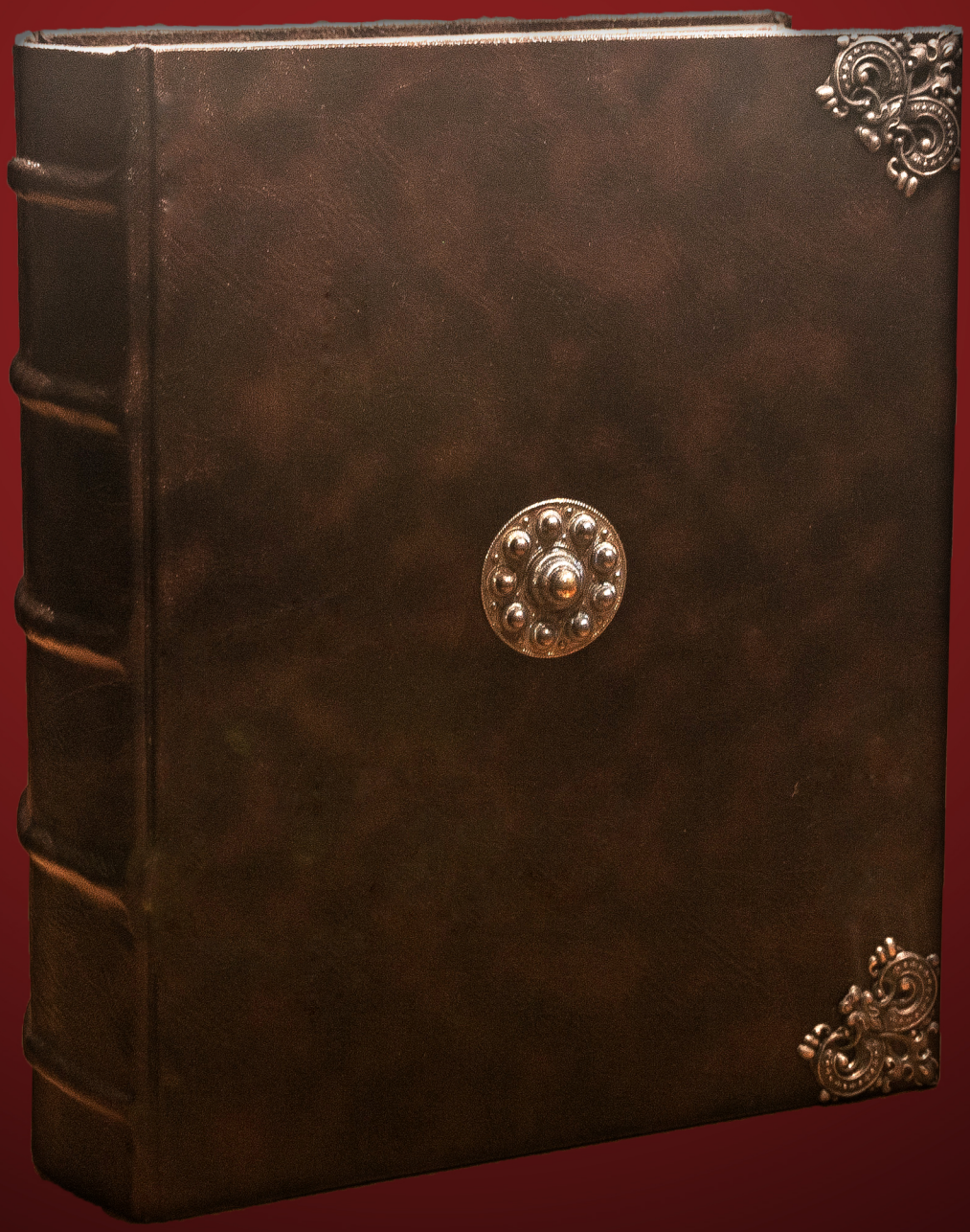
Markus Ruggiero

# LEARNING THE WONDERS

An introduction to creating great web applications with Project Wonder



# Learning the Wonders

An introduction to creating great web-applications with Project Wonder

by

**Markus Ruggiero**

**Learning the Wonders**

**Table of Contents**

## About this book

WebObjects is a very mature technology. Unfortunately WebObjects being a product of Apple Inc. has not seen much support from Apple for the last couple years. Due to political reasons Apple has set WebObjects' status to deprecated. Nevertheless there is a vivid community developing tools and frameworks to work with and enhance WebObjects. The toolset is known as WOLips, an Eclipse plugin. Project Wonder is an extensive collection of frameworks on top of and often replacing WebObjects. Both WOLips and Project Wonder are open source.

Several very good books about WebObjects have come out during the high time of WebObjects. Unfortunately with Apple pulling back the commercial side of creating new books and updating existing ones appeared not so interesting to large publishing houses. The last official commercial publication came out around 2005 with one book having an update in 2008. Since then the tools and the frameworks have made huge steps forward. Today the only documentation for all the new and enhanced features is available from *wiki.wocommunity.org* and some scattered private homepages. This is a vast collection of information written by programmers for programmers. However there was, and still is, nothing geared towards the beginner. WebObjects and Project Wonder have quite a steep learning curve; getting started just from the available documentation is difficult at best. All the old books are still valid resources for concepts but they are all based on the old toolset and thus confusing for the beginner.

Everyone in the community agrees that new programmers should be brought aboard. During WOWODC 2012, the World Wide Wonder Developer Conference, I decided that a modern book was needed. What you have in front of you is the result of that decision. I have written the book in my spare time, it is currently kind of a hobby project. However I do already have ideas for a successor.

## Who should read this book?

This book is intended for any programmer who wants to start creating powerful server applications with WebObjects and Project Wonder. You should be fluent with Java and the Eclipse IDE, albeit no need to be a guru. You should have a good basic understanding of object-oriented concepts. It also helps if you have basic knowledge of html and css, as well as know how to handle a relational database. The ability of being able to at least read some SQL helps. The book will extensively cover many conceptual details, but it is not meant as a guide to starting programming at all.

The primary focus for this book was to make things clear. Each and every example has been played through and everything has been documented with a screenshot. You should be able to follow every step and redo them on your own. The screenshots will show you how things are supposed to look. While the book tells you what to do, it also details the *why* and so enables you to do things differently if you want to.

## About the Author

Markus Ruggiero has been programming professionally since the mid 1980's. He started to use WebObjects in 2000. Around the same time Markus Ruggiero wrote his first course book. Since then he has written many more. All are widely being used in formal teaching of young application development professionals in Switzerland's dual-track system of vocational and professional education and training

Today Markus Ruggiero is part-time teaching all sorts of programming topics ranging from structured to object oriented programming, web technologies, and relational database design. When he is not teaching he works for his own company doing primarily project work for large international customers – of course all the big projects are Wonder based!

Markus Ruggiero has a master of science in applied physics and a degree in pedagogics.

You can reach him through the official web page for this book:

*http://learningthewonders.com*

## About the Book Cover

Some twenty years ago I rescued a bunch of old Olivetti boards on hinges. I can't remember exactly why I did this, because they were just useless junk. But they somehow looked like a book in need for a cover and that made me take the boards home. I stowed them away in the basement, thinking, that once I have enough time I am going to complete ***The Book***.

In 2011 the German computer and technology magazine *c't* called for a contest under the motto *"Mach Flott den Schrott"* (create something out of old computer hardware parts). Around the same time, I was introduced to a friend of a friend, who, I learned later, was a professional bookbinder. She was immediately fascinated when I told her about the old computer boards in need of a cover. She helped me create a beautiful book, looking some 200+ years old. The book won second price in the aesthetics category and was nominated third and fifth in two more categories.

*Markus Ruggiero, Summer 2013*

# Preface

By Charles (Chuck) Hill

Earlier this year, I needed to introduce a person that I recently hired to Wonder and WebObjects. Having no other resources, I grabbed some blank paper and a pen and sat John (his real name!) down. An hour or so later, John looked dazed, confused, and pretty much done for the day. There have been a few books on WebObjects published over the years. Sacha Mallais and I brought Practical WebObjects to the market in 2004 in an attempt to share some hard won knowlege. It was our hope to follow this up with an introductory book. Alas, that was never to happen. When Markus Ruggiero said he was going to write one, I thought he was crazy. I knew first hand how much work it was. What I did not know well enough was Markus! Not only did he write the missing introductory book, he also managed to cover Project Wonder which is a vital part of any modern WebObjects-based application. The Wonder community is fortunate to have this book written by someone with Markus' deep technical knowlege and pedagogical background.

The early days of WebObjects were characterized by fragmented add-ons, mostly closed source commerical products. NetStruxr's release of the original Projects Wonder frameworks as Open Source became a rallying point for new functionality. Wonder continues to be the cutting edge of new technology based on the WebObjects framework. It demonstrates the elegance with which technologies like REST and Ajax can be integrated with the original technology of a decade before. Wonder leverages the power of WebObjects and makes application development even more effective.

This book may change the way you think about software. Using this technology for the last 15 years has certainly changed mine. The core WebObjects technology started in the late 1990s It is old; that can't be denied. The origin was eons ago in technology terms. Beware: that does not make it outdated, this is not COBOL. It is still used today by those who know its secrets, its strengths, and its power. The engineers who designed and built the WebObjects foundation upon which Wonder was created were remarkable in their understanding of Object Orientation and how to design for extensibility and long-term maintainability. They were visionaries. Many went on to found today's popular technologies. We have much to learn from them.

What can Wonder and WebObjects teach you? It is a treasure trove of design patterns and well thought out design that has withstood the tests of time and highly varied environments. WebObjects was model driven years before Model Driven Architecture became a common term. It had one of the first commercially successful Object-Relational Mappers and inspired and influenced many who followed, incuding the CoreDate framework used by iOS. The component based, stateful UI layer provided an entirely different paradigm for writing web applications. The rule-based DirectToWeb (D2W) technology remains revolutionary to this day. Your software development palette will be much richer from exposure to these ideas, goals, and technologies.

Whether you are a student of software design, or looking to build a maintainable, scalable server application, this book will provide an excellent education. WebObjects/Wonder is the best choice for a server-based service for iOS applications. As I hire and train new people in WebObjects and Wonder, Learning the Wonders is going to be extremely valuable for me. Look deeply here; there is much of value to be learned. John will be pleased, but not as much as my next hire who won't have to suffer through my introduction!

# Acknowledgments

There are so many great people who helped create this book, most unknowingly. When I started to learn WebObjects and later Wonder the folks from the webobjects-dev mailing list have been and still are the ones that helped me the most. People like Chuck Hill, Ramsey Gurley, the various Davids (Avendasora, Holt, LeBer), Kieran Kelleher, Mike Schrag, Pascal Robert, and many many others (sorry, it's just impossible to mention every name here) answered mine and other's questions with great knowledge. In addition the various books about WebObjects and particularly the famous Practical WebObjects by Charles Hill and Sacha Mallais were an invaluable source of wisdom. Meeting you all at WOWODC (WebObjects and Wonder Developer Conference) was a great experience in its own. I do hope I can give something back to the community with his book.

Thank you!

Markus Ruggiero, Summer 2013

# Overview

The book is divided into five parts.

***Part 1*** will introduce you to WebObjects and Project Wonder and give you a bit of history. Main focus will then be setting up the development environment. You will learn what is needed and how to install all the necessary tools. It gives hands-on tips for directory layout and configuration options.

In ***Part 2*** we will cover all the basics of WebObjects and Project Wonder. You will learn how a wonder application works, how the tools work. We will extensively cover important concepts and see how to take advantage.

***Part C*** is dedicated to one of the most important issues: how to properly and elegantly access a relational database from a pure object-oriented application. You will learn about modeling your business objects and how to map objects, attributes, and relationships to a relational database.

***Part D*** is kind of a grab bag for various things. You will learn about cookies, some very sophisticated framework classes and mechanisms, and we will have a deeper look into debugging.

After successfully creating your application, you probably want to deploy it to a server. Deployment is the big topic in ***Part E***. We are going to see how you build your application for deployment, how to set up a deployment environment, and of course how to run your application inside the deployment environment.

### Conventions used in this book

We use *italics* for any filename, URL, name for an executable program or script, and names for our own classes and code elements.

Names for standard framework objects like class names are printed in a `mono-spaced font`. The same font is used for all `code fragments` and `method names` inside the flow of text. Method names like `takeValuesFromRequest()` always have opening and closing parentheses. Method parameters are usually not shown, unless they are important in the current context.

All user interface elements like buttons, menu items, or text field labels are shown in SMALL CAPS

Important new definitions, concepts, or terms are emphasized with ***bold/italics***

```
Code listings are in mono-spaced font and boxed
```

The dollar sign `$` in code sections denotes the command line prompt. If the distinction between Unix commands and DOS commands is important, `DOS>` may be used as a prompt for DOS command lines.

