Filename Completion

Filename completion lets you type just the first part of a filename and have the shell fill in the rest. The variations are: using the F key for basic filename completion, the D key if you want all the duplicates listed or Tab or Shift-Tab (BackTab) to move forward or backward through the list one-by-one.

Key	Meaning
Alt-F or Ctrl-F	Filename completion. Appending the "`*`" wildcard character onto the end, use the previous word as a wildcard pattern. If it matches a single file, substitute it in with a space following. (If you *don't* want a space following, use the `-N` option when you start the C shell.) If there were multiple matches, but they all had some common front-part that fully "used up" the pattern, substitute in just that common front-part and show it in the color specified by the `DUPLICATES` variable (default is green). If substitution wasn't possible, highlight the pattern in the color specified by the `MATCHFAIL` variable (default is bright red). (Any highlighting color is turned off when you press the next keystroke.)
Alt-D or Ctrl-D	Duplicate completions. Same wildcarding, but if there are multiple matches, show them all with a space following. (Here again, if you *don't* want a space following, use the `-N` option when you start the C shell.) If there were no matches, highlight the pattern in the color specified by the `MATCHFAIL` variable (default is bright red).
<Tab>	Next filename. Move one-by-one through the list of matching filenames. After the last, paste the original back in place, highlighting with the `MATCHFAIL` color, then continue, with the next Tab, cycling through the list again. (To type an ordinary tab character, use Ctrl-<Tab>. Alternately, if the C shell is started with the `-T` option, the Tab key generates a plain tab character and Ctrl-<Tab> is the filename completion key.)
Shift-<Tab>	Previous filename. Same as Tab, but cycling in reverse through the list.

Command Line Editing

Key	Meaning
<Enter>	Accept the command as typed. Move to the end (if not there already) and carriage return to a new line.
<Home>	Beginning of command line.
<End>	End of command line.
↑	Up one command in the history list. Each time it's pressed, it displays the preceding entry in the history list. Any " !..." or "%..." history references in the original text will have been fixed up unless it was the immediately preceding command and it had one these references that failed. If already at the first entry, the command line is highlighted in bright red.
↓	Down one command line in the history list. If already at the latest entry, the command line is highlighted in bright red.
←	One character left.
→	One character right.
Ctrl-<Home>	Move to the upper-leftmost character in the current screenful if the command is long enough that it actually wraps across several screens.
Ctrl-<End>	Move to the lower-rightmost character in the current screenful.
Ctrl-↑	Up one row on the screen if the command is long enough that it runs over a row.
Ctrl-↓	Down one row on the screen.
Ctrl-←	Backup word.
Ctrl-→	Forward word.
Alt-<Home>	Delete all preceding characters on the command line.
Alt-<End>	Delete all following characters.
Alt-↑	Delete up one row on the screen if the command runs over a row.
Alt-↓	Delete down one row.
Alt-← Ctrl-<Backspace>	Delete preceding word.
Alt-<Insert>	Toggle insert/overstrike mode. When inserting, the cursor is slightly thicker.
Ctrl-<Insert>	Insert the next word from the last section of deleted text. When it reaches the end of the deleted text, it starts over.
Alt-<Insert>	Insert all the rest of the previously deleted text.
<PageUp>	Backup to one past the last history reference. (Repeatedly typing <PageUp> <Enter> is a convenient way of picking up a whole series of commands from history.)
<PageDown>	Forward to the newest entry in the history list.
<Esc>	Clear the command line.

Note: Users lacking separate arrow keys must press Ctrl-Shift instead of Alt

History Recall

History recall allows a previous statement to be quickly recalled and re-executed. It’s a very fast shorthand, especially in the edit/compile/debug loop or to fix a typo. For convenience, “`!`” is taken as an ordinary character if followed by white space, “=“, “~” or “`(`“.

If you want, you can choose different characters to introduce history references by changing the `histchars` variable.

Command	Meaning
`!!`	Last command
`!^`	First argument word of last command
`!$`	Last word of last command
`!*`	All arguments of last command
`!`*n*	Command *n*
`!-`*n*	*n*th command from the last
`!`*str*	Last command starting with *str*
`!?`*str*`?`	Last command containing *str*
`%`*str1*`%`*str2*`%`	Substitute *str2* for *str1* in last command. (Used only at the beginning of a line.)

Command Completion

Command completion lets you type just part of a previous command and have the shell fill in the rest. As with filename completion, if no match is found, color highlighting as defined by the `MATCHFAIL` variable (default is bright red) will be used. Consecutive depressions cause the search to continue on back through the history list.

Key	Meaning
Ctrl-<Enter>	Search for the last command that starts with the characters in the previous word.
Alt-<Enter>	Search for the last command that contains the characters in the previous word anywhere on the command line.