

# VBScript

**VBScript** ("*Microsoft Visual Basic Scripting Edition*") is an Active Scripting language developed by Microsoft that is modeled on Visual Basic. It allows Microsoft Windows system administrators to generate powerful tools for managing computers with error handling, subroutines, and other advanced programming constructs. It can give the user complete control over many aspects of their computing environment.

VBScript uses the Component Object Model to access elements of the environment within which it is running; for example, the FileSystemObject (FSO) is used to create, read, update and delete files. VBScript has been installed by default in every desktop release of Microsoft Windows since Windows 98,<sup>[1]</sup> in Windows Server since Windows NT 4.0 Option Pack,<sup>[2]</sup> and optionally with Windows CE (depending on the device it is installed on).

A VBScript script must be executed within a host environment, of which there are several provided with Microsoft Windows, including: Windows Script Host (WSH), Internet Explorer (IE), and Internet Information Services (IIS).<sup>[3]</sup> Additionally, the VBScript hosting environment is embeddable in other programs, through technologies such as the Microsoft Script Control ([https://msdn.microsoft.com/en-us/library/aa227633\(v=vs.60\).aspx](https://msdn.microsoft.com/en-us/library/aa227633(v=vs.60).aspx)) (`msscript.ocx`).

## Contents

### History

### Environments

### Functionality

Language features

VBScript functionalities

Additional functionality

### Development tools

### Uses

### See also

### References

### External links

## VBScript



|   |  |
|---|--|
| <b>Developer</b>  | <u>Microsoft</u>   |
| <b>First appeared</b>                                   | <u>1996</u>  |
| <b>Stable release</b>                                   | <u>5.8</u>   |
| <b>OS</b>   | <u>Windows</u>   |
| <b>Filename extensions</b>                              | <u>.vbs</u> , <u>.vbe</u> , <u>.wsf</u> , <u>.wsc</u> ( <u>.asp</u> , <u>.hta</u> , <u>.htm</u> , <u>.html</u> )   |
| <b>Website</b>  | <u>docs.microsoft.com/en-us/previous-versions/t0aew7h6(v=vs.85)</u> ( <a href="https://docs.microsoft.com/en-us/previous-versions/t0aew7h6(v=vs.85)"><u>https://docs.microsoft.com/en-us/previous-versions/t0aew7h6(v=vs.85)</u></a> ) |
| <b>Major implementations</b>                            |  |
| <u>Windows Script Host</u> , <u>Active Server Pages</u> |  |
| <b>Influenced by</b>                                    |  |
| <u>Visual Basic</u>                                     |  |
| <b>Influenced</b>                                       |  |
| <u>Windows PowerShell</u>                               |  |

## History

VBScript began as part of the Microsoft Windows Script Technologies, launched in 1996. This technology (which also included JScript) was initially targeted at web developers. During a period of just over two years, VBScript advanced from version 1.0 to 2.0, and over that time it gained support from Windows system administrators seeking an automation tool more powerful than the batch language first developed in the early 1980s.<sup>[4]</sup>

On March 6, 1988, Alan Cooper showed Bill Gates his shell prototype that allowed widgets to be added dynamically. On March 20, 1991, Microsoft adopted "Quick Basic". This allowed users to create Windows apps quickly and easily with a GUI. Finally, on August 1, 1996, Internet Explorer was made with features that included VBScript.<sup>[5]</sup>

In version 5.0, the functionality of VBScript was increased with new features including regular expressions; classes; the *With* statement;<sup>[6]</sup> the *Eval*, *Execute*, and *ExecuteGlobal* functions to evaluate and execute script commands built during the execution of another script; a function-pointer system via *GetRef*,<sup>[7]</sup> and Distributed COM (DCOM) support.

In version 5.5, *SubMatches*<sup>[8]</sup> were added to the *regular expression* class in VBScript, to finally allow script authors to capture the text within the expression's groups. That capability had already been available in JScript.

With the advent of the .NET framework, the scripting team took the decision to implement future support for VBScript within ASP.NET for web development,<sup>[9]</sup> and therefore no new versions of the VBScript engine would be developed. It would henceforth be supported by Microsoft's *Sustaining Engineering Team*, who are responsible for bug fixes and security enhancements. For Windows system administrators, Microsoft suggests migrating to Windows PowerShell. However, the scripting engine will continue to be shipped with future releases of Microsoft Windows and IIS.

## Environments

---

When employed for client-side web development in Microsoft Internet Explorer, VBScript is similar in function to JavaScript. It is used to write executable functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page, to perform tasks not possible in HTML alone. However, other web browsers such as Firefox and Opera do not have built-in support for VBScript. This means that where client-side scripting and cross-browser compatibility are required, developers usually choose JavaScript over VBScript.

