
ATL-COM 学习笔记

1、如何替换默认的类厂？	3
2、如何让 COM 组件支持 Web 事件？	4
3、如何动态创建要求 License 的 ActiveX 控件？	5
4、如何在 Web 页面上生成基于 XML 的树形菜单？	5
5、如何在 Web 页面上创建动态表格？	9
6、如何为 Win32 服务程序提供事件源消息文件？	9
7、如何创建单实例（静态）COM 组件？	11
8、如何在 Delphi 6 及以上版本中使用可变常量（类型常量）？	12
9、几个有用的 ANSI C 和 VC++ 预定义宏.....	12
10、如何在自己的程序中使用脚本语言？	13
11、如何在基于 CHtmlView 的程序中使用 HTML 事件映射？	20
12、如何得到 Web 页面中的 COM 组件的原始接口？	27
13、如何生成全球唯一标识（UID）？	28
14、如何重写 MFC 的缺省控件容器（IOleContainer）？	31
15、如何重写 WebBrowser 控件的缺省安全处理功能？	32
16、如何重写 WebBrowser 控件的缺省下载管理器？	38
17、如何定制 IE 或 WebBrowser 控件的缺省打印和打印预览框？	40
18、如何扩展 IE 或 WebBrowser 控件提供的文档对象模型（DOM）？	40
19、如何给 WebBrowser 控件的扩展文档对象（文档. external）提供事件能力？	65
20、如何打开流中的 HTML 文档？	66
21、如何使 Web 页面中的对象可以拖动？	71
22、如何把 Web 页面中用到的图像等外部文件嵌入 DLL 或 Exe？	72
23、如何使用 #import 指令从文件定位特定类型库资源 ID？	73
24、如何让你的 MFC 类实现 IDispatch 接口？	73
25、如何自行管理 IE 缓存目录里的内容？	74
26、Windows Mobile 里如何使用 WebBrowser 控件？	74
27、为什么在 Windows Mobile 里无法使用宿主在 exe 里的 COM 组件？	75
28、如何扩展 IE 的协议，使其支持自定义的协议？	75

1、如何替换默认的类厂？

首先定义自己的类厂：

```
template<class T>
class CDictionaryClassFactory : public CComClassFactory
{
private:
    long m_lRef;
public:
    CDictionaryClassFactory(void) {m_lRef = 0;}
    virtual ~CDictionaryClassFactory(void) {};
public:
    HRESULT STDMETHODCALLTYPE CreateInstance(
        /* [unique][in] */ IUnknown *pUnkOuter,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void **ppvObject)
    {
        //添加自己的组件实例化代码。
        CComObject<T>* p = new CComObject<T>;
        return p->QueryInterface(riid, ppvObject);
    }

    HRESULT STDMETHODCALLTYPE LockServer(
        /* [in] */ BOOL fLock)
    {
        return _At1Module.Lock();
    }

    HRESULT STDMETHODCALLTYPE QueryInterface(
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject)
    {
        *ppvObject = NULL;
        if(riid==IID_IUnknown||riid==IID_IClassFactory)
            *ppvObject = this;
        if(*ppvObject)
        {
            AddRef();
            return S_OK;
        }
    }

    return E_NOINTERFACE;
```

```
}

ULONG STDMETHODCALLTYPE AddRef( void)
{
    return InterlockedIncrement(&m_lRef);
}

ULONG STDMETHODCALLTYPE Release( void)
{
    if(InterlockedDecrement(&m_lRef) ==0)
    {
        delete this;
        return 0;
    }

    return m_lRef;
}
};


```

然后可以通过下面的方法应用自己的类厂：

找到 COM 组件的组件类的声明，添加代码如下：

DECLARE_CLASSFACTORY_EX(你的类厂名称)，本例添加：
DECLARE_CLASSFACTORY_EX(CDictionaryClassFactory)

2、如何让 COM 组件支持 Web 事件？

首先让组件支持事件（出）接口，生成 COM 对象时选中支持连接点（support connect point）。

实现IProvideClassInfo2接口，方法是在COM组件类的派生类中添加
public IProvideClassInfo2Impl<&CLSID_XX, NULL, &LIBID_XXLib>
然后在接口映射中添加：

```
BEGIN_COM_MAP(xxx)
.....
COM_INTERFACE_ENTRY(IProvideClassInfo)
COM_INTERFACE_ENTRY(IProvideClassInfo2)
END_COM_MAP()
```

Web 页代码如下：

```
<OBJECT clsid=xxxx id=obj1></OBJECT>
<SCRIPT ID=obj1Script FOR="obj1" EVENT="onclick()" language="javascript">
    alert("event fired");
</SCRIPT>
```

或者

```
<SCRIPT ID=obj1Script FOR="obj1" EVENT="onclick()"  
language="javascript">  
function obj1::onclick()  
{  
    alert( "event fired" );  
}  
</SCRIPT>
```

3、如何动态创建要求 License 的 ActiveX 控件？

先用 VC++ ClassWizzard 创建控件包装类 CXXXX，
用 Licreqst.exe（微软网站提供下载）取得控件的 license 信息
WCHAR pwchLicenseKey[] =

```
{  
    0x0043, 0x006F, 0x0070, 0x0079, 0x0072, 0x0069,  
    0x0067, 0x0068, 0x0074, 0x0020, 0x0028, 0x0063,  
    0x0029, 0x0020, 0x0032, 0x0030, 0x0030, 0x0031,  
    0x0020, 0x004E, 0x0055, 0x0047, 0x0052, 0x004F,  
    0x004F, 0x0056, 0x005A  
};
```

```
BSTR bStrLicense = ::SysAllocStringLen(pwchLicenseKey,  
sizeof(pwchLicenseKey)/sizeof(WCHAR));
```

```
CXXXX* pWnd=new CXXXX();  
BOOL bRetBurn=pWnd->Create("Burn",  
    0,  
    CRect(0,0,0,0),  
    pWnd,  
    IDC_CDWRITERXPCTRL,  
    NULL,  
    FALSE,  
    bStrLicense);  
::SysFreeString(bStrLicense);
```

4、如何在 Web 页面上生成基于 XML 的树形菜单？

示例代码如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">  
<HTML>
```

```
<HEAD>
<TITLE> DSTree </TITLE>
<META NAME="Author" CONTENT="sTarsjz@hotmail.com" >
<style>
body,td{font:12px verdana}
#treeBox {background-color:#fffffa;}
#treeBox .ec{margin:0 5 0 5;}
#treeBox .hasItems{font-weight:bold;height:20px;padding:3       6       0
6;margin:2px;cursor:hand;color:#555555;border:1px solid #fffffa;}
#treeBox .Items{height:20px;padding:3 6 0 6;margin:1px;cursor:hand;color:#555555;border:1px
solid #fffffa;}
</style>
<base href="http://vip.5d.cn/star/dstree/" />
<script>
//code by star 20003-4-7
var HC = "color:#990000;border:1px solid #cccccc";
var SC = "background-color:#efefef;border:1px solid #cccccc;color:#000000;";
var IO = null;
function initTree()
{
    var rootn = document.all.menuXML.documentElement;
    var sd = 0;
    document.onselectstart = function(){return false;};
    document.all.treeBox.appendChild(createTree(rootn, sd));
}
function createTree(thisn, sd)
{
    var nodeObj = document.createElement("span");
    var upobj = document.createElement("span");
    with(upobj)
    {
        style.marginLeft = sd*10;
        className = thisn.hasChildNodes()?"hasItems":"Items";
        innerHTML = "<img src=expand.gif class=ec>" + thisn.getAttribute("text") + "";
        onmousedown = function()
        {
            if(event.button != 1) return;
            if(this.getAttribute("cn"))
            {
                this.setAttribute("open", !this.getAttribute("open"));
                this.cn.style.display = this.getAttribute("open")?"inline":"none";
                this.all.tags("img")[0].src =
this.getAttribute("open")?"expand.gif":"contract.gif";
            }
        }
    }
}
```

```
        }
        if(IO)
        {
            IO.runtimeStyle.cssText = "";
            IO.setAttribute("selected",false);
        }
        IO = this;
        this.setAttribute("selected",true);
        this.runtimeStyle.cssText = SC;
    }
    onmouseover = function(){
        if(this.getAttribute("selected"))return;
        this.runtimeStyle.cssText = HC;
    }
    onmouseout = function(){
        if(this.getAttribute("selected"))return;
        this.runtimeStyle.cssText = "";
    }
    oncontextmenu = contextMenuHandle;
    onclick = clickHandle;
}

if(thisn.getAttribute("treeId") != null){
    upobj.setAttribute("treeId",thisn.getAttribute("treeId"));
}
if(thisn.getAttribute("href") != null){
    upobj.setAttribute("href",thisn.getAttribute("href"));
}
if(thisn.getAttribute("target") != null){
    upobj.setAttribute("target",thisn.getAttribute("target"));
}

nodeObj.appendChild(upobj);
nodeObj.insertAdjacentHTML("beforeEnd","<br>

if(thisn.hasChildNodes()){
    var i;
    var nodes = thisn.childNodes;
    var cn = document.createElement("span");
    upobj.setAttribute("cn",cn);
    if(thisn.getAttribute("open") != null){
        upobj.setAttribute("open",(thisn.getAttribute("open")=="true"));
        upobj.getAttribute("cn").style.display = upobj.getAttribute("open")?"inline":"none";
        if( !upobj.getAttribute("open"))upobj.all.tags("img")[0].src ="contract.gif";
    }
}
```

```

        }

        for(i=0;i<nodes.length;cn.appendChild(createTree(nodes[i++],sd+1)));
        nodeObj.appendChild(cn);
    }
    else{
        upobj.all.tags("img")[0].src ="endnode.gif";
    }
    return nodeObj;
}
window.onload = initTree;
</script>

<script>
function clickHandle(){
    // your code here
}
function contextMenuHandle(){
    event.returnValue = false;
    var treeId = this.getAttribute("treeId");
    // your code here
}
</script>
</HEAD>
<BODY>
<xml id=menuXML>
<?xml version="1.0" encoding="GB2312"?>
<DSTreeRoot text="根节点" open="true" href="http://" treeId="123">

    <DSTree text="技术论坛" open="false" treeId="">
        <DSTree text="5DMedia" open="false" href="http://" target="box" treeId="12">
            <DSTree text="网页编码" href="http://" target="box" treeId="4353" />
            <DSTree text="手绘" href="http://" target="box" treeId="543543" />
            <DSTree text="灌水" href="http://" target="box" treeId="543543" />
        </DSTree>
        <DSTree text="BlueIdea" open="false" href="http://" target="box" treeId="213">
            <DSTree text="DreamWeaver & JS" href="http://" target="box" treeId="4353" />
            <DSTree text="FlashActionScript" href="http://" target="box" treeId="543543" />
        </DSTree>
        <DSTree text="CSDN" open="false" href="http://" target="box" treeId="432">
            <DSTree text="JS" href="http://" target="box" treeId="4353" />
            <DSTree text="XML" href="http://" target="box" treeId="543543" />
        </DSTree>
    </DSTree>
</xml>

```

```
</DSTree>

<DSTree text="资源站点" open="false" treeId="">
    <DSTree text="素材屋" href="http://" target="box" treeId="12" />
    <DSTree text="桌面城市" open="false" href="http://" target="box" treeId="213">
        <DSTree text="壁纸" href="http://" target="box" treeId="4353" />
        <DSTree text="字体" href="http://" target="box" treeId="543543" />
    </DSTree>
    <DSTree text="MSDN" open="false" href="http://" target="box" treeId="432">
        <DSTree text="DHTML" href="http://" target="box" treeId="4353" />
        <DSTree text="HTC" href="http://" target="box" treeId="543543" />
        <DSTree text="XML" href="" target="box" treeId="2312" />
    </DSTree>
</DSTree>

</DSTreeRoot>
</xml>
<table style="position:absolute;left:100;top:100;">
<tr><td id=treeBox style="width:400px;height:200px;border:1px solid #cccccc;padding:5 3 3 5;" valign=top></td></tr>
<tr><td style="font:10px verdana;color:#999999" align=right>by <font color="#660000>sTar</font><br> 2003-4-8</td></tr>
</table>
</BODY>
</HTML>
```

5、如何在 Web 页面上创建动态表格？

示例代码如下：

[打开页面](#)

6、如何为 Win32 服务程序提供事件源消息文件？

由于服务程序无法通过 GUI 与系统管理员进行交互，所以服务程序的运行状态一般记录在系统的事件日志中，系统事件的事件标识，事件类别，事件参数不能直接提供字符串描述而必须存放在消息文件（message file）中，使用消息文件的方法如下：

1. 创建消息文件：

```
***** MessageID.mc *****
LanguageNames=(British=0x809:MSG00809)

#define CAT_MWL_SERVICE 1
```

MessageId=0x1

Language=English

Running Log

Language=British

Greeting

MessageId=0x1000

SymbolicName=MC_SERVICE

Language=English

%1

Language=British

%1

用 mc.exe 编译 MessageID.mc,生成 MSG00001.bin, MSG00809.bin, MessageID.rc

2. 修改服务程序的资源文件 (*.rc), 添加以下行:

LANGUAGE 0x9,0x1

1 11 MSG00001.bin

LANGUAGE 0x9,0x2

1 11 MSG00809.bin

3. 修改服务的注册脚本文件 (*.rgs), 添加以下行:

HKLM

{

 NoRemove System

{

 NoRemove CurrentControlSet

{

 NoRemove Services

{

 NoRemove EventLog

{

 NoRemove Application

{

 MWLServer //你的服务名称

{

 val EventMessageFile = s '%MODULE%'

 val CategoryMessageFile = s '%MODULE%'

 val TypesSupported = d 00000007

 val CategoryCount = d 00000001

}

}

```
        }
    }
}
}
```

7、如何创建单实例（静态）COM 组件？

静态全局的COM组件常用来表示全局对象或无变化的对象，如：服务程序的抽象，Application对象；这种类型的对象随COM服务的启动而生成，COM服务的关闭而消亡；还有如字典对象，这种对象一般较大，构造较慢，但一旦生成就很少变化。

在新版本的ATL中可以这样实现：

直接在你的COM类里添加 DECLARE_CLASSFACTORY_SINGLETON（你的静态对象实例）。

如果你的ATL里没有这个宏，就需要自己来实现：

首先修改组件的默认IUnknown实现，去除引用计数管理：

```
template<class T>
class CMyObject:public CComObject<T>
{
    virtual HRESULT STDMETHODCALLTYPE QueryInterface(
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject)
    {
        return _InternalQueryInterface(riid, ppvObject);
    }

    virtual ULONG STDMETHODCALLTYPE AddRef( void ) {return 1;};

    virtual ULONG STDMETHODCALLTYPE Release( void ) {return 1;};
};
```

然后按照How To 1的方法自定义类厂，修改类厂的CreateInstance实现：

```
HRESULT STDMETHODCALLTYPE CreateInstance(
    /* [unique][in] */ IUnknown *pUnkOuter,
    /* [in] */ REFIID riid,
    /* [iid_is][out] */ void **ppvObject)
{
    //创建静态的组件对象。
    static CMyObject<T> StaticInst;
    return StaticInst.QueryInterface(riid, ppvObject);
}
```

8、如何在 Delphi 6 及以上版本中使用可变常量（类型常量）？

可变常量常用来表示仅在局部使用的全局性质的变量，声名方法如下：