Sometimes it is important that a command is written all on one line, but the printed line breaks. In such a case, the return arrow ↵ denotes that the command should not break.

# Part A - The Environment

# 1 Introduction

You are about to learn a lot. Yes, really. And it won't come easy. But the reward will be tremendous!

Let me give you an overview of what's to come:

We are going to learn how to use Project Wonder to create really great applications for the web. At the beginning we will play with rather simple HTML user interfaces but you will see that the Web 2.0 lies just ahead. Project Wonder is very sophisticated and mature but it needs some getting into. We will go there together.

No application can live without data, and in most cases data lives in a relational database. So we need a way to access relational databases and work with that data. You will learn how to bridge the gap between a modern Web 2.0 application that is programmed in pure object-oriented fashion and the relational world.

Oh, did I mention that Project Wonder is pure Java? The big vast Java world is out there just waiting to be called into your application.

For a start here is a bit of history.

## 1.1 A bit of history

In the 1990's a company called ***NeXT*** was trying to come to market with great new hardware and an even greater operating system called ***NeXTSTEP***. NeXT was founded by Steve Jobs, the man who created Apple Computer. One of the main goals of the NeXT operating system was to provide a clean object oriented interface to the programmer. For this NeXT used a programming language called ***Objective-C***. In 1996 Apple bought NeXT and made NeXTSTEP the foundation of the modern Mac OS X as well as later iOS, the operating system for iPhone and iPad. Still today a modern descendant of Objective-C is the implementation language for Mac and iOS programs.

In the NeXT era NeXT built a framework that allowed object oriented access to relational databases. This framework was called EOF Enterprise Objects Framework. When in the mid 1990's "The Web" started to become more wide-spread NeXT created another set of frameworks called ***WebObjects*** to generate applications with an html user interface so that the applications could be used from everywhere where there was an internet connection and a browser available. WebObjects and EOF were marketed together under the name of WebObjects.

The merger with NeXT brought WebObjects into Apple's hands. Around the year 2001/2002 Apple rewrote WebObjects (including EOF) in the Java language, thus making it more interesting to the "outside" world. At the same time, Apple lowered the price from $50,000 to $699.

Apple themselves used and still uses WebObjects heavily for their internal projects. This is continuing today with WebObjects being an important tool for Apple. Look at some of the URLs when you surf the Apple homepage. You will often see the string "WebObjects" embedded. No Apple Store, no iTunes Music Store, no Apple Developer Connection, and many more would have been possible to create in such a short time frame without WebObjects.

Apple had their own development tools for Objective-C and Java. However the non-Apple Java world started to use cross-platform tools, most often Eclipse, for developing Java applications. WebObjects was Java but Eclipse could not be used to develop WebObjects applications because

there were no WebObjects specific tools available for Eclipse. All the Apple / NeXT tools were written in Objective-C and over the years it become more and more difficult to keep them running as modern Mac OS X evolved. A group of open source developers started to create the necessary tools as plug-ins for Eclipse for developing WebObjects applications, resulting in the WOLips toolset. At the same time other groups using WebObjects started to build and provide to the community their own WebObjects extensions frameworks. These frameworks eventually were collected under the name Project Wonder.

Today ***Project Wonder*** is an immense set of frameworks providing all sorts of functionality from simple generation of html output, sending emails, creating RSS feeds, creating PDF and Excel files, working with WebServices, to full Web 2.0 Ajax applications. In addition, Project Wonder has implementations of augmented versions of many standard WebObjects classes fixing bugs and teaching them new tricks. WebObjects is still part of Project Wonder, but more and more of WebObjects has been superseded by Project Wonder classes.

## 1.2 Where do we stand today?

Project Wonder is a very mature set of frameworks for building the most sophisticated web applications. WebObjects is at the core of Project Wonder. Today we are not programming with WebObjects any more (well, sort of, this is not really true) but we create Project Wonder applications.

In the last couple years Apple has shifted focus from being a computer company to creating more consumer lifestyle products like iPod, iPhone, and more. WebObjects is clearly not a consumer thing and thus Apple, seeing a lively and grown up community to take over, decided to not provide support for WebObjects anymore. When you go to Apple's web site and search for WebObjects, you will not find much - and what you find is marked deprecated. This is unfortunate, but pure company politics. However do not despair: you have Project Wonder! Project Wonder is today's WebObjects. Project Wonder is open source and supported by a great community.

Now let's really start with this book and dive into the wonders of Project Wonder!

## 1.3 What do you need to create great Project Wonder applications?

First of all you need a computer. Not just any computer but it ought to be an Apple Mac running a modern version of Mac OS X. Apple licensing states that all WebObjects development must be done on Apple hardware platform. Deployment is allowed wherever you have a modern Java runtime environment (Mac OS X, Microsoft Windows, Linux, any Unix, and whatever else you might have and want to use). This license limitation is for the WebObjects frameworks only. Project Wonder has no such limitation!

As WebObjects is pure java there is no technical limitation for development on any platform as long as there is Eclipse available. People have Project Wonder/WebObjects development environments running on Windows and Linux. Think of a Windows installation running on an Apple Mac with Boot Camp. This is clearly a scenario where the development tools run on Apple hardware platform. Throughout this book we will run our development environment on Mac OS X but we will show you how to set it up on Windows, too! Where there are differences, we will point them out.

What else do you need besides a computer? You need the Project Wonder frameworks, WebObjects and the WOLips Eclipse plugins. And our course you need Eclipse. Oh, and not to forget you probably want some kind of relational database.

We will set up our development environment together a bit later on. Let's first get some overview of things.

## 1.4 Hi-level overview of the frameworks

Some of you may not be really clear what we mean when talking about frameworks. So let's first set things straight and then have a look at what frameworks we have available.

### 1.4.1 What is a Framework?

Unlike a class library that is just a collection of classes, using a framework is a lot more. A ***framework*** not only contains classes but also resources like images, templates, other types of files, and even embedded other frameworks. The purpose of the classes in a framework is to provide a more or less defined set of functionality. Often you can use single classes from a framework for your purpose, but normally you use the functionality of the framework as a whole. Classes in the framework play together to do what they are supposed to do. In most cases, the control of what is happening lies with the framework and not with your application. Your application just provides classes and objects to the framework. The framework objects do the job with the help of your application objects and classes.

The framework being in control is very different to classic programming, where your application code controls everything and you just make use of objects and classes from a library. This is known as ***inversion of control*** or ***Hollywood principle*** (as the agent tells the actor: "Don't call us, we'll call you").

In a typical Project Wonder application your `public static void main(String[] args)` - method contains one line of code where it immediately hands over control to the Project Wonder frameworks

Here is a first look at some of the frameworks available.

### 1.4.2 WebObjects Frameworks

WebObjects is a set of many frameworks; many of them have been superseded by Project Wonder frameworks. For the sake of compatibility, the Project Wonder frameworks kept the names and packages the same. So even if things are still called WebObjects-something, many classes are not "WebObjects" any more. Anyway here is a list of some of the most important WebObjects frameworks:

| Framework | Purpose |
|---|---|
| `JavaFoundation` | Contains the basic classes used throughout WebObjects and Wonder |
| `JavaWOExtensions` | As its name says provides more classes and basic functionality |
| `JavaWebObjects` | Well, as its name implies, this is the framework for the "WebObjects" functionality (generation of html) |

| Framework | Purpose |
|---|---|
| JavaWOWebServicesClient,<br>JavaWOWebServicesGeneration,<br>JavaWOWebServicesSupport | A set of several frameworks that make it possible to provide and consume web services (JEE integration) |
| JavaJDBCAdaptor | Provides connectivity via JDBC to relational databases |
| JavaEOAccess,<br>JavaEOControl,<br>JavaEOGeneration<br>and many more | An extensive set of frameworks that implement the whole object-relational mapping. These frameworks are probably the most sophisticated ones in the whole Project Wonder/WebObjects universe. |
| JavaDirectToWeb,<br>JavaDTWGeneration | Direct to Web is a technology that allows you to create a database web application without any line of code. |

There are many more.

### 1.4.3 Project Wonder Frameworks

Here is a list of some of the frameworks making up Project Wonder:

| Framework | Purpose |
|---|---|
| ERExtensions | A core framework. It provides all the basic Project Wonder functionality |
| ERDirectToWeb | Basic enhancements to WebObjects Direct To Web technology |
| ERPrototypes | Allows database vendor independent specification of data types |
| ERJavaMail | Uses standard java mail for working with emails |
| ExcelGenerator,<br>ERExcelLook | Create real Microsoft Excel files from your data |
| ERPDFGeneration | Create PDF files |
| Ajax,<br>AjaxLook,<br>ERJQuery<br>and many more | Use those to create great Web 2.0 rich interface applications |

And many more. If you have a particular problem you need to solve, Project Wonder probably has a framework for you. Need Captchas? Use `ERCaptcha`. You want to support *OpenID* for login into your application? There is `EROpenID`. And so on.

# 2 Setting up your development environment

Let's assume we start with a more or less virgin installation of our development computer.

## 2.1 What do you need?

Let's look at the hardware as well as software requirements.

### 2.1.1 Hardware suitable for Project Wonder development

To be able to create great Project Wonder applications you need a decent computer. Any computer that can run at least Apple Mac OS X 10.6 (Snow Leopard) or Microsoft Windows Vista, and is equipped with at least 4GB of RAM, should be sufficient.

On the Apple platform any recent Mac Mini, Mac Pro, iMac, Mac Book Air, or Mac Book (Pro) would be a good system for development.

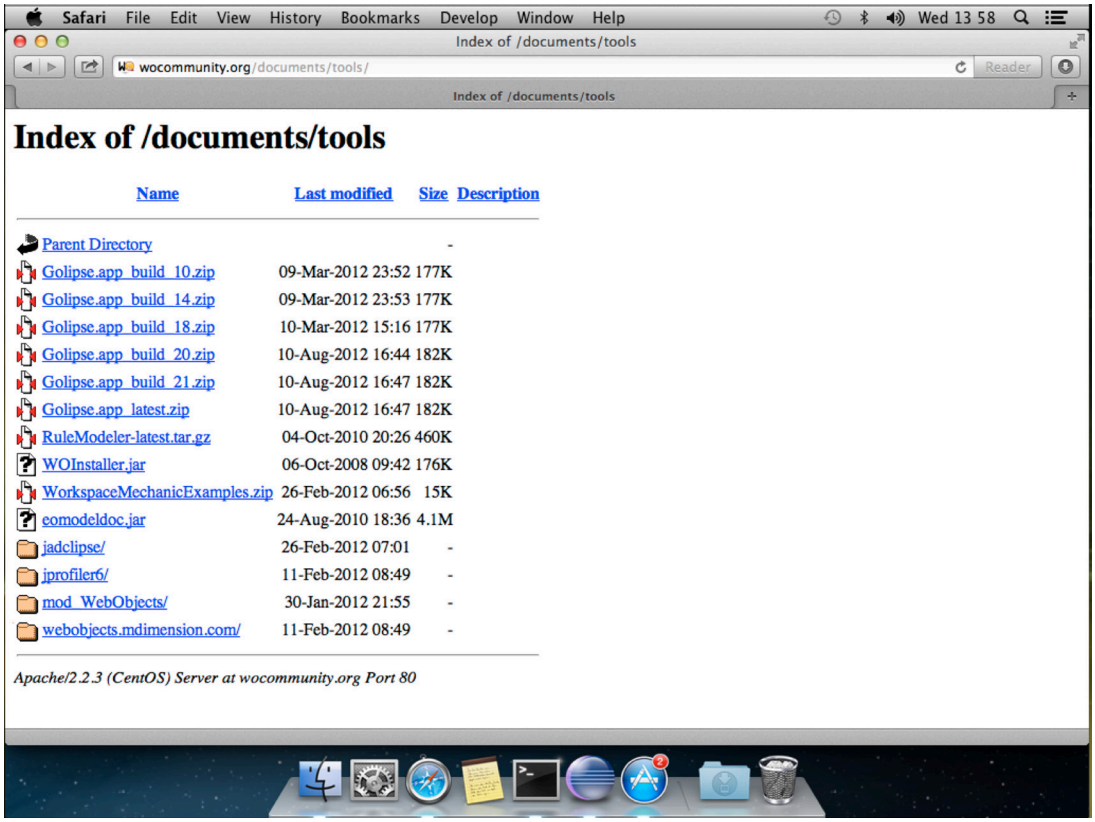
### 2.1.2 Software needed for Project Wonder development

If your operating system is at least Mac OS X Leopard or Windows Vista, you are fine. Development on Linux is possible as well, but as this is not so common, we will not cover it here. There is information available online.

