



Part I

Data-Driven Web Pages

In these chapters you'll be introduced to:

- ◆ The data-driven world that web and database developers need to master
- ◆ Dreamweaver MX and its interface
- ◆ Coding practices for successful team development in Dreamweaver MX
- ◆ HTML forms, the basic tool for data input on the web



Chapter 1

An Introduction to the Data-Driven World

THE INTERNET IS CHANGING the way many of us work and play—the ball is rolling so to speak. Although some of us are shoved into the fray by job pressures, most of us are clamoring for more. More what? More information. We want to know everything there is to know, and we want to know it now. As consumers, we'd like it for free, but we're willing to pay for it if the information is vital. As vendors, we're scrambling to keep our customers happy and coming back.

With all this information comes responsibility and choices. Sometimes choices aren't easy, but to remain competitive and successful, you'll have to face them head on. Chances are, if you're reading this book, some decisions have already been made, and now it's up to you to turn your plans into reality. With Dreamweaver MX and this book, you're definitely up to the challenge.

In this chapter, you'll learn about:

- ◆ The benefits of offering dynamic data
- ◆ New technologies and terms
- ◆ Scripting for servers
- ◆ Matching a database to your needs

The Benefits of Offering Dynamic Data

Give web fanatics a static page and they'll demand dynamic data! Well, it's too late. The horse is out of the barn, and everybody thinks they need access to data any time they want it. Frankly, if you have clients to appease, you may have to meet their demands or go the way of the dinosaur. There are a few really good reasons to meet this challenge:

- ◆ You can automate many business-to-business communication tasks.
- ◆ You can simplify many work-related processes at once by creating one single interface for all accessing, updating, and so on.
- ◆ You can give your employees and your clients immediate and remote access to the information they need.

Business-to-Business Communication: Extranets

Suppose you run a small manufacturing plant that supplies customized gadgets for a really large manufacturing plant, and their specifications are now processed online. The whole thing is automated, and specifications are sent to you as soon as they're entered and approved. Are you going to insist that someone on the other side print out a spec sheet and fax it to you? If you want to keep this customer, you're going to climb on the technological assembly line by training someone to access the necessary data electronically. More than likely, someone in your organization will have to spend some time at the client's facilities, learning their system. Once the client gives your employee the necessary clearance to connect to their database, your employee can then sit at her own computer and download order information as required. What you do with the specifications once you've got them is entirely up to you, but at least you've met the client's needs.

The above scenario benefits from what's known as an extranet. That's a website for customers (or those needing limited access) rather than the general public that transmits across the Internet. Companies use them to gain access to all (or at least most) of their databases. Some use them for research.

Internal Communication: The Company Intranet

Now, let's suppose you have employees entering orders, maintaining accounts receivable and accounts payable, and updating inventory and fulfillment—and it's all going on at different locations all over the country. Do you really want to create an individual application for every department? It would be much more efficient and simpler if everyone were talking to the same database and using the same interface. And if you have personnel who are frequently on the road and need information such as current prices and inventory numbers, remote access to your internal site can be provided.

Such a system might include several different input screens that grant or deny access to any number of database applications. That way, employees can gain access to the data they need from any computer that can connect to the system.

This type of in-house website is known as an intranet. It serves employees, and although intranet pages may link to the Internet, there's no public access to the site. Intranets take advantage of Internet protocols and hypertext links to provide a standard means of sharing data internally.

Communicating with the Outside World: The Internet

As widespread as those two applications have become, by far the most common use of data-driven web technology is to exchange information with the outside world via the Internet. Do you have customers that would benefit from ordering from you directly online, bypassing sales personnel and telephone calls? If so, a data-driven site may be just the tool you need. These are just a few examples of the kinds of data-driven sites that Dreamweaver MX—and the languages and database systems it works with—can help you build. Dreamweaver MX does a great job of helping you build just the right site and the right web application for your purposes. However, we don't want to mislead anyone. For one thing, there is a learning curve. Not only do you need to understand Dreamweaver MX and the language and database system you're using, you also need to design exactly how the site will function. You should also consider some other issues before you open shop:

- ◆ Software, hardware, and support personnel will add to the costs.
- ◆ A database isn't a lockbox; it needs constant management and maintenance.
- ◆ Once you're online, your database is vulnerable to hackers. You'll need a good security plan and software and personnel to implement it.

Dreamweaver MX and this book can help you with the first point, but you'll need to deal with the others on your own. The goal of this book is not to provide "recipes" that take you through the steps of building each type of data-driven site; it is to show you how to use Dreamweaver MX and the related tools to build sites that make data accessible to your users.

