

# 软航科技 NTKO OFFICE 文档控件

## JavaScript 编程指南

Web: <http://www.ntko.com>  
Email: [support@ntko.com](mailto:support@ntko.com)  
更新日期: 2004.08.15  
©Copyright: 本文档及技术资料版权所有，未经书面许可，禁止拷贝分发。

### 说明:

本文档为使用 NTKO OFFICE 文档控件的客户使用 JavaScript 控制控件中的 OFFICE 文档提供了一个简要的编程指南。仅供 NTKO OFFICE 文档控件的客户使用。

技能要求: Javascript。

### 目录:

一、 NTKO OFFICE 文档控件简介: .....	2
二、 原理介绍.....	2
三、 如何将宏转变为 Javascript 函数.....	3
四、 新增函数，方法，事件.....	4
1: 增加自定义菜单并处理菜单事件 .....	4
五、 常用函数示例.....	5
1: 控件界面控制.....	5
2: 控件功能控制.....	6
3: 控件文件菜单控制.....	6
4: 创建，从本地打开和保存文档: .....	7
5: 从 URL 打开文档: .....	8
6: 将在线编辑的文档以及 FORM 中的其他信息保存到 URL.....	8
7: 关闭文档.....	8
8: 打印控制（可控制前台或者后台打印） .....	8
9: 痕迹保留控制相关.....	9
10: 文档控制.....	11
11: VBA 编程 .....	12
12: 域值交换.....	14
13: 手写签名和电子印章.....	17
14: 插入图片和手工批注.....	20
15: 有关模板套红.....	21

## 一、NTKO OFFICE 文档控件简介：

NTKO OFFICE 文档控件是拥有完全自主知识产权的 ActiveX 控件。使用 NTKO Office 文档控件，能够在浏览器窗口中直接编辑 Word，Excel 等 Office 文档并保存到 Web 服务器。实现文档和电子表格的统一管理。同时支持强制痕迹保留，手写签名，电子印章等办公自动化系统必备的功能。是 OA 系统开发的最佳选择组件！控件采用标准 HTTP 上传协议，支持任意后台 Web 服务器（iis，domino，webaphere，apache 等），任意后台操作系统（win2k，win2003，linux，unix 等），以及任意后台编程语言（asp，asp.net，jsp，php，vb.net，c#等）和任意 web 应用服务器体系结构（j2ee，.net 等）。卓越的性能，低廉的价格，标准高效的编程接口，让您以最低的开发成本，轻松创建极具特色的办公自动化解决方案！

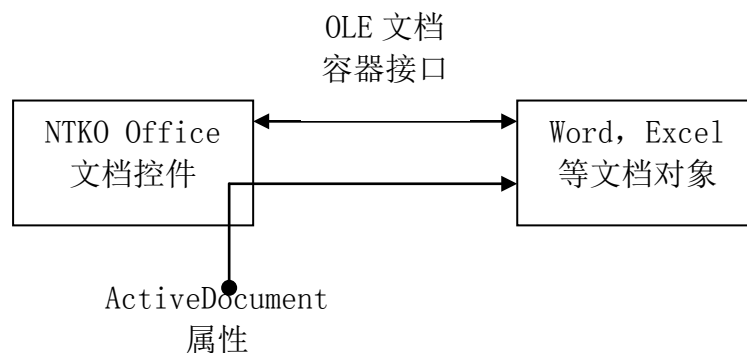
## 二、原理介绍

⚠ 注意，在本文档及以下部分，假设控件对象为 TANGER\_OCX\_OBJ。该对象可以在 HTML 页面的 onload 事件中通过 `TANGER_OCX_OBJ=document.all("控件 ID");`来赋值。

NTKO OFFICE 文档控件的 ActiveDocument 属性，如果是打开了 Word 文档，就是一个 Word 的 Document 对象。

如果是打开了 Excel 电子表，就是一个 Excel 的 Workbook 对象。这两个对象的详细信息分别在 Word 和 Excel 的联机帮助中与编程信息有关的部分可以查阅到。（需要完全安装 Word 和 Excel 才会有编程帮助信息）。