You need a couple of frameworks and some tools, namely Eclipse and the WOLips plugin.

You can download installers for the tools and the frameworks. Open up your browser and surf to the following address *http://wocommunity.org/documents/tools/*

Download *WOInstaller.jar* and *Golipse.app_latest.zip*. You might want to bookmark this page and come back later to download more interesting things. For the moment *WOInstaller* and *Golipse* is what we need.




*Picture 2-1 Download site for the installers*

Of course you need a relational database server. This can be any of the standard products like MySQL, Postgres, Oracle, and others. As long as you can get a JDBC driver for the database server, you should be all set. Install the database server on your development machine or on any other system; just make sure you have network access from your development system to the database. Throughout this book we will work with a locally installed MySQL database server. However where needed, we will give you tips for other products as well.

## 2.2 Where goes what?

On a Mac there are two main locations where frameworks are stored. */System/Library* is reserved for all Apple system software. The original WebObjects installation put all the frameworks into that location. This is not a good idea today. Leave */System* alone! Some tools and additional things got installed into */Library*. This is ok but we want full control over our installation (and the modern Project Wonder tools support this). So we are going to install into a completely different location. The following is our suggestion. You can choose whatever other location suits your style.

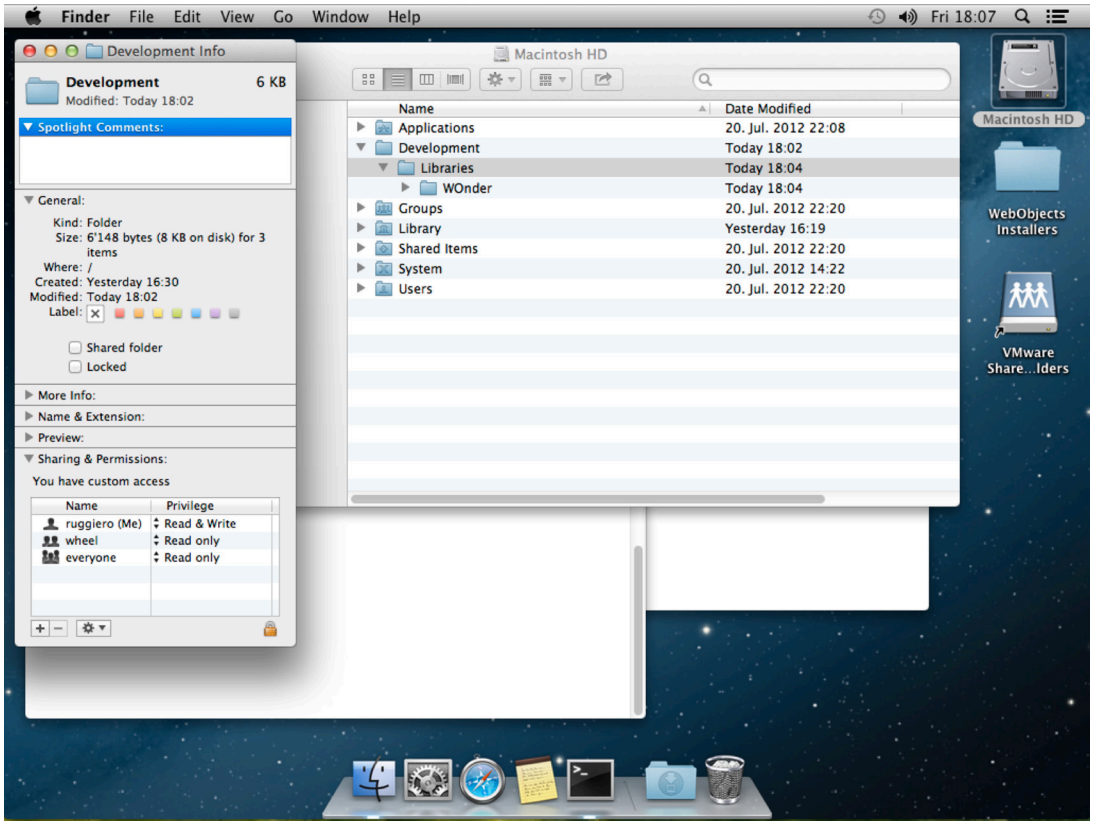
We create a Development-folder on the root level of our hard drive. Call it *Development*. On Windows this would normally be on the *C:* drive. You may need admin privileges to create the folder. Change the access rights of the *Development* folder so that it belongs to you.

***Caution***

*Please avoid spaces or international characters in file and path names. Some tools and frameworks might get upset otherwise and things might not work properly.*

Create another folder named *Libraries* inside the *Development* folder and inside this yet another one called *WOnder*. This will be the location where all the frameworks are going to end up.

Here is a screen shot from a Mac showing */Development/Libraries/WOnder*:




*Picture 2-2 The Development folder with its substructure*

On a Windows system you would have *C:\Development\Libraries\WOnder* (do you really need a screen shot from a Windows system, too?)

The basic idea is that everything goes inside this */Development* folder. At the end the */Development* folder will contain at least four subfolders:

*/Development/Libraries*
: This folder will contain all the compiled frameworks

*/Development/WonderSource*
: This folder will contain the source code to the frameworks.

*/Development/Tools*
: This is the location where all the tools will reside (e.g. Eclipse)

*/Development/workspace*
: This is your Eclipse workspace.

Of course you are free to choose another layout. However throughout this book we will work with this folder structure.

## 2.3 Installing the tools

You need Eclipse and the WOLips plugins. *Golipse* is a Mac application that downloads Eclipse and WOLips and configures everything for optimal Project Wonder development. If you are installing the development environment on Windows you, cannot use *Golipse*. Instead you must download and install Eclipse und WOLips manually.

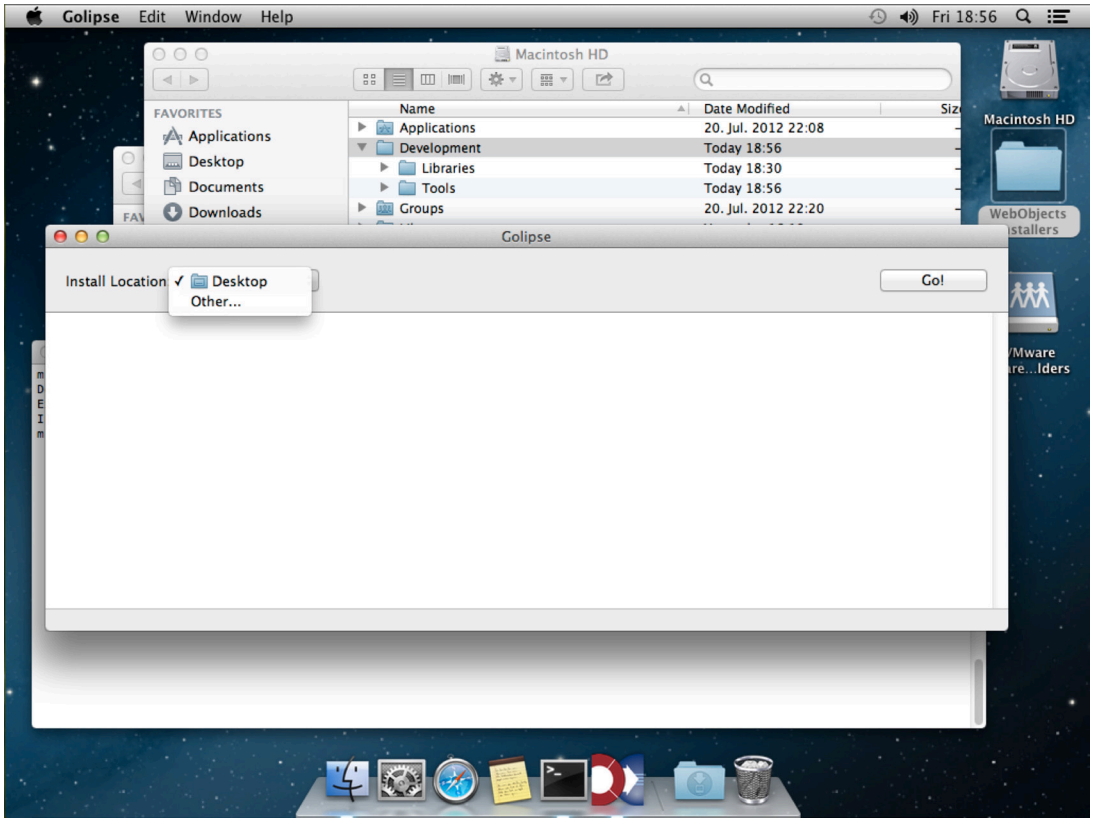
Manual installation would also be what you want if you already have Eclipse on your system and just need the WOLips plugin. We'll cover manual installation a bit further down.

### 2.3.1 Automatic installation of Eclipse and WOLips

Automatic installation will only work on Mac OS X.

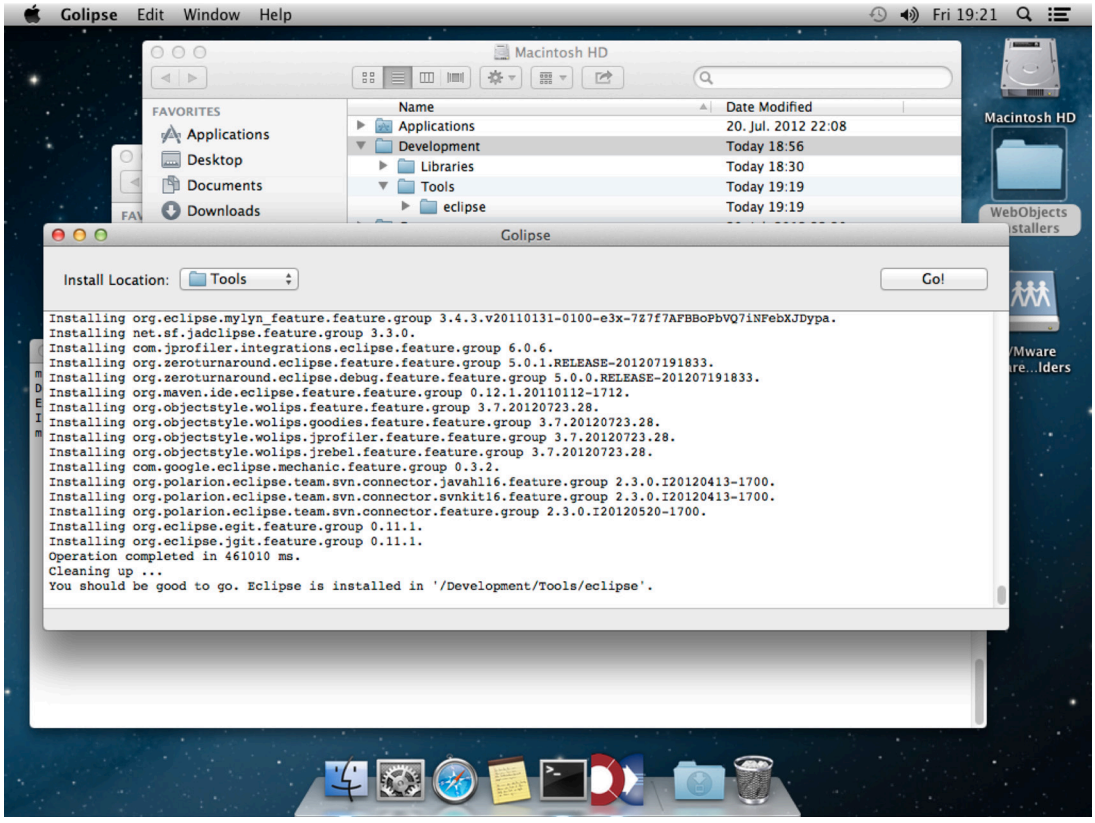
Unpack *Golipse.app_latest.zip* if not yet done and start *Golipse* by double clicking its icon.

From the popup select where to put Eclipse. You can select any location. Create a *Tools* folder inside your */Development* folder and select this new folder for the install location of *Golipse*.




*Picture 2-3 Golipse.app - Select install location for the tools*

Press the *GO!* button and have some coffee while Eclipse and WOLips are being downloaded and installed.



