

VC++ UDF Studio 2021R1 教程

[打开 English Version](#)

支持的版本:

Windows, Fluent, Visual Studio 版本	是否支持
Win XP~Win10 (x86/x64)	✓
Fluent 6.3~2021R1(x86/x64)	✓
Visual Studio 2008 SP1~2013 专业或旗舰版	✓
Visual Studio 2015 或更高	✗

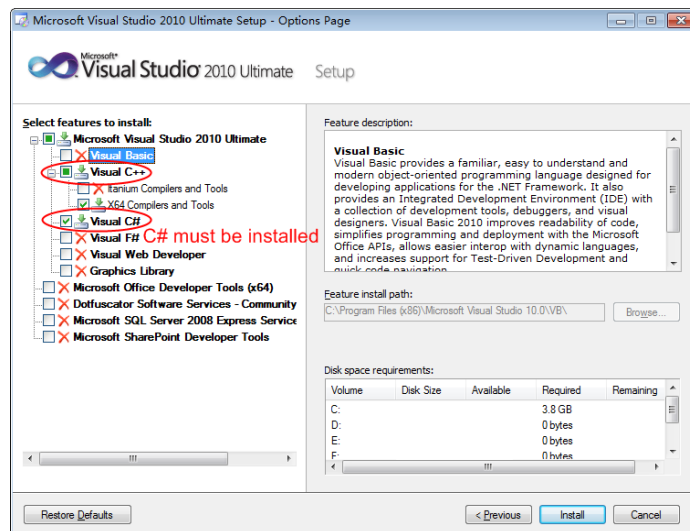
*推荐使用 Win10 + Visual Studio 2010 旗舰版+ Fluent 17.0 (或更高)

学术版与企业版对比:

功能	学术版		企业版	
	试用版	注册版	试用版	注册版
编译调试 (串行, 单精度)	✓ 最多 2 个宏	✓ 宏数不限	✓ 最多 2 个宏	✓ 宏数不限
编译调试 (串行, 双精度)	✗	✓	✗	✓
编译调试 (并行, 单/双精度)	✗	✓	✗	✓
调用 C++/Win32 API/MFC 函数	✓	✓	✓	✓
根据边界名字获取 id 号	✓	✓	✓	✓
调用第三方 LIB 库	✓	✓	✓	✓
设置第三方函数库目录	✗	✗	✓ 最多 1 个目录	✓ 目录数不限
Fluent 中加入用户菜单	✗	✗	✓ 最多 2 个子菜单	✓ 子菜单数目不限
UDF 中驱动 fluent 迭代	✗	✗	✗	✓
UDF 中调用 Scheme/TUI 命令	✗	✗	✗	✓
与 Matlab 耦合迭代计算	✗	✗	开发中	开发中

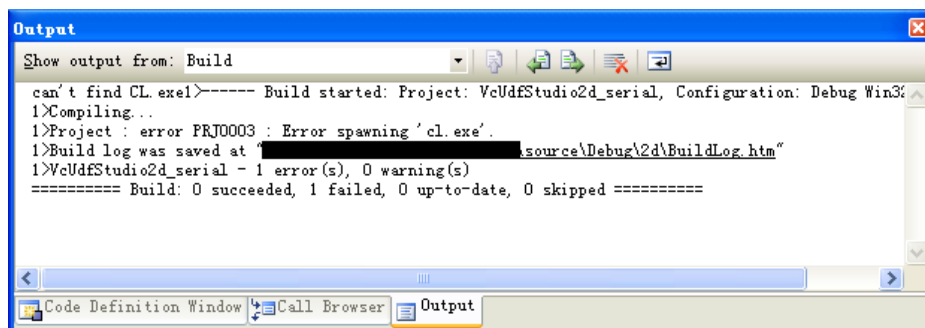
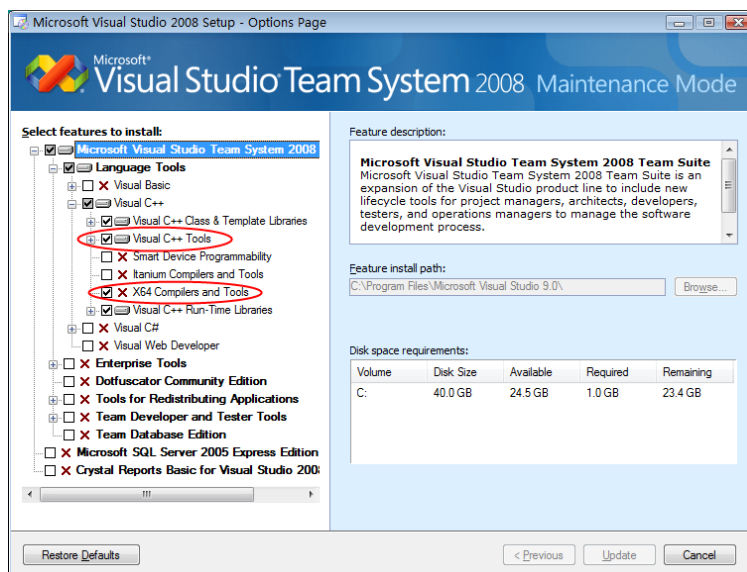
Visual Studio 安装注意事项:

1. 建议 Visual C#与 Visual C++同时安装。尽管对于某些 Visual Studio 版本这并不是必须的 (例如 VS2008 SP1 旗舰版)。但对于 VS2010 版本, Visual C#是必须安装的, 否则软件启动 Visual Studio 的时候会出错。



对于 VS2008, 安装时请勾选“Visual C++ 工具”(Visual C++ Tools)。此外, 对于 VS2008 标准版, Visual C# 是必须安装的, 否则会出现 “can’t find cl.exe” 的错误。安装完后还需在此基础上加装 Service Pack 1。

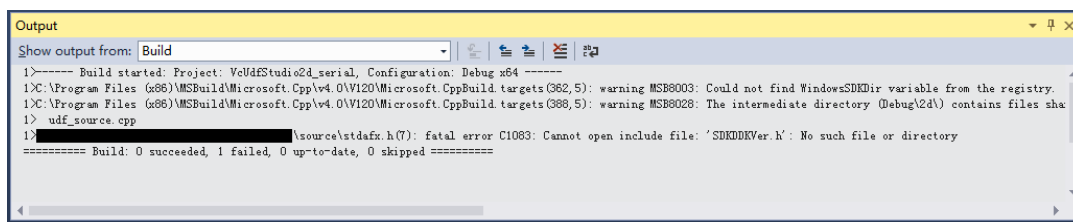
对于 64 位 Windows, 您只能使用 64 位 Fluent 并保证勾选安装 “X64 编译器和工具” (X64 compilers and Tools)。



