

VBScript Macros in TurboCAD V11

Introduction

VBA Macros were taken out of TurboCAD in V11, so in order to get working macros we need to resort to VBScript instead. This pack contains a number of files that ...

- Show you how to make your favourite scripts easily available from the Internet Explorer palette in TurboCAD
- Demonstrate some simple VBScripts in TurboCAD

Requirements

- I've only tested briefly, but this seems to work OK in V9, V10, and V11. I suspect that this works in most versions of V7 onwards, including Standard and Deluxe lines. The only real difference will be the install path.
- Throughout this document the term [YourTCPATH] is used to denote where your copy of TurboCAD is installed. For TC11 Pro this would usually be C:\Program Files\IMSI\TCWP11, but it will depend on which version you have and whether you installed in a custom location - so adjust as necessary.
- In order for scripts to work in TC, we use the internet palette. If you have installed Windows XP SP2 in particular (other versions as well to a degree) the security may be beefed up on your computer to the extent this is blocked. What you need to do is go to **Internet Explorer > Tools > Options > Advanced > Scroll down to Security** and make sure "Allow active content to run in files in My Computer" is **ON**.

Installing

- To install unzip TCVBScriptCentre.zip in [YourTCPATH]\Internet. It should create a few new files and a Folder called VBScripts.
- Check that the changes happened, in TC open the Internet Explorer palette. By default it will be open at [YourTCPATH]\Internet\palette.html ... click in the address bar and edit this to read [YourTCPATH]\Internet\palette_modV11.html* and press Enter.

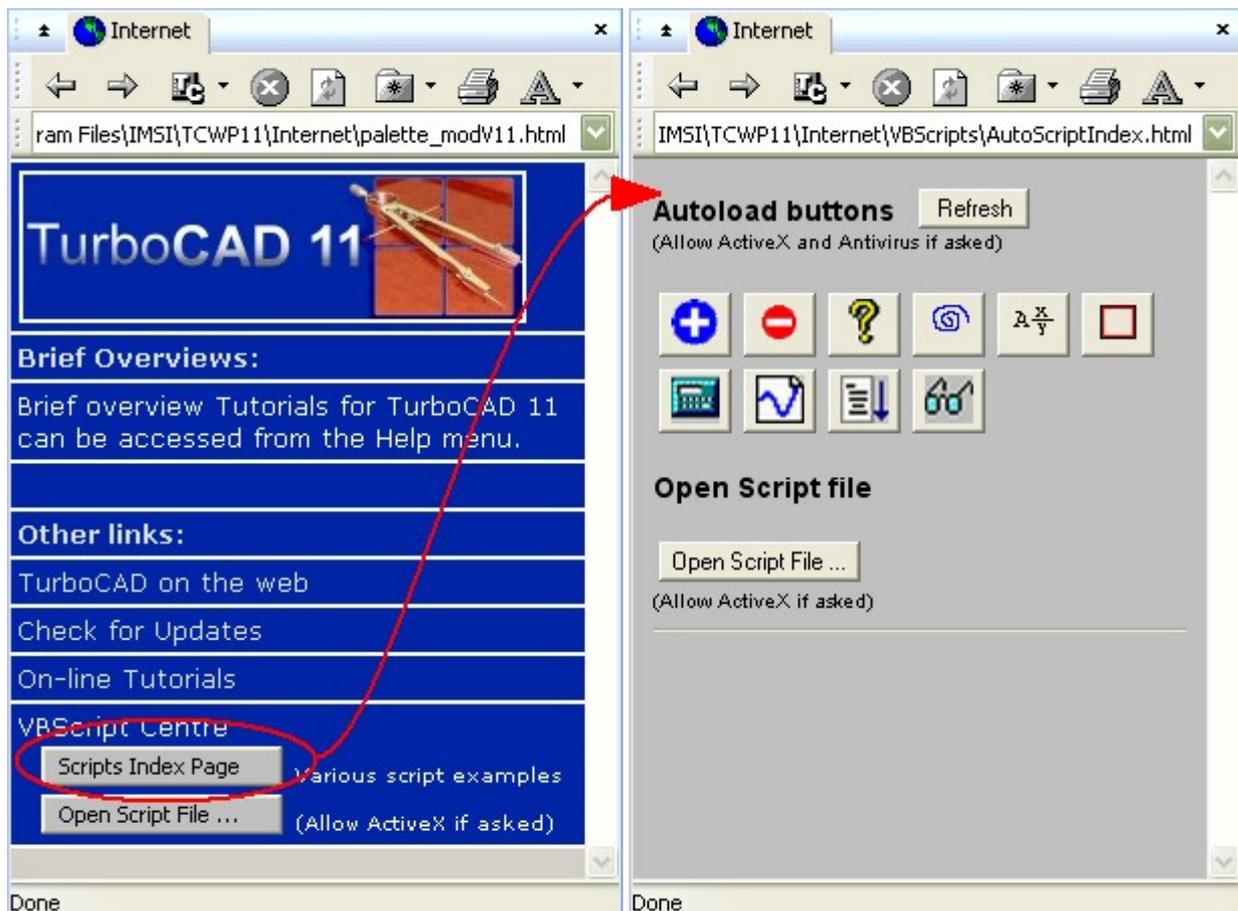
* or palette_modV10.html for V10, palette_modV9.html for V9. Similar palettes can be done for earlier versions.

