
Subject: Re: Tab completion of variables and routines at the IDL command line
Posted by [Conor](#) on Tue, 10 Jul 2007 12:43:13 GMT

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

On Jul 10, 4:23 am, Allan Whiteford

<allan.rem...@phys.remove.strath.ac.remove.uk> wrote:

> Dear all,
>
> Yesterday I got annoyed enough with the lack of tab completion inside a
> standard UNIX tty session (i.e. not the development environment) that I
> tried to implement something which I've suspected for a while should
> give me tab completion.
>
> Anyway, turned out it worked, after a fashion.
>
> (For background on tab completion if you don't know what I'm talking
> about see, e.g.:
>
> http://www.cmp.liv.ac.uk/docs/linux_guide/node28.html
>
> chances are though, if you didn't know what it was then you probably
> aren't too bothered about having it.)
>
> I now have an IDL command line that will tab complete variables in the
> current scope, compiled functions (and append a '(' to them) and
> compiled procedures (and append a ',' to them). System procedures and
> functions are also handled.
>
> What it DOESN'T do (yet) is:
>
> Tab complete structure tags.
>
> Realise you can't have a function call as the first item on the command
> line.
>
> Realise you can't have a procedure call as anything except the first
> item (or, I guess, after a '&').
>
> Tab complete keywords to routines. *
>
> Retain support for saving history between sessions.
>
> Handle Ctrl-C interrupts as gracefully as it should. *
>
> Expand on uncompiled routines. *
>
> Full understanding of when a new tab completion can start (e.g. after a
> ',' or a '(' - at the moment it's on space).

>
> Look at IDL system variables (or somewhere) to get options for how the
> tab completion should work.
>
> Tab complete filenames when appropriate (e.g. after a ".r").
>
> Gracefully handle exiting from inside an IDL procedure (by calling the
> "quit" routine). *
>
> With the exception of the starred items above, all of the others are
> really my immediate wishlist can/will probably be implemented this week.
>
> Now, the question...
>
> Is anyone else interested in this? If so then you're welcome to the code
> but if there is sufficient interest (more than five people, say) then
> I'll attempt to produce compilation instructions (of course, first I
> have to understand the compilation step) and possibly binaries for
> different versions of IDL.
>
> It could be that most people are using either IDLDE or IDLWAVE in which
> case I'm one of the few luddites left unwilling to abandon a standard
> TTY. Also, if this feature already exists in a standard TTY then break
> the news to me gently :).
>
> I should say that the way in which this is implemented is truly awful
> and there has to be a better way to do it. However, I gave up waiting
> for someone smarter (hint hint ITTVIS) to do it and just did it myself.
>
> As an additional warning I should point out that I've not written any
> serious C in years and that all of the malloc'ing and the realisation I
> could put most of the code in a macro seemed a lot more obvious after a
> glass of wine. Use at your peril :).
>
> Let me know if you're interested either by posting to the group or by
> e-mail and I'll assess demand. Depending on who is interested I'll
> perhaps do a pre-release to people who are better equipped than me to
> assess how dangerous the code is before I distribute it to many people.
> Please also indicate if you think you can help in a testing or code
> evaluation step.
>
> Thanks,
>
> Allan

That sounds quite intriguing, I'd certainly be willing to give it a
shot (sounds like it could be very useful). I have a strong dislike
of IDLDE, and I've never used IDLWAVE, so a plain ol' command line is

how I get things done.
