

Create a WebObjects Framework

- Introduction
- How To...
 - Adjust Your Filters
 - Create the Framework project in your Eclipse workspace.
 - Bundling 3rd Party Jars in your Framework
 - Create a package (or packages) for your reusable classes.
 - Add classes to your package. You can drag java files from the file system or XCode to the package:
 - Cleanup Time
 - Install The Framework

Introduction

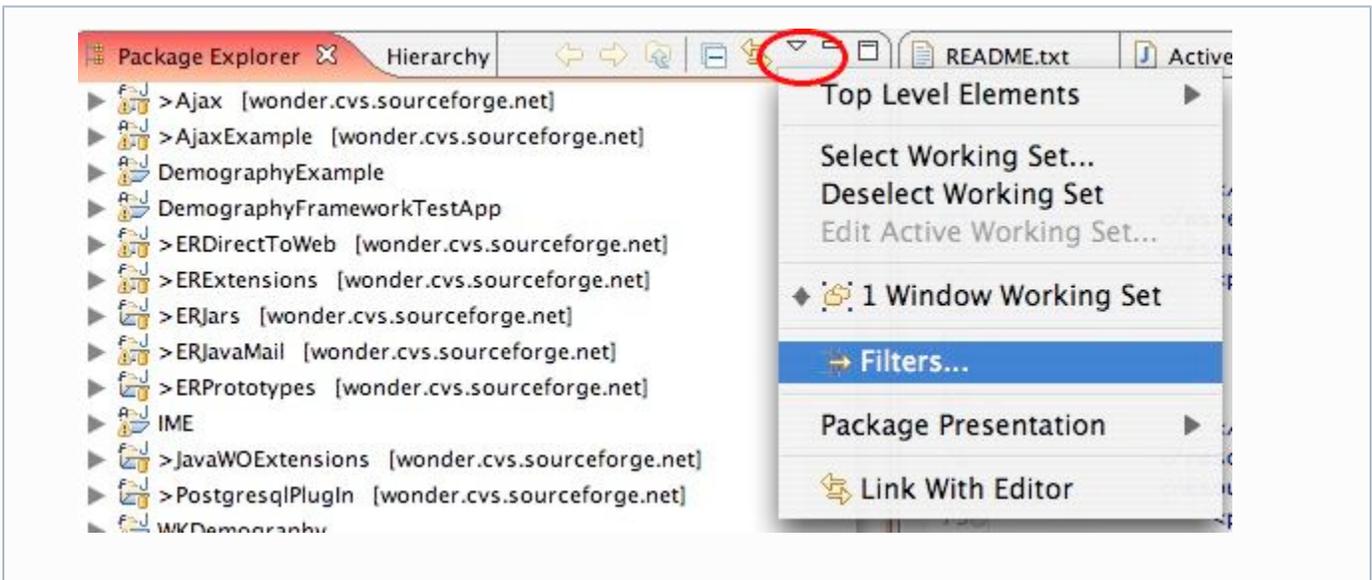
So, you have built the mother of all WebObjects apps in XCode or Eclipse with everything in the default package (or not 😊). The time has come where you want to make a new WebObjects application that re-uses some of the functionality you have already created in bigApp.woa. This article explains how to create a framework in Eclipse that can be used with your WebObjects projects. The resulting framework can be used in WOLips or XCode projects. In this example a framework is created that consists of class(es) that were in the default package of a WebObjects application and the 3rd party jar dependencies that were originally in /Library/WebObjects/Extensions. The resulting framework bundle will encompass the 3rd party jar and the custom java classes that are moved from the original WO project.

Disclaimer: I am still relatively new to Eclipse, so if you see a smarter way to do this, then feel free to edit this wiki page! Also, at the time of writing this article I was using Eclipse 3.2.1 with WOLips build #3769.

How To...