- If this brings up couple of extra items at the bottom of the page we are good to go!

Running Scripts from the Internet Palette

VBScripts are run from the Internet Explorer palette in TC. After installing, and changing the palette the address to ...\\palette_modV11.html as outlined above, it should now look like the left hand picture below. The thing to look for is the 'VBScript Centre' at the bottom of the page. It contains two new buttons

- **Scripts Index Page**, clicking this button takes you to an index page of example VBScripts, accessible by buttons (right hand picture below).
- **Open Script File**, clicking this button allows you to open and run any html type script in TurboCAD. By default the file browser opens in the /VBScripts folder created during the Install.



Scripts Index Page (VBScripts\AutoScriptIndex.html)

Now we will go into a bit more detail with the page opened by first of the **Scripts Index Page** button. One of the advantages of running scripts from the internet palette is it allows you can organise your scripts as links or buttons on an html page in any way you please. An example of this is AutoScriptIndex.html which has been set up to make scripts easily accessible

Autoload buttons

These buttons are specially formatted scripts that are automatically loaded by the

page itself. They all perform very simple tasks (hover the mouse over them to get a tool tip indicating their function), though they might just as easily perform complex tasks.

New scripts can be added to this page easily, the method is probably best explained by example...

- During installation an example was included. Look in [YourTCPATH] \Internet\VBScripts for the file called **AutoloadExample_CopyInPlace.zip**, and unzip it in VBScripts.
- Back in the palette, click **Refresh** and a new button should get added automatically. Note you may get **warnings** from IE regarding ActiveX and from your Antivirus regarding the script - You must 'allow' or 'authorize' in both cases.
- Note, to change the order of buttons you can edit the filenames. All Autoload files take the form *Autoload####_ClassName.vbs* They are effectively loaded in the order set by the #### number - so changing that changes the order. Change the #### section only and ensure that it has four digits.

See the Developer section below for instructions on creating Autoload scripts.

Open Script button

This button allows you to open any html page in the internet palette, that may or may not include scripts. This makes it easy to try out new scripts without having to integrate them into TC.

(Note the Open Script File button in palette_modV11.html is very similar)

Customizing the Internet Palette pages

There are three useful ways to customise this palette

- The easiest customisation is to to change it's Home page - just browse to the page you want, then click the **TC Links > Options > Current > OK** (TC Links is the drop-down button with the TC icon).

For example, you can set the Script Index page as your Home, and your scripts will always only be a single click away. If can't browse to the file you want easily use **Open Script File**.

- Alternatively, you can easily add script pages to your Favorites by browsing to them, then **right-click in the palette area > Add to Favorites**.
- Finally you can add pages/scripts to the TC Links drop-down menu. Open the TC Explorer palette, and edit the **Settings > Built-in > Web_Links** table. To add links make sure you name items so that they keep the Link_1, Link_2, ... Link_## sequence going - it doesn't seem to like gaps.

For Developers

This section explains what's going on behind the scenes in case you want to write your own html/vbs scripts, and describes a coding specification that will allow your scripts to be automatically loaded as buttons into the AutoScriptIndex.html page.