```
const  
  MyVar:integer=10;  
  MyVar:=1;
```

Delphi 5 以下版本可以使用没有问题，但从 Delphi 6 开始，默认状态下关闭了此用法，可用编译开关 {\$J+} 打开。

9、几个有用的 ANSI C 和 VC++ 预定义宏

ANSI C 宏	描述
<u>_DATE_</u>	当前源文件的编译时间，格式为： <i>Mmm dd yyyy.</i>
<u>_FILE_</u>	当前源文件名，扩展为双引号引起的文件名 You can create your own wide string version of <u>_FILE_</u> as follows: <pre>#include <stdio.h> #define WIDEN2(x) L ## x #define WIDEN(x) WIDEN2(x) #define _WFILE_ WIDEN(_FILE_) wchar_t *pwsz = _WFILE_; int main() { }</pre>
<u>_LINE_</u>	当前行在当前源文件中的行号。可由 #line 指令修改。
<u>_STDC_</u>	Indicates full conformance with the ANSI C standard. Defined as the integer constant 1 only if the /Za compiler option is given and you are not compiling C++ code; otherwise is undefined.
<u>_TIME_</u>	当前源文件的最近的编译时间，格式 <i>hh:mm:ss.</i>
<u>_TIMESTAMP_</u>	当前源文件的最后修改时间，格式 <i>Ddd Mmm Date hh:mm:ss yyyy</i> , <i>Ddd</i> 是星期 x , <i>Date</i> 是从 1 到 31 的数字。

VC++宏	描述
<u>FUNCTION</u>	返回当前函数名（定义/EP or /P 后无效）

10、如何在自己的程序中使用脚本语言？

微软提供了 msscript 控件以供第三方软件提供商使用脚本语言（jscript, vbscript 等）来增强软件功能和定制能力。

控件共包含以下文件： dispex.dll, jscript.dll, MSSCRIPT.HLP, MSSCRIPT.CNT, msscript.ocx, vbscript.dll。可从 Visual Studio 安装 CD 〈CDRoot\OS\SYSTEM〉 中取得。注册 msscript.ocx, 在自己的工程中添加 msscript 控件即可。具体用法可参见 msdn 搜索 msscript 或参见 MSSCRIPT.HLP。

下面的类 CScriptObject 重新包装了控件的大部分功能，定义如下：

ScriptObject.h

ScriptObject.cpp

```
*****
```

```
* FILENAME : ScriptObject.h
```

```
*
```

```
* class CScriptObject
```

```
*      This class implements MSScript control
```

```
*      It can interface script from resource file or text file
```

```
*
```

```
* INITIAL CODING : Ernest Laurentin (EL)
```

```
*****
```

```
#ifndef _SCRIPT_OBJECT_
```

```
#define _SCRIPT_OBJECT_
```

```
#include <xstring>
```

```
#include <list>
```

```
// Add this to your stdafx.h
```

```
//#pragma warning( disable: 4786)
```

```
// Adjust the following path if required
```

```
#import "msscript.ocx" no_namespace
```

```
#define LANGUAGE_DEFAULT _T("JScript")
```

```
#define RT_SCRIPT _T("SCRIPT")
```

```
#define LANGUAGE_NAME_LEN 40
```

```
#define ERROR_DESC_LEN 256
```

```
using namespace std;
```

```
#ifndef _UNICODE
typedef list<string> stl_string_list;
#else
typedef list<wstring> stl_string_list;
#endif

class CScriptObject
{
    // Construction
public:
    CScriptObject();
    ~CScriptObject();

    // Function
public:
    LPCTSTR GetLanguage();
    void SetLanguage(LPCTSTR szLanguage);
    int GetMethodsCount() const;
    LPCTSTR GetNameAt(int index);
    void Reset();
    bool LoadScript(LPCTSTR szFilename);
    bool LoadScriptResource(LPCTSTR lpName, LPCTSTR lpType, HINSTANCE hInstance);
    bool AddScript(LPCTSTR szCode);
    LPCTSTR GetErrorString();
    bool ExecuteStatement(LPCTSTR szStatement);
    bool RunProcedure(LPCTSTR szProcName, SAFEARRAY** saParameters,
                      VARIANT* varRet);
protected:
    void CommonConstruct();
    bool GetMethodsName();
    LPCTSTR GetScriptFunction(LPCTSTR name);

    IScriptControlPtr m_pScript;           // The one and only
    script control
    stl_string_list m_FunctionList;        // Function list
    TCHAR m_szLanguage[LANGUAGE_NAME_LEN+1]; // Current language
    TCHAR m_szError[ERROR_DESC_LEN+1];      // Description error
private:
};

#endif // _SCRIPT_OBJECT_
```

```
//////////  
/////////  
// File:      ScriptObject.cpp  
// Version:   1.0  
//  
// Author:     Ernest Laurentin  
// E-mail:    elauventin@sympatico.ca  
//  
// This class implements MSScript control  
// It can interface script from resource file or text file  
//  
// This code may be used in compiled form in any way you desire.  
This  
// file may be redistributed unmodified by any means PROVIDING it  
is  
// not sold for profit without the authors written consent, and  
// providing that this notice and the authors name and all  
copyright  
// notices remains intact.  
//  
// An email letting me know how you are using it would be nice as  
well.  
//  
// This file is provided "as is" with no expressed or implied  
warranty.  
// The author accepts no liability for any damage/loss of  
business that  
// this c++ class may cause.  
//  
// Version history  
//  
// 1.0 - Initial release.  
// 1.1 - Bug fixes for VC7 and Unicode  
//////////  
/////////  
#include "stdafx.h"  
#include <comutil.h>  
#include "ScriptObject.h"  
  
//////////  
/////////  
// Construction
```

```
CScriptObject::CScriptObject()
{
CommonConstruct(); // will throw exception if failed
}

CScriptObject::~CScriptObject()
{
// Destroy object- and release
m_pScript = NULL;
}

///////////
// Members

///////////
// CommonConstruct
void CScriptObject::CommonConstruct()
{
HRESULT hr = m_pScript.CreateInstance(_uuidof(ScriptControl));
_com_util::CheckError( hr ); // will throw an exception if failed

// will not come here if exception
_tcscpy(m_szLanguage, LANGUAGE_DEFAULT);
m_pScript->PutAllowUI( VARIANT_FALSE );
m_pScript->PutLanguage( _bstr_t(m_szLanguage) );
}

///////////
// GetLanguage : Get current script language
LPCTSTR CScriptObject::GetLanguage()
{
return m_szLanguage;
}

///////////
// SetLanguage : Set current script language
```

```
void CScriptObject::SetLanguage(LPCTSTR szLanguage)
{
    _tcscpy(m_szLanguage, szLanguage);

    if (m_pScript != NULL)
    {
        m_pScript->PutLanguage(_bstr_t( szLanguage ) );
        m_pScript->Reset();
        m_FunctionList.clear();
    }
}

/////////////////////////////
// GetMethodsCount
int CScriptObject::GetMethodsCount() const
{
    return m_FunctionList.size();
}

/////////////////////////////
// GetNameAt : Get method name at specified index
LPCTSTR CScriptObject::GetNameAt(int index)
{
    if( index >= 0 && index < m_FunctionList.size())
    {
        stl_string_list::iterator iter = m_FunctionList.begin();
        while( index > 0)
        {
            iter++;
            index--;
        }
        return (*iter).c_str();
    }
    return TEXT("");
}

/////////////////////////////
// Reset : Reset script control object
```

```
void CScriptObject::Reset()
{
    if (m_pScript != NULL)
    {
        m_pScript->Reset();

        // empty list...
        m_FunctionList.clear();
    }
}

///////////////////////////////
// GetErrorString : Get Script error string
LPCTSTR CScriptObject::GetErrorString()
{
    m_szError[0] = 0;
    if (m_pScript != NULL)
    {
        try{
            IScriptErrorPtr pError = m_pScript->GetError();
            if (pError != NULL)
            {
                _bstr_t desc = _bstr_t("Error: ") + pError-
>GetDescription() + _bstr_t(", ");
                desc += pError->GetText() + _bstr_t("; in line
");
                desc += _bstr_t(pError->GetLine());
                int count = __min(desc.length(), ERROR_DESC_LEN); // 
string may be truncated...
                _tcsncpy(m_szError, (LPCTSTR) desc, count);
                m_szError[count] = 0;
                pError->Clear();
            }
        }
        catch(_com_error& e)
        {
            TRACE( (LPSTR)e.Description() );
            TRACE( (LPSTR)"\n" );
        }
    }
    return m_szError;
}
```

```
//////////  
/////////  
// GetMethodsName: Get methods name list  
bool CScriptObject::GetMethodsName()  
{  
    bool bResult = false;  
    if (m_pScript != NULL)  
    {  
        IScriptProcedureCollectionPtr pIProcedures = m_pScript->GetProcedures();  
  
        // empty list...  
        m_FunctionList.clear();  
  
        try{  
            long count = pIProcedures->GetCount();  
            for(long index=1; index <= count; index++)  
            {  
                IScriptProcedurePtr pIProcPtr = pIProcedures->GetItem(_variant_t(index));  
                _bstr_t name = pIProcPtr->GetName();  
                m_FunctionList.insert(m_FunctionList.end(), (LPCTSTR)name );  
                pIProcPtr = NULL;  
            }  
  
            bResult = true;  
        }  
        catch(...)  
        {  
            // Just catch the exception, call GetErrorString()  
            // to retreive last error  
        }  
  
        pIProcedures = NULL;  
    }  
    return bResult;  
}
```

11、如何在基于 CHtmlView 的程序中使用 HTML 事件映射？

在基于 CDHTMLDialog 的程序中，我们可以用

```
DECLARE_DHTML_EVENT_MAP  
BEGIN_DHTML_EVENT_MAP  
    DHTML_EVENT()  
END_DHTML_EVENT_MAP
```

宏来相应来自于 HTML 页中的元素的事件，但是在基于 CHtmlView 的程序中却不能使用这个宏，经过对 CDHTMLDialog 的源码分析，可以看到 CDHTMLDialog 派生于 CDHtmlEventSink，是靠 CDHtmlEventSink 来处理事件的，我们可以移植 CDHtmlEventSink 到 CHtmlView 中，并把它绑定到 CHtmlView 的 IwebBroswer 即可。

下面是经过这样改进过的 CHtmlView 被命名为 CDHtmlViewEx，它同时派生于 CHtmlView 和 CDHtmlEventSink；使用时只需将你的基于 CHtmlView 的视改为从 CDHtmlViewEx 派生即可，你就可以使用所有上面所说的宏来处理来自 HTML 页的事件：

```
dhtmlviewex.h  
dhtmlviewex.cpp  
***** dhtmlviewex.h *****  
*****  
#pragma once  
#include "afxdhtml.h"  
  
class CDHtmlViewEx :  
public CDHtmlEventSink,  
public CHtmlView  
{  
private:  
    DWORD m_dwDHtmlEventSinkCookie;  
    CSimpleArray<CDHtmlControlSink *> m_ControlSinks;  
    CSimpleArray<CDHtmlElementEventSink *> m_SinkedElements;  
  
public:  
    CDHtmlViewEx()  
    {  
        m_dwDHtmlEventSinkCookie = 0;  
    };  
protected:  
    virtual void OnDocumentComplete(LPCTSTR lpszURL);
```

```
virtual void OnNavigateComplete2(LPCTSTR strURL);
HRESULT GetDHtmlDocument(IHTMLDocument2** pDoc);
HRESULT ConnectDHtmlEvents(IUnknown *punkDoc);
void DisconnectDHtmlEvents();
HRESULT ConnectDHtmlElementEvents(DWORD_PTR dwThunkOffset = 0);
HRESULT GetElement(LPCTSTR szElementId, IDispatch **ppdisp,
                   BOOL *pbCollection = NULL);
void DisconnectDHtmlElementEvents();
BOOL IsSunkedElement(IDispatch *pdispElem);
BOOL FindSinkForObject(LPCTSTR szName);

//*****************************************************************************
DECLARE_DHTML_EVENT_MAP()
//*************************************************************************/
};

//*****************************************************************************
dhtmlviewex.cpp ****
#include "StdAfx.h"
#include "dhtmlviewex.h"

//*****************************************************************************
BEGIN_DHTML_EVENT_MAP(CDHtmlViewEx)
    DHTML_EVENT_ONCLICK(_T("ButtonOK"), OnButtonOK)
END_DHTML_EVENT_MAP()
//*************************************************************************/

void CDHtmlViewEx::OnDocumentComplete(LPCTSTR lpszURL)
{
    CHtmlView::OnDocumentComplete(lpszURL);
    ConnectDHtmlElementEvents(((DWORD_PTR)static_cast<CDHtmlSinkHandler*>(this)) - (DWORD_PTR)this));
}

void CDHtmlViewEx::OnNavigateComplete2(LPCTSTR strURL)
{
    CHtmlView::OnNavigateComplete2(strURL);
    IDispatch* pdispDoc = GetHtmlDocument();
    ConnectDHtmlEvents(pdispDoc);
    pdispDoc->Release();
}

HRESULT CDHtmlViewEx::GetDHtmlDocument(IHTMLDocument2** pDoc)
{
    HRESULT hr = E_FAIL;
```

```
LPDISPATCH pDisp = GetHtmlDocument();
if(pDisp!=NULL)
{
    hr = pDisp->QueryInterface(_uuidof(IHTMLDocument2), (void**)&pDoc);
}

return hr;
}

HRESULT CDHtmlViewEx::ConnectDHtmlEvents(IUnknown *punkDoc)
{
    return ConnectToConnectionPoint(punkDoc, _uuidof(HTMLDocumentEvents),
&m_dwDHtmlEventSinkCookie);
}

void CDHtmlViewEx::DisconnectDHtmlEvents()
{
    CComPtr<IHTMLDocument2> sphtmlDoc;
    GetDHtmlDocument(&sphtmlDoc);

    if (sphtmlDoc == NULL)
        return;
    DisconnectFromConnectionPoint(sphtmlDoc, _uuidof(HTMLDocumentEvents),
m_dwDHtmlEventSinkCookie);
    DisconnectDHtmlElementEvents();
}

HRESULT CDHtmlViewEx::ConnectDHtmlElementEvents(DWORD_PTR dwThunkOffset /*= 0*/)
{
    HRESULT hr = S_OK;
    const DHtmlEventMapEntry* pEventMap = GetDHtmlEventMap();
    if (!pEventMap)
        return hr;

    for (int i=0; pEventMap[i].nType != DHTMLEVENTMAPENTRY_END; i++)
    {
        if (pEventMap[i].nType==DHTMLEVENTMAPENTRY_ELEMENT)
        {
            // an element name must be specified when using element events
            ASSERT(pEventMap[i].szName);

            // connect to the element's event sink
            CComPtr<IDispatch> spdispElement;
            GetElement(pEventMap[i].szName, &spdispElement);
        }
    }
}
```

