

WebOffice 文档控件

开发文档

北京点聚信息技术有限公司

2008/05/08

北京点聚信息技术有限公司（以下简称“点聚”）尽最大努力使这篇文章中的内容完善且正确。点聚对于由这篇文档导致的任何形式的直接或间接损失不负有责任。这篇文章的内容会跟随产品的升级而有所变化。

标记

名称: WebOffice 开发文档
状态: 发布
部门: 研发部
文档 ID: WebOffice-01-001

更新记录

版本	时间	备注
5.0.8.2	2008-05	设置了对文档安全的控制 清除了对 Word 模板的修改
5.0.8.0	2008-03	支持自定义的菜单，并跟 Javascript 绑定
5.0.1.4	2007-10	可以设置文档打开密码和可写密码, 支持 Word、Excel。 设置 Word 中的变量，可以更新关联的域。 扩展了这个接口，支持 Http 下的相对路径

软件开发协议

北京点聚信息技术有限公司（以下简称点聚）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有定单都受本协议的制约。如果您不愿接受这些条款，请在收到后的 7 天内将开发工具包寄回点聚，预付邮资和保险。我们会把货款退还给您，但要扣除运费和适当的手续费。

1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如开发指南中描述的那样保护该程序。您可以以存档为目的复制合理数量的拷贝。

2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件 and 产品的任何部分进行反向工程，或企图推导软件的源代码。禁止使用产品中的磁性或光学介质来传递、存储非本产品的原始程序或由点聚提供的产品升级的任何数据。禁止将软件放在服务器上传播。

3. 有限担保

点聚保证在自产品交给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

4. 修理限度

当根据本协议提出索赔时，点聚唯一的责任就是根据点聚的选择，免费进行替换或维修。点聚对更换后的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障 14 天内连同令人信服的证据交给点聚。当将产品返还给点聚或点聚的授权代理商时，须预付运费和保险。

除了在本协议中保证的担保之外，点聚不再提供特别的或隐含的担保，也不再对本协议中所描述的产品负责，包括它们的质量，性能和对某一特定目的适应性。

5. 责任限度

不管因为什么原因，不管是因合同中的规定还是由于刑事的原因，包括疏忽的原因，而使您及任何一方受到了损失，由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系，点聚对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下，点聚对于由于您不履行责任所导致的损失，或对于数据、利润、储蓄或其它的后续的和偶然的损失，即使点聚被建议有这种损失的可能性，或您根据第 3 方的索赔而提出的任何索赔均不负责任。

6. 协议终止

当您不能遵守本协议所规定的条款时，将终止您的许可和本协议。但条款 2, 3, 4, 5 将继续有效。

目 录

一、工作原理.....	7
1.1 开发流程.....	7
1.2 WEB页面调用控件:	8
二、接口说明.....	9
2.1 接口.....	9
2.1.1 AboutBox.....	9
2.1.2 AcceptAllRevisions.....	9
2.1.3 AddMenu.....	9
2.1.4 BookMarkOpt.....	10
2.1.5 CloseDoc.....	10
2.1.6 ConvertToAip.....	10
2.1.7 CreateApp.....	11
2.1.8 DelLocalFile.....	11
2.1.9 ExcProtectSharing.....	11
2.1.10 ExcUnprotectSharing.....	12
2.1.11 FtpConnect.....	12
2.1.12 FtpDisConnect.....	12
2.1.13 FtpGetFile.....	12
2.1.14 GetDocumentObject.....	13
2.1.15 GetFieldValue.....	14
2.1.16 GetTempFilePath.....	14
2.1.17 GetRevCount.....	14
2.1.18 GetRevInfo.....	15
2.1.19 GetBookMarkCount.....	15
2.1.20 GetBookMarkInfo.....	15
2.1.21 HideMenuItem.....	15
2.1.22 HttpInit.....	16
2.1.23 HttpAddPostFile.....	16
2.1.24 HttpAddPostBuffer.....	16
2.1.25 HttpAddPostValue.....	16
2.1.26 HttpAddPostString.....	17
2.1.27 HttpAddPostCurrFile.....	17
2.1.28 HttpAddPostCurrFile2.....	17
2.1.29 HttpPost.....	18
2.1.30 InsertFile.....	18
2.1.31 IsOpened.....	19

2.1.32	<i>IsSaved</i>	19
2.1.33	<i>InsertFile</i>	19
2.1.34	<i>LoadOriginalFile</i>	20
2.1.35	<i>OpenFileDlg</i>	21
2.1.36	<i>ProtectDoc</i>	21
2.1.37	<i>PrintDoc</i>	22
2.1.38	<i>PutSaved</i>	22
2.1.39	<i>Save</i>	22
2.1.40	<i>SaveTo</i>	23
2.1.41	<i>SaveAs</i>	23
2.1.42	<i>SetFieldValue</i>	23
2.1.43	<i>SetTrackRevisions</i>	24
2.1.44	<i>ShowRevisions</i>	25
2.1.45	<i>SetCurrUserName</i>	25
2.1.46	<i>SetToolBarButton2</i>	25
2.1.47	<i>SetCurrTime</i>	26
2.1.48	<i>SetKeyCtrl</i>	27
2.1.49	<i>SetSecurity</i>	27
2.1.50	<i>ShowDialog</i>	28
2.1.51	<i>SetValue</i>	28
2.1.52	<i>SetDocVariable</i>	29
2.1.53	<i>long UnProtectDoc (BSTR strPwd)</i>	29
2.1.54	<i>ZipFile</i>	30
2.2	属性.....	31
2.2.1	<i>DocType</i>	31
2.2.2	<i>Version</i>	31
2.2.3	<i>AppVersion</i>	31
2.2.4	<i>OptionFlag</i>	31
2.3	事件.....	32
2.3.1	<i>NotifyCtrlReady</i>	32
三、ASP例子说明	33
3.1	Access数据库结构	33
3.2	演示程序主要页面说明	33
3.3	开发必读.....	33
四、FAQ.....		34
4.1	如何把文件直接上传到服务器	34
五、附录.....		35
5.1	WebOffice文件Http上传原理.....	35

