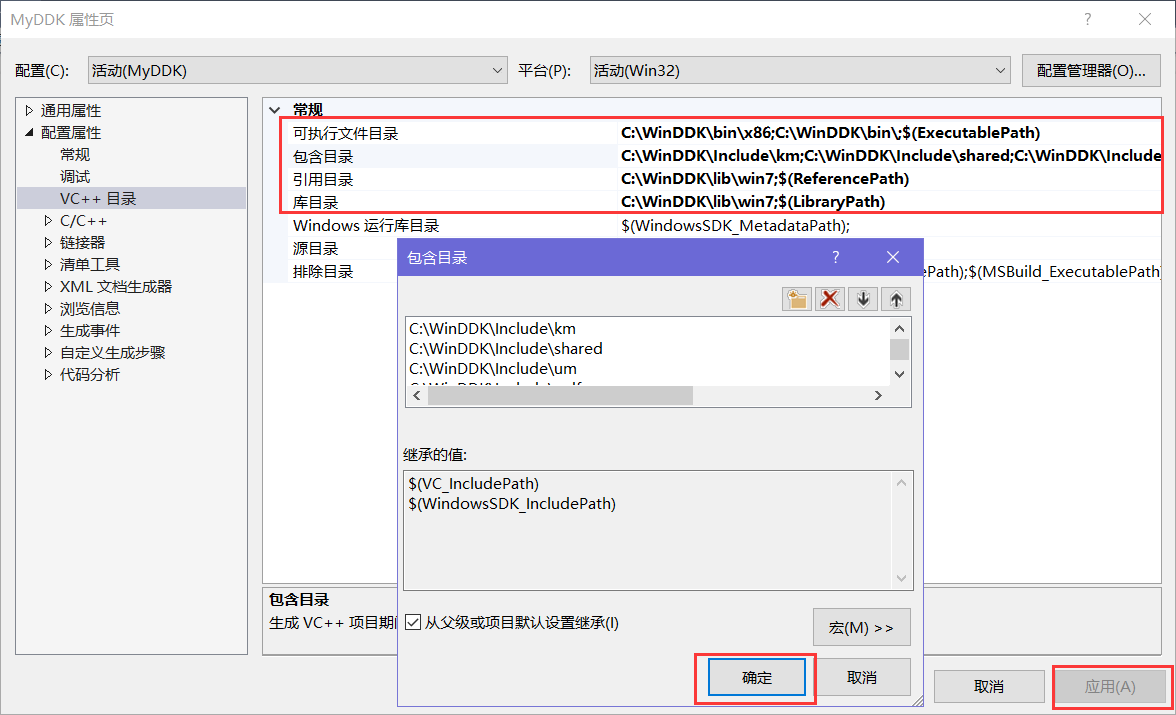
4.接着修改一下配置属性中的常规属性，点击菜单栏中的调试，选择 MyDDK属性 -> 配置 -> 常规 -> 修改以下几处。