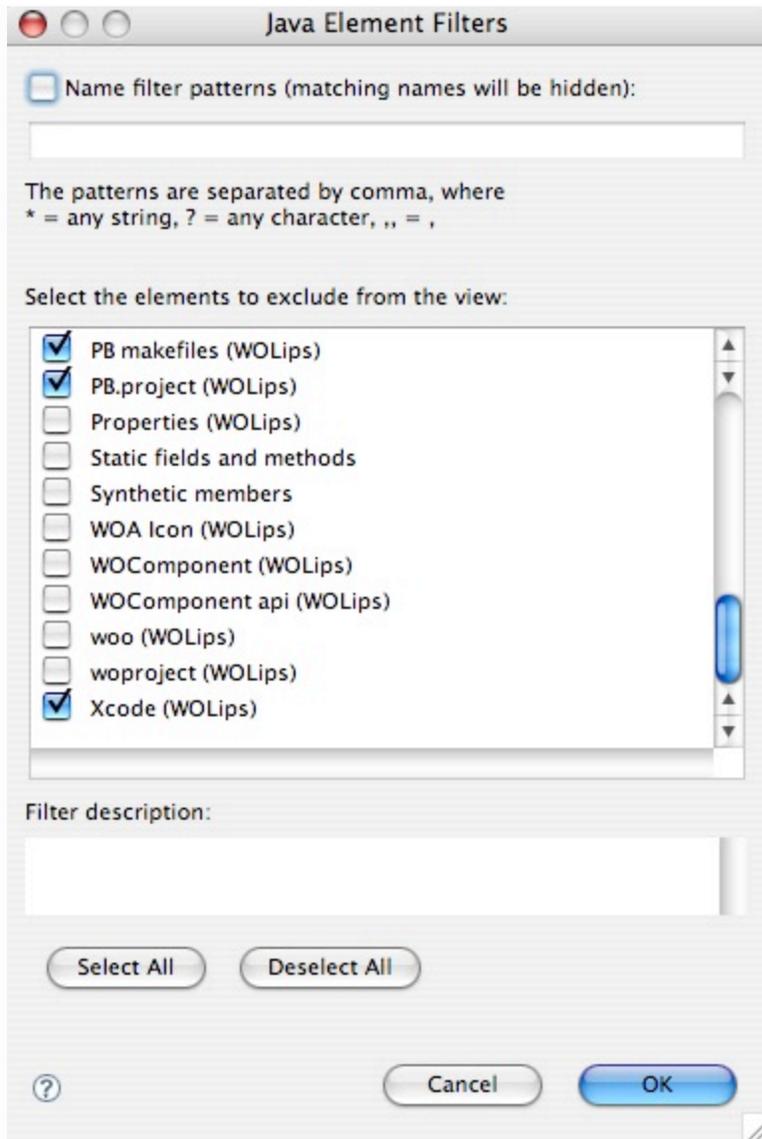
Adjust Your Filters

Before you start, edit your Package Explorer filters so that we can really "see" the files in our project, especially so we can check if our project was created correctly by the wizard later on. To do this simply select Filters from the Package Explorer contextual menu as shown below:



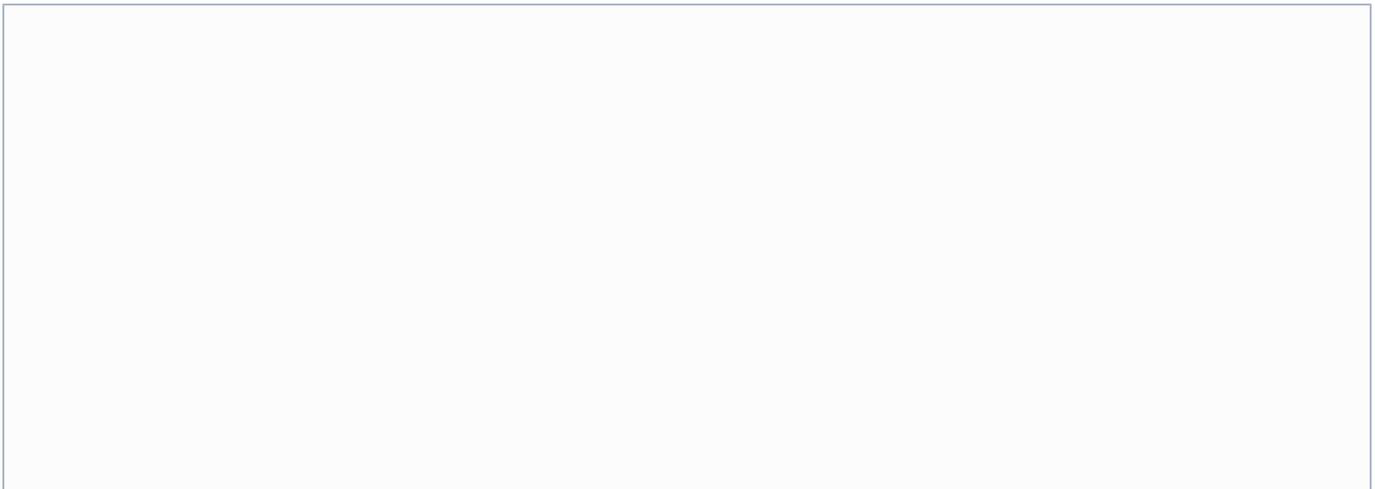
... and then deselect everything except the ones shown here. This of course is a user preference, but at least your screen will match my initial project picture shown here if you do as I tell you!