2. 如果想使用 Visual Studio 2013, 请安装最新的 **Visual Studio 2013 update5**。对于其它更早期的 2013 版本, 可能出现无法找到 “afxv_cpu.h” 文件, 甚至插件菜单混乱的情况。

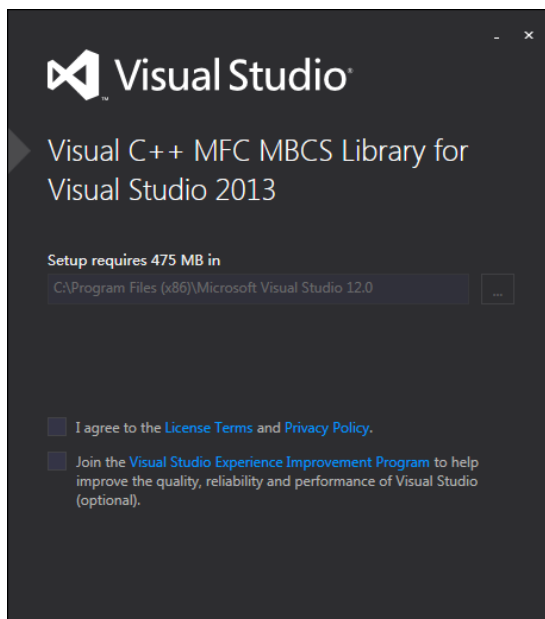
其次, 安装前保证网络畅通。否则, 可能会报告 “WindowSDKDir” 变量找不到的错误

(如下图所示)。

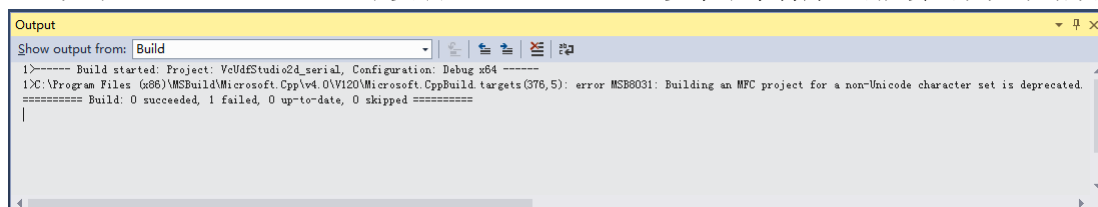


最后，Visual Studio 2013 还需下载安装 Visual C++ MFC 多字节字符库（Visual C++ MFC Multi-Byte-Character-Set Library）。网址是：