5.配置可执行文件路径与导入库路径，这里我们选择 -> VC++目录 依次将如下信息填入配置

配置属性

项，如果需要编译x64位驱动只需要将x86改为x64即可，此处以x86为例。



可 执 行 文 件 ： C:\WinDDK\bin C:\WinDDK\bin\x86

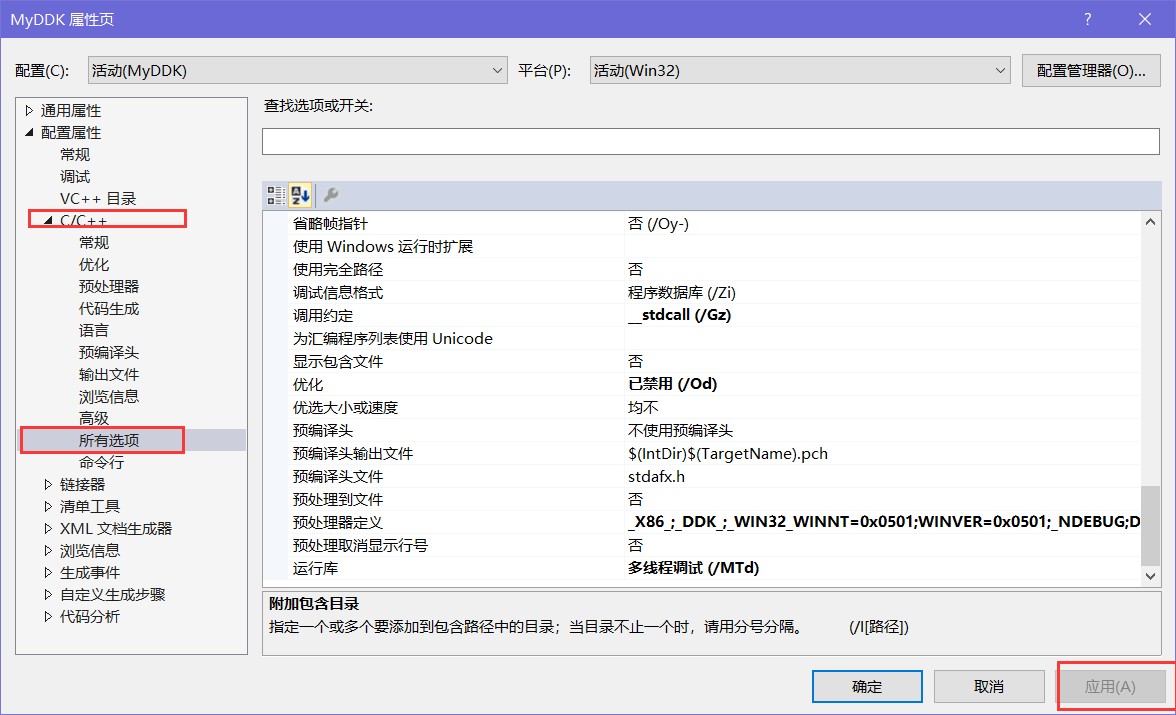
包含目录：

C:\WinDDK\Include\km C:\WinDDK\Include\um C:\WinDDK\Include\winrt C:\WinDDK\Include\shared

引用目录+库目录

C:\WinDDK\Lib\win7\km\x86

6.配置 C/C++优化选项 配置属性 -> C/C++ -> 所有选项 -> 依次修改下方几个关键处，如果你需要编译X64 驱动需将调用约定改为 fastcall (/Gr) 然后将预处理器定义中的 \_X86\_改为\_AMD64\_ 即可编译64位驱动了，此处以X86配置为例。



安全检查：禁用安全检查 (/GS-)

将警告视为错误：是 (/WX) 警告等级：等级 3 (/W3) 启用C++异常：否

调用约定： stdcall (/Gz)

优化：已禁用 (/Od)

运行库：多线程调试 (/MTd)

预处理器定义： \_X86\_;\_DDK\_;\_WIN32\_WINNT=0x0501;WINVER=0x0501;\_NDEBUG;DBG=0;%

(PreprocessorDefinitions)

7.接着需要 配置连接器 选项，选择连接器 -> 所有选项 -> 依次修改下方几个关键处。

附加选项：/IGNORE:4078 /safeseh:no

附加依赖项：ntoskrnl.lib;ndis.lib;Hal.lib;wdm.lib;wdmsec.lib;wmilib.lib 固定基址：此处需要为空

忽略所有默认库：是 (/NODEFAULTLIB) 启用增量链接：否 (/INCREMENTAL:NO) 驱动程序：驱动程序 (/Driver)

入口点：DriverEntry

生成清单：否 (/MANIFEST:NO)

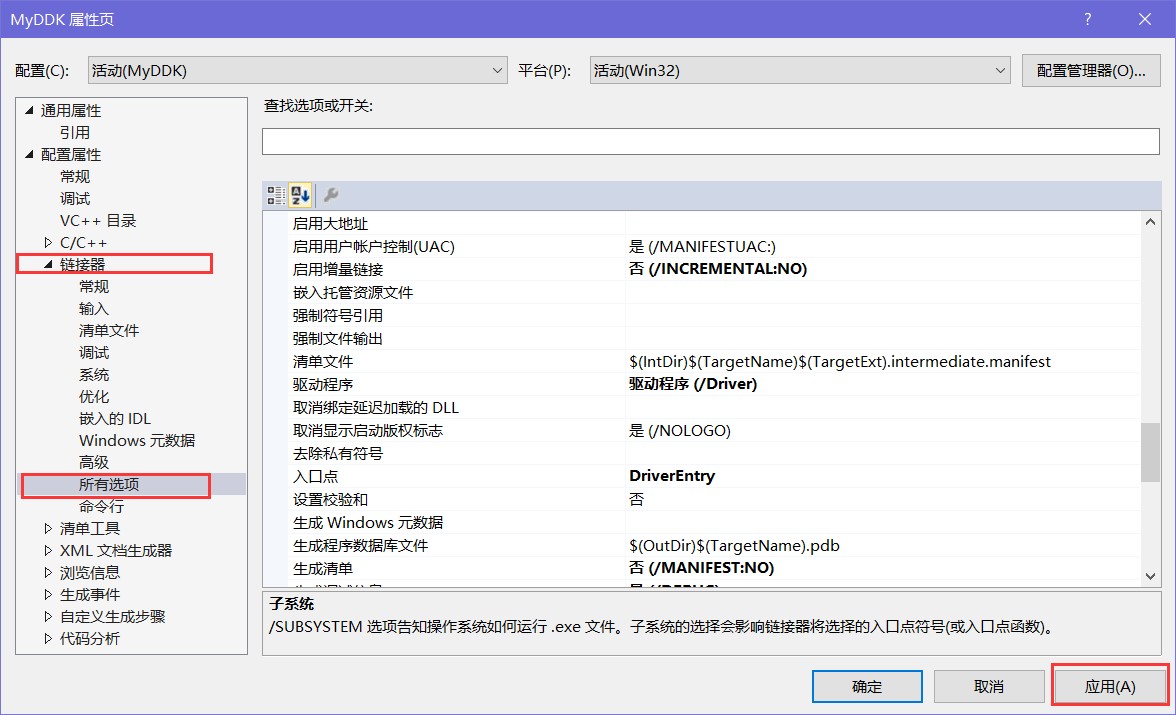
生成调试信息：是 (/DEBUG)