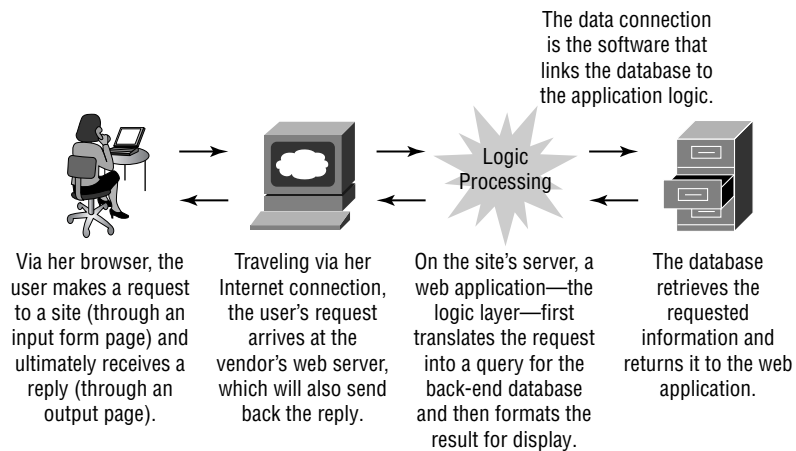
New Technologies and Terms

Technological advances are driven by need, and that evolution is marked by small and large spikes. One of those large spikes was the introduction of HTML (Hypertext Markup Language) because it introduced hypertext, which meant that text information could be accessed in a nonlinear fashion via hyperlinks. HTML was so successful because it was incredibly easy to learn and use and because it worked. But once unleashed, these new capabilities just made users want more—they wanted interactivity and they wanted it to be as easy as HTML. The result is a technological smorgasbord—the only problem may be finding a place in line.

Learning the web technology hierarchy may be the biggest challenge for the nonprogrammer or even the programmer with no web development experience, because there really isn't one—at least not one that's easily laid out in clear terms. The diagram in Figure 1.1 offers a basic look at what happens with a web request; this process is the context in which web development takes place.

FIGURE 1.1

Processing data via
the Internet



The user sits down at her computer keyboard and accesses your website, and information floods her monitor—seemingly with no effort. But there's a lot going on behind that monitor. First, her request is passed along to your web server via her Internet connection; the technology behind this connection is HTTP (Hypertext Transfer Protocol).

Your server interprets the request and sends it along to the appropriate application, which then processes the request. The results are then returned via the web server, which then routes it to her connection and on to her monitor. It's hard to believe all this happens in the few seconds she spent waiting for a request to be processed.

Through a process known as task switching (or multitasking), the web server manages to complete multiple tasks in a short period of time, making each user feel as though she is the only visitor at your

site. A *web server* is the software technology that connects a browser to your database. Here's how it works:

1. As the server receives each request, it stores them in a queue (a list of items that's constantly updated).
2. The server chooses a predetermined number of tasks—known as threads—from the queue.
3. The server loads the first thread and works on it for a while, maybe for a few millionths of a second.
4. The server then loads the second thread and works on it for a while. The server continues in this fashion until the first thread is complete.
5. Once a thread is complete, it's removed from the queue.

The more threads you have, the slower each task is completed, but the loss in performance is hardly noticeable in most systems.

This process sees a request and the resulting processed data traveling across several layers of technology:

Browser Accepts requests from the user and displays data. Microsoft Internet Explorer and Netscape Navigator are the most popular browsers today. You'll often see the browser referred to as the client in web development documentation.

Internet connection Most of us connect to the Internet via HTTP connections.

Web server The software technology that allows you to connect to another system via your HTTP connection and then share data. Windows systems use Internet Information Services (IIS) and Personal Web Server (PWS). The most popular non-Windows server is probably Apache.

Logic layer An application server or a scripting language that processes both client- and server-side requests in regard to manipulating, formatting, and displaying data. This layer works in both directions—accepting requests from the client to the server for data and returning data from the server to the client's browser. The web server and application server can be one and the same. For example IIS 5, Microsoft's web server, is also considered an application server and is referred to as such by Microsoft.

Data connection This layer links the logic layer to the actual data. Connection solutions are specific to the operating system or, sometimes, to the actual database software.

Database This layer could comprise any number of database application files. It's simply the storage medium for your data.

