

# Working with Wonder source in Eclipse

## Introduction

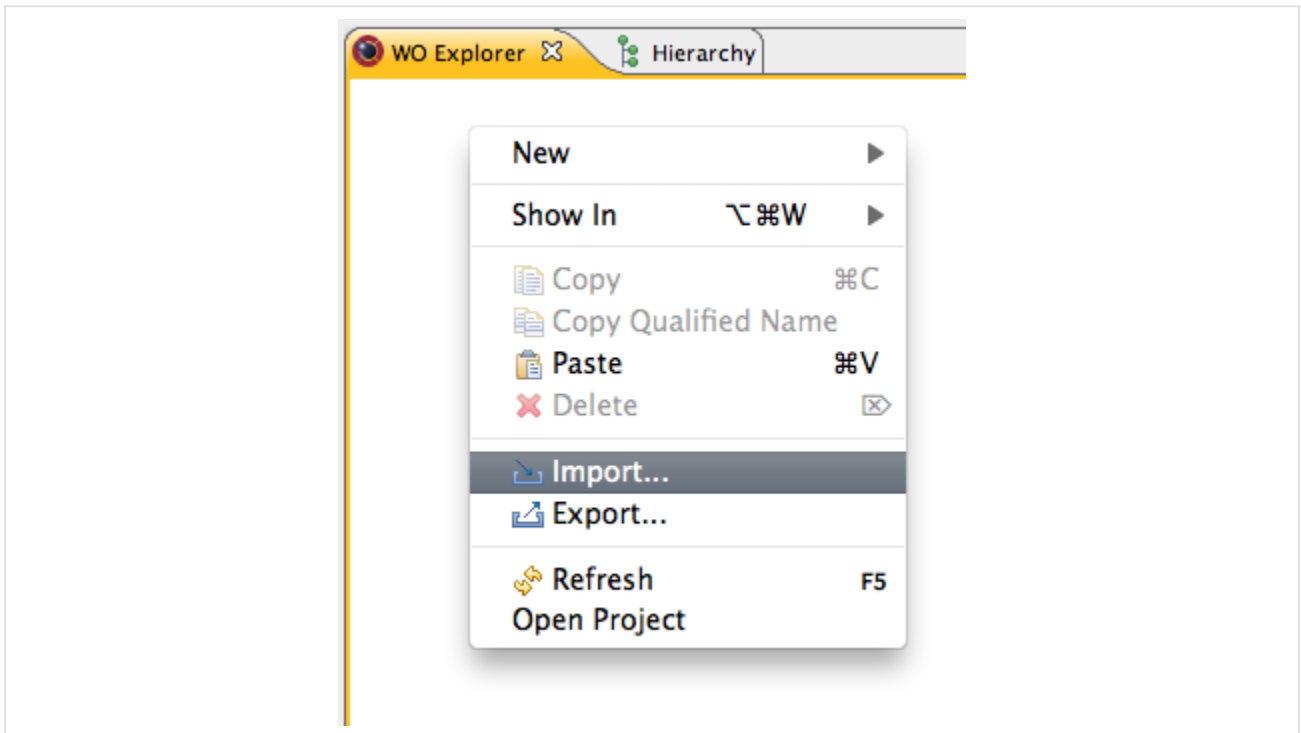
To work with Wonder source in Eclipse we import, without copying, the source projects from the Wonder source tree we downloaded in [an earlier tutorial](#). By doing so, we make sure our workspace Wonder source projects are the same source as what we have used for our build and install.

## Prerequisites

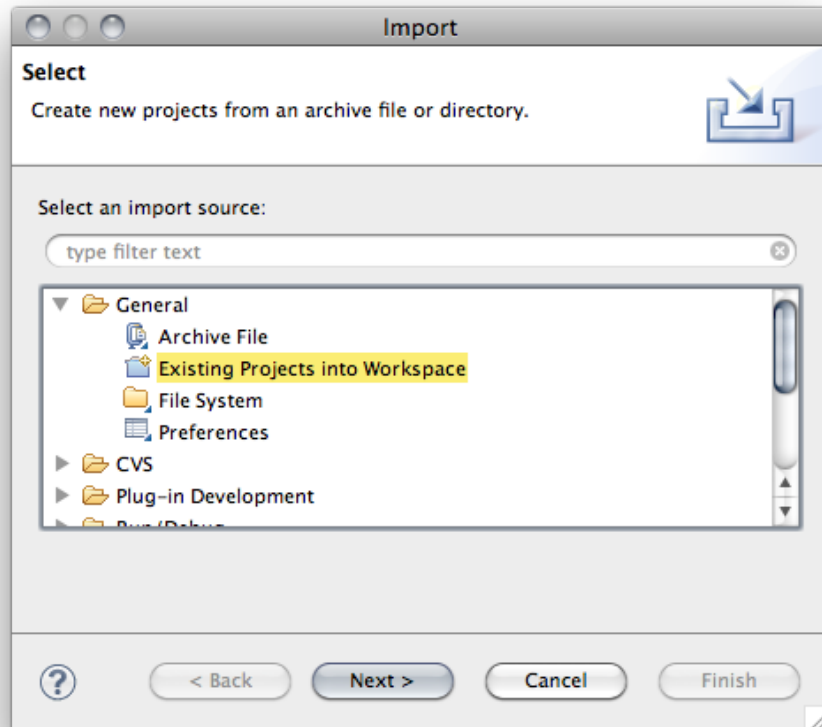
Wonder source tree is checked out from source control: [Getting the Wonder Source Code](#)

