

VBScript DLLCall using DynaWrap COM Library

<https://www.drdoobs.com/windows/an-automation-object-for-dynamic-dll-cal/210200078>

Using the DynamicWrapper Object

The **DynaWrapper Object** can be accessed as a **COM Library** after **registering** it.

Using **REGSVR32**, but keep in mind that this **Library is a 32Bit only !**

So you need to use the "**C:\Windows\SysWOW64\regsvr32.exe**" command to start going.

In **VbScript** you can access the **COM object** like this :

```
Dim UserWrap As Object
Set UserWrap = CreateObject("DynamicWrapper")
```

To make the library portable I created this manifest file so you don't have to register it on each PC or Server where you want to run the scripts

Registration Free COM Objects

Registration Free COM Objects technology is native part of the Windows OS since Windows XP.

It uses Manifest files to make the magic to work...

This is the **dynwrap.sxs.manifest** file to use :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">

  <assemblyIdentity
    type="win32"
    name="dynwrap.sxs"
    version="1.0.0.0" />

  <file name="dynwrap.dll">

    <comClass
      description="dynwrap.dll"
      clsid="{202774D1-D479-11d1-ACD1-00A024BBB05E}"
      threadingModel="Apartment"
      progid="DynamicWrapper"/>

  </file>
</assembly>
```

The **Manifest** file is included in the **ZIP file** for **download**.

VBScript DLLCall using DynaWrap COM Library

Initiating the RegFree COM Objects in VBScripts

You can access the **COM Library RegFree methods** and **Properties** like this :

```
Option Explicit

' USE 32Bit SysWOW64 script host only !!!

Dim fso, objShell, ScriptPath, ScriptDir

ScriptPath = WScript.ScriptFullName
Set fso = CreateObject("Scripting.FileSystemObject")
ScriptDir = fso.GetParentFolderName(ScriptPath)

Set objShell = CreateObject("Wscript.Shell")
' MsgBox objShell.CurrentDirectory ' show WorkingDirectory
objShell.CurrentDirectory = ScriptDir ' set WorkingDirectory

Dim oActCtx
Set oActCtx = CreateObject("Microsoft.Windows.ActCtx")
oActCtx.manifest = "dynwrap.sxs.manifest"

Dim obj
Set obj = oActCtx.CreateObject("DynamicWrapper")
```

Make sure to put the **COM Library DLL** and the **Manifest file** in the **same folder** as where you **run the scripts**

DLL Calls using VBScripts

The **COM Library** has on **1 Method** called **Register** to access the **Windows API's**. This is the way you can call **external DLL Functions**

```
[result = ] object.Register <dllname>, <procname> [,<tag>,...]
```

Messagebox Example :

```
int MessageBox(HWND hWnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType);
```

You can find the documentation at the Microsoft site :

<https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-messagebox>

Call the **USER32.DLL** that hold the **MessageBoxA** function like this :

```
obj.register "USER32.DLL", "MessageBoxA", "I=HsSu", "f=s", "R=1"
```

To call the functions you need to do some **Type conversion** based on this Input table options :

There are only **3 DynaWrap Parameters**:

i => The number and data type of the function's parameters (n.b., if no paramaters, **LEAVE OUT** this "i=" argument)

f => Type of call stdcall or cdecl. (Default to _stdcall. If that doesn't work use _cdecl).

r => Return data type.

VBScript DLLCall using DynaWrap COM Library

Flag	Parameter Type
a	IDispatch* (Automation object)
c	unsigned char
d	8 byte real
f	4 byte real
k	IUnknown* (COM object)
h	handle (4 bytes)
l	long (4 bytes)
p	pointer
s	string
t	short (2 bytes)
u	unsigned int
w	wide string

The **Data Type Declarations** are:

c => VT_I4: c signed char

d => VT_R8: d 8 byte real

f => VT_R4: f 4 byte real

h => VT_I4: h HANDLE

l => VT_I4: l long

p => VT_PTR: p pointer

s => VT_LPSTR: s string

t => VT_I2: t short

u => VT_UINT: u unsigned int

w => VT_LPWSTR: w wide string

r => VT_BYREF: (pass by reference) for strings only.

Over time some data types have been added see the **'r'** for example

The **'f'** **Call Format Parameters** looks like this :

Flag	Convention
m	Microsoft compatible (exclusive with b)
b	Borland compatible (exclusive with m)
s	Standard call _stdcall (exclusive with c)
c	C Call _cdecl (exclusive with s)
4	4 byte real value returned in ST (exclusive with 8)
8	8 byte real value returned in ST (exclusive with 4)

When all parameters are set correct to do the final invoke of the function you need to do it like this :

VBScript DLLCall using DynaWrap COM Library

```
obj.MessageBoxA Null, "Hello World", "VBScript", 3
```

Here is the example of a **MessageBox** call and adding **Text** to your **Clipboard**

```
Option Explicit
```

```
' USE 32Bit SysWOW64 script host !!!
```

```
Dim fso, objShell, ScriptPath, ScriptDir
```

```
ScriptPath = WScript.ScriptFullName
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
ScriptDir = fso.GetParentFolderName(ScriptPath)
```

