

Project WONDER-Quickstart

Quickstart

Using Project WONDER is surprisingly easy. There are only a few steps to get up-and-running:

1. Download and install the Project WONDER frameworks;
2. Add the ERJars.framework and ERExtensions.framework to your WebObjects project;
3. Change your Application class to extend er.extensions.ERXApplication instead of WOApplication;
4. Change your Application.main method to call er.extensions.ERXApplication.main instead of WOApplication.main;
5. Change your Session class to extend er.extensions.ERXSession instead of WOSession;
6. Change your DirectAction class to extend er.extensions.ERXDirectAction instead of WODirectAction;
7. Build.

When using Xcode out of the box, there will be an error upon launching this application (as of Feb 6, 2007):

"No ERX_MARKER field in NSMutableArray found."

This means your class path is incorrect. Adjust it so that ERExtensions come before JavaFoundation. To do this, you must change the order of your classpath so that ERExtensions appears above JavaFoundation. In my default classpath, JavaFoundation appears first in the list. If you are using Xcode, follow Jerry Walker's instructions:

1. Select "Application Server" in your Active Target popdown menu at the top left of your Xcode window.
2. Type command-option-e or choose the Project->Edit Active Target 'Application Server' menu item. This will open a target editing window.
3. Select 'Link Binary With Libraries' under 'Build Phases' in the left-hand view.
4. You can edit the framework load order by dragging framework up or down in the list on the right. Drag the ERExtensions.framework to the top.
5. Rearrange these frameworks in the editing pane into any order that you desire by dragging and dropping them within the editing pane.
6. Set your Active Target popdown menu back to the name of your project. **THIS IS IMPORTANT.**
7. Clean and rebuild your app and your classpath settings should be rearranged accordingly.

This error only occurs with Xcode and, obviously, this workaround applies to Xcode and not Eclipse. Also, there is also an Xcode project template for an ERXApplication available. (from where?)

If you are using WOLips, you can create a new "Project Wonder Application" which will setup a project with these steps already performed.

That's it! You are now using Project WONDER and you've begun your journey to easier WO application development.

Notes from David Teran

From David Teran Dec 1 2004: **Pointers for getting your head around the frameworks**

"PW consists of different frameworks, the most important one is ERExtensions, then for D2W apps additionally ERDirectToWeb, some if not all people are using ERCoreBusinessLogic. The other frameworks are build on top of one or multiple from these.

BUT: i would start reading the class description and then the javadoc, its generated from the sourcecode with eclipse within a second (ok, maybe 60...) and its read in about 1 - 3 hours.

Then i would take a look ERXApplication, ERXSession, ERXGenericRecord, ERXEC (very interesting!), ERXThreadStorage (very interesting, too), ERExtensions, ... and the various utility classes: ERXEOControlUtilities, ERXEOAccessUtilities, ERXArrayUtilities and so on."

Notes from Chris Meyer

Internationalization and Localization

Practical WebObjects outlines one technique for handling internationalization. It is incompatible with Project WONDER.

To handle internationalization in Project WONDER, delete the takeValuesFromRequest from your Session class (which should now be derived from ERXSession). Likewise, delete the createResponseInContext method from your Application class (which should now be derived from ERXApplication). Then add the following call to your Application constructor:
ERXMessageEncoding.setDefaultEncodingForAllLanguages("UTF-8");

Logging

Practical WebObjects outlines one technique for SQL logging. Project WONDER uses an entirely different technique.

In Project WONDER, put this in one of the properties files (I use ~/WebObjects.properties)

```
# turn off annoying INFO logs
log4j.logger.er=INFO
log4j.logger.er.extensions.ERXExtensions=OFF

# Enable delegate to emit SQL debugging info. The Logger used is
log4j.category.er.extensions.ERXAdaptorChannelDelegate.sqlLogging=DEBUG
# put this to true if you want to log sql stuff
er.extensions.ERXAdaptorChannelDelegate.enabled=true
# How long a statement must run to cause a log message. Messages with
longer than
# error also emit a stack-trace
er.extensions.ERXAdaptorChannelDelegate.trace.milliseconds.debug=0
er.extensions.ERXAdaptorChannelDelegate.trace.milliseconds.info=50
er.extensions.ERXAdaptorChannelDelegate.trace.milliseconds.warn=1000
er.extensions.ERXAdaptorChannelDelegate.trace.milliseconds.error=5000
# MaxLength of the message
er.extensions.ERXAdaptorChannelDelegate.trace.maxLength = 30000
# What entities to watch
er.extensions.ERXAdaptorChannelDelegate.trace.entityMatchPattern = .*
```

Backtracking

Project WONDER offers a few techniques to deal with backtracking.

My recommended way is to add a method to your WOComponent:

```
/**
 * Determine whether the user backtracked.
 * @return true or false indicating whether the user backtracked
 */
public boolean didBacktrack()
{
    return ((Session)session()).didBacktrack();
}
```

Then anywhere a form is submitted where backtracking might be a problem, add this code to the action method:

```
if (this.didBacktrack())
{
    ec.revert();
    // handle and prepare to report errors
    // this.errors.addObject("Unable to process page after back button
was pressed.");
    return this.context().page();
}
```

ERXWORepetition also contains some code to deal with backtracking. As of Project WONDER 3.0, the code seems to work inconsistently. When the page is backtracked and the form submitted, values in the form are just passed as null. This tends to cause more problems than it is worth, although your situation may be different. Also, there is an option to throw an exception when the page is backtracked, but this exception doesn't seem to be thrown consistently.

Javadocs

Prebuilt

- <http://wocommunity.org/documents/javadoc/wonder/latest>

Creating javadocs from the Wonder source (not using Eclipse)

If you want to create the javadoc and you haven't got Eclipse going yet:

1. Get the latest source from the nightly build server.
2. Once you untar it you'll have a Wonder folder created.
3. In the terminal application, type:

```
cd Wonder/Build/build
ant -f build-doc.xml
```

4. Wait several minutes while some good stuff scrolls by in your terminal window. If ant gives you errors, you may have to set it to use ant 1.4 instead of 1.5 by exporting: `JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.4/Home`
5. Find the new `dist/wonder-4.0/Documentation/api/index.html` page that has been created. The dist directory is at the same level as the top-level Wonder directory. So, you may try:

```
open ../../../../dist/wonder-4.0/Documentation/api/index.html
```

6. Or you can double-click the index.html page and the javadocs will open in your favorite browser

Creating javadocs from the Wonder source (using Eclipse)

Someone else will have to fill in this section...

```
ERXWORepetition
```