AutoScriptIndex.html and AutoScriptIndex.vbs

These two files between them comprise the 'Autoload system' which, if you follow coding conventions below, allows you to create packages that when unzipped in the correct Folder are automatically added as buttons into the AutoScriptIndex.html page in TurboCAD - a bit like adding a new Dragger.

Edit these two files with caution - in general you should never need to.

Two types of script

This document describes two types of scripts - **Stand-alone VBS** Autoload scripts perform simple functions that require no further user interaction other than pressing the script button; and **HTML Embedded VBS** Autoload scripts which are generally more complex and use html pages in place of VBA dialogs to acquire additional user input. From a user point of view they appear to be one and the same on the AutoScriptIndex.html page, but behind the scenes the two types require slightly different coding procedures as outlined in the following sections.

Autoload 'packages'

You can make scripts easily installable for users by zipping them into specially formatted packages which include

- A specially named .vbs file, taking the form Autoload####_ClassName.vbs where #### represents four digits that control loading order.
- A 24 x 23 (w x h) BMP or GIF for the button image, usually set to unzip in the subfolder ButtonImages\ (optional if you want to reuse an image already there)
- Optionally any number of additional html and vbs files

If setup correctly, such as **AutoloadExample_CopyInPlace.zip**, then all the user needs to do to install the script is unzip it in VBScripts and click **Refresh** in the Script Index page.

Autoload####_ClassName.vbs

Autoload####_ClassName.vbs is the most important file in the Autoload package, in the example package it is called Autoload0013_CopyInPlace.vbs and we will now look at it in detail - it's actually very simple

- It defines a class, with *exactly* the same `ClassName` that appears in the file name, so in the example it becomes the statement

```
Class CopyInPlace
    ...
    ...
End Class
```

- Inside the class, there is `Private Sub Class_Initialize` which is where you define five important variables

```
ScriptType      = "vbs"
                 ' "vbs" = Stand-alone script
                 ' "html" = HTML Embedded script

ThisClassName   = "CopyInPlace"
                 ' again exactly the same as ClassName

ToolTipText     = "Copy in Place (Selected Graphics)"
                 ' Tool tip text, when mouse hovered

ImageURL        = "ButtonImages\CopyInPlace.GIF"
                 ' Relative URL to the button image

LinkURL         = ""
                 ' Relative URL to the html page.
                 ' For HTML Embedded scripts only
                 ' (Not used for Stand-alone scripts)
```

- Also inside the class there is

```
Sub ExecScript
    ...
    ' This is the actual function code
    '
    ' Note that this Sub is not used for
    ' HTML Embedded scripts. Their functions are
    ' performed by the page specified in
    ' LinkURL
    ...
End Sub
```

So in conclusion, to set up an Autoload script you need to do three things

- give the Autoload VBS file a correctly formatted name
- Fill in the values in `Sub Class_Initialize`
- For Stand-alone scripts write code in `Sub ExecScript`
- For HTML Embedded scripts, the button simply links to the page specified in `LinkURL` - which may contain code itself, or links to as many other pages and/or scripts as required. The next section contains a bit more information on HTML Embedded scripts.

TC -> html/vbs connection in HTML Embedded scripts

Normally an internet explorer page/script would have absolutely no relation to TurboCAD, so a connection has to be established. For Stand-alone VBS scripts this is handled automatically by AutoScriptIndex.html, but for HTML Embedded VBS scripts you need to establish the connection for each html page separately (and a script might consist of more than one). Taking **BC_Involute.html** as an example, open it in Notepad

The most important code is in the <script> tag at the very top ...

```
Dim Application
'...
Sub window_onload()
    set Application = StTC.objApp
'...
End Sub
```

... and in the <object> tag at the very bottom...

```
<OBJECT height="0" id="StTC" name="StTC" style="DISPLAY: none;
LEFT: 0px; POSITION: absolute; TOP: 300px; Z-INDEX: 104"
type="text/x-scriptlet" width="0" VIEWASTEXT>
    <PARAM NAME="Scrollbar" VALUE="0">
    <PARAM NAME="URL" VALUE="CommonScripts\StTC.htm">
</OBJECT>
```

The <object> tag is something provided by IMSI/SoftDev and at this point I haven't had time figure out what happens in it except that it defines **StTC** as a pointer to the current TC instance and **StTC.objApp** points to it's Application object.

