



qq\_20828983的博客

RSS订阅



写评论



目录



收藏



微信



微博



QQ

## 原 MFC对话框里利用CHtmlView加载界面（三）——JavaScript 调用 MFC

2017年07月12日 08:51:27

### 前言

在 MFC对话框里利用CHtmlView加载界面（二）—— MFC 调用 JavaScript中，我们实现MFC对JavaScript方法的调用。

现在我们想既然MFC能调用JavaScript，那么又该如何实现JavaScript调用MFC呢？

只有MFC 能和 JavaScript 实现互相调用，那我们完全就可以将HTML作为我们的界面，将MFC作为我们底层逻辑的处理，实现漂亮的界面了。

相比MFC 调用 JavaScript 只需要获取COM接口，封装几个函数就能实现调用，JavaScript调用MFC就想到复杂很多。

里面涉及了ActiveX编程知识，需要根据浏览器定义的一些扩展接口，对其进行定制和扩展。

### CDocHostSite类实现

首先我们要实现CHtmlView的控制站点类，在此定义为CDocHostSite类。

CDocHostSite类继承自COleControlSite类，实现了IDocHostShowUI、IDocHostUIHandler、IDocHostUIHandler2、IInternetSecurityManager接口。

对于它的具体实现，由于其涉及了很多ActiveX编程知识，理解起来比较困难。但作为ActiveX小白，我们其实并不需要太多关心具体实现，只需要知道

需要注意的一点是：在实现中，我们只需要将CDocHostSite代码中的CBaseHtmlView类改成你自己实现的类就行了。

CDocHostSite类的申明如下：

```
1  #include <afxocx.h>
2
3  class CBaseHtmlView;
4
5  class CDocHostSite : public COleControlSite
6  {
7  public:
8      CDocHostSite(COleControlContainer * pOCC, CBaseHtmlView* pView);
9      virtual ~CDocHostSite();
10
11  private:
12      CBaseHtmlView* m_pView;
13
14  // IDocHostShowUI
15  protected:
16      BEGIN_INTERFACE_PART(DocHostShowUI, IDocHostShowUI)
17      INIT_INTERFACE_PART(CDocHostSite, DocHostShowUI)
18      STDMETHOD>ShowHelp)(
19          /* [in ] */   HWND hwnd,
20          /* [in ] */   LPOLESTR pszHelpFile,
21          /* [in ] */   UINT uCommand,
22          /* [in ] */   DWORD dwData,
23          /* [in ] */   POINT ptMouse,
24          /* [out] */   IDispatch __RPC_FAR *pDispatchObjectHit);
25      STDMETHOD>ShowMessage)(
26          /* [in ] */   HWND hwnd,
27          /* [in ] */   LPOLESTR lpstrText,
28          /* [in ] */   LPOLESTR lpstrCaption,
29          /* [in ] */   DWORD dwType,
```

```

30      /* [in] */ LPOLESTR lpstrHelpFile, 31 | /* [in] */ DWORD dwHelpContext,
31      /* [out] */ LRESULT __RPC_FAR *pIResult);
32  END_INTERFACE_PART(DocHostUIHandler)
33      0
34
35  // IDocHostUIHandler
36  protected:
37  BEGIN_INTERFACE_PART(DocHostUIHandler, IDocHostUIHandler)
38  STDMETHOD(ShowContextMenu)(/* [in] */ DWORD dwID,
39                             /* [in] */ POINT __RPC_FAR *ppt,
40                             /* [in] */ IUnknown __RPC_FAR *pcmdtReserved,
41                             /* [in] */ IDispatch __RPC_FAR *pdispReserved);
42  STDMETHOD(GetHostInfo)(/* [out][in] */ DOCHOSTUIINFO __RPC_FAR *pInfo);
43  STDMETHOD(ShowUI)(
44      /* [in] */ DWORD dwID,
45      /* [in] */ IOleInPlaceActiveObject __RPC_FAR *pActiveObject,
46      /* [in] */ IOleCommandTarget __RPC_FAR *pCommandTarget,
47      /* [in] */ IOleInPlaceFrame __RPC_FAR *pFrame,
48      /* [in] */ IOleInPlaceUIWindow __RPC_FAR *pDoc);
49  STDMETHOD(HideUI)(void);
50  STDMETHOD(UpdateUI)(void);
51  STDMETHOD(EnableModeless)(/* [in] */ BOOL fEnable);
52  STDMETHOD(OnDocWindowActivate)(/* [in] */ BOOL fEnable);
53  STDMETHOD(OnFrameWindowActivate)(/* [in] */ BOOL fEnable);
54  STDMETHOD(ResizeBorder)(
55      /* [in] */ LPCRECT prcBorder,
56      /* [in] */ IOleInPlaceUIWindow __RPC_FAR *pUIWindow,
57      /* [in] */ BOOL fFrameWindow);
58  STDMETHOD(TranslateAccelerator)(
59      /* [in] */ LPMSG lpMsg,
60      /* [in] */ const GUID __RPC_FAR *pguidCmdGroup,
61      /* [in] */ DWORD nCmdID);
62  STDMETHOD(GetOptionKeyPath)(
63      /* [out] */ LPOLESTR __RPC_FAR *pchKey,
64      /* [in] */ DWORD dw);
65  STDMETHOD(GetDropTarget)(
66      /* [in] */ IDropTarget __RPC_FAR *pDropTarget,
67      /* [out] */ IDropTarget __RPC_FAR *ppDropTarget);
68  STDMETHOD(GetExternal)(
69      /* [out] */ IDispatch __RPC_FAR *ppDispatch);
70  STDMETHOD(TranslateUrl)(
71      /* [in] */ DWORD dwTranslate,
72      /* [in] */ OLECHAR __RPC_FAR *pchURLIn,
73      /* [out] */ OLECHAR __RPC_FAR *ppchURLOut);
74  STDMETHOD(FilterDataObject)(
75      /* [in] */ IDataObject __RPC_FAR *pDO,
76      /* [out] */ IDataObject __RPC_FAR *ppDORet);
77  END_INTERFACE_PART(DocHostUIHandler)
78
79
80  #if (_WIN32_IE >= 0x0501) // IE 5.5 and higher
81  // IDocHostUIHandler2
82  protected:
83  BEGIN_INTERFACE_PART(DocHostUIHandler2, IDocHostUIHandler2)
84  STDMETHOD(GetOverrideKeyPath)(
85      /* [out] */ LPOLESTR __RPC_FAR *pchKey,
86      /* [in] */ DWORD dw);
87  STDMETHOD(ShowContextMenu)(/* [in] */ DWORD dwID,
88                             /* [in] */ POINT __RPC_FAR *ppt,
89                             /* [in] */ IUnknown __RPC_FAR *pcmdtReserved,
90                             /* [in] */ IDispatch __RPC_FAR *pdispReserved);
91  STDMETHOD(GetHostInfo)(
92      /* [out][in] */ DOCHOSTUIINFO __RPC_FAR *pInfo);
93  STDMETHOD(ShowUI)(
94      /* [in] */ DWORD dwID,
95      /* [in] */ IOleInPlaceActiveObject __RPC_FAR *pActiveObject,
96      /* [in] */ IOleCommandTarget __RPC_FAR *pCommandTarget,
97      /* [in] */ IOleInPlaceFrame __RPC_FAR *pFrame,
98      /* [in] */ IOleInPlaceUIWindow __RPC_FAR *pDoc);
99  STDMETHOD(HideUI)(void);
100  STDMETHOD(UpdateUI)(void);
101  STDMETHOD(EnableModeless)(/* [in] */ BOOL fEnable);