<https://www.microsoft.com/en-US/download/details.aspx?id=40770>

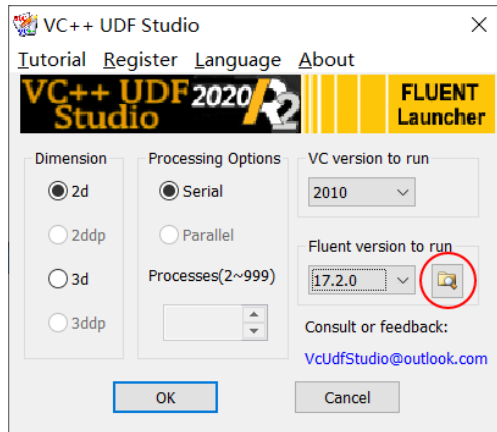


如果 Visual Studio 2013 未安装 Visual C++ MFC 多字节字符库，可能会出现如下错误。

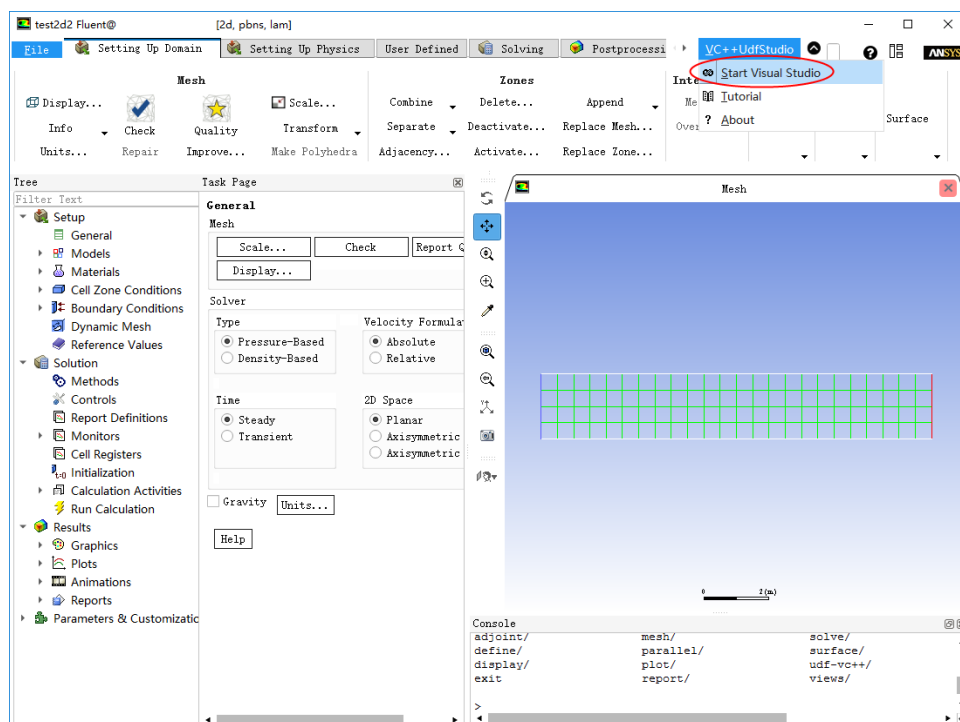


使用步骤实例：

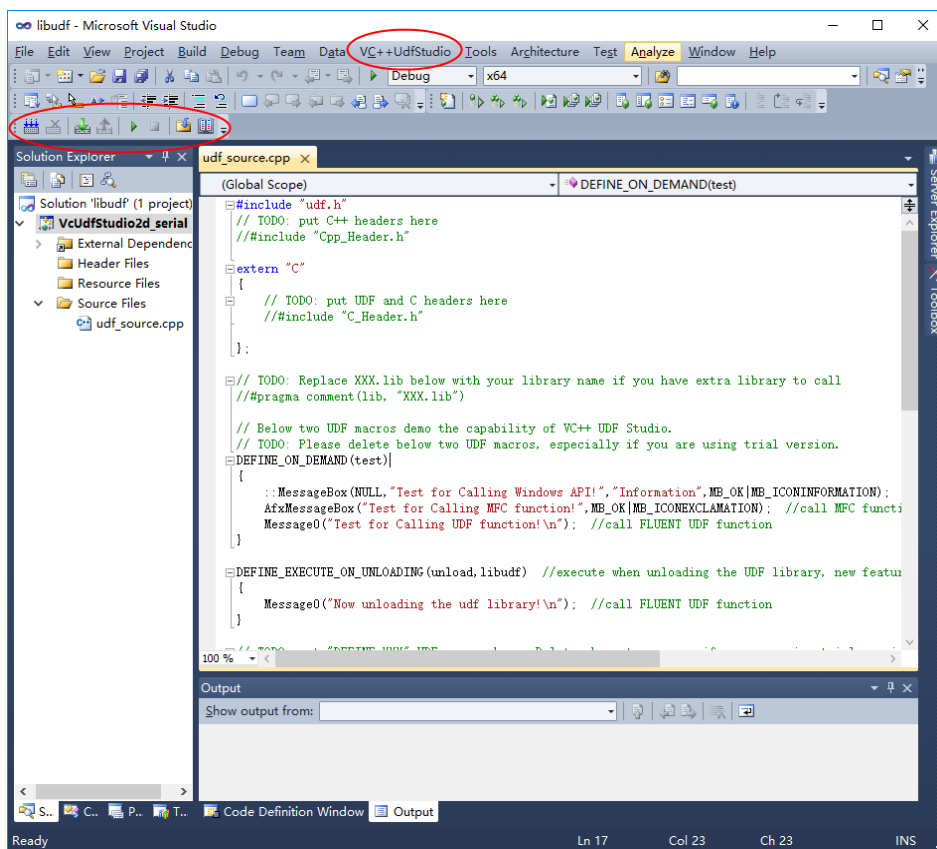
1. 安装 Visual Studio 时确保 Visual C#，Visual C++ 及 “Visual C++ Tools” 已勾选（具体注意事项见前一节 “Visual Studio 安装注意事项” 内容）。
2. 如果您使用 Visual Studio 2008，请在使用本软件前安装 Service Pack 1，其它版本可以不用安装 Service Pack。
3. 运行“VC++ UDF Studio”加载器并选择欲启动的 FLUENT 和 VC 版本，然后点击“OK”按钮。如果你想运行的 Fluent 版本不在列表里，你可以点击“Browse”按钮来选择 Fluent 版本对应的安装目录。



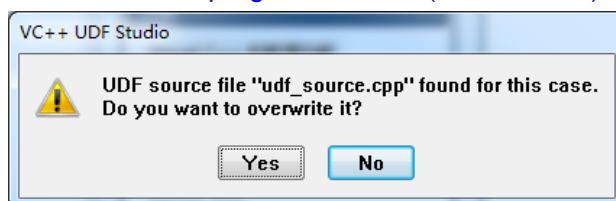
4. 读入一个 Fluent case 然后点击 “Start Visual Studio” 菜单开始 UDF 编程。

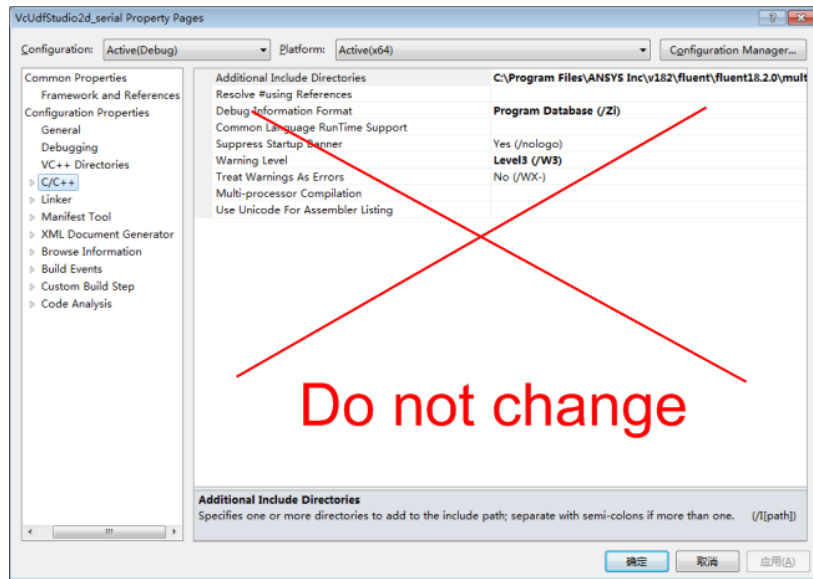


此时，Visual Studio 中能看到“VC++ UDF Studio”的工具栏和菜单。同时，软件会自动在 case 的相同目录建立一个名为“source”的文件夹，包含了所有 UDF 源代码。



如果“source”目录中已经存在“udf_source.cpp”文件，则会弹出如下的警告框，询问是否覆盖原文件。Visual Studio 关闭时，项目文件“*.vcproj”（VS2008）或“*.vcxproj”将被自动删除，因此请勿更改项目的设置。如果您想链接额外的库文件“XXX.lib”，您可以在“udf_source.cpp”中增加一句 `#pragma comment(lib, "XXX.lib")`

