

## Word Viewer OCX v3.1

Word Viewer OCX acts as an ActiveX document container for hosting Word documents in a custom form or Web page. The OCX is lightweight and flexible, and gives developers new possibilities for using Microsoft Word in a custom solution.

Home: <http://www.officeocx.com>

Online Demo: [http://www.officeocx.com/Word\\_Viewer\\_Online\\_Demo.htm](http://www.officeocx.com/Word_Viewer_Online_Demo.htm)

Support Mail: [support@officeocx.com](mailto:support@officeocx.com)

Sales online: <https://www.regnow.com/softsell/nph-softsell.cgi?item=14203-2>

Support Office 97, Office 2000, Office XP, or Office 2003 in Windows OS.

It can be easily integrated into applications written in languages that support ActiveX control such as Visual C++, Visual Basic, Delphi, C++ Builder and .Net languages. Support Office automating client to custom your application. Include abundant sample codes. You also [get the full c++ source codes](#).

**Name:** Word Viewer OCX or Word Viewer ActiveX Control

**CLSID:** 97AF4A45-49BE-4485-9F55-91AB40F22BF2

**Version:** 3,1,0,3

**Release Date:** 2006-9-16

**OCX Size:** 472KB

### Code a solution using the control

*Note: The install package includes some sample codes written with visual c++ 6.0, visual basic, Delphi, c++ builder, C#, ASP.Net, php and so on.*

The control is very customizable. You can change the color scheme of any of the control elements, as well as determine the border type and a custom caption. These can be set at run time or design time as needed.

#### 1. Create new documents

Function void CreateNew(); Description: Creates a new word document.
---

#### 2. Open Local documents

Function: boolean Open(BSTR Path); Description: Opens a document from a local file.
--

You can also open and edit Office documents that exist on a local drive. You can call the **Open** method directly and give the control a specific file to open.

#### 3. Open Web folder

Function: boolean OpenWebFile(BSTR FileUrl); Description: Opens a document from a web folder.
--

You can also open and edit Office documents that exist on web folder. Open takes either a qualified file path or a URL to a file on a remote Web server. For example, the following code opens a web file.

```
OA1.OpenWebFile "http://www.officeocx.com/demo/sample.doc"
```

#### 4. Show Open File Dialog

```
Function: boolean OpenLocalDialog ();
```

Description: Opens a document from a local file with the standard open file dialog.

#### 5. Save Documents

```
Function: boolean Save(BSTR strPath);
```

Description: Save the opened file to specify a save location.

#### 6. Save As

```
Function: boolean SaveAs(BSTR strPath, long nFileType);
```

Description: Save the opened file to common word document formats.

Office Document Type:

enum WdSaveFormat

```
{  
    wdFormatDocument = 0,  
    wdFormatTemplate = 1,  
    wdFormatText = 2,  
    wdFormatTextLineBreaks = 3,  
    wdFormatDOSText = 4,  
    wdFormatDOSTextLineBreaks = 5,  
    wdFormatRTF = 6,  
    wdFormatUnicodeText = 7,  
    wdFormatEncodedText = 7,  
    wdFormatHTML = 8,  
    wdFormatWebArchive = 9,  
    wdFormatFilteredHTML = 10,  
    wdFormatXML = 11  
};
```

If you want to save the opened word document into the html format, you can write codes as follow:

```
OA1.SaveAs "c:\1.html", 10
```

#### 7. Show Save File Dialog

```
Function: void SaveLocalDialog();
```

Description: Saves a document to a local file with the standard save file dialog.

#### 8. Save Web Folder

Function: void SaveWebFile(BSTR ServerUrl);  
Description: Saves a document to a web folder.