## Importing Wonder Projects into Eclipse

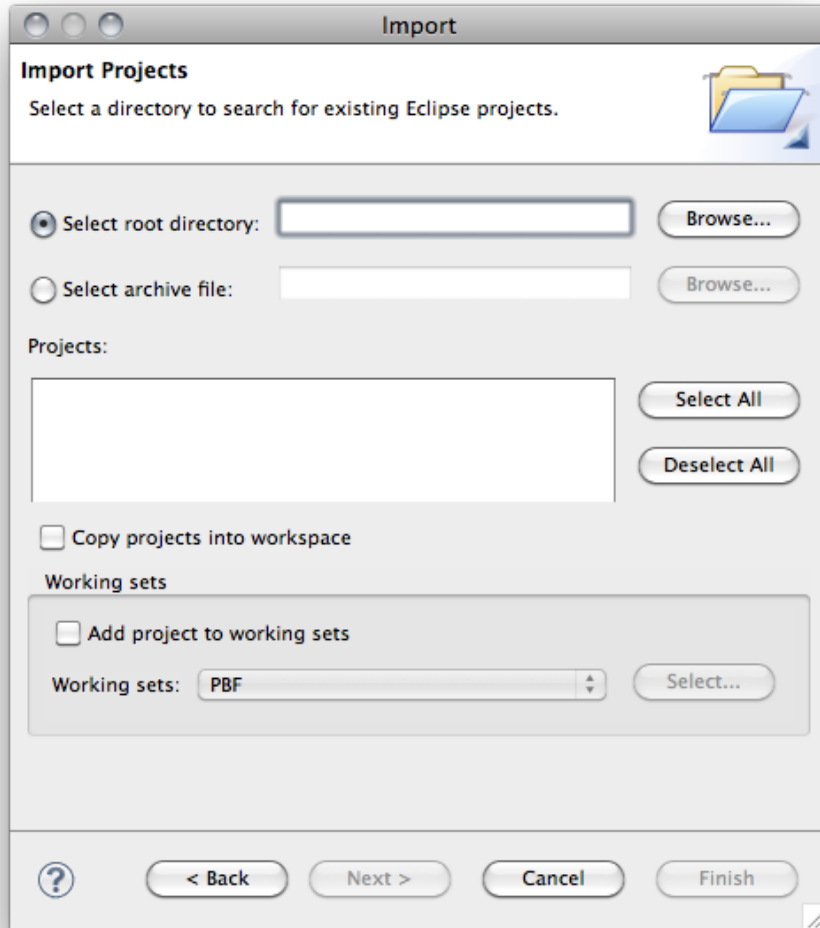
1. Right-click in WO Explorer and select "Import..." from the context menu.



