C++集成地图功能方法介绍

**工具**：VS2015（或VS2012，demo是使用VS2012写的）

**语言**：C++、HTML、JavaScript

**平台**：百度地图API、高德地图API

**实现方式**：有两种方式可以实现均采用MFC应用程序实现

1、新建基于对话框的MFC应用，使用HTML对话框；（该实现方式支持高德地图，不支持百度地图）

2、新建基于对话框的MFC应用，使用普通对话框，添加web browser控件显示地图（**推荐使用：**该实现方式同时支持高德地图、百度地图，且相对较第一种方式实现起来更容易控制）。

**实现步骤**：（采用第二种实现方式）

1、新建基于对话框的MFC应用；

2、添加Web Browser控件（右键—>插入Active X控件—>Microsoft Web Browser）；

3、设置控件属性、添加控件变量（例如变量名为browser）；

4、编写HTML文件，引导、展示地图（可参考百度地图API或高德地图API开发文档）；

5、为browser设置页面，设置方式如下：

browser.Navigate(\_T("file:///C:file/test.html"),NULL,NULL,NULL,NULL);

6、编写JavaScript文件，实现与地图相关的业务代码；

7、C++代码与JS代码相互调用，实现功能。

**主要技术**：

地图功能其实是调用了百度地图、高德地图或谷歌地图等平台的JavaScript API接口，通过JS代码实现地图有关业务，再使用C++与JS交互的接口方法，使C++代码与JS代码相互调用，完成该部分功能。

**1、百度地图API**（http://lbsyun.baidu.com/）；

**2、JavaScript**；

**3、C++调取JavaScript函数的接口方法**（WebPage类）；

1、将webpage源码导入工程，并在对应头文件中添加声明

2、使用示例代码

CWebPage web;

web.SetDocument(browser.get\_Document());

const CString funcName("test\_JS");//test\_JS为函数名

CComVariant varResult;

web.CallJScript(funcName,NULL,NULL, &varResult);//varResult为JS执行后返回值，可以在“NULL”位置附带任意数量的参数

资料参见同目录下文件：webPage.zip

参考网址：

<https://blog.csdn.net/tangyin025/article/details/8274519>

<https://blog.csdn.net/zt_xcyk/article/details/79729532>

**4、JavaScript调用C++函数的方法**（主要用于点击地图获取对应坐标值）

使用IDispatch接口作为该部分的主要技术支撑，项目中以轻量的方式使用IDispatch技术，若需要实现功能较为复杂或可维护性高的代码，需要围绕IDispatch对代码进行重构。

该项目中，IDispatch用法步骤如下：

1、使MFC生成的对话框类CxxDlg继承IDispatch

2、在CxxDlg中实现需要使用JS调用的C++方法

3、在头文件CxxDlg.h中声明与IDispatch有关的7个方法（见后文代码说明）

4、在源文件CxxDlg.cpp文件中实现与IDispatch有关的7个方法（同上，见后文代码说明）

5、在JS文件中实现调用C++方法的代码

参考网址：

<https://www.cnblogs.com/wgwyanfs/p/7230657.html>

代码说明：

1）CxxDlg.cpp中实现供JS调用的方法

void CGAODEMapDlg::ShowPointString(const wchar\_t \*msg) {

int pSize = WideCharToMultiByte(CP\_OEMCP, 0, msg, wcslen(msg), NULL, 0, NULL, NULL);

char\* pCStrKey = new char[pSize+1];

WideCharToMultiByte(CP\_OEMCP, 0, msg, wcslen(msg), pCStrKey, pSize, NULL, NULL);

pCStrKey[pSize] = '\0';

SetDlgItemText(IDC\_EDIT3,pCStrKey);//IDC\_EDIT1你的edit控件ID

return;

}

2）CxxDlg.h中声明与IDispatch有关的7个方法

virtual HRESULT STDMETHODCALLTYPE GetTypeInfoCount(UINT \*pctinfo);

virtual HRESULT STDMETHODCALLTYPE GetTypeInfo(UINT iTInfo, LCID lcid, ITypeInfo \*\*ppTInfo);

virtual HRESULT STDMETHODCALLTYPE GetIDsOfNames(REFIID riid, LPOLESTR \*rgszNames, UINT cNames, LCID lcid, DISPID \*rgDispId);

virtual HRESULT STDMETHODCALLTYPE Invoke(DISPID dispIdMember, REFIID riid, LCID lcid, WORD wFlags, DISPPARAMS \*pDispParams, VARIANT \*pVarResult, EXCEPINFO \*pExcepInfo, UINT \*puArgErr);

virtual HRESULT STDMETHODCALLTYPE QueryInterface(REFIID riid, void \*\*ppvObject);

virtual ULONG STDMETHODCALLTYPE AddRef();

virtual ULONG STDMETHODCALLTYPE Release();