一、工作原理

点聚 WebOffice 是基于客户端的 ocx 控件,它的功能是将 Office 文档(Word、Excel、WPS) 嵌入到浏览器中,并调用 Office 中各种接口,完成文档编辑工作,然后使用控件的 Http 接口模拟表单提交,发送到数据处理页面(此文档举例为 saveDoc.asp 页,代码见演示文件),完成文档的存档工作。

1.1 开发流程

└ 装载控件 <object>...</ object >

└ 触发初始化方法

```
<SCRIPT LANGUAGE=javascript FOR=WebOffice1 EVENT=NotifyCtrlReady>
<!--
    WebOffice1_NotifyCtrlReady()
//-->
</SCRIPT>
```

└ 执行初始化方法,打开新文档或装载已经存在的文档

```
function WebOffice1_NotifyCtrlReady() {
    document.all.WebOffice1.LoadOriginalFile("...", "...");
}
```

└ 根据当前步骤,来决定是否进入修订状态,如果是修订状态,最好设置当前的用户名

```
function WebOffice1_NotifyCtrlReady() {
    document.all.WebOffice1.LoadOriginalFile("...", "...");
    document.all.WebOffice1.SetCurrUserName("张三");
    document.all.WebOffice1.SetTrackRevisions(1);
}
```

└ 填写表单元素,编辑 Office 文档

└ 执行 Http 上传接口保存表单元素和 Office 文档

```
//保存文档之前,如果需要接受修订,需要调用
document.all.WebOffice1.SetTrackRevisions(4);
//通过 Http 接口上传
document.all.WebOffice1.HttpInit();
document.all.WebOffice1.HttpAddPostString("...", "...");
document.all.WebOffice1.HttpAddPostCurrFile("...", "...");
document.all.WebOffice1.HttpPost("...")
```

└ saveDoc.asp 获取表单数据和 Office 进行数据库操作,并返回操作是否正确。

└ 根据 saveDoc.asp 返回值进行相应操作。

1.2 WEB 页面调用控件:

```
<OBJECT id=WebOffice1 height=520 width="100%" style="LEFT: 0px; TOP: 0px"
classid="clsid:E77E049B-23FC-4DB8-B756-60529A35FAD5"
codebase=WebOffice.ocx#Version=3,0,0,0>
  <PARAM NAME="_Version" VALUE="65536">
  <PARAM NAME="_ExtentX" VALUE="2646">
  <PARAM NAME="_ExtentY" VALUE="1323">
  <PARAM NAME="_StockProps" VALUE="0"></OBJECT>
```

主要参数说明:

id	控件在网页的别名, 可以通过别名调用控件的方法、属性
width	控件的宽度, 可以使用绝对大小, 如 width="100px"。
height	控件的高度
codebase	设置控件自动下载的路径, 用户如果没有安装控件, 系统会自动访问 codebase 的路径下载控件; “#Version” 后的是控件的版本, 如果用户当前的软件版本低于 “#Version” 后面的版本, 系统也会自动下载最新的控件

注意: 如果控件的 width 和 height 属性使用 % 表示, 则要明确指明它所在的容器的大小, 如把控件放到表格 <td> 内, 则要明确指明表格 <td> 的 width 和 height 的绝对值。

二、接口说明

2.1 接口

2.1.1 *AboutBox*

原 型: void AboutBox();

功 能: 显示 WebOffice 版本信息

返回值:

示 例:

```
document.all.WebOffice1. AboutBox();
```

2.1.2 *AcceptAllRevisions*

原 型: long AcceptAllRevisions();

功 能: 接受当前文档所有修订

返回值:

0: 接受修订成功

非零: 接受修订失败

示 例:

```
document.all.WebOffice1.AcceptAllRevisions();
```

2.1.3 *AddMenu*

原 形: AddMenu(LPCTSTR cMenuName, LPCTSTR cChildName, long iIndex, LPCTSTR cBindJSFun)

功 能: (保留)

参 数:

返回值:

示 例:

说明: 添加菜单

参数:

cMenuName : 主菜单名称(不可以为空)

cChildName : 子菜单名称(如为空, 则表示非二级菜单)

iIndex : 菜单的位置, 低 8 位表示加到 Menu 的菜单项, 如大于 Menu 的总数, 则加到

Menu 的最后一个菜单。

如 iIndex 最高位为 1 即 (0x8000) 则表示新建菜单 (保留)。

如 iIndex 的第 2 高位为 1 即 (0x4000), 表示在 cMenuName 菜单的前面加上分割线。

如 iIndex 的第 3 高位为 1 即 (0x2000), 表示在 cChildName 菜单的前面加上分割线。

cBindJSFun : 绑定的 javascript 函数名

2.1.4 BookMarkOpt

原 型: long BookMarkOpt(BSTR strBookMarks, long lType);