Chances are you've visited a data-driven site. Most any time you request information and the browser displays the requested information, you're dealing with a *data-driven site*. What that means is that a database is actively providing information. Your web browser requests the information via a connection, the server passes that request on to the database, which then processes the request and returns it to your browser, via the server and the connection. Dreamweaver MX can help you build a fully dynamic website, even if you don't have the programming expertise to create such a site on your own.

Scripting for Servers

Dreamweaver MX is an HTML authoring tool and an application development tool that interprets multiple scripting languages to build data-driven websites. It runs on Windows and Mac systems and interacts with the majority of development operating systems. Dreamweaver MX supports most scripting languages, application servers, and database formats. In addition, Dreamweaver MX provides an easy-to-use graphical interface that integrates all these components.

Throughout this book, you'll see the term *script* in many forms. In a nutshell, a *script* is a set of statements pertaining to a particular scripting language. In this context, a scripting language is similar to any other language in that it's a combination of keywords, functions, and methods. However, scripting languages are considered *interpreted* languages. That means the server parses, compiles, and then executes the script all at once and as needed.

An interpreted language differs from other programming languages, otherwise known as compiled languages, in one respect. A compiled language can parse and compile code before the code is executed. As a result, the compiled language stores an already translated version of the code.

In this book, we'll use a variety of scripting languages in our examples. You can learn more about the following scripting languages in Chapters 12 through 16.

Active Server Pages (ASP)

Eventually, the quest for more flexible web languages led to several scripting languages. ASP is a server-side, language-independent scripting host from Microsoft. That means ASP interprets most scripting languages. In fact, ASP works with any scripting language that has a certified ActiveX Scripting Engine. The one requirement is that the language be compatible with the Microsoft Scripting Host. The following scripting languages are compatible with the Scripting Host:

VBScript A subset of Visual Basic for Applications (VBA). We mention this one first because it's the default ASP scripting language.

JScript Microsoft's counterpart to JavaScript. Neither is connected to Sun's Java language. Many developers consider JScript the common standard for browser scripting. JScript comes with ASP.

PerlScript A subset of Perl, which has been around for a long time. It's used mostly with text. It doesn't come with ASP, but you can download it from www.activestate.com.

ASP.NET

Active Server Pages has been the Microsoft web staple for many years and has been supported in all prior versions of Dreamweaver MX (Drumbeat and Ultradev). ASP.NET is a complete rewrite of ASP for the .NET framework and provides many advantages over standard ASP. ASP.NET supports new providers when working with databases, particularly with Microsoft SQL Server 2000. Unlike standard ASP, which restricted the developer to working in VBScript or JScript, ASP.NET allows you to choose virtually any scripting language.

Don't confuse ASP (discussed in the previous section) and ASP.NET, because they aren't the same thing. However, ASP.NET is compatible with ASP to the extent that almost all ASP code is supported in ASP.NET (for now) for backward compatibility.

Hypertext Preprocessor (PHP)

PHP is another server-side scripting language that puts dynamic capabilities into the hands of the masses. PHP runs on Linux, many Unix variants (including HP-UX, Solaris, and OpenBSD), Microsoft Windows, Mac OS X, and RISC OS. Supported web servers include Apache, IIS, PWS, Netscape, and iPlanet. For the most part, you'll find PHP running on Apache servers and interacting with MySQL (a database), although PHP supports a number of databases, including dBase, IBM DB2, Oracle, and Sybase. Currently, PHP is an open-source language that can be freely distributed. You can download PHP at www.php.net.

In a nutshell, you insert PHP code inside your HTML code. When a client makes a request, your server executes the PHP code, just like any other scripting language. PHP's many capabilities include the following:

- ◆ PHP supports data-driven applications and supports cookies (or we wouldn't include it in this discussion).
- ◆ PHP authenticates and tracks users.
- ◆ PHP supports threaded discussions at your site.
- ◆ PHP is scalable across multiplatforms.
- ◆ PHP supports Extensible Markup Language (XML).
- ◆ PHP supports command-line scripting, which is ideal for scripts regularly executed by an operating system.

JavaServer Pages (JSP)

JavaServer Pages is an application server that uses XML-like tags and Java-based scripts to generate web pages. The technology is platform-independent and consists of Java server-side modules, known as servlets, that support and extend the web server. JSP is available for free from Sun Microsystems at <http://java.sun.com/products/jsp/download.html>.

JSP pages consist of three elements:

- ◆ Static components—HTML or XML
- ◆ JSP tags (unique to JSP)
- ◆ Java code, known as scriptlets to the Java developer

Anyone knowing HTML or XML should pick up JSP quickly. In fact, part of its appeal—and hence its growing market share—is its simple language. You don't have to be an experienced web developer to successfully use JSP.