```
if (spdispElement)
{
    if (!IsSunkedElement(spd़ispElement))
    {
        CDHtmlElementEventSink *pSink = NULL;
        ATLTRY(pSink = new CDHtmlElementEventSink(this,
spdispElement));
        if (pSink == NULL)
            return E_OUTOFMEMORY;
        hr = AtlAdvise(spd़ispElement, pSink, __uuidof(IDispatch),
&pSink->m_dwCookie);
        if (SUCCEEDED(hr))
            m_SunkedElements.Add(pSink);
        else
            delete pSink;
#endif _DEBUG
        if (FAILED(hr))
            TRACE(traceHtml, 0, "Warning: Failed to connect to
ConnectionPoint!\n");
#endif
    }
}
else if (pEventMap[i].nType==DHTMLEVENTMAPENTRY_CONTROL)
{
    // check if we already have a sink connected to this control
    if (!FindSinkForObject(pEventMap[i].szName))
    {
        // create a new sink and
        // connect it to the element's event sink
        CComPtr<IDispatch> spdispElement;
        GetElement(pEventMap[i].szName, &spdispElement);
        if (spdispElement)
        {
            CComPtr<IHTMLObjectElement> sphtmlObj;
            spdispElement->QueryInterface(__uuidof(IHTMLObjectElement),
(void **) &sphtmlObj);
            if (sphtmlObj)
            {
                CComPtr<IDispatch> spdispControl;
                sphtmlObj->get_object(&spdispControl);
                if (spdispControl)
                {
                    // create a new control sink to connect to the
```

```
control's events
{
    CDHtmlControlSink *pSink = NULL;
    ATLTRY(pSink = new CDHtmlControlSink(spdispControl,
    this, pEventMap[i].szName, dwThunkOffset));
    if (pSink == NULL)
        return E_OUTOFMEMORY;
    m_ControlSinks.Add(pSink);
}
}

{
    CComPtr<IHTMLElementCollection> sphtmlAll;
    CComPtr<IHTMLElementCollection> sphtmlColl;
    CComPtr<IDispatch> spdispElem;
    CComVariant varName;
    CComVariant varIndex;
    HRESULT hr = S_OK;
    CComPtr<IHTMLDocument2> sphtmlDoc;

*ppdisp = NULL;

if (pbCollection)
    *pbCollection = FALSE;

hr = GetDHtmlDocument(&sphtmlDoc);
if (sphtmlDoc == NULL)
    return hr;

if(szElementId == NULL)
    return E_INVALIDARG;

varName.vt = VT_BSTR;
varName.bstrVal = T2BSTR_EX(szElementId);

#ifndef _UNICODE
if (!varName.bstrVal)
```

```

    {
        hr = E_OUTOFMEMORY;
        goto Error;
    }
#endif

    hr = sphtmlDoc->get_all(&sphtmlAll);
    if (sphtmlAll == NULL)
        goto Error;
    hr = sphtmlAll->item(varName, varIndex, &spdispElem);
    if (spdispElem == NULL)
    {
        hr = E_NOINTERFACE;
        goto Error;
    }

    spdispElem->QueryInterface(_uuidof(IHTMLElementCollection), (void **)
&sphtmlColl);
    if (sphtmlColl)
    {
        if (pbCollection)
            *pbCollection = TRUE;
#ifndef _DEBUG
        else
        {
            TRACE(traceHtml, 0, "Warning: duplicate IDs or NAMES. \n");
            ATLASSERT(FALSE);
        }
#endif
    }
#endif

}

Error:
if (SUCCEEDED(hr))
{
    *ppdisp = spdispElem;
    if (spdispElem)
        (*ppdisp)->AddRef();
}
return hr;
}

BOOL CDHtmlViewEx::FindSinkForObject(LPCTSTR szName)
{
    int nLength = m_ControlSinks.GetSize();

```

```
    for (int i=0; i<nLength; i++)
    {
        if (!_tcscmp(szName, m_ControlSinks[i]->m_szControlId))
            return TRUE;
    }
    return FALSE;
}

BOOL CDHtmlViewEx::IsSinkedElement(IDispatch *pdispElem)
{
    CComPtr<IUnknown> spunk;
    pdispElem->QueryInterface(__uuidof(IUnknown), (void **) &spunk);
    if (!spunk)
        return FALSE;
    for (int i=0; i<m_SinkedElements.GetSize(); i++)
    {
        if (spunk == m_SinkedElements[i]->m_spunkElem)
            return TRUE;
    }
    return FALSE;
}

void CDHtmlViewEx::DisconnectDHtmlElementEvents()
{
    const DHtmlEventMapEntry* pEventMap = GetDHtmlEventMap();

    if (!pEventMap)
        return;

    int i;

    // disconnect from element events
    for (i=0; i<m_SinkedElements.GetSize(); i++)
    {
        CDHtmlElementEventSink *pSink = m_SinkedElements[i];
        AtlUnadvise(pSink->m_spunkElem, __uuidof(IDispatch), pSink->m_dwCookie);
        delete pSink;
    }
    m_SinkedElements.RemoveAll();

    // disconnect from control events
    for (i=0; i<m_ControlSinks.GetSize(); i++)
    {
        DisconnectFromConnectionPoint(m_ControlSinks[i]->m_spunkObj,
```

```
    m_ControlSinks[i]->m_iid, m_ControlSinks[i]->m_dwCookie);
    delete m_ControlSinks[i];
}
m_ControlSinks.RemoveAll();
return;
}
```

12、如何得到 Web 页面中的 COM 组件的原始接口？

我们可以通过下面的函数获得 Web 页中任意元素的 IHTMLElement 接口：

```
//为了简单，剔除了所有错误处理代码
void GetElement(IHTMLDocument2* sphtmlDoc ,LPCTSTR szElementId,
IHTMLElement **ppElem)
{
CComVariant varName;
CComVariant varIndex;

varName.vt = VT_BSTR;
varName.bstrVal = T2BSTR_EX(szElementId);

CComPtr<IHTMLElementCollection> sphtmlAll;
sphtmlDoc->get_all(&sphtmlAll);
CComPtr<IDispatch> spdispElem;
sphtmlAll->item(varName, varIndex, &spdispElem);

spdispElem->QueryInterface(_uuidof(IHTMLElement), (void **) & ppElem);
}
```

那么如果已知 Web 元素是一个 COM 组件或 ActiveX 控件，怎么获得组件的原始接口呢，例如，如果页面中有一个 Ado: Recordset 组件，怎么获得 Recordset 的 Recordset21 接口呢：

```
<object classid="clsid:00000535-0000-0010-8000-00AA006D2EA4" id="Recordset">
</object>
```

经过测试发现，只需要利用前一步获得的 IHTMLElement 接口查询 Recordset21 接口就可以了，代码如下：

```
CComPtr<IHTMLElement> sphtmlObj;
GetElement(pDoc, " Recordset" , (void**)&sphtmlObj);
CComPtr<Recordset21> pRec;
sphtmlObj ->QueryInterface(_uuidof(Recordset21), (void **) & pRec);
//pRec->Open("1. xml", *****);
```

13、如何生成全球唯一标识（UID）？

我们可以通过下面的函数：

```
#include "Nb30.h"  
#pragma comment(lib,"Netapi32.lib")
```

```
/*****************************************************************************
```

为了节省时间值转为字串后的长度，采用此结构存放时间
此结构共占64bit（位），总有效位数49bit（位）各部分所占位数如下图示：

```
ULLONG nSecond:6;  
ULLONG nMinute:6;  
ULLONG nHour:5;  
ULLONG nDay:5;  
ULLONG nMonth:4;  
ULLONG nYear:14;  
ULLONG nStepID:9; //自增序号
```

排列顺序如下(64 Bit)：

高位	低位
nYear nMonth nDay nHour nMinute nSecond nStepID	
14 4 5 5 6 6 9	

转为字串后，最大串长15字节

```
*****  
ULLONG MakeTimeStamp(ULLONG nYear,  
                      ULLONG nMonth,  
                      ULLONG nDay,  
                      ULLONG nHour,  
                      ULLONG nMinute,  
                      ULLONG nSecond,  
                      ULLONG nStepID) //自增序号  
  
{  
    ASSERT(nSecond<=0x3F);  
    ASSERT(nMinute<=0x3F);  
    ASSERT(nHour<=0x1F);  
    ASSERT(nDay<=0x1F);  
    ASSERT(nMonth<=0xF);  
    ASSERT(nYear<=0x3FFF);  
    ASSERT(nStepID<=0x1FF);
```

```
    return ((nYear&0x3FF)<<35) +
        ((nMonth&0xF)<<31) +
        ((nDay&0x1F)<<26) +
        ((nHour&0x1F)<<21) +
        ((nMinute&0x3F)<<15) +
        ((nSecond&0x3F)<<9) +
        nStepID&0x1FF;
}

CString GetMacAddress();
CString GetNewUID(CString strPrefix)
{
    ASSERT(strPrefix.GetLength()<=20);

    //为了提高效率，仅第一次取Mac地址
    static WORD nStepID=0;      //自增序号
    static CString aMacAddr;   //网卡号
    if(nStepID==0)
        aMacAddr=GetMacAddress(); //获取网卡号
    else if(nStepID==511)
        nStepID=0;

    CString Result='.'+aMacAddr+'.'; //Result最长17位
    ASSERT(Result.GetLength()<=17);

    Result+=IntToStr(GetCurrentProcessId())+'.'; //Result最长28位
    ASSERT(Result.GetLength()<=28);

    COleDateTime DT=COleDateTime::GetCurrentTime();
    ULONGLONG ullTimeStamp=MakeTimeStamp(DT.GetYear(), //14900年不重复
                                         DT.GetMonth(),
                                         DT.GetDay(),
                                         DT.GetHour(),
                                         DT.GetMinute(),
                                         DT.GetSecond(),
                                         nStepID);

    Result+=Int64ToStr(ullTimeStamp); //Result最长43位
    ASSERT(Result.GetLength()<=43);

    nStepID++;

    ASSERT((strPrefix+Result).GetLength()<=63);
```

```
        return strPrefix+Result;//Result最长63位
    }

    CString GetMacAddress()
    {
        char aPart[8];
        memset(aPart, 0, 8);

        //枚举Lana
        NCB ncb;
        memset(&ncb, 0, sizeof(NCB));
        ncb.ncb_command = char(NCBENUM);
        LANA_ENUM lanaEnum;
        ncb.ncb_buffer = (PUCHAR)(&lanaEnum);
        ncb.ncb_length = sizeof(LANA_ENUM);
        if(Netbios(&ncb) == NRC_GOODRET)
        {
            //重置第一个Lana
            memset(&ncb, 0, sizeof(NCB));
            ncb.ncb_command = NCBRESET;
            ncb.ncb_lana_num = lanaEnum.lana[0];
            if(Netbios(&ncb) == NRC_GOODRET)
            {
                //获取第一个Lana的适配器信息
                memset(&ncb, 0, sizeof(NCB));
                ncb.ncb_command = NCBASTAT;
                ncb.ncb_lana_num = lanaEnum.lana[0];
                strcpy((char*)&ncb.ncb_callname[0], "*");
                ADAPTER_STATUS adapt;
                ncb.ncb_buffer = PUCHAR(&adapt);
                ncb.ncb_length = sizeof(ADAPTER_STATUS);
                if(Netbios(&ncb) == NRC_GOODRET)
                {
                    //获取Mac地址到pPart
                    ZeroMemory(aPart, 8);
                    memcpy(aPart, &adapt.adapter_address[0], 6);
                }
            }
        }

        return Int64ToStr(*(LONGLONG*)(&aPart));
    }
}
```

14、如何重写 MFC 的缺省控件容器(I0leContainer)?

缺省情况下，MFC 用 COleControlContainer 实现控件容器对象，如果需要改变它的缺省行为则必须重写 COleControlContainer，可按如下步骤：

1. 创建一个派生于 COleControlContainer 的类，因 COleControlContainer 没有缺省构造器所以你的类的构造器必须直接调用 COleControlContainer 的构造器

```
CMyOleControlContainer::CMyOleControlContainer(CWnd* pWnd) :  
    COleControlContainer(pWnd)  
{  
}
```

2. 创建一个派生于 C0ccManager 类。添加空构造器和虚的空析构器。

3. 重载 C0ccManager::CreateContainer 函数：

```
virtual COleControlContainer* CreateContainer(CWnd* pWnd)  
{  
    // Advanced control container applications may want to override.  
    return new CMyOleControlContainer(pWnd);  
}
```

4. 修改对 AfxEnableControlContainer 的调用（一般在 CYourApp::InitInstance），传入 C0ccManager 派生类的实例：

```
CMy0ccManager theManager;
```

```
// ...
```

```
BOOL CMfcaxscrvbApp::InitInstance()  
{  
    // Initialize OLE libraries  
    if (!AfxOleInit())  
        // ...  
    AfxEnableControlContainer(&theManager);
```

注意：对于 CHtmlView 的派生类，因为它的 Create 函数会调用 AfxEnableControlContainer() 这会导致我们的 C0ccManager 定制被冲掉，所以必须重载 CHtmlView::Create 并去掉对 AfxEnableControlContainer() 的调用。

5. 在你的 Cmy0ccManager, CMyOleControlContainer 实现文件里包含 Occimpl.h。

-
6. 在你的 CMyOleControlContainer 头文件里包含 AFX_DATA:

```
#undef AFX_DATA  
#define AFX_DATA AFX_DATA_IMPORT  
#include "c:\program files\devstudio\vc\mfc\src\occimpl.h"  
#undef AFX_DATA  
#define AFX_DATA AFX_DATA_EXPORT
```

15、如何重写 WebBrowser 控件的缺省安全处理功能？

CDHtmlDialog 和 CHtmlView 均使用 WebBrowser 控件进行服务，他们作为 WebBrowser 控件的控件容器，为控件提供安全处理等功能，控件通过容器的 IServiceProvider 接口方法 QueryService 查阅安全管理器接口（IInternetSecurityManager）的指针，然后调用安全管理器接口的 ProcessUrlAction 方法，如果返回 S_OK 则代表验证通过。我们可以重载安全管理器的这个方法以实现自定义的安全管理，请注意：如果你的实现不好的话，可能会引起严重的安全问题。具体步骤如下：

首先，重写 ColeControlSite 控件站点的默认实现，从 ColeControlSite 派生一个类，并实现 IServiceProvider 接口和 IInternetSecurityManager 接口：

```
*****  
myconsite.h  
*****  
#pragma once  
  
template <class T>  
class CMyControlSite:public T  
{  
public:  
    CMyControlSite(COLECONTROLCONTAINER* pContainer) : T(pContainer)  
    {  
    }  
    //IServiceProvider  
    BEGIN_INTERFACE_PART(IServiceProvider, IServiceProvider)  
        INIT_INTERFACE_PART(T, ServiceProvider)  
        STDMETHOD(QueryService)(REFGUID guidService, REFIID riid, void **ppv);  
    END_INTERFACE_PART(ServiceProvider)  
  
    //IInternetSecurityManager  
    BEGIN_INTERFACE_PART(IInternetSecurityManager, IInternetSecurityManager)  
        INIT_INTERFACE_PART(T, InternetSecurityManager)  
        STDMETHOD(SetSecuritySite)(IInternetSecurityMgrSite *pSite);  
        STDMETHOD(GetSecuritySite)(IInternetSecurityMgrSite **ppSite);  
        STDMETHOD(MapUrlToZone)(LPCWSTR pwszUrl, DWORD *pdwZone, DWORD dwFlags);  
        STDMETHOD(GetSecurityId)(LPCWSTR pwszUrl, BYTE *pbSecurityId,  
                               DWORD *pcbSecurityId,
```