VBScript is also used for server-side processing of web pages, most notably with Microsoft Active Server Pages (ASP). The ASP engine and type library, `asp.dll`, invokes `vbscript.dll` to run VBScript scripts. VBScript that is embedded in an ASP page is contained within `<%` and `%>` context switches. The following example of an ASP page with VBScript displays the current time in 24-hour format, but it is also used as the default script for error messages with the Windows operating systems.

```
1  <% Option Explicit %>
2  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
3  <html>
4      <head>
5          <title>VBScript Example</title>
6      </head>
7      <body>
8          <div><%
9              ' Grab current time from Now() function.
10             ' An '=' sign occurring after a context switch (<%) is shorthand
11             ' for a call to the Write() method of the Response object.
12             Dim timeValue = Now %>
13             The time, in 24-hour format, is
14                 <%=Hour(timeValue)%>:<%=Minute(timeValue)%>:<%=Second(timeValue)%>.
15         </div>
```

```
16     </body>
17 </html>
```

VBScript can also be used to create applications that run directly on a workstation running Microsoft Windows. The simplest example is a script that makes use of the Windows Script Host (WSH) environment. Such a script is usually in a stand-alone file with the file extension `.vbs`. The script can be invoked in two ways: `Wscript.exe` is used to display output and receive input through a GUI, such as dialog and input boxes. `Cscript.exe` is used in a command line environment.

VBScript can be included in two other types of scripting files: Windows Script Files, and HTML Applications.

A Windows Script File (WSF) is styled after XML. A WSF file can include multiple VBS files. As a result, WSF files provide a means for code reuse: a library of classes or functions can be stored in one or more `.vbs` files, which can be included in `.wsf` files, so the functionality can be reused in a modular way. The files have extension `.wsf` and can be executed using `wscript.exe` or `cscript.exe`, just as a `.vbs` file can.

An HTML Application (HTA) is styled after HTML. The HTML in the file is used to generate the user interface, and a scripting language such as VBScript is used for the program logic. The files have extension `.hta` and can be executed using `mshta.exe`.

VBScript (and JScript) can also be used in a Windows Script Component, an ActiveX-enabled script class that can be invoked by other COM-enabled applications.<sup>[10]</sup> These files have extension `.wsc`.

VBScript is simple to create and can be coded using an easy to use text editor like Notepad. A simple VBScript saved by the `.vbs` extension can be run on a Windows system by double clicking it.

## Functionality

---

### Language features

The language of VBScript is modeled on Visual Basic, and therefore can be reviewed using similar categories: procedures, control structures, constants, variables, user interaction, array handling, date/time functions, error handling, mathematical functions, objects, regular expressions, string manipulation, and so on.<sup>[11]</sup>

The following are some key points of introduction to the VBScript language by microsoft

A "procedure" is the main construct in VBScript for separating code into smaller modules. VBScript distinguishes between a function, which can return a result in an assignment statement, and a subroutine, which cannot. Parameters are positional, and can be passed by value or by reference.

Control structures include the usual iterative and conditional Do Loops, If-Then-Else statements, and Case statements, with some more complex variants, such as ElseIf and nested control structures.

As a memory aid in coding, and certainly for readability, there are a large number of constants, such as True and False for logical values, vbOKCancel and vbYesNo for MsgBox codes, vbBlack and vbYellow for color values, vbCR for the carriage return character, and many others.

Variables have "Variant" type by default, but it is possible (and sometimes necessary) to force a particular type (integer, date, etc.) using conversion functions (CInt, CDate, etc.)

User interaction is provided through the functions `MsgBox` and `InputBox` which provide a simple dialogue box format for messages and input. Both functions display prompting messages, with the former returning a standard response, and the latter returning one user-supplied text or numeric value. For more elaborate GUI interaction with controls, VBScript can be used in combination with HTML, for example, in an [HTML Application](#). Event-driven forms are not supported as in Visual Basic or Visual Basic for Applications.

Names are not case-sensitive. However, it is considered a best practice of VBScript style to be consistent and to capitalize judiciously.

## VBScript functionalities

When hosted by the [Windows Script Host](#), VBScript provides numerous features which are common to scripting languages, but not available from [Visual Basic 6.0](#). These features include direct or easy access to:

- Named and unnamed command line arguments,
- `Stdin` and `stdout`, which could be redirected,
- `WSH.Echo` which writes to the console and cannot be redirected,
- `WSH.ExitCode` which can be tested from DOS batch files, or by the process which invoked the script file,
- Network Printers,
- Network Shares,
- Special Folders, e.g. Desktop, Favorites, MyDocuments and so on,
- Network user information, such as group membership,
- Methods for runtime execution of text defined at runtime: `Eval` and `Execute`,
- Methods for executing scripts on remote machines,
- Windows Management Instrumentation (WMI)
- Functionality for embedding a VBScript engine in other applications, using a widely known language.

