Calculations and Design Checks Using an Internet Server

Bill T. Ray, Ph.D., P.E.
Southern Illinois University Carbondale

Abstract

This paper discusses the use of scripting (uncompiled) computer programming languages to provide students with interactive calculations and design examples over the Internet. Many engineering faculty already possess the programming skills necessary to perform these activities for course-related web sites. VBScript (Active Server Pages) and JavaScript (JScript) can perform a variety of programming functions including graphics, calculations, file storage, cookie (local data storage) creation and cookie retrieval.

Introduction

Internet scripting languages open a new conduit to provide students with a broader experience with various design processes. VBScript and JavaScript are relatively easy languages to learn. Anyone with experience in C++ or Visual Basic will have little difficulty making the transition to either of these languages. Those with only FORTRAN skills will find the transition only slightly more difficult. However, before you can use either scripting language you must have at least a rudimentary understanding of html.

Each scripting language offers advantages. VBScript, while it runs slower, allows the author to hide the calculations and store or retrieve data from a server hard disk. VBScript accepts input over the Internet, performs the requested calculations on the server, and then writes an html page that is returned to the student. This process consumes more time than JavaScript processes because of transiting the Internet. However, the student is not afforded the opportunity to view the code. On the other hand, the code in JavaScript is downloaded with the html page. All the calculations are performed on the client machine. No further Internet transit time is required. However, the user can view the code, seeing how the author approached the problem (if they can understand the JavaScript). Since JavaScript runs on the local machine, and does not require the round trip through the server, it is much faster. When you use VBScript over a local area network (LAN) this time factor will not be apparent.

Both languages offer a suite of mathematical functions, including all the standard math functions along with the ability to use arrays. Selected features include:

Parameter	Comments
Variable types	Single and double precision floating point numbers, single and double integers, strings
Math functions	Arithmetic, powers, roots, random number, much more
Logical statements	If, if then, if then else, while, select case
Arrays	Multidimensional
Input	From client (both) or data file (VBScript only)
Output	To screen, to cookie, to data file (VBScript only)

Using Active Server Pages

Active Server Pages (ASP) is a technology developed by Microsoft. It uses the VBScript programming language. This technology functions only on Microsoft's Internet servers: Windows NT or Windows 2000 with Internet Information Server. Since ASP writes a standard html page, it can be viewed on any of the newer Internet browsers, including Internet Explorer, Navigator, or Hot Java. You can tell if a page you are viewing is actually written in ASP by the file extension on the URL. If it ends in ".asp" the page is probably an active server page written in VBScript.

The VBScript language has much in common with Visual Basic, from which it draws its name (Visual Basic Script). However, it is not the same language. It does not contain all the functionality of Visual Basic, but it contains other functionality designed specifically for the Internet (things such as cookie calls).

Learning VBScript or Active Server Pages

Learning VBScript requires a computer capable of running ASP. Currently that includes Windows 95 or 98, or Windows NT or 2000. Windows NT or 2000 offer some distinct advantages over 95 or 98. And, you also need to install Internet Information Server or Personal Web Server from Microsoft. These are available for download from Microsoft. You will also need a program in which to write your programs. This can be as simple as Notepad, or a specialized development environment such as MS Interdev or Visual Studio. Other possibilities are HoTMetaL Pro or FrontPage 2000, although the author is not familiar with these for this application.

There are several books available, but *Beginning Active Server Pages 2.0*[1] is an excellent place to start. It offers good explanations and many example problems you can write and then run. Other books include *Learning VBScript*[2] and *Teach Yourself VBScript in 21 Days*[3]. For the more adventurous, *Professional Active Server Pages 2.0*[4] is a good book.

What JavaScript Pages Can Do

JavaScript pages can do much the same functions as ASP but much faster because all operations are performed on the client machine. Some differences include: 1) All the code is downloaded to the client browser so the client can view the pages. 2) JavaScript cannot save data to the server for future use. Another advantage is that JavaScript is server independent. Since it does its calculations on the client machine, all the web server does is send the page. You are not limited to MS Internet Information Server. As a general rule, JavaScript is preferable to ASP unless there is a specific reason to choose ASP, as noted above.