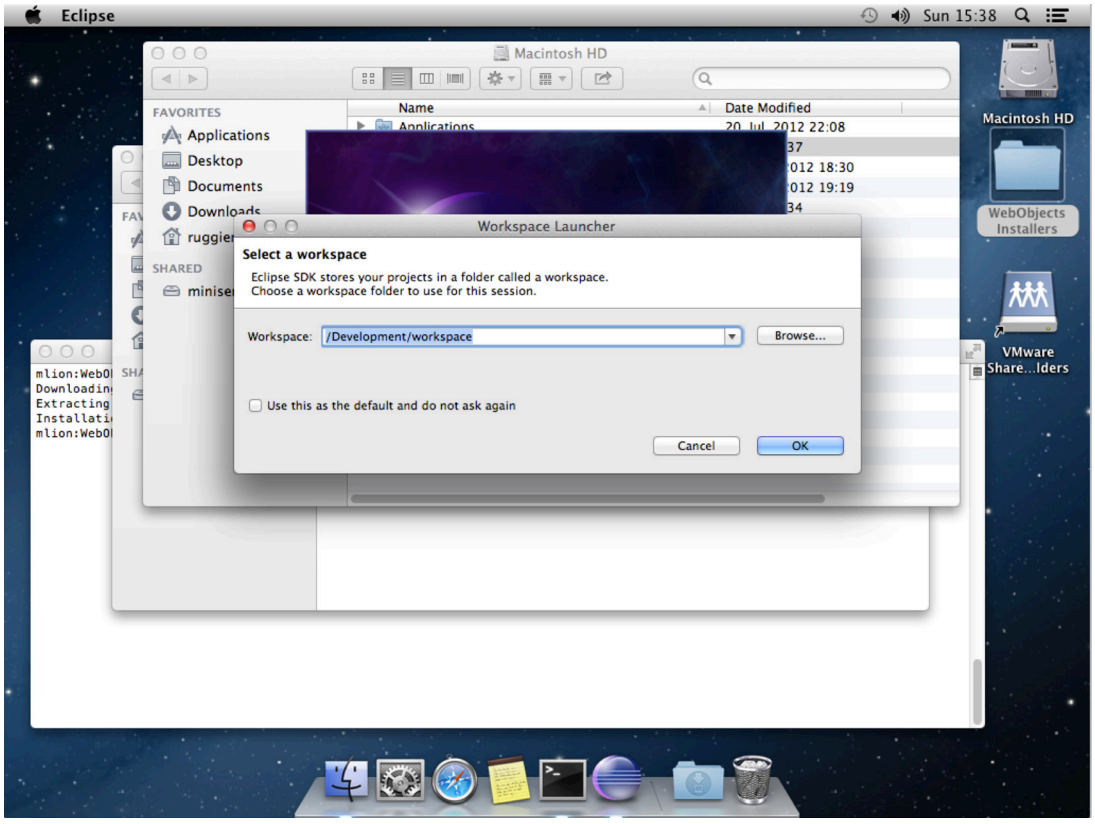

*Picture 2-4 Golipse.app - The tools are installed*

As you can see from the summary lines at the bottom of the log output, the download and installation took quite some time, but all is well now.

### 2.3.2 Manual installation of Eclipse and WOLips

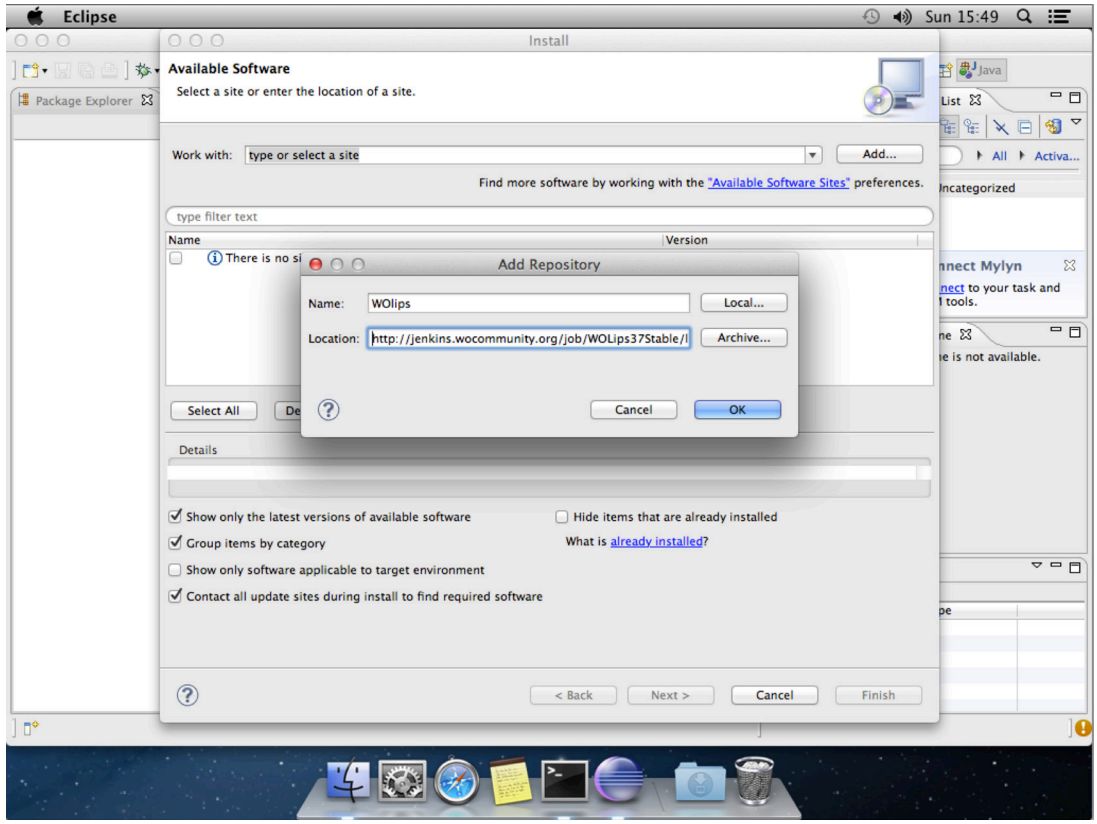
This is needed on Windows or if you already have an Eclipse IDE that you want to use. Go to *http://www.eclipse.org* and download Eclipse. As of the writing of this book Eclipse 3.7 was the latest stable version. Eclipse 4.2 was available but the WOLips tools were not yet ready. This might be different when you read this book. Important thing is that Eclipse and WOLips go together. So for Eclipse 3.7 you need WOLips 3.7, Eclipse 4.2 would then require WOLips 4.2.

Open your copy of Eclipse and go to *Help -> Install New Software*. When you start Eclipse it might ask which workspace to use. Go with the default or pick any empty directory. In our case we will create a workspace-folder inside */Development*.




*Picture 2-5 Eclipse creating a new Workspace*

When Eclipse is up and running, open the *HELP* Menu and select *INSTALL NEW SOFTWARE*. Add a new site and name it. You can name it anyway you like, but it makes a lot of sense naming things according to their meaning. Thus we name it *WOLips*.



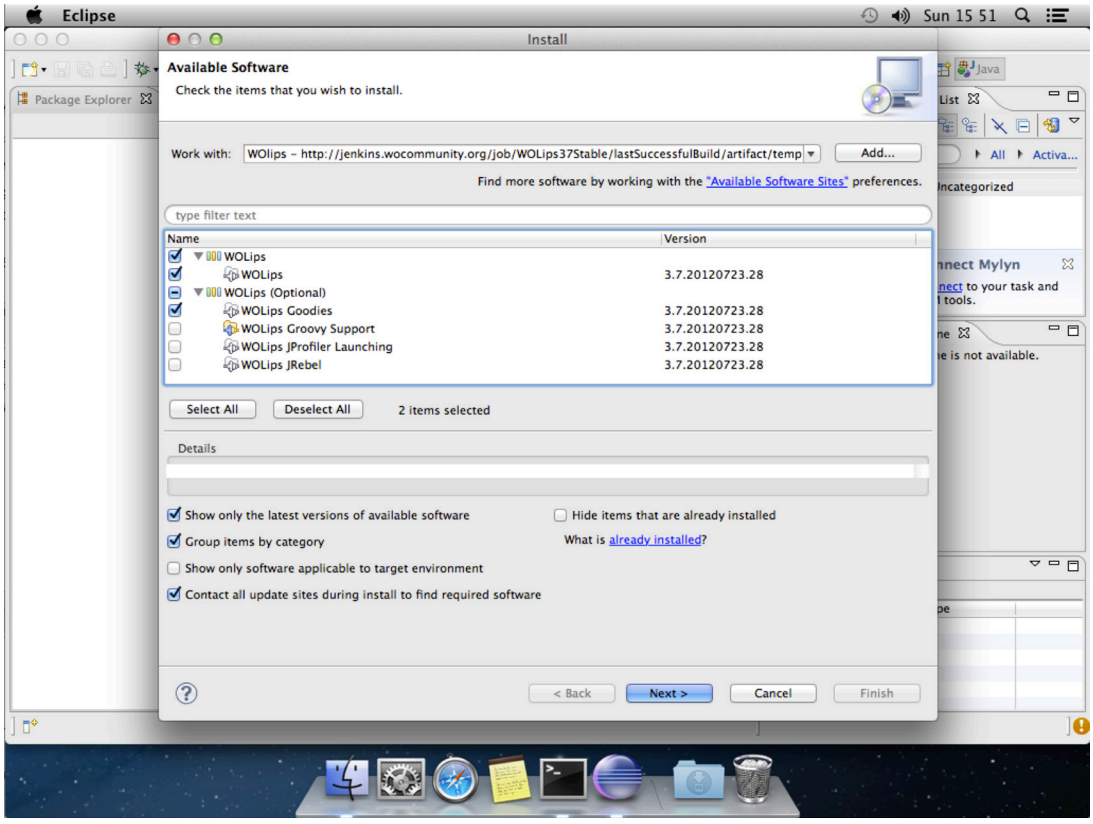

*Picture 2-6 Create a new software site*

Put the following URL into the location field:

*http://jenkins.wocommunity.org/job/WOLips37Stable/lastSuccessfulBuild/artifact/temp/dist*

Then press *OK*.

After a moment you will be presented with the following screen. If instead you get an error message you have probably mistyped the location URL.




*Picture 2-7 Select the parts of WOLips to install*

For the moment check *WOLIPS* and *WOLIPS (OPTIONAL)* -> *WOLIPS GOODIES*. Then press *NEXT>*. Confirm the installation, accept the license, and after some moments Eclipse is ready for restart.

## 2.4 Installing the frameworks

There are several frameworks to install for WebObjects and Project Wonder. WebObjects must be downloaded from Apple's site, Project Wonder is available from the community site. We can make use of an installer or install things manually. If things don't work the way you expect, it is sometimes necessary to manually fix the installation.

The Project Wonder frameworks are distributed in binary form as well as in source code for your own compilation. We will look into both ways to install Project Wonder.

### 2.4.1 Installing WebObjects

For the installation of WebObjects you need *WOInstaller.jar* which you hopefully already downloaded from *http://wocommunity.org/documents/tools/*.

Open a Terminal window and cd to your preferred download location. When you run *WOInstaller.jar* from the command line without any options, you will get the following output:

```
$ java -jar WOInstaller.jar
usage: java -jar WOInstaller.jar [5.3.3|5.4.3] [destinationFolder]

Example:
WO 5.3.3 on Windows
       java -jar WOInstaller.jar 5.3.3 C:\Apple

WO 5.3.3 on OS X (in alternate folder)
       java -jar WOInstaller.jar 5.3.3 /opt
$
```

Enter the following command (substitute your folder path). Note that we are going to download WebObject Version 5.4.3. This is the latest official set of frameworks. On a Macintosh the command looks like this:

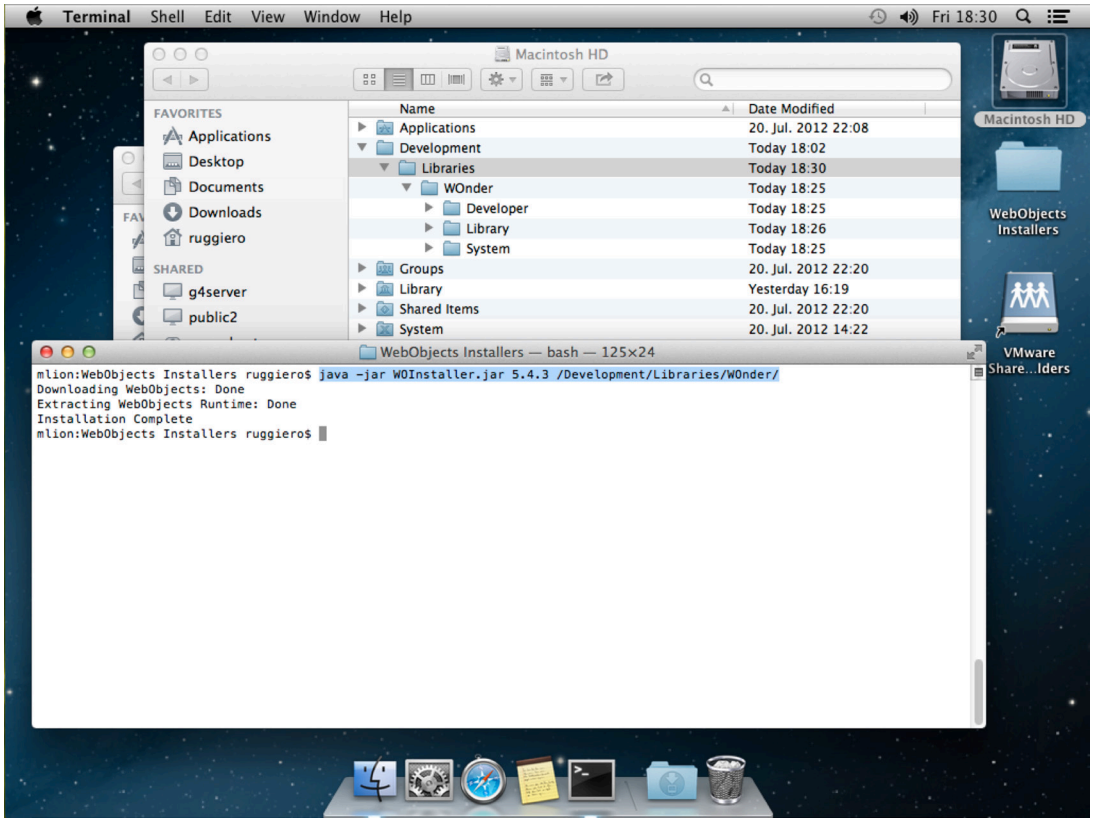
```
$ java -jar WOInstaller.jar 5.4.3 /Development/Libraries/WOnder/
```

On a Windows system substitute the proper Windows path:

```
DOS> java -jar WOInstaller.jar 5.4.3 C:\Development\Libraries\WOnder\
```

The program will tell you what it's doing, and there will be some progress output. Be patient, as this will take a couple minutes.

Here are the results:




*Picture 2-8 WebObjects is installed*

The installation has created sub-folders underneath */Development/Libraries/WOnder*. Feel free to explore the newly installed stuff.

Next comes the installation of Project Wonder.

### 2.4.2 Installing Project Wonder

Project Wonder is open source and you can install the frameworks either as a binary download or build them from source. Both ways are ok. Just pick what you feel better with. However we strongly recommend a source installation. Having the source and being able to debug right into it is an invaluable thing for learning.

#### Installing from source code

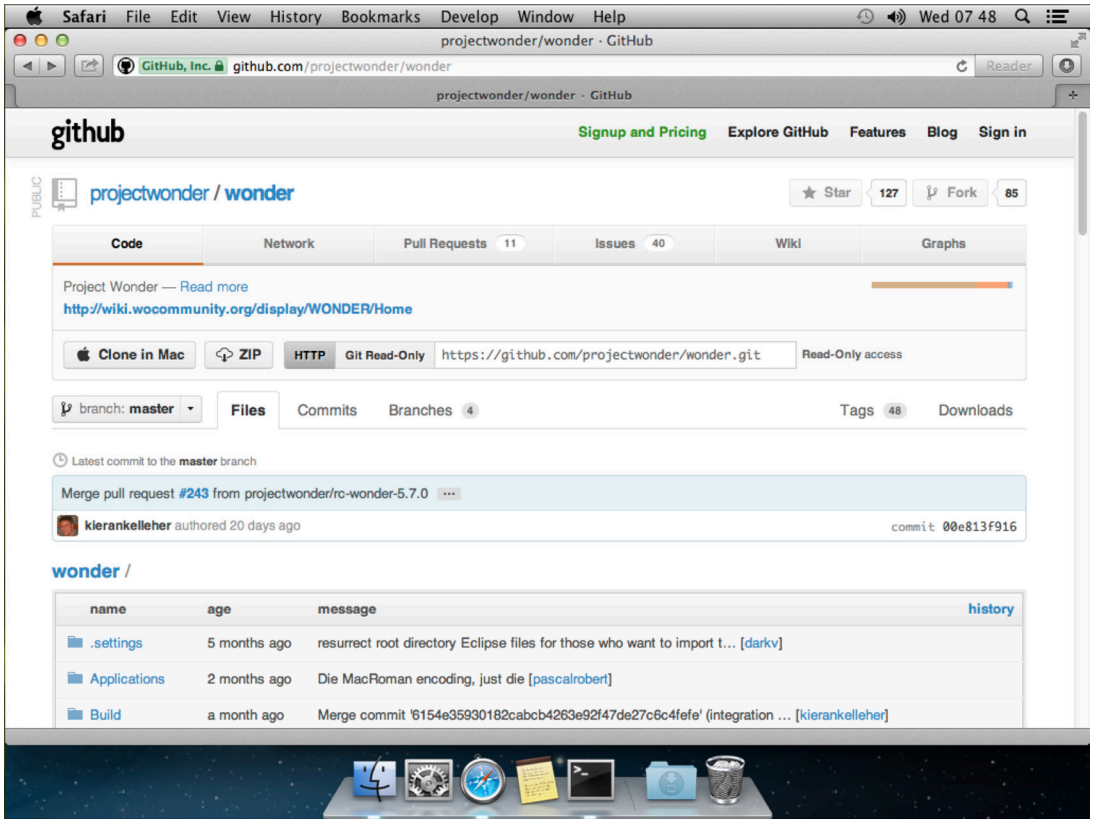
The source code for the Project Wonder frameworks is maintained in a public repository on *github*[1]. There are several command line and GUI tools available to access *git* repositories. An out-of-the-box Mac OS X system does not have a `git` command. You could download the developer tools from Apple's site, you could use *MacPorts* or any other package manager to get *git* or you can search for a

[1] Github is a large public repository for open source projects. It is based on the git version control system.

GUI-tool. There is also the possibility to simply download the complete source packaged as a *.zip*-file. In this tutorial for simplicity we use the downloadable *.zip* distribution.

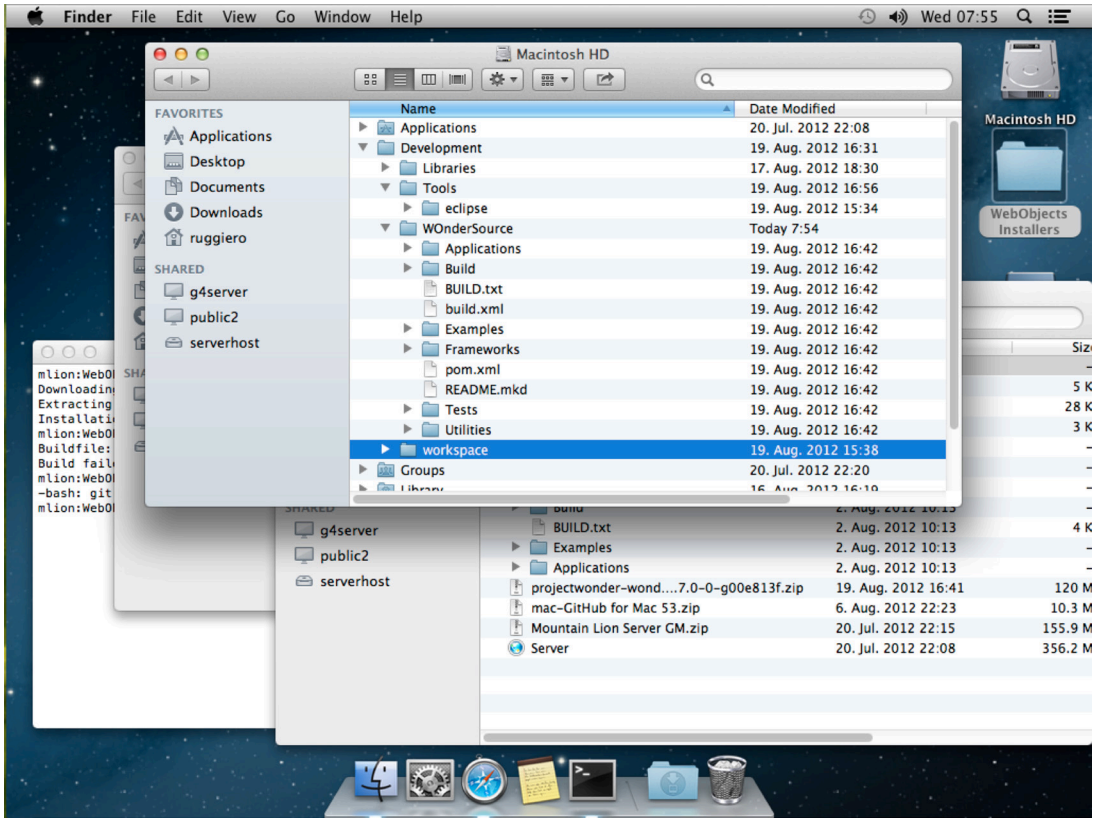
Open your browser and enter the following address (note the *https* protocol):

*https://github.com/projectwonder/wonder*




*Picture 2-9 Project Wonder on github*

Download the zipped source by clicking on the *ZIP* button. Unpack the file. Create another directory inside our */Development* folder and call it *WOnderSource*. Put the contents of the unzipped folder into this new directory.



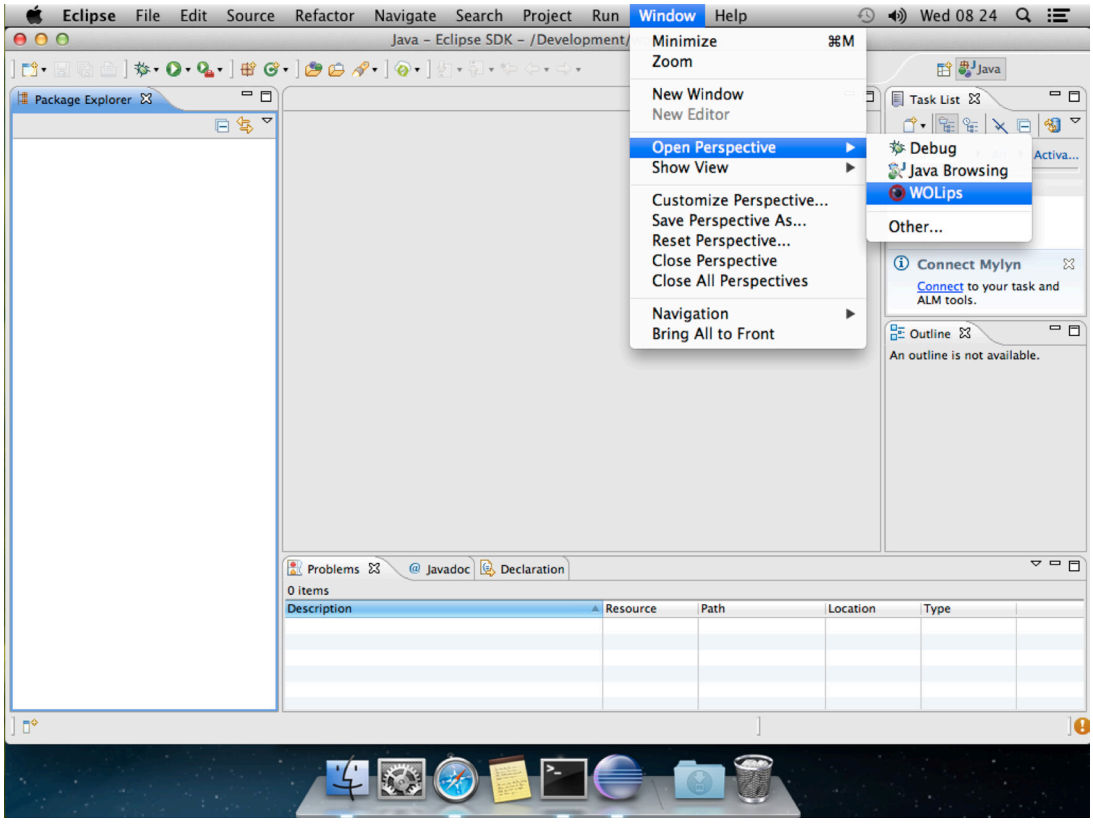

*Picture 2-10 Project Wonder source downloaded and unpacked*

### The wolips.properties file

There is one important file we need to create. This file specifies the correct locations for the built frameworks. You remember? We have created */Development/Libraries/WOnder*. That's where we want all the stuff to be installed. Now we need to tell WOLips where things are. To make life easy for us WOLips creates a default configuration file when you create your first application in WOLips. We are going to use WOLips to create such a default file and will then build upon that.