Sub `window_onload()` is run automatically* when the page is opened and as can be seen from the above, effectively sets up a *variable* called `Application` so that it looks and behaves like the `TC.Application object`. (*not auto in javascript I think)

Bit weird yes, but as stated above we need the connection between IE and TC, and it also makes VBA->VBS translation much easier because where you would have used the *Application object* in VBA, you can use the *Application variable* in it's place in VBS.

ActiveDrawing can be set up in a similar way, and the same goes for other common objects and constants.

AutoScriptIndex.html and AutoScriptIndex.vbs *again*

Just to complete the picture, the way this all works is

- When you click Refresh in AutoScriptIndex.html, it runs the WritePage function in AutoScriptIndex.vbs
- This scans the folder VBScripts for files conforming to the search string

'Autoload###_*.vbs', and overwrites AutoScriptIndex.html with a new version adding new <script> tags for each autoload script found.

Shortcomings

I actually think this system works quite well now, and if you set AutoScriptIndex.htm as your TC Home page it's actually more convenient than macros in many ways. However there are still a few minor problems

- Dialogs - while using HTML Embedded scripts is a bit of a workaround, they are really only practical for basic forms. Also I have only been able to make them work from within the palette - attempts at converting them to HTA applications didn't work. So we need a decent method of creating standalone dialogs/applications that can be easily launched from a VBScript, preferably free. Java Netbeans? Python? See more comments in the Translating tips below.
- There is a touch of weirdness running scripts from the Internet palette, because the palette seems to end up with the system focus instead of returning it to the TC drawing area. So they can be a little bit odd to use, though it's not really serious.
- If anyone knows of a free debugger/ide for VBScript please let me know.

Tips Translating VBA to VBS

If you want to translate your VBA macros to VBS scripts, this may help a bit

Code

Code is actually the easiest bit, provided it's embedded in an html page. Here's how I go about it

- You need the <script> ... Sub window_onload and <object> tags set up as outlined above. These tags are more or less the same for all html/vbs pages that work with TurboCAD - the only exception being you may have to edit the path in the URL <PARAM> if you have the script in a different folder.
- In all Dim statements delete the 'as ????' part (in VBScript everything is a variant)
- Look for With ... (Each ... In) ... End With statement and replace them either with a For Each ...Next loop or object variables.
- Optionally change any other lines where you know VBScript doesn't support the feature
- Try running the script, and fix errors as required. Importantly if the error is due to a missing top-level object or constant, just Dim and Set at the start of the Sub.
- Usually you can get you code working pretty quick like this.

Dialogs

OK, now for the bad news. At this stage to the best of my knowledge Dialogs have to be completely re-written to live as html pages in the palette. No shortcuts I'm afraid :-(. Here is my very short list of hints

- Try to stick to standard html form controls as far as you can, though note for our purposes, they do not need to be inside <form> tags
- I haven't tried it yet, but for multipage dialogs/wizards, I think I would use several <div> tags and use vbscript to hide/unhide different sections as required.
- If you want to use controls not normally in html (like treeviews etc) have a look at the examples provided by IMSI/SoftDev in VBScripts - there are some sophisticated pages here that I have not had the time to analyse yet.
- I haven't been able to get an .hta to work (trying to make it more like a dialog). Anyone able to?

Some VBS Resources on the web

- VBScript User Guide
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbscripttoc.asp>
- VB Script Language Reference
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbscripttoc.asp>
- VB Script debugger
http://msdn.microsoft.com/library/en-us/sdbug/Html/sdbug_1.asp?frame=true
(I cant get it to work - freezes TC when it hits a bug/break)
- Differences between VBA and VBS
<http://support.microsoft.com/default.aspx?scid=kb;en-us;190273>
- html forms
<http://www.w3.org/TR/REC-html40/interact/forms.html>
- If anyone knows how to reuse VBA Forms please let me know - maybe this recent article at the TCDRC gives us a hint
<http://www.bcitool.com/TCDRC/Tutorials/Using%20Forms%20with%20TurboCAD%20DLLs.pdf>