

Re: Using foreach loop to create radiobutton menu

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2006-04/msg01343.html>

- *From:* Bryan Oakley <oakley@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 28 Apr 2006 15:27:35 GMT
-

aj@xxxxxxxxxx wrote:

I appreciate the significance of your answer, however, my problem persists. Here is the updated code...

```
menubutton range -text "Set Range" -underline 0 -bd 0 -menu {
foreach range { 1 2 3 4 5 6 7 8 9 10} "
radiobutton range$range -label $range -command {puts hello}
"
}
```

When this code executes, I get this error:

```
"..... - can't read "range": no such variable"
```

That's an interesting application of my advice. Unfortunately, it is wrong on many levels.

Let's get back to your original loop:

```
> foreach range { 1 2 3 4 5 6 7 8 9 10} {
> radiobutton range$range -label $range -command {
> puts "hello: $range"
> }
> }
```

.... and your original complaint was that you got an error saying that the variable range doesn't exist. The complaint comes, not when you create the radiobutton, but when you click on it. Is that correct?

The reason it doesn't exist is that when the `-command` actually runs, it runs in the global scope. Your variable 'range' is a local variable. That is the crux of the problem. Your foreach loop is not the problem, the problem is how you define the `-command`.

Now, there are (at least) two solutions. You might be tempted to just declare range as global, but that will just give you a different problem. Remember that the `-command` runs sometime later. So, imagine some arbitrary point in time the following script runs:

```
puts "hello: $range"
```

Re: Using foreach loop to create radiobutton menu

The only possible thing the interpreter can do is substitute the current value of range (or throw an error if range isn't global), which is likely the last value once your loop exited. Remember, this puts statement isn't running inside a loop; it is running some time in the future when the loop has long sense gone away.

So, using a global variable won't work. What you want is the value of \$range at the time the button is created. How do we get that? What are the fundamental rules of tcl quoting? You tried this:

```
... -command {puts "hello: $range"}
```

The curly braces mean that \$range will not be substituted. You want the variable to be substituted, so clearly we can't use curly braces here. Do you understand that? One choice is double quotes. That's problematic since your script itself uses double quotes, but let's try anyway:

```
... -command "puts \"hello: $range\""
```

Can you see how that will work? The double quotes means that \$range gets expanded while in the loop, long before the puts command actually runs. This will solve your problem, albeit in a somewhat unreadable manner.

Most seasoned programmers will agree that using the list command to build up a proper tcl command is a better solution:

```
... -command [list puts "hello: $range"]
```

Notice how we still use double quotes to make sure \$range is substituted, but by using the list command we don't have to protect the double quotes surrounding your message. List takes care of making sure we are feeding the proper number of arguments to the "puts" command.

An even better solution is to stick to the rule of thumb that says, always call a proc from a binding or -command. That rule isn't stated in any documentation but it's something we say in this newsgroup from time to time.

The solution then becomes:

```
... -command [list doSomething $range]
proc doSomething {n} {
  puts "hello: $n"
}
```

Notice how all quoting problems vanish. It becomes clear that \$range is to be substituted at the time the radiobutton is created, and that value is passed to a proc. And, because we know that your ultimate problem is not to print "hello: \$n" but something more complex, you now have a nice, self-contained environment in which to write as complex a script as you want.

If you have other variables that need to be expanded at the time the radiobutton is created, you put them on the command line. If they need to be expanded at the time the script is run, you put them in the script.

--

Bryan Oakley

Re: Using foreach loop to create radiobutton menu

<http://www.tclscripting.com>