Learning JavaScript

JavaScript's level of difficulty is comparable to VBScript. It has the advantage of not requiring any particular server or development environment. Write the file in Notepad, save it on your local machine, and run it (or debug it as the case may be). Recommended books include *JavaScript, The Definitive Guide, 3rd Ed.*[5], *JavaScript Essentials*[6], *Instant JavaScript*[7], and *Teach Yourself JavaScript 1.3 in 24 Hours*[8].

Comparing The Two Languages

Feature	**ASP/VBScript**	**JavaScript**
Server Platform	MS Internet Information Server	Any Internet Server
Functions At	Server	Browser
Develop Using	Notepad, MS Visual Interdev	Notepad, MS Visual Interdev
Speed	Slow	Fast
Cookies	Yes	Yes
Save on Server	Yes	No

An Example

The author used ASP to create a web site[9] to do preliminary sizing of a municipal wastewater plant. The processes included are: bar screens, grit removal, primary sedimentation, high-rate activated sludge, packed tower, final sedimentation, and sludge production. The intent of this web is to allow students to quickly check if their homework or design calculations are reasonable. It also affords the student the opportunity to gain a better understanding of the size of various processes for differing wastewater flows. Since the web uses ASP for the calculations, the students are not afforded the opportunity to see exactly how the sizing and decisions were performed. They are presented with only the information the instructor wishes them to have available.

To use the site, the user inputs the wastewater average and peak flows (million gallons per day or MGD) and influent BOD_5 (5 day biochemical oxygen demand as mg/L) and TSS (total suspended solids as mg/L). Other parameters for activated sludge or trickling filter design have preset or default values that may be changed by the user. Any input parameter, whether required or optional is checked for validity. If the value is too large or too small, an error statement is returned to the user with a reminder of the allowable limits. To summarize, the input is the incoming wastewater strength and flow.

The web accepts the submitted input and writes a cookie containing all the data on the users machine. The cookie allows the user to return to the site at a later time (up to 90 days in this case) and go directly to the design section without reentering the data. After data submission, the input data is presented back to the user for confirmation or correction. The user may then select from several pages which provide preliminary sizing for different plant processes. The returned design data includes tank sizes (length, width, and depth or diameter and depth); the number of units required; channel widths, horsepower or air flow requirements, typical design standards, etc. The actual design is compared to the typical design standards for that particular process. Finally, a plan view of the process with piping is presented.

Shown below is the output page for primary sedimentation.

Primary Clarifier Design Output

Input Data Change Input Data

Parameter	Value
Average Plant Flow (million gallons/day)	12 MGD
Peak Plant Flow (million gallons/day)	22 MGD

Calculated Criteria

Parameter	Value
Peaking Factor	1.83
Minimum Area	10000 ft^2
Maximum Area	15000 ft^2

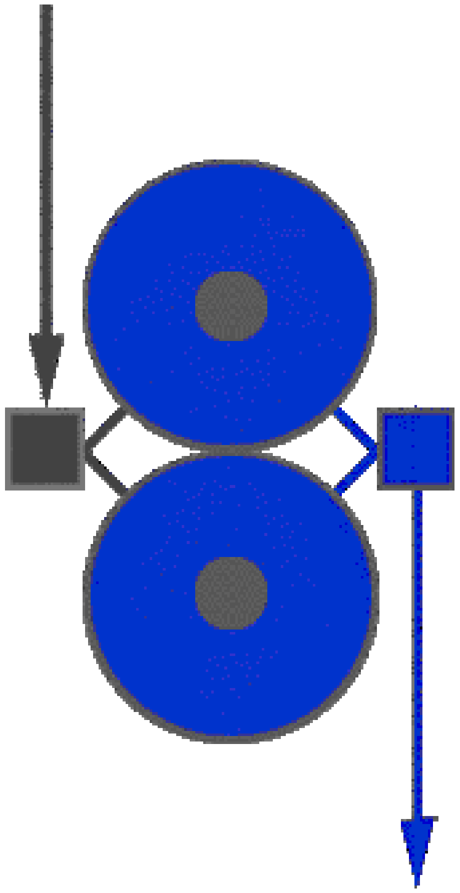
Minimum Unit Sizing Based on Average Flow