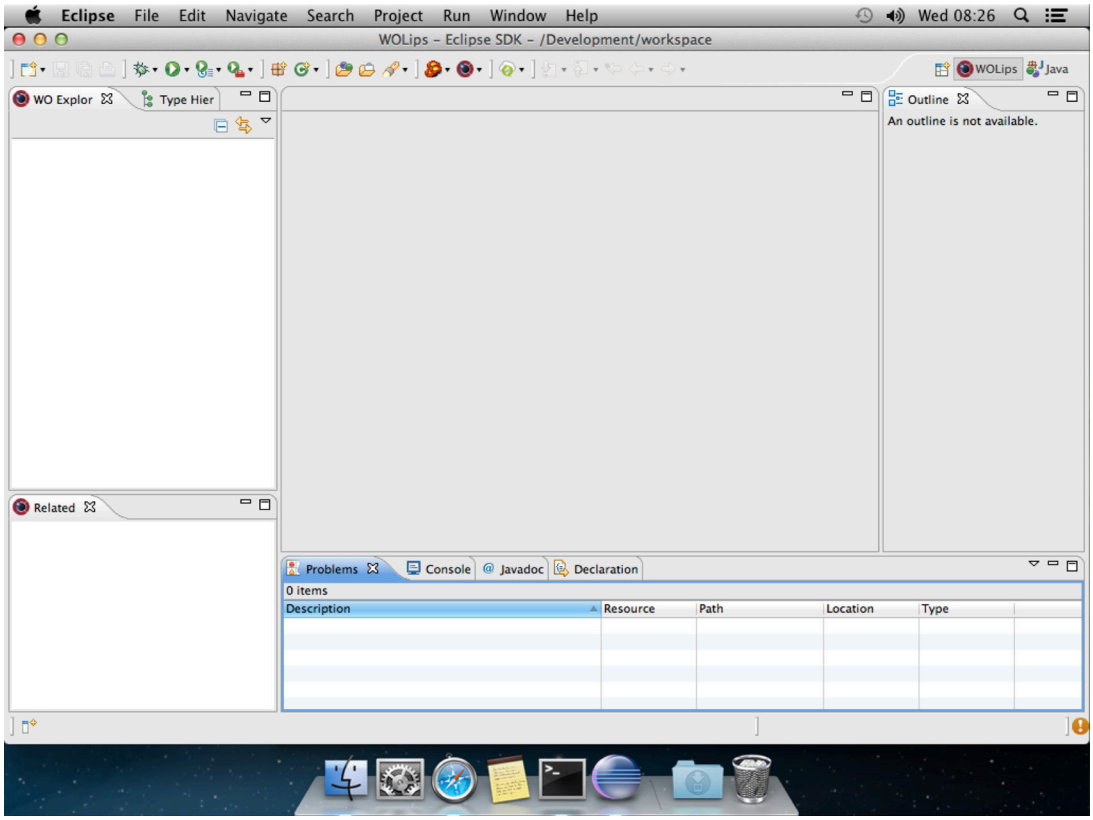
Follow the next steps. We are not particularly concerned with things. Later on we are going to look at what this all means. For now, just follow on.

Start Eclipse again, pointing it to the correct workspace when asked. Then open the menu *Window -> Open Perspective -> WOLips*.



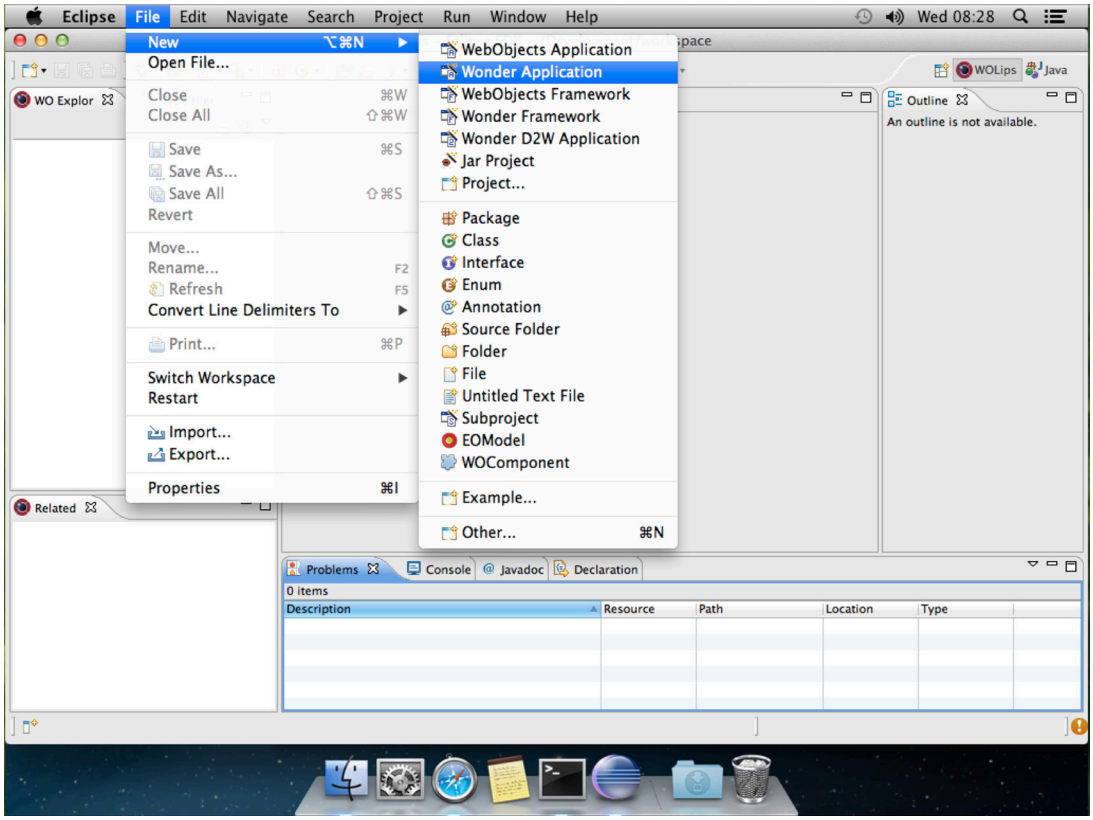

*Picture 2-11 Open WOLips perspective*

This will open the WOLips perspective, which will look something like this:




*Picture 2-12 The WOLips perspective in Eclipse*

Create a new Project Wonder application by going to menu *File -> New -> Wonder Application*.




*Picture 2-13 Create new Wonder application*

In the following dialog give that new application a name like *"BasicConcepts"* and click finish. This will create a project and (more importantly here) the default configuration file for WOLips.

At the moment, we are not interested in the project but will come back to it a bit later. *BasicConcepts* will serve as the project to explain all the basic concepts of a Project Wonder application (hence the name for this project). And as the problem report view in Eclipse tells you, the application cannot build anyway due to a handful of errors. We ignore the errors for the moment and terminate Eclipse.

When we created that first Project Wonder application WOLips created the default *wolips.properties* file. You will find this file in the following location (those directories might be hidden by your operating system's file browser, Finder or Explorer, you know how to make them visible, don't you?)

Mac OS X */Users/ruggiero/Library/Application Support/WOLips*

Windows 7 *C:\Users\ruggiero\AppData\Roaming\WOLips*

These are locations inside your home folder. Substitute your user name accordingly instead of *ruggiero*.

Here is the content of the default ***wolips.properties*** on a Mac OS X system

File: *wolips.properties* on Mac OS X

```
wo.api.root=/Developer/Documentation/DocSets/↵
com.apple.ADC_Reference_Library.WebObjectsReference.docset/Contents/Resources/↵
Documents/documentation/InternetWeb/Reference/WO542Reference
wo.apps.root=/Library/WebObjects/Applications
wo.bootstrapjar=/System/Library/WebObjects/JavaApplications/wotaskd.woa/↵
WOBootstrap.jar
wo.extensions=/Library/WebObjects/Extensions
wo.local.frameworks=/Library/Frameworks
wo.local.root=/
wo.network.frameworks=/Network/Library/Frameworks
wo.network.root=/Network
wo.system.frameworks=/System/Library/Frameworks
wo.system.root=/System
wo.user.frameworks=/Users/ruggiero/Library/Frameworks
wo.user.root=/Users/ruggiero
```

Note that all lines start with `wo`. Some lines are rather long and break over several print lines. Don't let this confuse you! The same remark applies to the *wolips.properties* on Windows:

File: *wolips.properties* on Windows

```
wo.api.root=/Developer/ADC%20Reference%20Library/documentation/WebObjects/↵
Reference/API
wo.apps.root=C\:\\Apple\\Library\\WebObjects\\Applications
wo.bootstrapjar=C\:\\Apple\\Library\\WebObjects\\JavaApplications\\wotaskd.woa\↵
\WOBootstrap.jar
wo.dir.local.library.frameworks=C\:\\Apple\\Local\\Library\\Frameworks
wo.dir.user.home.library.frameworks=C\:\\Users\\ruggiero\\Library\\Frameworks
wo.extensions=C\:\\Apple\\Local\\Library\\WebObjects\\Extensions
wo.local.frameworks=C\:\\Apple\\Local\\Library\\Frameworks
wo.local.root=C\:\\Apple\\Library\\Local
wo.network.frameworks=C\:\\Network\\Library\\Frameworks
wo.network.root=C\:\\Network
wo.system.frameworks=C\:\\Apple\\Library\\Frameworks
wo.system.root=C\:\\Apple\\Library
wo.user.frameworks=C\:\\Users\\ruggiero\\Library\\Frameworks
wo.user.root=C\:\\Users\\ruggiero
wolips.properties=wolips.properties
```

There is actually only one real difference between the Mac version and the Windows version. As on Windows the backslash and the colon are used in path names they must be properly escaped because the properties parser would not work. The escape character is the \.

**Editing wolips.properties**

As you can easily see, there are various file system paths noted in *wolips.properties*. We have to adapt those to our environment. Simply open *wolips.properties* with a text editor and make the following changes:

Change all paths starting with either */Library* or */System/Library* so that they read */Development/Libraries/WOnder/Library* and */Development/Libraries/WOnder/System/Library* (basically replace the initial "/" by "*/Development/Libraries/WOnder/*")

For a Windows *wolips.properties* replace "`C\:\\Apple\\`" by "`C\:\\Development\\`".

Of course this assumes you laid out your directory structure like we did. Adapt the paths to your structure if needed.

***Note***

*Not all paths in wolips.properties are really needed. E.g. wo.api.root looks weird and particularly on Windows makes no sense at all. We will come back later and fix things.*

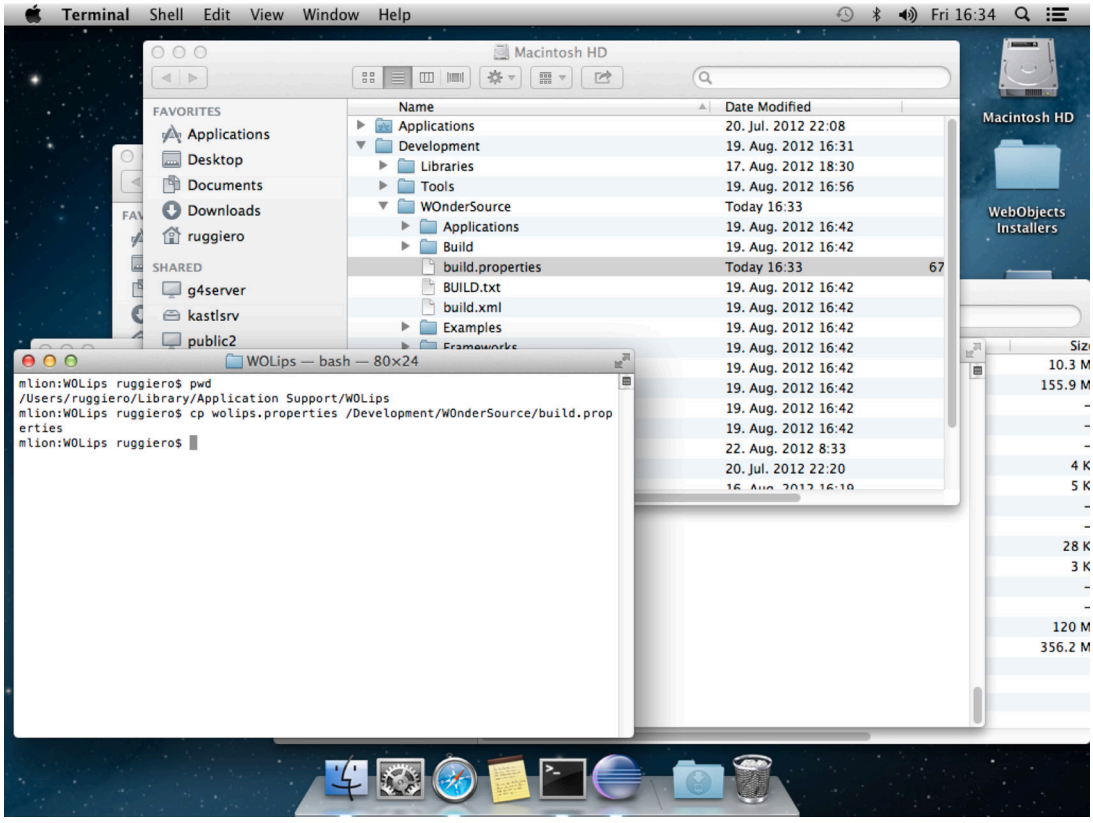
This is how your *wolips.properties* looks like when you followed our suggested folder layout (Mac version shown):