```

        DWORD_PTR dwReserved) ;
    STDMETHOD(ProcessUrlAction) (
        /* [in] */ LPCWSTR pwszUrl,
        /* [in] */ DWORD dwAction,
        /* [size_is][out] */ BYTE *pPolicy,
        /* [in] */ DWORD cbPolicy,
        /* [in] */ BYTE *pContext,
        /* [in] */ DWORD cbContext,
        /* [in] */ DWORD dwFlags,
        /* [in] */ DWORD dwReserved) ;
    STDMETHOD(QueryCustomPolicy) (
        /* [in] */ LPCWSTR pwszUrl,
        /* [in] */ REFGUID guidKey,
        /* [size_is][size_is][out] */ BYTE **ppPolicy,
        /* [out] */ DWORD *pcbPolicy,
        /* [in] */ BYTE *pContext,
        /* [in] */ DWORD cbContext,
        /* [in] */ DWORD dwReserved) ;
    STDMETHOD(SetZoneMapping) (
        /* [in] */ DWORD dwZone,
        /* [in] */ LPCWSTR lpszPattern,
        /* [in] */ DWORD dwFlags) ;
    STDMETHOD(GetZoneMappings) (
        /* [in] */ DWORD dwZone,
        /* [out] */ IEnumString **ppenumString,
        /* [in] */ DWORD dwFlags) ;
END_INTERFACE_PART(InternetSecurityManager)

private:
    static const AFX_INTERFACEMAP_ENTRY _interfaceEntries[];
protected:
    static const AFX_INTERFACEMAP interfaceMap;
    virtual const AFX_INTERFACEMAP* GetInterfaceMap() const
    {
        return &CMyControlSite<T>::interfaceMap;
    }
};

template <class T>
AFX_COMDAT const AFX_INTERFACEMAP CMyControlSite<T>::interfaceMap =
{ &T::GetThisInterfaceMap,
    &CMyControlSite<T>::_interfaceEntries[0],
};

```

```
template <class T>
AFX_COMDAT const AFX_INTERFACEMAP_ENTRY CMyControlSite<T>::_interfaceEntries[]
=
{
    INTERFACE_PART(CMyControlSite<T>, IID_IServiceProvider, ServiceProvider)
    INTERFACE_PART(CMyControlSite<T>, IID_IInternetSecurityManager,
InternetSecurityManager)

    { NULL, (size_t)-1 }
};

template <class T>
STDMETHODIMP_(ULONG) CMyControlSite<T>::XInternetSecurityManager::AddRef()
{
    METHOD_PROLOGUE_EX_(CMyControlSite, InternetSecurityManager)
    return (ULONG)pThis->ExternalAddRef();
}

template <class T>
STDMETHODIMP_(ULONG) CMyControlSite<T>::XInternetSecurityManager::Release()
{
    METHOD_PROLOGUE_EX_(CMyControlSite, InternetSecurityManager)
    return (ULONG)pThis->ExternalRelease();
}

template <class T>
STDMETHODIMP CMyControlSite<T>::XInternetSecurityManager::QueryInterface(
REFIID iid, LPVOID* ppvObj)
{
    METHOD_PROLOGUE_EX_(CMyControlSite, InternetSecurityManager)
    return (HRESULT)pThis->ExternalQueryInterface(&iid, ppvObj);
}

template <class T>
STDMETHODIMP CMyControlSite<T>::XServiceProvider::QueryService(REFGUID
guidService, REFIID riid, void **ppv)
{
    *ppv = NULL;
    HRESULT hr = E_NOINTERFACE;
    if((guidService==SID_SInternetSecurityManager)&&(riid==IID_IInternetSecurityManager))
    {
        METHOD_PROLOGUE(CMyControlSite, ServiceProvider);
        hr = (HRESULT)pThis->ExternalQueryInterface(&riid, ppv);
    }
}
```

```
}

    return hr;
}

template <class T>
STDMETHODIMP_(ULONG) CMyControlSite<T>::XServiceProvider::AddRef()
{
    METHOD_PROLOGUE_EX_(CMyControlSite, ServiceProvider)
    return (ULONG)pThis->ExternalAddRef();
}

template <class T>
STDMETHODIMP_(ULONG) CMyControlSite<T>::XServiceProvider::Release()
{
    METHOD_PROLOGUE_EX_(CMyControlSite, ServiceProvider)
    return (ULONG)pThis->ExternalRelease();
}

template <class T>
STDMETHODIMP CMyControlSite<T>::XServiceProvider::QueryInterface(
REFIID iid, LPVOID* ppvObj)
{
    METHOD_PROLOGUE_EX_(CMyControlSite, ServiceProvider)
    return (HRESULT)pThis->ExternalQueryInterface(&iid, ppvObj);
}

template <class T>
STDMETHODIMP
CMyControlSite<T>::XIInternetSecurityManager::SetSecuritySite(IInternetSecurityM
grSite *pSite)
{
    return INET_E_DEFAULT_ACTION;
};

template <class T>
STDMETHODIMP
CMyControlSite<T>::XIInternetSecurityManager::GetSecuritySite(IInternetSecurityM
grSite **ppSite)
{
    return INET_E_DEFAULT_ACTION;
};

template <class T>
```

```
STDMETHODIMP
CMyControlSite<T>::XInternetSecurityManager::MapUrlToZone(LPCWSTR pwszUrl, DWORD
*pdwZone, DWORD dwFlags)
{
    return INET_E_DEFAULT_ACTION;
};

template <class T>
STDMETHODIMP
CMyControlSite<T>::XInternetSecurityManager::GetSecurityId(LPCWSTR pwszUrl, BYTE
*pbSecurityId,
    DWORD *pcbSecurityId,
    DWORD_PTR dwReserved)
{
    return INET_E_DEFAULT_ACTION;
};

template <class T>
STDMETHODIMP CMyControlSite<T>::XInternetSecurityManager::ProcessUrlAction(
/* [in] */ LPCWSTR pwszUrl,
/* [in] */ DWORD dwAction,
/* [size_is][out] */ BYTE *pPolicy,
/* [in] */ DWORD cbPolicy,
/* [in] */ BYTE *pContext,
/* [in] */ DWORD cbContext,
/* [in] */ DWORD dwFlags,
/* [in] */ DWORD dwReserved)
{
    return S_OK;
};

template <class T>
STDMETHODIMP
CMyControlSite<T>::XInternetSecurityManager::QueryCustomPolicy(
/* [in] */ LPCWSTR pwszUrl,
/* [in] */ REFGUID guidKey,
/* [size_is][size_is][out] */ BYTE **ppPolicy,
/* [out] */ DWORD *pcbPolicy,
/* [in] */ BYTE *pContext,
/* [in] */ DWORD cbContext,
/* [in] */ DWORD dwReserved)
{
    return INET_E_DEFAULT_ACTION;
};
```

```

template <class T>
STDMETHODIMP CMyControlSite<T>::XInternetSecurityManager::SetZoneMapping(
/* [in] */ DWORD dwZone,
/* [in] */ LPCWSTR lpszPattern,
/* [in] */ DWORD dwFlags)
{
    return INET_E_DEFAULT_ACTION;
};

template <class T>
STDMETHODIMP CMyControlSite<T>::XInternetSecurityManager::GetZoneMappings(
/* [in] */ DWORD dwZone,
/* [out] */ IEnumString **ppenumString,
/* [in] */ DWORD dwFlags)
{
    return INET_E_DEFAULT_ACTION;
};

```

然后添加接口映射，方法是在你的控件站点的头文件里添加：
DECLARE_INTERFACE_MAP();

在实现文件里添加：

```

BEGIN_INTERFACE_MAP(CMyControlSite, COleControlSite)
    INTERFACE_PART(CMyControlSite, IID_IServiceProvider, ServiceProvider)
    INTERFACE_PART(CMyControlSite, IID_IInternetSecurityManager,
        InternetSecurityManager)
END_INTERFACE_MAP()

```

请注意：本例并未采用这种方法，因为采用了模版类，而是直接将这些宏展开，并直接修改_interfaceEntries函数。

然后应用你的控件站点，方法是重载你的 CDHtmlDialog 的派生类的 CreateControlSite，实例化你的控件站点。

```

virtual BOOL CreateControlSite(COleControlContainer* pContainer,
    COleControlSite** ppSite, UINT nID, REFLSID clsid)
{
    CMyControlSite<COleControlSite>* pSite = new
    CMyControlSite<COleControlSite>(pContainer);
    *ppSite = pSite;
    return TRUE;
}

```

请注意：对于 CHtmlView 的派生类，重载此方法时，不能直接实例化 COleControlSite，因为 CHtmlView 提供了自己的派生于 COleControlSite 的控件站点类 CHtmlControlSite，应该实例化为 CHtmlControlSite，你的重载函数应有如下形式：

```
virtual BOOL CreateControlSite(COleControlContainer* pContainer,
```

```
    COleControlSite** ppSite, UINT nID, REFCLSID clsid)
{
    CMyControlSite<CHtmlControlSite>* pSite = new
    CMyControlSite<CHtmlControlSite>(pContainer);
    *ppSite = pSite;
    return TRUE;
}

但 MFC 并未将 CHtmlControlSite 导出，所以你应该重写 CHtmlControlSite 类的声明，  
但不能写实现，最好是将 MFC 源码中的 CHtmlControlSite 类声明直接复制过来。对于  
VC++.net 2003 CHtmlControlSite 类的声明如下：  
class CHtmlControlSite : public COleControlSite  
{  
public:  
    CHtmlControlSite(COleControlContainer* pParentWnd);  
    ~CHtmlControlSite();  
  
    CHtmlView* GetView() const;  
  
    BEGIN_INTERFACE_PART(DocHostUIHandler, IDocHostUIHandler)  
        STDMETHOD(ShowContextMenu) (DWORD, LPPPOINT, LPUNKNOWN, LPDISPATCH);  
        STDMETHOD(GetHostInfo) (DOCHOSTUIINFO*);  
        STDMETHOD(ShowUI) (DWORD, LPOLEINPLACEACTIVEOBJECT,  
                          LPOLECOMMANDTARGET, LPOLEINPLACEFRAME, LPOLEINPLACEUIWINDOW);  
        STDMETHOD(HideUI) (void);  
        STDMETHOD(UpdateUI) (void);  
        STDMETHOD(EnableModeless) (BOOL);  
        STDMETHOD(OnDocWindowActivate) (BOOL);  
        STDMETHOD(OnFrameWindowActivate) (BOOL);  
        STDMETHOD(ResizeBorder) (LPCRECT, LPOLEINPLACEUIWINDOW, BOOL);  
        STDMETHOD(TranslateAccelerator) (LPMMSG, const GUID*, DWORD);  
        STDMETHOD(GetOptionKeyPath) (OLECHAR **, DWORD);  
        STDMETHOD(GetDropTarget) (LPDROPTARGET, LPDROPTARGET*);  
        STDMETHOD(GetExternal) (LPDISPATCH*);  
        STDMETHOD(TranslateUrl) (DWORD, OLECHAR*, OLECHAR **);  
        STDMETHOD(FilterDataObject) (LPDATAOBJECT, LPDATAOBJECT*);  
    END_INTERFACE_PART(DocHostUIHandler)  
  
    DECLARE_INTERFACE_MAP()
};
```

16、如何重写 WebBrowser 控件的缺省下载管理器？

CDHtmlDialog 和 CHtmlView 均使用 WebBrowser 控件进行服务，他们作为 WebBrowser

控件的控件容器，为控件提供安全处理等功能，控件通过容器的 IServiceProvider 接口方法 QueryService 查阅下载管理器接口（IDownloadManager）的指针，然后调用下载管理器接口的 Download 方法。我们可以重载下载管理器的这个方法以实现自定义的下载管理。具体步骤与 [15：如何重写 WebBrowser 控件的缺省安全处理功能](#) 相同。

下面仅列出不同部分：

首先，实现你的下载管理器组件，需要实现 IUnknown 和 IDownloadManager 接口。

然后，实现 IServiceProvider::QueryService，

STDMETHODIMP XServiceProvider::QueryService(REFGUID guidService,

 REFIID riid,

 void **ppv)

{

 HRESULT hr = E_NOINTERFACE;

 if (guidService == SID_SDownloadManager && riid == IID_IDownloadManager)

 {

 // Create new CDownloadMgr object using ATL.

 CComObject<CDownloadMgr>* pDownloadMgr;

 hr = CComObject<CDownloadMgr>::CreateInstance(&pDownloadMgr);

 // Query the new CDownloadMgr object for IDownloadManager interface.

 hr = pDownloadMgr->QueryInterface(IID_IDownloadManager, ppv);

 }

 return hr;

}

如果你想让你的定制下载管理器也能为 Internet Explorer 6 或更高版本所用，请在下面注册表位置填入你的下载管理器的 CLSID。

HKEY_LOCAL_MACHINE
 Software
 Microsoft
 Internet Explorer
 DownloadUI
HKEY_CURRENT_USER
 Software
 Microsoft
 Internet Explorer
 DownloadUI

另可参阅：<http://support.microsoft.com/default.aspx?scid=kb;en-us;327865>

17、如何定制 IE 或 WebBrowser 控件的缺省打印和打印预览框？

<http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/hosting/printpreview/reference/reference.asp>

18、如何扩展 IE 或 WebBrowser 控件提供的文档对象模型（DOM）？

先决条件和要求

为了能够理解和使用这个指南，你需要：

熟悉VC++和COM

精通ATL

安装了IE 6或更高版本

用于Internet Explorer 6 或更高的组件开发头文件和库；详细的说，你需要 Mshtmhsht.h.

许多定制特性可以在Internet Explorer 5 和Internet Explorer 5.5. Only a couple require Internet Explorer 6. 检查参考文档找到版本信息。

介绍

集成WebBrowser控件是一个进行快速应用开发的强有力的工具。通过集成使用易用的技术—Dynamic HTML (DHTML), HTML, and Extensible Markup Language (XML)一显示内容和开发UI. 然而，“out of the box,” WebBrowser控件可能不能完全满足你的要求。例如：在缺省情况下，用户能通过右键菜单，显示当前显示页面的源码。你可能想要禁用它或去除这个选项。你可能也想做其它比如用你自己的定制菜单替换缺省的快键菜单。

Aside from the customizations just mentioned, the advanced hosting features

of Internet Explorer enable you to:

除了刚才提到的定制外, Internet Explorer高级集成特性允许你:

用页面上的按钮或其他控件调用你的应用程序的内部方法, 有效的扩展DHTML对象模型.

改变拖放方式.

控制你的应用程序能访问那一页, 限制浏览, 例如: 规定页面, 域, 或站点.

截取用户击键并按你想要的方式处理它. 例如: 你可能想要截取CTRL+O来防止用户从Internet Explorer打开新的页面而不是宿主应用程序。

改变缺省字体和显示设置

控制欲下载的内容的类型, 和WebBrowser控件的缺省处理方式. 例如: 你可以阻止视频播放, 脚本运行, 用户单击连接时新窗口的打开, 或Microsoft ActiveX®控件的下载和执行。

限制显示源码。

捕捉搜索

捕捉浏览错误

Replace or modify shortcut menus and shortcut menu entries—disabling, replacing, customizing, or adding to them.

改变你的应用程序的注册设置

截获和改变WebBroswer控件打开的消息框

控制新窗口的创建

在下面的部分, 我们能看到许多--尽管不是所有--这些可能及其实现方法

WebBrowser定制化结构

介绍IDocHostUIHandler, IDocHostUIHandler2, IDocHostShowUI, and ICustomDoc三个接口是WebBrowser控件UI定制的核心: IDocHostUIHandler, IDocHostUIHandler2, and IDocHostShowUI. 也是你想要修改WebBrowser控件时要实现的接口. There are also a couple of service interfaces. ICustomDoc由MSHTML实现, 用于在某些情况下允许WebBrowser控件定制。IHostDialogHelper用于打开信任的对话框, without identification that marks them as Internet Explorer dialog boxes.

除此之外, 还用两件事可做。一个是: 通过在你的IDispatch实现内截取内嵌属性变化,

你能控制下载。第二，通过在你的IDispatch实现内截取DISPID_NEWWINDOW2，你能控制新窗口的打开。

How It Works

The mechanism for WebBrowser Control customization is designed to be automated when a container provides support for ActiveX controls. 当WebBrowser控件实例化的时候，他将尝试从主机找到IDocHostUIHandler, IDocHostUIHandler2 and IDocHostShowUI. The WebBrowser Control 通过在宿主的I0leClientSite上用QueryInterface查询这些接口。

对于一个实现了I0leClientSite接口并通过I0leObject::SetClientSite方法赋予I0leClientSite指针的应用来说，这个架构将会自动运转。典型的实现如下：

