



WebObjects 5

**WebObjects 5
Reviewer's Guide**

June 2001



Contents

3 Introduction

4 The WebObjects 5 Architecture

5 Key Features

- 5 Automated data access with Enterprise Objects
- 5 Streamlined page generation with Web Components
- 6 Rich Java clients using EOAssociations
- 7 Integrated tools

9 WebObjects 5 Advantages

- 9 Rapid time to market
- 9 Built-in quality
- 10 Scalable Java deployment
- 10 Market-proven solutions

11 Positioning and Target Customers

- 11 Education
- 11 Publishing
- 12 Business

13 WebObjects 5 Background

- 13 Born at NeXT
- 13 Joining Apple
- 14 Moving to Java

15 Pricing and Availability

16 System Requirements

- 16 For more information



Introduction

WebObjects 5 is the powerful new release of Apple's award-winning application server—now written in the language of the Internet: Java. This allows WebObjects to run on virtually any server; makes it easily accessible to millions of Java programmers; and enables it to easily integrate and interoperate with other Java-based solutions such as EJB containers, servlets, ORBS, and web services.

The combination of a Java runtime with advanced native tools for Mac OS X and Windows 2000 makes WebObjects the obvious environment for customers needing rapid development of flexible, scalable web applications.

Built using the same NeXT-derived tools that spawned the World Wide Web, the mature, fifth-generation WebObjects frameworks also have the Xerces parser from the Apache Software Foundation integrated directly into them. This allows developers to import and export XML datastreams as well as supporting SMIL for multimedia applications.

Although WebObjects is cross-platform—enabling development on both Mac OS X and Windows 2000 and deployment on any Java 2 Platform, Standard Edition v1.3 system—WebObjects 5 is especially designed to take full advantage of the power of Mac OS X. In fact, the WebObjects deployment is now bundled as part of the newest release of Mac OS X Server.

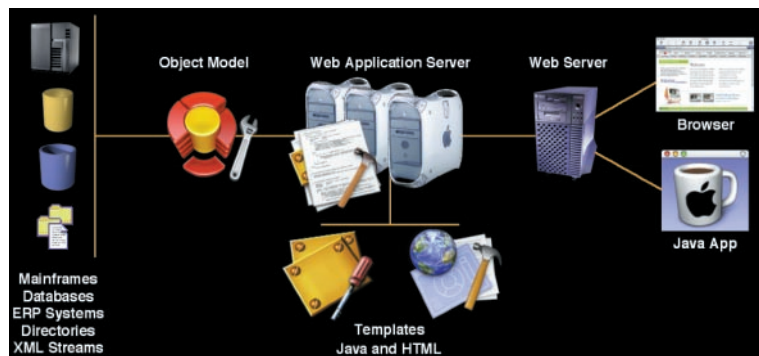
This document highlights key information about WebObjects 5. For those who wish to learn more about the product, WebObjects itself provides a wealth of documentation. Every copy of WebObjects includes *WebObjects Overview*, a 98-page book covering the rich architecture of the WebObjects frameworks. Also available from the Apple Developer Library is *Discovering WebObjects for HTML*, a 196-page step-by-step tutorial covering the key aspects of building three-tier web applications using WebObjects.



The WebObjects Architecture

All web application servers are designed to solve the same basic problem: how to create and maintain dynamic information. Static websites store all their content, presentation, and navigation information in HTML files. Not only does this make it very difficult for webmasters to do rapid or complex updates, it also means that every visitor sees exactly the same information in exactly the same way.

To solve this problem, application servers typically store their content in a database and use some form of scripting to create web pages dynamically, in real time. While this approach allows webmasters to create sites that have continuously updated content, such flexibility usually comes at the price of greatly increased complexity. This complexity lengthens development cycles, makes the website harder to maintain, and increases qualification efforts.



The WebObjects architecture.

WebObjects improves on the traditional application server architecture with an architecture that cleanly separates the database access layer, the application-specific Java code, and the web page or Java client presentation layer. Supported by rich object-oriented frameworks, this thoughtfully designed architecture is the foundation of the powerful technologies in WebObjects. Our patented object-relational mapping engine provides automated database access via Enterprise Objects. Web Components enable the efficient generation of HTML, XML, or SMIL from reusable templates. When HTML interfaces aren't enough, rich three-tier Java client applications are supported with EOAssociations.

