

Re: invoked "break" outside of a loop

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2006-09/msg00794.html>

- *From:* Darren New <dnew@xxxxxxxxxxx>
 - *Date:* Sun, 24 Sep 2006 16:59:56 GMT
-

Russell Trleaven wrote:

```
Sorry to belabor the point but I was hoping someone would explain why
"break"
"uplevel break"
and
"namespace eval :: break"
don't work like I thought they might.
```

As you've learned, [break] is simply a return with a special code attached. Hence, when you invoke [break], it invokes a command that returns that special code, which is caught inside your proc, which leads to the error. I.e., the proc is converting the return from [break] into a different type of return, since [break] is invoked several levels down the return stack from the [for] loop.

For example...

```
proc AA {} {
BB
puts "got here AA"
|
```

```
proc BB {} {
if {[catch {
uplevel "set x 0"
}]} { puts "caught something" }
puts "got here BB"
}
```

Now, given that [set x 0] there return no error, nothing prints.

Were the [set x 0] to be replaced with an error call, you would expect the catch to catch it, yes?

Hence, since [break] is simply a kind of error, that error return is returned to the caller for use in [catch]. Since there is no [catch], it propagates up to the environment of the proc, where proc is about to return, and says "Hey, why are you throwing a [break] without an appropriate [catch]?"

Does that help?

Re: invoked "break" outside of a loop

—

Darren New / San Diego, CA, USA (PST)
Just because you find out you are
telepathic, don't let it go to your head.

.