```
// Error checking omitted for clarity  
CComPtr<I0leObject> sp0le0bj;  
  
// Create WebBrowser--store pointer in class member variable m_spWebBrowser  
CoCreateInstance(CLSID_WebBrowser, NULL, CLSCTX_INPROC, IID_IWebBrowser2,  
(void**)&m_spWebBrowser);  
  
// Query WebBrowser for I0leObject pointer  
m_spWebBrowser->QueryInterface(IID_I0leObject, (void**)&sp0le0bj);  
  
// Set client site  
sp0le0bj->SetClientSite(this);  
  
// In-place activate the WebBrowser control  
RECT rcClient  
GetClientRect(&rcClient);  
sp0le0bj->DoVerb(OLEIVERB_INPLACEACTIVATE, NULL, this, 0,
```

```
GetTopLevelWindow(), &rcClient);

// Register container to intercept WebBrowser events
AtlAdvise(m_spWebBrowser, GetUnknown(), IID_DWebBrowserEvents2,
&m_dwCookie);

// Navigate to start page
m_spWebBrowser->Navigate(L"res://webhost.exe/startpage.htm", NULL, NULL,
NULL, NULL);
```

然而, 如果你的应用并未提供I0leClientSite, 你也不要失望. Internet Explorer 提供了ICustomDoc接口, 你可以通过它向Internet Explorer传送你的IDocHostUIHandler接口. 但在这种情况下, 你不能用IDocHostUIHandler2 and IDocHostShowUI。

当WebBrowser控件处理这些接口指针的时候, 相应的接口方法将会在控件的生命器内合适的时间被调用。例如: 当用户在控件的客户区域内点击鼠标右键时, 你的IDocHostUIHandler::ShowContextMenu将在IE显示菜单前被调用。这就给你一个显示自己菜单的机会。

这有几个问题要注意, 你的应用必须用OleInitialize而不是CoInitialize来启动COM. OleInitialize提供剪贴板, 拖放, OLE, 就地激活特性。用OleUninitialize关闭COM库。

ATL COM向导用CoInitialize 而不是OleInitialize 来打开COM 库, 所以如果你用向导, 你就需要将CoInitialize 和CoUninitialize 改成OleInitialize和 OleUninitialize. 对于MFC应用, 请确信调用了AfxOleInit, 他会调用OleInitialize。

如果你不想要拖放支持, 你可以调用IWebBrowser2::RegisterAsDropTarget传递 VARIANT_TRUE, 阻止拖放操作。

集成WebBrowser Control的应用也需要实现I0leInPlaceSite, I0leInPlaceSite从

I_{Ole}Window派生，应用需要实现I_{Ole}Window。 You need these implementations so that your application has a window in which to display the WebBrowser Control and so you can manage its display.

这些接口和I_{Ole}ClientSite接口可以最小化实现，甚至可以部分实现。 I_{Ole}ClientSite方法可返回E_NOTIMPL。 Some of the I_{Ole}InPlaceSite and I_{Ole}Window methods need an implementation beyond their return value. See the code sample for an example of a minimal implementation of I_{Ole}InPlaceSite and I_{Ole}Window.

Now that we have covered the initialization preliminaries, let's take a look at each of the interfaces for WebBrowser Control customization.

IDocHostUIHandler

IDocHostUIHandler在Internet Explorer 5以后可用。他提供15个方法。通常，更重要的方法是 IDocHostUIHandler::GetExternal, IDocHostUIHandler::GetHostInfo, IDocHostUIHandler::GetOptionKeyPath, IDocHostUIHandler::ShowContextMenu, and IDocHostUIHandler::TranslateAccelerator。当然，那个方法重要要视你的应用而定。

IDocHostUIHandler::GetHostInfo

你能用IDocHostUIHandler::GetHostInfo方法告诉MSHTML你的应用的能力和要求。通过它你能控制各种东西，例如：

You can disable the browser's 3-D border.

You can prevent scroll bars or change their appearance.

You can disable script.

You can define how you want to handle double-clicks.

IDocHostUIHandler::GetHostInfo有一个参数，一个指向DOCHOSTUIINFO结构的指针，由MSHTML分配存储。你的任务就是把你要告诉给MSHTML的信息填入。

DOCHOSTUIINFO结构中有五个成员，分别为

cbSize, 定义结构大小, 你应该按下面代码填入他.
dwFlags, 可以是DOCHOSTUIFLAG枚举值的任意组合 (OR)
dwDoubleClick, 可取DOCHOSTUIDBLCLK枚举的任一值.
pchHostCss. 你可以提供一个CSS样式表串的指针, 这些样式将应用于在控件里显示的任一页。

pchHostNs. 你可以提供一个指向双冒号分割的名空间串的指针, 当你用定制tags时会用到。This way you can have a global declaration for namespaces and don't have to declare them on each displayed page.

请确保使用CoTaskMemAlloc分配pchHostCss或pchHostNS的内存。

```
HRESULT GetHostInfo(DOCHOSTUIINFO *pInfo)
{
    WCHAR* szCSS = L"BODY {background-color:#ffcccc}";
    WCHAR* szNS = L"IE;MyTags;MyTags2='www.microsoft.com'";

#define CCHMAX 256
    size_t cchLengthCSS, cchLengthszNS;

    HRESULT hr = StringCchLengthW(szCSS, CCHMAX, &cchLengthCSS);
    // TODO: Add error handling code here.

    OLECHAR* pCSSBuffer = (OLECHAR*)CoTaskMemAlloc((cchLengthCSS + 1) *
sizeof(OLECHAR));
    // TODO: Add error handling code to make sure memory block was
allocated successfully.

    hr = StringCchLengthW(szNS, CCHMAX, &cchLengthszNS);
    // TODO: Add error handling code here.

    OLECHAR* pNSBuffer = (OLECHAR*)CoTaskMemAlloc((cchLengthszNS + 1) *
sizeof(OLECHAR));
```

```
// TODO: Add error handling code to make sure memory block was
allocated successfully.

hr = StringCchCopyW(pCSSBuffer, cchLengthCSS + 1, szCSS);

// TODO: Add error handling code here.

hr = StringCchCopyW(pNSBuffer, cchLengthszNS + 1, szNS);

// TODO: Add error handling code here.

pInfo->cbSize = sizeof(DOCHOSTUIINFO);

pInfo->dwFlags = DOCHOSTUIFLAG_NO3DBORDER | DOCHOSTUIFLAG_SCROLL_NO;

pInfo->dwDoubleClick = DOCHOSTUIDBLCLK_DEFAULT;

pInfo->pchHostCss = pCSSBuffer;

pInfo->pchHostNS = pNSBuffer;

return S_OK;
}
```

如果你没有需要告诉MSHTML的悄悄话，你可以直接返回E_NOTIMPL

IDocHostUIHandler::ShowContextMenu

通过实现本方法，你能得到右键菜单的控制权。你能阻止Internet Explorer显示他的默认菜单，直接返回S_OK。返回其他值，如：S_FALSE或E_NOTIMPL，可以让Internet Explorer继续显示他的默认菜单。

If you return S_OK from this method and nothing more, you can prevent any right-click behavior by the WebBrowser control. This may be all you desire in many scenarios but you can do more. Often, you use this method to create and display your own shortcut menu before returning S_OK. 如果你知道默认的开始菜单资源的位置，直到他如何被加载，你也可以直接修改默认菜单。让我们看一看该怎么做。

WebBrowser Control从Shdoclc.dll得到他的菜单资源。That knowledge and a few

#defines gives you a chance to manipulate the browser's menu. 我们假设你希望保留默认菜单但View Source菜单除外, 你想去除它. 下面的代码从Shdoclc. d11加载菜单, 然后选择正确的菜单, 删除相应于IDM_VIEWSOURCE命令的菜单项, 然后显示菜单。

```
HRESULT CBrowserHost::ShowContextMenu(DWORD dwID,
                                         POINT *ppt,
                                         IUnknown *pcmdTarget,
                                         IDispatch *pdspObject)

{

#define IDR_BROWSE_CONTEXT_MENU 24641
#define IDR_FORM_CONTEXT_MENU    24640
#define SHDVID_GETMIMECSETMENU 27
#define SHDVID_ADDMENUEXTENSIONS 53

HRESULT hr;
HINSTANCE hinstSHDOCLC;
HWND hwnd;
HMENU hMenu;
CComPtr<IOleCommandTarget> spCT;
CComPtr<IOleWindow> spWnd;
MENUITEMINFO mii = {0};
CComVariant var, var1, var2;

hr = pcmdTarget->QueryInterface(IID_IOleCommandTarget, (void**)&spCT);
hr = pcmdTarget->QueryInterface(IID_IOleWindow, (void**)&spWnd);
hr = spWnd->GetWindow(&hwnd);

hinstSHDOCLC = LoadLibrary(TEXT("SHDOCLC.DLL"));

if (hinstSHDOCLC == NULL)
```

```
{  
    // Error loading module -- fail as securely as possible  
    return;  
}  
  
hMenu = LoadMenu(hinstSHDOCLC,  
    MAKEINTRESOURCE(IDR_BROWSE_CONTEXT_MENU));  
  
hMenu = GetSubMenu(hMenu, dwID);  
  
// Get the language submenu  
hr = spCT->Exec(&CGID_ShellDocView, SHDVID_GETMIMESETMENU, 0, NULL,  
&var);  
  
mii.cbSize = sizeof(mii);  
mii.fMask = MIIM_SUBMENU;  
mii.hSubMenu = (HMENU) var.byref;  
  
// Add language submenu to Encoding context item  
SetMenuItemInfo(hMenu, IDM_LANGUAGE, FALSE, &mii);  
  
// Insert Shortcut Menu Extensions from registry  
V_VT(&var1) = VT_INT_PTR;  
V_BYREF(&var1) = hMenu;  
  
V_VT(&var2) = VT_I4;  
V_I4(&var2) = dwID;  
  
hr = spCT->Exec(&CGID_ShellDocView, SHDVID_ADDMENUEXTENSIONS, 0, &var1,  
&var2);
```

```
// Remove View Source

DeleteMenu(hMenu, IDM_VIEWSOURCE, MF_BYCOMMAND);

// Show shortcut menu

int iSelection = ::TrackPopupMenu(hMenu,
                                    TPM_LEFTALIGN | TPM_RIGHTBUTTON |
                                    TPM_RETURNCMD,
                                    ppt->x,
                                    ppt->y,
                                    0,
                                    hwnd,
                                    (RECT*)NULL);

// Send selected shortcut menu item command to shell

LRESULT lr = ::SendMessage(hwnd, WM_COMMAND, iSelection, NULL);

FreeLibrary(hinstSHDOCLC);

return S_OK;
}
```

安全警告 调用LoadLibrary不慎重可能会危及到你的应用的安全（可能装入不正确的 DLL文件）。Refer to the LoadLibrary documentation for information on how to correctly load DLLs with different versions of Microsoft Windows®.

IDocHostUIHandler::GetExternal: 扩展DOM

IDocHostUIHandler 用来扩展Internet Explorer Document Object Model (DOM) with objects, methods, and properties of your own, implemented in your own application. 通过给MSHTML提供一个你自己实现的对象的IDispatch指针，你就可以通过 document的external对象访问到你的扩展对象。

这个方法的实现非常简单，假设你的IDispatch接口和IDocHostUIHandler接口是同一个对象。

```
HRESULT CBrowserHost::GetExternal(IDispatch **ppDispatch)
{
    *ppDispatch = this;
    return S_OK;
}
```

一旦MSHTML有了指向你的IDispatch接口的指针，MSHTML将通过external对象访问到你 的方法：

```
<SCRIPT language="JScript">

function MyFunc(iSomeData)
{
    external.MyCustomMethod("Some text", iSomeData);
}

</SCRIPT>
```

你可以使用这种技术向Web页传送整个对象。采用下列方法可以做到这一点，在你的 IDispatch实现里生成一个方法，将你想要用到的对象返回。

```
<SCRIPT language="JScript">

function MyFunc(iSomeData)
{
    var oCustCalendarObj;
    external.GetCustomCalender(oCustCalenderObj);
    oCustCalenderObj.doStuffWithIt();
    .
}
```

```
    }  
</SCRIPT>
```

IDocHostUIHandler::GetOptionKeyPath

IDocHostUIHandler::GetOptionKeyPath is a very powerful tool for customizing the WebBrowser Control. Many of the WebBrowser Control display and behavior settings are stored in the registry under the HKEY_CURRENT_USER key. IDocHostUIHandler::GetOptionKeyPath gives you an opportunity to override these registry settings for your specific instance of the WebBrowser Control. It does this by letting you supply an alternate registry location from which the WebBrowser Control will read in registry settings.

An implementation of IDocHostUIHandler::GetOptionKeyPath passes a string to the WebBrowser Control for the registry location you want it to read from. The WebBrowser Control will look for this key under the HKEY_CURRENT_USER key.

Show Example

```
HRESULT CBrowserHost::GetOptionKeyPath(LPOLESTR *pchKey,  
                                      DWORD dwReserved)  
{  
    HRESULT hr;  
  
    #define CCHMAX 256  
    size_t cchLength;  
  
    if (pchKey)  
    {  
        WCHAR* szMyKey = L"Software\\MyCompany\\MyApp";
```

```
    hr = StringCchLengthW(szMyKey, CCHMAX, &cchLength);

    // TODO: Add error handling code here.