功 能: 获取文档书签列表(保留)

参 数:

返回值:

示 例:

2.1.5 CloseDoc

原 型: long CloseDoc(long blReqSaved);

功 能: 关闭当前文档

参 数:

blReqSaved:

0: 关闭文档, 不保存当前修改。

1: 关闭文档, 保存当前修改。

2: 如果文档已经被修改, 显示保存文档对话框, 否则直接关闭。

其它: 关闭文档, 保存当前修改。

返回值:

0: 失败

非零: 成功

示 例:

```
//关闭当前文档, 不保存当前修改。
```

```
document.all.WebOffice1.CloseDoc (0);
```

2.1.6 ConvertToAip

原 型: short ConvertToAip(short nAddOriFile, short nSupport)

功 能: 把当前通过[LoadOriginalFile](#)打开的原始文件直接转化成AIP文件。

参 数:

nAddOriFile: 是否装载原始文件。

0: 不装载。

非 0: 装载原始文件。

nSupport:

是否以普通方式（即解析文档的书签）打开。0: 不解析。非 0: 解析。

返回值:

0: 转化失败

非零: 转化成功

示 例: `document.all.WebOffice1.ConvertToAip (1, 1);`

2.1.7 *CreateApp*

原 型: `short CreateApp(BSTR strAppName);`

功 能: (保留)

参 数:

返回值:

示 例:

2.1.8 *DelLocalFile*

原 型: `long DelLocalFile(BSTR strFilePathName);`

功 能: 删除本地文件

参 数:

strFilePathName: 本地文件的绝对路径。

返回值:

0: 删除成功

非零: 删除失败

示 例:

```
document.all.WebOffice1.DelLocalFile("c:\\111.doc");
```

2.1.9 *ExcProtectSharing*

原 型: `long ExcProtectSharing(BSTR strPassword, BSTR strWriteResPassword, short bReadOnlyRecommended, BSTR strSharingPassword);`

功 能: (保留)

参 数:

返回值:

示 例:

2.1.10 *ExcUnprotectSharing*

原 型: long ExcUnprotectSharing(BSTR strSharingPassword);
功 能: (保留)
参 数:
返回值:
示 例:

2.1.11 *FtpConnect*

原 型: long FtpConnect(BSTR strURL, long lPort, BSTR strUser, BSTR strPwd);
功 能: FTP 相关(保留)
参 数:
返回值:
示 例:

2.1.12 *FtpDisConnect*

原 型: long FtpDisConnect();
功 能: FTP 相关(保留)
参 数:
返回值:
示 例:

2.1.13 *FtpGetFile*

原 型: long FtpGetFile(BSTR strRemoteFile, BSTR strLocalFile);
功 能: FTP 相关(保留)
参 数:
返回值:
示 例:

2.1.14 *GetDocumentObject*

原 型: IDispatch* GetDocumentObject();

功 能:

获取文档对象的 **ActiveDocument** 对象，此接口非常重要，基本所有的 VBA 功能都可以通过这个接口来实现。

参 数:

返回值: ActiveDocument 对象的 Dispatch

示 例:

/* 以下的 JavaScript 为采用 ActiveDocument 的方式直接操作 VBA，向 DOC 文档的书签中加入一个图片。

```
function AddPicture(strMarkName,strBmpPath,vType)
在书签位置插入图片,
strMarkName 书签的名称
strBmpPath 图片的路径
vType 插入后的图片的样式, 5 为浮动在文字上面
AddPicture("test1","d:\\111.bmp",5);
AddPicture("test1","d:\\111.bmp",128);
*/
function AddPicture(strMarkName,strBmpPath,vType)
{
//定义一个对象，用来存储ActiveDocument对象
var obj;
obj = new Object(document.all.WebOffice1.GetDocumentObject());
if(obj !=null){
    var pBookMarks;
    // VAB接口获取书签集合
    pBookMarks = obj.Bookmarks;
    var pBookM;
    // VAB接口获取书签strMarkName
    pBookM = pBookMarks(strMarkName);
    var pRange;
    // VAB接口获取书签strMarkName的Range对象
    pRange = pBookM.Range;
    var pRangeInlines;
    // VAB接口获取书签strMarkName的Range对象的InlineShapes对象
    pRangeInlines = pRange.InlineShapes;
    var pRangeInline;
    // VAB 接口通过 InlineShapes 对象向文档中插入图片
    pRangeInline = pRangeInlines.AddPicture(strBmpPath);
```

```
//设置图片的样式，5 为浮动在文字上面
pRangeInline.ConvertToShape().WrapFormat.TYPE = vType;
delete obj;
}
}
```

2.1.15 *GetFieldValue*

原 型: BSTR GetFieldValue(BSTR strFieldName, BSTR strSheetName);

功 能: 获得书签位置的数值

参 数:

strFieldName: 书签的名字

strSheetName: 预留字段（值为空）

返回值: 返回书签的值

示 例:

```
//获得书签的值
var strFieldValue=document.all.WebOffice1.GetFieldValue("mark_1","")
```

2.1.16 *GetTempFilePath*

原 型: BSTR GetTempFilePath();

功 能: 获得临时文件路径

参 数:

返回值: 当前文件的临时路径

示 例:

```
var strTempPath=document.all.WebOffice1.GetTempFilePath ();
```

2.1.17 *GetRevCount*

原 型: long GetRevCount();

