

Taking Komodo for a Test Drive

by Jim Brandt

cbrandt@perlfoundation.org



At OSCON this year, ActiveState was passing out CDs with their Komodo IDE. I grabbed one from the booth and when I got back decided to give it a test run while working on a patch for a module. The patch was for the `Workflow` module on CPAN and it was fairly simple, so it seemed like a good opportunity to try out a new working environment.

I started this trial clean in the sense that I had no previous experience with Komodo and very limited experience with other IDEs. As people often say jokingly, “`xemacs` is my IDE.” So while I’m not able to comment on changes between this version and previous versions, I can offer my experience getting up to speed as a new user.

■ Installation

I installed Komodo from the CD on my Apple PowerBook with no trouble. The first launch was a bit slow, but this is due to some initial setup it does. Subsequent launches were much quicker. On the Mac, I can launch via the standard GUI icon or from the command line.

It requires Perl 5.6 or later to use the debugger, and some features, such as background syntax checking, require the ActiveState version of Perl, which is available for free from ActiveState. In my case, I installed nothing extra and Komodo defaulted to the version of `perl` it found in my path, which is the Mac OS X system `perl` (5.8.6). I can specify a particular `perl` interpreter in the Komodo Preferences.

The first thing I did after launching it was to select Komodo > Preferences to see what sorts of things I can customize. As you might expect, there are a ton of ways to customize Komodo. One setting jumped out at me was Editor > Key Bindings (Figure 4, page 26). As an `xemacs` user, I figured it would ease my transition if I could still rely on some of my muscle memory by keeping familiar key bindings. The options are Emacs, Vi, and Windows; I picked Emacs.

■ Setting up

Before I start working on a patch, I typically unpack the module in a sandbox directory and make sure the tests all pass. I created a new directory, called `working`, and unpacked the `Workflow` module. Then I opened Komodo and tried to determine if I could run the tests from within the IDE.

To leverage some of the features of an IDE, it typically needs to know about all of the files in your project. In this case, the project is the `Workflow` distribution, so I needed to tell Komodo about it. To do this, I selected File > New > New

Project. In the New Project dialog, I gave it the project name `Workflow` with the default extension `.kpf`. I selected the top level directory of the `Workflow` project, `Workflow-0.28`, and clicked Save. This created the new project and Komodo seemed to be aware of all the files in the distribution because the directory tree and all of the files popped up in the Project tab on the left (Figure 1, below). On page 25, I show the entire Komodo workspace in Figure 2.

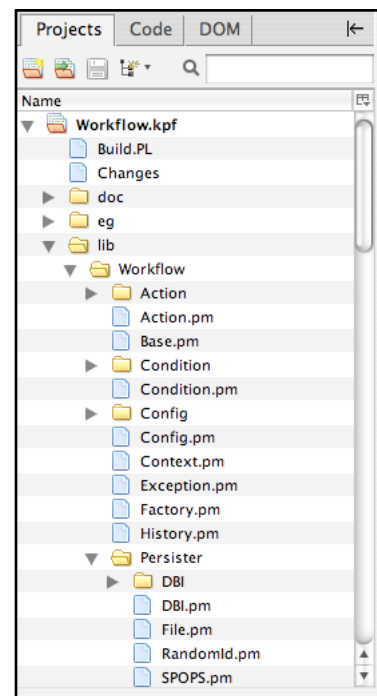


Figure 1: The Project tab of Komodo, showing the directory

■ Running tests

My next challenge was to figure out how to run the tests before I changed anything. Komodo doesn’t have a native way of doing this in the form of a Test button or menu, but it does support running commands within the IDE.

I selected Tools > Run Command and got the Run Command dialog (Figure 3, page 26). This dialog allows me to configure all the options for the command I want to run. I use the `prove` utility to run tests, so in the Run box I typed

```
prove --lib t
```