File: fixed *wolips.properties* on Mac OS X

```
wo.api.root=/Developer/Documentation/DocSets/↵
com.apple.ADC_Reference_Library.WebObjectsReference.docset/Contents/Resources/↵
Documents/documentation/InternetWeb/Reference/WO542Reference
wo.apps.root=/Development/Libraries/WOnder/Library/WebObjects/Applications
wo.bootstrapjar=/Development/Libraries/WOnder/System/Library/WebObjects/↵
JavaApplications/wotaskd.woa/WOBootstrap.jar
wo.extensions=/Development/Libraries/WOnder/Library/WebObjects/Extensions
wo.local.frameworks=/Development/Libraries/WOnder/Library/Frameworks
wo.local.root=/Development/Libraries/WOnder/
wo.network.frameworks=/Network/Library/Frameworks
wo.network.root=/Network
wo.system.frameworks=/Development/Libraries/WOnder/System/Library/Frameworks
wo.system.root=/Development/Libraries/WOnder/System
wo.user.frameworks=/Users/ruggiero/Library/Frameworks
wo.user.root=/Users/ruggiero
```

### Compiling the frameworks

Now we want to compile the Project Wonder frameworks. Where should the compiled frameworks go? Of course they should be put in a location where WOLips can find them - and this information is in your *wolips.properties* file. So make a copy of your *wolips.properties* file and put it into the folder where the Project Wonder source is. Rename the copied file to *build.properties*. The build-process will read this file and know where to put things.




*Picture 2-14 build.properties in your WOnderSource directory*

Let's build the frameworks. For this we need to make use of the ***ant*** tool. If ant is not yet installed on your computer you can go to *http://ant.apache.org* and download it from there. For details consult the online installation instructions at *http://ant.apache.org/manual/index.html*.

Open a terminal window (or a DOS box) and cd to the Project Wonder source directory. Issue the following command:

```
$ ant frameworks
```

The compilation process can take a minute or two. You will see messages fly by, some may be warnings but there should not be any errors. At the end, there should be a success message and the total time used.

When something does not work, ant will display an error message. A typical problem is that the compile process cannot find certain needed classes. Check the paths in your *build.properties* file. Most probably something is not correct there.

If you get a message telling you that ant cannot be found or is an unknown command or something else is wrong with ant itself, check if you have properly installed ant and the command is in your *PATH*.

On windows you may see messages flying by telling you something about un-mapable characters. This happens because some Java files contain comments in Japanese. You can safely ignore these warnings, the code compiles fine anyway.

The frameworks are now compiled, but they are not yet installed in the correct location. Issue the following command:

```
$ sudo ant frameworks.install
```

The install command must be run as super user. It needs the elevated privileges for a proper installation. On Windows run it from an admin account or open a console with admin privileges. No need for `sudo` there.

Hopefully you will be greeted by a *BUILD SUCCESSFUL* message.

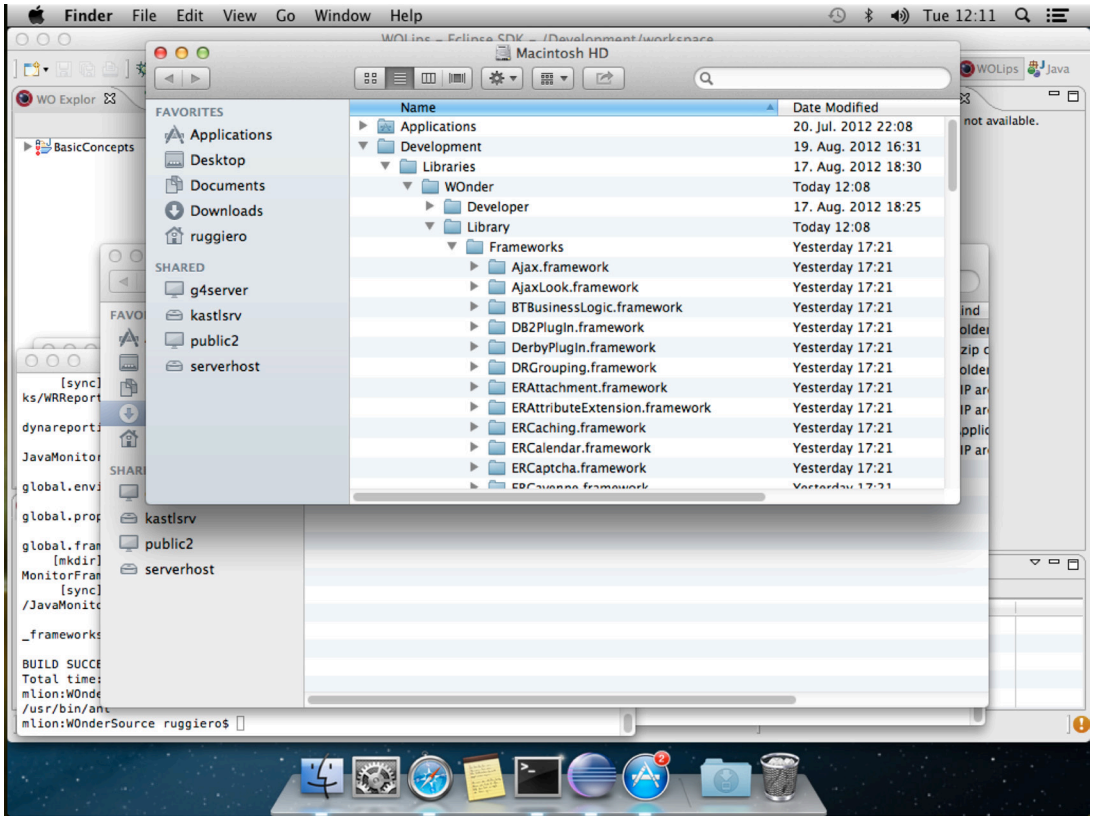
### Installing the binary frameworks

There is an archive available of all the Project Wonder frameworks ready built. Download it from *http://jenkins.wocommunity.org/job/Wonder/lastSuccessfulBuild/artifact/Root/Roots/Wonder-Frameworks.tar.gz*.

Use any available tool to unpack the archive. On Mac OS X double clicking the downloaded file in the Finder should be sufficient, on Windows you can use e.g. *WinZip* or *WinRar*. Of course the file can also be expanded from the command line in a terminal window with the following command:

```
$ tar -xzvf Wonder-Frameworks.tar.gz
```

You get a folder named *Wonder-Framework* that contains all the frameworks. Move the frameworks (not the folder itself) to */Development/Libraries/WOnder/Library/Frameworks* (or on Windows *C:\Development\Libraries\WOnder\Library\Frameworks*). Here is a screen shot that shows where things should be:



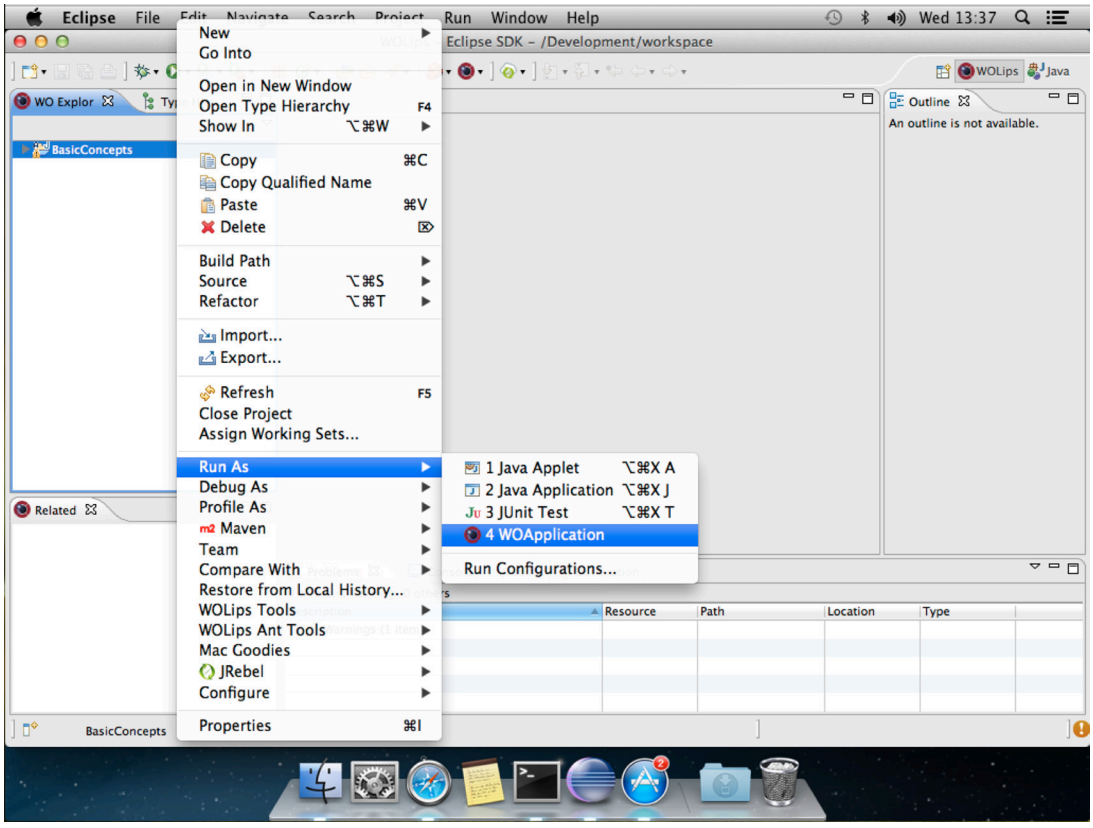

*Picture 2-15 Installed Project Wonder Frameworks*

That's it; you are done!

### 2.4.3 Final test if everything is installed properly

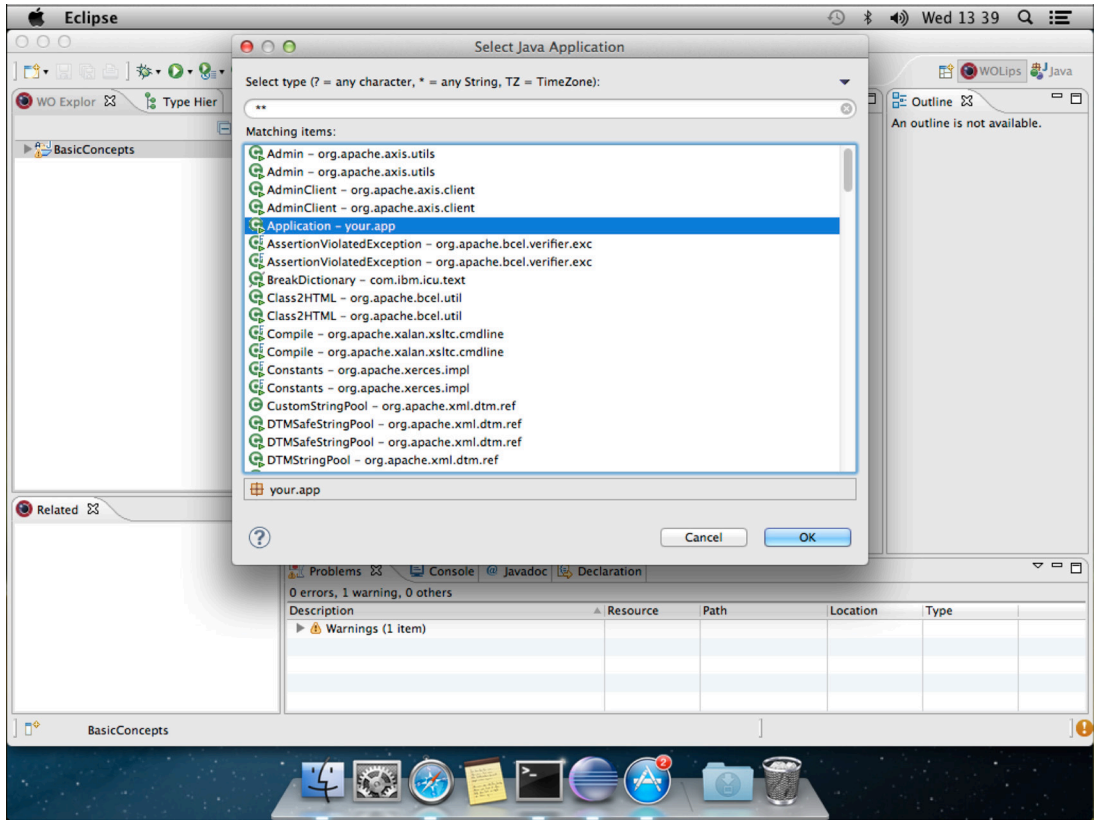
Open Eclipse again. You should still have that project *BasicConcepts*. This project did have a couple of build errors. These should now be gone.