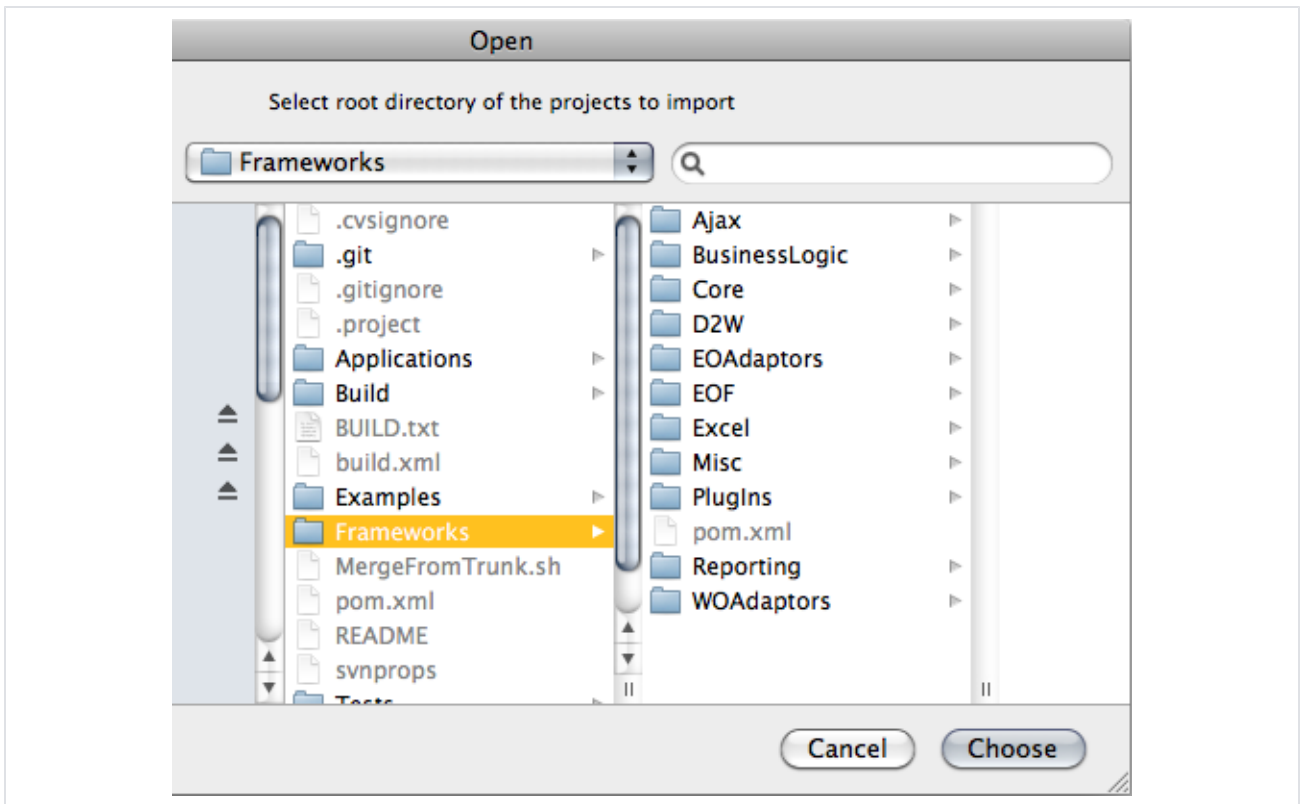
2. Select "Existing Projects into Workspace" as the import source.



3. Click "Browse..." to select the root directory containing the projects.



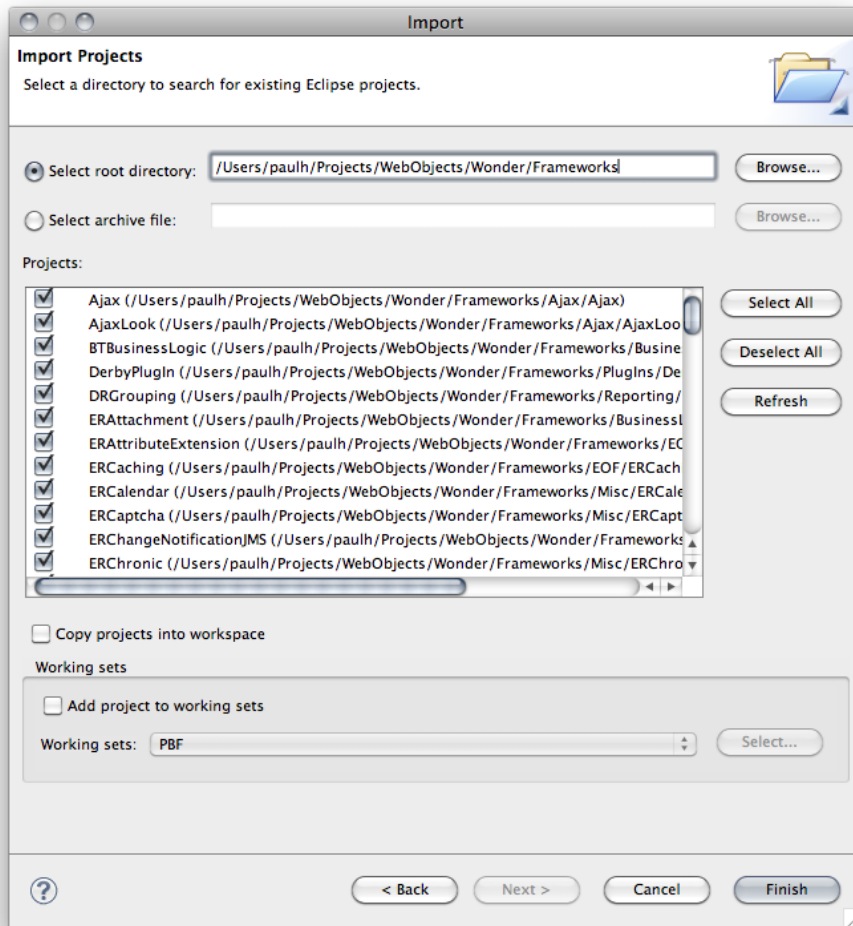
4. To import framework projects, for example, navigate to and select ".../Wonder/Frameworks" and click "Choose".