NTKO OFFICE 文档控件和其中打开的文档对象的关系如下：




因此，可以通过 Javascript 对 ActiveDocument 属性的访问，来达到控制控件内部的 Word,Excel 对象的目的。

比如，以下代码将在 Word 的当前光标插入文字：

```
TANGER_OCX_OBJ.ActiveDocument.Application.Selection.TypeText("test by tanger");
```


比如：以下代码设置控件中的第一个电子表格的 b2 单元格的值：

```
TANGER_OCX_OBJ.ActiveDocument.Sheets(1).Range("B2").Value="test by tanger";
```

 **注意：**在没有打开文档之前，**ActiveDocument** 属性是空值。

比如如下函数：

```
function testdoc()
{
    if (typeof(TANGER_OCX_OBJ.ActiveDocument) == "undefined")
    {    alert("undefined"); }
    else if (TANGER_OCX_OBJ.ActiveDocument == null)
    {    alert("null"); }
    else
    {    alert("ok"); }
}
```

 **注意：**在控件中没有打开文档时，将返回"null"。因此，要对 **ActiveDocument** 属性进行操作，请确认您已经打开了文档。或者在操作之前判断其值是否为空以防止出现错误。

### 三、如何将宏转变为 Javascript 函数

有时候您可以通过录制宏 vba 代码的方法并转换成 javascript 的方法来完成功能。宏中用到的是文字常量，比如，对于红色是 wdRed，在将 vba 宏代码转换成 javascript 代码的过程中，需要将这些常量转换成对应的数字。Word 和 Excel 中用到的常量请参考随本文档一同分发的两个 IDL 文件。您可以使用搜索功能在这两个 IDL 文件中找到所需的常量。

另外，vba 语法中有些命名参数的调用，比如，method a:=1, b:=2，这种调用方法是 Javascript 不支持的。如果有这种情况，您需要查阅 Word 和 Excel 的帮助，找到方法 method 的介绍，按照顺序传递给 javascript 参数。在 vba 中，可选参数是可以省略的，但是在 javascript 中，如果一个可选参数后面还有别的参数，您必须给前面省略的可选参数传递空值（null）。

如果您在编程中遇到问题，也可以直接跟我们联系。我们将提供帮助。

## 四、新增函数，方法，事件

### 1: 增加自定义菜单并处理菜单事件

NTKO OFFICE 文档控件新版本允许增加新的自定义文件菜单项目和主菜单项目并处理其事件。

首先，设定自定义主菜单的标题：（通过 CustomMenuCaption 属性）

```
<param name="CustomMenuCaption" value="我的菜单">
```

其次，在 HTML 文件引用的 js 文件或者 HTML 文件中定义如下 JS 函数，增加菜单项目：

```
function AddMyMenuItems()
{
    try
    {
        //在自定义主菜单中增加菜单项目
        TANGER_OCX.AddCustomMenuItem('我的菜单 1',false,false,1);
        TANGER_OCX.AddCustomMenuItem("");
        TANGER_OCX.AddCustomMenuItem('我的菜单 2',false,false,2);
        TANGER_OCX.AddCustomMenuItem("");
        TANGER_OCX.AddCustomMenuItem('我的菜单 3',false,true,3);
        //在文件菜单中增加菜单项目
        TANGER_OCX.AddFileMenuItem('创建 Word 文档',false,false,1);
        TANGER_OCX.AddFileMenuItem('创建 Excel 文档',false,false,2);
        TANGER_OCX.AddFileMenuItem('创建 PPT 文档',false,false,3);
        TANGER_OCX.AddFileMenuItem('关闭文档',false,true,4);
        TANGER_OCX.AddFileMenuItem("");
    }
    catch(err){
        alert("不能创建新对象: "+ err.number + ":" + err.description);
    }
    finally{
    }
}
}
```

第 3 步，在 Body 的 Onload 事件中调用上述方法：

```
<body bgcolor="#ffffff" topmargin="0" leftmargin="0" marginheight="0" marginwidth="0" onload="AddMyMenuItems ()">
```