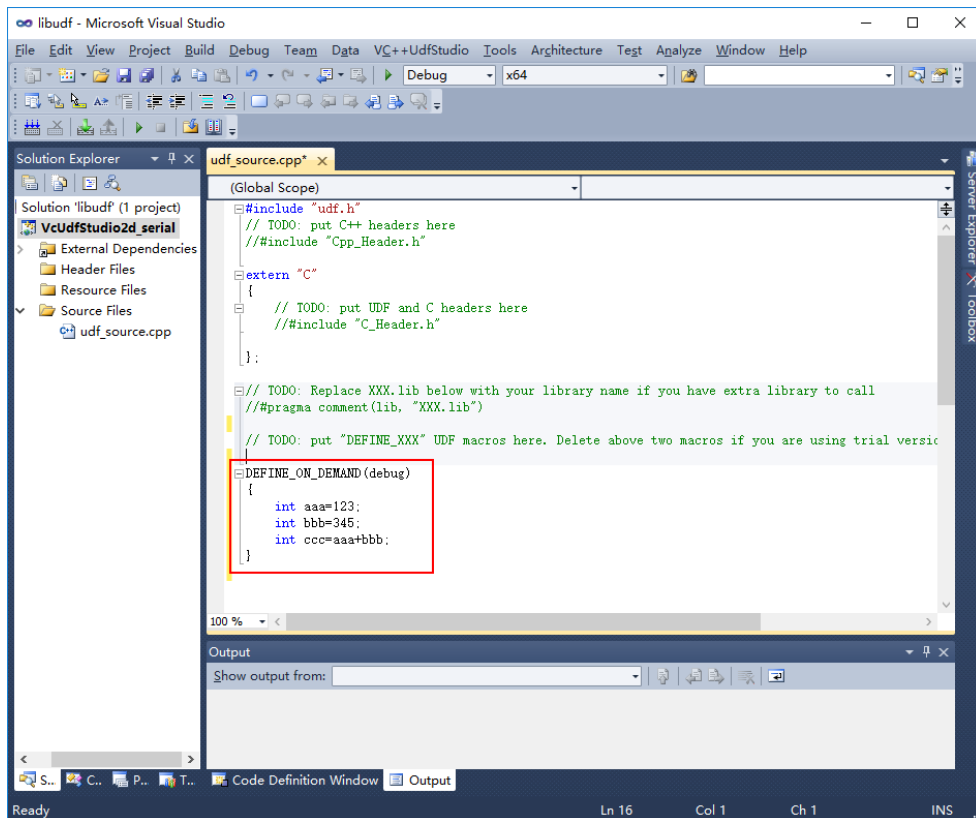




注意： Visual Studio 关闭时，除了“udf_source.cpp”和“udf_source.cpp.bak”以外，所有项目文件和临时文件夹（例如*.sln，*.suo，*.vcproj，*.vcxproj，*.user，*.filters，*.ncb，*.sdf，Debug 文件夹，Release 文件夹）将被自动删除。

5. 编辑 UDF 源码，在“udf_source.cpp”的末尾添加如下测试代码（如果是试用版，请删除自带的 DEFINE_ON_DEMAND 和 DEFINE_EXECUTE_ON_UNLOADING 两个宏，否则会报告宏总数超过允许数的错误）。

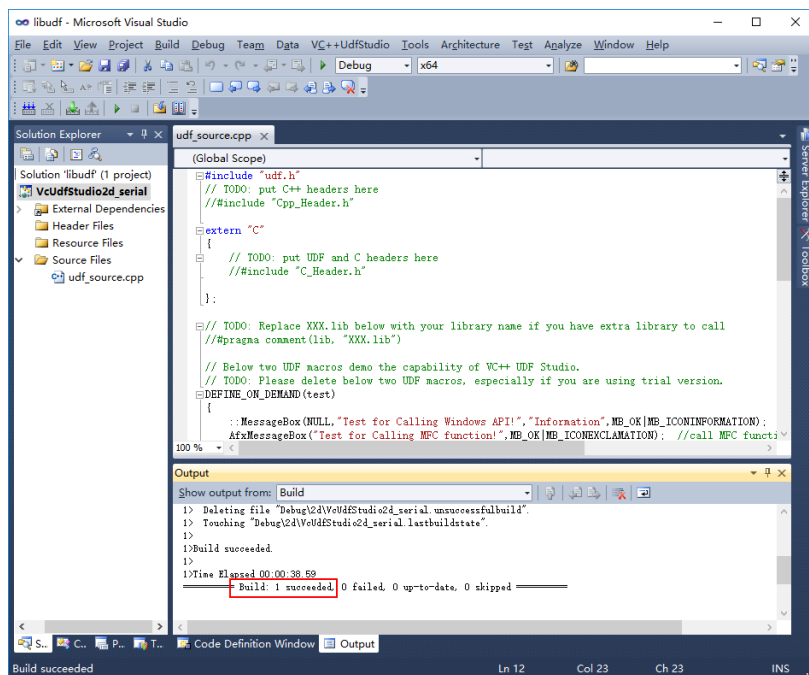
```
DEFINE_ON_DEMAND(debug)
{
    int aaa=123;
    int bbb=345;
    int ccc=aaa+bbb;
}
```



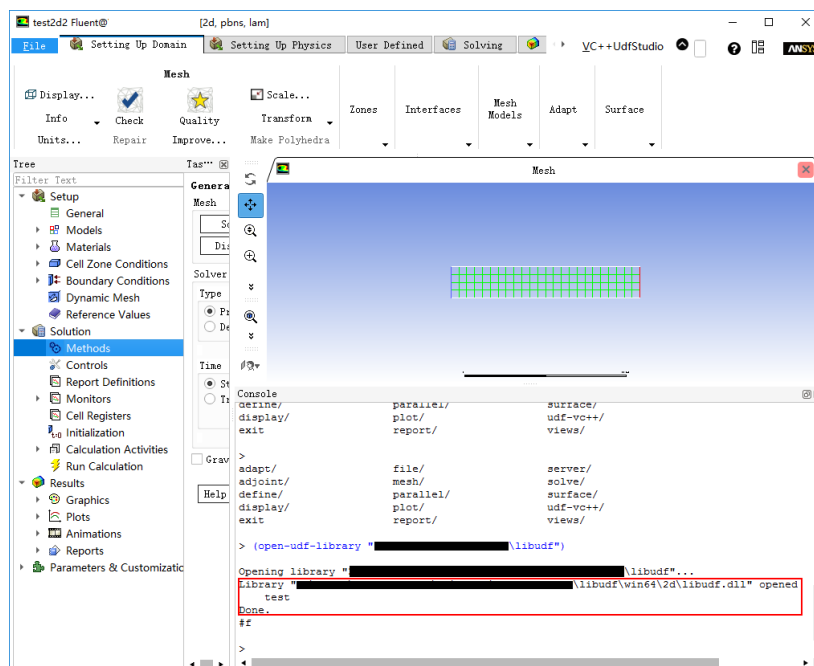
6. 点击“Build UDF library”按钮（或热键“F7”）开始编译 UDF 源码。



如果没有发现语法错误，则会报告编译成功，如下图所示。只有编译通过后，你才能进行后续的加载或调试。



7. 点击“Load UDF library to Fluent”按钮将编译好的库载入 Fluent。此时，Fluent 控制台界面应该有显示 libudf 库成功加载的响应。



