

Re: Great SWT Program

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-11/msg03495.html>

- *From:* twerpinator@xxxxxxxxxx
 - *Date:* Wed, 21 Nov 2007 12:05:13 -0800 (PST)
-

On Nov 19, 5:53 am, b...@xxxxxxxxxxxx (Bent C Dalager) wrote:

You misunderstand me -- deliberately again, I expect.

When I start up screen, it automatically launches five applications in different screen windows.

That is odd behavior for this software to have. At least from a standpoint of user satisfaction. It would make marketing sense, except that there's a worldwide market for maybe ten copies of this software.

Consider web browsers. Ordinarily people will go to different sites or do different things each time they run one, so the most logical starting point is either a blank page or a search engine, particularly Google. Yet they tend to come out-of-the-box configured to go someplace else, such as MSN or such, generally a site affiliated with the browser's maker. But I don't see such behavior benefiting screen's developers somehow, so having it go to five pre-programmed places seems senseless. And from a user standpoint, it should just start up a single internal terminal with a command prompt and wait for the user to decide what else it is to do. In fact I'm fairly sure this is exactly the behavior I observed with an older version of it some time in the past, so opening five terminals automatically and launching specific processes that aren't the system command interpreter in each of them must be something relatively recent.

The even stranger thing is that this being unix-type software it should be more configurable. I can configure Firefox for any desired start page or to start up with particular sites in tabs, though there isn't much point, since a single blank tab will do, pending my choosing a bookmark or using the google box in the upper right corner to launch a search. Likewise it should be possible for you to reconfigure screen to start up with only a single session open on a command prompt in your home directory, a central point that's reasonably convenient no matter what you plan to do today. Yet you haven't done this, so unless you always do the exact same thing all of the time, this would seem to indicate that it's stuck behaving the way

Re: Great SWT Program

you describe.

Then again, perhaps you do always do the exact same thing, although then all you would seem to need is one terminal that pops up running trn (or whatever it is you use to assault cljp with off-topic posts). :P

"screen vi blah.txt" will launch "vi blah.txt" in a new window.

Eh? Nesting screen instances? That sounds like an even bigger pain, even assuming it's fully reentrant. Won't the outer one eat the same keys that you'd need the inner one to see to operate the inner one?

(I'm overlooking your rather questionable use of the phrase "new window" to describe a primitive ancestral process that resembles opening a new window in much the way that a fish resembles a human being. Unfortunately we appear to lack a generic term analogous to "vertebrate" for describing all mechanisms, both crude and refined, for adding a multitasking session of some sort.)

And you reckon you have found a graphical windowing system that does /not/ have bugs?

I didn't say that. But none since the Windows 3.1 days where the windowing system is prone to hangs. WinXP may occasionally bluescreen, but that's associated with the same sorts of circumstances that would cause what you call a kernel panic over in unixland, generally hardware/driver problems. This is a possibility in any OS. The window system proper does not have such behavior. Windows 3.1's did, mind you; it could wedge solid under some circumstances. Thankfully it is no longer in widespread use. I wonder if the analogous bugs in screen have been fixed yet.

The excellence of screen in situations of misbehaving networks, as above, is that it will actually remember all your sessions for you and restore them when you log back in and and reattach to them.

That's not behavior I recall observing, probably because the circumstances were different: you're describing the network connection between your terminal and the remote machine running screen going kaput; if screen itself hangs or abends, presumably the session state doesn't get saved. Since in those instances noted above I couldn't kill it it was necessary to disconnect (or wait for the idle timeout to kick in) and reconnect, log back in, and since the old session was unresponsive it made me a new one. Which then ran into all kinds of

Re: Great SWT Program

trouble caused by the wedged session having been preserved: stale locks, for starters. Something on that system killed dead sessions after a fairly short time so I didn't have to bother myself with doing it manually, but for a few minutes things would be running double. The mail client didn't like this. The IRC client didn't mind, but for a while there'd be a silent copy of me on whatever server that was annoyingly sitting on my nick, which I'd have to wait for the server to boot with its idle timeout. And so forth.

Your killer feature becomes a source of pain when combined with a hang bug like described, as it just so happens.

Of course there's another problem with screen: it captures some keystrokes that might also be bound in other applications. ISTR C-w being one of these, used for its version of alt-tab. So much for using C-w in emacs then, eh?

Screen only captures C-a. I have remapped this to C-v since I like to use C-a in emacs. The installation you used may have been configured to capture C-w of course.

Still got to pick something. If all the apps you ever wanted to use inside it adhered to CUA, you could pick something relatively useless like C-x (and use shift-del for cut, or C-c del) that won't be bound to anything application-specific and it wouldn't impair your use of any application. Ditto if they all adhered to *some* common set of bindings, with at least one of those fairly superfluous. But unix text mode apps are like pre-CUA MS-DOS ones: they all have their own different and idiosyncratic ones and any choice at all will inevitably collide with something you want to use in at least one of them. C-v presumably is the least worst choice given the apps you tend to use, but ... ouch.

bindings, and there's no set of bindings a screen-like program could use that wouldn't mask something useful in some app or another you might run from inside it.

Which would explain why it's configurable.

That doesn't get rid of the problem; it just lets you move it around, and lets you put it where it's least inconvenient. See above.

Yet, there are those that do manage to jump these almost impossible hurdles.

Re: Great SWT Program

I don't doubt it can be done, but it's more burden placed on the user that could be avoided and, with Windows and the Mac and, I expect, KDE and Gnome, does get avoided.

.