```

```

102     STDMETHOD(OnDocWindowActivate)(/* [in] */ BOOL fEnable);
103     STDMETHOD(OnFrameWindowActivate)(/* [in] */ BOOL fEnable);

104     STDMETHOD(ResizeBorder,
105         /* [in] */ LPCRECT pBorder,
106         /* [in] */ IUnknown* pUIWindow,
107         /* [in] */ BOOL fWindow);
108     STDMETHOD(TranslateAccelerator)(
109         /* [in] */ LPMSCROLLINFO,
110         /* [in] */ const __RPC_FAR *pguidCmdGroup,
111         /* [in] */ DWORD dwCmdID);
112     STDMETHOD(GetOptionKey)(
113         /* [out] */ LPOLYCHAR __RPC_FAR *pchKey,
114         /* [in] */ DWORD dwOptions);
115     STDMETHOD(GetDropTarget,
116         /* [in] */ IDropTarget __RPC_FAR *pDropTarget,
117         /* [out] */ IDropTarget __RPC_FAR *ppDropTarget);
118     STDMETHOD(GetExternalDispatch)(
119         /* [out] */ IDispatch __RPC_FAR *ppDispatch);
120     STDMETHOD(TranslateURL,
121         /* [in] */ DWORD dwTranslate,
122         /* [in] */ OLECHAR __RPC_FAR *pchURLIn,
123         /* [out] */ OLECHAR __RPC_FAR *ppchURLOut);
124     STDMETHOD(FilterDataObject)(
125         /* [in] */ IDataObject __RPC_FAR *pDO,
126         /* [out] */ IDataObject __RPC_FAR *ppDORet);
127     END_INTERFACE_PART(DocHostUIHandler2)
128 #endif
129
130 // Implement IInternetSecurityManager
131 BEGIN_INTERFACE_PART(InternetSecurityManager, IInternetSecurityManager)
132     STDMETHOD(SetSecuritySite)(IInternetSecurityMgrSite*);
133     STDMETHOD(GetSecuritySite)(IInternetSecurityMgrSite**);
134     STDMETHOD(MapUrlToZone)(LPCWSTR, DWORD*, DWORD*);
135     STDMETHOD(GetSecurityId)(LPCWSTR, BYTE*, DWORD*, DWORD_PTR);
136     STDMETHOD(ProcessUrlAction)(
137         /* [in] */ LPCWSTR pwszUrl,
138         /* [in] */ DWORD dwAction,
139         /* [size_is][out] */ BYTE __RPC_FAR *pPolicy,
140         /* [in] */ DWORD cbPolicy,
141         /* [in] */ BYTE __RPC_FAR *pContext,
142         /* [in] */ DWORD cbContext,
143         /* [in] */ DWORD dwFlags,
144         /* [in] */ DWORD dwReserved = 0);
145     STDMETHOD(QueryCustomPolicy)(LPCWSTR, REFGUID, BYTE**, DWORD*, BYTE*, DWORD, DWORD);
146     STDMETHOD(SetZoneMapping)(DWORD, LPCWSTR, DWORD);
147     STDMETHOD(GetZoneMappings)(DWORD, IEnumString**, DWORD);
148     END_INTERFACE_PART(InternetSecurityManager)
149
150
151 ///////////////////////////////////////////////////
152 // Implement IServiceProvider
153 BEGIN_INTERFACE_PART(ServiceProvider, IServiceProvider)
154     STDMETHOD(QueryService)(REFGUID, REFIID, void**);
155     END_INTERFACE_PART(ServiceProvider)
156
157 DECLARE_INTERFACE_MAP()
158 };

```

实现部分为：

```

1 #include "stdafx.h"
2
3 #if _MFC_VER >= 0x0700
4 #include <afxoccur.h>
5 #else
6 #ifdef _AFXDLL
7 #undef AFX_DATA
8 #define AFX_DATA AFX_DATA_IMPORT
9 #endif
10 #include <..\src\ocimpl.h>

```

```

11 | #ifdef _AFXDLL
12 | #undef AFX_DATA
13 | #define AFX_DATA AFX_DATA_IMPL
14 | #endif
15 | #endif
16 |
17 | #include <mshtml.h>
18 |
19 | #include "DocHostSite.h"
20 | #include "BaseHtmlView.h"
21 |
22 | BEGIN_INTERFACE_MAP(CDocHostSite, COleControlSite)
23 | INTERFACE_PART(CDocHostSite, IID_IDocHostShowUI, DocHostShowUI)
24 | INTERFACE_PART(CDocHostSite, IID_IDocHostUIHandler, DocHostUIHandler)
25 | #if (_WIN32_IE >= 0x0501)
26 | INTERFACE_PART(CDocHostSite, IID_IDocHostUIHandler2, DocHostUIHandler2)
27 | #endif
28 | INTERFACE_PART(CDocHostSite, IID_IInternetSecurityManager, InternetSecurityManager)
29 | INTERFACE_PART(CDocHostSite, IIDIServiceProvider, ServiceProvider)
30 | END_INTERFACE_MAP()
31 |
32 | CDocHostSite::CDocHostSite(COleControlContainer * pOCC, CBaseHtmlView* pView)
33 | : COleControlSite( pOCC )
34 | {
35 |     m_pView = pView;
36 | }
37 |
38 | CDocHostSite::~CDocHostSite()
39 | {
40 | }
41 |
42 | ULONG CDocHostSite::XDocHostShowUI::AddRef()
43 | {
44 |     METHOD_PROLOGUE(CDocHostSite, DocHostShowUI);
45 |
46 |     return pThis->ExternalAddRef();
47 | }
48 |
49 | ULONG CDocHostSite::XDocHostShowUI::Release()
50 | {
51 |     METHOD_PROLOGUE(CDocHostSite, DocHostShowUI);
52 |
53 |     return pThis->ExternalRelease();
54 | }
55 |
56 | HRESULT CDocHostSite::XDocHostShowUI::QueryInterface(REFIID riid, void ** ppvObj)
57 | {
58 |     METHOD_PROLOGUE(CDocHostSite, DocHostShowUI);
59 |
60 |     return pThis->ExternalQueryInterface( &riid, ppvObj );
61 | }
62 |
63 | HRESULT CDocHostSite::XDocHostShowUI::ShowHelp(HWND hwnd,
64 |     LPOLESTR pszHelpFile,
65 |     UINT nCommand,
66 |     DWORD dwData,
67 |     POINT ptMouse,
68 |     IDispatch * pDispatchObjectHit)
69 | {
70 |     METHOD_PROLOGUE(CDocHostSite, DocHostShowUI);
71 |
72 |     return E_NOTIMPL;
73 | }
74 |
75 | HRESULT CDocHostSite::XDocHostShowUI::ShowMessage(HWND hwnd,
76 |     LPOLESTR lpstrText,
77 |     LPOLESTR lpstrCaption,
78 |     DWORD dwType,
79 |     LPOLESTR lpstrHelpFile,
80 |     DWORD dwHelpContext,
81 |     LRESULT * pResult)

```

```

82 {
83     METHOD_PROLOGUE(CDocHostSite, DocHostShowUI);
84
85     return pThis->m_0->OnShowMessage(hwnd, lpstrText, lpstrCaption, dwType, lpstrHelpFile, dwHelpContext, plResult);
86 }
87
88 ULONG CDocHostSite::XDocHostUIHandler::AddRef()
89 {
90     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
91     return pThis->ExternalAddRef();
92 }
93
94 ULONG CDocHostSite::XDocHostUIHandler::Release()
95 {
96     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
97     return pThis->ExternalRelease();
98 }
99
100 HRESULT CDocHostSite::XDocHostUIHandler::QueryInterface(REFIID riid, void ** ppvObj)
101 {
102     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
103     return pThis->ExternalQueryInterface(&riid, ppvObj);
104 }
105
106 HRESULT CDocHostSite::XDocHostUIHandler::GetHostInfo(DOCHOSTUIINFO * pInfo)
107 {
108     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
109     return pThis->m_pView->OnGetHostInfo(pInfo);
110 }
111
112 HRESULT CDocHostSite::XDocHostUIHandler::ShowUI(DWORD dwID,
113     IOleInPlaceActiveObject * pActiveObject,
114     IOleCommandTarget * pCommandTarget,
115     IOleInPlaceFrame * pFrame,
116     IOleInPlaceUIWindow * pDoc)
117 {
118     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
119     return pThis->m_pView->OnShowUI(dwID, pActiveObject, pCommandTarget,
120     pFrame, pDoc);
121 }
122
123 HRESULT CDocHostSite::XDocHostUIHandler::HideUI()
124 {
125     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
126     return pThis->m_pView->OnHideUI();
127 }
128
129 HRESULT CDocHostSite::XDocHostUIHandler::UpdateUI()
130 {
131     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
132     return pThis->m_pView->OnUpdateUI();
133 }
134
135 HRESULT CDocHostSite::XDocHostUIHandler::EnableModeless(BOOL fEnable)
136 {
137     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
138     return pThis->m_pView->OnEnableModeless(fEnable);
139 }
140
141 HRESULT CDocHostSite::XDocHostUIHandler::OnDocWindowActivate(BOOL fEnable)
142 {
143     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
144     return pThis->m_pView->OnDocWindowActivate(fEnable);
145 }
146
147 HRESULT CDocHostSite::XDocHostUIHandler::OnFrameWindowActivate(BOOL fEnable)
148 {
149     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
150     return pThis->m_pView->OnFrameWindowActivate(fEnable);
151 }
152

