



Linux Ubuntu 26.04 Manual Pages on command 'lesskey.1'

\$ man lesskey.1

LESSKEY(1) General Commands Manual LESSKEY(1)

NAME

lesskey - customize key bindings for less

SYNOPSIS (deprecated)

lesskey [-o output] [--] [input]

lesskey [--output=output] [--] [input]

lesskey -V

lesskey --version

SCOPE

This document describes the format of the lesskey source file, which is used by less version 582 and later. In previous versions of less, a separate program called lesskey was used to compile the lesskey source file into a format understood by less. This compilation step is no longer required and the lesskey program is therefore deprecated, although the file format remains supported by less itself.

DESCRIPTION

A lesskey file specifies a set of key bindings and environment variables to be used by subsequent invocations of less.

FILE FORMAT

The input file consists of one or more sections. Each section starts with a line that identifies the type of section. Possible sections are:

#command

Customizes command key bindings.

#line-edit

Customizes line-editing key bindings.

`#env` Defines environment variables.

Blank lines and lines which start with a hash mark (#) are ignored, except as noted below.

COMMAND SECTION

The command section begins with the line

`#command`

If the command section is the first section in the file, this line may be omitted. The com?

mand section consists of lines of the form:

`string <whitespace> action [extra-string] <newline>`

Whitespace is any sequence of one or more spaces and/or tabs. The string is the command key(s) which invoke the action. The string may be a single command key, or a sequence of up to 15 keys. The action is the name of the less action, from the list below. The characters in the string may appear literally, or be prefixed by a caret to indicate a control key. A backslash followed by one to three octal digits may be used to specify a character by its octal value. A backslash followed by certain characters specifies input characters as follows:

`\b` BACKSPACE (0x08)

`\e` ESCAPE (0x1B)

`\n` NEWLINE (0x0A)

`\r` RETURN (0x0D)

`\t` TAB (0x09)

`\k` followed by a single character represents the char(s) produced when one of these keys is pressed:

`\kb` BACKSPACE (the BACKSPACE key)

`\kB` ctrl-BACKSPACE

`\kd` DOWN ARROW

`\kD` PAGE DOWN

`\ke` END

`\kh` HOME

`\ki` INSERT

`\kl` LEFT ARROW

`\kL` ctrl-LEFT ARROW

`\kr` RIGHT ARROW

\kR ctrl-RIGHT ARROW

\kt BACKTAB

\ku UP ARROW

\kU PAGE UP

\kx DELETE

\kX ctrl-DELETE

\k1 F1

A backslash followed by any other character indicates that character is to be taken literally. Characters which must be preceded by backslash include caret, space, tab, hash mark and the backslash itself.

An action may be followed by an "extra" string. When such a command is entered while running less, the action is performed, and then the extra string is parsed, just as if it were typed in to less. This feature can be used in certain cases to extend the functionality of a command. For example, see the "{" and ":t" commands in the example below. The extra string has a special meaning for the "quit" action: when less quits, the ASCII value of the first character of the extra string is used as its exit status.

EXAMPLE

The following input file describes the set of default command keys used by less. Documentation on each command can be found in the less man page, under the key sequence which invokes the command.

#command

\r forw-line

\n forw-line

e forw-line

j forw-line

\kd forw-line

^E forw-line

^N forw-line

k back-line

y back-line

^Y back-line

^K back-line

^P back-line

J	forw-line-force
K	back-line-force
Y	back-line-force
d	forw-scroll
^D	forw-scroll
u	back-scroll
^U	back-scroll
\40	forw-screen
f	forw-screen
^F	forw-screen
^V	forw-screen
\kD	forw-screen
b	back-screen
^B	back-screen
\ev	back-screen
\kU	back-screen
z	forw-window
w	back-window
\e\40	forw-screen-force
F	forw-forever
\eF	forw-until-hilite
R	repaint-flush
r	repaint
^R	repaint
^L	repaint
\eu	undo-hilite
\eU	clear-search
g	goto-line
\kh	goto-line
<	goto-line
\e<	goto-line
p	percent
%	percent

<code>\e(</code>	left-scroll
<code>\e)</code>	right-scroll
<code>\kl</code>	left-scroll
<code>\kr</code>	right-scroll
<code>\e{</code>	no-scroll
<code>\e}</code>	end-scroll
<code>{</code>	forw-bracket {}
<code>}</code>	back-bracket {}
<code>(</code>	forw-bracket ()
<code>)</code>	back-bracket ()
<code>[</code>	forw-bracket []
<code>]</code>	back-bracket []
<code>\e^F</code>	forw-bracket
<code>\e^B</code>	back-bracket
<code>G</code>	goto-end
<code>\e></code>	goto-end
<code>></code>	goto-end
<code>\ke</code>	goto-end
<code>\eG</code>	goto-end-buffered
<code>=</code>	status
<code>^G</code>	status
<code>:f</code>	status
<code>/</code>	forw-search
<code>?</code>	back-search
<code>\e/</code>	forw-search *
<code>\e?</code>	back-search *
<code>n</code>	repeat-search
<code>\en</code>	repeat-search-all
<code>N</code>	reverse-search
<code>\eN</code>	reverse-search-all
<code>^O^N</code>	osc8-forw-search
<code>^On</code>	osc8-forw-search
<code>^O^P</code>	osc8-back-search

