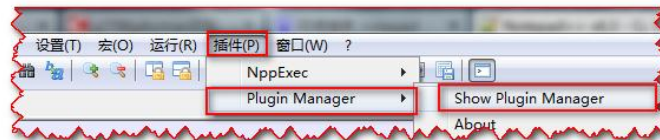


1、下载 notepad++

进入 Notepad++ 主页 <http://notepad-plus-plus.org/> 随便下载某个版本的 notepad++ ,并进行安装。

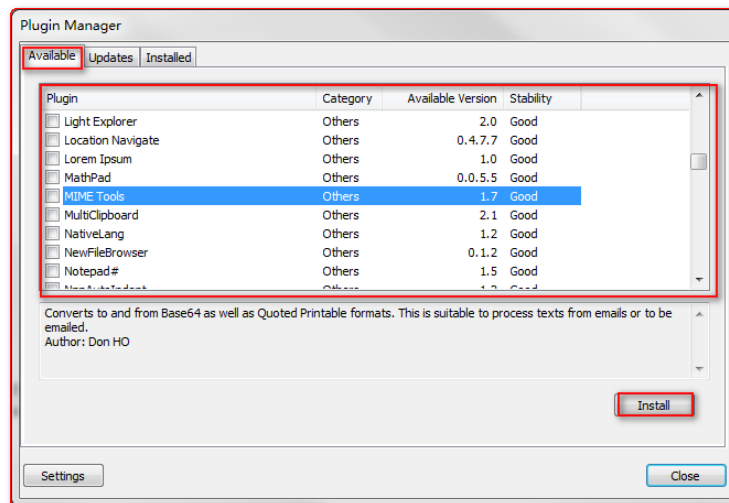
安装过程略。。。

安装结束后运行 notepad++ ,点击菜单 “插件(P)” -> “Plugin Manager” -> "Show Plugin Manager"。

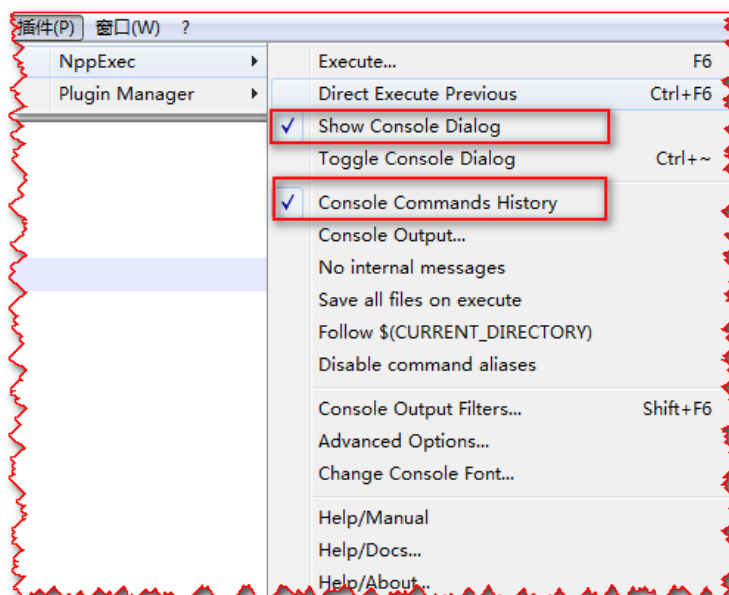


在弹出的 “Plugin Manager” 对话框中的 “Available” 标签下选择 “NppExec” ，点击 “Install” 。

注：“Available” 标签下的可选插件要在联网时才会出现。



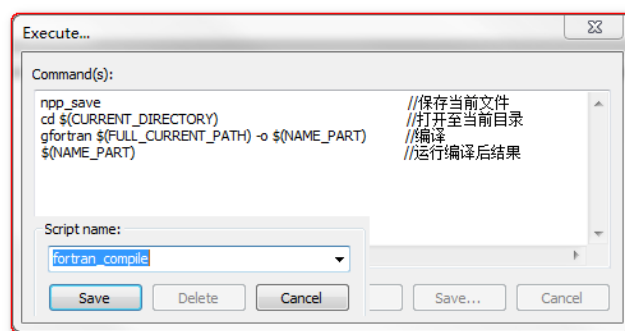
NppExec 插件安装完后，可以在菜单 “插件 (P)” 下看到，楼主一般勾选下面两项。



