 来源：http://blog.csdn.net/charlessimonyi/article/details/18447471

[VC与JavaScript交互(一) ———— 如何实现](http://blog.csdn.net/charlessimonyi/article/details/18447471)

分类： [【Win32开发】](http://blog.csdn.net/CharlesSimonyi/article/category/1297611)2014-01-18 15:35 3154人阅读 [评论](http://blog.csdn.net/charlessimonyi/article/details/18447471#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/charlessimonyi/article/details/18447471#report)

为什么要让VC与JavaScript交互？

1.有时候我们需要让自己的软件打开一个网页，来获取页面上的一些数据。这时，可以用mshtml解析HTML提取出数据，也可以向HTML文档动态写入我们准备好的JS代码，用JS代码获取HTML上的数据，然后用VC调用该JS代码取得数据。

2.有时候我们需要让自己的软件打开一个网页并操控该网页，填写表单，提交等动作。这时，可以用mshtml操作HTML，给文本框赋值，模拟点击按钮。也可以向HTML文档动态写入我们准备好的JS代码，用JS代码实现填单，提交等动作，然后用VC调用一下JS代码即可。

3.有时候我们需要用HTML网页做界面，用JS管理HTML页面，用VC调用JS传递进数据，JS把这些数据通过HTML显示在界面上。CHtmlDialog正是这种产物。

4.有时候我们需要基于HTTP协议与WEB服务器直接交互。比如基于HTTP协议来登录QQ空间。但是对浏览器抓包发现QQ号是明文传输，但QQ密码是密文，它是如何加密的呢？这些加密算法肯定都在页面上的JS里，因为这个加密过程是在客户浏览器上实现的。我们可以找到加密相关的JS代码，仔细阅读分析，改写为C++代码实现，但比较复杂。一个简单的办法就是直接把这些用于加密的JS代码复制出来，用VC调用JS代码，让它的JS代码完成加密过程，然后我们把密文拿过来，用HTTP协议POST发送到WEB服务器，即完成了登陆动作。

5.等等，还有很多用途。

那么，在Windows平台上用VC开发的程序，如何与JavaScript交互？

通常，我们都是用WebBrowser加载包含JS代码的HTML，然后通过WebBrowser对象获取IHTMLDocument接口(对于ActiveX的WebBrowser来说是get\_Document方法，对于CHtmlView对象来说是GetHtmlDocument方法)。其中IHTMLDocument2接口有一个get\_Script方法，可以获取用于控制JS代码的IDispatch接口。VC调用JS函数，都是通过这个IDispatch接口的Invoke方法来完成。使用IDispatch接口的GetIDsOfNames方法根据JS函数名获取调度标识符DISPID，使用Invoke来调用JS函数。Invoke最后两个参数用于返回错误信息，可为NULL。这个IDispatch接口调用起来很麻烦，下文将会介绍如何简便的调用。网上有个外国人写了个CWebPage类实现VC与JS交互，用的正是这种方法。

<http://www.codeproject.com/Articles/2352/JavaScript-call-from-C>

关于WebBrowser：

在VC中使用WebBrowser，一般有两种方法。MFC中有个CHtmlView封装了WebBrowser，用起来很方便。虽然CHtmlView派生于CView，是个视图类，但它也派生于CWnd，将其用于对话框上，完全没有问题，只是在某些地方需要小修改一下。其中需要注意的两个问题就是：

1. CHtmlView的构造函数是protected的，不允许直接构造一个CHtmlView对象。必须从CHtmlView派生后再构造。

2.如果是在栈上创建CHtmlView对象，必须重载PostNcDestroy并什么也不写。因为默认的PostNcDestroy会delete  this;而出错。如果是在堆上创建CHtmlView对象则要注意防止二次delete。

另一种方法是使用WebBrowser的ActiveX控件，这种方法可以在MFC项目中使用，也可以在非MFC项目中使用。

关于IDispatch：

我们知道IDispatch是COM双接口中的调度接口。一般用于供脚本语言调用COM组件。对于编译型的C++语言，让它调用这种接口，是很麻烦的。毕竟用IDispatch接口调用COM对象的各种方法、设置与获取COM对象的属性、让COM对象回调我们，都是用IDispatch的Invoke方法来实现。一个Invoke就要实现那么多功能，用起来当然很麻烦。不过好在ATL智能指针类中的CComDispatchDriver(即CComQIPtr<IDispatch>)封装了IDispatch接口，使用我们用起来大大的方便！CComDispatchDriver对GetIDsOfNames和Invoke进一步进行了封装，只需更少的参数即方便可调用。

获取和设置COM对象属性可以用CComDispatchDriver的这些方法：

GetProperty

GetPropertyByName

PutProperty

PutPropertyByName

其实使用IDispatch调度接口来设置、获取COM属性，调用COM方法，都是使用GetIDsOfNames和Invoke。实际上这四个方法都是对GetIDsOfNames和Invoke的封装。简化调用的复杂性。

