

Tips for Teaching Web Programming

Vance Etnyre

Information Systems, University of Houston – Clear Lake
Houston, Texas, 77059
etnyre@cl.uh.edu

and

Kent Cook

Information Systems, University of Houston – Clear Lake
Houston, Texas, 77059
kent.cook@earthlink.net

Abstract

This paper presents three tips for teachers dealing with web programming. The three tips focus on security and deployment issues in web programming. The first tip deals with configuration conflicts and security vulnerabilities which occur when you try to install web servers for Java-based web applications and Microsoft-based web applications on the same physical server. The second tip deals with overcoming deployment problems for large Microsoft-based web applications developed on Visual Studio.NET. The third tip deals with overcoming deployment problems for smaller Microsoft-based web applications when the solution given in the second tip is not available.

Keywords: Web programming, security, deployment strategies, teaching

Web programming is an extremely popular field among information technology students today. Students want to learn how to build interactive, easily maintainable websites that can deliver accurate and timely information. Teaching students how to create these highly sophisticated websites is both interesting and challenging.

There are many sets of tools available to the web developer. The easiest tools to use are packaged development products. These tools provide graphic user interfaces with drag-and-drop capabilities, alleviating the user of the necessity to write code, in many instances. Within minutes, technically astute developers can use these software products to create working website models. At some point, however, these packaged development tools present the developer with a set of limitations. Web programming allows the developer to reach beyond these boundaries to provide a higher degree of flexibility and interactivity to the website.

The most important criteria for web development are service effectiveness, security, ease of use and cost effectiveness. A web site is service effective if it provides the desired services to the intended audience in a meaningful way. A web site is secure if it consistently and reliably provides the intended services and cannot be used to provide unintended access to the source

code, other systems or facilities on the server or provider network. A website is cost effective if the benefits it provides are greater than the cost of building and maintaining the site. Other aspects such as attractiveness, portability and scalability are important after the primary concerns of service effectiveness, security, ease of use and cost effectiveness have been satisfied.

During the analysis phase of a website project, the website developer must determine the content and level of interactivity required by the users. This will help to determine the optimal technology to be used in building the site. If the site only provides static information, select options from a predetermined set, and provides links to other existing sites; packaged development tools will likely suffice, since little or no web programming is necessary. Most website development packages have drag-and-drop capability which allows the developer to add such features as buttons, menus, text boxes and drop-down lists to the website without writing a line of true programming code. However, if the site requires support for database access, complex logic or sophisticated calculations, web programming must be considered.

However, as the web has matured, the demand for 'real-time' information has increased at a dramatic pace. Now, as the content requirements usually include regular information and data updates, the static content model has become woefully inadequate. For instance, to avoid disappointing potential customers with an out-of-stock message, an online retailer may wish to offer only those products that are available in inventory on its website. To keep up with the constant update requirements, the retailer would either have to employ a team of people to keep the site updated; or the website may be designed to update automatically from the retailer's inventory information system. The potentially higher cost of initial development costs are offset by lower on-going maintenance costs and higher customer satisfaction when orders made are promptly filled.

As indicated, initial development costs are likely to be higher since highly qualified programmers will be needed to create the 'middleware' applications needed to extract the necessary data for presentation on the website. Since the retailer's inventory system is being employed, security issues are also raised and must be countered, since these database connections can make the retailer's inventory system vulnerable to unauthorized access.

Fortunately, over the last decade, two methods of web-based programming have evolved. Initially, Java, a programming language originally developed by Sun Microsystems, evolved into an early standard for web-based programming languages. Other software providers, including IBM, endorsed and enhanced Java as a web programming standard.

Microsoft followed a different path for web development by introducing Active Server Pages (ASP). ASP used a combination of server-side code written in Visual Basic script, embedded the JavaScript and HTML within a client-side form. Since that initial effort, Microsoft has introduced its .NET initiative which adds substantially greater functionality to its earlier ASP technology. The new initiative is designated ASP.NET.

ASP.NET is Microsoft's new technology for web-programming applications. ASP.NET offers many significant advantages over previous ASP applications in which server-side code was embedded within HTML files. ASP.NET now includes fully functional object-oriented

programming languages, Visual Basic, C#, and J#, which interact with web-based forms to provide very powerful active server pages.

One welcome feature of ASP.NET is the ability to separate server-side code from client-side presentation HTML and JavaScript. This new web programming technology provides two approaches. One method, called 'code-behind' is automatically provided by Microsoft's Visual Studio.NET development platform. This approach separates the definition of the client-side form and the server-side program code into two different files with different filename extensions.

The other approach, called single file 'Form Plus Script Section' (FPSS), separates the definition of the form and the server-side code into separate sections of the same file. Of the two approaches (code-behind and FPSS) code-behind method provides the greatest level of security.

Those who teach web programming should include lessons on security and deployment as part of their courses. One of the requirements associated with web programming is the need to provide sufficient security for the website being developed. In simpler applications where the web developer stays within the boundaries of a packaged development tool, security provisions provided by that tool are usually sufficient to protect the site from most threats. However, when a developer decides to use a general programming language to extend the functionality of the site to include other resources, such as the customer's inventory information system, the developer must take on additional responsibilities for providing site security.

Code-behind provides two different levels or layers of security to a website. The first is where the website creation team is divided into different groups, such as design and development. The design group may include personnel who are not familiar with programming languages and could unintentionally modify the server-side code. Since code-behind provides a physical separation between the presentation web form and the ASP programming code, this unfortunate scenario can easily be avoided by only providing the design team with the more familiar HTML tags (somewhat modified for use with ASP.NET) in a separate file.

