
Subject: Re: I for Interactive Programming? (was: widget_control and group_leader)

Posted by [John-David T. Smith](#) on Fri, 05 Jan 2001 00:22:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

Vapuser wrote:

- >
- > For those willing to learn emacs and Carsten Dominik's marvelous
- > idlwave_shell mode, you would have no more problems with saving the
- > results of your 'tapping' at the keyboard. Just start up an instance
- > of idlwave_shell in emacs and tapple to your hearts delight, then
- > write the idlwave-shell buffer out to a file. Or, kill/yank
- > (i.e. cut/paste) it to another buffer, edit it and write that out to a
- > file.
- >
- > This is almost the *only* way I develop code anymore, I try stuff
- > out in emacs/idlwave-shell and then cut/paste it to the 'work in
- > progress' file. (doesn't work for widgets programming, of course, but
- > in this area it's the debugging capabilities, which are totally
- > awesome *d00d* that recommend it)
- >
- > No more need to remember to turn journalling on or to set
- > variables to assure that you save enough of your work.
- >
- > It does have a steep learning curve, especially if you're not used to
- > emacs or you're on a Windows/Mac system. If that's the case,
- > journalling is probably the best bet. But if you're moderately used to
- > emacs and on a unix system, emacs/idlwave-shell mode simply can't be
- > beat (IMHO) and it's well worth the effort!
- >
- > Kudo's to Carsten for his fine work!

Hear Hear! I've got something even better (from Carsten, of course). I'm not sure if this has been put into IDLWAVE yet as a more palatable form (I vote for it), but if you have version >4.7b (meaning when 4.8 comes out, if you're leary of alpha's) you can simply use in your .emacs:

```
(add-hook 'idlwave-shell-mode-hook
  (lambda ()
    (set (make-local-variable 'comint-input-ring-file-name)
      "~/.idlwhist")
    (comint-read-input-ring))))
```

```
(add-hook 'idlwave-shell-sentinel-hook 'comint-write-input-ring)
```

I.e. just set up a "comint input ring filename" to ~/.idlwhist (or whatever you like), and ensures it is read in when the shell starts (in

the `idlwave-shell-mode-hook`), and written out after it exits (in the `idlwave-shell-sentinel-hook`).

This way, you always get all of your input history (which is actually maintained by `comint`, not IDL), between sessions, etc. Now restarting IDL is not nearly so painful. The commands are still right there, at your fingertips.

Never wonder where your command history went again. And for the really brave, try:

```
(setq comint-input-ring-size 1024)
```

outside of your `idlwave-shell-mode-hook` (since `comint` runs before `idlwave-shell`). Now you'll have the silly command you tried 3 weeks ago, somewhere in there. By the way, try `[C-c C-l]` to see your entire history in another buffer, with ability to scroll through it and click on the command you're after (or cut and paste between many such commands). `.idlwhist` is a plain file, which you can search through using `grep` or a pager or however you like, if you don't want to load IDL.

Good luck,

JD

Subject: Re: I for Interactive Programming? (was: `widget_control` and `group_leader`)

Posted by [Richard French](#) on Fri, 05 Jan 2001 04:00:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

JD Smith wrote:

```
> Hear Hear! I've got something even better (from Carsten, of course).
> I'm not sure if this has been put into IDLWAVE yet as a more palatable
> form (I vote for it), but if you have version >4.7b (meaning when 4.8
> comes out, if you're leary of alpha's) you can simply use in your
> .emacs:
>
> (add-hook 'idlwave-shell-mode-hook
> (lambda ()
> (set (make-local-variable 'comint-input-ring-file-name)
> "~/idlwhist")
> (comint-read-input-ring)))
>
> (add-hook 'idlwave-shell-sentinel-hook 'comint-write-input-ring)
>
```

Finally taking the plunge and trying to learn this system. I'm using a PC with xterm software on it to connect to my unix workstation. Trying to figure out how to use the Alt-TAB command inside emacs- Alt-TAB is a windows thing that lets you hop between running programs. I tried to use ESC-TAB but it does not seem to do the right completion. Instead, I get
Use symbol help mode:
and I am not sure what to do next. I've tried to find a way to bind a different key to be my ALT key but I haven't succeeded. Any emacs users out there who can help me out here? As should be obvious, I have not used emacs before.