CScript, the command line runner, provides options for:

- Interactive or Batch Mode,
- Invoking Debug Mode from the command line,
- Error reporting including the line number.

## Additional functionality

File system management, file modification, and streaming text operations are implemented with the Scripting Runtime Library `scrrun.dll`. This provides objects such as `FileSystemObject`, `File`, and `TextStream`, which expose the Windows file system to the programmer.

Binary file and memory I/O are provided by the "ADODB.Stream" class, which can also be used for string builders (to avoid excessive string concatenation, which can be costly), and to interconvert byte arrays and strings. Database access is made possible through [ActiveX Data Objects](#) (ADO), and the [IIS Metabase](#) can be manipulated using the `GetObject()` function with sufficient permissions (useful for creating and destroying sites and virtual directories). XML files and schemas can be manipulated with the [Microsoft XML Library Application Programming Interfaces](#) (`msxml6.dll`, `msxml3.dll`), which also can be used to retrieve content from the World Wide Web via the `XMLHTTP` and `ServerXMLHTTP` objects (class strings "MSXML2.XMLHTTP.6.0" and "MSXML2.ServerXMLHTTP.6.0", respectively).

Functionality can also be added through ActiveX technologies. Security concerns have led to many ActiveX controls being blacklisted in the Internet Explorer process by Microsoft, which deploys the killbit via monthly Windows security updates to disable vulnerable Microsoft and third party code.<sup>[12][13]</sup>

Programmers can utilize the extensibility via COM (ActiveX) modules to specifically equip the Script Host and VBScript with required or desired functions. The "VTool" component, for instance, adds a number of dialog windows, binary file access, and other functionality.<sup>[14]</sup>

## Development tools

---

Microsoft does not routinely make available an IDE (Integrated Development Environment) for VBScript, although the Microsoft Script Editor has been bundled with certain versions of Microsoft Office.

For debugging purposes the Microsoft Script Debugger can still be used in current Windows versions, even though the tool has not been updated in years. It allows the user to set break points in the VBScript code but the user interface is more than clumsy.

There are VBScript debuggers available from third-party sources,<sup>[15][16]</sup> and many text editors offer syntax highlighting for the language.

During execution, when an error occurs, the script host issues a message stating the type of error and the number of the offending line.

## Uses

---

Although VBScript is a general-purpose scripting language, several particular areas of use are noteworthy. First, it is widely used among system administrators in the Microsoft environment.<sup>[17]</sup> This situation may change with the promotion and increased use of Windows PowerShell. Second, VBScript is the scripting language for Quick Test Professional, a test automation tool.<sup>[18]</sup> A third area to note is the adoption of VBScript as the internal scripting language for some embedded applications, such as industrial operator interfaces and human machine interfaces. The hierarchical DBMS InterSystems Caché (which has its roots in the language MUMPS) also supports an implementation of VBScript, Cache BASIC, for programming stored code.<sup>[19]</sup>

VBScript omits several useful features of the full Visual Basic, such as strong typing, extended error trapping and the ability to pass a variable number of parameters to a subroutine. However, its use is relatively widespread because it is easy to learn and because those who implement code in the language need not pay royalties to Microsoft as long as the VBScript trade mark is acknowledged. When an organization licenses Visual Basic for Applications (VBA) from Microsoft, as companies such as Autodesk, StatSoft, Great Plains Accounting and Visio (subsequently acquired by Microsoft) have done, it is allowed to redistribute the full VBA code-writing and debugging environment with its product.

VBScript is used in place of VBA as the macro language of Outlook 97.

VBScript can be effectively used for automating day to day office tasks as well as monitoring in the Windows-based environment. It's a first choice tool for server administrators, MS Office users and other tech contributors. It can also be used in collaboration with ADODB ActiveX Data Objects (ADODB) for effective database connectivity.