Note: In order to save and upload the opened file in the PowerPoint Viewer ActiveX Control, you must write a method in your web server to receive the stream. (Please review the full sample codes in the install package)

**ASP:**

```
private void Save()
{
    Stream stream = Request.InputStream;
    StreamReader sr = new StreamReader(stream);
    string uploadString = sr.ReadToEnd();
    sr.Close();
    stream.Close();
    try
    {
        FileStream fs = new FileStream(@"c:\Test1\test.doc",
            FileMode.OpenOrCreate, FileAccess.Write, FileShare.None);
        byte[] _fs = Request.ContentEncoding.GetBytes(uploadString);
        fs.Write(_fs, 0, _fs.Length);
        fs.Flush();
        fs.Close();
    }
    catch(Exception ex)
    {
        Response.Write(ex.ToString());
    }
}
```

**ASP.NET:** (See \install path\sample\ASP.NET)

```
protected void Page_Load(object sender, EventArgs e)
{
    String Action = Request.QueryString["Action"]
    If(Action == "savefile"){
        string fileName = Request.QueryString["FileName"];
        if (fileName == "") fileName = "temp.ppt";

        Stream stream = Request.InputStream;
        byte[] fileByte = new byte[stream.Length];
        stream.Position = 0;
        stream.Read(fileByte, 0, fileByte.Length);
        stream.Close();
        string strTempPath = "c:\\";
```

```

        string packageFilePath = strTempPath + fileName;
        using (FileStream fileStream = new FileStream(packageFilePath,
        FileMode.CreateNew))
        {
            fileStream.Write(fileByte, 0, fileByte.Length);
            fileStream.Flush();
            fileStream.Close();
        }
    }
}

```

**PHP:** (See \install path\sample\SaveFile.php)

```

<?php
/* PUT data comes in on the stdin stream */
$putdata = fopen("php://stdin", "r");

/* Open a file for writing */
$fwp = fopen("test.ppt", "w");

/* Read the data 1 KB at a time
and write to the file */
while ($data = fread($putdata, 1024))
    fwrite($fwp, $data);

/* Close the streams */
fclose($fwp);
fclose($putdata);
?>

```

## 9. Close Document

Function: boolean Close();  
Description: Closes the currently open document.

## 10. Read-Only

Function: void SetReadOnly(boolean bReadOnly);  
Description: Set the read-only state

It is possible to change the viewer in a way it can only view the word documents.

```
OA1.SetReadOnly true
```

It is possible to change the viewer in a way it can edit the word documents.

```
OA1.SetReadOnly false
```

### 11. Print Out

```
Function: void Print();  
Description: Print out the document
```

### 12. Get the toolbars' state

```
Function: boolean GetToolbarsIsShow ();  
Description: Get the toolbars' state .
```

### 13. Show or hide the toolbars

```
Function: void ShowToolbars(boolean bShow);  
Description: Show/Hide whether toolbars should be displayed.
```

You can open an office document without toolbars as follow:

```
If OA1.GetToolbarsIsShow = True Then  
OA1.ShowToolbars False  
Else  
OA1.ShowToolbars True  
End If
```

### 14. Is Modified

```
Function boo IsDirty();  
Description: Returns True/False if file has been altered or needs save.
```

It returns true when the document has been altered or needs save.

```
BOOL bDirty = OA1.IsDirty
```

### 15. Upload and Download file

```
Function: 1) boolean HttpUploadFile(BSTR ServerUrl, BSTR LocalFilePath);  
2) boolean HttpDownloadFile(BSTR ServerUrl, BSTR LocalFileUrl);  
3) boolean FTPUploadFile(BSTR ServerUrl, BSTR LocalFilePath, BSTR UserName, BSTR  
Password);  
4) boolean FTPDownloadFile(BSTR ServerUrl, BSTR LocalFilePath, BSTR UserName, BSTR  
Password);  
Description: Upload and download the file with Http or FTP method.
```

The ActiveX Control supports upload and download file with HTTP or FTP method.

Download file with HTTP method:

```
OA1.HttpDownloadFile "http://www.officeocx.com/demo/Samples.doc"
```

Download file with FTP method:

```
OA1.FTPDownloadFile "http://www.officeocx.com/demo/Samples.doc", "c:\Samples.doc", "", ""
```

Upload file with HTTP method:

```
OA1.HttpUploadFile "http://www.officeocx.com/demo/SaveFile.php", "c:\Samples.doc"
```

Note: you must write a Stream receive method in the server.

