NAME ranger - visual file manager

SYNOPSIS ranger [–version] [–help] [–debug] [–clean] [–cachedir=directory] [–confdir=directory] [–datadir=directory] [–copy-config=which] [–choosefile=target] [–choosefiles=target] [–choosedir=target][–selectfile=filepath] [–show-only-dirs] [–list-unused-keys] [–list-tagged-files=tag] [–profile] [–cmd=command] [path]

DESCRIPTION ranger is a console file manager with VI key bindings.

RESOURCES This manual contains instructions on how to use and configure ranger.

    Inside ranger, you can press ? for a list of key bindings, commands or settings.

    The README contains install instructions.

    The file HACKING.md contains guidelines for code modification.

    The directory doc/configs contains configuration files. They are usually installed to
    /usr/share/doc/ranger/config and can be obtained with ranger's --copy-config option.

    The directory examples contains reference implementations for ranger plugins, sample conf
    some programs for integrating ranger with other software. They are usually installed to
    /usr/share/doc/ranger/examples.

    The man page of rifle(1) describes the functions of the file opener

    The section LINKS of this man page contains further resources.

OPTIONS -d, –debug Activate the debug mode: Whenever an error occurs, ranger will exit and print a full traceback. The default behavior is to merely print the name of the exception in the statusbar/log and try to keep running.

    -c, --clean   Activate the clean mode: ranger will not access or create any configuratio
                  leave any traces on your system. This is useful when your configuration is
                  to avoid clutter, etc.

    --cachedir=dir
                  Change the cache directory of ranger from $XDG_CACHE_HOME or ~/.cache/range

    -r dir, --confdir=dir
                  Change the configuration directory of ranger from $XDG_CONFIG_HOME or ~/.co

    --datadir=dir Change the data directory of ranger from $XDG_DATA_HOME or ~/.local/share/r

    --copy-config=file
                  Create copies of the default configuration files in your local configuratio

ones will not be overwritten. Possible values: all, commands, commands_ful

Note: You may want to disable loading of the global configuration files by
RANGER_LOAD_DEFAULT_RC=FALSE in your environment. See also: FILES, ENVIRON

--copy-config=commands will copy only a small sample configuration file wit
commented example. It is recommended to keep this file tidy to avoid getti
ranger upgrades. The full default commands.py can be copied with --copy-co
but that file will be ignored by ranger and serves only as a reference for
commands.

--choosefile=targetfile
            Allows you to pick a file with ranger. This changes the behavior so that w
            ranger will exit and write the absolute path of that file into targetfile.

--choosefiles=targetfile
            Allows you to pick multiple files with ranger. This changes the behavior s
            file, ranger will exit and write the absolute paths of all selected files i
            one newline after each filename.

--choosedir=targetfile
            Allows you to pick a directory with ranger. When you exit ranger, it will
            directory into targetfile.

--selectfile=targetfile
            Open ranger with targetfile selected.

--show-only-dirs
            Display only the directories. May be used in conjunction with --choosedir=t

--list-unused-keys
            List common keys which are not bound to any action in the "browser" context
            complete, you can bind any key that is supported by curses: use the key coc
            "getch()".

--list-tagged-files=tag
            List all files which are tagged with the given tag. Note: Tags are single
            default tag is "*"

--profile   Print statistics of CPU usage on exit.

--cmd=command Execute the command after the configuration has been read. Use this optior
            multiple commands.

--version   Print the version and exit.

```
     -h, --help    Print a list of options and exit.
```

CONCEPTS This part explains how certain parts of ranger work and how they can be used efficiently.

TAGS Tags are single characters which are displayed left of a filename. You can use tags however you want. Press "t" to toggle tags and "ut" to remove any tags of the selection. The default tag is an Asterisk ("*"), but you can use any tag by typing".