```
Set objShell = CreateObject("Wscript.Shell")
```

```
' MsgBox objShell.CurrentDirectory ' show WorkingDirectory
```

```
objShell.CurrentDirectory = ScriptDir ' set WorkingDirectory
```

```
Dim oActCtx
```

```
Set oActCtx = CreateObject("Microsoft.Windows.ActCtx")
```

```
oActCtx.manifest = "dynwrap.sxs.manifest"
```

```
Dim obj
```

```
Set obj = oActCtx.CreateObject("DynamicWrapper")
```

```
Msgbox "You are calling RegFree Windows API's from VBScript !"
```

```
' call MessageBoxA(), first register the API function
```

```
obj.register "USER32.DLL", "MessageBoxA", "I=HsSu", "f=s", "R=1"
```

```
Msgbox "Press YES to copy Text to your Clipboard + press CTRL + V"
```

```
' call the MessageBoxA
```

```
Dim val
```

```
val = obj.MessageBoxA (Null, "MessageBox (ANSI)", "VBS Next Level From DynaWrap  
Object - Check your Clipboard !!", 3)
```

```
Set obj = Nothing
```

```
Set oActCtx = Nothing
```

```
If val = 6 Then
```

```
msgbox "You pressed YES data copied ..."
```

```
' clipboard set data
```

```
Dim oActCtx1
```

```
Set oActCtx1 = CreateObject("Microsoft.Windows.ActCtx")
```

```
oActCtx1.manifest = "dynwrap.sxs.manifest"
```

```
Dim data, objc, b, c
```

```
Set objc = oActCtx1.CreateObject("DynamicWrapper")
```

```
data = "VBScript has copied data using Win API Clipboard :-)"
```

```
objc.register "kernel32.dll", "GlobalAlloc", "i=uu", "f=s", "r=1"
```

```
objc.register "kernel32.dll", "GlobalLock", "i=l", "f=s", "r=1"
```

```
objc.register "kernel32.dll", "lstrcpy", "i=hs", "f=s", "r=h"
```

```
objc.register "user32.dll", "OpenClipboard", "i=h", "f=s", "r=1"
```

```
objc.register "user32.dll", "EmptyClipboard", "f=s", "r=1"
```

```
objc.register "user32.dll", "SetClipboardData", "i=uh", "f=s", "r=1"
```

```
objc.register "user32.dll", "CloseClipboard", "f=s", "r=1"
```

VBScript DLLCall using DynaWrap COM Library

```
b = objc.lstrcpy(objc.GlobalLock(objc.GlobalAlloc(0, cint(len(data)+1))),
cstr(data))

c = objc.OpenClipboard(0)
c = objc.EmptyClipboard()
c = objc.SetClipboardData(1, b)
c = objc.CloseClipboard()

Set objc = nothing

ElseIf val = 7 Then
    msgbox "You pressed NO, nothing happened ... "
Else
    msgbox "You pressed CANCEL"
End If
```

Of course the **DynaWrap COM Library** can be used by any **COM Compliant scripting or programming language**.

```
; AutoIt DynWrap Example (RegFree)
Local $oActCtx = ObjCreate("Microsoft.Windows.ActCtx")
$oActCtx.manifest = @ScriptDir & "\dynwrap.sxs.manifest"

$userWrap = $oActCtx.CreateObject("DynamicWrapper")

Local $oActCtx = ObjCreate("Microsoft.Windows.ActCtx")
$oActCtx.manifest = @ScriptDir & "\dynwrap.sxs.manifest"

$userWrap.Register("MSVCRT.DLL", "sin", "f=mc8", "i=d", "r=d")
$ret = $userWrap.sin($iVal)
MsgBox(0, "DynWrap (RegFree)", "Sinus : " & $ret & @CRLF)

; VbScript Example (Refree)
Local $Code = 'Option Explicit'
$code=$code & @CRLF & ''
$code=$code & @CRLF & 'Dim oActCtx'
$code=$code & @CRLF & 'Set oActCtx = CreateObject("Microsoft.Windows.ActCtx")'
$code=$code & @CRLF & 'oActCtx.manifest = "dynwrap.sxs.manifest"'
$code=$code & @CRLF & ''
$code=$code & @CRLF & 'Dim UserWrap'
$code=$code & @CRLF & 'Set UserWrap = oActCtx.CreateObject("DynamicWrapper")'
$code=$code & @CRLF & ''
$code=$code & @CRLF & 'Dim ret, val'
$code=$code & @CRLF & 'val = 1/2'
$code=$code & @CRLF & ''
$code=$code & @CRLF & 'UserWrap.Register "MSVCRT.DLL", "sin", "f=mc8", "i=d",
"r=d"'
$code=$code & @CRLF & 'ret = UserWrap.sin(val)'
$code=$code & @CRLF & 'msgbox "VBScript sin : " & ret'
$code=$code & @CRLF & 'UserWrap.Register "MSVCRT.DLL", "cos", "f=mc8", "i=d",
"r=d"'
$code=$code & @CRLF & 'ret = UserWrap.cos(val)'
$code=$code & @CRLF & 'msgbox "VBScript cos : " & ret'
$code=$code & @CRLF & 'UserWrap.Register "MSVCRT.DLL", "sinh", "f=mc8", "i=d",
"r=d"'
$code=$code & @CRLF & 'ret = UserWrap.sinh(val)'
$code=$code & @CRLF & 'msgbox "VBScript sinh : " & ret'
$code=$code & @CRLF & 'UserWrap.Register "MSVCRT.DLL", "cosh", "f=mc8", "i=d",
"r=d"'
$code=$code & @CRLF & 'ret = UserWrap.cosh(val)'
$code=$code & @CRLF & 'msgbox "VBScript cosh : " & ret'
```

VBScript DLLCall using DynaWrap COM Library

```
Local $vbs = ObjCreate("ScriptControl")  
$vbs.language="vbscript"  
$vbs.addcode($code)
```



Enjoy !!