功 能: (保留)

参 数:

返回值:

示 例:

2.1.18 *GetRevInfo*

原 型: BSTR GetRevInfo(long lIndex, long lType);

功 能: (保留)

参 数:

返回值:

示 例:

2.1.19 *GetBookMarkCount*

原 型: long GetBookMarkCount();

功 能: (保留)

参 数:

返回值:

示 例:

2.1.20 *GetBookMarkInfo*

原 型: BSTR GetBookMarkInfo(long lIndex, long lType);

功 能: (保留)

参 数:

返回值:

示 例:

2.1.21 *HideMenuItem*

原 型: long HideMenuItem(long lMenuType);

功 能: (保留)

参 数:

返回值:

示 例:

2.1.22 *HttpInit*

原 型: long HttpInit();

功 能: 初始化上传接口

参 数:

返回值:

示 例:

```
//上传前必须执行此方法  
document.all.weboffice1.HttpInit();
```

2.1.23 *HttpAddPostFile*

原 型: long HttpAddPostFile(BSTR strName, BSTR strFile);

功 能: Http 上传相关(保留)

参 数:

返回值:

示 例:

2.1.24 *HttpAddPostBuffer*

原 型: long HttpAddPostBuffer(BSTR strName, BSTR strRelateFile, BSTR pbBuffer, long lLen);

功 能: Http 上传相关(保留)

参 数:

返回值:

示 例:

2.1.25 *HttpAddPostValue*

原 型: long HttpAddPostValue(BSTR strName, long lValue);

功 能: Http 上传相关(保留)

参 数:

返回值:

示 例:

2.1.26 *HttpAddPostString*

原 型: long HttpAddPostString(BSTR strName, BSTR strValue);

功 能: 添加要上传的表单域

参 数:

strName: 表单域的名称

strValue: 表单域的值

返回值:

示 例:

```
//添加表单域, 可多次执行此方法
document.all.WebOffice1.HttpAddPostString("DocTitle",
myform.DocTitle.value);
```

2.1.27 *HttpAddPostCurrFile*

原 型: long HttpAddPostCurrFile(String field, String newFileName);

功 能: 添加要上传的 Word 或者 Excel 文件

参 数:

Field: 要上传文件的 id

newFileName: 上传后的新文件名。该参数可以为空, 系统将自动为文件命名

返回值:

示 例:

```
document.all.WebOffice1.HttpAddPostCurrFile("DocContent", "");
```

2.1.28 *HttpAddPostCurrFile2*

原 型: long HttpAddPostCurrFile2(BSTR strName)

功 能: (保留)

参 数:

返回值:

示 例:

2.1.29 *HttpPost*

原 型: BSTR HttpPost(BSTR strHttpURL);

功 能: 将当前表单和文件数据提交到指定 URL 路径

参 数:

strHttpURL: 处理表单和文件数据的文件 URL 路径

返回值:

Succeed: 上传成功

failed: 上传失败

示 例:

```
//相对路径
var returnValue=document.all.WebOffice1.HttpPost("../saveDoc.jsp");
//绝对路径
var
returnValue=document.all.WebOffice1.HttpPost("http://127.0.0.1/saveDoc.jsp")
)
```

2.1.30 *InsertFile*

原 形: short InsertFile(LPCTSTR strFilePathName, short nCurPos)

功 能: 插入合并文件, 和插入图片。

参 数: strFilePathName:

文件或者图片的路径, 可以是 Http. 或者 Ftp 路径

nCurPos:

位置和图片控制符。nCurPos 或者 nCurPos -8

0: 当前位置

1: 文件开头

2: 文件末尾

如果第四位为 1: 表示插入的是图片

返回值:

0: 正确

其他: 错误

示 例:

```
//文件头部
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",1);
//文件尾部
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",2);
//当前光标位置
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",0);
//文件头部
```

```
document.all.WebOffice1.InsertFile("http://XX.com/XX.jpg", 9);  
//文件尾部  
document.all.WebOffice1.InsertFile("http://XX.com/XX.jpg", 10);  
//当前光标位置  
document.all.WebOffice1.InsertFile("http://XX.com/XX.jpg", 8);
```

2.1.31 *IsOpened*

原 型: long IsOpened();

功 能: 判断当前是否打开文件

参 数:

返回值:

0: 表示当前未打开文档

非 0: 表示已经打开文档

FILE_TYPE_WORD = 11,

FILE_TYPE_WORD = 11,

FILE_TYPE_EXCEL = 12,

FILE_TYPE_PPT = 13,

FILE_TYPE_RTF = 14,

FILE_TYPE_WPS = 21,

FILE_TYPE_PDF = 31,

FILE_TYPE_UNK = 127

示 例:

```
var isOpened=document.all.WebOffice1.IsOpened();
```

2.1.32 *IsSaved*

原 型: long IsSaved();

功 能: 判断文件是否被修改

参 数:

返回值:

0: 文件已被修改

非零: 文件未被修改

示 例:

```
var isSaved=document.all.WebOffice1.IsSaved();
```

2.1.33 *InsertFile*

原 型: long InsertFile(BSTR strFilePathName, short nCurPos);

功 能: 插入、合并文件或插入图片

参 数:

strFilePathName: 文件或者图片的路径, 可以是 Http.或者 Ftp 路径

nCurPos: 位置和图片控制符。nCurPos 或者 nCurPos -8

0: 当前位置

1: 文件开头

2: 文件末尾

如果第四位为 1: 表示插入的是图片

返回值:

0: 插入成功

非 0: 插入失败

示 例:

```
//文件头部
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",1);
//文件尾部
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",2);
//当前光标位置
document.all.WebOffice1.InSertFile("http://XX.com/XX.doc",0);
//文件头部
document.all.WebOffice1.InSertFile("http://XX.com/XX.jpg",9);
//文件尾部
document.all.WebOffice1.InSertFile("http://XX.com/XX.jpg",10);
//当前光标位置
document.all.WebOffice1.InSertFile("http://XX.com/XX.jpg",8);
```

2.1.34 LoadOriginalFile

原 型: long LoadOriginalFile(BSTR pcFileNameOrUrl, BSTR pcType);

功 能: 新建、装载 Office 文档

参 数:

pcFileNameOrUrl:

要装载文件的本地路径或者 URL, 值为空表示新建文档, 如果是在 Http 的 Web 环境下, 可以传相对路径.

pcType:

doc: 打开、新建 Word 文件

xls: 打开、新建 Excel 文件

ppt: 打开、新建 PPT 文件

wps: 打开、新建 WPS 文件

返回值:

0: 失败

非 0:成功

示 例:

```
// 通过 URL 打开一个服务器文件, getdoc.asp 代码见演示文件
document.all.WebOffice1.LoadOriginalFile("http://127.0.0.1/getdoc.asp?id=8", "doc");
// 通过 URL 打开一个服务器文件, 相对路径
document.all.WebOffice1.LoadOriginalFile("/getdoc.asp?id=8", "doc");
document.all.WebOffice1.LoadOriginalFile("./getdoc.asp?id=8", "doc");
// 新建 Excel
document.all.WebOffice1.LoadOriginalFile("", "xls");
// 通过 Word 打开本地文件
document.all.WebOffice1.LoadOriginalFile("C:\\demo.doc", "doc");
```

2.1.35 *OpenFileDialog*

原 型: short OpenFileDialog();

功 能: 打开本地文件显示在 WebOffice 中

参 数: pcFileNameOrUrl: 保存的文件路径

返回值:

示 例:

```
//打开对话框
document.all.WebOffice1.OpenFileDialog();
```

2.1.36 *ProtectDoc*

原 型: long ProtectDoc(long lProOrUn, long lType, BSTR strPWD);

功 能: 文档保护

参 数:

lProOrUn:

└ 1: 为保护文档

└ 0: 为解锁文档

lType :

└ 0 : 文档保护, 只允许修订

└ 1 : 文档保护, 只允许批注

└ 2 : 文档完全保护, 不允许任何操作

strPWD: 文档保护、解锁时所需设置的密码

返回值:

示 例:

```
// 保护文档, 只允许修订
document.all.WebOffice1.ProtectDoc(1, 0, "12345");
// 解锁文档
```

```
document.all.WebOffice1.ProtectDoc(0, 0, "12345");
```

2.1.37 PrintDoc

原 型: long PrintDoc(long blShowDlg);

功 能: 打印当前文档

参 数: blShowDlg:

0: 不显示打印对话框, 直接打印

非零: 显示打印对话框

返回值:

0: 打印失败

非零: 打印成功

示 例:

```
//显示打印对话框  
document.all.WebOffice1.PrintDoc(1);
```

2.1.38 PutSaved

原 型: long PutSaved(boolean bSaved);

功 能: 设置当前文档的修改状态

参 数:

0: 处于修改状态

1: 处于保存状态

返回值:

示 例:

```
//设置当前文档处于修改状态  
document.all.WebOffice1.PutSaved(0);
```

2.1.39 Save

原 型: long Save();

功 能:

将文件保存原来位置, 第一次保存时弹出对话框

参 数:

返回值:

0: 保存失败

非零: 保存成功

示 例:

```
document.all.WebOffice1.Save();
```

2.1.40 SaveTo

原 型: long SaveTo(BSTR pcFileNameOrUrl);

功 能: 将文件另存为至其它位置

参 数:

pcFileNameOrUrl: 保存的文件路径

返回值:

0: 保存失败

非零: 保存成功

示 例:

```
//将文件另存为  
document.all.WebOffice1.SaveTo("c:\\webOffice.doc");
```

2.1.41 SaveAs

原 型: long SaveAs(BSTR strFileName, long dwFileFormat);

功 能: 保存文件到本地

参 数:

strFileName: 要保存的文件路径和文件名

dwFileFormat: 保存的文件类型

└ 0 Word 类型

└ 2 Txt 文件

└ 6 Rtf 文件

└ -4143 Excel 类型

返回值:

示 例:

```
document.all.WebOffice1.SaveAs("C:\\test.doc", 0); // 另存为 Word  
文件  
document.all.WebOffice1.SaveAs("C:\\test.txt", 2); // 另存为 Txt  
文件  
document.all.WebOffice1.SaveAs("C:\\test.rtf", 6); // 另存为 Rtf  
文件  
document.all.WebOffice1.SaveAs("C:\\test.xls", -4143); // 另存为 Excel  
文件
```

2.1.42 SetFieldValue

原 型: long SetFieldValue(BSTR strFieldName, BSTR strValue, BSTR strSheetName);

