Which Web Development Tool Is Right for You?

Dennis R. Copeland, Raymond C. Corbo, Susan A. Falkenthal, James L. Fisher, and Mark N. Sandler

This article might not settle dogmatic developer arguments, but it offers objectivity and technical detail to help you assess the strengths, weaknesses, and appropriateness of some popular Web development tools. Selecting the right tool for Web development projects is becoming more important as enterprises move from static Web sites to more sophisticated interactive, secure, database-backed Web sites. The need to develop and deploy new Web applications in Internet time feeds the need to select the right tool for the project.

In writing this article, each of us, as Web developers, began by picking our favorite tool and presenting its strengths and weaknesses. The tools we presented were

- Dreamweaver
- Active Server Pages (ASP)
- Domino
- Practical Extraction and Report Language (Perl)

Then we challenged one another’s lists of strengths and weaknesses. The sessions were sometimes passionate, but we managed to keep the discussions collegial and objective. This article reflects the best from those sessions.

**THE BROAD SCOPE OF WEB DEVELOPMENT**

Web sites range from static, personal, and corporate “Web presence” sites to highly interactive, secure, e-commerce or customer relationship management sites with high transaction volumes. High-end sites include Amazon.com, Microsoft.com, eBay.com, and countless others, handling millions of hits, fetching gigabytes from their databases, and processing thousands of credit card transactions per day.

We have categorized Web development functions into five areas:

- preparing page structure,
- organizing and managing content hierarchy,
- serving content to user (fetch and send information to the user’s browser),
- catching user input, and
- performing back-end processing and integration.

The tools we discuss here are some of the most popular in Web site development today. As Figure 1 shows, each selected tool provides different coverage of these functions, and each has a different learning curve. Some of the tools are multifunctional; others are more limited in function. Table 1 summarizes the pros and cons of each Web development tool.

**DREAMWEAVER**

On 8 November 1999, Macromedia announced version 3.0 of its Web-authoring tool, Dreamweaver. This product has grown in popularity with professional Web developers since its first release in late 1997. Dreamweaver 3.0 is a powerful Web-authoring tool, yet it is easy to use, even for first-
time users. The ability to replicate a remote Web site locally greatly simplifies using Dreamweaver with existing Web sites.

Available separately, Dreamweaver can also be purchased bundled with Macromedia’s graphics software package, Fireworks 3.0. Dreamweaver’s competitors include Adobe GoLive and Microsoft FrontPage.

Dreamweaver is a professional visual editor for creating and managing Web pages. Web developers can visually design and manage cross-browser Web sites without sacrificing Hypertext Markup Language (HTML) control. It also helps developers streamline workflow among development team members and supports graphics- and content-supportive applications such as Photoshop, Fireworks, and Microsoft Office. Dreamweaver also lets Web developers customize the environment and automate tasks.

Like other tools in this class, Dreamweaver’s value is in automating the slow process of building a Web site by hand with a user-friendly point-and-click, drag-and-drop environment. By generating the HTML source that corresponds to what the developer creates visually, Dreamweaver eliminates the need for knowing HTML. The HTML inspector feature gives the developer total visibility and control of the generated HTML source. Although other HTML generation tools we’ve used always required manual corrections, we were pleased with Dreamweaver’s output.

Our consensus is that using a conversion tool rather than

---

**Figure 1. Functional coverage versus skill required for four popular Web development tools.**

**Table 1. Functional comparison of four Web development tools.**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Functions</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dreamweaver</td>
<td>Prepare, Organize</td>
<td>HTML knowledge not required</td>
<td>Server required for back-end process development and testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No server needed for testing</td>
<td>Additional tools required for processing functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allows collaborative development</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FTP capability</td>
<td></td>
</tr>
<tr>
<td>ASP</td>
<td>Catch input, Perform back-end processing</td>
<td>Compatible with many object-oriented components</td>
<td>Runs only on Microsoft platform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supports database connectivity</td>
<td>Requires experienced developer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Good integration with IIS</td>
<td></td>
</tr>
<tr>
<td>Domino</td>
<td>Prepare, Organize, Serve content, Catch input, Perform back-end processing</td>
<td>Many built-in collaborative functions, Sophisticated access controls, Good design tools and templates, Built-in data and document management, Sophisticated indexer and search engine</td>
<td>Lacks relational database, Limited capability for high-volume, high-performance applications, High learning curve</td>
</tr>
<tr>
<td>Perl</td>
<td>Catch input, Perform back-end processing</td>
<td>Excellent code portability, Runs on desktops and servers, Free Win32 extensions available, Open-source nature brings constant functionality growth, Good integration with the ActiveX engine</td>
<td>Requires experienced programmers, Potentially high maintenance and evolution costs in dynamic applications</td>
</tr>
</tbody>
</table>