调用COM对象的方法可以用这些方法：

Invoke0    //调用0个参数的方法

Invoke1    //调用1个参数的方法

Invoke2    //调用2个参数的方法

InvokeN    //调用多个参数的方法

这些函数都有两个版本，一个是接受调度标示符DISPID，需要自己先调用GetIDsOfNames来获取。一个是接受OLE字符串的版本，这个版本在内部会调用GetIDsOfNames来获取DISPID。这些函数用起来很方便，不需要我们自己填充DISPPARAMS结构，但是它对原始Invoke的调用时，最后两个参数都是NULL，即不需要获取错误信息。如果需要获取错误信息，我们需要自行调用原始Invoke方法。

注意，这些方法是ALT的CComDispatchDriver封装的方法，调用时应使用"."而不是"->"。因为"->"获得的是CComDispatchDriver内部的Dispatch指针。

另一个要注意的问题是，一定要等Navigate完全加载一个html文档后（触发OnDocumentComplete）。才能获取IHTMLDocument2和Script。否则会出现空指针或找不到JS函数。所以不能在调用Navigate打开HTML后就紧接着获取IHTMLDocument2和Script，要等HTML文档加载完。

上面说了这么多COM对象，和VC调用JS有什么关系？别忘了我们用IHTMLDocument2接口的get\_Script方法获取到了代表HTML文档中JS代码的IDispatch接口，我们用IDispatch接口，把HTML文档中的这堆JS代码当作一个COM对象，来操控它。上面说的Invoke0,Invoke1,Invoke2,InvokeN，正是分别被我们用来调用0个参数的JS函数，1个参数的JS函数，2个参数的JS函数，N个参数的JS函数。

[VC与JavaScript交互(二) ———— 调用JS函数](http://blog.csdn.net/charlessimonyi/article/details/18450783)

分类： [【Win32开发】](http://blog.csdn.net/CharlesSimonyi/article/category/1297611)2014-01-18 15:35 3740人阅读 [评论](http://blog.csdn.net/charlessimonyi/article/details/18450783#comments)(19) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/charlessimonyi/article/details/18450783#report)

这一章，我们来动手实践VC调用JS函数。

我们动手写一个HTML，其中包含这样一段JS代码:

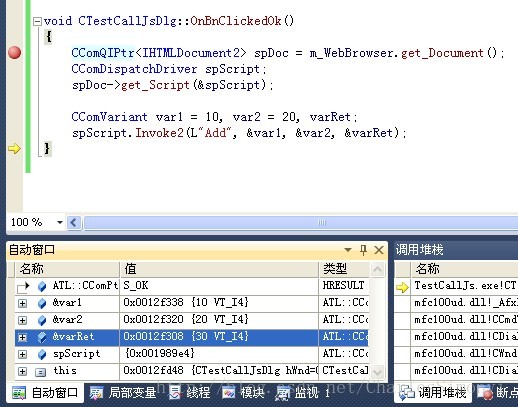
**[html]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. **<script** type="text/javascript"**>**
2. function Add(value1, value2) {
3. return value1 + value2;
4. }
5. **</script>**

然后我们用WebBrowser加载这个HTML后，在VC中这样来调用这个函数名为Add的JS函数:

**[cpp]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. //别忘了#include <MsHTML.h>
2. //m\_WebBrowser是一个WebBrowser的Activex控件对象。
3. CComQIPtr<IHTMLDocument2> spDoc = m\_WebBrowser.get\_Document();
4. CComDispatchDriver spScript;
5. spDoc->get\_Script(&spScript);
7. CComVariant var1 = 10, var2 = 20, varRet;
8. spScript.Invoke2(L"Add", &var1, &var2, &varRet);



spScript.Invoke2的作用是调用JS函数中名为Add的函数，传入两个参数，用varRet接收返回值。

可以看到，Invoke2调用成功后，varRet得到了返回值30。

但这样的话一次只能接受一个返回值。

如果要一次接受多个返回值的话，怎么办呢？

我们可以让JS返回一个JS中的Array数组或Object对象。

当JS函数return一个Array或一个Object对象时，VC这边的varRet将接受到一个代表该对象的IDispatch接口。我们仍然用CComDispatchDriver来管理这个IDispatch。用上一篇文章介绍的CComDispatchDriver的四个方法：

GetProperty

GetPropertyByName

PutProperty

PutPropertyByName

来从这个Array或Object对象中取出我们要的数据。

实践是检验真理的唯一标准，让我们再来写一个JS函数:

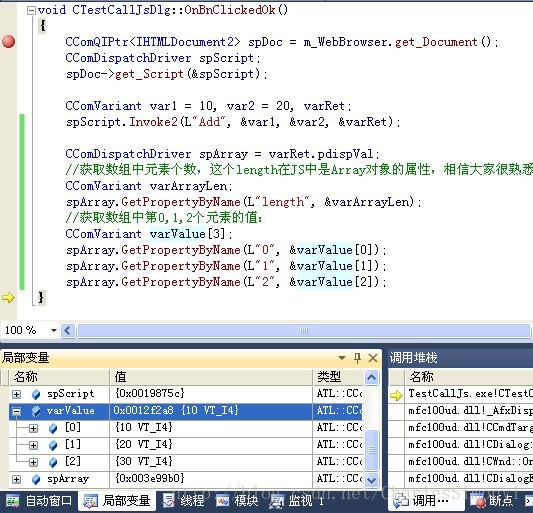
**[html]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. **<script** type="text/javascript"**>**
2. function Add(value1, value2) {
3. var array = new Array();
4. array[0] = value1;
5. array[1] = value2;
6. array[2] = value1 + value2;
7. return array;
8. }
9. **</script>**