PREVIEWS By default, only text files are previewed, but you can enable external preview scripts by setting the option "use_preview_script" and "preview_files" to true.

```
    This default script is ~/.config/ranger/scope.sh. It contains more documentation and call
    and elinks for html, highlight for text/code, img2txt for images, atool for archives, pdf
    mediainfo for video and audio files.

    Install these programs (just the ones you need) and scope.sh will automatically use them.

    Independently of the preview script, there is a feature to preview images by drawing them
    terminal. To enable this feature, set the option "preview_images" to true and enable one
    modes:

    w3m

    This does not work over ssh, requires certain terminals (tested on "xterm" and "urxvt") a
    with tmux, although it works with screen.

    To enable this feature, install the program "w3m" and set the option "preview_images_meth

    iTerm2

    This only works in iTerm2 compiled with image preview support, but works over ssh.

    To enable this feature, set the option "preview_images_method" to iterm2.

    This feature relies on the dimensions of the terminal's font.  By default, a width of 8 a
    used.  To use other values, set the options "iterm2_font_width" and "iterm2_font_height"

    urxvt

    This only works in urxvt compiled with pixbuf support. Does not work over ssh.

    Essentially this mode sets an image as a terminal background temporarily, so it will brea
    image background.

    To enable this feature, set the option "preview_images_method" to urxvt.
```

urxvt-full

The same as urxvt but utilizing not only the preview pane but the whole terminal window.

To enable this feature, set the option "preview_images_method" to urxvt-full.

SELECTION The selection is defined as "All marked files IF THERE ARE ANY, otherwise the current file." Be aware of this when using the :delete command, which deletes all files in the selection.

You can mark files by pressing <Space>, v, etc. A yellow Mrk symbol at the bottom right are marked files in this directory.

MACROS Macros can be used in commands to abbreviate things.

```
%f    the highlighted file
%d    the path of the current directory
%s    the selected files in the current directory
%t    all tagged files in the current directory
%c    the full paths of the currently copied/cut files
%p    the full paths of selected files
```

The macros %f, %d, %p, and %s also have upper case variants, %F, %D, %P, and %S, which re To refer to specific tabs, add a number in between. (%7s = selection of the seventh tab.

%c is the only macro which ranges out of the current directory. So you may "abuse" the c other purposes, like diffing two files which are in different directories:

 Yank the file A (type yy), move to the file B, then type
 @diff %c %f

Macros for file paths are generally shell-escaped so they can be used in the "shell" comm

Additionally, if you create a key binding that uses <any>, a special statement which acc macro %any (or %any0, %any1, %any2, ...) can be used in the command to get the key that

The macro %rangerdir expands to the directory of ranger's python library, you can use it this command:
   alias show_commands shell less %rangerdir/config/commands.py

%confdir expands to the directory given by --confdir.

%datadir expands to the directory given by --datadir.

The macro %space expands to a space character. You can use it to add spaces to the end of needed, while preventing editors to strip spaces off the end of the line automatically.

To write a literal %, you need to escape it by writing %%.

BOOKMARKS Type m to bookmark the current directory. You can re-enter this directory by typing '. can be any letter or digit. Unlike vim, both lowercase and uppercase bookmarks are persistent.

Each time you jump to a bookmark, the special bookmark at key ` will be set to the last c "``" gets you back to where you were before.

Bookmarks are selectable when tabbing in the :cd command.

Note: The bookmarks ' (Apostrophe) and ` (Backtick) are the same.

RIFLE Rifle is the file opener of ranger. It can be used as a standalone program or a python module. It is located at ranger/ext/rifle.py. In contrast to other, more simple file openers, rifle can automatically find installed programs so it can be used effectively out of the box on a variety of systems.

It's configured in rifle.conf through a list of conditions and commands. For each line t checked and if they are met, the respective command is taken into consideration. By defa matching rule is used. In ranger, you can list and choose rules by typing "r" or simply "<rulenumber><enter>". If you use rifle standalone, you can list all rules with the "-l' rule with "-p <number>".

The rules, along with further documentation, are contained in ranger/config/rifle.conf.

FLAGS Flags give you a way to modify the behavior of the spawned process. They are used in the commands ":open_with" (key "r") and ":shell" (key "!").

    f   Fork the process. (Run in background)
    c   Run the current file only, instead of the selection
    r   Run application with root privilege (requires sudo)
    t   Run application in a new terminal window

There are some additional flags that can currently be used only in the "shell" command: (df")

    p   Redirect output to the pager
    s   Silent mode. Output will be discarded.
    w   Wait for an Enter-press when the process is done

By default, all the flags are off unless specified otherwise in the rifle.conf configurat specify as many flags as you want. An uppercase flag negates the effect: "ffcccFsf" is e

The terminal program name for the "t" flag is taken from the environment variable $TERMCM exist, it tries to extract it from $TERM, uses "x-terminal-emulator" as a fallback, and t fails.