最后，在引用控件的 HTML 文件中增加如下的代码以接管自定义菜单事件：

```
<script language="JScript" for="TANGER_OCX"
event="OnCustomMenuCmd(menuIndex,menuCaption,menuID)">
alert("第" + menuIndex + "个菜单项,menuID="+menuID+",菜单标题为"+menuCaption+"的命令
被执行.");
</script>
<script language="JScript" for="TANGER_OCX"
event="OnCustomFileMenuCmd(menuIndex,menuCaption,menuID)">
alert("第" + menuIndex + "个菜单项,menuID="+menuID+",菜单标题为"+menuCaption+"的命令
被执行.");
switch(menuID)
{
    case 1:
        TANGER_OCX.CreateNew("Word.Document");
        break;
    case 2:
        TANGER_OCX.CreateNew("Excel.Sheet");
        break;
    case 3:
        TANGER_OCX.CreateNew("Powerpoint.Show");
        break;
    case 4:
        TANGER_OCX.Close();
        break;
}
</script>
```

## 五、常用函数示例

以下是常用的函数调用示例。详细参数及使用方法请参考用户编程手册。

① 技巧：出于稳定性考虑，在真正运行时，可以使用 `try{}catch{}封装` 下面的函数，以避免出现因 js 错误而引起的不便。调试时，可以在 `catch` 中显示错误信息。

### 1: 控件界面控制

以下 Javascript 函数使用控件属性来控制控件是否显示标题栏，菜单栏，状态栏，工具栏。

```
function ShowTitleBar(bShow)
{
    TANGER_OCX_OBJ.Titlebar = bShow;
```

```
}  
function ShowMenubar(bShow)  
{  
    TANGER_OCX_OBJ.Menubar = bShow;  
}  
function ShowStatusbar(bShow)  
{  
    TANGER_OCX_OBJ.Statusbar = bShow;  
}  
function ShowToolbars(bShow)  
{  
    TANGER_OCX_OBJ.Toolbars = bShow;  
}
```

## 2: 控件功能控制

以下函数可用于允许或禁止用户从控件拷贝数据。

```
function TANGER_OCX_SetNoCopy(boolvalue)  
{  
    TANGER_OCX_OBJ.IsNoCopy = boolvalue;  
}
```

以下函数可用于工具菜单项控制。(隐藏工具菜单项和审阅工具栏及右键菜单可用于强制用户在痕迹保留状态下工作)

```
function ShowToolMenu(bShow)  
{  
    TANGER_OCX_OBJ.IsShowToolMenu = bShow;  
}
```

## 3: 控件文件菜单控制

以下函数用于对控件文件菜单的控制。

//允许或禁止文件—>新建菜单

```
function TANGER_OCX_EnableFileNewMenu(boolvalue)  
{  
    TANGER_OCX_OBJ.FileNew = boolvalue;  
}  
//允许或禁止文件—>打开菜单
```

```

function TANGER_OCX_EnableFileOpenMenu(boolvalue)
{
    TANGER_OCX_OBJ.FileOpen = boolvalue;
}
//允许或禁止文件—>关闭菜单
function TANGER_OCX_EnableFileCloseMenu(boolvalue)
{
    TANGER_OCX_OBJ.FileClose = boolvalue;
}
//允许或禁止文件—>保存菜单
function TANGER_OCX_EnableFileSaveMenu(boolvalue)
{
    TANGER_OCX_OBJ.FileSave = boolvalue;
}
//允许或禁止文件—>另存为菜单
function TANGER_OCX_EnableFileSaveAsMenu(boolvalue)
{
    TANGER_OCX_OBJ.FileSaveAs = boolvalue;
}
//允许或禁止文件—>打印菜单
function TANGER_OCX_EnableFilePrintMenu(boolvalue)
{
    TANGER_OCX_OBJ.FilePrint = boolvalue;
}
//允许或禁止文件—>打印预览菜单
function TANGER_OCX_EnableFilePrintPreviewMenu(boolvalue)
{
    TANGER_OCX_OBJ.FilePrintPreview = boolvalue;
}

