# 关于

作者：张佩

2010/11/26

本文档介绍怎么使用本书的源码工程。首先请读者按照下面的两个步骤进行操作：

1. 设置名为BASEDIR 的环境变量，变量的值为用户本地WDK的目录，比如：e:\WinDDK\7600.16385.1。

如何设置环境变量呢？鼠标右键“我的电脑”，选“属性”，在“高级”标签中点击“环境变量”按钮即可。如下图所示：

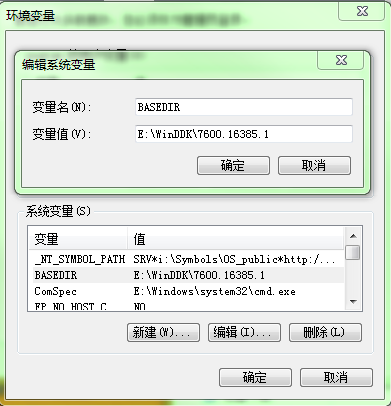


图1 环境变量

1. 将当前目录下的**my\_build.bat**和**my\_clean.ba**t两个批文件拷贝到系统目录（一般是：c:\windows\system32）下。这两个批文件，前者用来编译驱动项目，后者用来清除编译生成的临时文件、目录。

上面两步完成后，读者最好进行一次重启操作。因为新设置的环境变量只有重新启动才能生效。下面我们讲讲如何编译驱动工程。以目录06中的simClass项目为例讲解。simClass项目中有一个VC项目文件，我们的目地是要能够在Visual Studio集成编译环境中对驱动项目进行编辑和编译工作。

我们知道在一般情况下，通过VS软件是不能够生成一个内核驱动工程的，自然也无法直接在VS集成环境中编译项目。但通过上面的1、2两个步骤，我们却能够实现simClass在VS中的直接编译。如下图所示：

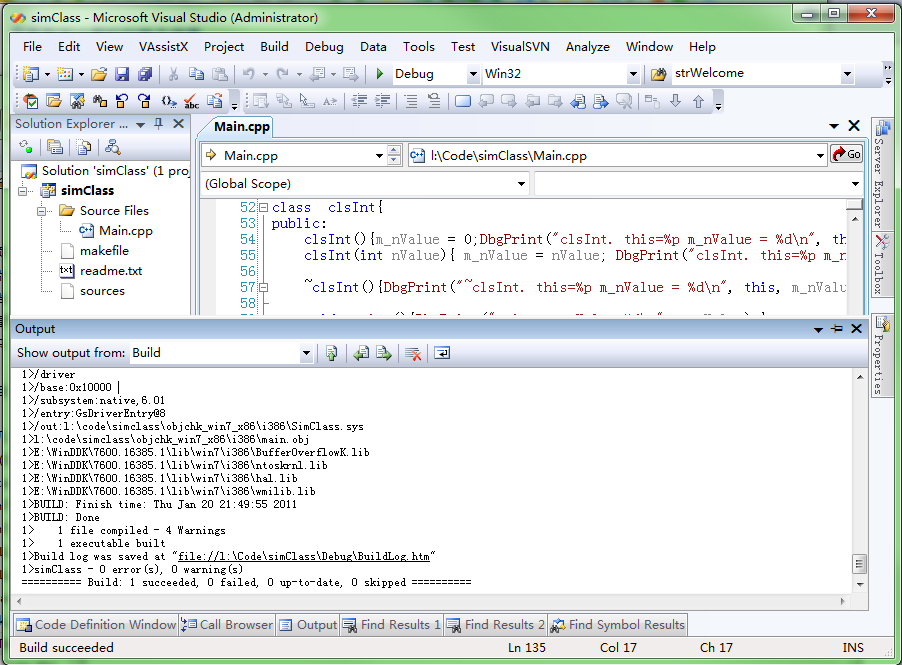


图2 VS中编译驱动工程

仔细看上图下方的Output输出信息，能看到下面的信息：

**1>/out:l:\code\simclass\objchk\_win7\_x86\i386\SimClass.sys**

1>l:\code\simclass\objchk\_win7\_x86\i386\main.obj

1>E:\WinDDK\7600.16385.1\lib\win7\i386\BufferOverflowK.lib

1>E:\WinDDK\7600.16385.1\lib\win7\i386\ntoskrnl.lib

1>E:\WinDDK\7600.16385.1\lib\win7\i386\hal.lib

1>E:\WinDDK\7600.16385.1\lib\win7\i386\wmilib.lib

1>BUILD: Finish time: Thu Jan 20 21:49:55 2011

1>BUILD: Done

1> 1 file compiled - 4 Warnings

1> 1 executable built

1>Build log was saved at "file://l:\Code\simClass\Debug\BuildLog.htm"