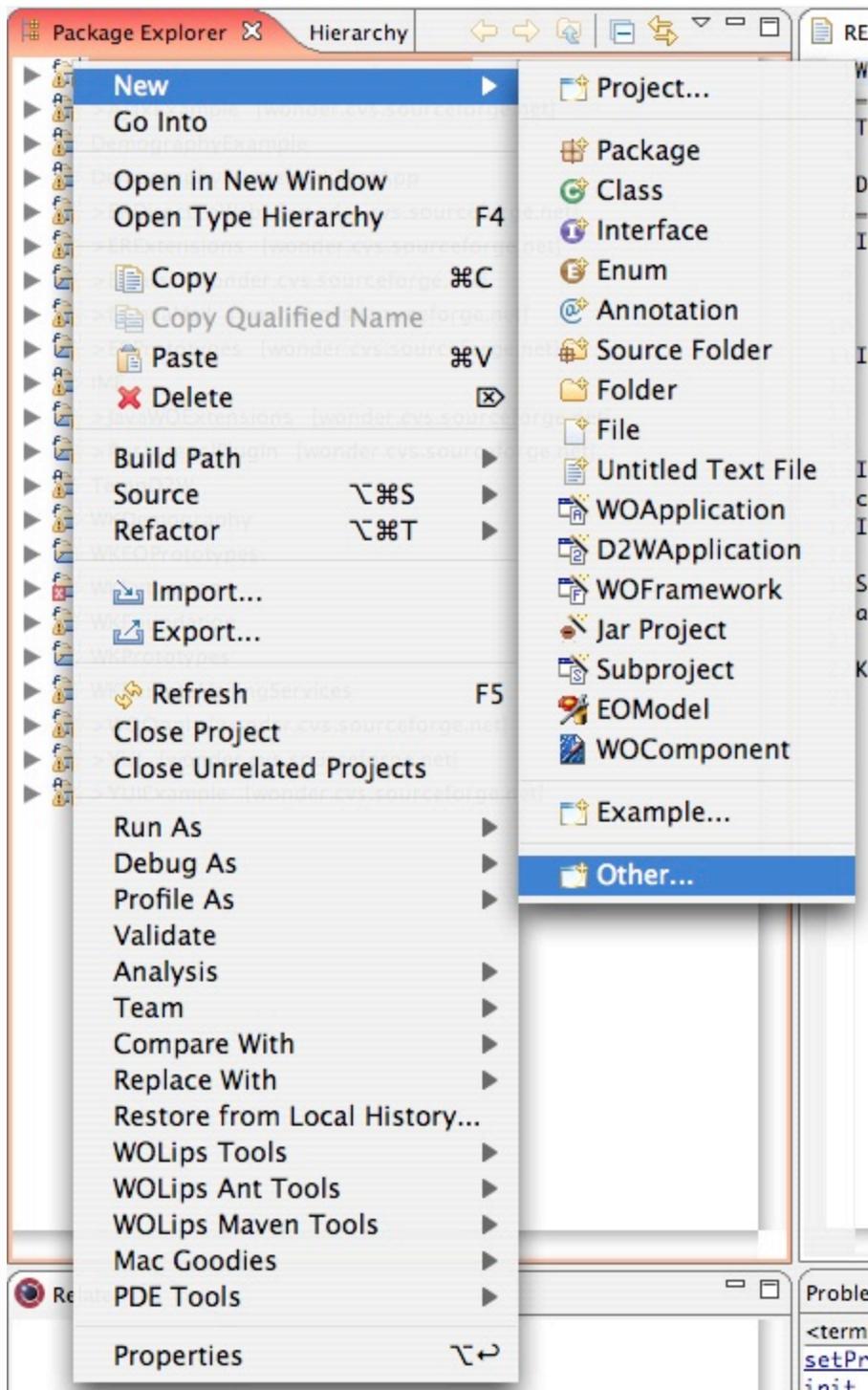




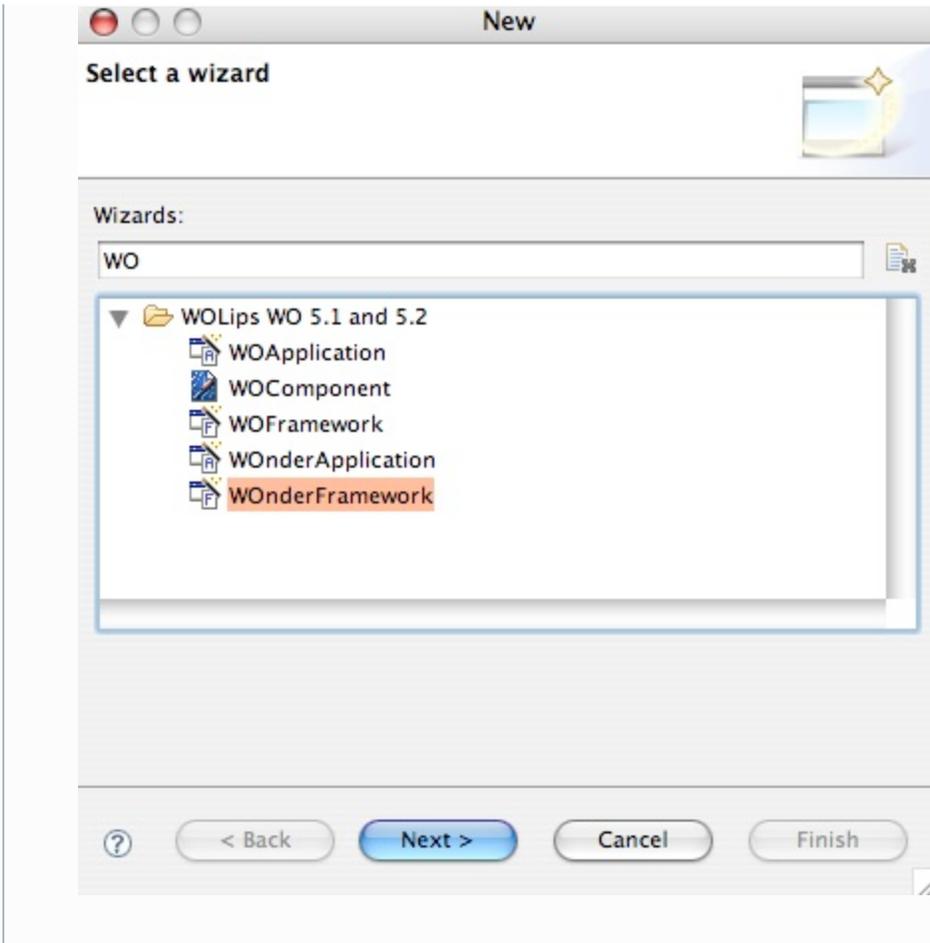
Create the Framework project in your Eclipse workspace.