Download file with HTTP method:

```
OA1.HTTPDownloadFile "http://www.officeocx.com/demo/Samples.doc", "c:\Samples.doc"
```

#### 16. Switch to the Read Mode

```
Function: void SwitchToReadMode ();
```

Description: Switch the opened file into the read mode..

#### 17. Switch to the Normal View

```
Function: void SwitchToNormalView ();
```

Description: Switch the opened file into the normal view .

#### 18. Switch to the Page View

```
Function: void SwitchToPageView ();
```

Description: Switch the opened file into the Page view..

#### 19. Switch to the Web View

```
Function: void SwitchToWebView ();
```

Description: Switch the opened file into the Web view..

#### 20. Go to Page

```
Function: GotoPage(short pageNum);
```

Description: Go to the appointed page .

#### 21. Open the Standard Dialog in the MS Word

```
Function: void ShowWordStandardDialog (short nDialogType);
```

Description: Open the standard dialog in the MS Word. .

The follow codes can open the Find dialog:

```
OA1.ShowWordStandardDialog(112);
```

The follow codes can open the summary information dialog:

```
OA1.ShowWordStandardDialog(86);
```

You can visit <http://www.officeocx.com/Dialog-Type.htm> to see all the enumeration values.

## 22. Automating Office Document

Function: IDispatch\* GetIDispatch();

Description: Returns the Automation interface of the document object.

The control also supports a property called **GetIDispatch** that allows you to obtain a reference to the **IDispatch** interface of the embedded object. From this interface you can automate the object to perform tasks, edit parts of the document, or gather information about what a user has added or removed. For example, if you have a Word document open, you can use code that resembles the following to add a line of text:

```
Dim oDoc As Word.Document
Set oDoc = OA1.GetIDispatch
oDoc.Content.Text = "This was added by Automation"
```

The ability to control the object while the object is embedded is very powerful.