生成映射文件：是 (/MAP)

数据执行保护：是 (/NXCOMPAT)

随机基址：此处需要清空

子系统：本机 (/SUBSYSTEM:NATIVE)



8.上方的配置已经基本完成了，接着我们编写一段驱动初始化代码，然后按下 F7 即可完成驱动的编译。

#include <ntddk.h>

VOID UnDriver(PDRIVER\_OBJECT driver)

{

DbgPrint(("Uninstall Driver Is OK \n"));

}

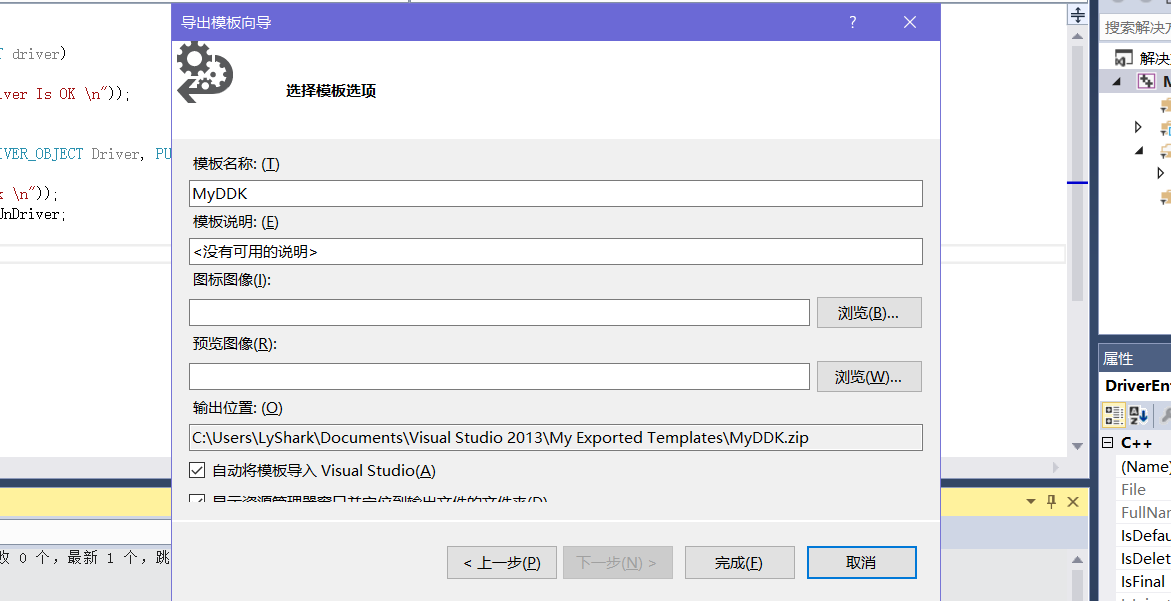
NTSTATUS DriverEntry(IN PDRIVER\_OBJECT Driver, PUNICODE\_STRING RegistryPath)

{

DbgPrint(("hello lyshark \n")); Driver->DriverUnload = UnDriver; return STATUS\_SUCCESS;

}

9.最后我们生成一个驱动开发模板，依次选择 文件 -> 导出模板 -> 项目模板 -> 下一步 -> 完成 即可完成模板的导出。



此时关闭VS工具，再次打开，就能直接使用我们的模板来开发驱动了，不需要每次都配置。