```

```

153 HRESULT CDocHostSite::XDocHostUIHandler::ResizeBorder(LPCRECT prcBorder, 154 | IOleInPlaceUIWindow * pUIWindow,
155     BOOL fFrameWindow)
156 {
157     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
158     return pThis->m_pView->ResizeBorder( prcBorder, pUIWindow, fFrameWindow );
159 }
160
161 HRESULT CDocHostSite::XDocHostUIHandler::ShowContextMenu(DWORD dwID,
162     POINT * ppt,
163     IUnknown * pcmdtReserved,
164     IDispatch * pdispReserved)
165 {
166     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
167     return pThis->m_pView->ShowContextMenu( dwID, ppt, pcmdtReserved,
168     pdispReserved );
169 }
170
171 HRESULT CDocHostSite::XDocHostUIHandler::TranslateAccelerator(LPMSG lpMsg,
172     const GUID * pguidCmdGroup,
173     DWORD nCmdID)
174 {
175     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
176     return E_NOTIMPL;
177     return pThis->m_pView->OnTranslateAccelerator( lpMsg,
178     pguidCmdGroup, nCmdID );
179 }
180
181 HRESULT CDocHostSite::XDocHostUIHandler::GetOptionKeyPath(LPOLESTR * pchKey,
182     DWORD dw)
183 {
184     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
185     return pThis->m_pView->OnGetOptionKeyPath( pchKey, dw );
186 }
187
188 HRESULT CDocHostSite::XDocHostUIHandler::GetDropTarget(IDropTarget * pDropTarget,
189     IDropTarget ** ppDropTarget)
190 {
191     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
192     return pThis->m_pView->OnGetDropTarget( pDropTarget, ppDropTarget );
193 }
194
195 HRESULT CDocHostSite::XDocHostUIHandler::GetExternal(IDispatch ** ppDispatch)
196 {
197     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
198     return pThis->m_pView->OnGetExternal( ppDispatch );
199 }
200
201 HRESULT CDocHostSite::XDocHostUIHandler::TranslateUrl(DWORD dwTranslate,
202     OLECHAR * pchURLIn,
203     OLECHAR ** ppchURLOut)
204 {
205     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
206     return pThis->m_pView->OnTranslateUrl( dwTranslate, pchURLIn, ppchURLOut );
207 }
208
209 HRESULT CDocHostSite::XDocHostUIHandler::FilterDataObject(IDataObject * pDO,
210     IDataObject ** ppDORet)
211 {
212     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler);
213     return pThis->m_pView->OnFilterDataObject( pDO, ppDORet );
214 }
215
216 #if (_WIN32_IE >= 0x0501) // IE 5.5 and higher
217
218 ULONG CDocHostSite::XDocHostUIHandler2::AddRef()
219 {
220     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler2);
221
222     return pThis->ExternalAddRef();
223 }

```

```

224 | 225 | ULONG CDocHostSite::XDocHostUIHandler2::Release()
226 | {
227 |     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler2);
228 |     0
229 |     return pThis->ExternalRelease();
230 | }
231 | 写评论
232 | HRESULT CDocHostSite::XDocHostUIHandler2::QueryInterface(REFIID riid, void ** ppvObj)
233 | {
234 |     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler2);
235 |     目录
236 |     return pThis->ExternalQueryInterface(&riid, ppvObj);
237 | }
238 | 收藏
239 | HRESULT CDocHostSite::XDocHostUIHandler2::GetHostInfo(DOCHOSTUIINFO *)
240 | {
241 |     ASSERT(FALSE);
242 |     微博
243 |     return S_FALSE;
244 | }
245 | QQ
246 | HRESULT CDocHostSite::XDocHostUIHandler2::ShowUI(DWORD /*dwID*/,
247 |     IOleInPlaceActiveObject * /*pActiveObject*/,
248 |     IOleCommandTarget * /*pCommandTarget*/,
249 |     IOleInPlaceFrame * /*pFrame*/,
250 |     IOleInPlaceUIWindow * /*pDoc*/)
251 | {
252 |     ASSERT(FALSE);
253 |
254 |     return S_FALSE;
255 | }
256 |
257 | HRESULT CDocHostSite::XDocHostUIHandler2::HideUI()
258 | {
259 |     ASSERT(FALSE);
260 |
261 |     return S_FALSE;
262 | }
263 |
264 | HRESULT CDocHostSite::XDocHostUIHandler2::UpdateUI()
265 | {
266 |     ASSERT(FALSE);
267 |
268 |     return S_FALSE;
269 | }
270 |
271 | HRESULT CDocHostSite::XDocHostUIHandler2::EnableModeless(BOOL /*fEnable*/)
272 | {
273 |     ASSERT(FALSE);
274 |
275 |     return S_FALSE;
276 | }
277 |
278 | HRESULT CDocHostSite::XDocHostUIHandler2::OnDocWindowActivate(BOOL /*fActivate*/)
279 | {
280 |     ASSERT(FALSE);
281 |
282 |     return S_FALSE;
283 | }
284 |
285 | HRESULT CDocHostSite::XDocHostUIHandler2::OnFrameWindowActivate(BOOL /*fActivate*/)
286 | {
287 |     ASSERT(FALSE);
288 |
289 |     return S_FALSE;
290 | }
291 |
292 | HRESULT CDocHostSite::XDocHostUIHandler2::ResizeBorder(LPCRECT /*prcBorder*/,
293 |     IOleInPlaceUIWindow * /*pUIWindow*/,
294 |     BOOL /*fRameWindow*/)
295 | {

```

```

296     ASSERT(FALSE);
297
298     return S_FALSE;
299 }
300
301 HRESULT CDocHostSite::XDocHostUIHandler2::ShowContextMenu(DWORD /*dwID*/,
302     POINT* /*pptPos*/,
303     IUnknown* /*pComponent*/,
304     IDispatch* /*pDispatch*/,
305     {
306         ASSERT(FALSE);
307
308         return S_FALSE;
309     }
310
311 HRESULT CDocHostSite::XDocHostUIHandler2::TranslateAccelerator(LPMSG /*lpMsg*/,
312     const GUID* /*pCmdGroup*/,
313     DWORD /*nCmdID*/,
314     {
315         ASSERT(FALSE);
316
317         return S_FALSE;
318     }
319
320 HRESULT CDocHostSite::XDocHostUIHandler2::GetOptionKeyPath(BSTR* /*pbstrKey*/, DWORD)
321 {
322     ASSERT(FALSE);
323
324     return S_FALSE;
325 }
326
327 HRESULT CDocHostSite::XDocHostUIHandler2::GetDropTarget(IDropTarget* /*pDropTarget*/,
328     IDropTarget** /*ppDropTarget*/)
329 {
330     ASSERT(FALSE);
331
332     return S_FALSE;
333 }
334
335 HRESULT CDocHostSite::XDocHostUIHandler2::GetExternal(IDispatch** /*ppDispatch*/)
336 {
337     ASSERT(FALSE);
338
339     return S_FALSE;
340 }
341
342 HRESULT CDocHostSite::XDocHostUIHandler2::TranslateUrl(DWORD /*dwTranslate*/,
343     OLECHAR* /*pchURLIn*/,
344     OLECHAR** /*ppchURLOut*/)
345 {
346     ASSERT(FALSE);
347
348     return S_FALSE;
349 }
350
351 HRESULT CDocHostSite::XDocHostUIHandler2::FilterDataObject(IDataObject* /*pDO*/,
352     IDataObject** /*ppDORet*/)
353 {
354     ASSERT(FALSE);
355
356     return S_FALSE;
357 }
358
359 HRESULT CDocHostSite::XDocHostUIHandler2::GetOverrideKeyPath(LPOLESTR* pchKey, DWORD dw)
360 {
361     METHOD_PROLOGUE(CDocHostSite, DocHostUIHandler2);
362     return S_FALSE;
363     //return pThis->m_pView->OnGetOverrideKeyPath(pchKey, dw);
364 }
365 #endif
366