功 能: 添加书签, 套红头模板

参 数: strFieldName: 书签的名字

strValue: 书签的值, 该处可以是个用于返回数据的 URL, 用于套红头模板

strSheetName: 命令的类型

└ ::ADDMARK:: 添加书签

└ ::DELMARK:: 删除书签

└ ::JPG:: 添加图片, 可以利用这个来实现图片电子印章

└ ::FILE:: 调用本地或者 URL 的文件设置书签值 (套红头)

返回值:

示 例:

```
// 添加书签 mark_1, 并为它赋值
document.all.WebOffice1.SetFieldValue("mark_1", "北京点聚",
"::ADDMARK::");
// 为已存在的书签 mark_1 赋值
document.all.WebOffice1.SetFieldValue("mark_1", "电子签章", "");
// 套本地红头
document.all.WebOffice1.SetFieldValue("mark_1", "c:\\test.doc",
"::FILE::");
// 套服务端红头 1, getDoc.asp 页用于返回要装载 Word 的二进制值, 代码见演示文件
document.all.WebOffice1.SetFieldValue("mark_1",
"http://127.0.0.1/getdoc.asp?id=8", "::FILE::");
// 套服务端红头 2
document.all.WebOffice1.SetFieldValue("mark_1", "http://127.0.0.1/test.doc",
"::FILE::");
// 添加图片电子印章
document.all.WebOffice1.SetFieldValue("mark_1", "http://127.0.0.1/seal.bmp",
"::JPG::");
```

2.1.43 SetTrackRevisions

原 型: long SetTrackRevisions(long lValue);

功 能: 修订文档

参 数:

lValue: 文档修订状态

└ 1: 开始修订文档

└ 2: 进入非留痕状态

└ 4: 接受当前修订

返回值:

示 例:

```
// 开始修订
document.all.WebOffice1.SetTrackRevisions(1);
// 接受修订
```



```
document.all.WebOffice1.SetTrackRevisions(4);
```

2.1.44 ShowRevisions

原型: long ShowRevisions(long nNewValue);

功能: 显示、隐藏修订

参数:

nNewValue: 要保存的文件路径和文件名

└ 1: 显示修订

└ 0: 隐藏修订

返回值:

示例:

```
document.all.WebOffice1.ShowRevisions(0); // 隐藏修订
```

2.1.45 SetCurrUserName

原型: long SetCurrUserName(BSTR strCurrUserName);

功能: 设置当前编辑 Word、Excel 的用户名, 主要用于修订文档, 标识不同人进行的不同操作

参数: strCurrUserName: 当前编辑人的用户名

返回值:

示例: document.all.WebOffice1.SetCurrUserName("张三");

2.1.46 SetToolBarButton2

原型: short SetToolBarButton2(VARIANT lIndex, VARIANT lID, VARIANT lControl);

功能:

设置所有菜单项或者菜单项按钮的隐藏和现实

参数:

lIndex: 要显示隐藏的元素名称。根据 Word/Excel 的不同版本, 可能有部分不同

└ Standard: 常用工具条

└ Formatting: 格式化工具条

└ Menu Bar: 菜单栏

当 lIndex 为 0 的时候, lID 代表的是按钮的 ID 数值, 关于各按钮的 ID 值, 请参照附件。

lID: 要操作的菜单元素的位置

lControl: 按钮状态控制, 按位控制, 如该值为 1 时, 它的二进制为 0001, 表示该菜单元素是显示但不可用

第一位为: 是否显示。0: 隐藏; 1: 显示

第二位为：是否可用。0：不可用；1：可用。

第三位为：工具栏和菜单栏是否恢复到默认状态。0：不恢复；
1：恢复。

第四位为：是否隐藏整个菜单栏或工具栏，1 为隐藏，0 为显示

返回值：

示 例：

```
// 常用工具栏上的新建按钮 显示但不可用
document.all.WebOffice1.SetToolBarButton2("Standard",1,1);
// 常用工具栏上的打开按钮 隐藏。
document.all.WebOffice1.SetToolBarButton2("Standard",2,0);
// 恢复常用工具栏为初始状态
document.all.WebOffice1.SetToolBarButton2("Standard",1,4);
// 格式工具栏上的格式按钮 显示但不可用
document.all.WebOffice1.SetToolBarButton2("Formatting",1,1);
// 恢复格式工具栏为初始状态
document.all.WebOffice1.SetToolBarButton2("Formatting",1,4);
// 菜单栏文件选项，显示但不可用。
document.all.WebOffice1.SetToolBarButton2("Menu Bar",1,1);
// 菜单栏文件选项，显示但不可用。
document.all.WebOffice1.SetToolBarButton2("Menu Bar",1,1);
// 隐藏整个菜单栏（如果需要隐藏之后再显示，请设置第三个参数为 11）
document.all.WebOffice1.SetToolBarButton2("Menu Bar",1,8);
// 隐藏整个格式工具栏（如果需要隐藏之后再显示，请设置第三个参数为 11）
document.all.WebOffice1.SetToolBarButton2("Formatting",1,8);
// 隐藏整个常用工具栏（如果需要隐藏之后再显示，请设置第三个参数为 11）
document.all.WebOffice1.SetToolBarButton2("Standard",1,8);
```

2.1.47 SetCurrTime

原 型：short SetCurrTime(BSTR dateAndTime);

