

CHAPTER

1

WebObjects and OpenBase

IN THIS CHAPTER:

WebObjects History

How This Book Is Organized

WebObjects: The Big Picture

WebObjects: The Details

Before You Start: Two Cautions



According to Apple, WebObjects 5 for Java is “the flexible, scalable way to develop and deploy high-performance network applications.” If that sounds a little murky, that’s because WebObjects is many things:

- ▶ It’s an application server—software that runs Web sites and mediates between users’ browsers and databases or legacy applications.
- ▶ It’s a rapid application development (RAD) environment geared specifically to developing Web-based solutions.
- ▶ It’s a tool for dynamically creating Web pages and for managing complex media (such as video).

This chapter provides an overview of WebObjects and a roadmap to the book.

WebObjects History

Here is how this book began prior to May 2000: This book provides a complete overview of WebObjects and its related technologies. WebObjects is a back-of-the-shop tool—an application server—that integrates databases and legacy applications with dynamic Web pages. It features a rapid application development environment that makes even the development of traditional, static Web pages remarkably easy. It is a tool for Web site developers and managers.

In bringing together the Web and databases, WebObjects allows for the rapid deployment of e-commerce solutions. The WebObjects user base is an impressive collection of some of the world’s major corporations: Standard & Poor’s, Toyota, AT&T, the BBC, Fleet Bank, and more (including the Apple Store). As a critical component of these companies’ Web presence, WebObjects runs on a variety of platforms, including Windows 2000, Mac OS X Server, HP-UX, and Solaris; in addition, it interacts directly with databases such as Oracle, Sybase, and Informix. It is frequently deployed in complex environments with multiple copies of the software running to manage a heavy load of transactions.

As is the case with such high-end products geared to the corporate market, pricing is determined on a sliding scale based on usage: the number of processors at a given site on which it runs, for example. The price of WebObjects ranges from just over \$1,000 to \$50,000.

Apple changed the rules in May 2000—not just for WebObjects, but for the entire application server market segment. Apple stopped pricing WebObjects on a sliding scale; a uniform price of \$699 was set. (Yes, \$50,000 marked down to \$699.) Apple

also announced the next version of WebObjects—WebObjects 5 for Java. Instead of maintaining separate code for the various platforms, WebObjects now runs as a Java application on any platform that supports Java.

The price change was significant because it signaled a major shift in the use of WebObjects. None of the features changed and no functionality was dropped, but the substantially lower price meant the following:

- ▶ E-commerce functionality and dynamic Web pages were accessible to individuals and small businesses as well as to large corporations. They could be implemented *the same way* they were on the largest sites.
- ▶ The modern development techniques that are part of WebObjects (object-oriented programming and the rapid application development tools) let everyone develop for the Web in the same way. No longer are there two types of development environments: the corporate information technology world with large development budgets and the make-do world of CGI scripts, Perl, and hand-written Hypertext Markup Language (HTML).
- ▶ The WebObjects deployment tools enable sites to be managed on high-speed connections from home offices and small business.

The last time large-scale corporate technology was made available to individual users, small offices, and home offices, a revolution swept through the world. Desktop publishing was made possible by the relatively low-priced laser printers, What You See Is What You Get (WYSIWYG) graphics, and the easy-to-use graphical user interface (GUI) of personal computers. (Not surprisingly, Apple was at the forefront of that advance, too.) You may remember a world of mimeograph machines, carbon paper, and mechanical paste-ups. It was a world in which the look of printed communications clearly separated the “pros” from the hoi polloi. That world is gone forever. High-quality printed communications are within everyone’s reach (although not everyone chooses or is able to achieve the highest standards).

WebObjects (at its new, lower price) promises a similar revolution. Many individuals have tried (and succeeded to varying degrees) to put up e-commerce sites and dynamic pages. It is always a shock to someone with a small Web site to discover the difficulty of implementing e-commerce, shopping carts, or even a welcome page that greets a returning user by name. It *can* be done, and it *is* done. However, many of the solutions are jury-rigged; furthermore, they are often profoundly unscalable. That is, they work (barely) for the intended purpose, but it is hard to change or expand them. WebObjects is not only powerful but also scalable (as witness the many corporate sites that use it). For smaller users, WebObjects is

the beginning of a reliable, flexible, and powerful development and production environment; it replaces other environments that aggressive and innovative small-scale users have pushed to their limits.

How will this change the computing environment? Most likely, the Web equivalents of mimeograph machines and mechanical paste-ups will seem quaint in a few years. Those equivalents are handcrafted HTML pages, scripts written in Perl or other scripting languages, and awkward bridges to and from legacy systems and databases. The rapid application development tools of WebObjects can yield significant productivity improvements to Web site designers and implementers.