Right-click in Package Explorer view of the WOLips perspective and select New->Other.

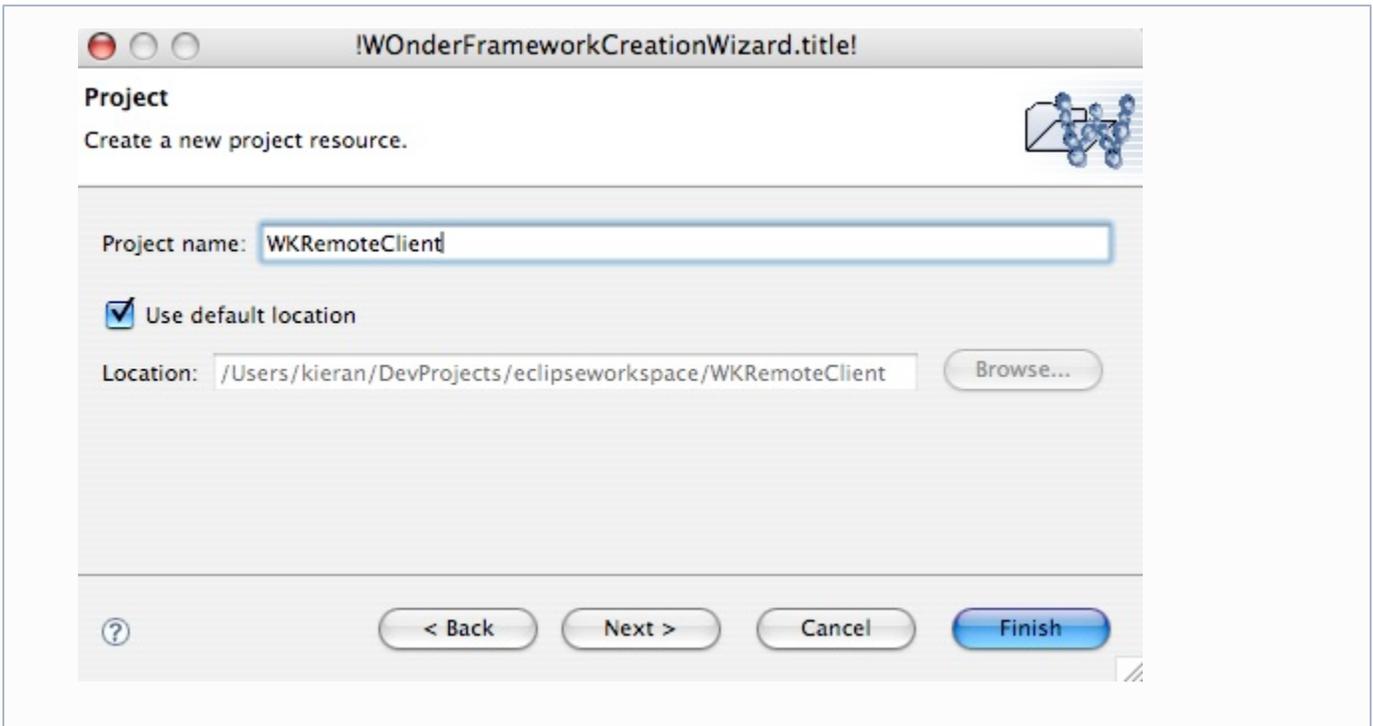




Next type WO in the filter field and select WOnderFramework from the list and click Next.