An integrated suite of graphical tools streamlines application development. Unlike other graphical tools that generate hard-to-read code, the WebObjects tools dynamically bind application components with XML-like data structures, greatly simplifying application maintenance. Application components can be reused, and incremental development is easy. Both of those capabilities reduce your total cost of ownership over the life cycle of an application.

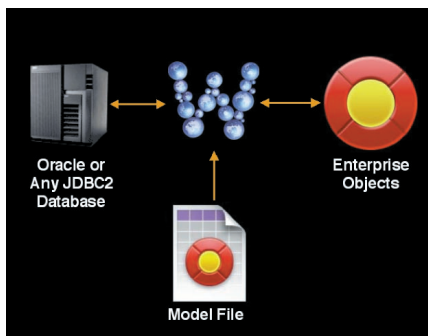


Key Features

WebObjects 5 offers a number of powerful features designed to speed application development, reduce errors, and permit easy updating and modification of your applications. These features include

- Automated data access
- Streamlined page generation
- Rich Java clients
- Integrated development and deployment tools

For additional information, please see the WebObjects 5 data sheet in the “Tech Specs” section of www.apple.com/webobjects.



Apple's patented object-relational mapping engine lets you write all your business logic using objects. WebObjects automatically fetches, caches, and updates data for you from any JDBC 2.0 database or XML stream. You may never write another line of SQL.

Automated data access with Enterprise Objects

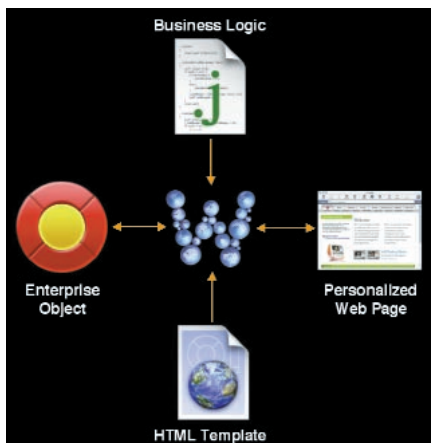
For a typical web application, the most time-consuming and error-prone step is managing database access. Generating correct SQL, understanding foreign keys and dependencies, ensuring that transactions are committed in the proper order—all these are difficult processes for a programmer to master and implement properly. And the price of failure is high. One erroneous entry could rapidly propagate throughout your application, corrupting the entire database.

To prevent such errors, WebObjects takes the programmer out of the data access loop. Instead, all database access is handled automatically via Enterprise Objects (EOs). The developer simply uses the graphical EOModeler tool to create a plist (an XML-like structured text file used for configuration information) called an EOModel. This model specifies how to map tables into objects, and encodes all the dependencies and the “to-one” and “to-many” relationships between them. The WebObjects frameworks use this model file to create Enterprise Objects, and transparently handle all the fetching, caching, and updating to and from the database.

After the EOModel is created, all the programmer has to do is to write straightforward Java code that accesses, modifies, and commits the Enterprise Objects. EOs can be thought of as entity beans with complete container-managed persistence. Perhaps just as important, this process is managed via configuration files rather than generated code. The database administrator can change databases, reorganize schema, switch to an ERP system, or even use multiple data sources at once. As long as the model file is properly updated, the programmer (and the code) never knows the difference.

Streamlined page generation with Web Components

The next challenge is turning objects into web pages. The traditional way to do this has been via some sort of server-side scripting, whether Perl, ASP, or JSP. However, this method can lead to a complicated mix of code and markup in the same file, and often requires one person to be both an HTML designer and a Java developer.



Web Components efficiently generate HTML, XML, or SMIL output from reusable templates. This clearly separates the presentation layer from your business logic and your data model, allowing each piece to evolve independently.

Today, most experts recommend using a template engine to generate web pages, an approach that isolates the layout and formatting issues from the logic of the application. This separation allows designers to create aesthetically pleasing HTML, and programmers to focus on efficiently generated code.

WebObjects pioneered this idea (way back in 1996) with Web Components (sometimes called WOComponents, after the class name). Web Components represent dynamic elements, which can be either a complete page or a single item such as a table. WebObjects includes a vast library of existing Web Components; you can easily subclass them to create your own library of reusable components.

Originally developed to generate HTML files, Web Components can also be used to generate XML or SMIL output, as well as to embed arbitrary client-side controls such as JavaScript. (SMIL is an XML-compliant language used to describe interactive synchronized multimedia on the web.) Because these templates use standard HTML, it's easy to create output in any language, including two-byte languages such as Japanese.