Review how to insert an image into word. (VC Codes)

```
LPDISPATCH lpDisp;
lpDisp = m_pOA->GetIDispatch();

//Add text to the first line of the document
_Document m_WordDoc;

//set _Document doc to use lpDisp, the IDispatch* of the
//actual document.
m_WordDoc.AttachDispatch(lpDisp);
Shape m_wordShape;
Shapes m_wordShapes;

COleVariant vOpt((long)DISP_E_PARAMNOTFOUND, VT_ERROR);
COleVariant vTrue((short)TRUE),vFalse((short)FALSE);

CString m_strPath;
char exeFullPath[MAX_PATH];
GetModuleFileName(NULL,exeFullPath,MAX_PATH);
m_strPath.Format("%s",exeFullPath);
int ndx = m_strPath.ReverseFind('\\');
m_strPath = m_strPath.Left(ndx);
CString filename = m_strPath + "\\test.bmp";
```

```
_Application m_WordApp;

m_wordShapes = m_WordDoc.GetShapes();
CString FileName =filename;

m_WordApp=m_WordDoc.GetApplication();
InlineShapes m_WordInlineShapes;
InlineShape m_WordInlineShape;
Selection m_WordSelection;
m_WordSelection=m_WordApp.GetSelection();
m_WordInlineShapes=m_WordSelection.GetInlineShapes();

m_WordInlineShape = m_WordInlineShapes.AddPicture(FileName,vFalse,vTrue,vOpt);

m_wordShape = m_WordInlineShape.ConvertToShape();

Shapes shs;
Shape shp;
shs=m_WordDoc.GetShapes();
VARIANT var;
var.vt=VT_I4;
var.IVal=shs.GetCount();
shp=shs.Item(&var);
shp.Select(&var);

ShapeRange m_wordShapeRange;
WrapFormat m_wordWrapFormant;

m_WordApp = m_WordDoc.GetApplication();

m_wordShape = m_WordSelection.GetShapeRange();
m_wordWrapFormant = m_wordShape.GetWrapFormat();

m_wordWrapFormant.SetType(3);
m_wordShapeRange.ZOrder(4);
PictureFormat m_wordPictureFormat = m_wordShape.GetPictureFormat();
m_wordPictureFormat.SetTransparentBackground(TRUE);
m_wordPictureFormat.SetTransparencyColor(0xFFFFFFFF);

FillFormat m_wordFillFormat = m_wordShape.GetFill();
m_wordFillFormat.SetVisible(FALSE);

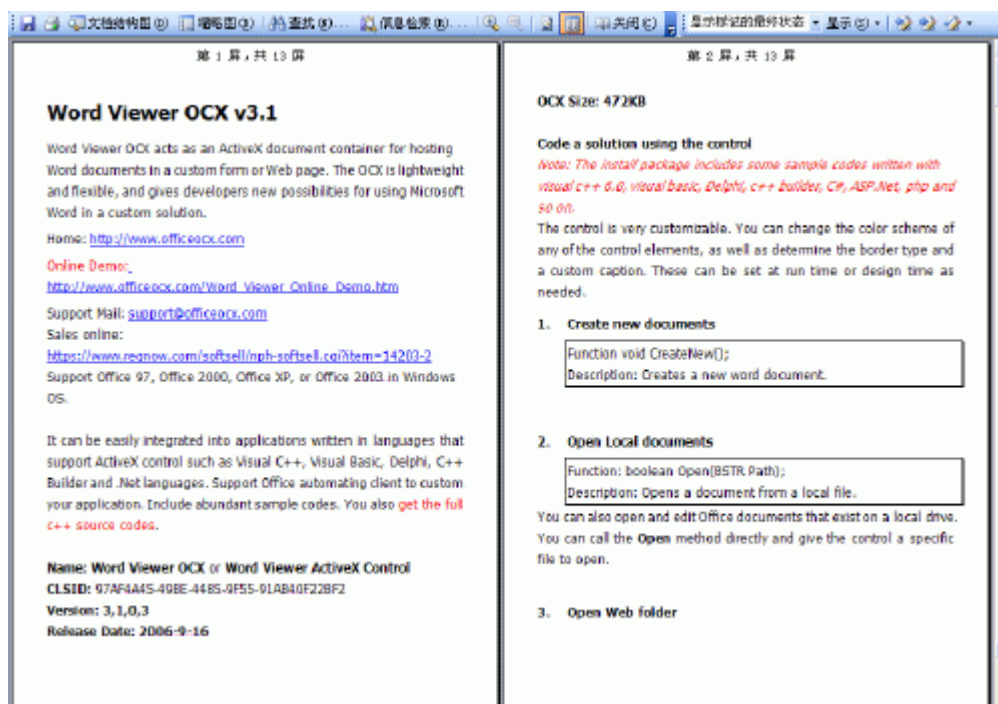
shs.ReleaseDispatch();
shp.ReleaseDispatch();
```

```

m_wordPictureFormat.ReleaseDispatch();
m_wordFillFormat.ReleaseDispatch();
m_wordWrapFormant.ReleaseDispatch();
m_wordShapeRange.ReleaseDispatch();
m_wordShape.ReleaseDispatch();
m_wordShapes.ReleaseDispatch();
m_WordSelection.ReleaseDispatch();
m_WordDoc.ReleaseDispatch();
m_WordApp.ReleaseDispatch();

```

### Screenshot:



### Other Office ActiveX Controls

**[Office Viewer ActiveX Control](#)** – Office Viewer ActiveX Control enables your application to display and interact with all Microsoft Office files such as Word, Excel, PowerPoint, Project and Visio! Simply place the OCX on your form, then you can have all the office functions immediately. Support Office automating client to custom your application. Include abundant sample codes.

**[PowerPoint Viewer OCX](#)** –PowerPoint Viewer OCX enables your application to play the slideshow file in a form or a web site.

**[Excel Viewer OCX](#)** – Excel Viewer OCX enables your application to display and interact with all Microsoft Excel sheet with Read-Only or Edit properties. Support some important excel automating interfaces. It's very easy to customize your own programs.