```



```

367 // InternetSecurityManager Methods
368 HRESULT FAR EXPORT CDocHostSite::XInternetSecurityManager
369 ::QueryInterface(REFIID riid, void** ppvObj)
370 {
371     0
372     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
373     HRESULT hr = (HRESULT)CDocHostSite->ExternalQueryInterface(&riid, ppvObj);
374     return hr;
375 }
376
377 ULONG FAR EXPORT CDocHostSite::XInternetSecurityManager::AddRef()
378 {
379     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
380     return pThis->ExternalAddRef();
381 }
382
383 ULONG FAR EXPORT CDocHostSite::XInternetSecurityManager::Release()
384 {
385     微博
386     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
387     return pThis->ExternalRelease();
388 }
389
390 HRESULT CDocHostSite::XInternetSecurityManager
391 ::SetSecuritySite (IInternetSecurityMgrSite *pSite)
392 {
393     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
394     return INET_E_DEFAULT_ACTION;
395 }
396
397 HRESULT CDocHostSite::XInternetSecurityManager
398 ::GetSecuritySite(IInternetSecurityMgrSite **ppSite)
399 {
400
401     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
402     return INET_E_DEFAULT_ACTION;
403 }
404
405 HRESULT CDocHostSite::XInternetSecurityManager
406 ::MapUrlToZone(LPCWSTR pwszUrl, DWORD *pdwZone, DWORD dwFlags)
407 {
408     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
409     return INET_E_DEFAULT_ACTION;
410 }
411
412
413 HRESULT FAR EXPORT CDocHostSite::XInternetSecurityManager
414 ::GetSecurityId(LPCWSTR pwszUrl,
415                BYTE *pbSecurityId,
416                DWORD *pcbSecurityId,
417                DWORD dwReserved)
418 {
419
420     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
421     return INET_E_DEFAULT_ACTION;
422 }
423
424
425 STDMETHODCALLTYPE CDocHostSite::XInternetSecurityManager
426 ::ProcessUrlAction( /* [in] */ LPCWSTR pwszUrl,
427                   /* [in] */ DWORD dwAction,
428                   /* [size_is][out] */ BYTE __RPC_FAR *pPolicy,
429                   /* [in] */ DWORD cbPolicy,
430                   /* [in] */ BYTE __RPC_FAR *pContext,
431                   /* [in] */ DWORD cbContext,
432                   /* [in] */ DWORD dwFlags,
433                   /* [in] */ DWORD dwReserved)
434 {
435
436     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
437

```

```

438     DWORD dwPolicy=URLPOLICY_ALLOW;439
440     BOOL bNoActiveX=FALSE;
441     BOOL bNoJava=FALSE;
442     BOOL bNoScript=FALSE 0
443     BOOL bCrossData=FALSE
444
445     if (dwAction <= URLACTION_ACTIVEX_MAX && dwAction >= URLACTION_ACTIVEX_MIN)
446         dwPolicy = bNoActiveX ? URLPOLICY_DISALLOW : URLPOLICY_ALLOW;
447     else if ((dwAction <= URLACTION_JAVA_MAX && dwAction >= URLACTION_JAVA_MIN) ||
448             URLACTION_HTML_JAVA_RUN == dwAction)
449         if (bNoJava)
450             dwPolicy = URLPOLICY_JAVA_PROHIBIT;
451         else
452             return INET_E_DEFAULT_ACTION;
453     else if (dwAction <= URLACTION_SCRIPT_MAX && dwAction >= URLACTION_SCRIPT_MIN)
454         dwPolicy = bNoScript ? URLPOLICY_DISALLOW : URLPOLICY_ALLOW;
455     // !! If the compiler cannot find URLACTION_CROSS_DOMAIN_DATA, make sure you are building with
456     // !! the latest version of the IE headers -- URLMON.H specifically -- from MSDN Downloads for the
457     // !! Web Workshop or the Platform SDK
458     else if (URLACTION_CROSS_DOMAIN_DATA == dwAction)
459         dwPolicy = bCrossData ? URLPOLICY_ALLOW : URLPOLICY_DISALLOW;
460     else
461         return INET_E_DEFAULT_ACTION;
462
463     if (cbPolicy >= sizeof (DWORD))
464     {
465         *(DWORD*) pPolicy = dwPolicy;
466         return S_OK;
467     }
468     else
469     {
470         return S_FALSE;
471     }
472 }
473
474 HRESULT CDocHostSite::XInternetSecurityManager
475 ::QueryCustomPolicy(LPCWSTR pwszUrl,
476                    REFGUID guidKey,
477                    BYTE **ppPolicy,
478                    DWORD *pcbPolicy,
479                    BYTE *pContext,
480                    DWORD cbContext,
481                    DWORD dwReserved)
482 {
483     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
484     return INET_E_DEFAULT_ACTION;
485 }
486
487 HRESULT CDocHostSite::XInternetSecurityManager
488 ::SetZoneMapping(DWORD dwZone,
489                 LPCWSTR lpszPattern,
490                 DWORD dwFlags)
491 {
492     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
493     return INET_E_DEFAULT_ACTION;
494 }
495
496 HRESULT CDocHostSite::XInternetSecurityManager
497 ::GetZoneMappings(DWORD dwZone,
498                  IEnumString **ppenumString,
499                  DWORD dwFlags)
500 {
501     METHOD_PROLOGUE(CDocHostSite, InternetSecurityManager)
502     return INET_E_DEFAULT_ACTION;
503 }
504
505
506 // ServiceProvider Methods
507 ULONG FAR EXPORT CDocHostSite::XServiceProvider::AddRef()
508 {
509     METHOD_PROLOGUE(CDocHostSite, ServiceProvider)

```

```

510     return pThis->ExternalAddRef();
511 }
512
513 ULONG FAR EXPORT CDocHostSite::XServiceProvider::Release()
514 {
515     METHOD_PROLOGUE(CDocHostSite, ServiceProvider)
516     return pThis->ExternalQueryInterface(&IID_IUnknown, (void**)0);
517 }
518
519 HRESULT FAR EXPORT CDocHostSite::XServiceProvider
520 ::QueryInterface(REFIID riid, void** ppvObject)
521 {
522     METHOD_PROLOGUE(CDocHostSite, ServiceProvider)
523     HRESULT hr = (HRESULT)CDocHostSite->ExternalQueryInterface(&riid, ppvObject);
524     return hr;
525 }
526
527 STDMETHODCALLTYPE CDocHostSite::XServiceProvider
528 ::QueryService(REFGUID guidService, REFIID riid, void** ppvObject)
529 {
530 }
531 {
532
533     if (guidService == SID_SInternetSecurityManager &&
534         riid == IID_IInternetSecurityManager)
535     {
536         METHOD_PROLOGUE(CDocHostSite, ServiceProvider)
537         HRESULT hr = (HRESULT)pThis->ExternalQueryInterface(&riid, ppvObject);
538         return hr;
539     }
540     else
541     {
542         *ppvObject = NULL;
543     }
544     return E_NOINTERFACE;
545 }
546 }