You can run the application by selecting from the context menu *RUN AS -> WOAPPLICATION*



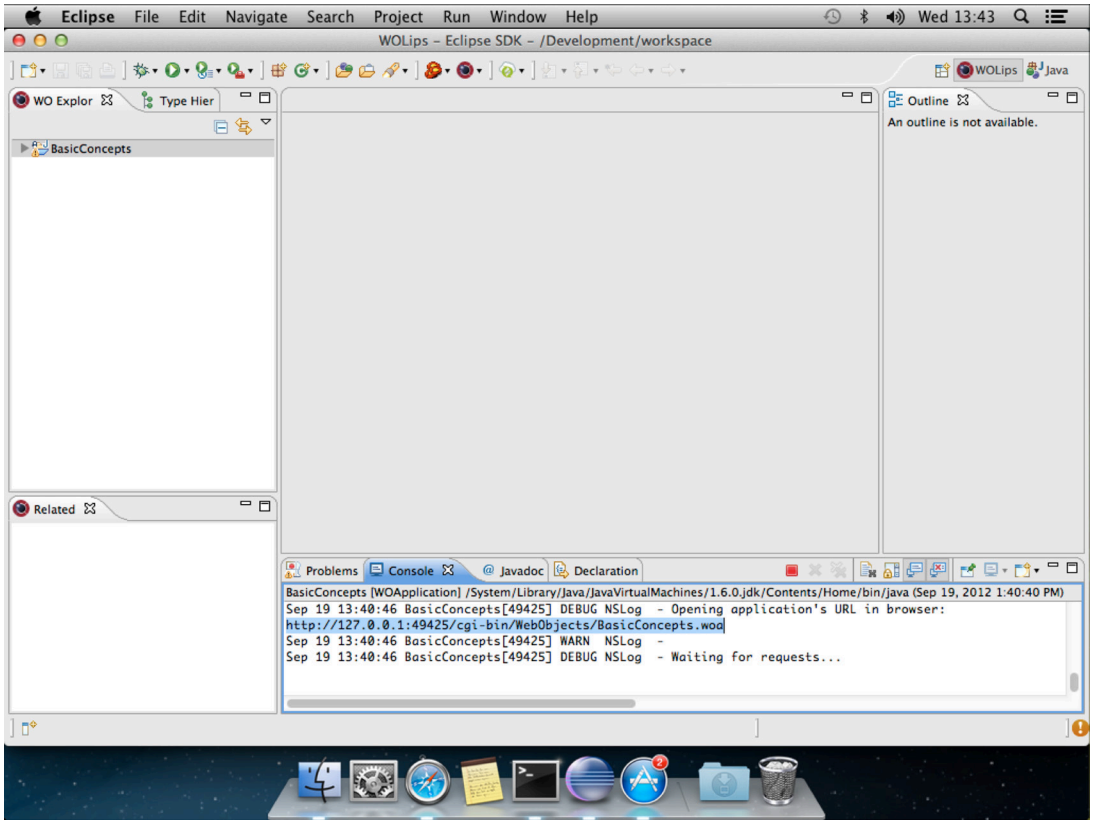

*Picture 2-16 Run the Application*

Eclipse will search for classes containing `main()` methods. Pick `Application` from package `your.app` and hit *OK*.




*Picture 2-17 Pick the right main class*

After a moment your default browser should open and display the famous greeting to the world. If you do this on a Windows system, you must open the browser manually. Look in the Console window in Eclipse. At the very end of the startup messages you should see a line telling you what the URL of your application is. Copy this to your browser, hit enter, and the greeting should appear.




*Picture 2-18 Startup messages and URL for the application*

What if you get an error message instead of a nice running application? There is one typical problem that can show up:

You get an error message telling you that the main method cannot be found because the class *Application* in package your.app is not known.

That is easy to fix! Remember, you created the *BasicConcepts* project without a proper *wolips.properties* file. So WOLips configured your project with default paths for the frameworks. Now we have the frameworks and the *wolips.properties* file has the correct paths, but your project is still wrong. Go into *PROJECT* menu and *CLEAN…* the project. This should rebuild everything based on the correct properties.

# 3 Where to get help

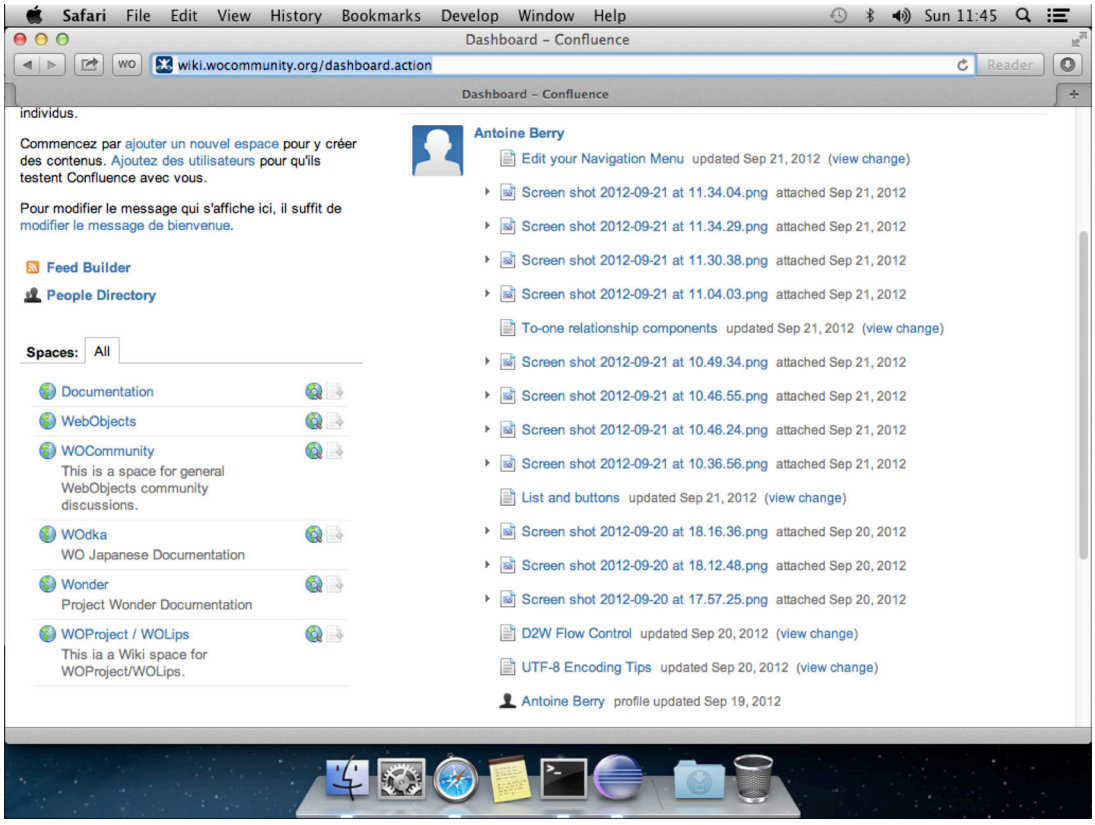
Sometimes things do not work as expected, sometimes you need some hand holding (yes, even seasoned programmers need hand holding from time to time), oftentimes you want to know more about a class, or how to do certain things, or how to use a framework.

Here are the most important resources:

### The Wonder Wiki

*http://wiki.wocommunity.org/dashboard.action*

This is your starting point for everything Wonder related. Wonder has a very lively community and the wiki provides all the information you need.




*Picture 3-1 The wiki entry point for everything Wonder related*

### WebObjects classes and API documentation

*http://api.webobjects.me/wo542/*

The documentation has not changed significantly from version 5.4.2 to the current one 5.4.3, so it was not updated.

### Wonder frameworks, classes and API documentation

*http://jenkins.wocommunity.org/job/Wonder/lastSuccessfulBuild/javadoc/?*

### Mailing list

There are several mailing lists available related to Wonder and WebObjects. Apple hosts the most important one. You can find it here: *https://lists.apple.com/mailman/listinfo/webobjects-dev*.

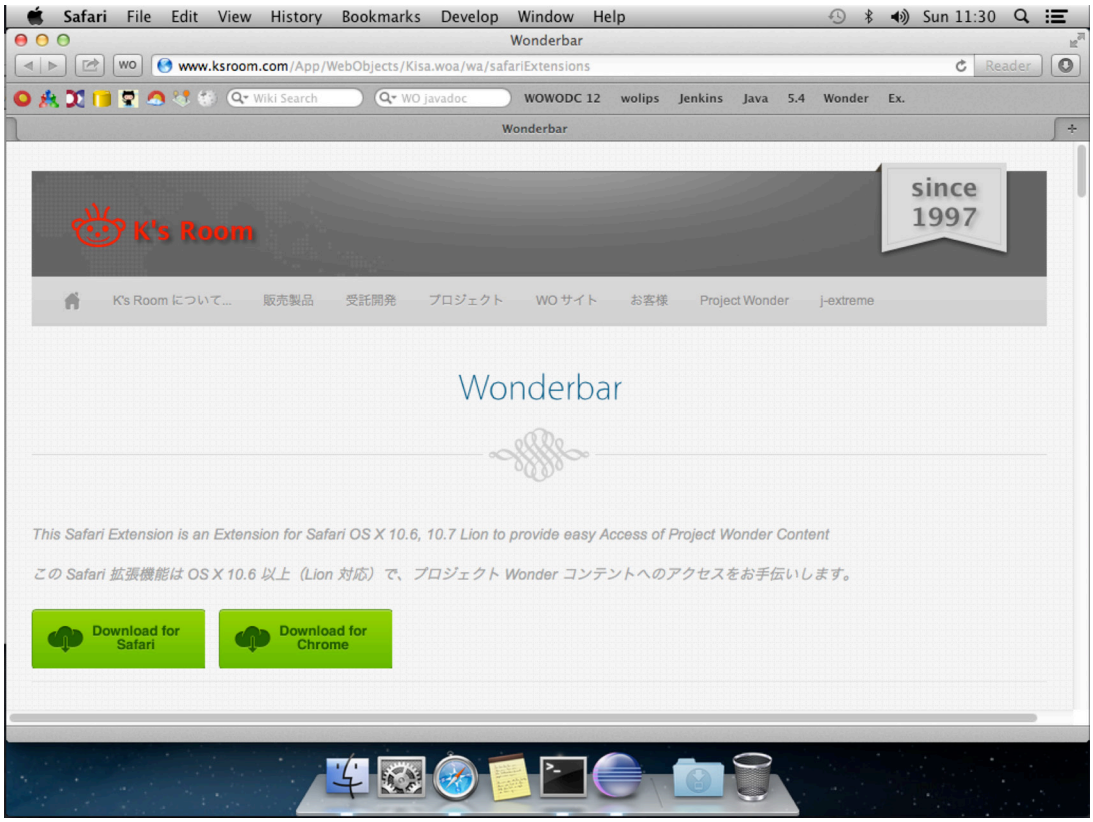
Ask questions, post tips, and maybe even give answers! All the Wonder "gurus" and all the ordinary folks hang out there. Do not be afraid of asking "stupid" questions. Never forget

The most stupid question is the one you have and dare not ask!

All the members on this list are more than happy to help you. We want you! And we can only get you on board by helping you!

### Wonderbar

This is an extension for Apple's Safari and Google Chrome from Ken Ishimoto. Go to *http://www.ksroom.com/App/WebObjects/Kisa.woa/wa/safariExtensions* and download ***Wonderbar***. It installs right into your browser and gives you direct access to most all online resources. Here is *Wonderbar* in action:




*Picture 3-2 Wonderbar installed in Safari*

You will probably use the following resources the most:

- Java API Documentation
- WebObjects Classes and API Documentation
- Wonder Classes and API Documentation