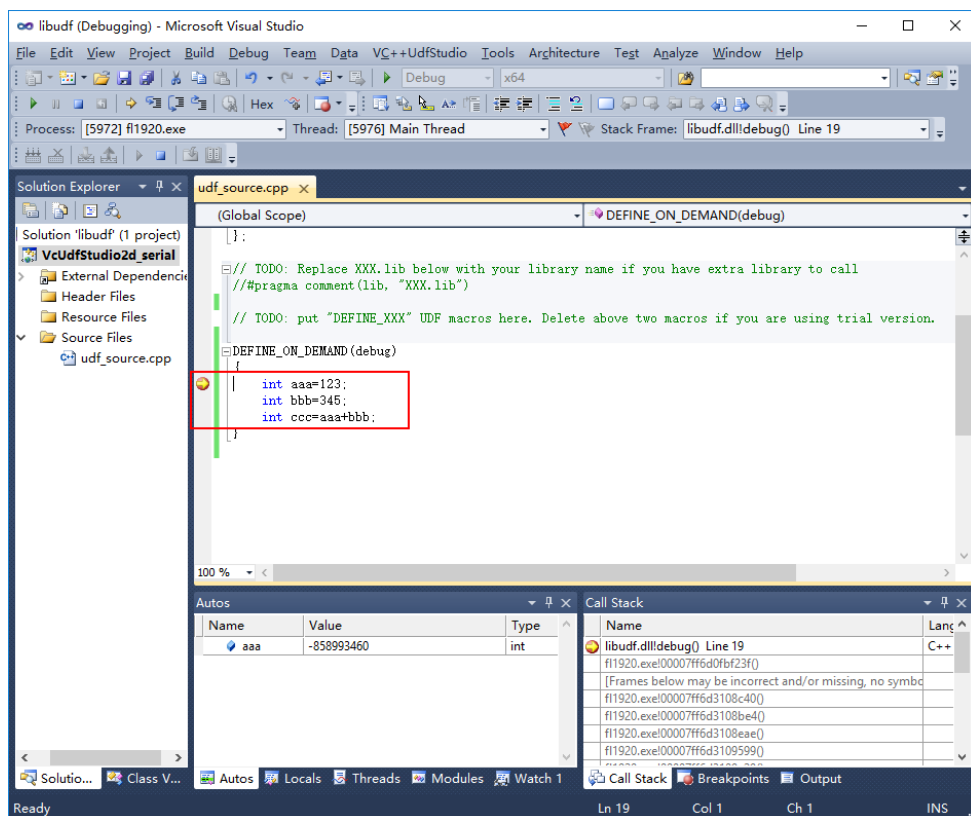
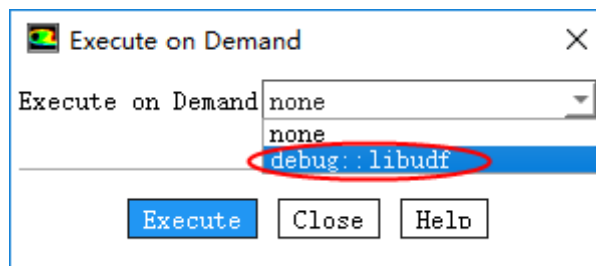
8. 在“int aaa=123;”语句前设置断点（鼠标停在该行热键“F9”），然后点击“Start debugging

UDF library”按钮。当然如果不加载直接点该按钮也是可以的，软件会自动先帮您加载，但必须编译通过。

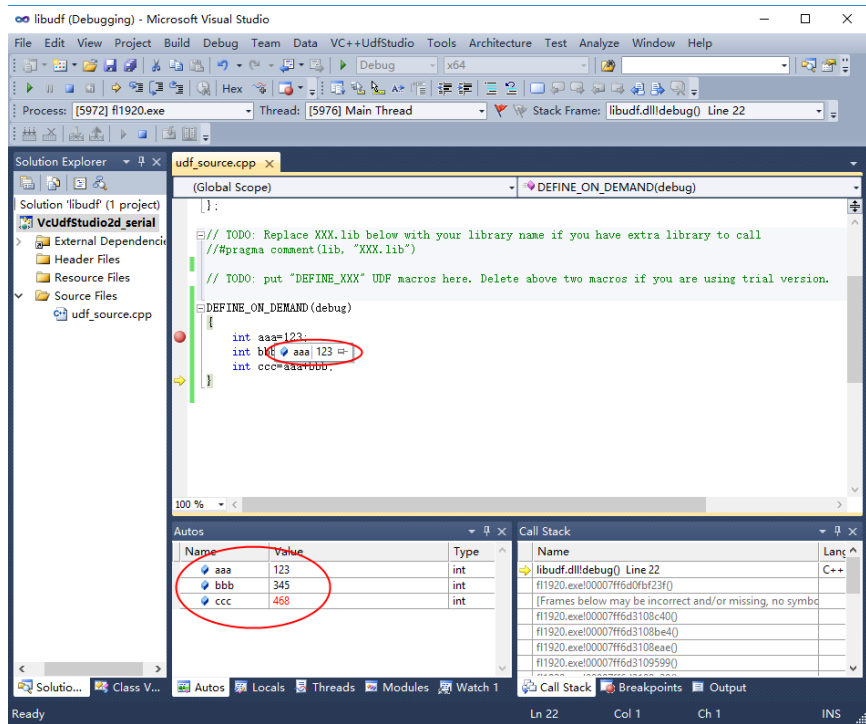


9. 在 Fluent 中执行 “debug::libudf” 函数，Visual Studio 会自动停在刚才设置的断点处，您可以看到所有变量的值了。注意：有些宏需要在 Fluent 迭代计算过程中才能调试，例如 DEFINE_SOURCE, DEFINE_PROFILE 等等。

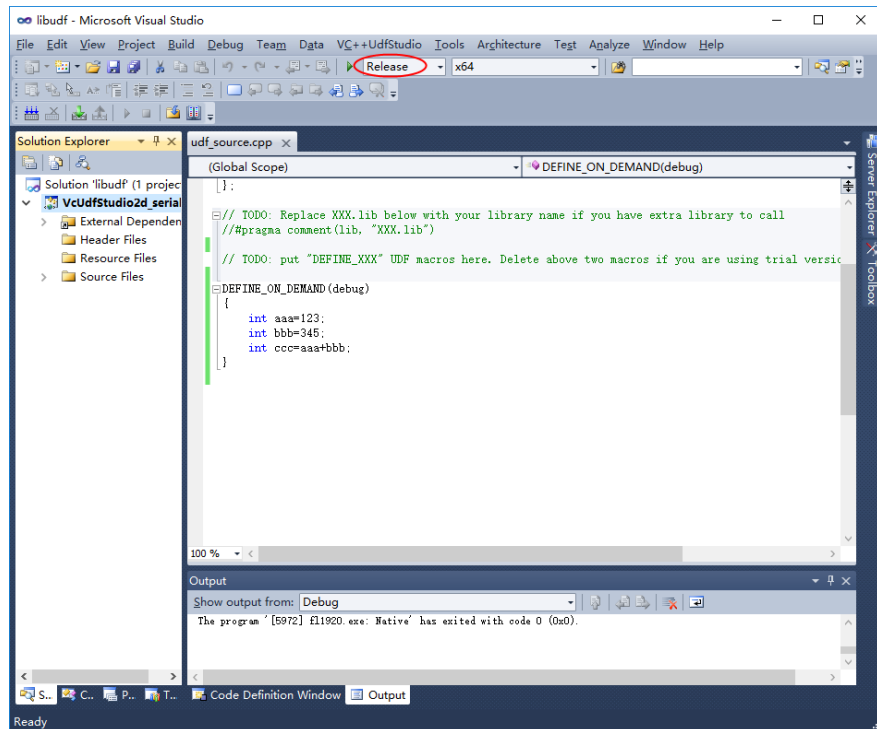
注意：请首先理解您要调试的宏何时被 Fluent 调用。即使在某个宏中下了断点，但 Fluent 未调用该宏，断点也是不起作用的。例如 DEFINE_SOURCE 一般在 FLUENT 迭代计算过程中被调用，DEFINE_INIT 是在初始化时被调用。如果在 DEFINE_SOURCE 宏中下了断点，但没有开始迭代，程序是不会停在断点处的。