```

## CMyOleDropTarget类的实现

其次，我们还需要实现一个类:CMyOleDropTarget

其申明如下：

```

1  #pragma once
2
3  #include <afxole.h>
4
5  // 拖动类型
6  #define DRAGTYPE_NONE 0
7  #define DRAGTYPE_FILE 1 // 文件
8
9  class CMyOleDropTarget : public COleDropTarget
10 {
11     DECLARE_DYNAMIC(CMyOleDropTarget)
12     DECLARE_INTERFACE_MAP()
13     DECLARE_MESSAGE_MAP()
14
15 public:
16     CMyOleDropTarget();
17     virtual ~CMyOleDropTarget();
18
19 protected:
20     LPDROPTARGET m_pIEDropTarget;
21
22 public:
23     void SetIEDropTarget(LPDROPTARGET pDropTarget);
24     void ReleaseIEDropTarget();
25 }

```

```

26 public:
27     BEGIN_INTERFACE_PART(MyDropTarget, IDropTarget)
28     INIT_INTERFACE_PART(CMyDropTarget, MyDropTarget)
29     STDMETHOD(DragEnter)(THIS, COleObject, DWORD, POINTL, LPDWORD);
30     STDMETHOD(DragOver)(THIS, COleObject, DWORD, POINTL, LPDWORD);
31     STDMETHOD(DragLeave)(THIS);
32     STDMETHOD(Drop)(THIS, LPDATA1, COleObject, DWORD, POINTL pt, LPDWORD);
33     END_INTERFACE_PART(MyDropTarget, IDropTarget)
34 };

```

[目录](#)

实现部分为：

[收藏](#)
[微信](#)
[微博](#)
[QQ](#)

```

1  #include "stdafx.h"
2  #include "MyOleDropTarget.h"
3
4
5  IMPLEMENT_DYNAMIC(CMyOleDropTarget, COleDropTarget)
6
7  BEGIN_MESSAGE_MAP(CMyOleDropTarget, COleDropTarget)
8  END_MESSAGE_MAP()
9
10 BEGIN_INTERFACE_MAP(CMyOleDropTarget, COleDropTarget)
11 INTERFACE_PART(CMyOleDropTarget, IID_IDropTarget, MyDropTarget)
12 END_INTERFACE_MAP()
13
14 CMyOleDropTarget::CMyOleDropTarget()
15 {
16     m_pIEDropTarget = NULL;
17 }
18
19 CMyOleDropTarget::~CMyOleDropTarget()
20 {
21     ReleaseIEDropTarget();
22 }
23
24 void CMyOleDropTarget::SetIEDropTarget(LPDROPTARGET pDropTarget)
25 {
26     if (m_pIEDropTarget)
27         ReleaseIEDropTarget();
28     m_pIEDropTarget = pDropTarget;
29     m_pIEDropTarget->AddRef();
30 }
31
32 void CMyOleDropTarget::ReleaseIEDropTarget()
33 {
34     if (!m_pIEDropTarget)
35         return;
36     m_pIEDropTarget->Release();
37     m_pIEDropTarget = NULL;
38 }
39
40 STDMETHODIMP_(ULONG) CMyOleDropTarget::XMyDropTarget::AddRef()
41 {
42     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
43     return pThis->ExternalAddRef();
44 }
45
46 STDMETHODIMP_(ULONG) CMyOleDropTarget::XMyDropTarget::Release()
47 {
48     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
49     return pThis->ExternalRelease();
50 }
51
52 STDMETHODIMP CMyOleDropTarget::XMyDropTarget::QueryInterface(
53     REFIID iid, LPVOID* ppvObj)
54 {
55     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
56     return pThis->ExternalQueryInterface(&iid, ppvObj);
57 }
58

```

```

59 | STDMETHODCALLTYPE CMyOleDropTarget::XMyDropTarget::DragEnter(THIS_ LPDATAOBJECT lpDataObject, 60 | DWORD dwKeyState, POINTL pt
61 | {
62 |     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
63 |     ASSERT_VALID(pThis);
64 |
65 |     if(pThis->m_pIEDropTarget)
66 |         pThis->m_pIEDropTarget->DragEnter(lpDataObject, dwKeyState, pt, pdwEffect);
67 |     return pThis->m_pIEDropTarget->DragEnter(lpDataObject, dwKeyState, pt, pdwEffect);
68 | }
69 |
70 | STDMETHODCALLTYPE CMyOleDropTarget::XMyDropTarget::DragOver(THIS_ DWORD dwKeyState,
71 |     POINTL pt, LPDWO Effect)
72 | {
73 |     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
74 |     ASSERT_VALID(pThis);
75 |
76 |     if(pThis->m_pIEDropTarget)
77 |         pThis->m_pIEDropTarget->DragOver(dwKeyState, pt, pdwEffect);
78 |     return pThis->m_pIEDropTarget->DragOver(dwKeyState, pt, pdwEffect);
79 | }
80 |
81 | STDMETHODCALLTYPE CMyOleDropTarget::XMyDropTarget::DragLeave(THIS)
82 | {
83 |     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
84 |     ASSERT_VALID(pThis);
85 |
86 |     if(pThis->m_pIEDropTarget)
87 |         pThis->m_pIEDropTarget->DragLeave();
88 |     return pThis->m_pIEDropTarget->DragLeave();
89 | }
90 |
91 | STDMETHODCALLTYPE CMyOleDropTarget::XMyDropTarget::Drop(THIS_ LPDATAOBJECT lpDataObject,
92 |     DWORD dwKeyState, POINTL pt, LPDWO pdwEffect)
93 | {
94 |     METHOD_PROLOGUE_EX(CMyOleDropTarget, MyDropTarget)
95 |     ASSERT_VALID(pThis);
96 |
97 |     if(pThis->m_pIEDropTarget)
98 |         pThis->m_pIEDropTarget->Drop(lpDataObject, dwKeyState, pt, pdwEffect);
99 |     return pThis->m_pIEDropTarget->Drop(lpDataObject, dwKeyState, pt, pdwEffect);
100 | }

```

到目前为止，作为ActiveX小白的我，其实也没有彻底理解上述两个类的具体实现和作用，只知道其实现了对IE浏览器的定制和扩展。不过没关系，先用起来，后面慢慢对于相关知识，我转载了一篇文章：[浏览器的定制和扩展](#)，可能对我们的理解会有一定的帮助。

## CBaseHtmlView中的相关函数实现

有了CDocHostSite 和 CMyOleDropTarget类。我们当然还需要在CBaseHtmlView中实现相应的函数。

我们定义了一个CMyOleDropTarget类的成员变量 m\_DropTargat, 同时实现了5个函数，声明如下：

```

1 | /*****
2 | ***** Javascript call MFC 相关函数*****
3 | *****/
4 | //JavaScript call MFC 的几个相关函数
5 | CMyOleDropTarget m_DropTarget;
6 | virtual BOOL CreateControlSite(COleControlContainer* pContainer, COleControlSite** ppSite,
7 |     UINT nID, REFCLSID clsid);
8 | virtual HRESULT OnGetDropTarget(LPDROPTARGET pDropTarget, LPDROPTARGET* ppDropTarget);
9 | // 定制宿主信息
10 | virtual HRESULT OnGetHostInfo(DOCHOSTUIINFO * pInfo);
11 | // 设置脚本扩展
12 | virtual HRESULT OnGetExternal(LPDISPATCH *lppDispatch);
13 | // MesseBox
14 | virtual HRESULT OnShowMessage(HWND hwnd, LPOLESTR lpstrText, LPOLESTR lpstrCaption,
15 |     DWORD dwType, LPOLESTR lpstrHelpFile, DWORD dwHelpContext, LRESULT * pResult);

```

相关实现：

### 1、CreateControlSite:

```

1 | BOOL CBaseHtmlView::CreateControlSite(COleControlContainer* pContainer,
2 |                                     COleControlSite** ppSite, UINT nID, REFCLSID clsid)
3 | {
4 |     *ppSite = new CDocSite(pContainer, this); // 创建自己的控制站点实例
5 |     return (*ppSite) != NULL ? S_OK : S_FALSE;
6 | }