Thanks,
Dick

Subject: Re: I for Interactive Programming? (was: widget_control and group_leader)

Posted by [Vapuser](#) on Fri, 05 Jan 2001 15:54:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

JD Smith <jdsmith@astro.cornell.edu> writes:

<snip>

> Hear Hear! I've got something even better (from Carsten, of course).
> I'm not sure if this has been put into IDLWAVE yet as a more palatable
> form (I vote for it), but if you have version >4.7b (meaning when 4.8
> comes out, if you're leary of alpha's) you can simply use in your
> .emacs:
>
> (add-hook 'idlwave-shell-mode-hook
> (lambda ()
> (set (make-local-variable 'comint-input-ring-file-name)
> "~/idlwhist")
> (comint-read-input-ring)))
>
> (add-hook 'idlwave-shell-sentinel-hook 'comint-write-input-ring)
>
> I.e. just set up a "comint input ring filename" to ~/idlwhist (or
> whatever you like), and ensures it is read in when the shell starts (in
> the idlwave-shell-mode-hook), and written out after it exits (in the
> idlwave-shell-sentinel-hook).
>
> This way, you always get all of your input history (which is actually
> maintained by comint, not IDL), between sessions, etc. Now restarting
> IDL is not nearly so painful. The commands are still right there, at

> your fingertips.
>
> Never wonder where your command history went again. And for the really
> brave, try:
>
> (setq comint-input-ring-size 1024)
>
> outside of your idlwave-shell-mode-hook (since comint runs before
> idlwave-shell). Now you'll have the silly command you tried 3 weeks
> ago, somewhere in there. By the way, try [C-c C-l] to see your entire
> history in another buffer, with ability to scroll through it and click
> on the command you're after (or cut and paste between many such
> commands). .idlwhist is a plain file, which you can search through
> using grep or a pager or however you like, if you don't want to load
> IDL.
>
> Good luck,
>
> JD

Man! I have to put you on my Christmas Card list!

Thanks!

whd

--

William Daffer: 818-354-0161: William.Daffer@jpl.nasa.gov

Subject: Re: I for Interactive Programming? (was: widget_control and group_leader)

Posted by [Vapuser](#) on Fri, 05 Jan 2001 16:42:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Richard G. French" <rfrench@wellesley.edu> writes:

<snip>

> Finally taking the plunge and trying to learn this system. I'm
> using a PC with xterm software on it to connect to my unix
> workstation. Trying to figure out how to use the Alt-TAB command
> inside emacs- Alt-TAB is a windows thing that lets you hop between
> running programs. I tried to use ESC-TAB but it does not seem to do
> the right completion. Instead, I get
>
> Use symbol help mode:
>
> and I am not sure what to do next. I've tried to find a way

- > to bind a different key to be my ALT key but I haven't succeeded.
- > Any emacs users out there who can help me out here? As should be
- > obvious, I have not used emacs before.
- >
- > Thanks,
- > Dick

Let me see if I understand. 'Alt-Tab' is a *windows* command which has become disabled when the active window is the xterm window in which you're running emacs and you want the windows functionality back? Is that the problem? If this is not a correct rephrasing of your problem, what are you expecting from the ALT-TAB key sequence? That sequence isn't bound to anything in my (20.4.2) version of emacs.

I'm going to proceed on the assumption that what you want is the window's functionality back.

What happens when you make some other window the active window? Does the ALT-TAB functionality return?

Which version of emacs? And what xserver software are you running. My guess is that it's the xserver software that's intercepting the ALT key because that key is significant to many xservers (it's how the system menus on many X windows are accessed from the keyboard) For instance, on my system, an SGI, alt-F9 will minimize the current active window. It's been years since I've run any such software, but I remember it allowing you to set the function of various control keys as part of a specific connection configuration. If I've got your question right, and what you want is the Windows functionality back, you really want the Windows GUI to intercept the 'Alt-Tab' so that it never gets sent through the xserver software to emacs *at all*. Perhaps 'unsettling' it in the xserver configuration???

Be careful what you wish for, however. There might be some serious side effects from taking the ALT key away from emacs, not to mention the xserver. I've never seen any xserver software on windows (or macs, for that matter) that didn't have some ... ummm... idiosyncracies. If all you're missing is the ability to ALT-TAB your way around the active Win9x windows, I'd consider that a fair price.

But that's just me.

William Daffer

--

William Daffer: 818-354-0161: William.Daffer@jpl.nasa.gov