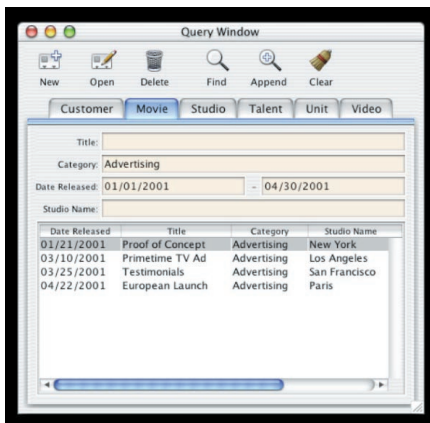
Each Web Component consists of an HTML file, a Java class, and a mapping file (which is just another plist). The HTML file is standard HTML with the addition of special *webobjects* tag. Each such tag refers to an entry in the mapping file, which defines a "WOAssociation." A WOAssociation uses various methods or instance variables in the associated Java class to generate HTML. When the HTML file is parsed by the WebObjects template engine, the tags are replaced by dynamic content from the associated class, using the WOAssociations.

These Java classes are automatically instantiated for each new session, and use Enterprise Objects as their persistence mechanism. Thus, the developer generally only needs to write a few lines of code covering each piece of application-specific business logic; all the session management and persistence is handled automatically by WebObjects. The loose coupling provided by the mapping file also makes it easy to update the Java code without having to touch the HTML file (or vice versa).

Rich Java clients using EOAssociations

Sometimes an HTML interface isn't rich enough or fast enough for interacting with large, complex data sets. To address these situations, WebObjects supports the creation of three-tier Java client applications using a technique called EOAssociations. An EOAssociation is a bidirectional relationship between a user interface widget and an underlying data object. EOAssociations are related to the WOAssociations used to manage the mapping in a Web Component, except that EOAssociations can talk directly to the user interface, avoiding the need for an intermediary.

The use of EOAssociations eliminates the need for the user interface to know about the web server (or even the database server, for that matter). All the UI designer has to do is associate a user interface widget with an appropriate data object, and the EOAssociation works with the underlying frameworks to manage all the appropriate transport, caching, and persistence. In particular, Java client applications maintain their own client-side object caches for optimum performance. You can easily move logic between the client and the server for optimum security.



When HTML isn't enough, WebObjects makes it easy to create highly interactive desktop clients using rich Swing widgets, while still preserving the security and flexibility of three-tier applications.



Integrated tools

To help you manage all these abstractions and their associated property lists, WebObjects provides a comprehensive suite of tightly integrated tools. These tools are available for both Windows 2000 and Mac OS X (v10.0), although the versions for Mac OS X are newer and take full advantage of the Aqua user interface.



Project Builder

At the heart of the WebObjects development process is Project Builder, a multilanguage integrated development environment (IDE) used extensively by Mac OS X developers. In addition to helping you edit, compile, and debug WebObjects applications, Project Builder organizes all your components (including localized resources for multiple languages); provides templates and assistants covering the common application types; and, on Mac OS X, integrates with source code repositories such as CVS.



EOModeler

Perhaps the most powerful tool in the WebObjects arsenal is EOModeler. EOModeler manages the object-relational mappings used by the data access layer, and can even help you organize your business logic. It can also be used to manage your database schema, either by reverse engineering a model from the schema or by using a new model to create the schema. In EOModeler, graphical entity-relationship diagrams aren't just pretty pictures; they represent the actual object relationships created and maintained by the data access frameworks.



WebObjects Builder

Although Web Components use standard HTML that can be edited with any text editing tool, WebObjects also includes a comprehensive layout tool called WebObjects Builder (WOB) that can manage the associated mapping files. WOB can be used for general HTML layout and provides the usual preview mode and drag-and-drop palettes common in such tools.

However, the real power of WebObjects Builder is its support for mappings. You can drag a dynamic element into your web page, specify the mappings, and set the properties. WOB automatically generates the appropriate HTML and mapping information, letting you focus on the aesthetics of your design. (And letting someone else focus on writing the associated Java code to implement the mapped methods.)