```

#### 4: 创建，从本地打开和保存文档：

创建新 Word 文档：

```
TANGER_OCX_OBJ.CreateNew("Word.Document");
```

创建新 Excel 电子表格：

```
TANGER_OCX_OBJ.CreateNew("Excel.Sheet");
```

提示用户选择本地文件打开：

```
TANGER_OCX_OBJ.ShowDialog (1);
```

提示用户选择本地文件保存：

```
TANGER_OCX_OBJ.ShowDialog (3);
```

不提示用户，直接打开指定的本地文件：

```
TANGER_OCX_OBJ.OpenLocalFile ("c:\\test.doc");
```

不提示用户，直接保存为指定的本地文件：

```
TANGER_OCX_OBJ.SaveToLocal ("c:\\test.doc",false);//第二个参数为true 将覆盖已存在的文件
```

## 5：从 URL 打开文档：

以下调用从 readdoc.asp 中打开文件。readdoc.asp 可以从数据库或者服务器磁盘中读取文件并返回。

```
TANGER_OCX_OBJ.OpenFromURL ("readdoc.asp");//可以是相对或者绝对 URL
```

## 6：将在线编辑的文档以及 FORM 中的其他信息保存到 URL

以下调用将控件中编辑的文件保存到 upload.asp。从后台 upload.asp 程序来看，只需要处理当前 HTML 页面的 FORM(0)的信息，以及一个名称为 newdoc.doc，输入域为 EDITFILE 的上传文件。一般第二个参数可以被 upload.asp 用来区分是控件中上传的文件，还是当前 HTML 页面 FORM 中的其他上传文件。另外，后台的 upload.asp 程序可以将文件保存到服务器磁盘或者数据库的 BLOB 字段。

```
TANGER_OCX_OBJ.SaveToURL ("upload.asp","EDITFILE","", "newdoc.doc",0);
```

## 7：关闭文档

```
TANGER_OCX_OBJ.Close();
```

## 8：打印控制（可控制前台或者后台打印）

以下函数用于设置页面布局：

```
function TANGER_OCX_ChgLayout()
{
    try
    {
        TANGER_OCX_OBJ.showdialog(5); //设置页面布局
    }
}
```



```
catch(err){
    alert("错误: " + err.number + ":" + err.description);
}
finally{
}
}
```

以下 Javascript 函数可以用来打印当前文档。isBackground 参数可以控制是前台打印还是后台打印。对于比较大的文档，如果确认必须等待用户打印完毕，可以调用 TANGER\_OCX\_PrintDoc(false);函数来进行前台打印。

```
function TANGER_OCX_PrintDoc(isBackground)
{
    var oldOption;
    try
    {
        var objOptions = TANGER_OCX_OBJ.ActiveDocument.Application.Options;
        oldOption = objOptions.PrintBackground;
        objOptions.PrintBackground = isBackground;
    }
    catch(err){};
    TANGER_OCX_OBJ.printout(true);
    try
    {
        var objOptions = TANGER_OCX_OBJ.ActiveDocument.Application.Options;
        objOptions.PrintBackground = oldOption;
    }
    catch(err){};
}
```

## 9: 痕迹保留控制相关

设置当前 Office 的用户名（痕迹的用户名）

```
//设置用户名
function TANGER_OCX_SetDocUser(cuser)
{
    with(TANGER_OCX_OBJ.ActiveDocument.Application)
    {
        UserName = cuser;
    }
}
```

```
}

```

进入或退出强制痕迹保留状态，调用该函数下面定义的两个函数。一般可直接调用本函数

```
function TANGER_OCX_SetMarkModify(boolvalue)
{
    TANGER_OCX_SetReviewMode(boolvalue);
    TANGER_OCX_EnableReviewBar(!boolvalue);
}
```

允许或禁止显示修订工具栏和工具菜单（保护修订,用户不能更改当前修订状态）:

```
function TANGER_OCX_EnableReviewBar(boolvalue)
{
    TANGER_OCX_OBJ.ActiveDocument.CommandBars("Reviewing").Enabled = boolvalue;
    TANGER_OCX_OBJ.ActiveDocument.CommandBars("Track Changes").Enabled = boolvalue;
    TANGER_OCX_OBJ.IsShowToolMenu = boolvalue; //关闭或打开工具菜单
}
```