---
Word’s Save-as-HTML feature is better for converting Microsoft Word documents to HTML. Nevertheless, Dreamweaver’s Clean-Up-Word-HTML feature is very useful for those using Word to generate content. Dreamweaver strips out unnecessary tags, such as Word-inserted metatags, browser-specific items, and various other Word artifacts, and returns optimized code that maintains the original Word material’s visual appearance (see Figure 2).

A notion of Dreamweaver’s most popular features is Roundtrip HTML. This feature describes Dreamweaver’s ability to read documents that move between Dreamweaver and a text-based HTML editor, with little or no impact on the content and structure of the document’s HTML source code. Roundtrip HTML can ignore almost any kind of server markup. It can also identify syntactically invalid HTML code created in an outside text-based HTML editor.

Dreamweaver strengths

Dreamweaver provides tools for developing browser-neutral Web sites graphically and quickly through point-and-click, drag-and-drop operations, without requiring manual HTML coding. Moreover, Dreamweaver establishes a local site as a storage location for all documents and files belonging to the Web site project. This function lets Web developers begin developing and organizing a Web site’s directory structure from their desktop or other shared space. This, in turn, alleviates the need to use Web server storage for development purposes. Finally, Dreamweaver can establish an FTP connection to a remote Web site through which developers can push and pull files from a local system.

Dreamweaver weaknesses

Using Dreamweaver locally to build a Web site is convenient. However, you must still move to the Web server to test and run server-side processes—an important part of an interactive Web site that collects user-entered data.

Also, Dreamweaver is a powerful Web development tool. If you need to edit or modify a few existing Web pages, using Dreamweaver may be overkill.

ACTIVE SERVER PAGES

ASP is a server-side scripting environment for creating dynamic Web pages or building other interactive Web applications. ASP pages are files that contain HTML tags, text, and script commands. They can call ActiveX components to perform tasks such as communicating with a database or performing a calculation. ASP lets developers add interactive content to Web pages or build entire Web applications that use HTML pages as the user interface.

ASP scripts give HTML authors an easy way to begin creating interactive pages. ASP provides a relatively simple mechanism for:

- collecting information from an HTML form,
- personalizing an HTML document with a customer’s name, or
- using browser-specific HTML features.

If you wanted to collect information from an HTML form, you would typically use a programming language to build a common gateway interface (CGI) application. A SP lets you collect and analyze data from a form by using simple instructions embedded directly into HTML documents. So you need not learn or use a full programming language or compile separate executables to create interactive pages.

Developers who already know a scripting language, such as Microsoft Visual Basic Scripting Edition (VBScript), JavaScript, or Perl, know how to use ActiveX components to perform tasks such as communicating with a database or performing a calculation. A SP lets developers add interactive content to Web pages or build entire Web applications that use HTML pages as the user interface.

A SP scripts give HTML authors an easy way to begin creating interactive pages. A SP provides a relatively simple mechanism for:

- collecting information from an HTML form,
- personalizing an HTML document with a customer’s name, or
- using browser-specific HTML features.

If you wanted to collect information from an HTML form, you would typically use a programming language to build a common gateway interface (CGI) application. A SP lets you collect and analyze data from a form by using simple instructions embedded directly into HTML documents. So you need not learn or use a full programming language or compile separate executables to create interactive pages.

Developers who already know a scripting language, such as Microsoft Visual Basic Scripting Edition (VBScript), JavaScript, or Perl, know how to use ActiveX components to perform tasks such as communicating with a database or performing a calculation. A SP lets developers add interactive content to Web pages or build entire Web applications that use HTML pages as the user interface.

