

NAME

editrc — configuration file for editline library

SYNOPSIS

editrc

DESCRIPTION

The **editrc** file defines various settings to be used by the `editline(3edit)` library.

The format of each line is:

```
[prog:]command [arg ...]
```

command is one of the `editline(3edit)` builtin commands. Refer to **BUILTIN COMMANDS** for more information.

prog is the program name string that a program defines when it calls `el_init(3)` to set up `editline(3edit)`, which is usually *argv[0]*. *command* will be executed for any program which matches *prog*.

prog may also be a `regex(3)` style regular expression, in which case *command* will be executed for any program that matches the regular expression.

If *prog* is absent, *command* is executed for all programs.

BUILTIN COMMANDS

The **editline** library has some builtin commands, which affect the way that the line editing and history functions operate. These are based on similar named builtins present in the `tcsh(1)` shell.

The following builtin commands are available:

bind [-aeklrsv] [*key* [*command*]]

Without options and arguments, list all bound keys and macros, and the editor command or input string to which each one is bound. If only *key* is supplied, show the binding for that key or macro. If *key command* is supplied, bind the editor *command* to that key or macro.

The options are as follows:

- a** List or change key bindings in the `vi(1)` mode alternate (command mode) key map.
- e** Bind all keys to the standard GNU Emacs-like bindings.
- k** *key* is interpreted as a symbolic arrow key name, which may be one of ‘up’, ‘down’, ‘left’ or ‘right’.
- l** List all editor commands and a short description of each.
- r** Remove the binding of the key or macro *key*.
- s** Define a keyboard macro rather than a key binding or command macro: *command* is taken as a literal string and appended to the input queue whenever *key* is typed. Bound keys and macros in *command* are themselves reinterpreted, and this continues for ten levels of interpretation.
- v** Bind all keys to the standard `vi(1)`-like bindings.

The `editline(7edit)` manual documents all editor commands and contains more information about macros and the input queue.

key and *command* can contain control characters of the form ‘^*character*’ (e.g. ‘^A’), and the following backslashed escape sequences:

```

\a  Bell
\b  Backspace
\e  Escape
\f  Formfeed
\n  Newline
\r  Carriage return
\t  Horizontal tab
\v  Vertical tab
\nnn

```

The ASCII character corresponding to the octal number *nnn*.

‘\’ nullifies the special meaning of the following character, if it has any, notably ‘\’ and ‘^’.

echo*tc* [-**sv**] *arg* . . .

Exercise terminal capabilities given in *arg* If *arg* is ‘baud’, ‘cols’, ‘lines’, ‘rows’, ‘meta’, or ‘tabs’, the value of that capability is printed, with “yes” or “no” indicating that the terminal does or does not have that capability.

–**s** returns an empty string for non-existent capabilities, rather than causing an error. –**v** causes messages to be verbose.

edit [on | off]

Enable or disable the **editline** functionality in a program.

history *list* | *size* *n* | *unique* *n*

The *list* command lists all entries in the history. The *size* command sets the history size to *n* entries. The *unique* command controls if history should keep duplicate entries. If *n* is non zero, only keep unique history entries. If *n* is zero, then keep all entries (the default).

set*tc* *cap val*

Set the terminal capability *cap* to *val*, as defined in `termcap(5)`. No sanity checking is done.

set*ty* [-**a**] [-**d**] [-**q**] [-**x**] [+*mode*] [-*mode*] [*mode*] [*char=c*]

Control which tty modes that **editrc** won’t allow the user to change. –**d**, –**q** or –**x** tells **setty** to act on the ‘edit’, ‘quote’ or ‘execute’ set of tty modes respectively; defaulting to –**x**.

Without other arguments, **setty** lists the modes in the chosen set which are fixed on (‘+mode’) or off (‘-mode’). –**a** lists all tty modes in the chosen set regardless of the setting. With +*mode*, –*mode* or *mode*, fixes *mode* on or off or removes control of *mode* in the chosen set.

Setty can also be used to set tty characters to particular values using *char=value*. If *value* is empty then the character is set to `_POSIX_VDISABLE`.

tell*tc*

List the values of all the terminal capabilities (see `termcap(5)`).

ENVIRONMENT

EDITRC Names the default configuration file for the `editline(3edit)` library.

FILES

`~/editrc` Last resort, if no other file is specified, user configuration file for the `editline(3edit)` library.

SEE ALSO

`editline(3edit)`, `regex(3)`, `termcap(5)`, `editline(7edit)`

AUTHORS

The **editline** library was written by Christos Zoulas, and this manual was written by Luke Mewburn, with some sections inspired by `tcsh(1)`.