打开或者关闭修订模式:

```
function TANGER_OCX_SetReviewMode(boolvalue)
{
    TANGER_OCX_OBJ.ActiveDocument.TrackRevisions = boolvalue;
}
```

显示/不显示修订文字

```
function TANGER_OCX_ShowRevisions(boolvalue)
{
    TANGER_OCX_OBJ.ActiveDocument.ShowRevisions = boolvalue;
}
```

打印/不打印修订文字

```
function TANGER_OCX_PrintRevisions(boolvalue)
{
    TANGER_OCX_OBJ.ActiveDocument.PrintRevisions = boolvalue;
}
```

接受所有修订

```
function TANGER_OCX_AcceptAllRevisions()
{
    TANGER_OCX_OBJ.ActiveDocument.AcceptAllRevisions();
}
```

```
}

```

拒绝所有修订

```
function TANGER_OCX_RejectAllRevisions()
{
    TANGER_OCX_OBJ.ActiveDocument. RejectAllRevisions ();
}
```

## 10: 文档控制

切换文档的只读状态（文档保护状态）

```
function TANGER_OCX_SetReadOnly(boolvalue)
{
    var i;
    try
    {
        if (boolvalue) TANGER_OCX_OBJ.IsShowToolMenu = false;
        with(TANGER_OCX_OBJ.ActiveDocument)
        {
            if (TANGER_OCX_OBJ.DocType == 1) //word
            {
                if ( (ProtectionType != -1) &&  !boolvalue)
                {
                    Unprotect();
                }
                if ( (ProtectionType == -1) &&  boolvalue)
                {
                    Protect(2,true,"");
                }
            }
            else if(TANGER_OCX_OBJ.DocType == 2)//excel
            {
                for(i=1;i<=Application.Sheets.Count;i++)
                {
                    if(boolvalue)
                    {
                        Application.Sheets(i).Protect("",true,true,true);
                    }
                    else
                    {

```

```

        Application.Sheets(i).Unprotect("");
    }
}
if(boolvalue)
{
    Application.ActiveWorkbook.Protect("",true);
}
else
{
    Application.ActiveWorkbook.Unprotect("");
}
}
}
}
}
catch(err){
    //alert("错误: " + err.number + ":" + err.description);
}
finally{}
}

```

## 11: VBA 编程

此函数用来加入一个自定义的文件头部:

```

function TANGER_OCX_AddDocHeader( strHeader )
{
    var i,cNum = 30;
    var lineStr = "";
    try
    {
        for(i=0;i<cNum;i++) lineStr += "_"; //生成下划线
        with(TANGER_OCX_OBJ.ActiveDocument.Application)
        {
            Selection.HomeKey(6,0); // go home
            Selection.TypeText(strHeader);
            Selection.TypeParagraph(); //换行
            Selection.TypeText(lineStr); //插入下划线
            // Selection.InsertSymbol(95,"",true); //插入下划线
            Selection.TypeText("★");
            Selection.TypeText(lineStr); //插入下划线
            Selection.TypeParagraph();
        }
    }
}

```

```
//Selection.MoveUp(5, 2, 1); //上移两行, 且按住 Shift 键, 相当于选择两行
Selection.HomeKey(6,1); //选择到文件头部所有文本
Selection.ParagraphFormat.Alignment = 1; //居中对齐
with(Selection.Font)
{
    NameFarEast = "宋体";
    Name = "宋体";
    Size = 12;
    Bold = false;
    Italic = false;
    Underline = 0;
    UnderlineColor = 0;
    StrikeThrough = false;
    DoubleStrikeThrough = false;
    Outline = false;
    Emboss = false;
    Shadow = false;
    Hidden = false;
    SmallCaps = false;
    AllCaps = false;
    Color = 255;
    Engrave = false;
    Superscript = false;
    Subscript = false;
    Spacing = 0;
    Scaling = 100;
    Position = 0;
    Kerning = 0;
    Animation = 0;
    DisableCharacterSpaceGrid = false;
    EmphasisMark = 0;
}
Selection.MoveDown(5, 3, 0); //下移 3 行
}
}
catch(err){
    //alert("错误: " + err.number + ":" + err.description);
}
finally{
}
}
```

