

# The journal of Apple technology.

**Volume Number: 19 (2003)**

**Issue Number: 4**

**Column Tag: Section 7**

## Tk: A Portable GUI Toolkit

### Expressiveness and Simplicity Count for a lot

*by Rich Morin*

Last month, I mentioned Tk, referring to it as an X11 toolkit. This month, I'd like to cover Tk in a bit more detail. I should begin, however, by clarifying that Tk is not just an X11 toolkit; in fact, it is a platform- and language-independent GUI toolkit.

Tk can be used with C, Perl, Python, Ruby, Tcl (Tk's original counterpart), and many other programming languages. It can be used with Aqua, PalmOS, X11, and assorted Microsoft APIs. In summary, any modern computer with the appropriate hardware probably has Tk support.

Tk also wins big on expressiveness and simplicity. Most GUI toolkits (e.g., Cocoa and assorted X11 "widget sets") are complex and have steep learning curves. Tk, in contrast, is remarkably simple to program, once a few basic concepts have been mastered. Although Tk may not have all the features of Cocoa, it supports most of the widgets you are likely to need; it also has ways to add new widgets, if you're feeling sufficiently motivated.

### Tcl/Tk

Because Tk started as a toolkit for the Tcl language, most of the Tk resources you'll find are actually for Tcl/Tk, the language/toolkit combination. Also, because Tcl/Tk is the focus of development, the other language bindings (e.g., Perl/Tk, Tkinter) occasionally lag behind. Finally, if you want to extend Tk, you'll probably find yourself coding some Tcl in the process.

In short, you should plan to learn a bit of Tcl, even if you intend to use Tk with some other language. Fortunately, this isn't particularly painful; Tcl syntax only takes about five minutes to learn (most of which you'll spend accepting the idea of how simple the syntax really is :-).

Pick up a copy of the Mac OS X Tcl/Tk port from:

[http://www.apple.com/downloads/macosx/unix\\_open\\_source/tcltk.html](http://www.apple.com/downloads/macosx/unix_open_source/tcltk.html)

Be sure to look at the ReadMe files; they contain a number of useful pointers and critical details. After the (dmg, mpkg) installation finishes, you should find a copy of "Wish Shell":

```
/Applications/Utilities/Wish Shell.app
```

When you start up the app, two windows will appear. The "Console" window allows you to run Tcl commands, viewing the output:

```
set a 123
```

```
123
() 2 % set b [expr $a * 2]
246
() 3 %
```

Tcl isn't a conventional language and it isn't processed in a conventional manner. Although some parts of Tcl syntax look a bit like C or Perl, the interpreter is simply performing a sequence of commands and string substitutions. Thus, the first line sets a variable ("a") to the text string ("123"). You may think of this as a numeric value, but the Tcl interpreter doesn't.

The second line does three things. First, it performs a "variable substitution", replacing "\$a" with the value "123". Next, it performs a "command substitution", replacing the expression "[expr \$a \* 2]" with the result (246). Finally, it sets "b" to the resulting value.

Now, let's bring Tk and the other (Wish Shell) window into play:

```
() 3 % button .b -text "Hello, world!" -command exit
.b
() 4 % pack .b
() 5 %
```

If all went as expected, the "Wish Shell" window should now contain a button labeled "Hello, world!". Pushing that button should cause the Wish Shell to exit (and both of its windows to disappear). In two lines of code, you have created and run a (tiny) windowing application!

If you simply want to program in Tcl/Tk, you are now in a position to do so. The results will look nicely Aquafied (Aquatic?) and you don't have to tell anyone how little code you wrote! If you become intrigued with Tcl/Tk, be sure to pick up John Ousterhout's excellent (if dated) "Tcl and the Tk Toolkit" (Addison-Wesley). John created both Tcl and Tk, so he's eminently qualified to say how they should be used.

## Perl/Tk

Although Tcl is syntactically interesting, extremely portable, and quite flexible, you may prefer (as I do) to do your actual programming in some other language. Perl/Tk allows Perl programmers to use Tk widgets, while keeping the rest of their code in Perl. As hinted above, analogous libraries are available for C, Python, Ruby, and other programming languages.

Having said this, I should note that Tk calls may not be a smooth fit into the syntax of the chosen language. Here is a Perl/Tk translation of our simple window example:

```
use Tk;
my ($b, $mw);
$mw = new MainWindow;
$b = $mw->Button(-text => 'Hello, world!',
               -command => sub { exit; });
$b->pack();
Mainloop();
```

This code is substantially larger, but the original Tk commands and arguments are still present (albeit buried in Object-Oriented Perl syntax :-). Several lines of overhead code bring in the Tk module, declare a couple of variables, create a window, and finally, start up the Tk event loop. Also note that we are no longer using the Tcl exit command; instead, we have created an anonymous Perl function which uses Perl's own exit command.

If you want to play with Perl/Tk, your first step should be to purchase a copy of "Learning Perl/Tk" (Nancy Walsh, O'Reilly). Buy Ousterhout's book, too, if you can; you won't regret it!

Now for the bad news: installing Perl/Tk on Mac OS X is quite complex and (worse) still in a state of flux. It is possible, with some difficulty, to make Perl/Tk work with Apple's X11 server and Developer Kit. I don't know of a way to get Perl/Tk to work with the native Tk port (as installed above), but I hope to have more information by the time this column is in print. Look for instructions on

<http://www.cfcl.com/rdm/section7/perltk.html>

I also suggest, for the brave-hearted, the following links:

---

**Rich Morin** has been using computers since 1970, Unix since 1983, and Mac-based Unix since 1986 (when he helped Apple create A/UX 1.0). When he isn't writing this column, Rich runs Prime Time Freeware ([www.ptf.com](http://www.ptf.com)), a publisher of books and CD-ROMs for the Free and Open Source software community. Feel free to write to Rich at [rdm@ptf.com](mailto:rdm@ptf.com).