```

### 2、OnGetDropTarget

```

1 | HRESULT CBaseHtmlView::OnGetDropTarget(LPDROPTARGET pDropTarget, LPDROPTARGET* ppDropTarget)
2 | {
3 |     m_DropTarget.SetDropTarget(pDropTarget);
4 |
5 |     LPDROPTARGET pMyDropTarget;
6 |     pMyDropTarget = (LPDROPTARGET)m_DropTarget.GetInterface(&IID_IDropTarget);
7 |     if (pMyDropTarget)
8 |     {
9 |         *ppDropTarget = pMyDropTarget;
10 |         pMyDropTarget->AddRef();
11 |         return S_OK;
12 |     }
13 |
14 |     return S_FALSE;
15 | }

```

### 3、OnGetHostInfo :

```

1 | HRESULT CBaseHtmlView::OnGetHostInfo(DOCHOSTUIINFO * pInfo)
2 | {
3 |     pInfo->cbSize = sizeof(DOCHOSTUIINFO);
4 |     pInfo->dwFlags = DOCHOSTUIFLAG_THEME | DOCHOSTUIFLAG_NO3DBORDER;
5 |     pInfo->dwDoubleClick = DOCHOSTUIDBLCLK_DEFAULT;
6 |
7 |     return S_OK;
8 | }

```

### 4、OnGetExternal :

```

1 | HRESULT CBaseHtmlView::OnGetExternal(LPDISPATCH *lppDispatch)
2 | {
3 |     *lppDispatch = GetIDispatch(TRUE);
4 |     return S_OK;
5 | }

```

### 5、OnShowMessage :

```

1 | HRESULT CBaseHtmlView::OnShowMessage(HWND hwnd, LPOLESTR lpstrText, LPOLESTR lpstrCaption, DWORD dwType, LPOLESTR lpstrHelpFile, DWORD dwHelpContext,
2 | {
3 |     // 窗口标题"Microsoft Internet Explorer"的资源标识
4 | #define IDS_MESSAGE_BOX_TITLE 2213
5 |     // 载入Shdoclc.dll 和IE消息框标题字符串
6 |     HINSTANCE hinstSHDOCLC = LoadLibrary(TEXT("SHDOCLC.DLL"));
7 |     if (hinstSHDOCLC == NULL)

```

```
8         return S_FALSE; 9
10     CString strBuf, strCaption(lpstrCaption);
11     strBuf.LoadString( tSHDocLC, IDS_MESSAGE_BOX_TITLE);
12
13     // 比较IE消息框标题和lpstrCaption
14     // 如果相同, 用白 替换
15     if (strBuf == lpstrCaption)
16         strCaption = L" HTMLVIEW";
17
18     // 创建自己的消息 显示
19     *pIResult = Message(CString(lpstrText), strCaption, dwType);
20
21     // 卸载Shdoclc.dll
22     FreeLibrary(hinstLC);
23
24     return S_OK;
25 }
```



## JavaScript消息的声明

在实现了上面的函数和类后，我们就可以利用MFC宏，通过消息传到，实现JavaScript调用MFC了。

首先在头文件中申明消息宏：

DECLARE\_DISPATCH\_MAP() // 构建dispatch映射表以暴露方法或属性

然后在CPP 中声明 消息宏：

```
1 BEGIN_DISPATCH_MAP(CDemoHtmlView, CBaseHtmlView)
2     DISP_FUNCTION(CDemoHtmlView, "CloseWindow", CloseWindow, VT_EMPTY, VTS_NONE)
3     DISP_FUNCTION(CDemoHtmlView, "MinimizeWindow", MinimizeWindow, VT_EMPTY, VTS_NONE)
4     DISP_FUNCTION(CDemoHtmlView, "ClickCaption", ClickCaption, VT_EMPTY, VTS_WBSTR)
5     DISP_FUNCTION(CDemoHtmlView, "slideIn", SetSlideIn, VT_EMPTY, VTS_I2)
6     DISP_FUNCTION(CDemoHtmlView, "getLineData", getLineData, VT_EMPTY, VTS_WBSTR)
7     DISP_FUNCTION(CDemoHtmlView, "getPieData", getPieData, VT_EMPTY, VTS_WBSTR)
8     DISP_FUNCTION(CDemoHtmlView, "getThisDayData", getThisDayData, VT_EMPTY, VTS_WBSTR)
9 END_DISPATCH_MAP()
```

在DISP\_FUNCTION中5个参数分别为：类名、JavaScript中调用MFC的函数名、响应函数名、返回类型、参数类型

以getLineData为例，JavaScript调用方式为：

```
window.external.getLineData(sendData);
```

其无返回值，传入参数为string，通常为JSON格式的字符串。

为此，我们需要实现一个响应函数：void getLineData(LPCTSTR lpszLineData); //折线图

```
1 void CDemoHtmlView::getLineData(LPCTSTR lpszLineData)
2 {
3     ::PostMessage(GetParent()->GetSafeHwnd(), WM_MAINJS_LineData, (LPARAM)lpszLineData, NULL);
4     return;
5 }
```

在此实现中，我们直接向主对话框发送自定义消息WM\_MAINJS\_LineData，让主对话框进行处理。考虑到阻塞和效率问题，通常情况下我们不直接在上面响应函数中直消息形式，让后台去处理。为此，我们需要自己实现自定义消息的响应。这里就不进行介绍了。

## 一个小技巧

在上面，细心的读者可能发现，我们的类名不是CBaseHtmlView，变成CDemoHtmlView了，其实这里是我们的一个小技巧。

我们再写了一个CDemoHtmlView类，继承自CBaseHtmlView，我们将上述消息在此类中实现，同时我们也将主对话框类中定义的CBaseHtmlView

类声明如下：

```
1 #pragma once
2
3 #ifndef _WIN32_WCE
4 #error " CHtmlView is not supported for Windows CE."
5 #endif
6 #include "BaseHtmlView.h"
7 // CHtmlView html view
8
9 class CHtmlView : public BaseHtmlView
10 {
11     DECLARE_DYNCREATE(CHtmlView)
12     DECLARE_MESSAGE_MAP()
13     DECLARE_DISPATCH_MAP() // 构建dispatch映射表以暴露方法或属性
14 public:
15     CHtmlView(); // protected constructor used by dynamic creation
16     virtual ~CHtmlView();
17
18 protected:
19     virtual void OnInitialUpdate();
20 public:
21     void CloseWindow(); // 关闭窗口
22     void MinimizeWindow(); // 最小化窗口
23     void ClickCaption(LPCTSTR _bstr); // 拖动窗口
24     void SetSlideIn(short iShort); // 右侧缩放
25
26     void getLineData(LPCTSTR lpszLineData); // 折线图
27     void getPieData(LPCTSTR lpszPieData); // 饼状图
28     void getThisDayData(LPCTSTR lpszDayData); // MONITOR 页面数据渲染
29     virtual void OnDocumentComplete(LPCTSTR lpszURL);
30 };
31
32
```

可以看到，我们除了实现JavaScript调用MFC的方法外，还特意继承了CBaseHtmlView的两个虚函数：OnInitialUpdate() 和 OnDocumentComplete。

OnInitialUpdate() 函数 实现 html的加载， OnDocumentComplete函数 是加载完成后的通知。

这样我们就完全实现了对CBaseHtmlView的封装，以后添加相关html页面，我们也只需要添加一个继承自CBaseHtmlView 的具体类，并实现上述函数就行，完全不用对CBaseHtmlView进行任何修改。

## 总结

好了，通过我的前两篇文章：MFC对话框里利用CHtmlView加载界面（一）————加载网页、MFC对话框里利用CHtmlView加载界面（二）——和本文的介绍，我们就实现了MFC 和 JavaScript的互相调用。此外在MFC对话框里利用CHtmlView加载界面（四）————分层窗口设置中，我框背景，完全将HTML作为界面的方法。根据此四篇文章，我们就基本掌握了一种以MFC对话框为基础的，实现HTML界面和交互的方法。

本文的示范代码下载路径为：[http://download.csdn.net/detail/qq\\_20828983/9900343](http://download.csdn.net/detail/qq_20828983/9900343)

文章标签：

CHtmlView

JavaScript

MFC

windows

html

个人分类：

MFC

MFC 和 HTML 交互

## MFC 对话框加载菜单

step1: 在Resource View里添加一个菜单,假设ID为:IDR\_MU\_MAIN step2: 在对话框的初始化内 BOOL CLPC1114UpdaterDlg::OnInitDialog...