功 能：设置客户端的修订时间，防止客户端和服务端的时间误差

参 数：

dateAndTime：参数的形式必须为 yyyy-mm-dd hh-mm-ss 的形式

返回值：

示 例：

```
document.all.WebOffice1.SetCurrTime("2006-02-07 11:11:11");
```

2.1.48 SetKeyCtrl

原 型: long SetKeyCtrl(long lCtrl, long lChar, long lType);

功 能: 屏蔽、运行 Office 的快捷键

参 数:

lCtrl: 要屏蔽、运行的快捷键的值。

Ctrl=512 Shift=256 Alt=1024

例: Ctrl + S = 512 + 83 (大写 S 的 ASCII 码值) = 595

例: Ctrl + O = 512 + 79 = 591

例: Ctrl + Shift + E = 512 + 256 + 69 = 837

注意: 字母使用大写的 ASCII 码值

lChar: 控制类型

└ -1: 屏蔽快捷键

└ 1: 执行快捷键命令

└ 0: 恢复快捷键

└ lTyp: 扩展接口, 目前暂设为 0

返回值:

示 例:

```
//屏蔽 Ctrl + S
document.all.WebOffice1.SetKeyCtrl(595,-1,0)
//屏蔽 Ctrl + O
document.all.WebOffice1.SetKeyCtrl(591,-1,0)
```

2.1.49 SetSecurity

原 型: long SetSecurity(long lFlag)

功 能:

设置文档安全选项

参 数:

lFlag: 按位标记的控制开关。

0x01 表示不允许打印(当第一位为 1 时有效)

0x02 表示不允许保存(当第二位为 1 时有效)

0x04 表示不允许复制(当第三位为 1 时有效)

0x08 表示不允许拖动(当第四位为 1 时有效)

0x8000 表示清除开关(当最高位为 1 时有效)

返回值:

示 例:

```
//禁止打印
Object.SetSecurity(0x01);
```

```
//恢复允许打印
Object.SetSecurity(0x01 + 0x8000);
//禁止保存
Object.SetSecurity(0x02);
//恢复允许保存
Object.SetSecurity(0x02 + 0x8000);
//禁止复制
Object.SetSecurity(0x04);
//恢复允许复制
Object.SetSecurity(0x04 + 0x8000);
//禁止打印、保存、复制
Object.SetSecurity(0x01+0x02+0x04);
//恢复允许打印、保存、复制
Object.SetSecurity(0x01+0x02+0x04+0x8000);
```

2.1.50 ShowDialog

原 型: long ShowDialog(long dwIndex);

功 能:

显示对话框

参 数:

dwIndex: 对话框的 ID 数值, 详细的数值列表请参照附件。

返回值:

示 例:

2.1.51 SetValue

原 形: short SetValue(LPCTSTR strValue, LPCTSTR strName)

功 能: 设置 Word、Excel 文档基本信息, 目前仅支持 只读和可写密码

参 数: strValue: 设置的基本信息的数值

strName: 基本信息的标识, 如只读密码的标识为":DOCPROP:PassWord"

可改密码的标识为":DOCPROP:WritePW"

返回值:

0: 正确设置

-1: 不支持此命令, 请确定您的第二个参数为正确。

示 例:

```
//设置当前文档的打开密码为" password"
document.all.WebOffice1.SetValue("password", ":DOCPROP:PassWord");

//设置当前文档的可写密码为" password"
document.all.WebOffice1.SetValue("password", ":DOCPROP: WritePW");
```

2.1.52 SetDocVariable

原 形: short SetDocVariable(LPCTSTR strVarName, LPCTSTR strValue, long 10pt)

功 能: 设置 Word 中的变量的值，更可以更新与此变量相关联的域。

参 数: strVarName

变量名称

strValue

变量设置的数值

10pt

一个按位操作的数字位，第一位为 1: 表示 update 域关联的

第二位为 1: 表示如果没有这个变量则添

加

返回值:

1: 正确设置

0: 错误

示 例:

```
//设置变量 DocName 的值为“文档名称”，如不存在 DocName 的变量则返回  
document.all.WebOffice1.SetDocVariable("DocName","文档名称",0);
```

```
//设置变量 DocName 的值为“文档名称”，如不存在 DocName 的变量则添加  
document.all.WebOffice1.SetDocVariable("DocName","文档名称",2);
```

```
//设置变量 DocName 的值为“文档名称”，并更新 DocName 所关联的域  
document.all.WebOffice1.SetDocVariable("DocName","文档名称",1);
```

2.1.53 long UnProtectDoc(BSTR strPwd)

原 型: long UnProtectDoc(BSTR strPwd);

功 能: 解除当前文档保护

参 数: strPwd 文档保护时所设置的密码

返回值:

0: 解锁成功

非零: 解锁失败

示 例:

```
document.all.WebOffice1.UnProtectDoc("12345");
```

2.1.54 ZipFile

原 型: short ZipFile(BSTR cSourceFile, BSTR cAimFile, long wFileType, long bEnc);

功 能: 压缩文档 (保留)

参 数:

返回值:

示 例:

2.2 属性

2.2.1 *DocType*

原 型: long DocType;

说 明: 文件类型, 只读

数 值:

```
FILE_TYPE_WORD = 11,  
FILE_TYPE_WORD = 11,  
FILE_TYPE_EXCEL = 12,  
FILE_TYPE_PPT = 13,  
FILE_TYPE_RTF = 14,  
FILE_TYPE_WPS = 21,  
FILE_TYPE_PDF = 31,  
FILE_TYPE_UNK = 127
```