Interface Builder

For the creation of rich Java client applications, WebObjects includes the award-winning Interface Builder application, considered the crown jewel of the NEXTSTEP application environment that gave it birth. In WebObjects, Interface Builder defines the layout and database connections of three-tier Java client applications using a simple drag-and-drop paradigm. On Mac OS X, this same technology can also be used to create native Cocoa clients that talk directly to a database.

Deployment tools

In addition to all of the developer applications listed above, WebObjects provides a rich suite of deployment tools. The centerpiece of these tools is Monitor, itself a WebObjects application that manages all the different WebObjects applications and the servers on which they are running. You can use Monitor to start and stop instances, generate statistics, and configure how applications are run. Other tools enable you to record and play back sessions for both functional and stress testing.

Monitor

Applications

Hosts

Site

Preferences

Help

MediaServer

This page automatically updates every 60 seconds.

[Refresh Now](#)

Name	Host Port	Status	Start Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics					Configure	Delete	
							Trans- actions	Active Sessions	Average Trans- action	Average Idle Period	Deaths			WOSats
MediaServer-1	mascsvr07.2002						247510	75	0.059	0.078	-	 WOSats	 Config	 Delete
MediaServer-2	mascsvr01.2001						217839	53	0.088	0.033	-	 WOSats	 Config	 Delete
MediaServer-3	mascsvr05.2002						294076	87	0.155	0.023	-	 WOSats	 Config	 Delete
ALL INSTANCES:							759425	215	0.104	0.043	Clear Deaths	574.20 TPM	 Config	

The WebObjects runtime is written entirely in Java, allowing your applications to run on virtually any Java 2 platform. With the low-cost licensing and smart load balancing of WebObjects, it's easy to scale an application for as many users as necessary.



WebObjects Advantages

The comprehensive tools and frameworks of WebObjects make it an incredibly powerful way to create Java server applications. In particular, WebObjects provides four key advantages over competing solutions: rapid time to market, built-in quality, scalable Java deployment, and market-proven solutions.

Rapid time to market

WebObjects has long been the tool that demanding customers turn to when time is money. Its modular architecture enables database configuration, Java programming, and page design to proceed in parallel. Its rich object-oriented frameworks do most of the hard work for you, and its integrated tools streamline the process of application design and development.

An added bonus is the ability to create fully functional prototypes in seconds using the Direct to Web and Direct to Java Client assistants. Leveraging the power of the WebObjects frameworks, these tools apply heuristics to the data model to define a complete three-tier HTML or Java application instantly. The appearance of the application is driven by a sophisticated rule engine, which can be configured inside the application by using Java applets or with the comprehensive RuleEditor included in the WebObjects developer tools. These prototypes allow you to browse and edit the database, and can provide a useful starting point for creating more sophisticated interfaces.

Built-in quality

Developers are all too familiar with Murphy's law—namely, that anything that can go wrong *will* go wrong. WebObjects provides “Murphy protection” by reducing the number of things that can possibly go wrong. The WebObjects frameworks were created by what many people consider the best object-oriented design team in the history of our industry. Now, in their fifth generation, these frameworks have been fine-tuned to handle all the basic tasks of application server development for you.

Because the vast majority of a WebObjects application is well-tested framework code—not generated code—there's very little application-specific code to debug. Also, the isolation of data, logic, and presentation elements greatly improves both the flexibility and security of your product. Key information such as database referential integrity constraints (identifying dependencies between objects) is embedded directly in the model, rather than being wholly dependent on programmer discipline.



Scalable Java deployment

Most exciting of all, with this release, the power of WebObjects is now fully accessible to Java programmers. The WebObjects runtime is written to the Java 2 Platform, Standard Edition v1.3, allowing it to run virtually anywhere.

Now that WebObjects is available on Java, it is instantly familiar and accessible to the millions of programmers who prefer that language. The use of standard Java also makes it easy for WebObjects to interoperate with other Java solutions, such as EJB containers, servlets, ORBS, and web services.

Another benefit is that the Xerces-J XML parser from the Apache Software Foundation is integrated directly into WebObjects. WebObjects uses this parser to import and export XML data streams directly into your application, where you can manipulate the data using the power of Enterprise Objects.

Compared to other solutions in this price range, WebObjects has the scalability to grow along with your business. WebObjects makes it easy to dynamically add, manage, and remove servers in order to handle all your deployment needs.

Market-proven solutions

These benefits are not just empty theory. In its five years of life, WebObjects has proven itself time and time again as having the most mature object-oriented application server frameworks. This leadership has been recognized by customers ranging from Apple's core markets of education and creative professionals to *Fortune* 500 corporations.