Examples: ":open_with c" will open the file that you currently point at, even if you have

```
     ":shell -w df" will run "df" and wait for you to press Enter before switching back to ran
```

PLUGINS ranger's plugin system consists of python files which are located in
~/.config/ranger/plugins/ and are imported in alphabetical order when starting
ranger. A plugin changes rangers behavior by overwriting or extending a function
that ranger uses. This allows you to change pretty much every part of ranger,
but there is no guarantee that things will continue to work in future versions as
the source code evolves.

```
    Adding new commands via a plugin as simple as specifying them like you would do in the co

    There are some hooks that are specifically made for the use in plugins.  They are functio
    hook_ and can be found throughout the code.

     grep 'def hook_' -r /path/to/rangers/source

    Also try:

     pydoc ranger.api

    Note that you should NOT simply overwrite a function unless you know what you're doing.
    existing function and call it from your new one.  This way, multiple plugins can use the
    several sample plugins in the /usr/share/doc/ranger/examples/ directory, including a hel
    describes this procedure.
```

KEY BINDINGS Key bindings are defined in the file ranger/config/rc.conf.
Check this file for a list of all key bindings. You can copy it to your local
configuration directory with the –copy-config=rc option.

```
    Many key bindings take an additional numeric argument.  Type 5j to move down 5 lines, 2l
    2, 10<Space> to mark 10 files.

    This list contains the most useful bindings:
```

MAIN BINDINGS h, j, k, l Move left, down, up or right

```
    ^D or J, ^U or K
                   Move a half page down, up

    H, L         Move back and forward in the history

    gg           Move to the top

    G            Move to the bottom

    [, ]         Move up and down in the parent directory.

    ^R           Reload everything
```

F   Toggle freeze_files setting. When active (indicated by a cyan FROZEN messa
             directories and files will not be loaded, improving performance when all th
             already loaded. This does not affect file previews, which can be toggled w
             disabling the preview of directories with zP.

^L   Redraw the screen

i   Inspect the current file in a bigger window.

E   Edit the current file in $VISUAL otherwise $EDITOR otherwise "vim"

S   Open a shell in the current directory

?   Opens this man page

W   Opens the log window where you can review messages that pop up at the botto

w   Opens the task window where you can view and modify background processes th
             ranger. In there, you can type "dd" to abort a process and "J" or "K" to
             a process. Only one process is run at a time.

^C   Stop the currently running background process that ranger has started, like
             loading directories or file previews.

<octal>=, +<who><what>, -<who><what>
             Change the permissions of the selection. For example, "777=" is equivalen
             "+ar" does "chmod a+r %s", "-ow" does "chmod o-w %s" etc.

yy   Copy (yank) the selection, like pressing Ctrl+C in modern GUI programs. (Y
             to add files to the copy buffer, "yr" to remove files again, or "yt" for to

dd   Cut the selection, like pressing Ctrl+X in modern GUI programs. (There are
             "dt" shortcuts equivalent to "ya", "yr" and "yt".)

pp   Paste the files which were previously copied or cut, like pressing Ctrl+V i

po   Paste the copied/cut files, overwriting existing files.

pP, pO   Like pp and po, but queues the operation so that it will be executed after
             Reminder: type "w" to open the task window.

pl, pL   Create symlinks (absolute or relative) to the copied files

phl   Create hardlinks to the copied files

pht   Duplicate the subdirectory tree of the copied directory, then create hardli