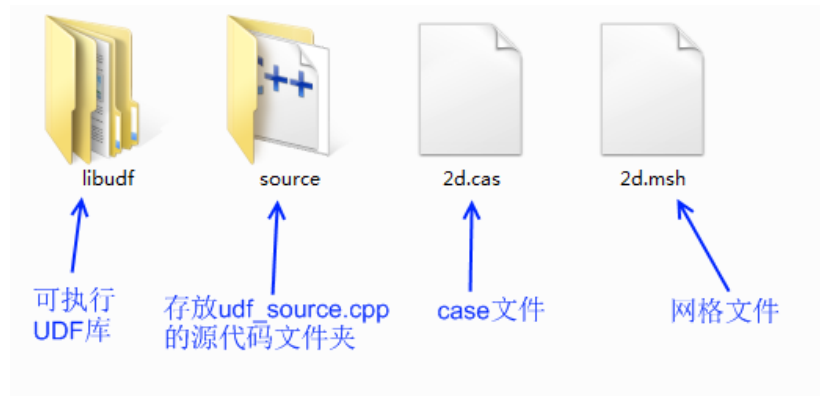
10. 单步调试跟踪（或“F10”热键），您可以采用平时调试 C++ 程序的方式那样观察所有变量的值。



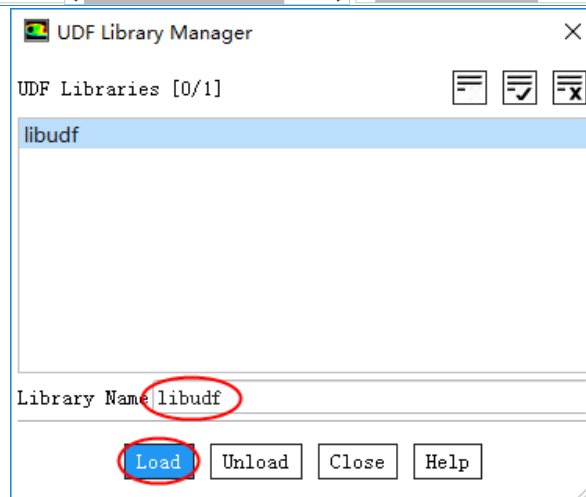
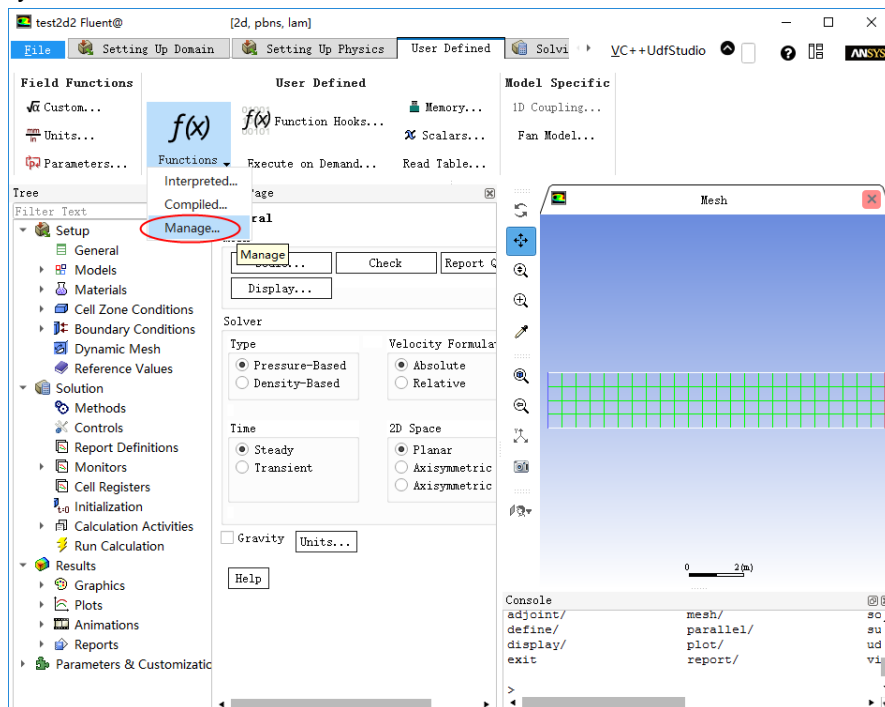
11. 所有 bug 修正后，您可以将 Debug 模式改为 Release 模式，然后在 Release 下重新编译一次以便发布您的 UDF 库。



12. 现在，您的 case 目录一般如下图所示。其中，“libudf”文件夹包含了您要发布的 UDF 库，而“source”文件夹保存了您的 UDF 源代码文件“udef_source.cpp”。

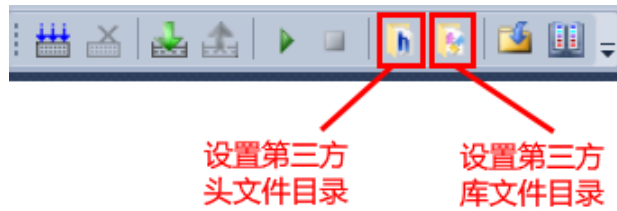


13. 重新编译通过 Release 版本的 UDF 库后，您如果想只计算 case（不改动或编译 UDF 源码）那么可以不必从 VC++ UDF Studio 加载器启动 Fluent。只要按照通常方式启动 Fluent，加载 UDF 库可以用“Define->User-Defined->Functions->Manage”菜单。在“Library Name”编辑框中输入库名“libudf”，然后点击“Load”按钮即可。