2.2.2 *Version*

原 型: BSTR Version;

说 明: 打开文件的版本号, 只读

数 值:

2.2.3 *AppVersion*

原 型: BSTR Version;

说 明: 打开文件的版本号, 只读

数 值:

2.2.4 *OptionFlag*

原 型: long OptionFlag;

说 明: 系统标志位, 可读写。此属性比较重要, 可以作为各种控制开关。

数 值:

```
OPTIONFLAG_WORD_07SAVETO030x0001  
OPTIONFLAG_EXCEL_07SAVETO03 0x0002  
OPTIONFLAG_HTTPSAVEENC      0x0004
```

2.3 事件

2.3.1 *NotifyCtrlReady*

原 型: void NotifyCtrlReady();

说 明: 当控件准备完毕后触发, 一般在这个事件中完成初始化的操作。

三、ASP 例子说明

3.1 Access 数据库结构

表 doc

=====	
id	自动编号
DocID	文本 (10)
DocTitle	文本 (10)
DocType	文本 (15)
DocData	日期/时间
DocContent	OLE 对象
=====	

说明: DocContent 为上传的 Office 文档的数据, 为二进制数据类型, 如果使用 SQLServer 请改为 Image 类型。

3.2 演示程序主要页面说明

config.asp 取得服务器的项目文件目录, 赋值给 Application 对象, 供接口调用
docEdit.asp 调用 weboffice 控件在线编辑 word, 并提交给 savedoc.jsp 页面
上传保存 word 文档
saveDoc.asp 保存表单元素和上传的 word 文件
getDoc.asp 取得数据库中的记录并返回给调用它的页面
debug.htm 用于调试数据传送到 saveDoc.asp 页
template 用于存放红头文件的模板

3.3 开发必读

- 1) 调用控件的时候尽量在控件 ID 前加 “document.all”, 如:
`document.all.WebOffice1`
- 2) Javascript 区分大小写, 使用时请注意。
- 3) WebOffice 控件为 VC++6 编写, 控件的所有方法和参数区分大小写, 请严格参考本手册。
- 4) 在本控件的演示程序中, 有个 debug.html 文件用于测试表单元素和文件上传。

四、FAQ

4.1 如何把文件直接上传到服务器

WebOffice 通过 Http 协议实现文件直接上传到服务器，原理见附件一《WebOffice 文件 Http 上传原理》

五、附录

5.1 WebOffice 文件 Http 上传原理

WebOffice 文件 Http 上传原理分析

点聚 WebOffice 控件使用了自身的文件 Http 上传接口实现文件和表单域的上传保存，本文将简要介绍该上传方式的原理和使用方法。注：点聚 WebOffice 的上传接口和 AIP 是一致的。

原理

因为 Web 页面上的 WebOffice 打开的文件无法随表单记录提交到处理页面，所以我们将文件模拟成表单的文件域

主要上传接口：

- HttpInit();
该方法用于初始化上传接口，不需要设置参数
- HttpAddPostString(String FieldName, String someValue);
添加要提交的表单域，FieldName 是要上传表单域名称，someValue 是表单域值

例一：

```
HttpAddPostString("username", "zhangSan");  
它类似域表单域的  
<input type=text name=username value=zhangSan>
```

例二：

```
HttpAddPostString("username", WebForm.username.value);  
....  
<form name= WebForm >  
    <input type=text name=username value=zhangSan>  
</form>
```

注意：如果使用点聚上传接口，那么 form 表单里的表单域不会被提交到处理页面。需要手工添加到上传接口（HttpAddPostString 接口），然后通过控件发送到处理页面。例二使用了“WebForm.username.value”获取表单域值。

- HttpAddPostCurrFile(String fileName, String ReName);
添加当前 WebOffice 编辑的文件给上传接口，该接口把文件模拟到成一个表单的文件域，并提交到处理页面。

例一：

```
HttpAddPostCurrFile("AipFile", "");
```

该接口模拟表单文件域如下：

```
<input type=file name=AipFile >
```

注意：第二个参数为空表示控件自动为文件命名，如：dj_temp.aip，如果需要重命名可以在第二个参数指明文件名称，也可在保存页面指定文件名称。

- **HttpPost(String url);**

执行上传操作将文件和表单域提交到 url 的页面进行保存操作，并获取 url 页面返回的执行结果（是否成功，可以根据这个结果来获取服务器端执行的结果）。

url 支持绝对路径和相对路径。

绝对路径，例如：http://192.168.0.1/savedoc.asp

相对路径要以“./”或者“../”开头。例如：./savedoc.asp

完整的上传接口演示

```
function SaveDoc() {  
    var returnValue; // 保存页面的返回值  
    document.all.WebOffice1.HttpInit(); // 初始化 Http 引擎  
    // 添加相应的 Post 元素  
    document.all.WebOffice1.HttpAddPostString("username", WebForm.username.value);  
    // 添加上传文件  
    document.all.WebOffice1.HttpAddPostCurrFile("AipFile", "");  
    // 提交上传文件  
    returnValue = document.all.WebOffice1.HttpPost("../saveDoc.jsp");  
    if("true" == returnValue){  
        alert("文件上传成功");  
    } else {  
        alert("文件上传失败")  
    }  
}
```

关于保存

WebOffice 上传接口将数据流以 Http 方式提交到保存页面，至于对数据处理就和普通的文件上传处理一样了，具体代码参考附件（asp 和 jsp 的实现）。