ActiveX scripts give HTML authors an easy way to begin creating interactive pages. A SP provides a relatively simple mechanism for:

- collecting information from an HTML form,
- personalizing an HTML document with a customer’s name, or
- using browser-specific HTML features.
such as Visual Basic will find that ASP provides a flexible way to quickly create Web applications. By adding script commands to HTML pages, developers can create an HTML interface to an application. By creating ActiveX components, developers can encapsulate an application’s business logic into reusable modules that they can call from a script, another component, or another program.

**The ASP model**

An ASP script begins to run when a browser requests an .asp file from the Web server. The Web server then calls ASP, which reads the requested file from top to bottom, executes any script commands, and sends the resulting Web page to the browser.

Because the scripts run on the server rather than the client, the server does all the work involved in generating the Web pages sent to the requesting browsers. So the developer need not be concerned about whether the requesting browser can process the scripts. Furthermore, the user cannot easily copy server-side scripts, because only the script’s result—not the script or processing logic itself—returns to the browser.

ASP lets Web authors intersperse delimited script-code directives in an HTML source file, called a template. This file resides on the server with an extension. When a client’s HTML code references an .asp file, the Internet Information Server (IIS) passes the request to the ASP runtime parser. The parser acts as a template file processor that parses the .asp file and takes the correct actions, as specified by the delimited directives. From within these directives, scripts can evaluate expressions, branch conditionally, and invoke methods on server-side components. The directive’s output funnels to the respective template areas in the resultant HTML stream, which then travels down to the client machine via http.

The nature of this process allows dynamic and specialized HTML creation (see Figure 3).

**ASP strengths**

Perhaps the most important feature of Active Server Pages is that it lets you instantiate and use programmable components. You can create these components in tools such as Visual Basic, Visual C++, Visual J++, Borland Delphi, and Powersoft PowerBuilder. Thus, you can integrate Web applications with existing client-server systems.

**ASP weaknesses**

A SP has two main limitations: It runs only on Microsoft platforms using IIS as a Web application server, and using it requires experience and programming skills.

**DOMINO (RELEASE 5)**

R 5 is Lotus’ latest release of its Web application and back-office server, Domino. In addition to supporting Web applications, it also supports a client-server architecture and development environment based on a typical graphical user interface. Domino R 5 runs on Windows NT, Windows 95 and 98, Solaris, Linux, AIX, and other operating systems.

**Technical description**

Domino provides Web developers and IT managers a one-stop approach to delivering Web applications. The database server is Domino’s foundation. A Domino database stores all of an application’s design elements and data. Domino also provides a full-text indexer component. This component is similar to other Web site index builders that provide the capability to search an entire Web site or database for content via character strings. Domino also has an http server component that delivers all this functionality to the Web. Figure 4 shows a high-level view of the Domino server’s architecture and components. Domino’s http server is essentially a standard http Web
server with an application programming interface that allows calls to a Domino database. You can replace the Domino http server with other http servers, such as Microsoft's IIS or Apache's http server, and still provide full access to a Domino database running on the Domino server. This capability provides a convenient way to integrate Domino into a legacy infrastructure.

The API is where the dynamic conversion takes place. It automatically converts a Domino database’s design, logic, scripting, and information content into HTML and JavaScript. Then it sends this information to the http server. The server, in turn, presents the page to the browser. This conversion occurs on demand in real time. Domino supports many of today’s Internet standards. For example, the Internet Interoperable ORB Protocol enables the deployment of distributed objects based on various object-oriented languages such as Java. Thus, Domino can help you develop distributed applications for the Internet.

**Domino database**

The database is the mechanism that stores everything related to a particular application: HTML Web pages, graphics and image files, data, queries, scripts and code, and background agents. Thus, a single operating system file contains the entire application. This simplifies maintaining the application because the design elements and content are not scattered across the operating environment’s storage devices.

Domino’s database storage techniques differ from those of many other vendors. Developers do not design tables and relationships as with a relational database management system such as Oracle. Instead, the developer designs forms that store information in documents, so that one database may have several different forms that create many different types of documents. For example, the system-user database contains one form for creating individual user accounts and one for creating user groups. Each user or user group can have many different access, editing, and control privileges.