想对作者说点什么？

我来说一句



## 【win32 app调用html方法】解决VS2008中CHtmlView函数OnBeforeNavigate2不能禁止跳转的BUG

VS2008中CHtmlView的OnBeforeNavigate2函数有个莫名其妙的问题,竟然不能实现禁止导航。函数定义如下: void CHtmlView::OnBeforeNavigate...

 pirate97    2012-03-13 10:46:25    0    数：1491

老公自从吃了它 每晚要我好几次!

美思源 · 顶新

写评论

目录

## LOADING页面小技巧

给一段简单的在读取某个时间较长网页时显示一个简单的LOADING信息的办法 Loading.....

 beiouwolf    2008-06-05 12:38:00    微信    数：1864

## 处理Menu功能菜单及"loading"对话框

一般的Android手机上都会有Menu键，我们怎么来处理Menu的菜单项和它的点击事件呢？下面这个例子就示范了“关于”对话框和“离开”等功能。要为菜单添加菜单项...

 li0531    2013-07-21 10:21:45    QQ    : 932

## <em>MFC</em>显示<em>加载</em>等待图标,显示gif动画.rar

<em>MFC</em>实现显示GIF动画,解决类似不知道进度比例,需要等待的状态。很值得下载。包内容: 1.<em>Loading</em>的系列gif动画集合; 2.显示GIF动画原理 3.显示下载    2018年05月07日 00:00

## 仿微信，自定义加载中透明框

先看效果图吧...

 q59521038    2014-09-30 11:42:21    阅读数：1059

老公自从吃了它 每晚要我好几次!

美思源 · 顶新

## C# winfrom读取数据时候出现的等待<em>loading</em>界面

在应用程序进行数据<em>加载</em>的时候出现<em>loading</em>等待界面,...<em>MFC</em>中软件开启等待界面的实现方法 Android实现长时间...c#等待提示对话框上传...

下载    2018年05月09日 00:00

## 详解MFC框架窗口、文档和视图

尽管窗口、文档和视图是MFC的基础，但可能也是最不容易理解的部分，因为其概念比传统编程所需要的Windows函数更强一些，因此，须在本章做进一步详细讨论框技巧。6.1框架...

 mail\_cm    2012-01-09 16:38:40    阅读数：4125

## LoadLibrary加载动态库失败的解决办法

方式一 采用LoadLibraryEx 转载：http://blog.sina.com.cn/s/blog\_62ad1b8101017qub.html 若DLL不在调用方的同一目录下，可以用LoadLibraryEx...

 segen\_jaa    2013-01-28 17:29:01    阅读数：50598

## 【Android】Android开发之常用的loading等待效果实现，仿微博等待动画。两种实现方式详解

作者：程序员小冰（转载请说明出处）博客地址：http://blog.csdn.net/qq\_21376985 长期维护的Android项目，里面包括常用功能实现，以及知识点详解，当然还有Java...

 fengyeNom1    2017-06-28 15:47:12    阅读数：991

## 提高页面加载速度支一招

ControlJS官网:http://stevesouders.com/controljs/ 下面是转自http://www.cnblogs.com/tiwinlin/archive/2011/12/12...

 pipi0714    2012-10-17 09:32:57    阅读数：3701

## 自定义ProgressBar的加载效果

三种方式实现自定义圆形页面加载中效果的进度条 To get a ProgressBar in the default theme that is to be used on white/light ...

 gundumw100

2017-04-10 18:41:50

阅读数：429

MFCHtmltest

2017年07月16日582KB

下载

写评论



老司机教你一个数学

目录


懂天下英语

收藏

0

### MFC对话框里利用CHtmlView加载界面（四）——分层窗口设置

在学会了 利用CHtmlView 在MFC对话框中显示网页，并支持JavaScript 和 MFC 之间的互相通信后。当然就想：我们为什么不能只用HTML 作为前端界面，而底层用MFC来写业务逻辑呢？

 qq\_20828983

2017-05-23 18:41

阅读数：739

微信

微博

### MFC界面--利用CHtmlView加载HTML制作新风格的界面（包括CView和Dialog）

一、文档视图形式，以一个视图cview（或cdialog）作为首页界面 转自：<http://liuxiang031130.blog.163.com/blog/static/117666552011...>

 WaitForFree

2013-05-19 10:41:51

阅读数：9772

### MFC中CHtmlView和JS的相互调用

没废话，直接上步骤：1、创建一个继承自CHtmlView

 AndyCoder

2014-05-15 11:20:27

阅读数：1055

### 在CHtmlView中，判断一个页面加载完成的准确方法,解决OnDocumentComplete多次调用问题

在CHtmlView中，判断一个页面加载完成的准确方法 凌晨了，但是关于这个问题一定要记录下来，要不然以后不记得了，还得走弯路。今天在对一个小系统做一个测试C写了...

 z331719654

2014-02-25 22:28:21

阅读数：2844

### MFC对话框里利用CHtmlView加载界面（一）——加载网页

简单了解下CHtmlView类，CHtmlView类是MFC类，它继承自CView，属于MFC视图类。但我们看它的具体实现时，我们会发现其最大的本质在于IWebBrowser2（WebBrowserInterface）


 qq\_20828983

2017-07-10 08:44:56

阅读数：488

短信接口验证码

如何实现全国全网验证码类短信接口的调用



### 使用MFC的CHtmlView创建浏览器控件

```
PlazaBrowser.h #ifndef PLAZA_BROWSER_HEAD_FILE #define PLAZA_BROWSER_HEAD_FILE #pragma once #include<...>
```


 qq125096885

2016-05-18 18:03:22

阅读数：2143

### 20150622 CEF嵌入MFC对话框程序

最近在学习CEF，发现自己的编程能力实在太弱，看不懂应该怎么使用这个库，也不知道可以向谁请教，尽管官方说提供的cefclient示例程序已经很清楚了啊，但是我看的十分痛苦。最...

 w1reg

2015-06-22 23:37:21

阅读数：1562

### Qt界面动态库被MFC对话框调用

2016年08月01日5.43MB

下载

### MFC中用CHtmlView作的浏览器

2008年07月03日1.87MB

下载

### MFC界面基于单文档调用对话框

2010年09月17日

11.23MB

下载



教你1招有效防止Unity代码被反编译！

无需手动加密Assen，防止反编译代码，自动编译mono，防止反编译

写评论

### CHTMLDome1

2017年07月10日

130KB

下载

目录

收藏

### 在VC对话框中用ChtmlView显示"HTML"

导读：微软的MFC在Visual Studio 6.0中增加了一个新类CHtmlView，利用这个类，我们可以实现在基于文档视图结构的程序中显示HTML文件。但是它是否可以用来在对话框中显示HTML文件呢？

zys123456

2007-07-04 10:02:00

阅读数：521

微博

### MFC 对话框里面编辑HTML

2012年05月24日

34.11MB

下载

QQ

### VC6下CHtmlView中最简单最全面的程序与网页交互方法

简单来说，终极目标-----VC6和网页相互调用对方的数据和方法；而调用方法时重点要捕获返回值。VC6下远没有VC7及其之后提供的操作丰富方便。一些深度应用的解释...

liehuoee

2012-03-23 09:05:39

阅读数：2370

### CHTMLDome2

2017年07月11日

645KB

下载



便宜云虚拟主机

云服务器和云虚拟主机哪个更好

### MFC调用其它DLL的资源中包含自定义控件，显示不正常的解决方法

我遇到的问题：我又一个A.exe调用B.dll中的对话框，对话框中有一个自定义控件。对话框在DoModal()时没有任何反应，OnInitDialog也没走。跟了一遍DoModal，猜测是资源中包含自定义控件的问题。

u012789356

2017-05-02 11:22:08

阅读数：262

### MFC对话框里利用CHtmlView加载界面（二）————MFC调用JavaScript

在MFC对话框里利用CHtmlView加载界面（一）中，对如何利用CHtmlView加载网页进行了介绍。本文，我们将在上一篇文章的基础上，对如何利用CHtmlView加载HTML文件进行介绍。

qq\_20828983

2017-07-11 08:50:30

阅读数：490

### MFC Tab Control 多层 嵌套 子对话框

以下教程需要先在鸡啄米的MFC教程中学习Tab Control的基本知识，才可学习。资源网址：http://download.csdn.net/detail/linyijiong/97932401...