2、在 NotePad++ 中利用 NppExec 配置 fortran IDE

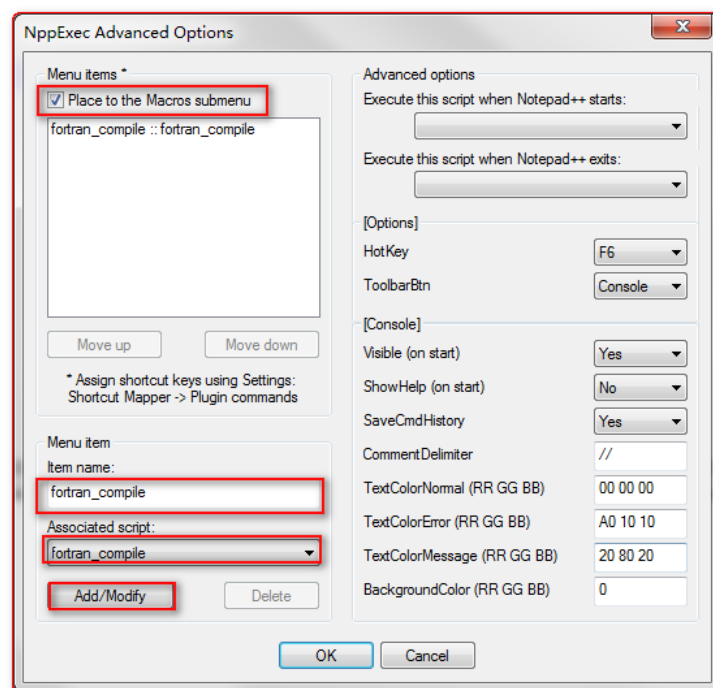
首先用 Notepad++ 随便打开一个 fortran 文件 按 F6 或者从菜单“插件 (P)”->“NppExec”->“Execute...”。在“Execute...”对话框(见下图)中输入如下命令,后保存为 fortran_compile(这里也可以删除已经定义的 command):

```
npp_save //保存当前文件
cd $(CURRENT_DIRECTORY) //打开至当前目录
gfortran $(FULL_CURRENT_PATH) -o $(NAME_PART) //编译, -g 选项是为了在用 GDB 调试器时加入相
关信息
$(NAME_PART) //运行编译后结果
```

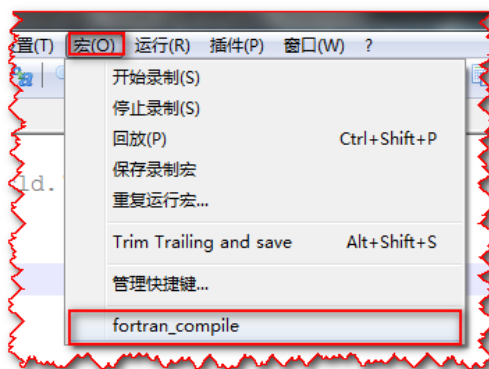


注:若之前没有设置环境变量(安装包中有说明),不能直接使用“gfortran”命令编译,而应该改用 gfortran.exe 所在的完整路径,我的是“C:\MinGW\bin\gfortran.exe”,一定要带“.exe”。

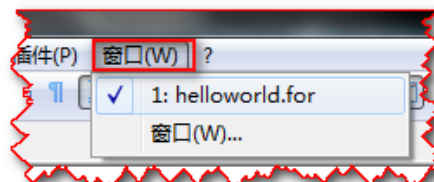
然后在 NppExec 的 Advanced Options 中配置 marco 菜单,首先打开 NppExec 的高级选项对话框



在 “Associated script:” 下拉列表中 ,选择 “fortran_compile” 并点击 “Add/Modify” ,最后勾选 “Place to the Macros submenu” 。在菜单 “宏 (O)” 下会出现子菜单 “fortran_compile” ,下次要编译 fortran 程序就点这个按钮即可。



