

# Home

## About the WOCCommunity

The WOCCommunity mission is to support, motivate, and enable WOCCommunity activities and projects. Our main project is Project Wonder, a collection of frameworks that sits on top of Apple's WebObjects Web application server and frameworks.

## About Project Wonder

Project Wonder is the largest open source collection of reusable WebObjects frameworks, applications and extensions. Also included in Project Wonder are deployment software and web server adaptors. Project Wonder development is done within the Eclipse IDE with the WOLips plug-in tools.

WebObjects was initially created at NeXT. There was a framework called DBKit, which provided a way to abstract legacy databases to an object-relational graph. This framework later became the Enterprise Objects Framework (EOF). EOF was used to build nibware, runnable applications for NeXTStep/OpenStep that were built with Project Builder and which used interfaces that were designed in Interface Builder. If this sounds familiar, it is because the ideas from that time have only been refined, and not fundamentally changed, to become the Cocoa frameworks for applications running on Mac OS X and iOS today.

WebObjects was built on top of DBKit/EOF to provide a way to create dynamic applications for the web. The first versions of WebObjects were created at NeXT around the latter half of 1992.

## Our sponsors



**ENTERPRISE OBJECTS FRAMEWORK RELEASE 1.1**

The screenshot shows a complex graphical user interface for the Enterprise Objects Framework. It includes a main window with a table of data, a bar chart, and various control panels. The Oracle logo is visible in the bottom right corner of the interface.

*The Enterprise Objects Framework seamlessly extends the power of NEXSTEP's object-oriented paradigm to the development of client/server database applications.*

The Enterprise Objects™ Framework builds on NeXT's leadership in object-oriented software development tools by dramatically reducing the costs of building and maintaining database applications. The Framework is unique in its ability to bring the benefits of object-oriented programming to relational database application development. The Enterprise Objects Framework enables developers to construct reusable business objects that combine business logic with persistent storage in industry-standard relational databases. Developers create business objects once and reuse those objects to manage enterprise-wide operations.

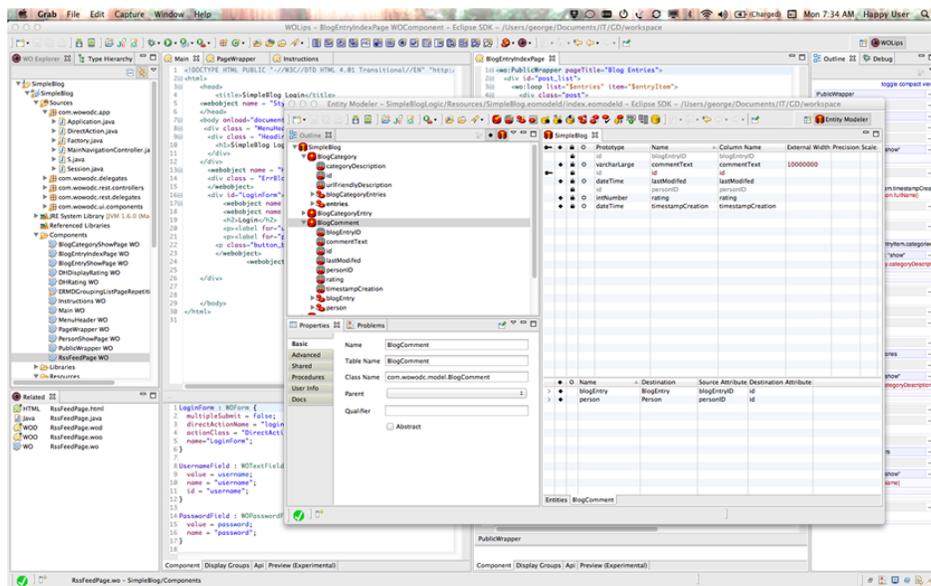
Applications developed with the Enterprise Objects Framework support a three-tier client/server architecture by maintaining a clear separation between the user interface, business objects, and the database. Business objects are conceptually insulated from the underlying data structures used for data storage. Any changes to the database schema are easily accommodated without recoding. The same business object can be reused even when data is stored differently throughout the organization. The Enterprise Objects Framework also allows business objects to be developed independently from actual data sources. The architecture allows arrangement of individual objects or entire applications to new data sources, including relational databases, hierarchical databases, and on-line news feeds, without recoding.

Portable Distributed Objects™ (PDO™) extends NEXSTEP's industry-leading object model across heterogeneous operating systems. With PDO, developers have an infinitely flexible choice of object deployment strategies. Business objects can be deployed in NEXSTEP clients and a variety of server class machines running HP-UX, SunOS, Solaris, and Digital VMS operating systems. The pairing of PDO with the Enterprise Objects Framework creates a distributed computing environment that enables applications to scale by leveraging all computing resources across the enterprise.

Nearly 20 years later, Project Wonder and the [WebObjects Community](#) actively support the development of WebObjects based applications.