然后在VC中这样写：

**[cpp]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. CComQIPtr<IHTMLDocument2> spDoc = m\_WebBrowser.get\_Document();
2. CComDispatchDriver spScript;
3. spDoc->get\_Script(&spScript);
5. CComVariant var1 = 10, var2 = 20, varRet;
6. spScript.Invoke2(L"Add", &var1, &var2, &varRet);
8. CComDispatchDriver spArray = varRet.pdispVal;
9. //获取数组中元素个数，这个length在JS中是Array对象的属性，相信大家很熟悉
10. CComVariant varArrayLen;
11. spArray.GetPropertyByName(L"length", &varArrayLen);
12. //获取数组中第0,1,2个元素的值：
13. CComVariant varValue[3];
14. spArray.GetPropertyByName(L"0", &varValue[0]);
15. spArray.GetPropertyByName(L"1", &varValue[1]);
16. spArray.GetPropertyByName(L"2", &varValue[2]);



可以看到，10,20,30，这三个JS函数返回的值已经躺在我们的varValue[3]里了。

当然，如果不知道JS返回的Array对象里面有几个元素，我们可以在VC这边获取它的length属性，然后在一个循环中取出数组中的每个值。

如果我们的JS函数返回一个包含有多个属性值的Object对象，VC这边该如何接收呢？

让我们再来写一个JS函数:

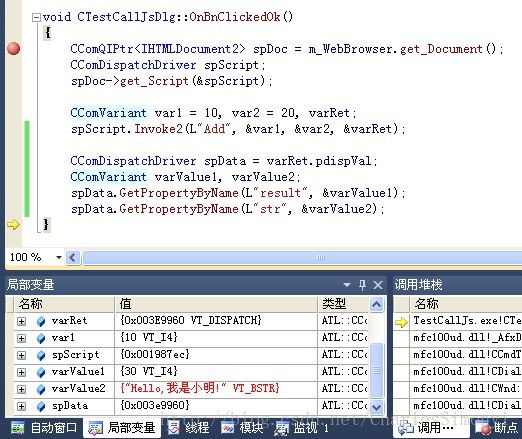
**[html]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. **<script** type="text/javascript"**>**
2. function Add(value1, value2) {
3. var data = new Object();
4. data.result = value1 + value2;
5. data.str = "Hello,我是小明!";
6. return data;
7. }
8. **</script>**

然后在VC中我们这样接收：

**[cpp]** [view plaincopy](http://blog.csdn.net/charlessimonyi/article/details/18450783)[在CODE上查看代码片](https://code.csdn.net/snippets/161967)

1. CComQIPtr<IHTMLDocument2> spDoc = m\_WebBrowser.get\_Document();
2. CComDispatchDriver spScript;
3. spDoc->get\_Script(&spScript);
5. CComVariant var1 = 10, var2 = 20, varRet;
6. spScript.Invoke2(L"Add", &var1, &var2, &varRet);
8. CComDispatchDriver spData = varRet.pdispVal;
9. CComVariant varValue1, varValue2;
10. spData.GetPropertyByName(L"result", &varValue1);
11. spData.GetPropertyByName(L"str", &varValue2);



我们从JS返回的Object对象里取出了它的两个属性，result和str，分别是一个整形数据和一个字符串。

这里JS代码是我们自己写的，在VC这边当然事先知道这个JS函数返回的对象有result和str这两个属性。

如果JS代码不是我们写的，或者它的属性是事先不能确定的，该怎么办呢？答案是使用IDispatchEx接口来枚举这个对象的相关信息(方法名、属性名)。

这个现在暂时不讲，在后续的文章中会讲。

当然，JS不只可以返回Object对象，返回什么对象都可以，当返回一个对象而非基本数据类型（整形、浮点、字符串）时，VC这边收到的返回值是一个IDispatch，然后我们需要调用GetPropertyByName方法从这个IDispatch代表的对象中取出它的属性来。

这样一来，VC调用JS函数，传递参数给JS和JS返回返回值给VC，大致就都会了。

对于CComVariant包装的VARIANT这种智能型变量，不了解的可以到网上看下相关资料。《深入解析ATL》之类的书上均有介绍。

值得注意的是ATL提供的这些CCom开头的智能包装类，并不依赖于ATL的动态库。因为我在VC项目中并没有选择链接ATL，程序调试运行时进程加载的模块中也有没有ATL100.dll之类的模块载入。大家可以放心使用而不用担心依赖上ATL。

VC调用JS函数没问题了。那么JS函数如何调用VC呢？我们将在下一篇文章中慢慢道来。