    *pchKey = (LPOLESTR)CoTaskMemAlloc((cchLength + 1) * sizeof(WCHAR));

    if (*pchKey)

        hr = StringCchCopyW(*pchKey, cchLength + 1, szKey);

        // TODO: Add error handling code here.

    hr = (*pchKey) ? S_OK : E_OUTOFMEMORY;

}

else

    hr = E_INVALIDARG;

return hr;
}
```

As with IDocHostUIHandler::GetHostInfo, be sure to allocate memory for your strings using CoTaskMemAlloc.

Telling the WebBrowser Control where to look for your registry settings is the first step—actually, it's the second step as far as program execution is concerned. Your program must set the keys at the location specified by IDocHostUIHandler::GetOptionKeyPath so the WebBrowser Control can read them. There are a variety of ways to do this. One way would be with a registry script that runs when the application is installed. Another way might be to do it programmatically when the application starts. Here's a function that sets keys to change the default font, size, and color.

Show Example

HRESULT SetSomeKeys()

```
{  
    HKEY hKey = NULL;  
    HKEY hKey2 = NULL;  
    HKEY hKey3 = NULL;  
    DWORD dwDisposition = NULL;  
    LONG lResult = NULL;  
#define CBMAX 256  
    size_t cbLength;  
  
    RegCreateKeyEx(HKEY_CURRENT_USER, _T("Software\\MyCompany\\MyApp"),  
        NULL, NULL, REG_OPTION_NON_VOLATILE, KEY_SET_VALUE,  
        NULL, &hKey, &dwDisposition);  
  
    RegCreateKeyEx(hKey, _T("Main"), NULL, NULL, REG_OPTION_NON_VOLATILE,  
        KEY_SET_VALUE, NULL, &hKey2, &dwDisposition);  
  
    RegSetValueEx(hKey2, _T("Use_DlgBox_Colors"), NULL, REG_SZ,  
        (CONST BYTE*)_T("no"), sizeof(_T("no")));  
  
    RegCloseKey(hKey2);  
  
    RegCreateKeyEx(hKey, _T("Settings"), NULL, NULL,  
        REG_OPTION_NON_VOLATILE,  
        KEY_SET_VALUE, NULL, &hKey2, &dwDisposition);  
  
    RegSetValueEx(hKey2, _T("Anchor Color"), NULL, REG_SZ,  
        (CONST BYTE*)_T("0, 255, 255"), sizeof(_T("0, 255, 255")));  
  
    RegSetValueEx(hKey2, _T("Text Color"), NULL, REG_SZ,  
        (CONST BYTE*)_T("255, 0, 255"), sizeof(_T("255, 0, 255")));
```

```
RegCloseKey(hKey2);

RegCreateKeyEx(hKey, _T("International\\Scripts"), NULL, NULL,
    REG_OPTION_NON_VOLATILE, KEY_SET_VALUE, NULL,
    &hKey2, &dwDisposition);

BYTE bDefaultScript = 0x3;

RegSetValueEx(hKey2, _T("Default_Script"), NULL, REG_BINARY,
    &bDefaultScript, sizeof(bDefaultScript));

RegCreateKeyEx(hKey2, _T("3"), NULL, NULL, REG_OPTION_NON_VOLATILE,
    KEY_SET_VALUE, NULL, &hKey3, &dwDisposition);

BYTE bSize = 0x4; // Value from 0 - 4. 2 is medium.

TCHAR* szFontName = _T("Comic Sans MS");
TCHAR* szFixedFontName = _T("Courier");

HRESULT hr = StringCbLength(szFontName, CBMAX, &cbLength);
// TODO: Add error handling code here.

RegSetValueEx(hKey3, _T("IEPropFontName"), NULL, REG_SZ,
    (CONST BYTE*) szFontName, cbLength + sizeof(TCHAR));

hr = StringCbLength(szFixedFontName, CBMAX, &cbLength);
// TODO: Add error handling code here.

RegSetValueEx(hKey3, _T("IEFixedFontName"), NULL, REG_SZ,
    (CONST BYTE*) szFixedFontName, cbLength + sizeof(TCHAR));

RegSetValueEx(hKey3, _T("IEFontSize"), NULL, REG_BINARY, &bSize,
```

```
    sizeof(bSize));  
  
    RegCloseKey(hKey3);  
    RegCloseKey(hKey2);  
    RegCloseKey(hKey);  
  
    return S_OK;  
}
```

IDocHostUIHandler2

IDocHostUIHandler2 has a single method,

IDocHostUIHandler2::GetOverrideKeyPath. It performs a function very similar to IDocHostUIHandler::GetOptionKeyPath. It points your hosted WebBrowser to registry settings to modify the default Internet Explorer registry settings. An implementation of IDocHostUIHandler2::GetOverrideKeyPath will look much the same as an implementation of IDocHostUIHandler::GetOptionKeyPath.

GetOptionKeyPath and GetOverrideKeyPath Compared

You probably don't see any difference between

IDocHostUIHandler::GetOptionKeyPath and IDocHostUIHandler2::GetOverrideKeyPath. The difference between them is subtle, but significant. If you implement IDocHostUIHandler::GetOptionKeyPath, your WebBrowser Control instance will ignore any user settings for Internet Explorer. These settings are stored in the registry under HKEY_CURRENT_USER/Software/Microsoft/Internet Explorer. If you implement IDocHostUIHandler2::GetOverrideKeyPath, your WebBrowser Control instance will incorporate any user settings—font settings, menu extensions, and so forth—into the way it displays and behaves.

To illustrate the difference between IDocHostUIHandler::GetOptionKeyPath and IDocHostUIHandler2::GetOverrideKeyPath, let's take a look again at the code

example for the section on IDocHostUIHandler::ShowContextMenu. Remember the line:

```
spCT->Exec (&CGID_ShellDocView, SHDVID_ADDMENUEXTENSIONS, 0, &var1, &var2);
```

If you've implemented IDocHostUIHandler::GetOptionKeyPath, this line would add no menu items to the shortcut menu because menu extension information is stored in the registry for the current user. If you've implemented IDocHostUIHandler2::GetOverrideKeyPath, this line would add any extensions defined for the current user under HKEY_CURRENT_USER/Software/Microsoft/Internet Explorer/MenuExt, unless you explicitly supply an empty or alternative MenuExt key in your custom registry key.

Controlling Navigation

You may have wondered why the section on IDocHostUIHandler didn't mention IDocHostUIHandler::TranslateUrl as a method to implement when you wish to control page navigation. The reason is that this method is not the most general purpose technique with which to control navigation. Unless you are hosting MSHTML directly, this method will have no effect on navigation. Instead, you can control navigation by implementing your IDispatch::Invoke method to handle DISPID_BEFORENAVIGATE2. As an example, the following code prevents navigation to a particular URL, displaying a "No Navigation Allowed" error page if the user attempts to do so.

Show Example

```
case DISPID_BEFORENAVIGATE:  
{  
    CComBSTR url = (*pDispParams).rgvarg[5].pvarVal->bstrVal;
```

```
if (url == "http://www.adatum.com" || url == "http://www.adatum.com/")
{
    CComPtr<IWebBrowser2> spBrowser;
    CComPtr spDisp = ((*pDispParams).rgvarg)[6].pdispVal;
    spDisp->QueryInterface(IID_IWebBrowser2, (void**)&spBrowser);
    spBrowser->Stop();
    CComBSTR newURL = "L"res://webhost.exe/nonavigate.htm";
    spBrowser->Navigate(newURL, NULL, NULL, NULL, NULL);
    ((*pDispParams).rgvarg)[0].boolVal = TRUE;
}
break;
}
```

IDocHostShowUI

This interface gives you control over the message boxes and help files displayed by the WebBrowser Control. It works the same way as IDocHostUIHandler and IDocHostUIHandler2 in that you implement it so the WebBrowser Control can call your IDocHostShowUI methods before it displays any messages or help menus of its own. This gives you a chance to prevent the WebBrowser Control from displaying anything and enables you to display your own custom message or help instead. IDocHostShowUI has two methods, IDocHostShowUI::ShowMessage and IDocHostShowUI::ShowHelp.

IDocHostShowUI::ShowMessage

Return S_OK to disable WebBrowser Control messages. Any other return value, like S_FALSE or E_NOTIMPL, allows the WebBrowser Control to display with its message box.

One nice thing you can do with this method is customize the message box captions for your application so they don't read "Microsoft Internet Explorer."

You can do this by comparing the caption string in lpstrCaption with the string resource Internet Explorer uses, which is stored in Shdoclc.dll. It is identified by the symbol IDS_MESSAGE_BOX_TITLE, whose value is 2213. The following code snippet shows how you might do this.

Show Example

```
HRESULT CBrowserHost::ShowMessage(HWND hwnd,
                                    LPOLESTR lpstrText,
                                    LPOLESTR lpstrCaption,
                                    DWORD dwType,
                                    LPOLESTR lpstrHelpFile,
                                    DWORD dwHelpContext,
                                    LRESULT *plResult)

{

    USES_CONVERSION;

    TCHAR pBuffer[50];

    // resource identifier for window caption "Microsoft Internet Explorer"
#define IDS_MESSAGE_BOX_TITLE 2213

    // Load Shdoclc.dll and the IE message box title string
    HINSTANCE hinstSHDOCLC = LoadLibrary(TEXT("SHDOCLC.DLL"));

    if (hinstSHDOCLC == NULL)
    {
        // Error loading module -- fail as securely as possible
        return;
    }

    LoadString(hinstSHDOCLC, IDS_MESSAGE_BOX_TITLE, pBuffer, 50);
```

```
// Compare the IE message box title string with lpstrCaption
// If they're the same, substitute your own Caption
if (_tcscmp(OLE2T(lpstrCaption), pBuffer) == 0)
    lpstrCaption = L"Custom Caption";

// Create your own message box and display it
*p1Result = MessageBox(OLE2T(lpstrText), OLE2T(lpstrCaption), dwType);

// Unload Shdoclc.dll and return
FreeLibrary(hinstSHDOCLC);

return S_OK;
}
```

Security Alert Using LoadLibrary incorrectly can compromise the security of your application by loading the wrong DLL. Refer to the LoadLibrary documentation for information on how to correctly load DLLs with different versions of Windows.

IDocHostShowUI::ShowHelp

This method is called whenever Internet Explorer Help would be shown, for instance when the F1 key is pressed, and works analogously to IDocHostShowUI::ShowMessage. Return S_OK to override Internet Explorer Help, or another HRESULT value to let Internet Explorer proceed with its Help.

Controlling Download and Execution

The WebBrowser Control gives you control over what it downloads, displays, and executes. To gain this control, you need to implement your host's IDispatch so it handles DISPID_AMBIENT_DLCONTROL. When the WebBrowser Control is instantiated, it will call your IDispatch::Invoke with this ID. Set pvarResult to a combination of following flags, using the bitwise OR operator, to indicate

your preferences.

DLCTL_DLIMAGES, DLCTL_VIDEOS, and DLCTL_BGSOUNDS: Images, videos, and background sounds will be downloaded from the server and displayed or played if these flags are set. They will not be downloaded and displayed if the flags are not set.

DLCTL_NO_SCRIPTS and DLCTL_NO_JAVA: Scripts and Java applets will not be executed.

DLCTL_NO_DLACTIVEXCTLS and DLCTL_NO_RUNACTIVEXCTLS : ActiveX controls will not be downloaded or will not be executed.

DLCTL_DOWNLOADONLY: The page will only be downloaded, not displayed.

DLCTL_NO_FRAMEDOWNLOAD: The WebBrowser Control will download and parse a frameSet, but not the individual frame objects within the frameSet.

DLCTL_RESYNCHRONIZE and DLCTL_PRAGMA_NO_CACHE: These flags cause cache refreshes. With DLCTL_RESYNCHRONIZE, the server will be asked for update status. Cached files will be used if the server indicates that the cached information is up-to-date. With DLCTL_PRAGMA_NO_CACHE, files will be re-downloaded from the server regardless of the update status of the files.

DLCTL_NO_BEHAVIORS: Behaviors are not downloaded and are disabled in the document.

DLCTL_NO_METACHARSET_HTML: Character sets specified in meta elements are suppressed.

DLCTL_URL_ENCODING_DISABLE_UTF8 and DLCTL_URL_ENCODING_ENABLE_UTF8: These flags function similarly to the DOCHOSTUIFLAG_URL_ENCODING_DISABLE_UTF8 and DOCHOSTUIFLAG_URL_ENCODING_ENABLE_UTF8 flags used with IDocHostUIHandler::GetHostInfo. The difference is that the DOCHOSTUIFLAG flags are checked only when the WebBrowser Control is first instantiated. The download flags here for the ambient property change are checked whenever the WebBrowser Control needs to perform a download.

DLCTL_NO_CLIENTPULL: No client pull operations will be performed.

DLCTL_SILENT: No user interface will be displayed during downloads.

DLCTL_FORCEOFFLINE: The WebBrowser Control always operates in offline mode.

DLCTL_OFFLINEIFNOTCONNECTED and DLCTL_OFFLINE: These flags are the same.

The WebBrowser Control will operate in offline mode if not connected to the Internet.

DISPID_AMBIENT_DLCONTROL and the flag values are defined in mshtmdid.h.

Initially, when the function call to IDispatch::Invoke starts, the VARIANT to which the parameter pvarResult points is of type VT_EMPTY. You must switch the type to VT_I4 for any settings to have an effect. You can place your flag values in the lVal member of the VARIANT.

Most of the flag values have negative effects, that is, they prevent behavior that normally happens. For instance, scripts are normally executed by the WebBrowser Control if you don't customize its behavior. But if you set the DLCTL_NOSCRIPTS flag, no scripts will execute in that instance of the control. However, three flags—DLCTL_DLIMAGES, DLCTL_VIDEOS, and DLCTL_BGSOUNDS—work exactly opposite. If you set flags at all, you must set these three for the WebBrowser Control to behave in its default manner vis-a-vis images, videos and sounds.

The following code sample causes a WebBrowser Control instance to download and display images and videos, but not background sounds, since the DLCTL_BGSOUNDS is not explicitly set. Also, script execution on pages displayed by the WebBrowser Control is disabled.

Show Example

```
STDMETHODIMP CAt1BrCon::Invoke(DISPID dispidMember, REFIID riid,
```

```
LCID lcid, WORD wFlags,
```

```
DISPPARAMS* pDispParams,
VARIANT* pvarResult,
EXCEPINFO* pExcepInfo,
UINT* puArgErr)

{

switch (dispidMember)

{

case DISPID_AMBIENT_DLCONTROL:

    pvarResult->vt = VT_I4;
    pvarResult->lVal = DLCTL_DLIMAGES | DLCTL_VIDEOS | DLCTL_NO_SCRIPTS;
    break;

default:

    return DISP_E_MEMBERNOTFOUND;
}

return S_OK;
}
```

IHostDialogHelper

IHostDialogHelper is an interface you can use to create dialog boxes according to your liking. This interface has one method, IHostDialogHelper::ShowHTMLDialog. This method provides the same service as the function ShowHTMLDialog, but it's a little easier to use.

To use IHostDialogHelper, you create the dialog helper object from scratch. Here's how you would do it using CoCreateInstance. The interface and IDs are defined in mshtmhst.h.

Show Example

```
IHostDialogHelper* pHDH;
IMoniker* pUrlMoniker;
BSTR bstrOptions = SysAllocString(L"dialogHeight:30;dialogWidth:40");
BSTR bstrPath = SysAllocString(L"c:\\dialog.htm");

CreateURLMoniker(NULL, bstrPath, &pUrlMoniker);

// Create the dialog helper object
CoCreateInstance(CLSID_HostDialogHelper,
                 NULL,
                 CLSCTX_INPROC,
                 IID_IHostDialogHelper,
                 (void**) &pHDH);

// Call ShowHTMLDialog to create your dialog box
pHDH->ShowHTMLDialog(NULL,
                      pUrlMoniker,
                      NULL,
                      bstrOptions,
                      NULL,
                      NULL);

// Free resources
SysFreeString(bstrPath);
SysFreeString(bstrOptions);
pUrlMoniker->Release();
pHDH->Release();
```

Controlling New Windows

One important way to take control of the WebBrowser Control is to control navigation. You saw earlier how you can intercept DISPID_BEFORNAVIGATE2 in an IDispatch::Invoke implementation to control where your WebBrowser Control will navigate. Another important aspect of navigation is to control how the navigation occurs, especially when opening new windows. Let's say, for instance, that the user clicks the right mouse button over a link and chooses "Open in New Window" or that a page contains a script like this:

```
window.open("www.msn.com");
```

By default, the WebBrowser Control deals with this code by opening a new instance of Internet Explorer to display the page. This may be fine for your application. But then again, it may not. Perhaps you'll want all links to open in your current WebBrowser Control instance. Or perhaps you'll want to open a link in a new WebBrowser Control instance under your control, with your user interface and with your branding.

You can intercept an event, DWebBrowserEvents2::NewWindow2, in your IDispatch implementation to control this. Your control needs to connect to the DWebBrowserEvents2 connection point to intercept this event.

Once you're connected to DWebBrowserEvents2, implement your IDispatch::Invoke so that it handles DISPID_NEWWINDOW2. During the IDispatch::Invoke function call for DISPID_NEWWINDOW2, the array pDispParams contains two parameters. The first one, at index zero, is a Boolean value that tells the WebBrowser Control whether to cancel the new window or not. By default, it is FALSE and a new window will open. If you want to cancel new window creation completely, set the flag to TRUE.

The parameter at index one is a pointer to an IDispatch interface. You can

set this parameter to the IDispatch of a WebBrowser Control that you've created. When you pass back an IDispatch like this, MSHTML will use the control you've given it to open the link.

另请参见 msdn:

<http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/hosting/wbcustomization.asp>

19、如何给 WebBrowser 控件的扩展文档对象（文档. external）提供事件能力？

通过上一知识库（18），你应该知道了如何扩展 WebBrowser 的文档对象模型，但是随后你就会发现在 Web 页中没有办法响应你的对象的事件，即便你提供了事件接口也不行。

那该怎么办呢？

让我们回顾一下 document 等 IE 默认对象的事件处理方式。例如，如果我们要处理 document 对象的 onmousemove 事件，脚本代码如下：

```
<script language=' javascript'>
function OnMyMouseMove()
{
    //do my processing
}
document.onmousemove=OnMyMouseMove;
</script>
```

通过分析 IHTMLDocument2 接口的 onmousemove 定义，我们会发现：onmousemove 只不过是 IHTMLDocument2 接口的一个普通属性而已，唯一的特殊点是它必须为 VARIANT 类型，且实际类型为 VT_IDISPATCH，也就是说 onmousemove 属性的 pdispVal 成员记录了脚本中 onmousemove 事件处理函数对象的 IDispatch 接口指针，而且经测试可以发现 onmousemove 事件处理函数对象的第一个成员（即 dispid 为 0 的成员）就是脚本中给出的处理函数，本例为：OnMyMouseMove 函数。

按照这种思路，我们可以给自己的 DOM 扩展对象（external）定义一个表示事件处理函数的属性，并在事件发生时调用此属性的 Invoke 方法（dispid 为 0），实例如下：

```
//CMyObj 是提供给 Web 页的 external 对象，IMyObj 接口暴露了 OnShow 属性并影射到 spOnShow 变量
class CMyObj: IMyObj
{
public:
```

```
CComPtr<IDispatch> spOnShow;
//在事件发生时调用Fire_OnShow函数会激发Web页中提供的响应函数
Fire_OnShow(BSTR bstr)
{
    CComVariant avarParams[1];
    avarParams[0] = bstr;    avarParams[0].vt = VT_BSTR;
    CComVariant varResult;

    DISPPARAMS params = { avarParams, NULL, 1, 0 };
    HRESULT hr = p->Invoke(0, IID_NULL, LOCALE_USER_DEFAULT,
DISPATCH_METHOD, &params, &varResult, NULL, NULL);
}
};
```

相应的页面脚本代码如下：

```
<HTML>
    <BODY>
        <script language="javascript">
            function OnShow(v)
            {
                alert(v);
            }
            external.OnShow=OnShow;
        </script>
    </BODY>
</HTML>
```

20、如何打开流中的 HTML 文档？

The [IPersistStreamInit](#) interface, and its associated methods, can be used to load HTML content from a stream using the [WebBrowser](#) control and Microsoft® Visual C++®.

This article discusses the steps required to load HTML content from a stream and is divided into the following sections.

- [Navigating to about:blank](#)
- [Availability of the DHTML Object Model](#)
- [Using QueryInterface to Obtain the IPersistStreamInit Interface](#)
- [Using the IPersistStreamInit Interface to Load HTML Content](#)

Navigating to about:blank

The **IWebBrowser2::Navigate2** method of the **IWebBrowser2** interface enables you to navigate the browser to a URL. In the following sample code, the **IWebBrowser2::Navigate2** method is used to navigate to the about:blank

page. Navigating to this empty page ensures that MSHTML is loaded and that the HTML elements are available through the Dynamic HTML (DHTML) Object Model.

This example demonstrates how to navigate the **WebBrowser** control to an empty page. The variable **m_pBrowser** contains the **IWebBrowser2** interface pointer obtained from the **WebBrowser** control.

```
m_pBrowser->Navigate2( _T("about:blank"), NULL, NULL, NULL, NULL  
);
```

Availability of the DHTML Object Model

The DHTML Object Model is used to access and manipulate the contents of an HTML page and is not available until the page is loaded. Your application determines that a page is loaded by handling the [DWebBrowserEvents2::DocumentComplete](#) event of the **WebBrowser** control. This event may be fired once for each frame in the page, and once when the top frame of the document is loaded. You can determine if the **DWebBrowserEvents2::DocumentComplete** event is for the top frame by comparing the [IDispatch](#) interface pointer passed by this event with that of the **WebBrowser** control.

This sample handler code for the **WebBrowser DWebBrowserEvents2::DocumentComplete** event demonstrates how to determine if this event is for the top frame, which indicates that the HTML page has loaded. This sample also demonstrates how to create a stream from a block of memory—in this case a string that contains the HTML content to be displayed.

```
void myObject::DocumentComplete(LPDISPATCH pDisp, VARIANT* URL)  
{  
    HRESULT hr;
```

```
IUnknown* pUnkBrowser = NULL;

IUnknown* pUnkDisp = NULL;

IStream* pStream = NULL;

HGLOBAL hHTMLText;

static TCHAR szHTMLText[] = "<html><h1>Stream Test</h1><p>This
HTML content is/
being loaded from a stream.</html>";

// Is this the DocumentComplete event for the top frame window?

// Check COM identity: compare IUnknown interface pointers.

hr = m_pBrowser->QueryInterface( IID_IUnknown,
(void**) &pUnkBrowser );

if ( SUCCEEDED(hr) )

{

    hr = pDisp->QueryInterface( IID_IUnknown,
(void**) &pUnkDisp );

    if ( SUCCEEDED(hr) )

    {

        if ( pUnkBrowser == pUnkDisp )

        {   // This is the DocumentComplete event for the top

            // frame - page is loaded!

            // Create a stream containing the HTML.

            // Alternatively, this stream may have been passed
to us.

size_t = cchLength;
```

```
// TODO: Safely determine the length of szHTMLText  
in TCHAR.  
  
hHTMLText = GlobalAlloc( GPTR, cchLength+1 );  
  
if ( hHTMLText )  
{  
    size_t cchMax = 256;  
  
    StringCchCopy((TCHAR*)hHTMLText, cchMax + 1,  
szHTMLText);  
  
    // TODO: Add error handling code here.  
  
    hr = CreateStreamOnHGlobal( hHTMLText, TRUE,  
&pStream );  
  
    if ( SUCCEEDED(hr) )  
    {  
        // Call the helper function to load the  
        browser from the stream.  
  
        LoadWebBrowserFromStream( m_pBrowser,  
pStream );  
  
        pStream->Release();  
    }  
  
    GlobalFree( hHTMLText );  
}  
}  
pUnkDisp->Release();  
}  
pUnkBrowser->Release();
```

```
    }  
  
}
```

Using QueryInterface to Obtain the IPersistStreamInit Interface

The [IWebBrowser2::get_Document](#) property on the **WebBrowser** control retrieves the [document](#) object that represents the DHTML Object Model for the top frame. MSHTML implements the **IPersistStreamInit** interface to provide the ability to load and save HTML using a stream, through the **document** object. The **IDispatch** interface for the **document** object can be queried for the **IPersistStreamInit** interface pointer using [QueryInterface](#) with an identifier of **IID_IPersistStreamInit**, as shown in the following code example.

```
HRESULT LoadWebBrowserFromStream(IWebBrowser* pWebBrowser,  
IStream* pStream)  
  
{  
  
HRESULT hr;  
  
IDispatch* pHtmlDoc = NULL;  
  
IPersistStreamInit* pPersistStreamInit = NULL;  
  
  
    // Retrieve the document object.  
  
    hr = pWebBrowser->get_Document( &pHtmlDoc );  
  
    if ( SUCCEEDED(hr) )  
  
    {  
  
        // Query for IPersistStreamInit.  
  
        hr = pHtmlDoc->QueryInterface( IID_IPersistStreamInit,  
(void**)&pPersistStreamInit );  
  
        if ( SUCCEEDED(hr) )  
  
        {  
  
            // Initialize the document.  
  
            hr = pPersistStreamInit->InitNew();
```

```
    if ( SUCCEEDED(hr) )

    {

        // Load the contents of the stream.

        hr = pPersistStreamInit->Load( pStream );

    }

    pPersistStreamInit->Release();

}

}

}
```

Using the **IPersistStreamInit** Interface to Load HTML Content

The **IPersistStreamInit** interface has **InitNew** and **Load** methods that are used to initialize and load an HTML document from a stream. The **InitNew** method initializes the stream to a known state and the **Load** method loads the HTML content from the stream.

In the previous sample code, the HTML document is initialized and the HTML content is loaded from the stream.

Note In Microsoft Internet Explorer 5, more than one call to the **Load** method of the [IPersist](#) interfaces is supported. In earlier versions, only one call to **Load** per instance of MSHTML is supported.

另请参阅：

<http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/webbrowser/tutorials/webocstream.asp>

21、如何使 Web 页面中的对象可以拖动？

首先，在页面中加入如下代码：

```
<style>
<!--
.drag{position:relative;cursor:hand}
-->
```

```
</style>
<script language="JavaScript1.2">
<!--
var dragapproved=false
var z, x, y
function move() {
if (event.button==1&&dragapproved) {
z.style.pixelLeft=temp1+event.clientX-x
z.style.pixelTop=temp2+event.clientY-y
return false
}
}
function drags() {
if (!document.all)
return
if (event.srcElement.className=="drag") {
dragapproved=true
z=event.srcElement
temp1=z.style.pixelLeft
temp2=z.style.pixelTop
x=event.clientX
y=event.clientY
document.onmousemove=move
}
}
document.onmousedown=drags
document.onmouseup=new Function("dragapproved=false")
//-->
</script>
```

然后，把你想要拖动的对象的css样式设成drag，例如：

```
<input type="button" class="drag" >
```

22、如何把Web页面中用到的图像等外部文件嵌入DLL或Exe？

在我们写基于Web的应用时，我们为了安全，往往把Web页面嵌入到DLL或Exe中，但如果页面中用到了图片等外部文件时，就不得不写成绝对路径或干脆把页面也放到外面，那么能不能把外部文件也放到DLL或Exe中呢？

通过研究IE可以发现，完全可以这样。我们使用IE的时候，如果遇到连不上网的情况，IE

会显示一个错误信息页，这个页面来自于IE的shdoclc.dll，可以注意到其中有一个刷新图标，通过打开页面源文件，可以看到图标的文件名叫refresh.gif，但是，搜遍Windows目录也找不到这个文件，也就是说他一定存在于DLL或其它地方，我们使用搜索关键字的方法可以找到refresh.gif就位于shdoclc.dll中，我们用VC++的资源编辑器打开shdoclc.dll，可以发现refresh.gif位于资源类型为2110的资源下，资源名就是它的文件名refresh.gif；

也就是说，我们可以在我们的工程的rc文件里加入类型为2110的资源，然后在页面中就可以用它的资源名调用它。步骤如下：

例如我们要把da01.gif放入资源，在rc文件里加入下面行：

```
da01.gif 2110 "res\\da01.gif"
```

在我们的页面里可以这样调用，

```
<body background= "da01.gif" >或  
<body background= "res://mydll/da01.gif" >
```

实验发现，也可以这么做：

1. da01.gif HTML "res\\da01.gif" 或
2. da01.gif IMAGES "res\\da01.gif"

第一种写法与前面的一样，但后一种引用时要这样：

```
<body background= "images/da01.gif" >
```

其实，IMAGES 也可以换成其他，例如：JS，CSS 等，引用方式也与 IMAGES 相同。

23、如何使用 #import 指令从文件定位特定类型库资源 ID？

本分步指南介绍如何使用 **#import** 指令从单个 .dll 文件或单个 .exe 文件定位特定类型库资源 ID。默认情况下，类型库是从第一个类型为 **TypeLib** 的资源中提取的。您可以结合 **#import** 指令使用 **tlbid(n)** 属性，其中 **n** 是要使用的类型库的资源 ID。

使用 #import

如果有多个类型库内置到单个 .dll 文件或单个 .exe 文件，您可以使用 **#import** 指令定位某个特定类型库资源。例如，要定位类型库 **resource 2**，可使用以下 **tlbid** 开关：

```
#import "SERVER.DLL" tlbid(2)
```

此语句从文件 Server.dll 中加载类型库 **resource 2**，然后构建头文件以纳入来自类型库 **resource 2** 的信息。

默认情况下，如果缺少 **tlbid** 指令，将从类型为 **TypeLib** 的第一个资源中提取类型库。当使用此 **tlbid** 开关时，获得的结果与使用 **LoadTypeLib("SERVER.DLL\2")** API 调用获得的结果相同。

24、如何让你的 MFC 类实现 IDispatch 接口？

如果完全手动实现 IDispatch 接口，就要实现 IDispatch 和 IUnknown 的所有方法，非常的繁琐。其实，MFC 提供了更方便的方法实现 IDispatch 接口，方法如下：

你的类必须派生于 CCmdTarget 或其子类，声明宏 DECLARE_DISPATCH_MAP()；

示例代码如下：

```
class CXMLDOMSamples : public CCmdTarget
{
public:
    void DOMDocOnDataAvailable() {};

    DECLARE_DISPATCH_MAP();
};
```

然后用下面的宏映射你的每个IDispatch函数，示例如下：

```
BEGIN_DISPATCH_MAP(CXMLDOMSamples, CCmdTarget)
    DISP_FUNCTION_ID(CXMLDOMSamples, "Data Available Event
Handler", DISPID_FUNC, DOMDocOnDataAvailable, VT_EMPTY, 0)
END_DISPATCH_MAP()
```

这样你的类就实现了 IDispatch 接口，通过 GetIDispatch 方法可以得到你的 IDispatch 接口指针。

25、如何自行管理 IE 缓存目录里的内容？

如果想要自行管理 IE 缓存的内容，例如：删除某一个文件，可以通过 WinInet 函数来实现，DeleteUrlCacheEntry 函数可以删除来自于给定 Url 的文件；URLDownloadToCacheFile 可以下载文件到 IE 缓存；还有其他函数可以参考 MSDN WinInet Functions。

26、Windows Mobile 里如何使用 WebBrowser 控件？

如果在 Windows Mobile、Ce 项目里用通常添加 ActiveX 的方法添加 WebBrowser 控件，会发现虽然能够添加成功，但是却无法运行，控件不能够实例化。

可以用下面的方法创建：

```
#include <htmlctrl.h>
#include <piedocvw.h>
#pragma comment(lib, "Htmlview.lib")

HINSTANCE hInst = ::LoadLibrary(_T("Htmlview.dll"));
InitHTMLControl(hInst);
HWND m_hwndHTML = CreateWindow(WC_HTML, NULL,
    WS_CHILD | WS_VISIBLE | HS_NOSCROLL,
    0, 0, 100, 100, m_hWnd,
    NULL, hInst, NULL);
```

//m_hWnd 是控件的父窗口句柄，一般是 Dialog 的句柄。

那么如何打开网页呢？实际上，WebBrowser 控件可以通过发消息的方式来控制，比如打开网页：