1>simClass - 0 error(s), 0 warning(s)

========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========

清单1. 编译输出信息

这说明驱动编译成功，并生成了SimiClass.sys文件。

下面我们来看看它的实现原理，**有兴趣的读者可以看一看**。右键工程名打开属性，可看到如下图所示的界面：

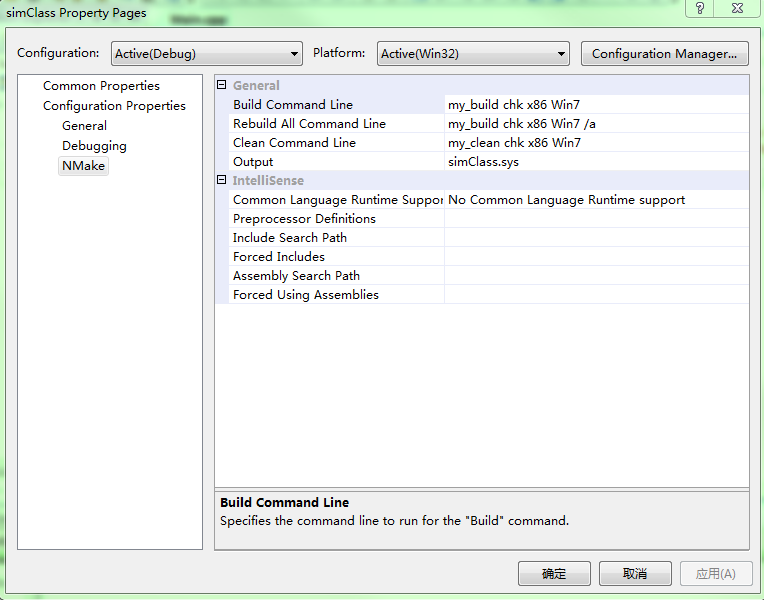


图3 驱动工程属性

印象深刻的读者，把上图的属性界面，与诸如MFC、Win32控制台等项目的属性界面相比较，就能发现很大的不同：

首先，它有一个名为“NMake”的属性项。这是VS软件自身预定义的一个属性项，一般情况下不被显示。仔细看NMake这个名字，大家对NMake.exe或许比较熟悉，它是Windows平台下，以命令行方式手动编译工程项目的工具。当然在IDE完善发展的现在，NMake.exe已经早不为大家熟悉了。而对于驱动项目，我们只能返璞归真，回到元初的NMake工具。

其次再看NMake属性下的设置，摘录于下：

Build Command Line: my\_build chk x86 Win7

Clean Command Line: my\_clean chk x86 Win7

清单2 NMake命令

担心大家会吃到夹生饭，所以把清单2中的两行翻译于下：

1. 当用户点击Build按钮时，VS软件将自动执行命令“my\_build chk x86 Win7”；
2. 当用户点击Clean按钮时，VS软件将自动执行命令“my\_clean chk x86 Win7”

上面两条里的“自动”，是VS软件自身能够保证的，因为前文也说过了，NMake属性项是软件预定义的。

只要执行了“my\_build chk x86 Win7”这条命令，就能够得到类似图2所示的编译结果。读者可能会问：my\_build是什么？它做了什么动作？

命令中的my\_build就是前面读者拷贝到系统目录下的my\_build.bat批文件。看看到底这个批文件是怎么做的，下面是它的主要内容：

call %BASEDIR%\bin\setenv.bat %BASEDIR% %1 %2 %3

build

清单3 my\_build批文件

简简单单的两行，不明就里的读者或者会糊涂，不明白它在干什么。其实很简单，这个批文件在完成两个工作：设置WDK编译环境（Line1），并在设置好的环境中编译工程（Line2）。

详细讲第一项。它调用了WDK包中的setenv.bat批文件。这个批文件是做什么用的呢？读者可以直接到WDK中把它打开看看，好大一个批文件。作用就是设置编译环境。我们现在来做一个实验：

在本机的开始菜单中，在Windows Driver Kit项中，找到Win7 X86 CHK编译环境并右键点击其属性，弹出对话框后点击“快捷方式”页，在名为“目标”的编辑框中，看到下面的内容（为清晰故，我把参数进行了分行）：