## 12: 域值交换

如下 Javascript 函数可将 html form 的域值拷贝到 Word 文档的标签中

```
function CopyTextToBookMark(inputname,BookMarkName)
{
    try
    {
        var inputValue="";
        var j,elObj,optionItem;
        var elObj = document.forms[0].elements(inputname);
        if (!elObj)
        {
            alert("HTML 的 FORM 中没有此输入域: "+ inputname);
            return;
        }
        switch(elObj.type)
        {
            case "select-one":
                inputValue = elObj.options[elObj.selectedIndex].text;
                break;
            case "select-multiple":
                var isFirst = true;
                for(j=0;j<elObj.options.length;j++)
                {
                    optionItem = elObj.options[j];
                    if (optionItem.selected)
                    {
                        if(isFirst)
                        {
                            inputValue = optionItem.text;
                            isFirst = false;
                        }
                        else
                        {
                            inputValue += " " + optionItem.text;
                        }
                    }
                }
                break;
        }
    }
}
```

```

        default: // text,Areatext,selecte-one,password,submit,etc.
            inputValue = elObj.value;
            break;
    }
    //do copy
    //DEBUG
    //alert(inputname+"="+inputValue+" Bookmarkname="+BookMarkName);
    var bkmkObj = TANGER_OCX_OBJ.ActiveDocument.BookMarks(BookMarkName);
    if(!bkmkObj)
    {
        alert("Word 模板中不存在名称为: \""+BookMarkName+"\" 的书签! ");
    }
    var saverange = bkmkObj.Range
    saverange.Text = inputValue;
    TANGER_OCX_OBJ.ActiveDocument.Bookmarks.Add(BookMarkName,saverange);
}
catch(err){}
finally{
}
}

```

将 html form 的域值拷贝到 Word 文档的 MailMerge 域中

```

function CopyTextToField(inputname,FieldName)
{
    try
    {
        var inputValue="";
        var j,elObj,optionItem;
        var elObj = document.forms[0].elements(inputname);
        if(!elObj)
        {
            alert("HTML 的 FORM 中没有此输入域: "+inputname);
            return;
        }
        switch(elObj.type)
        {
            case "select-one":
                inputValue = elObj.options[elObj.selectedIndex].text;
                break;
            case "select-multiple":
                var isFirst = true;

```

```
        for(j=0;j<elObj.options.length;j++)
        {
            optionItem = elObj.options[j];
            if (optionItem.selected)
            {
                if(isFirst)
                {
                    inputValue = optionItem.text;
                    isFirst = false;
                }
                else
                {
                    inputValue += " " + optionItem.text;
                }
            }
        }

        break;
default: // text,Areatext,selecte-one,password,submit,etc.
    inputValue = elObj.value;
    break;
}

var Fields = TANGER_OCX_OBJ.ActiveDocument.MailMerge.Fields;
for(var iField=1; iField<=Fields.Count;iField++)
{
    //alert(Fields(iField).Code.Text);
    if ( Fields(iField).Code.Text.indexOf(FieldName) >= 0)
    {
        Fields(iField).Select();
        TANGER_OCX_OBJ.ActiveDocument.Application.Selection.Text = inputValue;
    }
}

}

catch(err){
    //alert(err.number + ":" + err.description);
}

finally{
}

}
```



从 Word 文档中读出内容，并传递给 inputname。

```
function ReadFromDoc(BookMarkName,inputname)
{
    var inputValue="";
    var elObj = document.forms[0].elements(inputname);
    if (!elObj)
    {
        alert("HTML 的 FORM 中没有此输入域: "+ inputname);
        return;
    }
    elObj.value = TANGER_OCX_OBJ.ActiveDocument.Content.Text;
}
```