Project Wonder started its life as a collection of open source frameworks build on top of Apple's WebObjects. Over time, Wonder became as large as WebObjects and not only provided complimentary frameworks but also provided fixes for WebObjects bugs, and development and deployment tools. Starting with Snow Leopard, Apple stopped providing releases of WebObjects outside Apple, but the community love WebObjects so much that we decided to continue development with Project Wonder.

It provides a powerful and mature set of Object-Oriented frameworks for managing Object-Relational Mapping (ORM) to any JDBC-compliant database, Session Management, Undo/Redo/Revert, Ajax, Web Services, full Java Client applications, Rapid Application Development, file upload/download and many other popular internet and enterprise application technologies.



## Java on the Web, Done Right!

Enjoy working in a clean and elegantly designed set of coherent, consistent frameworks. Delight in spending your time implementing features instead of fussing with infrastructure. You don't have to deal with EJB, J2EE containers, Struts and the like. There is no need to hand edit XML and use dependency injection to get the pieces to co-operate. Of course, WebObjects has full access to the universe of Java libraries (JavaMail, Apache projects, etc.) and the cross platform portability that Java developers depend on.

You can develop and deploy on any OS that can run Java (the WebObjects license does say that you have to develop on Apple hardware, so developing in Windows or Linux on a Mac is ok). This allow you to use other Java libraries with Project Wonder, many in the community use libraries and tools like Apache Commons inside their WebObjects apps.

## What's in for Cocoa developers

Since WebObjects was in Objective-C in the past, it shares some design patterns with Cocoa. You use Cocoa because you like clean APIs, good tools, faster development process and turnaround, and KVC ? WebObjects, because of its NeXT foundation, shares a lot of concepts coming from Cocoa (especially Foundation). You don't use `java.util.Vector`, you use `com.webobjects.foundation.NSArray`, which should be familiar to you. You can also use Project Wonder as a back-end for your Cocoa (and Cocoa touch) application, in fact many members of the community use Project Wonder as the back-end of their iOS applications.

## It's to build Web applications, not Web sites

You might have notice that Project Wonder is not used on many "public" Web sites, and the main

reason to that is that Project Wonder was made to build Web applications, not Web sites. If you only want to have a contact form or make a read-only Web site, Project Wonder is not a tool for that. But if you need to build a Web app that needs to fetch and update content, WebObjects is your friend. It's funny to see people go crazy over some Web frameworks to create Web apps, because WebObjects was doing that since day one.

## I'm a Cocoa developer, why should I use Project Wonder?

Well, you use Cocoa because you like clean APIs, good tools, faster development process and turnaround, and KVC ? WebObjects, because of its NeXT foundation, shares a lot of concepts coming from Cocoa (especially Foundation). You don't use `java.util.Vector`, you use `NSArray`, which should be familiar to you. You can also use Project Wonder as a back-end for your Cocoa (and Cocoa touch) application.

## I'm a J2EE developer, why should I use Project Wonder?

If you are a J2EE developer, forget what you learned about EJB and Struts, you will be glad to use EOF and components instead. But the fun part is that you don't need to learn a new language. You can reuse some of the code of your J2EE apps, and the official Project Wonder IDE is based on Eclipse. If you ever used Apache Cayenne or Tapestry, you will feel even more at home.

## What Project Wonder and WebObjects are not

WebObjects is not a web page design tool. In fact, WebObjects does not have a WYSIWYG HTML editor (but the component editor is quite powerful). WebObjects uses your web pages (called templates) and adds dynamic content on the fly to make your application run. Also, WebObjects is not a framework for writing client side code. The objects you work with in WebObjects exist on the server only. If you are writing a complex client side JavaScript based application you can serialize your server objects via XML, JSON (by using [ERRest](#)) or SOAP and send them to a browser.

## Contents

- [Getting Started](#)
  - [Project Wonder Installation](#)
  - [Your First Project - Hello World](#)
  - [Your First Rest Project](#)
  - [Your First Framework](#)
  - [Your First Deployment](#)
- [Binary Frameworks Installation and Upgrade](#)
- [Quickstart](#)
- [Example Applications from Wonder Source](#)
- [Frameworks](#)
- [History](#)
- [Using Properties Files](#)
- [Wonder Tutorials](#)
- [Best Practices](#)
- [JavaDoc API](#)
- [Testing](#)
- [Application Startup Troubleshooting](#)
- [Acknowledgements](#)

## Deployment Tools

WebObjects 5.4 was released with the source code for JavaMonitor and wotaskd as Example Source. If you have WO 5.4 installed, these should be on your file system at `/Developer/Examples/JavaWebObjects`. These projects were imported into Wonder, bug fixed and enhanced.

- [Deployment Overview](#)
- [Wonder JavaMonitor & wotaskd](#)
- [WOAdaptorInfo](#)
- [WOStats](#)

## Wonder Frameworks and Components

### Bug Reports and Enhancement Requests

- [Wonder Bug/Issue Tracker](#)

### JavaDoc

Nightly build [JavaDoc API](#) on [WOCCommunity.org](#).

And there is [JavaDoc API](#) generated by doxygen.