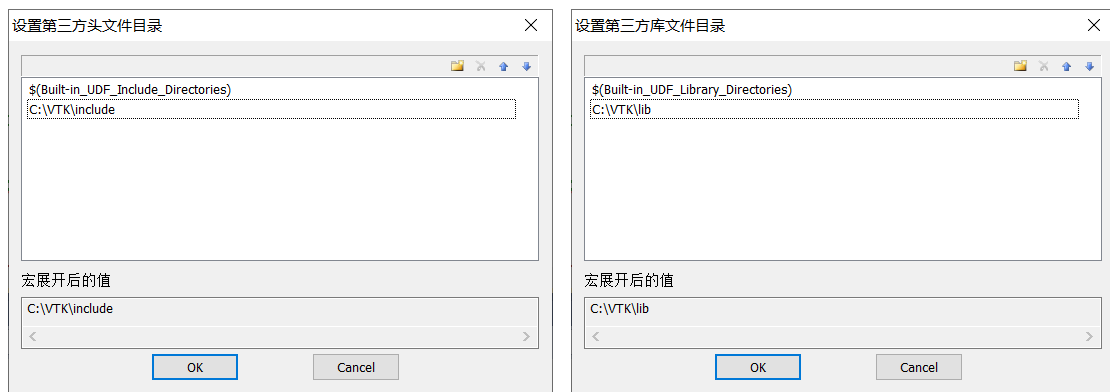


设置第三方库目录（仅企业版）：

企业版用户可以用如下两个按钮设置第三方头文件目录和库文件目录。



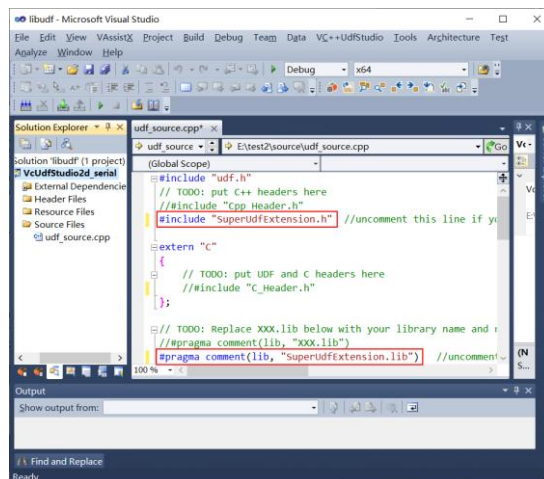
点击按钮后，分别出现如下对话框，可以通过手动或浏览加入第三方头文件和库文件的目录。其中\$(Build-in_UDF_Include_Directories)和\$(Build-in_UDF_Library_Directories)代表 VC++UdfStudio 所自带需要的头文件和库文件的目录，不允许修改，但可以调整它们和额外第三方库的前后顺序。



开启拓展函数库：

本工具已将一些常用功能以第三方库的形式封装好供用户直接调用，从而大大拓展 UDF 的功能。如下图所示，如需调用拓展的第三方函数，用户只需要将如下两句语句去掉注释。

```
#include "SuperUdfExtension.h"  
#pragma comment(lib, "SuperUdfExtension.lib")
```



拓展函数详解：

1. void SuperUdf_Initialize(HMODULE hLibudfDllModule)

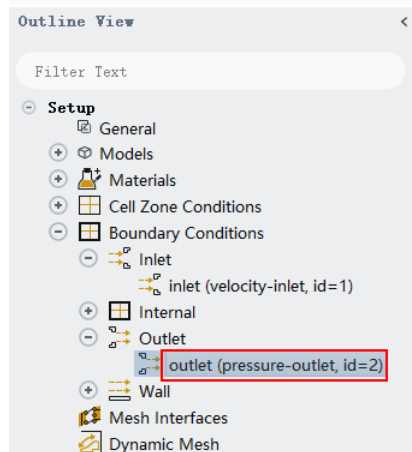
此函数负责拓展库的初始化，hLibudfDllModule 为 udf 库的模块句柄，可以使用 AfxGetInstanceHandle() 获取（参见下一节实例）。

注意：此函数必须在调用其它拓展函数前调用。较佳的调用该函数的地方是 DEFINE_EXECUTE_ON_LOADING 宏中。

2. int SuperUdf_GetZoneIdByName(char* strZoneName)

根据 zone 或边界的名字来获取对应的 zone ID，如下图所示的 case 的 UDF 源码中调用 SuperUdf_GetZoneIdByName("outlet") 函数将返回 2。

该函数主要用来解决源码通用性问题，由于 UDF 中经常会调用 LookUp_Thread(domain, zone_ID) 来获取 Thread，里面的 zone_ID 是关键参数，但是会随着网格变化而变化，很多用户每次更换网格后只能手动查到对应 zone ID 后修改源码再重新编译来实现，十分不方便。有了此函数后只需要画网格时取固定名字，该函数会自动根据名字获取其 ID。



注意：此函数只能在 serial 或 host 上调用，node 上调用会返回-1。但可以在 serial 或 host 上获取以后再调用 host_to_node_int 系列函数将值传给 node。

3. HWND SuperUdf_GetFluentMainWnd();

获得 Fluent 的主窗体句柄（仅企业版，详见编程手册）

4. void SuperUdf_Steady_Iterate(int nTimes)

驱动 Fluent 进行稳态迭代 n 步（仅企业版，详见编程手册）

5. void SuperUdf_ExecuteConsoleCommand(char* strAnsiConsoleCommand)

驱动 Fluent 执行 TUI 或 Scheme 命令（仅企业版，详见编程手册）

6. void SuperUdf_AddUserMenu(UINT uMenuResourceID)

在 Fluent 中插入用户菜单（仅企业版，详见企业版编程手册）

7. void SuperUdf_EnableMenuItem(UINT uTargetMenuID, BOOL bEnabled)

将用户菜单变灰禁止或恢复可用（仅企业版，详见编程手册）

8. void SuperUdf_SetMenuBmpAndFun(MenuItemBmpAndFun menuBmpAndFuns[], ULONG nCount)
设置用户菜单对应的位图和点击响应函数（仅企业版，详见编程手册）
9. void SuperUdf_SetMenuSelectCallBack(MENUSELECTPROC UserCallBackFunction)
设置选择菜单时响应回调函数，可在此回调函数中动态使菜单变灰或恢复（仅企业版）

