

Overview-Objective-C to Java Transition

Windows

WebObjects 5 projects will require converting with the JavaConverter, whether or not they are already written entirely in Java. This is best run from a Bourne Shell, and is at \$NEXT_ROOT/Library/WebObjects/JavaApplications/JavaConverter.woa/JavaConverter.cmd. Run it with the full path of your chosen project and it will create a new project, whose name will be the original name with "Java" appended.

You should then be able to open the new project and browse the source code. The Java Converter has non-trivial tasks to perform, more so if you've come from Objective-C, so you'll want to read through your source code and check it's done a job you're happy with. As an alternative, you could search merely for its comments (labelled JC_WARNING or JC_ERROR) and review each one.

Note that the Makefile will be removed by the conversion process, and for some reason best not asked about, Project Builder will recreate the Makefile only if files are added or removed from the project - and not in the rather more obvious case where you try to build and there is no Makefile. So you'll need to add and remove a dummy class, for example, before your project will build.

EOModels will need converting to the JDBC adaptor. Since JDBC is itself a multiple-database protocol, the JDBC adaptor itself does not have the usual knowledge of external types - but if you give it an appropriate database URL and it can locate the appropriate plugin, it will then know which types are possible, so you can use the external type pulldown list to get them right. The JDBC adaptor and EOF 5 is, of course, a whole new learning curve of little fiddly bits, so expect to have to spend a bit of time experimenting and reading up in order to make all your database types work correctly, especially if it is complex. As the community (and the database vendors) discover more about the environment, this stage is becoming easier and easier.

Once your application is built you can run it, of course. Be aware WebObjects 5 uses some bizarre scripts to launch applications, and these are worse than usual on Windows, where the command line has a maximum length, so your classpath may not fit. My personal opinion (MalcolmCleaton) is that this is an extremely ill-conceived method for launching Java applications, and you'd be best off doing away with it as soon as possible, but that's another discussion.