从 Word 文档中读出指定书签的内容，并传递给 inputname。

```
function ReadBookMarkFromDoc(inputname)
{
    var inputValue="";
    var elObj = document.forms[0].elements(inputname);
    if (!elObj)
    {
        alert("HTML 的 FORM 中没有此输入域: "+ inputname);
        return;
    }
    var bkmkObj = TANGER_OCX_OBJ.ActiveDocument.BookMarks(BookMarkName);
    if (!bkmkObj)
    {
        alert("Word 模板中不存在名称为: \""+BookMarkName+"\"的书签! ");
    }
    elObj.value = bkmkObj.Range.Text;
}
```

### 13: 手写签名和电子印章

从本地增加印章到当前光标所在的段落的指定位置。

```
function AddSignFromLocal()
{
    // alert(TANGER_OCX_key);
    if(TANGER_OCX_bDocOpen)
    {
```

```

        TANGER_OCX_OBJ.AddSignFromLocal(
        TANGER_OCX_username,//当前登陆用户
        "",//缺省文件
        true,//提示选择
        0,//left
        0,
        TANGER_OCX_key,
        1,
        100,
        0)
    }
}

```

从 URL 增加印章到当前光标所在的段落的指定位置。

```

function AddSignFromURL(URL)
{
    // alert(TANGER_OCX_key);
    if(TANGER_OCX_bDocOpen)
    {
        TANGER_OCX_OBJ.AddSignFromURL(
        TANGER_OCX_username,//当前登陆用户
        URL,//URL
        50,//left
        50,
        TANGER_OCX_key,
        1,
        100,
        0)
    }
}

```

开始全屏手写签名

```

function DoFullScreenHandSign()
{
    // alert(TANGER_OCX_key);
    if(TANGER_OCX_bDocOpen)
    {
        TANGER_OCX_OBJ.DoHandSign2(

```

```

TANGER_OCX_Username, //当前登陆用户 必须
TANGER_OCX_key,
0, //left //可选参数
0, //top
0, //relative=0, 表示按照屏幕位置批注
100 //缩放 100%, 表示原大小
);
}
}

```

开始手写签名

```

function DoHandSign()
{
// alert(TANGER_OCX_key);
if(TANGER_OCX_bDocOpen)
{
TANGER_OCX_OBJ.DoHandSign(
TANGER_OCX_Username, //当前登陆用户 必须
0, //笔型 0—实线 0—4 //可选参数
0x000000ff, //颜色 0x00RRGGBB //可选参数
2, //笔宽 //可选参数
100, //left //可选参数
50, //top //可选参数
false, //可选参数
TANGER_OCX_key
);
}
}

```

检查签名结果

```

function DoCheckSign()
{
// alert(TANGER_OCX_key);
if(TANGER_OCX_bDocOpen)
{
var ret = TANGER_OCX_OBJ.DoCheckSign
(
false, /*如果为 FAISE, 表示弹出验证对话框, 否则, 只是返回验证结果到返回值*/
TANGER_OCX_key
); //返回值, 验证结果字符串
}
}

```

```

        //alert(ret);
    }
}

```

## 14: 插入图片和手工批注

本地增加图片到文档指定位置

```

function AddPictureFromLocal()
{
    if(TANGER_OCX_bDocOpen)
    {
        TANGER_OCX_OBJ.AddPicFromLocal(
            "", //路径
            true, //是否提示选择文件
            true, //是否浮动图片
            100, //如果是浮动图片, 相对于左边的 Left 单位磅
            100); //如果是浮动图片, 相对于当前段落 Top
    };
}

```

从 URL 增加图片到文档指定位置

```

function AddPictureFromURL(URL)
{
    if(TANGER_OCX_bDocOpen)
    {
        TANGER_OCX_OBJ.AddPicFromURL(
            URL, //URL 注意: URL 必须返回 Word 支持的图片类型。
            true, //是否浮动图片
            0,
            0,
            1, //当前光标处
            100, //无缩放
            1 //文字上方
        );
    };
}

```

开始手工绘图, 可用于手工批注

```

function DoHandDraw()
{

```

```

if(TANGER_OCX_bDocOpen)
{
    TANGER_OCX_OBJ.DoHandDraw(
        0,//笔型 0—实线 0—4 //可选参数
        0x00ff0000,//颜色 0x00RRGGBB//可选参数
        3,//笔宽//可选参数
        200,//left//可选参数
        50);//top//可选参数
    }
}

