Best Practices-Starting From Scratch

In construction

This chapter is under construction. Do not expect it to be useful... yet! 🙂

Starting From Scratch

Building a good WO application is not just about nice data models and fast algorithms. Making the right decisions from the start and taking advantage of code written by the community is also very important. Here we'll talk about some stuff that you should care about from the start.

Use WONDER

Why?

WONDER is a huge set of frameworks that fix and extend WebObjects in many, many ways. For more than five years, Apple barely updated WebObjects, and that shows. There are bugs that date from 2003 still unfixed, there are almost no new features. WebObjects 5.4 is a nice release that includes a lot of fixes and some new functionality, but it was only released in late 2007, with Leopard. During all that time, though, the community hasn't been sleeping. The WONDER team wrote thousands of lines of code, and made them available for the rest of us.

The WONDER framework most people use is ERExtensions. This framework is probably the biggest in WONDER, and contains a lot of great functionality you don't want to live without. Some of it's features are:

- XHTML generation. Although WebObjects 5.4 finally provides XHTML compliant code generation, WONDER was the only way to achieve that in previous WebObjects versions. I mention this feature first because it actually was the feature that made my start using WONDER in first place.
- Automatic EOEditingContext locking. This feature is huge. We'll talk about locking a lot in later sections, but trust me, you'll want WONDER to handle the locking for you. No more unlocked contexts to be corrupted while you edit them, and no more deadlocks and locked contexts lying around.
- Collections re-implementation supporting Java Generics. This means that you may create an NSArray, NSDictionary or any other "NS collection" typed to a class or interface. I don't actually like this feature, because I'm a LISP lover and I see this as unnecessary constraints. Anyway, I mention this because I admit it may improve the code quality in some ways, and because WONDER implementation has the advantage that it can be used in pre-5.4 WO versions (Apple introduced generics support for collections in 5.4). The are also some common-used methods, like pageWithName(), that were implemented in WONDER using generics.
- COUNT SQL query generation. This is a very, very useful feature, at least for me. There are many times when you just want to know how many objects are selected by a query, but you don't want the objects themselves, specially because we may be talking about tens of thousands, or more. Think something like Amazon querying its system for the number of books sold this month. The query results would be huge and unpractical to manage in memory and time, and an unacceptable overhead if all you want is the count. WONDER provides a very easy way to do COUNT (and COUNT UNIQUE) queries.
- Extra qualifiers. Sometimes WO qualifiers are simply not enough. WONDER provides some handy qualifiers. Some of them are buggy and may not work, it's not one of the best quality WONDER features, but at least it's there, and it's open source, so you can fix any bugs you find.

There are many, many more nice things about EXExtentions framework. You would be really bored if I described only 10% of them. Besides ERExtentions, there are also a number of cool extra frameworks in WONDER. Here are some of them:

- Ajax Framework. Provides a really easy way to add Ajax calls to your application. Ajax Framework integrates Ajax in the WebObjects Request-Response loop in a very intelligent way. You program all of the Ajax stuff as you program normal component actions. An action method is called, you do whatever you want, and return a component (usually null). The defined block of the page is updated. No page reload, the user is happy, and you are more than happy because you actually had to do nothing (well, almost). We'll look at Ajax Framework later.
- PostgreSQL Plugin. This allows WO to be used with the popular PostgreSQL open source database. The plugin is needed to that WO knows how to deal with primary keys, and some other details.
- ERJavaMail. Useful framework for managing email.
- ERPrototypes. A collection of data model prototypes for popular databases. You may use this framework or create your own prototypes, but **do** use prototypes. If you are lost, don't worry, we'll look at this later.
- ExcelGenerator - A POI wrapper framework for generating Excel documents, a very useful feature in many web applications.
- ERPlot - A JFreeChart wrapper for generating nice-looking charts on your web application.

By now, you must be thinking why would you make the learning curve even worse by learning WONDER and WO at the same time. After all, it's a lot of frameworks, a lot of information and, unfortunately, not a lot of documentation.

The point is: **using WONDER will actually make it easier to learn WebObjects**. WONDER fixes a lot of bugs and provides a lot of easier solutions, like EOEditingContext automatically locking, that would otherwise drive you mad if you use pure WO. So, the best option for you to learn WO easily is to use WONDER, at least ERExtensions. Although it seems you are complicating your life, you are really making it (a little) easier.

What WONDER version should you use?

Many of us, WebObjects developers, are very conservative about upgrading and living on the bleeding edge of the technologies. WebObjects is used for real business, where many millions of Euros are involved. So, normally the advice would be: use the stable version.

Well, WONDER is different. WONDER is updated very fast, and at the same time the WONDER team makes a big effort to keep it stable. There are many small fixes every week, that will improve WONDER reliability, and new big features are usually included as optional, off by default. So, my advice on this is, use the nightly build, or, even better, the CVS head source code. Keeping the source code on disk will make it easier if you want to fix a bug and submit a patch. Instructions for compiling and installing the framework are included in the CVS. To install the nightly build, just uncompress it and dump all the frameworks into your /Library/Frameworks/ directory.

How to integrate WONDER with your application

If you are creating a new application, you can just do the following after installing WONDER in your system:

1. In Eclipse, choose "New / Other..." from the File Menu. A window will appear asking you what type of file do you want to create.
2. Open the WOLips folder, by clicking on its disclosure triangle, and choose "WOnder Application".

That's it. This will create a new WO application, ready to be used with WONDER.

If you want to integrate WONDER with and existing Application, you must follow **all** the steps described in the integration page of the WONDER wiki. Note that it's not just adding the frameworks to the project, you really have to make all the code changes described there (it's easy to do, but don't skip it, or you'll have problems).

Keep all the model functionality on a framework

Many web application are not really a single application. Even if the end user interacts with only one application, many times you need to create back-offices, maintenance application that run scheduled jobs, etc.

When you need to do it, you basically have two options:

- Duplicate all the model code you need. That's, obviously, bad bad bad.
- Create the new application, link it to your model framework, and you're done.

It helps keeping most of your code in the model. Maintenance tasks usually have to do complex operations that share code with the main application itself. So, keeping it all in one place avoids code duplication and makes your life easier.

Besides, it's really easy to create a framework:

1. In Eclipse, choose "New / Other..." from the File Menu. A window will appear asking you what type of file do you want to create.
2. Open the WOLips folder, by clicking on its disclosure triangle, and choose "WOnderFramework" (or "WebObjects Framework" if you don't want to use WONDER).

Then, you need to link your application to the framework:

1. Right-click on your application project icon, and choose "Properties" from the contextual menu.
2. Choose "Java Build Path" from the list on the left.
3. Click on the "Libraries" tab on the right.
4. Click the "Add Library" button, select "WebObjects Frameworks", click "Next" and choose your framework. Click "OK" to close the Build Path window.
5. Done. Now you're project is linking with your framework. If you think this is overkill, think again when you need to do a maintenance application of a backoffice. You'll love the time you spent separating the model code from the application. Also, this helps you write better code, because model code shouldn't need to know anything about the presentation layer. If you try to do that in your framework, you'll get an error, so you really can't do it! 🙂

If you don't want to create the model framework now, at least take the advice of keeping the model layer fully independent from the presentation layer. This will make it easier if you decide, in the future, to separate the model code to a framework as described here.

« Overview Model »