# 、修复OD消息断点Bug

消息断点其实还是条件断点。要下消息断点首先必须获得被调试窗口的窗口过程函数。获得窗口过程函数的方法有两个一个是SetWindowLong，一个是GetClassLong。OD的消息断点是通过GetClassLong这个函数来实现的。

## Bug产生原因

OD在下消息断点过程中要使用GetClassLong函数。在Windows操作系统中该函数有两个版本，GetClassLongA版和GetClassLongW版。分代表的是ASCII和Unicode含义。OD作者在实现消息断点功能的时候只考虑到了ASCII版本，没有考虑到还有Unicode版本。所以在OD调试Unicode窗口程序的时候，下消息断点时，调用的依然是A版GetClassLong，从而而产生错误。

## 修复思路

通过注入手段，Hook系统APIGetClassLong。使其跳转到我们的代码，由我们自己来进行A版和W版判断，并且由我们自己手工调用A版和W版，将正确的返回值（窗口过程函数的地址）给OD就好了。一次性解决类似所有问题。注入手段有Dll劫持，插件编写等方法。

## 修复的具体步骤

1. OD调试OD，定位消息断点Bug产生位置。实现手法是用有Bug的OD\_A调试窗口程序，用OD\_B附加OD\_A，并且在OD\_B中对。GetClassLongA下断点，在OD\_A中打开Windows窗口。这样就可以定位Bug产生位置了。（有关信息：GetClassLong在User32.dll中）

在具体执行过程中：中发现GetClassLong并不是在下消息断点的时候被调用，而是在打开OD\_A的Windows窗口的时候就会调用。



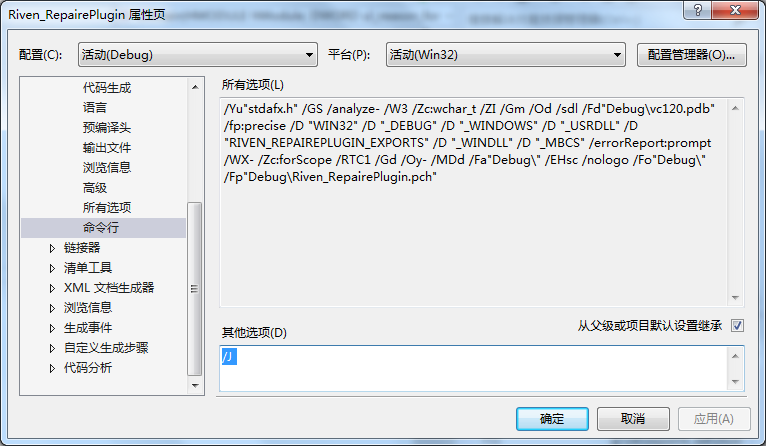
1. 编写插件，修改跳转。经过跟踪分析，这个API调用方式是通过导入表实现。所以call会跳到一个jump [mem]。[mem]其实就是导入表。所以我们只要，编写一个插件，修改导入表中GetClassLongA对应的位置的值。让其跳转到我们自己的代码就好了。

## 插件编写

插件编写，首先要获得插件开发包。其实就是一个帮助文档，加头文件定义，和几个插件的例子。

1. VS2013建立控制台项目，选择Dll工程。导入头文件，添加编译选项\J，测试是否正常。
2. 导出插件的约定的关键回调函数。关键回调函数有2个，约定好的回调有12个（不包括关键回调）。回调都以ODBUG\_Plugin做为前缀命名，编写关键回调。

注意事项：在编译插件的过程中要修改编译选项。这个选项的功能是，让所有的char变成无符号的char。



同时在判断是否是Unicode窗口的时候可以使用API IsWindowUnicode（HWND）。

在编写插件的过程中试图调用Addtolist的时候报错，报错原因分析，是由于LIB导出命名有问题。为了解决。