Next enter a project name and click Finish

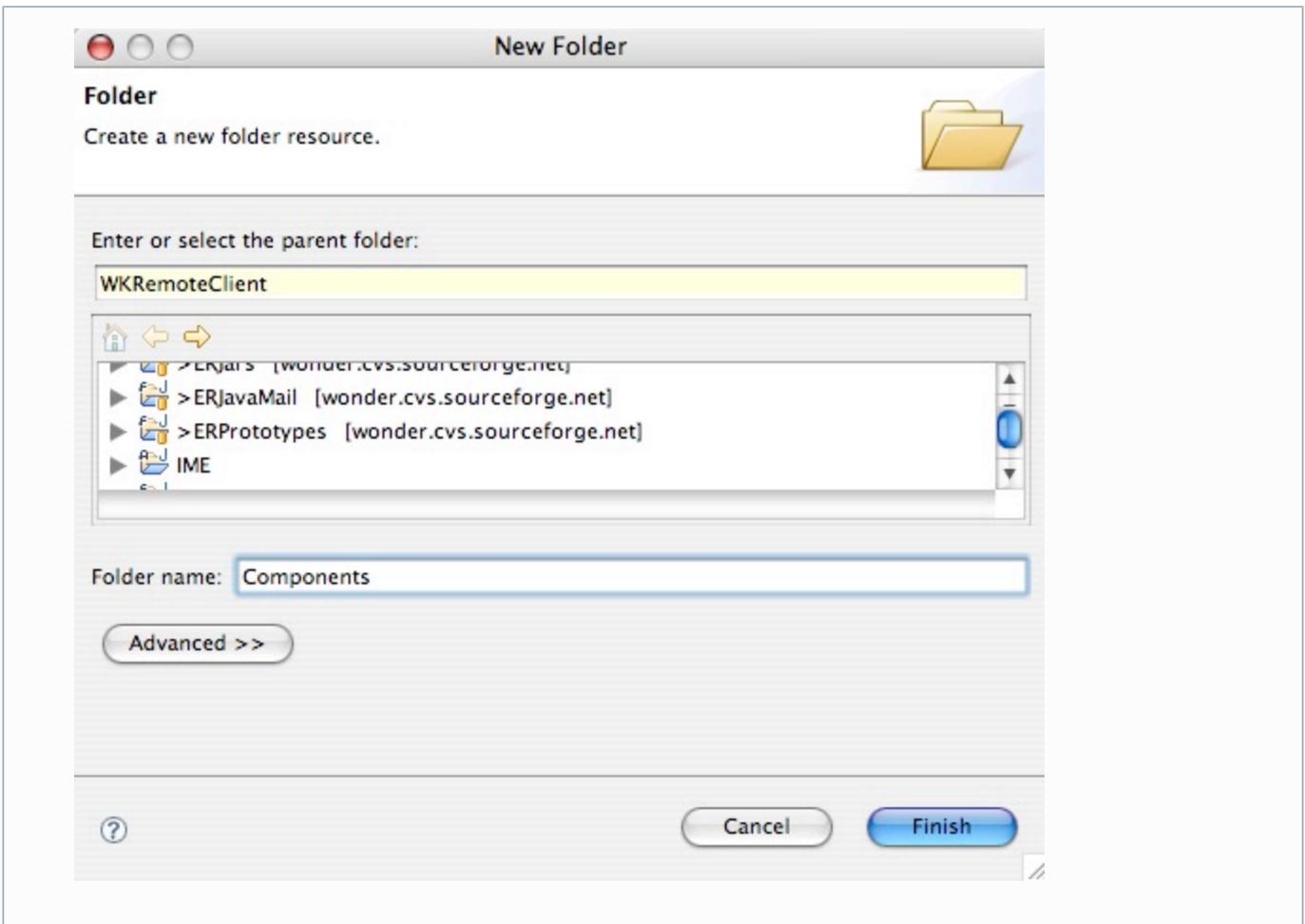


WOLips Bug: If your new framework project DOES NOT look like the one in the picture below, then you have come across a known elusive WOLips bug that the team is trying to solve. In the case that your project does not look like this, the fix is to select the project and then select the Project->Close in the main menu bar. Afterward select Project->Open and then expand your project and it should be

OK.