                                    7

```
                   file into the new directory tree.

mX                 Create a bookmark with the name X

`X                 Move to the bookmark with the name X

n                  Find the next file.  By default, this gets you to the newest file in the di
                   search something using the keys /, cm, ct, ..., it will get you to the next

N                  Find the previous file.

oX                 Change the sort method (like in mutt)

zX                 Change settings.  See the settings section for a list of settings and thei

u?                 Universal undo-key.  Depending on the key that you press after "u", it eith
                   tabs (uq), removes tags (ut), clears the copy/cut buffer (ud), starts the 
                   (uV) or clears the selection (uv).

f                  Quickly navigate by entering a part of the filename.

Space              Mark a file.

v                  Toggle the mark-status of all files

V                  Starts the visual mode, which selects all files between the starting point
                   you press ESC.  To unselect files in the same way, use "uV".

/                  Search for files in the current directory.

:                  Open the console.

!                  Open the console with the content "shell " so you can quickly run commands

@                  Open the console with the content "shell  %s", placing the cursor before th
                   quickly run commands with the current selection as the argument.

r                  Open the console with the content "open with " so you can decide which pro
                   current file selection.

cd                 Open the console with the content "cd "

Alt-N              Open a tab. N has to be a number from 0 to 9. If the tab doesn't exist yet,

gn, ^N             Create a new tab.
```

```
    gt, gT        Go to the next or previous tab. You can also use TAB and SHIFT+TAB instead.

    gc, ^W        Close the current tab.  The last tab cannot be closed this way.

    M             A key chain that allows you to quickly change the line mode of all the file
                  directory.  For a more permanent solution, use the command "default_linemod
```

READLINE-LIKE BINDINGS IN THE CONSOLE ^B, ^F Move left and right
(B for back, F for forward)

```
    ^P, ^N        Move up and down (P for previous, N for Next)

    ^A, ^E        Move to the start or to the end

    ^D            Delete the current character.

    ^H            Backspace.
```

MOUSE BUTTONS Left Mouse Button Click on something and you'll move
there. To run a file, "enter" it, like a directory, by clicking on the preview.

```
    Right Mouse Button
        Enter a directory or run a file.

    Scroll Wheel
        Scrolls up or down.  You can point at the column of the parent directory while scroll
        directories.
```

SETTINGS This section lists all built-in settings of ranger. The valid types for
the value are in [brackets]. The hotkey to toggle the setting is in , if a hotkey
exists.

```
    Settings can be changed in the file ~/.config/ranger/rc.conf or on the fly with the comma
    Examples:

     set column_ratios 1,2,3
     set show_hidden true

    Toggling options can be done with:

     set show_hidden!

    The different types of settings and an example for each type:

     setting type   | example values
     ---------------+--------------------------
     bool           | true, false
     integer        | 1, 23, 1337
     string         | foo, hello world
```

```
 list          | 1,2,3,4
 none          | none
```

You can view a list of all settings and their current values by pressing "3?" in ranger.

automatically_count_files [bool]
    Should ranger count and display the number of files in each directory as soon as it's
    slow with remote file sytems. Turning it off will still allow you to see the number
    entering the directory.

autosave_bookmarks [bool]
    Save bookmarks (used with mX and `X) instantly? This helps to synchronize bookmarks
    ranger instances but leads to *slight* performance loss. When false, bookmarks are s
    exited.

autoupdate_cumulative_size [bool]
    You can display the "real" cumulative size of directories by using the command :get_c
    typing "dc". The size is expensive to calculate and will not be updated automaticall
    update it automatically though by turning on this option.

cd_bookmarks [bool]
    Specify whether bookmarks should be included in the tab completion of the "cd" comman

cd_tab_case [string]
    Changes case sensitivity for the "cd" command tab completion. Possible values are:

     sensitive
     insensitive
     smart

cd_tab_fuzzy [bool]
    Use fuzzy tab completion with the "cd" command. For example, :cd /u/lo/b<TAB> expands
    /usr/local/bin.

clear_filters_on_dir_change [bool]
    If set to 'true', persistent filters would be cleared upon leaving the directory

collapse_preview [bool] <zc>
    When no preview is visible, should the last column be squeezed to make use of the whi

colorscheme [string]
    Which colorscheme to use? These colorschemes are available by default: default, jung
    monochrome scheme, jungle replaces blue directories with green ones for better visibi
    terminals.

column_ratios [list]

How many columns are there, and what are their relative widths? For example, a value
evenly sized columns. 1,1,1,1,4 means 5 columns with the preview column being as larg
combined.

confirm_on_delete [string]
    Ask for a confirmation when running the "delete" command? Valid values are "always"
    "multiple". With "multiple", ranger will ask only if you delete multiple files at on

dirname_in_tabs [bool]
    Display the directory name in tabs?