In addition, like the desktop publishing revolution, the overall quality of the output (in that case, printed communications, but in this case, Web sites) will probably improve. As the new technology becomes more widely available and accessible to more people, it will be feasible and cost-effective to use it for even the smallest project. (If you don't believe this, look in a file or archive for a church or school newsletter from 20 years ago; you will probably be shocked at the low quality—by today's standards—of the design.)

Finally—and again, just as with desktop publishing—many intermediaries will disappear from the picture or find their roles redefined. This does not mean that everyone can become a Web designer overnight (or ever). What it does mean is that the possibility of experimentation is open to all; it also means that everyone has more control over the work that they do or want done. A complex Web site may still have a large staff devoted to creating and maintaining it; however, less and less of that work will be the drudgery that is the nasty little secret of Web site development today.

Another intermediary whose role may well be changed is the Web site host. The advent of broadband connections (including cable and DSL) and of highly reliable Web servers at low cost means that it is feasible for many individuals and small businesses to run their own sites. In the excitement over the growth of the Web, it is not often noted that most Web sites get very little traffic. Many sites may get only a few hits (if that) over the course of a week. Because of the overhead involved in creating and maintaining a Web site, many smaller sites are unable to be managed on their own; people use “free” pages on commercial services such as AOL instead. Coupled with the broadband connections, the deployment tools of WebObjects can change all of this.

Those are the changes that Apple set in motion. As a result of the lower pricing, a vast market of individuals and small businesses has access to the functionality, development, and deployment tools that heretofore were the sole dominion of large corporations. The consequences of this breakthrough are likely to be significant improvements in productivity in creating and maintaining Web sites, an increase in the overall quality of Web sites (at least those that use WebObjects), and an

increase in efficiency as intermediaries and barriers to Web site development and support are reduced.

Notwithstanding the vast expansion of the WebObjects world, the traditional WebObjects users remain at the core of the Apple market. For existing and new corporate customers, WebObjects continues to provide the features they are accustomed to using. Productivity improvements and efficiency are all relative: in a one-person operation, they may mean the difference between being able to support a Web site and not doing so, but in a global enterprise, they may mean the difference between profitability and the reverse. (Note, too, that as with desktop publishing, ease-of-use and widespread access to new technology also empowers individuals and small workgroups within the organization.)

WebObjects: The Big Picture

What exactly is WebObjects and where did it come from? The answers to those questions help you understand how to use WebObjects.

WebObjects is an application server. It is software that runs on a Web server and generates HTML pages to be downloaded to users' browsers. In generating those HTML pages, it uses templates, and it draws on connections to databases and other applications.

One of the features that makes WebObjects distinctive is its rich suite of development tools and its adherence to object-oriented design. These provide an easy-to-use and powerful development environment. In addition to being an application server, WebObjects is also a development environment.

WebObjects originated at NeXT, which Steve Jobs started after he left Apple Computer in the late 1980s. The company manufactured computers; in the process, it developed its own operating system. The hardware part of NeXT was dropped, but the operating system (called NeXTStep) was ported to Intel and other hardware. NeXTStep gained a reputation of being an excellent development platform and of being very stable, and it gained a new name—OpenStep. In the late 1990s, Apple Computer purchased NeXT to obtain the operating system, which became the basis for its next-generation operating system, Mac OS X.

WebObjects arose at NeXT in the early 1990s as an outgrowth of their database kit that let developers easily integrate databases into NeXTStep applications. When Apple bought NeXT, WebObjects came along.

Apple's long-standing leadership in computer graphics and video played a role in the evolution of WebObjects as a tool for providing rich graphics. To its two original

roles—application server and development environment—you can now add Web publishing, particularly for sophisticated media.

The next step for WebObjects is its widespread availability. No other product of this sophistication or complexity is priced at this level (although other application server vendors are in the process of rethinking their pricing).

WebObjects: The Details

WebObjects has a number of components and features. Each is described in depth later in this book. These components and features interact with one another, and you will find that you may need to jump back and forth, depending on your specific needs and background, so that you get all of the pieces together in a way that makes sense for you. For now, here is a brief guide to WebObjects components and features.

Enterprise Objects Framework

Enterprise Objects Framework is a framework—an object-oriented set of components—that encapsulates database data and the operations you need to perform on it. You write WebObjects code that interacts with Enterprise Objects Framework, which defines the specific database for use by WebObjects but without any of the product-specific concerns for the database manager that actually is to be used. Separately, you connect Enterprise Objects Framework to a specific database and database manager. Your WebObjects code is thus immune to most of the vagaries of distinctions among database managers.

Databases today are remarkable for their similarity and stability. They are, after all, among the most mature products of the computer age. All major databases today use SQL and they use the relational model.

Using Enterprise Objects Framework, you can describe the data that you need to work on in a WebObjects application. In a sense, Enterprise Objects Framework lets you define the nouns of your application: WebObjects—with its operations—defines the verbs.