```

## 15: 有关模板套红

下列函数可以将 URL 指定的模板文件插入到当前文档的头部。可用于简单的套红。

```

function TANGER_OCX_DoTaoHong(URL)
{
    try{
        TANGER_OCX_OBJ.ActiveDocument.Application.Selection.HomeKey(6);
        TANGER_OCX_OBJ.AddTemplateFromURL(URL);
    }
    catch(err)
    {};
};

```

下列函数可以将当前编辑的文档插入到 URL 指定的模板文件的”ZhengWen”书签位置。可用于稍微复杂的套红。

```

function TANGER_OCX_DoPaiBan(URL)
{
    try{
        //选择对象当前文档的所有内容
        var curSel = TANGER_OCX_OBJ.ActiveDocument.Application.Selection;
        TANGER_OCX_SetMarkModify(false);
        curSel.WholeStory();
        curSel.Cut();
        //插入模板
        TANGER_OCX_OBJ.AddTemplateFromURL(URL);
        var BookMarkName = "zhengwen";
        if(!TANGER_OCX_OBJ.ActiveDocument.BookMarks.Exists(BookMarkName))
        {
            alert("Word 模板中不存在名称为: \""+BookMarkName+"\"的书签! ");
        }
    }
}

```

```

        return;
    }

    var bkmkObj = TANGER_OCX_OBJ.ActiveDocument.BookMarks(BookMarkName);
    var saverange = bkmkObj.Range
    saverange.Paste();
    TANGER_OCX_OBJ.ActiveDocument.Bookmarks.Add(BookMarkName,saverange);
    TANGER_OCX_SetMarkModify(true);
}
catch(err)
{
    //alert("错误: " + err.number + ":" + err.description);
};
}

```

下列函数可以在当前编辑的文档前面和后面插入两个指定的模板文件，并设置当前文档中文本格式为模板中”zhengwen”书签的字体及段落格式。可用于非常复杂的套红。

```

function TANGER_OCX_DoPaiBanFuZa(URL1,URL2)
{
    try{
        TANGER_OCX_SetMarkModify(false);//暂时退出痕迹保留状态
        var doc = TANGER_OCX_OBJ.ActiveDocument;
        var curSel = doc.Application.Selection;
        curSel.HomeKey(6); //跳转到文档开头
        //插入模板头部
        TANGER_OCX_OBJ.AddTemplateFromURL(URL1);
        //拷贝正文书签格式
        doc.BookMarks("zhengwen").Select();
        curSel.CopyFormat();
        //选择到尾部
        curSel.EndKey(6,1);
        curSel.PasteFormat();
        doc.BookMarks("zhengwen").Select();
        curSel.Delete();
        //删除段落标记
        curSel.Delete(1,1);
        curSel.EndKey(6); //跳转到文档尾部
        //插入模板尾部
        TANGER_OCX_OBJ.AddTemplateFromURL(URL2);
        //设置其他书签
        /*

```

```
doc.BookMarks("wenjiantoubu").Range.Text = "NTKO OFFICE 文档控件";
doc.BookMarks("wenhao").Range.Text = "111-111-111";
doc.BookMarks("biaoti").Range.Text = "关于采用 NTKO OFFICE 文档控件的决定";
doc.BookMarks("danwei").Range.Text = "尊敬的用户:";
doc.BookMarks("Notekeyword").Range.Text = "痕迹保留 电子印章 手写签名 复杂模板套
红 电子政务 同意采用 文档管理";
doc.BookMarks("Publishto").Range.Text = "各合作伙伴和用户单位";
doc.BookMarks("PublishUnit").Range.Text = "千航网络产品部";
doc.BookMarks("PublisDate").Range.Text = "2004 年 3 月 28 日";
/**/
TANGER_OCX_SetMarkModify(true); //进入痕迹保留状态
}
catch(err)
{
    alert("错误: " + err.number + ":" + err.description);
};
}
```