^Op	osc8-back-search
^O^O	osc8-open
&	filter
m	set-mark
M	set-mark-bottom
\em	clear-mark
'	goto-mark
^X^X	goto-mark
E	examine
:e	examine
^X^V	examine
:n	next-file
:p	prev-file
t	next-tag
T	prev-tag
:x	index-file
:d	remove-file
-	toggle-option
:t	toggle-option t
s	toggle-option o
## Use a long option name by starting the	
## extra string with ONE dash; eg:	
## s toggle-option -log-file\n	
_	display-option
	pipe
v	visual
!	shell
#	pshell
+	firstcmd
H	help
h	help
V	version
0	digit

1	digit
2	digit
3	digit
4	digit
5	digit
6	digit
7	digit
8	digit
9	digit
q	quit
Q	quit
:q	quit
:Q	quit
ZZ	quit

PRECEDENCE

Commands specified by lesskey take precedence over the default commands. A default command key may be disabled by including it in the input file with the action "invalid". Alternatively, a key may be defined to do nothing by using the action "noaction". "noaction" is similar to "invalid", but less will give an error beep for an "invalid" command, but not for a "noaction" command. In addition, ALL default commands may be disabled by adding this control line to the input file:

```
#stop
```

This will cause all default commands to be ignored. The #stop line should be the last line in that section of the file.

Be aware that #stop can be dangerous. Since all default commands are disabled, you must provide sufficient commands before the #stop line to enable all necessary actions. For example, failure to provide a "quit" command can lead to frustration.

LINE EDITING SECTION

The line-editing section begins with the line:

```
#line-edit
```

This section specifies new key bindings for the line editing commands, in a manner similar to the way key bindings for ordinary commands are specified in the #command section. The line-editing section consists of a list of keys and actions, one per line as in the example

below.

EXAMPLE

The following input file describes the set of default line-editing keys used by less:

```
#line-edit

\t      forw-complete
\17     back-complete
\et     back-complete
^L      expand
^V      literal
^A      literal
\el     right
\kr     right
\eh     left
\kl     left
\eb     word-left
\ekl    word-left
\ew     word-right
\ekr    word-right
\ei     insert
\ex     delete
\kx     delete
\eX     word-delete
\ekx    word-delete
\eb     word-backspace
\e0     home
\kh     home
\e$     end
\ke     end
\ek     up
\ku     up
\ej     down
^G      abort
```

The environment variable section begins with the line

```
#env
```

Following this line is a list of environment variable assignments. Each line consists of an environment variable name, an equals sign (=) and the value to be assigned to the environment variable. White space before and after the equals sign is ignored. Variables assigned in this way are visible only to less. If a variable is specified in the system environment and also in a lesskey file, the value in the lesskey file takes precedence.

If the variable name is followed by += rather than =, the string is appended to the variable's existing value. This currently works only if any += lines immediately follow the same variable's original definition (with an = line), without any intervening definitions of other variables. It can append only to a variable defined earlier in the file; it cannot append to a variable in the system environment. The string is appended literally, without any extra whitespace added, so if whitespace is desired, it should be appended to the end of the preceding line. (It cannot be added to the beginning of the += string because space after the equals sign is ignored, as noted above.)

In the string after the = sign, a substring of the form \${NAME} is replaced with the value of the environment variable "NAME". The value of the variable may come from either the system environment, an earlier lesskey file, or an earlier definition in the current lesskey file. Simple text replacements can be performed by using the syntax \${NAME/STRING/REPL}. This replaces all instances of "STRING" in the named environment variable with the text "REPL". STRING is matched using a simple text comparison; no metacharacters are supported. An instance of slash or right curly bracket in STRING or REPL must be escaped by preceding it with two backslashes. If REPL is an empty string, all instances of STRING are removed. A slash immediately before the right curly bracket may be omitted. Multiple replacements may be performed by using the syntax \${NAME/STRING1/REPL1/STRING2/REPL2} and so on.

CONDITIONAL CONFIGURATION

If a line begins with #version followed by a relational operator and a version number, the remainder of the line is parsed if and only if the running version of less (or lesskey) matches the operator. This can be helpful if a lesskey file is used by different versions of less.

For example, suppose that a new command named 'sideways-search' is added in less version 777. Then the following line would assign the command to the Q key, but only in versions of less which support it. The line would be ignored by versions earlier than 777.

```
#version >= 777 Q sideways-search
```

These six operators are supported:

> Greater than

< Less than

>= Greater than or equal to

<= Less than or equal to

= Equal to

!= Not equal to

The `#version` feature is not supported in `less` and `lesskey` before version 594. In those older versions, all `#version` lines are ignored.

EXAMPLE

The following input file sets the `-i` and `-S` options when `less` is run and, on version 595 and higher, adds a `--color` option.

```
#env
## (Note that there must be a space at the end of the next line,
## to separate the --color option from the -S option.)
LESS = -i -S
#version >= 595 LESS += --color=Hkc
```

SEE ALSO

`less(1)`

WARNINGS

On MS-DOS and OS/2 systems, certain keys send a sequence of characters which start with a NUL character (0). This NUL character should be represented as `\340` in a `lesskey` file.

COPYRIGHT

Copyright (C) 1984-2024 Mark Nudelman

`less` is part of the GNU project and is free software. You can redistribute it and/or modify it under the terms of either (1) the GNU General Public License as published by the Free Software Foundation; or (2) the Less License. See the file `README` in the `less` distribution for more details regarding redistribution. You should have received a copy of the GNU General Public License along with the source for `less`; see the file `COPYING`. If not, write to the Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA. You should also have received a copy of the Less License; see the file `LICENSE`.

`less` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without

even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

AUTHOR

Mark Nudelman

Report bugs at <https://github.com/gwsw/less/issues>.

Version 668: 06 Oct 2024

LESSKEY(1)