The preeminence of WebObjects is reflected in the wide range of awards it has received, from best server to best tools to product of the year. In 2000, we won the *Java Developer's Journal* Readers Choice Award for the second year in a row, with almost twice the votes of the second-place application server.

Apple believes so strongly in WebObjects that we use it ourselves as the basis for our entire Internet strategy. WebObjects is our primary tool for delivering dynamic information to our customers, providing the engine behind Apple's iCards, iReview, and iTools portals. It also powers the Apple Store, one of the best e-commerce sites. WebObjects even plays a crucial role behind the scenes in managing www.apple.com, arguably the most popular computer company site on the Internet.

WebObjects is also a crucial aspect of Apple's server product strategy. We bundle an unlimited WebObjects deployment as part of Mac OS X Server, and many customers use WebObjects to help deliver QuickTime Streaming Server solutions.



Positioning and Target Customers

WebObjects is Apple's incredibly powerful "Internet engine" for addressing the custom development, enterprise integration, and web publishing needs of educators, creative professionals, and other major Apple customers. It leverages the power of Java, XML, and open source code to enable the rapid creation of database-driven, cross-platform solutions for e-learning, multimedia workflows, and e-commerce. It's even more powerful when it's run on the UNIX-derived Mac OS X platform. Together, WebObjects, Mac OS X Server, and QuickTime Streaming Server provide a complete Apple solution for the delivery of rich, interactive content over the Internet.

Education

WebObjects is a key part of Apple's strategy for education. In K–12 schools, WebObjects will be used as part of Apple's next generation of the PowerSchool school management system. PowerSchool enables students, teachers, administrators, and parents to access comprehensive student and school information online.

In higher education, WebObjects is used by universities (often with the help of Apple iServices) to create portals, labs, and administrative solutions, such as the new student information site at the University of Michigan. Many universities are looking to use WebObjects as the basis of their web development courses, allowing students (or continuing education professionals) to create complete web applications in a single semester.

Publishing

WebObjects is the lifeblood of some of the most important publishing solutions for creative markets. Most notable is Adobe's InScope software for publishing workflows, which is built entirely around WebObjects. Other products for this market that are based on WebObjects include

- ReportMill, for the dynamic generation of PDF, QuickTime, and Flash output in web or video applications
- MAMBO from WebWare, for web-based asset and brand management
- SiteGenesis from Cassini Division, for comprehensive site management (used, for example, by the BBC)
- CreativeStandard, which enables web designers to create complete e-commerce solutions without becoming programmers

In addition, hundreds of other components and services have grown up around WebObjects during the years it's been on the market. Many of these can be integrated into your project easily due to the robust object-oriented structure of WebObjects.



Reviewer's Guide

WebObjects 5

12

Business

A combination of outstanding technology and innovative partners has attracted some of the best-known brands in the world to WebObjects. *Fortune* 500 companies have trusted it for years for Intranet application development, personalized Internet content delivery, and e-commerce. WebObjects-based solutions in this market include cutting-edge Internet sites such as the BBC, Skymall, AAA, and MCI. WebObjects is also being used in powerful online applications for asset management or customer services, including tracking United Parcel Service packages and providing online services to U.S. Postal Service customers.



WebObjects Background

WebObjects 5 continues a proud, longstanding heritage of industry-leading web technology. From predecessors that helped give birth to the World Wide Web to the current Java-based release, WebObjects has become ever more powerful, comprehensive, and sophisticated.

Born at NeXT

WebObjects has its roots in NEXTSTEP, the award-winning object-oriented development tools used by Tim Berners-Lee to create the original web browser at CERN. NEXTSTEP employed a number of powerful design patterns that greatly simplified the creation of rich, graphical applications. This ease of use led to its adoption by a number of large enterprises that wanted to rapidly create sophisticated database applications.

To better serve this market, NeXT created a technology known as Enterprise Objects, the first commercially successful object-relational modeling framework. Because Enterprise Objects abstracted away all the work of database access, it quickly became the technology of choice for financial institutions creating sophisticated client/server applications.

In 1995, NeXT realized that the World Wide Web would soon become the dominant format for client/server applications. Therefore, NeXT programmers applied that same object-oriented knowhow to create a sophisticated set of frameworks for managing HTTP requests and HTML generation. In January 1996, NeXT announced WebObjects 1.0, the world's first object-oriented web application server. The power of WebObjects, leveraging the strengths of Enterprise Objects, quickly made it the dominant product in NeXT's portfolio.