```
    ::SendMessage(m_hwndHTML, DTM_NAVIGATE, 0,  
        (LPARAM)_T("file:///\\Storage Card\\test.htm"));
```

其他可用的消息值可以参考 MSDN（可以通过 InitHTMLControl 关键字查到），我们可以通过 DTM_DOCUMENTDISPATCH 消息拿到文档对象的 IDispatch 接口：

```
CComPtr<IDispatch> dispDoc;  
::SendMessage(m_hwndHTML, DTM_DOCUMENTDISPATCH, 0, (LPARAM)&dispDoc);
```

相似的方法也可以拿到 WebBrowser 对象的 IDispatch 接口。

值得说明的是，尽管我们能够拿到文档对象和 WebBrowser 对象的 IDispatch 接口，但却不容易拿到进一步的 IPIEHTMLDocument2, IWebBrowser2 或 IBrowser3 等接口，原因是：在某些基于 CE 的项目里，比如 Windows Mobile 5.0 PocketPC 里，我们找不到这些接口的定义，如果把 PocketPC 2003 SDK 里面的的定义（定义在 webvw.h 里）复制过来的话，会发现尽管查询接口成功，但是调用某些属性或方法时出现错乱，原因是尽管这些平台都是基于 WinCE 的，但是他们所实现的接口并不相同，相同接口的 IID 可能没有变化，但是接口里所包含的属性或方法却有删减。所以最好的方法是不要拿具体的 vtable 接口，而是直接使用 IDispatch 接口。可以通过 IDispatch 接口的 GetIDsOfNames 得到所用属性或方法的 DispId，然后调用 Invoke。

27、为什么在 Windows Mobile 里无法使用宿主在 exe 里的 COM 组件？

如果你在 Mobile 里使用过宿主在 exe 里的 COM 组件的话，可能会发现 Mobile 里根本不支持这样的组件，一个替代方法就是用 DLL 的形式，如果希望组件使用者，组件 (DLL) 和你的 exe（就是你本来想做为 COM 组件宿主的那个 exe）能够通信的话，可以在 COM 宿主 DLL 里加其他通信函数或直接用 COM 接口。

28、如何扩展 IE 的协议，使其支持自定义的协议？

尽管 IE 支持很多种协议，例如：http, https, ftp, res 等，但有时我们还是希望 IE 能够支持自定义的协议，比如，我们可能希望 IE 能够直接访问位于压缩包里的文件，就像这样子：zip://my.zip!/images/foo.gif。其实，IE 提供了这方面的扩展能力，可以通过异步可插入协议（Asynchronous Pluggable Protocols）的方式实现。通过实现一个异步可插入协议，并将自己的协议实现注册到系统中，可以让 IE 支持并使用自己提供的协议实现来处理自定义的 URL。

相关的接口有：

IInternetProtocolInfo
IInternetProtocolRoot
IInternetProtocol

IInternetProtocolSink

IInternetSession

异步可插入协议是一个单元线程（Apartment thread）的 COM 组件，当客户程序发出请求时，urlmon 将会在系统注册表中查找对应 scheme 的协议处理器（handler），如果找到了，机会实例化它，并将后续的请求交给它处理。如果希望自己的异步可插入协议成为某个协议的处理器，需要在注册里注册自己的 CLSID 和协议的 scheme：

HKEY_CLASSES_ROOT\PROTOCOLS\Handler\protocol_scheme

添加字串值 CLSID，值为自定义协议的 CLSID

比如自定义协议 zip，CLSID 是 {AB648571-CCDC-4671-B18B-A5E10D3F99CF}

HKEY_CLASSES_ROOT\PROTOCOLS\Handler\zip

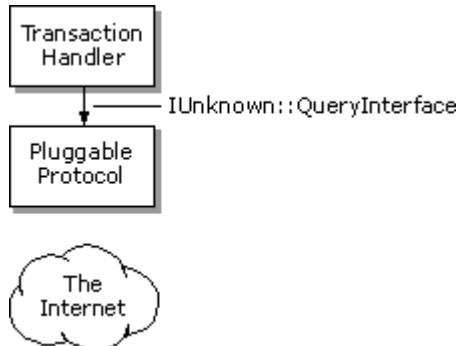
CLSID="{AB648571-CCDC-4671-B18B-A5E10D3F99CF}"

也可以不用写入注册表，而仅在当前进程里注册自定义协议：

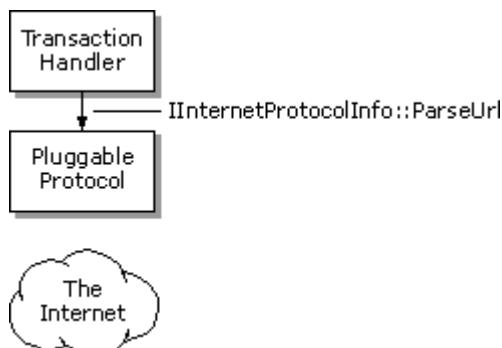
```
CComPtr<IInternetSession> spSession;
CoInternetGetSession(0, &spSession, 0);
spSession->RegisterNameSpace(spClassFactory, __uuidof(ZipProtocolHandler), _T("zip"), 0, 0,
0);
```

异步可插入协议需要实现 IInternetProtocol，IInternetProtocolInfo 接口，IInternetProtocolInfo 是可选的，一般可以不实现。urlmon 调用的过程如下：

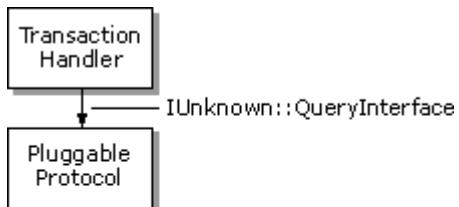
1. 首先 QueryInterface 查询 IInternetProtocolInfo 接口，



2. 如果实现了 IInternetProtocolInfo，调用 IInternetProtocolInfo::ParseUrl



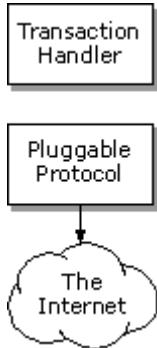
3. QueryInterface 查询 IInternetProtocol 接口，



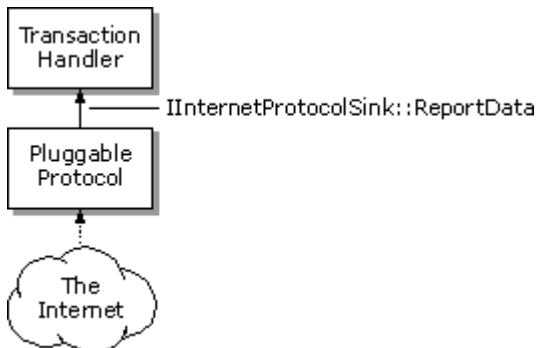
4. 调用 IIInternetProtocolRoot::Start



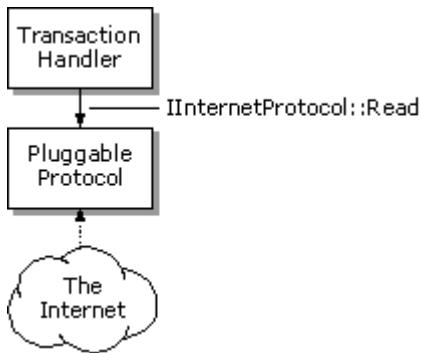
5. 此时你的协议处理器应该开始获取数据,



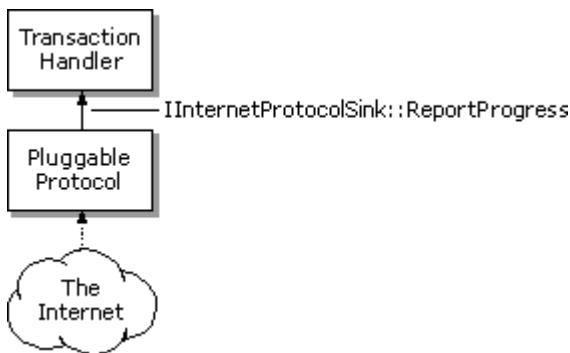
6. 在你的处理器下载数据的时候, 调用 IIInternetProtocolSink::ReportData



7. Urlmon 调用 IIInternetProtocol::Read 来读取数据

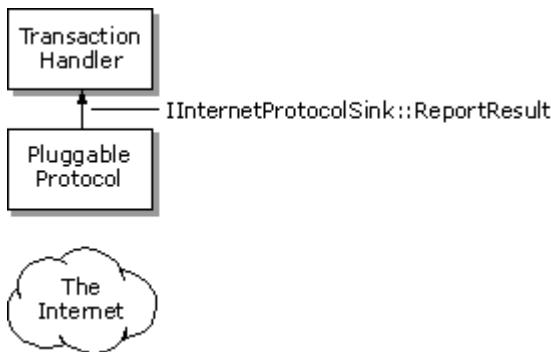


8. 你的协议处理器调用 IInternetProtocolSink::ReportProgress 进度，可以使用 BINDSTATUS_MIMETYPEAVAILABLE 通知 content-type

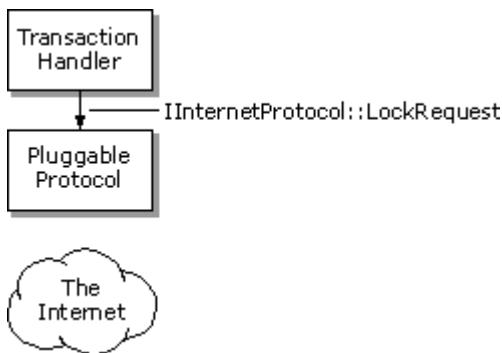


9. 6-8 持续进行，知道你下载完所有数据。

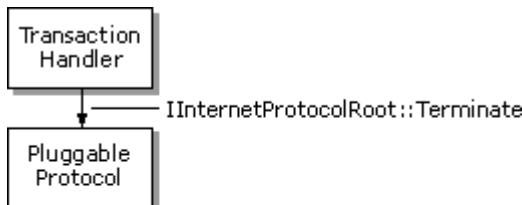
10. 你的协议处理器调用 IInternetProtocolSink::ReportResult



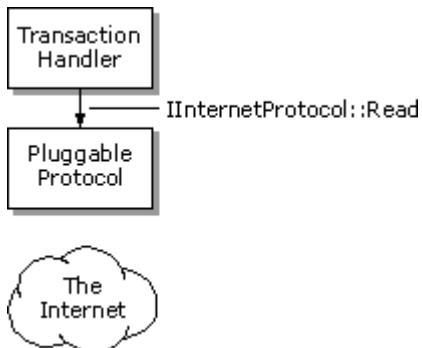
11. Urlmon 调用你的处理器的 IInternetProtocol::LockRequest



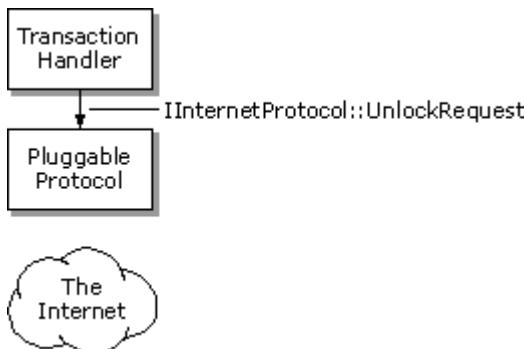
12. Urlmon 调用你的处理器的 IInternetProtocolRoot::Terminate



13. Urlmon 调用你的处理器的 IInternetProtocol::Read，知道所有数据取回



14. Urlmon 调用你的处理器的 IInternetProtocol::UnlockRequest



下面给出一个用 c#实现的例子：

```

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Runtime.InteropServices;
using System.Text.RegularExpressions;

namespace S235.WindowsAPI
{
    public abstract class ProtocolHandlerImpl
        : IInternetProtocol
    {
        Stream _resStream;
    }
}
  
```

```

    ///<summary>
    /// you may report the progress and MIME type by Sink.ReportData and
    Sink.ReportProgress
        ///         Sink.ReportData(BSCF.BSCF_FIRSTDATANOTIFICATION,      0,
    (_resStream.Length);
        ///         Sink.ReportProgress(BINDSTATUS.BINDSTATUS_MIMETYPEAVAILABLE,
    "text/html");
    ///</summary>
    ///<param name="szURL"></param>
    ///<param name="Sink"></param>
    ///<returns></returns>
protected abstract Stream LoadData(string szURL, IInternetProtocolSink Sink);

#region IInternetProtocol Members

public HRESULT LockRequest(uint dwOptions)
{
    return HRESULT.S_OK;
}

public HRESULT Read(IntPtr pv, uint cb, out uint pcbRead)
{
    pcbRead = 0;
    if (_resStream == null)
    {
        //return HRESULT.INET_E_DOWNLOAD_FAILURE;
        return HRESULT.S_FALSE;
    }

    byte[] bytes = new byte[cb];
    pcbRead = (_resStream.Read(bytes, 0, (int)bytes.Length));
    Marshal.Copy(bytes, 0, pv, (int)pcbRead);

    HRESULT hr = (pcbRead == 0) ? HRESULT.S_FALSE : HRESULT.S_OK;
    return hr;
}

public HRESULT Seek(ulong dlibMove, SeekOrigin dwOrigin, out ulong
plibNewPosition)
{
    plibNewPosition = 0;
    HRESULT hr = HRESULT.S_OK;
    if (_resStream != null)
    {

```

```

        try
        {
            plibNewPosition = (ulong)_resStream.Seek((long)dlibMove, dwOrigin);
        }
        catch(Exception exp)
        {
            hr = (HRESULT)Marshal.GetHRForException(exp);
        }
    }

    return hr;
}

public HRESULT UnlockRequest()
{
    ((IDisposable)this).Dispose();
    return HRESULT.S_OK;
}

#endregion

#region IIInternetProtocolRoot Members

public HRESULT Abort(int hrReason, uint dwOptions)
{
    return HRESULT.E_NOTIMPL;
}

public HRESULT Continue(ref _tagPROTOCOLDATA pProtocolData)
{
    return HRESULT.E_NOTIMPL;
}

public HRESULT Resume()
{
    return HRESULT.E_NOTIMPL;
}

public HRESULT Start(string szURL, IIInternetProtocolSink Sink, IIInternetBindInfo
pOIBindInfo, uint grfPI, uint dwReserved)
{
    HRESULT hr = HRESULT.S_OK;
    try
    {
        string localPath = GetLocalPath(szURL);

```

```
        _resStream = LoadData(localPath, Sink);

        BSCF.flags = BSCF.BSCF_LASTDATANOTIFICATION;
        flags |= (_resStream.Length == 0) ? BSCF.BSCF_AVAILABLEDATASIZEUNKNOWN : BSCF.BSCF_DATAFULLYAVAILABLE;
        Sink.ReportData(flags, (uint)_resStream.Length, (uint)_resStream.Length);
        Sink.ReportResult(HRESULT.S_OK, 0, null);
    }

    catch (Exception exp)
    {
        hr = (HRESULT)Marshal.GetHRForException(exp);
        Sink.ReportResult(hr, (uint)hr, exp.Message);
    }

    return hr;
}

public HRESULT Suspend()
{
    return HRESULT.E_NOTIMPL;
}

public HRESULT Terminate(uint dwOptions)
{
    return HRESULT.S_OK;
}

#endregion
}
```