Learning WebObjects

This document is written for someone who wants to start developing with WebObjects and has no prior experience with it. It details:

1. What WebObjects is and does
2. How to learn the concepts necessary to code with WebObjects
3. How to get started with development

What WebObjects Is

WebObjects is a server side development environment that allows you (a developer) to write a database driven web based application by creating Java objects and working with them. You can bind the properties of an object directly to user interface elements like text, form fields, buttons, and drop down boxes. WebObjects hides the complexities of web browsers from you, and also that the data store is a database. WebObjects handles creating and managing user sessions without the use of cookies, and fetches and saves your objects to and from a database without your needing to write SQL or tracking changes. That also means it is easy to switch databases after an application is written. In addition, WebObjects can be used to write a Web Service without a user interface. You can use a Web Service to interface with other web application servers or with a JavaScript (AJAX) application.

WebObjects includes a rapid prototyping tool called Direct to Web. You can graphically define your object model in the development environment and immediately launch a series of web pages that look somewhat like a Microsoft Access front end; you can list the objects in your database, query the database, and add new entries. While using the Direct to Web pages, you can customize them in real time with a Java tool that takes your changes and saves them back to the web application server. When you are finished you can "freeze" the pages and then use them (or parts of them) in your finished application.

A WebObjects application can be written on any platform that supports Java development. It is compiled into 100% Java byte code that uses a runtime environment also written in Java; it can be deployed on any web server capable of running Java. WebObjects applications perform very well and are very scalable; Apple's online store and the iTunes store use WebObjects.

WebObjects is not open source, but it is free to use and deploy. Many excellent third party open source extensions exist on the web.

What WebObjects Is Not

WebObjects is not a web page design tool. In fact, WebObjects does not have a WYSIWYG HTML editor. When using WebObjects you either hand code your HTML pages or use your favorite HTML designer application to create your web pages. This enforces the concept that web developers write code and web designers make web pages. 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 or SOAP and send them to a browser, but you will then have to deserialize them in your client code and deal with them. WebObjects is maintained by Apple and in my opinion Apple will not ever try to make it into a framework for developing an entire cloud or Web 2.0 / Application (for example the Google Apps). However it could be useful as either a back end for such an app, or as a means to develop a server driven application that uses extensive client side scripting to make the user interface more responsive (for example client side code to validate user input). You will have to write such code yourself. WebObjects (or rather extensions to it written by third parties) provides hooks to define and execute client side actions (known as AJAX).

Why I Use WebObjects

I once wrote a Microsoft Windows. Net application that consisted of a native Win32 front end coupled with an MS SQL Server back end. I needed to extend it to work outside of the local network, so I needed a web front end. MS ASP.Net would have been a natural extension. However, I was frustrated by how much time I spent writing code to interface with the database and how difficult it was to represent my data with custom objects. For example, in my .Net code, I created steps to explicitly perform queries on a database and the results are returned as dataTable objects which contain dataRow objects. A dataRow is similar to an array of strings. After they are loaded in memory, the row objects have no relations to each other; for instance if in my database a person is linked to a department with a foreign key, in memory a dataRow object representing a person does NOT have a pointer to the dataRow object representing his / her department. So I can't write a statement like:

```java
String myDepartment = aPerson.department.name;
```

Nor could I easily bind a user interface element to a custom property of an object. For instance I could not easily create a Person property called fullName (Firstname + " " + Lastname) and bind it to a drop down list. In order to do that, I had to create a custom Person object, copy all of my dataRow objects into a collection of Person objects, work with them, and when I was done write my own code for tracking which objects changed.
WebObjects is a mature and time proven method for writing business logic in Java, using all of your favorite frameworks and packages, which manages both the mess of database storage and HTML interface binding.

How to Learn WebObjects

To use WebObjects you need to use both the Apple frameworks and third party extensions along with open source tools. Apple developers also use these tools. Unfortunately not too long ago WebObjects development used an Integrated Development Environment (IDE) that is now obsolete and many references to it are still present in the documentation. To be clear, you will develop your applications using the Eclipse editor, which is an open source editor written in Java. On top of that you will install the WOLips plug in which provides an IDE for modeling your data and writing your code. Also, you will install a set of frameworks called Project WONDER which include bug fixes to the WebObjects framework and extensions that will make your life easier by performing common tasks.

Prerequisites

To start developing in WebObjects you need a basic understanding of Java, HTML, and what a database is and how tables, columns, rows, and relationships work. Having any knowledge of SQL or any experience with NEXT or Apple Cocoa development is a plus.

Concepts

To learn WebObjects I suggest first reading the (out of date) Apple documentation. Everything in it is correct except the references to the obsolete Apple development tools, and how to deploy WebObjects applications. Here is a good list of helpful documents, and a suggested reading order:

<table>
<thead>
<tr>
<th>Document</th>
<th>Subject</th>
<th>Ignore references to</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebObjects Overview</td>
<td>Conceptual Overview of WebObjects</td>
<td>Apple development tools, creating a full Java client (Swing) application running outside of a browser.</td>
</tr>
<tr>
<td>Java for WebObjects Developers</td>
<td>Java Skills Necessary for WebObjects</td>
<td></td>
</tr>
<tr>
<td>WebObjects Tutorial</td>
<td>Walkthrough of a Simple Application</td>
<td>The references to version numbers. As of this writing WOLips can be installed in Eclipse 3.5</td>
</tr>
<tr>
<td>WebObjects Web Applications Programming Guide</td>
<td>Common WebObjects Development Tasks</td>
<td>Apple development tools (in particular creating a new project, but your steps will be similar).</td>
</tr>
<tr>
<td>EOModeler User Guide</td>
<td>How to Model Your Objects</td>
<td>This instruction book references the obsolete Apple modeling tool. The WOLips modeler is VERY similar.</td>
</tr>
<tr>
<td>WebObjects Enterprise Objects Programming Guide</td>
<td>Working with Your Objects</td>
<td>Apple development tools and Java client applications.</td>
</tr>
<tr>
<td>WebObjects Direct to Web Guide</td>
<td>How to Rapidly Prototype an Application</td>
<td>Apple development tools, especially Builder (HTML editor) and RuleEditor.</td>
</tr>
<tr>
<td>WebObjects Web Services Programming Guide</td>
<td>How to Hook Other Applications Into WebObjects</td>
<td>Apple development tools.</td>
</tr>
<tr>
<td>WebObjects Dynamic Elements Reference</td>
<td>What Dynamic Elements are Available</td>
<td>Note: Third parties have extended this list considerably.</td>
</tr>
<tr>
<td>WebObjects Application Properties Reference</td>
<td>Advanced Configuration of an Application</td>
<td></td>
</tr>
<tr>
<td>WebObjects XML Serialization Guide</td>
<td>How to Serialize Your Objects Into XML</td>
<td>Apple development tools and Java client applications.</td>
</tr>
</tbody>
</table>

For advanced concepts and best practices, buy Practical WebObjects, a great book.

For general information, check out the WebObjects Project homepage, and the WebObjects Community homepage.

For community support join the WebObjects mailing list.
After you’ve familiarized yourself with these documents, you can begin coding!

**Getting Started**

First you need to install WebObjects. Start at the [WOLips Project Install Page](#).

To see some simple projects built with WOLips, go to [Don Linsay's tutorials](#).

For a walkthrough of a more complex application than Apple’s tutorial and various podcasts of development see the WebObjects Community homepage- Getting Started (link above).

For information on additional components available in the WONDER see the WebObjects Project homepage (link above).