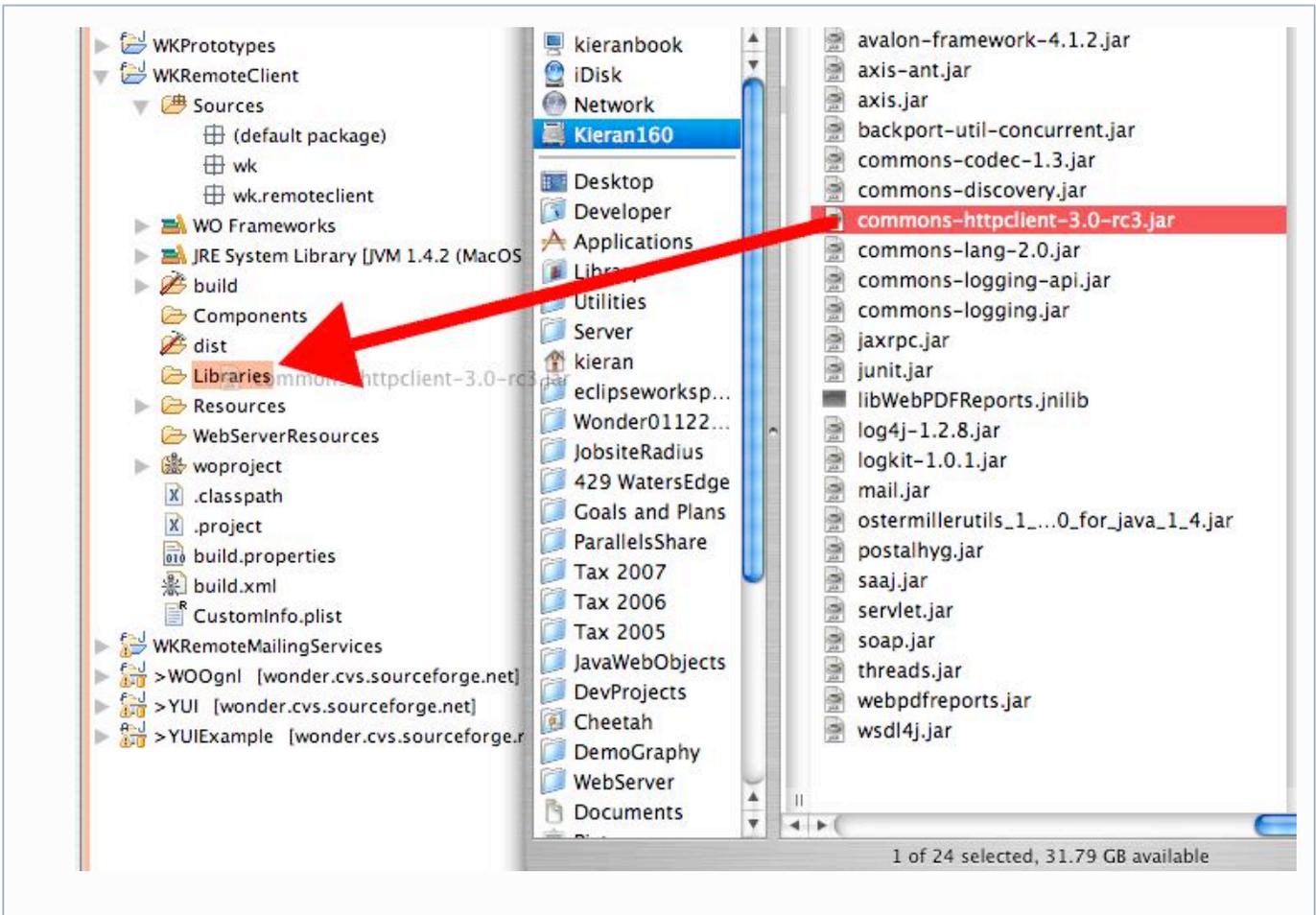
Another Minor Bug: In the latest build I was using, the default framework project does not contain a folder named 'Components' which it needs to have a successful build later. So just create a new folder in the project named Components. Right-click the project -> New -> Folder



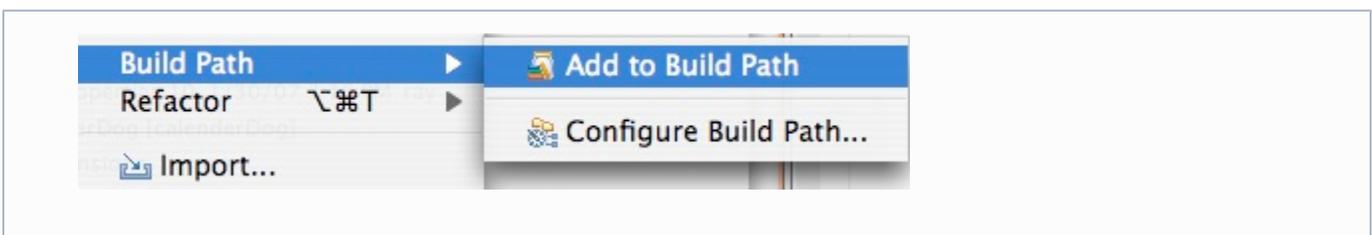
Bundling 3rd Party Jars in your Framework

Next add any 3rd party jar files to your project that you want bundled with this framework. This is probably the "best practice" way to make 3rd party jars available to your WebObjects apps rather than stuffing them into /Library/WebObjects/Extensions, or even worse, putting them in /Library/Java/Extensions, which can lead to deployment hassles later.

To do this simply locate the jar in the Finder and drag and drop it on the Libraries folder in your project as shown here:



Next simply right-click on the jar file itself and select Build Path -> Add to Build Path.



IMPORTANT:

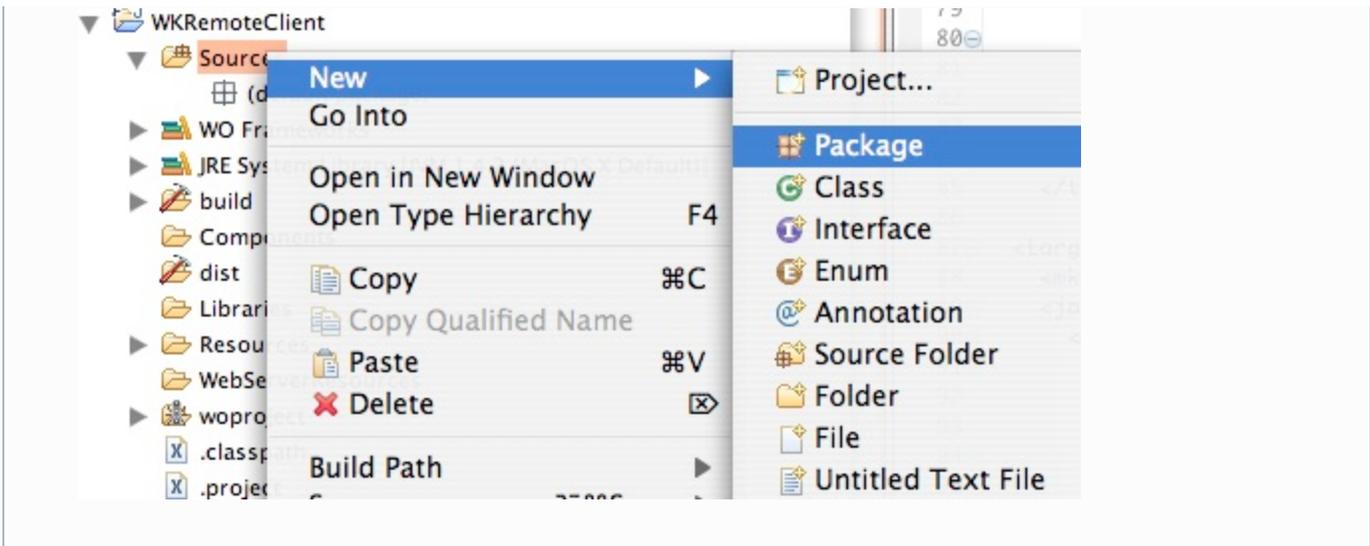
- Next right-click on the project and select Build Path -> Configure Build Path...
- Then to to the Order and Export tab and turn ON the checkbox next to the jar that you just added to the build path. By doing this, the jars in the framework are automatically available to any WebObjects apps/frameworks that depend on this framework in your workspace.

Adding dependencies on other frameworks can be done at this stage if you wish. The procedure is the same as for adding framework dependencies to a WOApplication which also has a tutorial. Keep in mind that the WOnder Framework wizard adds default framework links which you can adjust and turn off easily (right-click WOFrameworks icon and select Configure)

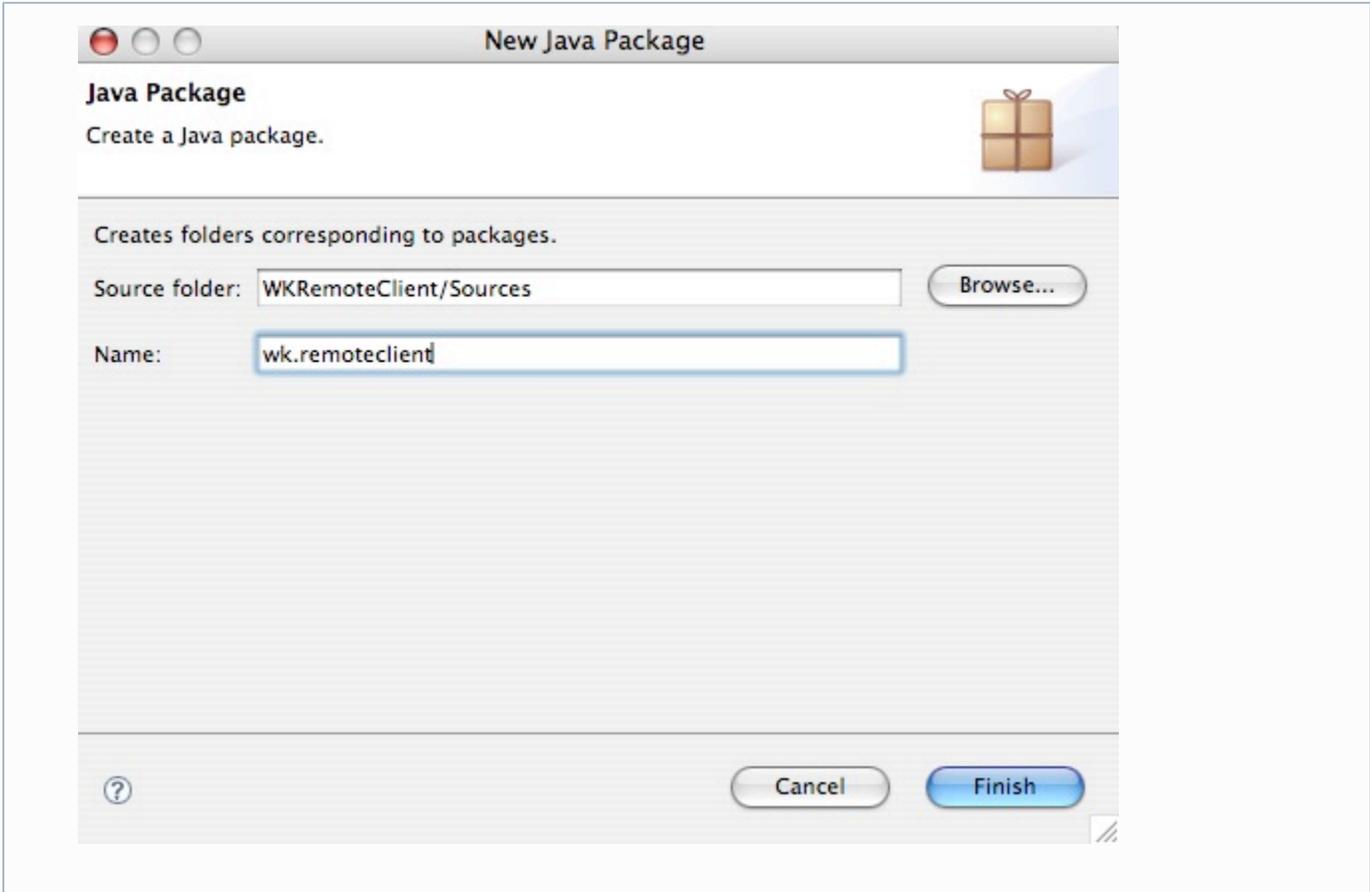
Create a package (or packages) for your reusable classes.