Development Tools

Project Builder, Interface Builder, WebObjects Builder, and Enterprise Objects Modeler are the tools that you use to develop WebObjects applications. Each has a powerful and sophisticated graphical user interface that lets you drag and drop objects, draw connections, and otherwise implement your application with a minimum of typing. Each tool focuses on a different part of the WebObjects application:

- ▶ **HTML pages** Use WebObjects Builder to create HTML pages and to link their elements to business objects from the database or within the application.
- ▶ **Java Clients** Use Interface Builder to design your interface and to link its elements to your Java code.
- ▶ **Business objects** Use EOModeler to create business objects and interact with your database.
- ▶ **WebObjects projects** Manage the entire WebObjects project with Project Builder.

Deployment Tools

In addition to development tools, WebObjects comes with powerful deployment tools: Monitor, and Record and Playback Manager. These tools were developed to support WebObjects in its original environment: large-scale Web sites in large enterprises. They are among the most powerful deployment tools available to Web site and Web application developers.

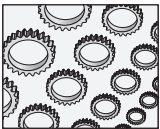
Monitor provides a means to manage and create configurations for your WebObjects application. WebObjects can run in very complex environments with multiple copies of itself running on multiple servers. Tuning network-based applications is extraordinarily tricky, and Monitor helps you do that.

Record and Playback Manager does just what its name says: it lets you record individual Web sessions and play them back later for testing. Using Record and Playback Manager, you can create test suites to test not only the functionality of your WebObjects application, but also its stability and performance under stress (when you play back multiple sessions at the same time).

Production Environments

WebObjects itself runs on Mac OS X, Windows 2000, Solaris, and HP-UX.

WebObjects 5 for Java is the first version written totally in Java. That means that it runs on any computer that supports Java; however, Apple certifies it with thorough testing only on some platforms.



NOTE

The WebObjects software that runs on all platforms under Java is the runtime software that interacts with a Web or HTTP server. The development tools run on fewer platforms. But the WebObjects applications that you develop can then run on any of the supported runtime platforms.

Development Environments

Most WebObjects applications interact with databases. WebObjects uses JDBC adapters to access these databases. In this book, OpenBase is used for all database examples.

Database Environments

A lot of WebObjects applications interact with databases. This interaction is through the use of adaptors for various data sources. The data sources supported directly by WebObjects are Oracle, Sybase, OpenBase, and Informix. Data sources can also be legacy and enterprise systems; WebObjects supports adaptors for PeopleSoft and SAP. In addition, WebObjects adapters are supported for open database connectivity (ODBC)—an open interface to relational databases ranging from mainframe products to personal computers.

OpenBase

WebObjects serves up rich data over the Web and usually that data comes from a database. Since 1991, OpenBase has been a leading relational database on platforms such as Mac OS X (before that on NextStep), Windows, and Linux. It has been used with WebObjects and with Enterprise Objects Frameworks on many projects, including some of the examples that ship with WebObjects. Although its functionality is comparable in many ways to that of larger, enterprise-scales databases such as Oracle and Informix, its pricing makes it a feasible choice for small to medium sized enterprises, as well as for larger enterprises that need its reliability and ease of installation and support.

Although adhering to platform-neutral SQL standards, OpenBase takes advantage of the Aqua interface to Mac OS X as well as WebObjects and Enterprise Objects Frameworks. It is tied to those products and supports them more than any other database. It can be used with other applications servers and frameworks, too. Because of its integration with and support of WebObjects and Enterprise Objects Framework and in view of the fact that it is readily available (trial versions are available with WebObjects itself and for free download from <http://www.openbase.com>), OpenBase is used in the examples in this book.

In addition, the second part of this book focuses on the use of OpenBaseManager in creating and maintaining databases. You can use it for stand-alone databases as well as for databases accessed by other software applications than WebObjects. That part of the book will help you in each of these situations.

Before You Start: Two Cautions

Before getting started with WebObjects, you should note two cautions. Its ease of use does not always translate into reduced development and implementation time, and it can live quite happily on a private intranet or local area network (LAN) as well as on the Internet.

Some people look at the ease-of-development and think it means brief development time. In many cases, that's true. However, it can also mean significantly higher quality because more iterations of design, development, and testing can fit into a project's schedule. (In practice, both brief development time and high quality are often achieved.)

This book describes the development process primarily as a straight road from project inception to deployment and maintenance. In reality, however, there are many detours, experiments, and false starts. This is an absolutely normal part of development, and WebObjects makes it easy.

You can use WebObjects to deploy applications on individual computers and on various networks that are not connected to the Internet. You often use the Internet, but it's not necessary. Likewise, although you often use databases in WebObjects applications, they are not required.