**Domino development tools**

Domino gives Web developers several application development tools. It supports embedded client-side Java and JavaScript in an HTML page. A developer might use client-side scripting for data validation on a form before it goes to the server. For server-side processing, Domino supports Java and LotusScript. For example, a developer can use server-side scripting to process user input. Domino agents, triggered when a document is saved to the database, can help accomplish this server-side scripting. Besides using many Web-development Internet standards, Domino also provides a designer tool. This tool provides a graphical user interface for developing WYSIWYG (what you see is what you get) forms, views, and HTML pages. Domino views are predefined queries that display a filtered subset of database documents. Views give an easy mechanism for organizing users data and documents.

**Domino strengths**

Domino inherits most of its sophisticated functions from the mature Lotus Notes groupware application. Therefore, it is excellent for document management and workflow applications. Moreover, Domino provides its own access control mechanism. Domino uses its own user directory services, so you can avoid giving users access to the operating system. This makes Domino’s security model consistent and adaptable across the many operating systems it runs on. The Domino directory is Lightweight Directory Access Protocol accessible, making integration with other applications easy.

Another Domino advantage is the database engine’s hierarchical nature. This, along with Domino’s access control mechanisms, greatly simplifies implementing robust online bulletin boards, collaborative spaces, and discussion groups. Domino also provides a good prototyping and production-development environment. Lotus gives developers a designer tool to build applications, as well as many sample database and application templates. You can easily customize these templates, yet they are readily usable out of the box.
Finally, the full-text indexer recognizes over 130 file types. In addition to indexing stored HTML pages, the indexer can also index the contents of stored documents. The indexer’s associated search engine provides sophisticated search functions such as sentence or paragraph proximity and weighted searches.

**Domino weaknesses**

Domino is limited by its lack of a true relational-database capability. Domino’s database server is a hierarchical database engine and does not allow the implementation of databases requiring primary and foreign keys. Data is stored either as flat records or hierarchically. In addition, although its functions could provide the applications required for a transaction-based Web application such as an e-commerce site, Domino’s database engine is not particularly suited for high-volume, high-performance relational-database applications. Finally, novice developers may find that learning Domino development is difficult. Usually, senior Domino developers have two to three years of hands-on experience with the Domino architecture and environment, and 10 to 15 days of formal training.

**PERL**

The most popular Web server platforms with their integrated programming environments include Microsoft’s Internet Information Server with Active Server Pages, Netscape’s Enterprise Server with Java, and Apache Software Foundation’s A pache http Server (http://www.apache.org/httpd.html) with Professional Home Page. Although A SP, Netscape’s Enterprise Server, and the Apache http Server have versions for both UNIX and Windows platforms, this discussion of Perl is limited to Web application development on Windows platforms.

Selecting the right tool depends on both the available development environment and the application’s sophistication.
INTERNET DEVELOPMENT

Figure 5. Coding for database access.

Dim objConn, objRec
Set objConn = Server.CreateObject("ADODB.Connection")
Set objRec = Server.CreateObject("ADODB.Recordset")
objConn.Open "DSN=my_mdb_file"
objRec.Open "SELECT * FROM Employees", objConn
(a)

use Win32::ODBC;
$db = new Win32::ODBC("DSN=my_mdb_file");
$db->Sql('SELECT * FROM Employees');
while ($db->FetchRow()) {...}
(b)

(a) In ASP, database calls require the “Server” object and must be done on the server side; (b) in Perl, the ODBC object handles database interactions and can be executed on the server or client side.

Or you can run Perl on the client side by enclosing the code within the similar tags:

<SCRIPT Language=PerlScript>
  ...
  ...
</SCRIPT>

Although client-side VBScript differs slightly from server-side VBScript (or, more accurately, Active Server Pages), Perl functions are the same in both client- and server-side scripts. Programming database access illustrates this point well. With A SP server-side coding, calls to the “Server” object handle database connections and record-set management, as shown in Figure 5a.

For testing and debugging, the developer must have access to a Web server platform (typically an NT 4.0 server box). For example, the developer cannot continue work at home using a Windows 95/98 platform, because the server object is not available on the client side. In contrast, as Figure 5b shows, database connections in Perl are the same and rely on the ODBC object, whether on the client side (for example, at home) or on the server side (for example, on the main Web server). A different error-detection code would be necessary in a truly operational system. Thus, PerlScript provides the basis for developing very powerful Web functions with minimal development environment tools.