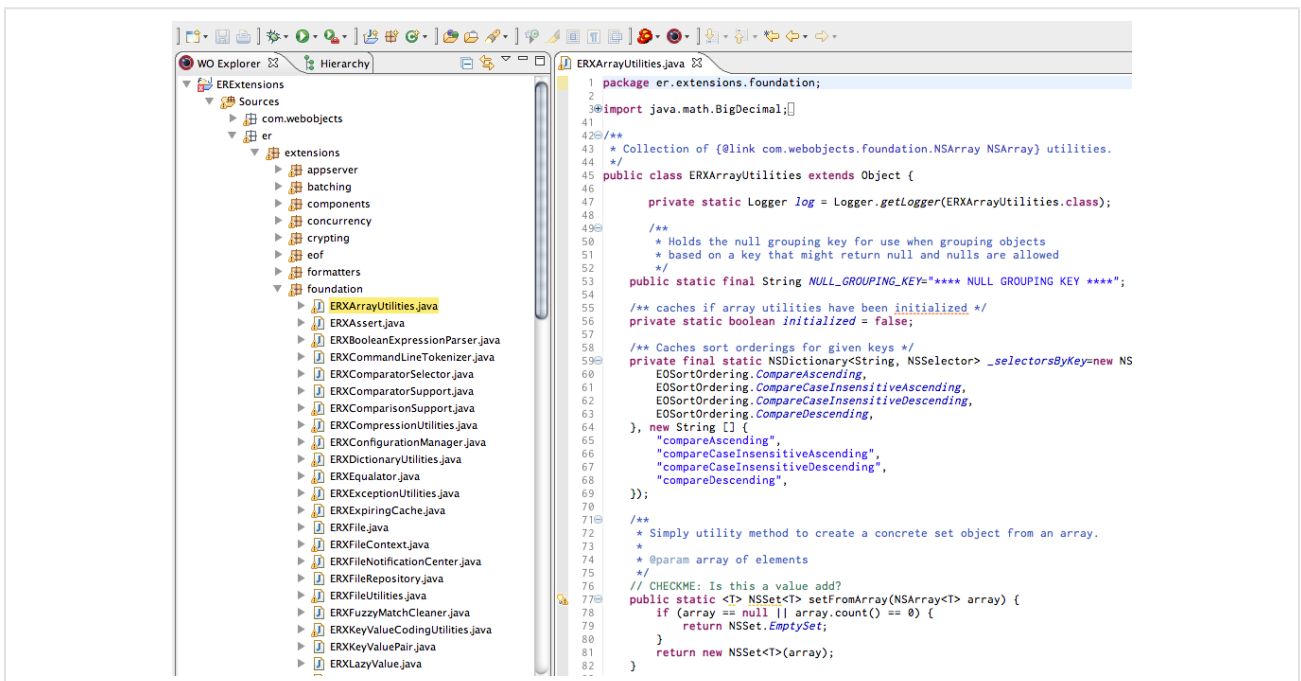
Wonder isn't just Frameworks!

Project Wonder also has replacements for JavaMonitor and wotaskd and several example applications that are great to learn from. To import the example application projects, select the "Examples" directory instead of "Frameworks" above.

5. The list of projects inside that folder appear checked. Uncheck the "Copy Projects into Workspace" checkbox and uncheck the projects you don't want and click "Finish".



6. Eclipse will build the projects and you can now browse, inspect and use Wonder source in Eclipse.



## Ant Builds on your Development Machine



WOLips' "Incremental Builder" is an incredibly helpful feature during development, but it does things that the Ant deployment build does not do. If you are **either** using WOLips to build your WebObjects Applications (*WOLips Ant Tools > Install*) **or** are [running Hudson/Jenkins](#) locally to do it, you **must** also build and install your workspace's Frameworks that your Application depends upon – including your own, Project Wonder's and any others. **The standard Ant build does NOT do this for you.**