Joining Apple

In 1997, NeXT was acquired by Apple, primarily for its engineering talent and the UNIX-based operating system technology that eventually became the core of Mac OS X. WebObjects played a key role at Apple during this time, driving Apple's public Internet efforts such as the Apple Store and iTools. However, WebObjects as a product was sold only through the Apple Enterprise Software (AES) group, which also provided training, consulting, and support services for it. This combination of direct sales and a high price (up to \$50,000 for a multi-CPU deployment license) kept WebObjects out of the reach of the vast majority of Apple developers.

Last year, after Mac OS X was solidly under way, Apple decided it was time to make both WebObjects and AES more relevant to the broader Apple marketplace. At the WorldWide Developers Conference in May 2000, Apple announced that it was cutting the price of WebObjects to only \$699 for a single box containing both developer tools and a full deployment license, which would be sold via both the Apple Store and traditional retail channels.



At the same time, the professional services portion of AES was reformulated as Apple iServices, with a charter to focus on all of Apple's Internet technologies, including WebObjects, Mac OS X Server, and QuickTime Streaming Server.

Moving to Java

In its first several releases, WebObjects was written in Objective-C, a simple object-oriented language that provided much of the inspiration for Java. As Java became more and more important for server-side development, WebObjects 4.x added Java APIs. However, these APIs accessed the Objective-C frameworks via a "bridge" that reduced performance and complicated debugging. Therefore, in May 2000, Apple announced its intention to make WebObjects a true Java application.

With WebObjects 5, Apple has delivered on that promise, and the original web application server is now available in the language of the Internet: Java. This Java compatibility allows WebObjects to run on virtually any server, and to easily integrate and interoperate with other Java-based solutions. We expect the combination of a Java runtime with advanced native tools for Mac OS X and Windows 2000 to enhance the already solid popularity of WebObjects with customers needing rapid development of flexible, scalable web applications.



Pricing and Availability

To help customers and the marketplace better appreciate the power of WebObjects, and to make it more affordable to smaller organizations, Apple has set the price of WebObjects 5 at \$699 (only \$99 for educational institutions). A single package contains both the developer tools and a full deployment license. The license has no transaction limits, and allows you to install the runtime on a separate machine—even a high-end server with multiple CPUs—at no additional charge.

Now that WebObjects 5 is available in Java and runs on Apple's new UNIX-based Mac OS X operating system, we will be working hard to make the Java developer community aware of the incredible power of WebObjects. WebObjects had a dedicated track at Apple's WorldWide Developers Conference in May 2001 that included 17 separate sessions, and it was a prominent part of Apple's booth at JavaONE. Time-limited WebObjects evaluation CDs and downloads are being distributed by the Apple Developer Connection, even to free online members. The new Mac OS X Server (v10.0) includes a full deployment license for WebObjects, making it the first server software platform to include an enterprise-scale web application server.

WebObjects is available for sale via the Apple Store worldwide, as well as from various catalogs and selected retailers. Up-and-running support is provided by our award-winning AppleCare organization (www.info.apple.com/webobjects). API-based developer technical support is available through the Apple Developer Connection (connect.apple.com). For more in-depth assistance, our Apple iServices professional services organization (www.apple.com/iservices) provides a wide range of WebObjects solutions, including consulting, training, and personalized technical support.



System Requirements

Development platforms

- Mac OS X (v10.0)
- Windows 2000 Pro

Deployment platforms

WebObjects applications are designed to work with any Java 2 Platform, Standard Edition v1.3 runtime environment, using JDBC 2.0 for universal database connectivity. Following is the current list of deployment platforms that Apple has qualified for use with WebObjects 5 and for which we offer technical support. For updates, please visit www.apple.com/webobjects.

- Mac OS X Server (v10.0)
- Windows 2000 Pro
- Solaris 8

Supported data sources

WebObjects applications are designed to use JDBC 2.0 for universal database connectivity. The following database has been qualified to work with WebObjects 5. For updates, please visit www.apple.com/webobjects.

- Oracle 8i

For more information

For technical or sales information about WebObjects 5, please visit www.apple.com/webobjects. For developer information, please see www.developer.apple.com/webobjects.