Rocky Enterprise Linux 9.2 Manual Pages on command 'dumpkeys.1'

$ man dumpkeys.1

DUMPKEYS(1) General Commands Manual DUMPKEYS(1)

NAME

dumpkeys - dump keyboard translation tables

SYNOPSIS

dumpkeys [-h --help -i --short-info -l -s --long-info -n --numeric -f

--full-table -1 --separate-lines -Sshape --shape=shape -t --funcs-only

-k --keys-only -d --compose-only -ccharset --charset=charset -v --ver?

bose -V --version]

DESCRIPTION

dumpkeys writes, to the standard output, the current contents of the

keyboard driver's translation tables, in the format specified by

keymaps(5).

Using the various options, the format of the output can be controlled

and also other information from the kernel and the programs dumpkeys(1)

and loadkeys(1) can be obtained.

OPTIONS

-h --help

Prints the program's version number and a short usage message to

the program's standard error output and exits.

-i --short-info

Prints some characteristics of the kernel's keyboard driver. The items shown are:

Keycode range supported by the kernel

This tells what values can be used after the keycode key? word in keytable files. See keymaps(5) for more informa? tion and the syntax of these files.

Number of actions bindable to a key

This tells how many different actions a single key can output using various modifier keys. If the value is 16 for example, you can define up to 16 different actions to a key combined with modifiers. When the value is 16, the kernel probably knows about four modifier keys, which you can press in different combinations with the key to ac? cess all the bound actions.

Ranges of action codes supported by the kernel

This item contains a list of action code ranges in hexa? decimal notation. These are the values that can be used in the right hand side of a key definition, ie. the vv's in a line

keycode xx = vv vv vv vv

(see keymaps(5) for more information about the format of key definition lines). dumpkeys(1) and loadkeys(1) sup? port a symbolic notation, which is preferable to the nu? meric one, as the action codes may vary from kernel to kernel while the symbolic names usually remain the same. However, the list of action code ranges can be used to determine, if the kernel actually supports all the sym? bols loadkeys(1) knows, or are there maybe some actions supported by the kernel that have no symbolic name in your loadkeys(1) program. To see this, you compare the range list with the action symbol list, see option

--long-info below.

Number of function keys supported by kernel

This tells the number of action codes that can be used to output strings of characters. These action codes are tra? ditionally bound to the various function and editing keys of the keyboard and are defined to send standard escape sequences. However, you can redefine these to send common command lines, email addresses or whatever you like. Es? pecially if the number of this item is greater than the number of function and editing keys in your keyboard, you may have some "spare" action codes that you can bind to AltGr-letter combinations, for example, to send some use? ful strings. See loadkeys(1) for more details.

Function strings

You can see you current function key definitions with the command

dumpkeys --funcs-only

-l -s --long-info

This option instructs dumpkeys to print a long information list? ing. The output is the same as with the --short-info appended with the list of action symbols supported by loadkeys(1) and dumpkeys(1), along with the symbols' numeric values.

-n --numeric

This option causes dumpkeys to by-pass the conversion of action code values to symbolic notation and to print the in hexadecimal format instead.

-f --full-table

This makes dumpkeys skip all the short-hand heuristics (see keymaps(5)) and output the key bindings in the canonical form. First a keymaps line describing the currently defined modifier combinations is printed. Then for each key a row with a column for each modifier combination is printed. For example, if the current keymap in use uses seven modifiers, every row will have

seven action code columns. This format can be useful for example to programs that post-process the output of dumpkeys.

-Sshape --shape=shape

-1 --separate-lines

This forces dumpkeys to write one line per (modifier,keycode) pair. It prefixes the word plain for plain keycodes.

-t --funcs-only

When this option is given, dumpkeys prints only the function key string definitions. Normally dumpkeys prints both the key bind? ings and the string definitions.

-k --keys-only

When this option is given, dumpkeys prints only the key bind? ings. Normally dumpkeys prints both the key bindings and the string definitions.

-d --compose-only

When this option is given, dumpkeys prints only the compose key combinations. This option is available only if your kernel has compose key support.

-ccharset --charset=charset

This instructs dumpkeys to interpret character code values ac? cording to the specified character set. This affects only the translation of character code values to symbolic names. Valid values for charset currently are iso-8859-X, Where X is a digit in 1-9. If no charset is specified, iso-8859-1 is used as a de? fault. This option produces an output line `charset "iso-8859-X"', telling loadkeys how to interpret the keymap. (For example, "division" is 0xf7 in iso-8859-1 but 0xba in iso-8859-8.)

-v --verbose

-V --version

Prints version number and exits.

FILES

/usr/lib/kbd/keymaps

The recommended directory for keytable files.

SEE ALSO

loadkeys(1), keymaps(5)