3）CxxDlg.cpp中实现与IDispatch有关的方法

enum

{

FUNCTION\_ShowMessageBox = 1,

FUNCTION\_GetProcessID = 2,

};

HRESULT STDMETHODCALLTYPE CGAODEMapDlg::GetTypeInfoCount(UINT \*pctinfo)

{

return E\_NOTIMPL;

}

HRESULT STDMETHODCALLTYPE CGAODEMapDlg::GetTypeInfo(UINT iTInfo, LCID lcid,

ITypeInfo \*\*ppTInfo)

{

return E\_NOTIMPL;

}

HRESULT STDMETHODCALLTYPE CGAODEMapDlg::GetIDsOfNames(REFIID riid, LPOLESTR \*rgszNames, UINT cNames, LCID lcid, DISPID \*rgDispId)

{

if (wcscmp(rgszNames[0], L"ShowPointString") == 0) //JS调用C++方法的名称为 ShowPointString

{

\*rgDispId = FUNCTION\_ShowMessageBox;

return S\_OK;

}

else

return E\_NOTIMPL;

}

HRESULT STDMETHODCALLTYPE CGAODEMapDlg::Invoke(DISPID dispIdMember, REFIID riid, LCID lcid, WORD wFlags, DISPPARAMS \*pDispParams, VARIANT \*pVarResult, EXCEPINFO \*pExcepInfo,

UINT \*puArgErr)

{

//通过ID我就知道JavaScript想调用哪个方法

if (dispIdMember == FUNCTION\_ShowMessageBox)

{

//检查是否仅仅有一个參数

if (pDispParams->cArgs != 1)

return E\_NOTIMPL;

//检查这个參数是否是字符串类型

if (pDispParams->rgvarg[0].vt != VT\_BSTR)

return E\_NOTIMPL;

//放心调用

ShowPointString(pDispParams->rgvarg[0].bstrVal);

return S\_OK;

}

else

return E\_NOTIMPL;

}

HRESULT STDMETHODCALLTYPE CGAODEMapDlg::QueryInterface(REFIID riid, void \*\*ppvObject)

{

if (riid == IID\_IDispatch || riid == IID\_IUnknown)

{

//对的，我是一个IDispatch，把我自己(this)交给你

\*ppvObject = static\_cast<IDispatch\*>(this);

return S\_OK;

}

else

return E\_NOINTERFACE;

}

ULONG STDMETHODCALLTYPE CGAODEMapDlg::AddRef()

{

return 1;

}

ULONG STDMETHODCALLTYPE CGAODEMapDlg::Release()

{

return 1;

}

4）JS中调用C++代码

//传入C++对象，供JS调用C++代码

var cpp\_object; //被JS调用方法的C++对象

function SaveCppObject(obj) {

cpp\_object = obj;

}

//地图中右击鼠标，获取对应坐标

map.addEventListener("rightclick", function (c) {

showPointValueInHtml(c);//在地图上使用便签显示经纬度值

//在该方法体内处理获得的经纬度数据（传到C++或者其他途径显示），直接使用c.point 即可获得当前点坐标

cpp\_object.ShowPointString(c.point.lng + "," + c.point.lat); //调用C++代码

});

4）将C++对象传给JS，供JS调用

在Web Browser的DocumentComplete中实现如下代码，使MFC初始化完成后，C++主动向JS传输C++对象

void CGAODEMapDlg::DocumentCompleteExplorer1(LPDISPATCH pDisp, VARIANT\* URL)

{

// TODO: 在此处添加消息处理程序代码

CComQIPtr<IHTMLDocument2> document = Browser.get\_Document();

CComDispatchDriver script;

document->get\_Script(&script);

CComVariant var(static\_cast<IDispatch\*>(this));

script.Invoke1(L"SaveCppObject", &var);

}

运行程序即可。