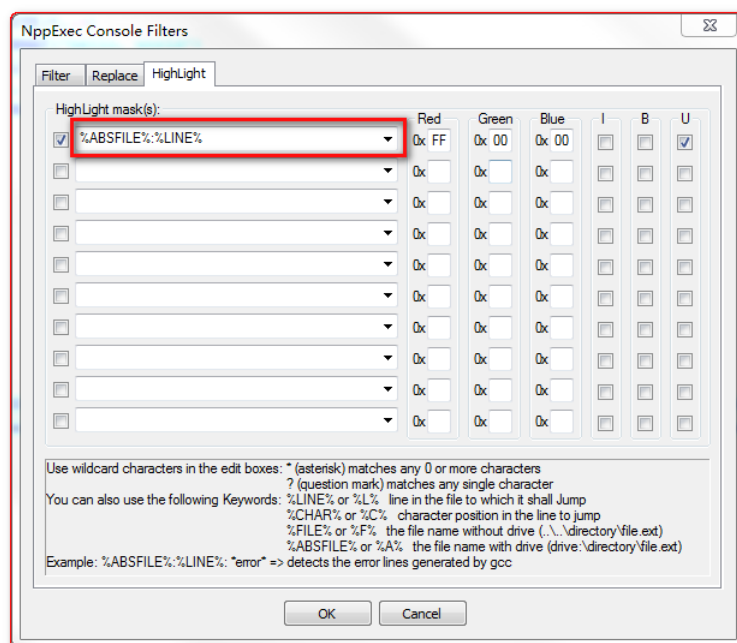
注 :所要编译的 fortran 程序必须在 notepad++ 中是当前激活的窗口,通过菜单 “窗口 (W)” 查看,打钩表示当前文本处于激活状态。



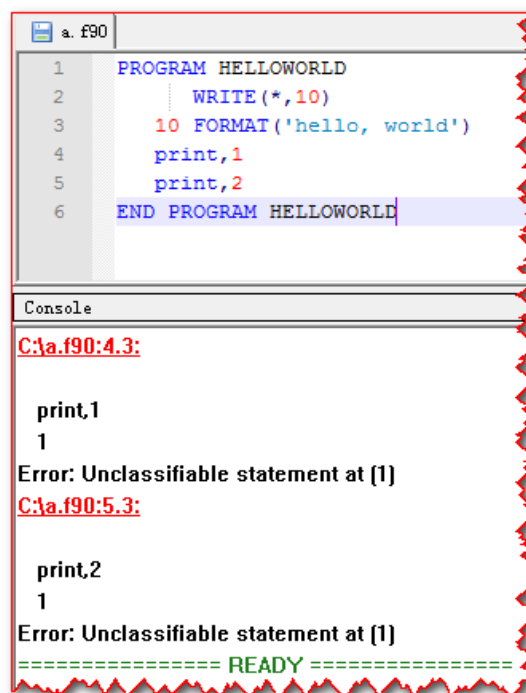
3、设置错误跳转

为了实现在出现编译错误时能准确、快速的定位到源代码的错误位置 , 还需要设置 Console 输出过滤选项, 打开 “NppExec Output Filters...” 或快捷键 shift+F6, 设置如下 (经试验只能这样设置, 其它设置都不能正常使用, 囧!!!) :

%ABSFILE%:%LINE%



在出现错误时，双击下面红色即可自动定位错误位置，例如：



```
1 PROGRAM HELLOWORLD
2     WRITE(*,10)
3     10 FORMAT('hello, world')
4     print,1
5     print,2
6 END PROGRAM HELLOWORLD
```

Console

C:\a.f90:4.3:

print,1
1
Error: Unclassifiable statement at (1)

C:\a.f90:5.3:

print,2
1
Error: Unclassifiable statement at (1)

===== READY =====

4、fortran 调试过程注意事项

1) NppExec 插件还可以创建更多的调试选项，如

①实现 fortran 编译并生成.exe，另存以下命令为“fortran_compile”。

NPP_SAVE

cd \$(CURRENT_DIRECTORY) //打开至当前目录

gfortran \$(FULL_CURRENT_PATH) -o \$(NAME_PART) -g //调试并生成.exe，-g 选项是为了在用 GDB 调试器时加入相关信息