Processing input with Perl

Perl also provides other standard Web server functions. The CGI.pm module makes catching user input from a submitted form easy. CGI.pm also provides hooks for catching uploaded files. Those who have tried to implement this functionality in ASP or have searched the Web for a free component can appreciate Perl’s power and flexibility. Because Perl has a rich collection of pattern-matching and string-manipulation capabilities, it is an excellent tool for interpreting and transforming user input.

We have used Perl to implement many of our Web services. At a recent online Y2K command center, analysts periodically completed and uploaded Microsoft Word or Excel report templates. The analysts uploaded files through an HTTPS connection (secure connection via HTTP over SSL) and were required to have an NT account with appropriate privileges on the server. The NT operating system authenticated the user, and Perl determined the user’s login name, full name, and group membership. These reports, with file names similar to LAN_summary_v3.xls (where the analysts manually assigned the version number _v3 for local record keeping), were stripped of their version number and augmented with a
time stamp. Then they were placed in a dedicated directory. Perl scripts then dynamically detected the presence of new files and displayed them on a Web page under one file name, with a pull-down menu of all available upload times. This automated procedure completely freed the Web masters from having to manage the file structure and page content.

In another project, researchers captured the contents of highly dynamic, flexibly formatted data into a four-sheet Excel workbook. Perl scripts automated the processes of:

- reading the Excel worksheets,
- checking for internal consistency in key fields,
- requesting changes (if necessary) before proceeding, and
- generating both static Web page content and dynamic JavaScript code.

Using similar scripting techniques, many Web masters have also used Perl for automated accounts generation; remote log analysis; periodic data security checks; and other, otherwise tedious, system-administration and maintenance tasks.

Perl strengths

First, Perl offers excellent code portability across operating systems such as Unix, Linux, Wintel, and WinA Ipa. Second, it allows developers to use any Wintel platform—it does not require a server. Third, Perl has many free Win32 extensions available for additional functionality. Fourth, Perl’s open-source status adds to its stability and frequent addition of new modules. Fifth, Perl provides good integration into the ActiveX scripting engine.

Perl weaknesses

On the downside, Perl requires experienced, cognizant programmers to take advantage of its many methods of interacting with back-end processes. The syntax can be daunting to new programmers. AIso, special installation is required on the client side, if desired. A though Perl programming does not require the same development environment tools as C/C++ programming, it often requires equally talented programmers, thus increasing the average development and maintenance costs.

The Web and its evolving technologies and tools will be part of our environment for the foreseeable future. As you compare these tools to determine which is right for you, keep your options open. You will continue to see an evolution of tools and techniques focused on helping enterprises develop and deploy robust, secure Web-based business applications quickly. Undoubtedly, tool vendors will continue to improve their offerings and attempt to overshadow the competition with better, more user-friendly and efficient integrated development tools.

Dennis R. Copeland leads the Internet Technologies and e-Commerce development team at Mitretek Systems, McLean, Virginia. Contact him at copeland@mitretek.org; http://www.mitretek.org.

Raymond C. Corbo is the manager of an Internet systems group at Mitretek Systems. Contact him at rcorbo@mitretek.org.

Susan A. Falkenthal is a member of Mitretek’s e-commerce group. Contact her at sfalkent@mitretek.org.

James L. Fisher is a principal engineer with Mitretek Systems. He has been running public Web sites since 1992, long before it was fashionable. Contact him at jlf@mitretek.org.

Mark N. Sandler is the associate department manager of Mitretek’s Advanced Software Technologies Department. Contact him at sandman@mitretek.org.

Tools, Check. Process?

You have your favorite Web development tool and you’re on a strict deadline, so now you’re ready to hack away at that new application for your company’s e-commerce site. But before you hit the ground running, have you stopped to plan out the project? Applying engineering principles to Web applications—even those created on Internet time—can help prevent critical software blunders, says Roger S. Pressman in the January/February issue of IEEE Software.

By establishing a solid framework for Web-based applications, the author argues, the new generation of software developers can avoid the kind of haphazard approaches that failed miserably a few generations back. Find out how and why to keep your WebApp development project disciplined in “What a Tangled Web We Weave,” available online at computer.org/software/.