If you're a Sun developer or if you're already familiar with Java, the jump to JSP will be painless. On the other hand, if you're coming from a Microsoft background, you'll find JSP very different from ASP. The main differences boil down to differences in shop technology—you're Sun or you're Microsoft. You can use JSP with Windows and IIS, but you can't use ASP with Sun.

The JSP engine is really a specialized servlet, supported by the servlet engine. JSP only deals with text, so you must use servlets to communicate with Java applets and applications. As a result of this arrangement, you may find servlets better suited to nondynamic tasks such as authentication, validation, and so on. Rely on JSP pages for your data-driven content.

Unlike most of the scripting languages we've reviewed, JSP code is stored separately from the static presentation within external JavaBeans components. The JSP page uses special tags to call these components as they're needed. When a change is made to the presentation template, the JSP engine automatically recompiles and reloads the JSP page.

JSP consists of three components: directives, scripts, and actions. Directives tell the JSP engine what to do with the JSP page. These directives are enclosed with the `%@...%` tag. Fragments of Java code, known as scriptlets, are enclosed in the `<%...%>` tag. Actions perform tasks such as instantiating objects and communicating with the server.

ColdFusion MX

Macromedia's ColdFusion MX is a web application server that lets developers create scripts for controlling data integration, logic, and user interface components for a website. In other words, ColdFusion MX processes the logic and scripting in your page and then lets your web server build the HTML to hand back to your user's browser.

NOTE *An application server is a server that supports your web server. When your browser requests a ColdFusion MX page, your web server passes that request on to the ColdFusion MX application server, which then processes the request and hands it back to the web server for delivery to your browser.*

Using ColdFusion MX, you can quickly produce dynamic web pages without actually writing your own scripts—which can be tedious work even for the experienced web developer. This makes ColdFusion MX or any similar product the ideal solution for the nonprogrammer.

ColdFusion MX quickly produces HTML-like tags via its many wizards and auto-generating tags. For instance, you can combine ColdFusion MX tags to automatically generate a validation script (which you'd otherwise have to write yourself). The automatically generated script is ColdFusion Markup Language (CFML)—a tag-based server scripting language like HTML. You can download a demo copy of ColdFusion MX from www.macromedia.com.

Match a Database to Your Needs

Dreamweaver MX can connect to almost any database that runs on Unix or Windows. If an ODBC (Open Database Connectivity) or OLE DB driver exists, Dreamweaver MX will produce the code necessary to connect to and interact with the database. In this book, we'll review two of the largest full-system databases on the market: Microsoft SQL Server 2000 and Oracle9i. We'll also review the popular desktop database application, Microsoft Access, and the open-source relational database MySQL.

More than likely, as the application developer, you really had no choice over the database you'll be using. Your company chose it for a number of reasons: current costs and future needs were probably discussed. However, you probably ended up with the relational database that you already have. We're not endorsing a database, but we will use all the above in our examples throughout this book.

Working with Microsoft SQL Server 2000 and Access 2002

SQL Server 2000 is a powerful database and is fully web-enabled. You'll have no problem interfacing with it via Dreamweaver MX. It's also expensive and requires experienced personnel to administer and maintain the database. If you're a Windows fan and have the money, go with SQL Server 2000.

Access 2002 (also known as Access XP) is an alternative if you're running Windows, have limited resources, but anticipate a small amount of online traffic.

Relying on Open Source: MySQL

MySQL is fast, stable, runs on Unix and Windows, and it's free—do we need to say more? You can download MySQL at www.mysql.com. MySQL makes its money by selling technical support because the down side is that what you gain in fast, stable, and free open-source code, you lose in functionality. MySQL is a great alternative for the money-challenged but ready-to-learn-and-support crowd.

Oracle9i

Oracle is a SQL Server 2000 competitor and has been around for a long time. This industrial-strength database has friends in many places, and Dreamweaver MX has no problem talking to it.

Summary

We've tried to introduce you to the web-driven world, if that's possible in one short chapter. There are advantages and disadvantages, and as with any business decision, you have to weigh your unique needs and resources. But Dreamweaver MX can help you balance any disadvantages, such as the costs and the highly paid personnel that often accompany a web venture. In the end, there's simply nothing to replace experience. The remaining chapters will open the world of databases and show you how to use scripting languages to interact with databases. By the end of the book, you'll know the ins and outs of retrieving and interacting with dynamic data in several database formats.

In the next chapter, you'll launch Dreamweaver MX and have a look around.