Parameter	Value
Number of Units	2
Diameter	80 ft.
Depth	14 ft.
Actual Area	10053 ft.^{2}
Actual Overflow Rate at Average Flow	1194 Gal./ft.^{2}-day
Actual Overflow Rate at Peak Flow	2188 Gal./ft.^{2}-day
Actual Detention Time at Average Flow	2.1 hours

Standard loading criteria used for these calculations:

Parameter	Value
Overflow rate at average flow	800 to 1200 Gal./ft.^{2}-day
Overflow rate at peak flow	2000 to 3000 Gal./ft.^{2}-day
Detention time at average flow	1.5 to 2.5 hours

Possible Schematic (Incoming wastewater is gray, outgoing wastewater is blue. The primary clarifiers are the two circular units below.)



Note: If the peaking factor is 2.5 or less, the size is based on accepted overflow rates for average flow. If the peaking factor exceeds 2.5 the size is based on accepted overflow rates for peak flow.

++++++++++++++++++ End of Output ++

The reader is referred to the following URL for a demonstration: http://civil.engr.siu.edu/Ray_Design/.

The previous example is intended only to demonstrate that a web server in combination with a scripting language can be used by faculty to provide design calculations for engineering courses. The web based program accepts the input, performs the calculations, and returns the results based on the structure of the program. It acts no differently than any other computer application, except that it is available to anyone accessing the web. A wastewater treatment example was chosen because the author developed the web for courses related to wastewater treatment. Internally, the program calculates the required minimum area for both the average and peak flows. These areas are based on accepted industry design standards. The diameter is then calculated and rounded up to the nearest foot. The area is recalculated and the result is compared to typical design values. The program then determines the number of units required based on typical maximum diameters. The depth is then calculated based on accepted detention times. The mathematics for this example is relatively simple. Much of the program is working with decisions about sizes and numbers of units. However, both scripting languages mentioned earlier offer a rather complete set of math functions.

Course Applications

Students in courses having a wastewater treatment component (our program has two, Wastewater Design and Water and Wastewater Treatment) use the design web to confirm their homework and design calculations. They also use the web to gain an understanding of how flow rates affect the size and number of units in a given treatment process. Feedback from students using the web has been positive.

Conclusions

1. Scripting languages such as JavaScript and VBScript are relatively easy to grasp for engineering faculty already possessing a knowledge of other computer languages such as FORTRAN, BASIC, or C++.
2. Scripting languages offer many features of more complex languages, including: decision branching; looping; and numerical calculations.
3. Scripting languages can manipulate graphic images.
4. Scripting languages offer faculty the ability to develop interactive web applications for a variety of engineering problems.
5. Interactive course webs offer students access anywhere they can connect to the web.

Bibliography

1. Francis, Brian, et. al. *Beginning Active Server Pages 2.0*, Wrox Press Ltd., UK, 1998. (*Beginning Active Server Pages 3.0* should be available January 2000.)
2. Lomax, Paul and Petrusha, Ronald, *Learning VBScript*, O'Reilly & Associates, 1977.
3. Brophy, Keith, and Koets, Timothy, *Teach Yourself VBScript in 21 Days*, Sams Publishing, 1996.
4. Federov, Alex, et. al. *Professional Active Server Pages 2.0*, Wrox Press Ltd., UK, 1998. (*Beginning Active Server Pages 3.0* should be available January 2000.)
5. Flanagan, David, *JavaScript, The Definitive Guide, 3rd Ed.*, O'Reilly & Associates, Inc., 1998.
6. Manger, Jason J., *JavaScript Essentials*, Osborne McGraw-Hill, 1996.
7. McFarlane, Nigel, *Instant JavaScript*, Wrox Press Ltd., 1997.
8. Moncur, Michael, *Teach Yourself JavaScript 1.3 in 24 Hours*, Sams Publishing, 1999.
9. URL: http://civil.engr.siu.edu/Ray_Design/;

BILL T. RAY
Bill T. Ray is an Associate Professor of Civil Engineering at Southern Illinois University Carbondale. Dr. Ray is a registered professional engineer in Illinois. He teaches courses in environmental engineering and is also active in developing and maintaining the Department's computer network and web servers.