学术版拓展函数实例：

以下为学术版中拓展函数的编程实例（企业版拓展函数实例见编程手册）。

```
#include "udf.h"
#include "SuperUdfExtension.h"
#pragma comment(lib, "SuperUdfExtension.lib")

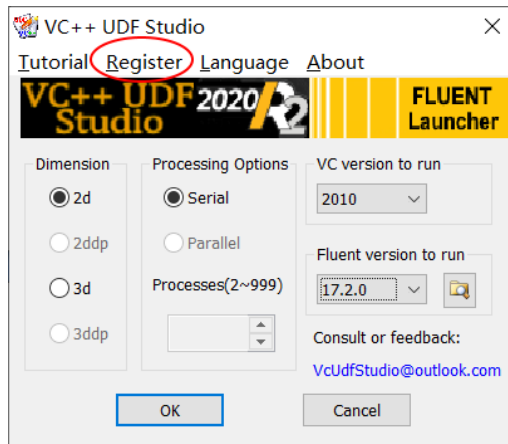
DEFINE_ON_DEMAND(GetOutletId)
{
    int outlet_id;
    face_t f;
    Thread*tf;
    Domain*domain=Get_Domain(1);
    #if !RP_NODE
        outlet_id=SuperUdf_GetZoneIdByName("outlet"); //get the id of zone whose name is "outlet"
    #endif
    host_to_node_int_1(outlet_id);

    #if !RP_HOST
        if(-1==outlet_id)
            Message("Can't get the ID on myid=%d\n",myid);
        else
        {
            tf=Lookup_Thread(domain, outlet_id);
            Message("myid=%d, outlet id=%d\n",myid, outlet_id);
            begin_f_loop(f,tf)
            {
                if(PRINCIPAL_FACE_P(f,tf))
                {
                    // loop over faces on "outlet"
                }
            }
            end_f_loop(f,tf)
        }
    #endif
}

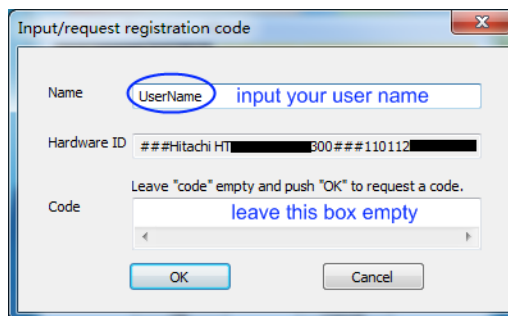
DEFINE_EXECUTE_ON_LOADING(load,libudf)
{
    SuperUdf_Initialize(AfxGetInstanceHandle());
}
```

如何注册:

1. 打开启动器并点击“Register”菜单。



2. 输入你的用户名，“Code”文本框留空，然后点击“OK”。您的用户名和硬件信息会被存放到文本文件“user.ini”。



3. 将“user.ini”文件作为附件 email 发送给 vcUdfStudio@outlook.com (国际) 或 vcUdfStudio@sohu.com (中国). “user.ini”的内容具有如下形式:

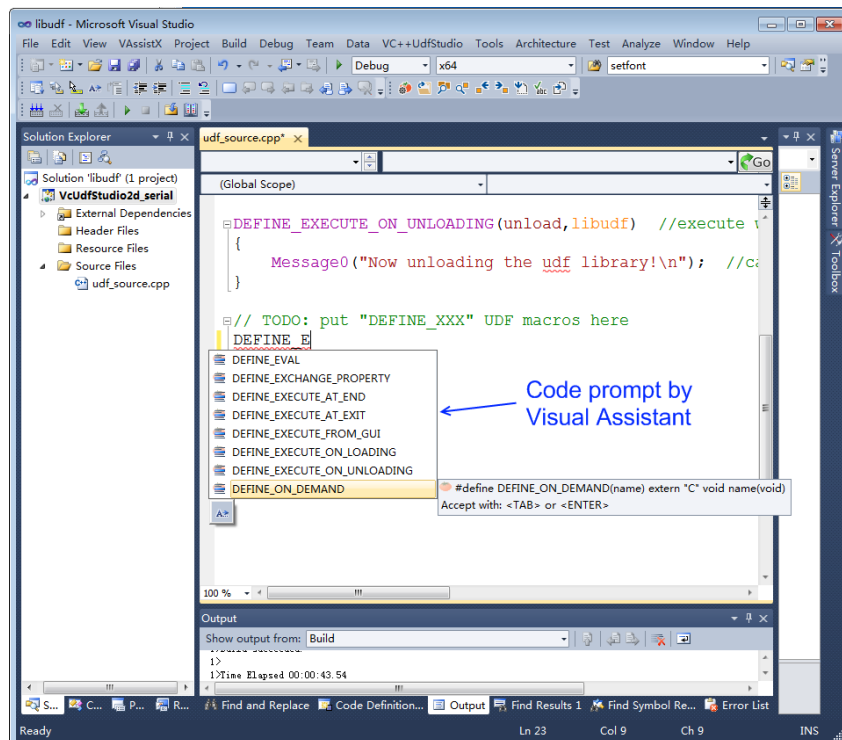
Name=UserName

HardWare ID=###Hitachi HTSABC434CD300###110782PCDN403M7H6KM

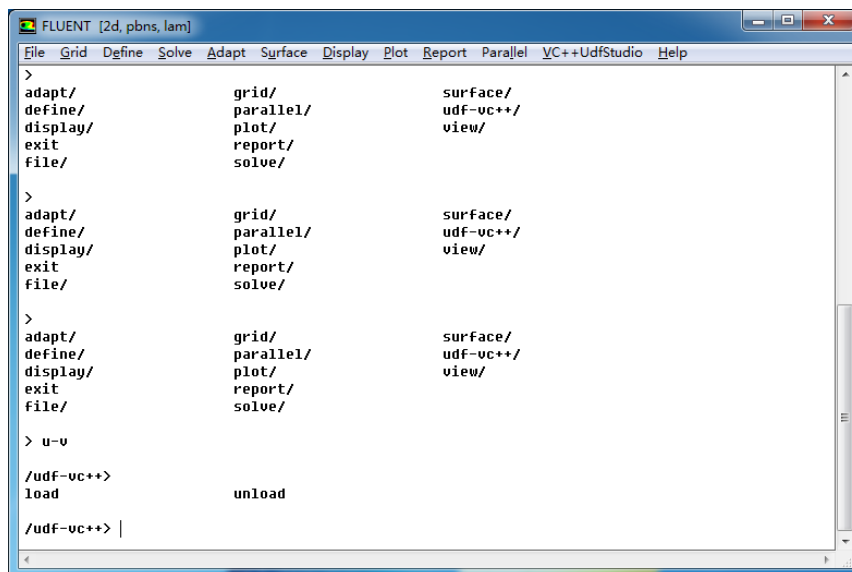
4. 联系 vcUdfStudio@outlook.com (国际)或 vcUdfStudio@sohu.com (中国)并付款，收到返回的带有注册号的 email 后，以管理员权限重新运行启动器。再次点击“Register”菜单，输入您的用户名和注册号，所有的功能即可解锁。

小贴士:

1. 强烈推荐安装“Visual assistant”(www.wholetomato.com) 插件。该插件具有很多扩展功能(例如代码自动补全, 括号匹配, 用户自定义关键词颜色等等)。



2. 利用 TUI 命令 `udf-vc++/load` 或 `udf-vc++/unload` 可以加载或卸载 Fluent 中的 VC++UdfStudio 菜单。



更多信息详见网站 <https://vcUdfStudio.bitbucket.io>

订购注册版或报告错误，请联系 vcUdfStudio@outlook.com（国际）或 vcUdfStudio@sohu.com（中国）。