## See also

---

- [AppleScript](#)
- [FastTrack Scripting Host](#)
- [HTML Components](#)
- [JavaScript](#)
- [JScript .NET](#)
- [JScript](#)
- [PerlScript](#)
- [Windows PowerShell](#)
- [Windows Script File](#)

## References

---

1. *WSH Version Information* ([http://msdn2.microsoft.com/en-us/library/x66z77t4\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/x66z77t4(VS.85).aspx)), on MSDN
2. *VBScript Version Information* (<http://msdn.microsoft.com/en-us/library/4y5y7bh5.aspx>), on MSDN
3. *What is VBScript?* (<http://msdn.microsoft.com/en-us/library/1kw29xwf.aspx>), in MSDN Library
4. *86-DOS*
5. "The History of Visual Basic" (<http://www.johnsmiley.com/visualbasic/vbhistory.htm>). [www.johnsmiley.com](http://www.johnsmiley.com).
6. *Visual Basic Scripting Edition: With Statement* (<http://msdn2.microsoft.com/en-us/library/zw39ybk8.aspx>), on MSDN
7. *GetRef Function* (<http://msdn2.microsoft.com/en-us/library/ekabbe10.aspx>), on MSDN
8. *Visual Basic Scripting Edition: SubMatches Collection* (<http://msdn2.microsoft.com/en-us/library/y27d2s18.aspx>), on MSDN
9. *What About VBScript?* (<http://msdn.microsoft.com/en-us/library/ms974588.aspx#scripting0714topic1>), within the article *Introducing JScript .NET* by Andrew Clinick of Microsoft Corporation, in Scripting Clinic on MSDN (July 14, 2000)
10. *Introducing Windows Script Components* ([http://msdn2.microsoft.com/en-us/library/07zhfkh8\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/07zhfkh8(VS.85).aspx)), on MSDN
11. "VBScript Features" ([http://msdn.microsoft.com/en-us/library/273zc69c\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/273zc69c(v=VS.85).aspx)). [msdn.microsoft.com](http://msdn.microsoft.com).
12. "How to stop an ActiveX control from running in Internet Explorer" (<http://support.microsoft.com/kb/240797>). Microsoft. 2007-08-24. Retrieved 2009-06-29.
13. "Microsoft Security Advisory (960715): Update Rollup for ActiveX Kill Bits" (<http://www.microsoft.com/technet/security/advisory/960715.mspx>). Microsoft. 2009-01-17. Retrieved 2009-06-29.
14. "VTool" script component (<http://eriedel.info/en/files/vtool/vtool.html>) – GUI and functional enhancements for WSH/VBS
15. "VbsEdit - VBScript Editor with Debugger - VBS Editor" (<http://www.vbsedit.com/>). [www.vbsedit.com](http://www.vbsedit.com).
16. Corp., Spline Technologies. "SplineTech VBS Debugger, VBScript Debugger. Debug VBS" ([http://www.remotedebugger.com/vbs\\_debugger/vbs\\_debugger.asp](http://www.remotedebugger.com/vbs_debugger/vbs_debugger.asp)). [www.remotedebugger.com](http://www.remotedebugger.com).
17. Script Center (<https://technet.microsoft.com/en-us/scriptcenter/default.aspx>), Microsoft web site targeting system administration scriptors
18. "Quick Test Professional – Basics of VBScript" ([http://knol.google.com/k/quick-test-professional-software-test-automation-tool#Basics\\_of\\_vbscript](http://knol.google.com/k/quick-test-professional-software-test-automation-tool#Basics_of_vbscript)).
19. "Caché for Unstructured Data Analysis" (<http://www.intersystems.com/cache/technology/components/script/index.html>). InterSystems. Retrieved 2018-09-24.

## External links

---

- [Is VBScript Dead?](https://isvbscriptdead.com) (<https://isvbscriptdead.com>)
  - [VBScript](http://msdn2.microsoft.com/en-us/library/t0aew7h6.aspx) (<http://msdn2.microsoft.com/en-us/library/t0aew7h6.aspx>) in the Microsoft Developer Network
  - [Windows Script 5.6 Documentation](http://www.microsoft.com/downloads/details.aspx?FamilyId=01592C48-207D-4BE1-8A76-1C4099D7BBB9&displaylang=en) (<http://www.microsoft.com/downloads/details.aspx?FamilyId=01592C48-207D-4BE1-8A76-1C4099D7BBB9&displaylang=en>)
  - [VBScript @ codewiki](http://codewiki.wikidot.com/vbscript) (<http://codewiki.wikidot.com/vbscript>)
  - [VBScript @ stealthbot](http://www.stealthbot.net/wiki/VBScript_function) ([http://www.stealthbot.net/wiki/VBScript\\_function](http://www.stealthbot.net/wiki/VBScript_function))
  - [WMI Overview](https://technet.microsoft.com/en-us/library/ee198925.aspx) (<https://technet.microsoft.com/en-us/library/ee198925.aspx>), Microsoft TechNet
- 

Retrieved from "<https://en.wikipedia.org/w/index.php?title=VBScript&oldid=956155332>"

---

This page was last edited on 11 May 2020, at 19:51 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.