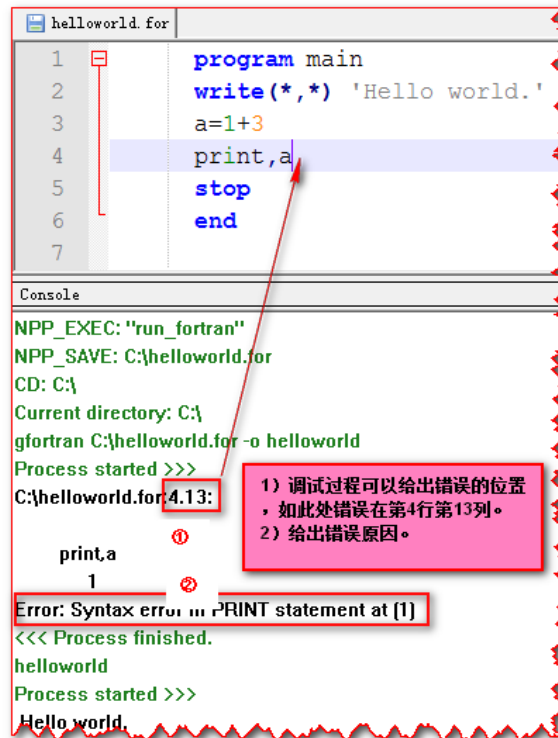
②实现运行生成的.exe，另存以下命令为“fortran_run”。

\$(NAME_PART) //只运行编译生成的.exe

③实现启用 GDB 调试，另存以下命令为“GDB_Debug”。

gdb \$(NAME_PART).EXE

2) 调试会给出错误位置及原因



3) 在“插件” -> “NppExec” 中选择 “Follow \$(CURRENT_DIRECTORY)” 菜单项，表示 Console 窗口的当前目录随同文件；如果不选择的话，在建立脚本文件时就得自己使用 cd 命令切换到当前目录了。

4) 这样的环境非常适合用来编写小程序，一般步骤：新建文档-->notepad++中 写代码-->保存为 fortran 格式-->调试，若无错误提示则编译成功，测试运行结果，否则根据提示行和错误信息修改代码。

5) notepad++环境变量介绍

Notepad++定义的这些环境，可以作为参数传递给外部工具，NppExec 插件可以利用了这些环境变量。

表 1 Notepad++定义的环境变量

变量名	含义	例子 (以 c:\my document\helloworld.for 为例)
FULL_CURRENT_PATH	文件路径名称	c:\my document\helloworld.for
CURRENT_DIRECTORY	文件目录	c:\my document\
FILE_NAME	文件全名称	helloworld.for
NAME_PART	文件名称	helloworld
EXT_PART	文件扩展名	for

6) Notepad++中还有一些比较常用的插件

Tagsview 插件：可以显示程序中定义了哪些变量、subroutine，当然它不仅仅可以用于显示 fortran 程序，还可以用于其它编程语言。楼主很喜欢它！！！！

Tagsjump 插件：实现程序内函数等的跳转，方便程序的阅读。

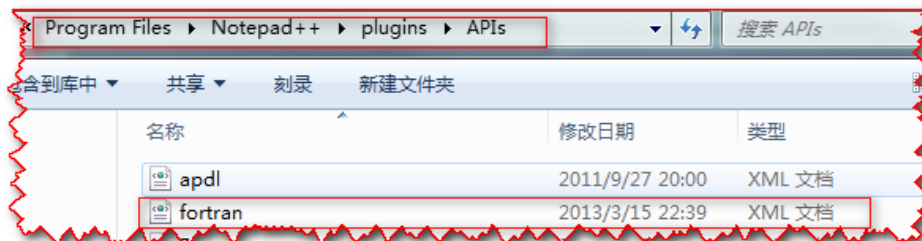
compare 插件：这个看名字就知道怎么用，它实现两个文件的比较。很有用呀，墙裂推荐！！！楼主在编程计算时，需要对程序改动前后输出结果文件进行比较，光靠眼力去比较，会死人的，有木有呀！！

explorer 插件：用于文件的查看还是比较方便的，里面的 favourite 功能还是很贴心的。

XBrackets lite 插件：实现括号的自动补全，不错！！

7) Notepad++ V6.2 以上的版本貌似不能实现函数的自动补全，可以参考一下解决方法。

在 Notepad++ 安装目录的 plugins 文件夹下新建文件夹 APIs，将楼主提供的 fortran.xml (<http://dl.vmall.com/c0n53khr8h>) 放在 APIs 文件夹下。



然后设置 notepad++ 的自动补全功能，设置如下：

