## Subject: Re: Readline and/or easy command line history access Posted by JDS on Mon, 27 Apr 2009 19:55:32 GMT

View Forum Message <> Reply to Message

```
On Apr 27, 9:30 am, sav...@nsidc.org wrote:
> Adam.
>
> Keflavich <keflav...@gmail.com> writes:
>> Is there any sort of readline support in IDL more advanced
>> than a simple command history? I can hit 'up' at the command line to
>> retrieve old commands, but this is tedious and sometimes difficult in
>> large interactive sessions.
>
> I'm not sure what you're using either the IDE or idlwave-shell, but JD Smith
> wrote idlwave-shell routines for exactly this. Just drop these into you
 .emacs file.
 Hope this helps.
>
> Matt
  ;; Set up some functions that we need to define before we can use them.
 :: the actual author is
> ;; JD Smith <jdsmith@****.edu>
> :: These commands make the up/down arrows do regular expression searches that
> ;; include whatever is already on the idlwave-shell's command line.
> ;; So if you did a plot command ten commands ago, you would put plot on the
 :: shell line.
> ;; IDL> plot,
> ;; and then hit the up arrow* to get to the command
 ;; *(if you bind this commmand to the up arrow like we do later in this file)
>
  (defvar ids-search-regexp nil)
>
  (defun jds-search-pre-command-hook ()
   (unless (or (eq this-command 'jds-history-search-up)
>
           (eg this-command 'jds-history-search-down))
>
     (remove-hook 'pre-command-hook 'jds-search-pre-command-hook t)
>
     (setq jds-search-regexp nil)))
>
  (defun jds-history-search (dir)
   (when (null ids-search-regexp)
>
     (let ((str (buffer-substring (point) (save-excursion (comint-bol nil)
>
                                      (point)))))
      (if (or (string= str "") (not (looking-at "\\s-*$")))
```

```
(setg ids-search-regexp ".")
>
       ;; This is what JDS had.
>
       ;; (setq jds-search-regexp (concat "^" (regexp-quote str)))
>
       (setq jds-search-regexp (regexp-quote str)))
>
      (add-hook 'pre-command-hook 'jds-search-pre-command-hook nil t)))
>
   (comint-previous-matching-input jds-search-regexp dir))
>
>
  (defun jds-history-search-up ()
>
   (interactive)
>
   (ids-history-search 1))
>
>
> (defun jds-history-search-down ()
   (interactive)
>
   (jds-history-search -1))
```

This a just a subset of the command line searching already built into IDLWAVE (courtesy of COMINT mode). I use M-r to search back using any regular expression, C-c C-l to show a selectable list of prior commands, middle click on any old input line in the shell to paste it to the current, etc.

As for Matthew's customization, recent versions of Emacs have a new command C-c M-r to do just what this .emacs stuff does (so it's no longer needed). For an easier binding, I'd probably stay away from the up arrow for this (multi-line editing is built-in to the coming IDLWAVE release), but might try:

```
(add-hook 'idlwave-shell-mode-hook
  (lambda ()
    (local-set-key [M-up] 'comint-previous-matching-input-from-input)
    (local-set-key [M-down] 'comint-next-matching-input-from-input)))
```

to have Meta-up and Meta-down do this. Note that it works on input up to the cursor, so position your cursor on a long command to cap the section you want to search, and fire. Successive searches will do the right thing.

JD

P.S. IDLWAVE is now working with the upcoming IDL help system, with direct linking to keyword help sections, etc.