1. c:\Windows\System32\cmd.exe // 运行控制台程序
2. /k // 这个参数表示执行后面的命令
3. c:\WinDDK\7600.16385.1\bin\setenv.bat // 环境脚本
4. c:\WinDDK\7600.16385.1\ // WinDDK目录
5. chk x86 WIN7 // 这是一组编译选项

清单4 控制台编译环境配置

很有意思的一组参数，明眼人立刻就可以看到，清单4中从第三行开始的参数内容，与清单3中的第一项内容，是完全一样的。它们都是在完成同样一件事情：配置编译环境。

环境编译好后，编译就很简单了。所以清单3的第二项，是一个简单的**build**指令。

上面的内容虽然简单，笔者颇花了些笔墨讲述，也许显得有些多此一举。但根据我的个人经验，还是有很多读者尚不了解这些知识的。关于如何将VS和驱动编程相结合这一问题，一直缠扰着部分初学者。希望对他们能有所帮助。

讲明了原理之后，最后我们说一下如何创建一个类似的驱动工程。因为在VS中创建一个项目，都是通过工程向导（AppWizard）来完成的，不会手动做。而软件自带的向导，并不包含对驱动工程的支持。所以就有几种可行解决方法，分别介绍如下：

1. 插件形式。

以插件形式存在的，较有名的是以前的DriverStudio，但这个软件早已经关张，不好再谈。最近出现的VisualDDK，是非常好的一个DDK工程插件，安装使用也非常简单。读者可以很容易从网上搜到它的免费安装包。安装好之后，直接从工程向导中就能找到它，如图5所示。

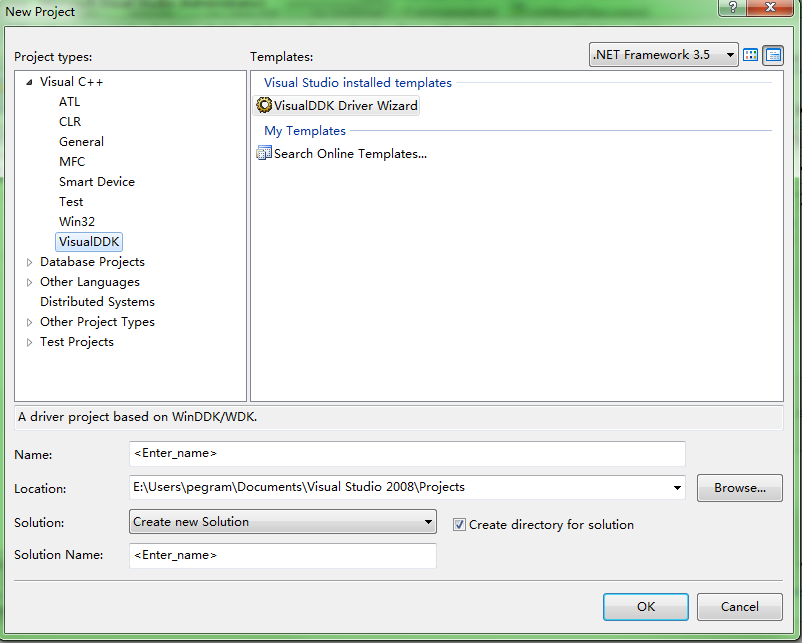


图5 VisualDDK的安装向导

1. 工具生成形式。

其原理是以一些简单的模本文件（包括VS工程文件，Sources文件等等），根据用户输入的设置（比如项目名称等），进行关键词替代，最后生成基本的工程文件。比较好的例子是easySYS这个工具。但笔者见到的EasySYS版本较老，生成的VS工程还是VC6.0的老版本。它的使用界面如下图所示：

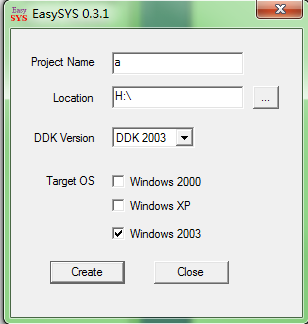


图6 使用EasySYS生成驱动工程

1. 手动法

笔者时而使用这种手动法，其实也挺方便。读者只要把simClass中的simClass.vcproj文件拷贝到某个空目录中，改变其名称为所期望的名称，打开工程，把里面所有的文件都删除掉再添加新的驱动文件，这样，一个新工程也就算可以了。