The second type of security code-behind offers is the ability to compile the ASP source code into a machine code, optimized for the server. The compiled source code is stored separately and is not directly accessible by the web user. By using this method, the developer is able to prevent the source code from being modified by unauthorized personnel, and to better protect his intellectual rights, when applicable.

Additionally, the new object-oriented programming languages, such as Java, C#, and the latest version of Visual Basic all provide classes of objects which were specifically designed to enhance website security, such as encoding and decoding. Other "good practices" in coding procedures can create additional layers of security, such as creating custom error-handling pages to avoid inadvertently revealing sensitive information to the website user.

However, security problems are further compounded and potential solutions may be unavailable, when the website is deployed on a server other than the one used to develop the application (from the development server to the production server). Security-based objects and practices must be combined with sound deployment practices in order to provide a secure operational site.

Although there are general security and deployment concepts which can be applied to various platforms, the implementation of these general concepts varies from platform to platform. Many of the textbooks used to teach Java programming recommend using Apache Tomcat on the host server for executing servlets and .jsp files. Textbooks used to teach C# or Visual Basic.Net programming recommend using Microsoft Internet Information Server (IIS) on the host server for executing Active Server Pages. Many system administrators say that running Microsoft IIS and Apache Tomcat services on the same physical server creates severe configuration conflicts and introduces additional security problems.

Tip #1: You can run all web programs (Java and Microsoft-based) on the same server.

The Problem:

Microsoft ASP.NET applications will not run on servers running non-Microsoft operating systems. Installing and running most pure Java-based services such as Apache Tomcat on the same machine with IIS introduces configuration and resource-demand conflicts, as well as opens security vulnerabilities.

Possible Solution Approaches:

Applications using Java programming can be run in two ways on Microsoft servers. First (and somewhat superficially), Java code can be compiled as J#, which will compile into Microsoft Intermediate Language (MSIL) and run under IIS. This, however, will elicit howls of indignation from pure Java advocates.

Another method is to keep the Java servlets and .jsp files intact as pure Java code. These files can then be deployed to a server program which accepts and compiles pure Java files into MSIL which will run on the Microsoft IIS server. One such server program is ServletExec which is distributed by NewAtlanta.com. This program allows .jsp files and Java Servlets to run under IIS on servers which also run Microsoft's web-based applications.

Tip # 2. Use Visual Studio's 'Copy Project' option with FrontPage Server Extensions to deploy most ASP.NET applications

The Problem:

By default, Visual Studio.NET produces multi-file application projects for web-based programming. In addition to the several files used to compile the application and link it to other resources, Visual Studio.NET produces separate files for the web form (with an .aspx extension) and the server-side code (with a .vb or .cs extension). The many files produced by Visual Studio must be properly linked together to produce the intended result. Visual Studio does this automatically during the development process. When the application must be deployed to another server, properly linking these files can be a daunting problem.

Solution:

Visual Studio.NET provides a 'Copy Project' service as an option under its 'Project' menu options. This method utilizes the full code-behind benefits, to include compiling the source code. The developer need only supply the URL of the target website on the server and designate

'FrontPage Extensions' as the method to implement the 'Copy Project' service. Visual Studio will create all of the necessary script commands to copy the necessary files and properly link them together on the server. Without this very useful service, deployment can be quite difficult.

Tip # 3. If FrontPage Extensions are not available on the server, create small, easily deployable (one file) ASP.NET applications.

The Problem:

Visual Studio.NET is a very easy platform to use for developing web-based applications. If FrontPage extensions are not available on the server, however, the dozen or so files produced by Visual Studio.NET create several possible deployment problems.

A Solution:

Microsoft distributes (free of charge) another development package called Web Matrix. This development package provides an integrated development environment similar to that of Visual Studio. The output product of Web Matrix, however, is a single 'File Plus Script Section' application file with a .aspx extension. For simple web programming applications, the single output file is self-contained, with no links to other files. The single .aspx file can easily be copied to a server running IIS, where it executes as easily as it executes on the development environment. However, in return for this ease of deployment, as stated earlier, FPSS does not provide the level of security provided by a separate code-behind file.

Conclusions

Web programming is an exciting, rapidly evolving field. The authors have tried to point out areas where challenges occur in teaching this exciting field. In three specific areas, the authors have offered suggestions to solve problems which they have encountered. The authors hope that readers will find one or more of these suggestions useful in teaching web programming.

References

Deitel, H.M., and Deitel, P.J. (2004), *Java: How to Program*,

Upper Saddle River, NJ: Prentice Hall

Deitel, H.M., Deitel, P.J., and Nietro, T, (2002), *Visual Basic.NET: How to Program*,

Upper Saddle River, NJ: Prentice Hall

Deitel, H.M., Deitel, P.J. and Steinbuhler, K. (2001), *E-Business & e-Commerce for Managers*,

7 th Edition, Upper Saddle River, NJ: Prentice Hall

Jessup, L. and Valacich, J (2003) *Information Systems Today*

Upper Saddle River, NJ: Prentice Hall

Laudon, Kenneth and Laudon, Jane (2002), *Management Information Systems*, 7 th Edition,

Upper Saddle River, NJ: Prentice Hall

Webb, J. (2003), *Developing Web Applications: MCAD/MCSD Training Kit*,

Richmond, WA: Microsoft Press