Right-click on the Sources folder and select New -> Package as shown here:



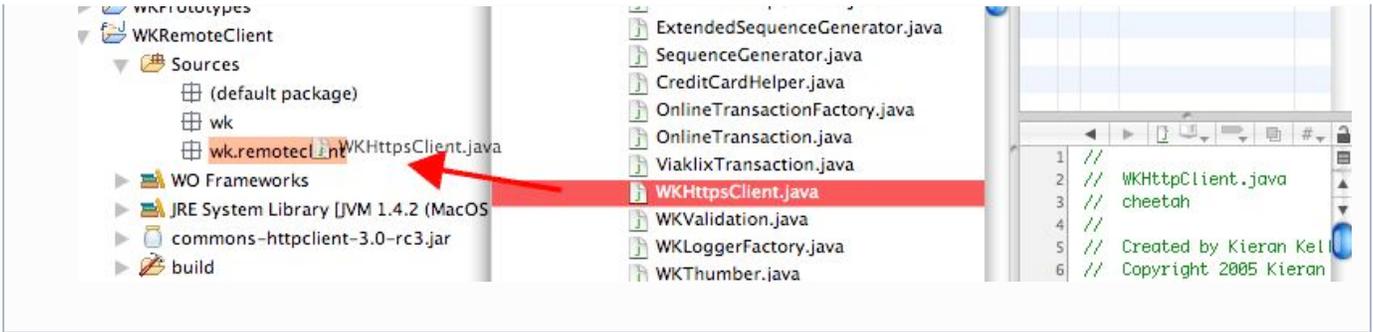


Enter a name for the package and Eclipse automatically creates the appropriate folder structure in the file system:



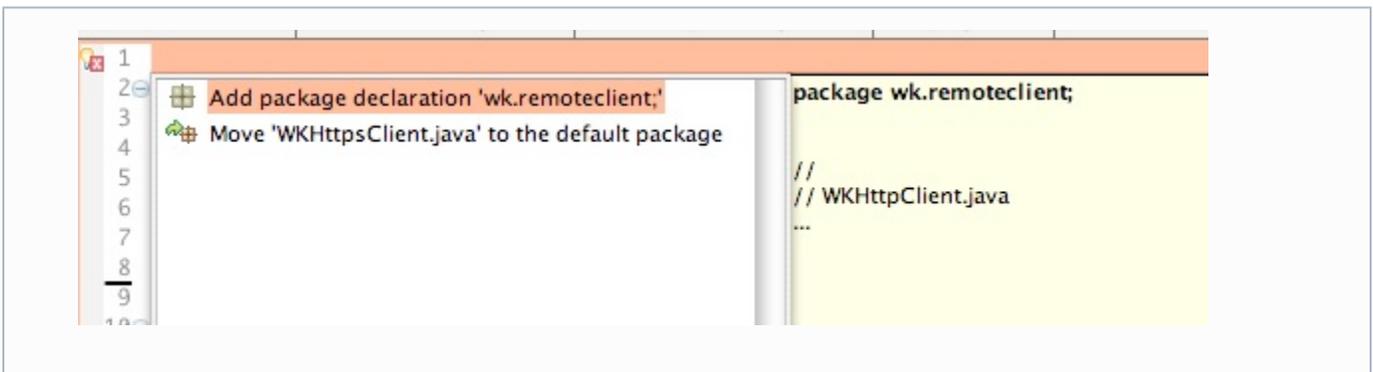
Add classes to your package. You can drag java files from the file system or XCode to the package:





Cleanup Time

Eclipse will check and compile your classes and sprinkle red x's and yellow exclamations on the areas needing your attention. (Isn't Eclipse just great ... intelligent assistance, refactoring, etc.) Anyway go ahead and fix/refactor to fix code problems and resolve dependencies. There is nothing like trying to package a large "default package" app to show you where you have opportunities to clean up dependencies or to show you dependencies you did not realize you had!



Install The Framework

Building and/or installing is the same as for WOApplications in Eclipse