linyijiong

2017-03-24 22:33:01

阅读数：1001

### 基于对话框的MFC图形界面程序的选择文件读取文件

上一篇文章，我们写了我们可能第一个用C++写的图形界面程序，现在我们要实现我们可能最需要的一个功能，从电脑的文件中读取文件的操作。实际上非常简单 下面按钮的监听操作。（）还...

weng126weng

2016-03-04 14:22:28

阅读数：2594

### MFC中CHtmlEditCtrl类与CHtmlView类的使用

工作中需要用到标题中的两个类，有很多不明白之处，在网上查找了些资料，整理后写了两个函数：1、CHtmlView类加载内存中的HTML文本函数(CMyHtmlCtrl是从C CHtmlEditCtrl类派生出来的)

tg2003

2009-11-11 03:37:00

阅读数：1086



### 程序员不会英语怎么行？

老司机教你一个数字 懂天下英语

### 如何调用MFC dll 来显示其对话框

一、dll部分 同过VS向导建立一个MFC工程，在工程中创建自己的MFC类，来实现将要展示的界面。新建一个cpp工程，添加头文件#include "StdAfx.h"和界面类...  
hd498222594 2011-12-01 14:00 阅读量：2307

### 基于MFC对话框工程如何使用单一工程下的对话框

1、拷贝所需要的对话框资源 在VS2010其他版本中打开现有的解决方案，然后在左上角文件-->打开按钮，打开你想调用的对话框所属工程的中后缀为.rc的文件Tes...  
cwt19902010 2016-11-11 16:25 阅读量：1081

### MFC几种给对话框添加背景图片方法

1，目标 总结几种给MFC对话框添加背景图片的方法。2，实现 方式一：（适用与多种格式）用CImage类可以加载多种格式的图片，在OnPaint中Draw即可。 void CB...  
dpsying 2013-12-12 21:18:43 阅读量：19027

### MFC基于对话框程序界面之皮肤美化(Skin Magic)--附案例源码

由于项目原因，需要对MFC界面进行美化，然后问大叔和度娘要了相关资源，以下主要是从实用角度介绍Skin Magic这个皮肤的其中一种用法。资源列表如下（楼主自个）。 ...  
CImage 2013-09-16 23:59:02 阅读量：4833

### MFC中软件开启等待界面的实现方法

2017年07月14日 29.71MB 下载



### 短信接口验证码

网站短信验证码接口

### 使用CHtmlView制作浏览器的简单记录

这几天做了一个简易浏览器，其中碰到了一些问题，在这里记录下来，以供之后查阅。 1、简单的浏览器一些基本功能，例如回到主页、向前、后退、刷新、停止，都可...  
hrxd1270 2014-03-12 23:18:45 阅读量：2937

### MFC 界面线程和工作线程--非模态对话框

每个系统都有线程，而线程的最重要的作用就是并行处理，提高软件的并发率。针对界面来说，还能提高界面的响应力。 线程分为界面线程和工作线程，界面实际就...  
xiuxiaoman 2013-09-17 14:13:31 阅读量：1606

### VS2005中在mfc对话框中嵌入一个matlab的figure图形

2015年05月08日 7.91MB 下载

### MFC 自定义模态等待对话框（费时操作等待）

//////////////////////////////////////随手的一个简单代码 //CMainDlg.cpp 简化代码如下。 CWaitDlg \*Waitdalg=NULL; CW...  
zhao\_yi\_ming 2016-09-20 15:03:43 阅读量：1403

### MFC C++监听网页窗口的点击超链接跳转问题(即打开新的浏览器窗口)

使用MFC的CHtmlView类，注册ON\_EVENT事件进行监听。 实现的基本代码是: BEGIN\_EVENTSINK\_MAP(CxxxView, CHtmlView) ON\_EVENT(Cx...  
pengmutian 2015-06-10 13:31:10 阅读量：1649



### 50万码农评论：英语对于程序员有多重要！

不背单词和语法，老0你一个数学公式秒懂天下英语

### MFC不同dll中对话框的调用

碰到一个问题，在不同dll中对话框的调用，报错，调试发现在OnInitDialog()的时候系统死掉，网上一查才知道还是MFC了解的太少。 网上提供了2种方法，一是将DL改...

a5476466661 2016-11-16 11:20 阅读数：744

### MFC 下调用Qt编写的含界面dll

接到任务要做一个既可以在纠结了好几天，

libin88211 2014-07-27 23:23:57 阅读数：5008

### MFC不同工程（解决方案）中对话框资源的复制与重用方法

项目是一个基于MFC的VS2013工程A，工程A实现一个通过GPRS网络接收一个传感器系统采集的数据并存储的功能。最近需要添加一个带有温度计、罗盘等控件的对话框善，而新控件以前并...

lihui126 2015-05-07 09:53:15 阅读数：2808

### MFC 对话框中动态加载OCX控件

以前在MFC中创建OCX控件后，再到对话框中加载用的都是看似简单但实际上最麻烦的办法： 那就是在对话框上右键->insert ActiveX control....然后给插入的控件绑

yumulinxiang 2017-10-29 15:42:57 阅读数：857

### 基于MFC的对话框程序加启动进度条

最终效果 开始启动画面 主对话框初始化完成之后(sleep模拟)显示出主对话框 实现过程 以前写的、包括网上很多添加启动画面都是阻塞型的，作用只是好看..... 针对比转以为了让...

ZengDong\_1991 2016-09-03 14:51:27 阅读数：2358



订单管理系统  
EasyBiz订单管理软件

### MFC中非模态对话框去刷新当前视图

我创建了一个SDI，在wizard中我选择了不支持 文档/视图结构。之后我产生了一个非模态对话框，我想按下非模态对话框中的按钮后，就在视图区域显示一些文字，请S...

shiluankuang5671 2017-01-19 09:56:15 阅读数：344

### 2步让Win32应用程序/控制台应用使用MFC类库支持显示对话框

Win32应用程序/控制台应用使用MFC类库 第一步、以VS2013为例，新建2个控制台工程，prj1(不勾选MFC)和prj2(勾选MFC) 可以发现在prj2的stdafx.h中关于支...

qq851914012 2015-05-09 16:20:33 阅读数：1407

没有更多推荐了，[返回首页](#)

#### 个人资料



白白皎皎

关注

原创	粉丝	喜欢	评论
21	7	2	2

等级： 博客 2  
访问：2万+  
积分：439  
排名：11万+

0

写评论

目录

收藏

微信

微博

QQ

所在国家 +0086 中国内地(China)

手机号码 157

验证码 27978 重新发送(7)

用户名 推送移动

设置密码

确认密码

短信接口验证码

最新文章

C++ 字符、字符串大小写的转换

通配符字符串匹配

TCP的连接和断开

C++ 类型隐式转换学习

sln、db、opendb、vcxproj、filters、user 文件跟踪说明

个人分类

MFC9篇

c++基础4篇

MFC 和 HTML 交互5篇

网络编程3篇

windows8篇

展开

归档

2017年9月2篇

2017年8月2篇

2017年7月10篇

2017年5月2篇

2017年3月4篇

展开

热门文章

SQL语句多表联合查询  
阅读量：10150

MFC动态绘图+OpenCV画图  
阅读量：2188

C++模板类链表实现  
阅读量：1088

我对内核对象的理解  
阅读量：799

MFC对话框里利用CHtmlView加载界面（四）——分层窗口设置  
阅读量：736

最新评论

C++模板类链表实现  
c2436477256：第287行“if (lift > right)”，lift和right是T类型，不能...

我对内核对象的理解  
qingzai\_：博主讲解的不错，谢谢分享