

Re: VBScript, OpenTextFile(), Strings and Unicode – Possible?

Source:

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.vbscript/2007-10/msg00298.html>

- *From:* "Paul Randall" <paulr901@xxxxxxxxxxxxx>
 - *Date:* Tue, 16 Oct 2007 17:22:39 -0600
-

"Axel Dahmen" <KeenToKnow@xxxxxxxxxxxxxxxxxxxxx> wrote in message news:OexKEYEEIHA.1208@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hi,

I've written a VBScript to manually update the version number in an AssemblyInfo.cs file. But all I get from the ReadAll() function is rubbish. The string I'm assigning the function result to only contains loads of empty squares when watched from within Visual Studio Debugger.

I can't even write the string back without modifications into the AssemblyInfo.cs file. Visual Studio shows only a bunch of Chinese characters when opening the saved file. (Perhaps the result of a new espionage attempt by the Axis of Evil? :))

Moreover, I cannot modify the string. To VBScript the string doesn't seem contain anything near to a valid character.

Can anybody help me on how to access and save the Unicode file's content from within VBScript?

TIA,
Axel Dahmen

Here's my test source code:

```
' VBScript source code
Const ForReading = 1, ForWriting = 2, TristateTrue = -1
Dim fso,stream,strg,re

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

Re: VBScript, OpenTextFile(), Strings and Unicode – Possible?

```
fso.OpenTextFile("C:\Temp\AssemblyInfo.cs",ForReading,False,TristateTrue)
strg = stream.ReadAll()
stream.Close
Set stream = Nothing
```

```
Set re = New RegExp
re.Global = True
re.IgnoreCase = False
re.Multiline = True
re.Pattern = "\d+\.\d+\.\d+\.\d+"
strg = re.Replace(strg,"9.9.9.9")
```

```
Set stream =
fso.OpenTextFile("C:\Temp\AssemblyInfo2.cs",ForWriting,True,TristateTrue)
stream.Write(strg)
stream.Close
Set stream = Nothing
```

One person's rubbish is another person's treasure :-)

I suspect that the file in question is encoded in some form of Unicode or some other encoding. Various encodings of HTML files can be displayed successfully because the name of the encoding is included in the file. I suppose your AssemblyInfo2.cs file might include encoding information; I don't know what program reads that file's format.

If you think that you would recognise the name of the encoding, you can try setting VBScript's locale to various settings and see how the string looks in a message box. Search the scripting help file for the two-word phrase locale id, for a list of locale IDs and descriptions.

Microsoft programs that write Unicode files, such as Notepad in WXP, write a Byte Order Mark (BOM), the two bytes FFFE, at the beginning of the file, followed by two bytes per character for the text in Notepad. You can get some clues to what you have by looking at the file with a hexadecimal file editor.

–Paul Randall