display_size_in_main_column [bool]
    Display the file size in the main column?

display_size_in_status_bar [bool]
    Display the file size in the status bar?

display_tags_in_all_columns [bool]
    Display tags in all columns?

draw_borders [bool]
    Draw borders around columns?

draw_progress_bar_in_status_bar [bool]
    Draw a progress bar in the status bar which displays the average state of all current
    support progress bars?

flushinput [bool] <zi>
    Flush the input after each key hit? One advantage is that when scrolling down with '
    scrolling instantly when you release the key. One disadvantage is that when you type
    some keys might get lost.

freeze_files [bool] <F>
    When active, directories and files will not be loaded, improving performance when al
    are already loaded. This does not affect file previews.

global_inode_type_filter [string]
    Like filter_inode_type, but globally for all directories. Useful in combination with

      ranger --choosedir=/tmp/x --cmd='set global_inode_type_filter d'

hidden_filter [string]
    A regular expression pattern for files which should be hidden. For example, this pat
    files that start with a dot or end with a tilde.

      set hidden_filter ^\.|~$

11

idle_delay [integer]
    The delay that ranger idly waits for user input, in milliseconds, with a resolution o
    reduces lag between directory updates but increases CPU load.

iterm2_font_height [integer]
    Change the assumed font height in iTerm2, which may help with iTerm image previews

iterm2_font_width [integer]
    Change the assumed font width in iTerm2, which may help with iTerm image previews

line_numbers [string]
    Show line numbers in main column. Possible values are:

     false   turn the feature off
     absolute   absolute line numbers for use with "<N>gg"
     relative   relative line numbers for "<N>k" or "<N>j"

max_console_history_size [integer, none]
    How many console commands should be kept in history? "none" will disable the limit.

max_history_size [integer, none]
    How many directory changes should be kept in history?

metadata_deep_search [bool]
    When the metadata manager module looks for metadata, should it only look for a ".meta
    current directory, or do a deep search and check all directories above the current on

mouse_enabled [bool] <zm>
    Enable mouse input?

one_indexed [bool]
    Start line numbers from 1. Possible values are:

     false   start line numbers from 0
     true   start line numbers from 1

open_all_images [bool]
    Open all images in this directory when running certain image viewers like feh or sxi
    selected files by marking them.

padding_right [bool]
    When collapse_preview is on and there is no preview, should there remain a little pad
    This allows you to click into that space to run the file.

preview_directories [bool] <zP>

Preview directories in the preview column?

preview_files [bool] <zp>
    Preview files in the preview column?

preview_images [bool]
    Draw images inside the console with the external program w3mimgpreview?

preview_images_method [string]
    Set the preview image method. Supported methods: w3m, iterm2, urxvt, urxvt-full, term
    section.

preview_max_size [int]
    Avoid previewing files that exceed a certain size, in bytes. Use a value of 0 to dis

preview_script [string, none]
    Which script should handle generating previews? If the file doesn't exist, or use_pr
    ranger will handle previews itself by just printing the content.

save_backtick_bookmark [bool]
    Save the "`" bookmark to disk. This bookmark is used to switch to the last directory

save_console_history [bool]
    Should the console history be saved on exit? If disabled, the console history is res
    ranger.

save_tabs_on_exit [bool]
    Save all tabs, except the active, on exit? The last saved tabs are restored once when
    session. Multiple sessions are stored in a stack and the oldest saved tabs are restor

scroll_offset [integer]
    Try to keep this much space between the top/bottom border when scrolling.

shorten_title [integer]
    Trim the title of the window if it gets long? The number defines how many directorie
    once. A value of 0 turns off this feature.

show_cursor [bool]
    Always show the terminal cursor?

show_hidden_bookmarks [bool]
    Show dotfiles in the bookmark preview window? (Type ')

show_hidden [bool] <zh>, <^H>
    Show hidden files?

show_selection_in_titlebar [bool]
    Add the highlighted file to the path in the titlebar

sort_case_insensitive [bool] <zc>
    Sort case-insensitively? If true, "a" will be listed before "B" even though its ASC

sort_directories_first [bool] <zd>
    Sort directories first?

sort_reverse [bool] <or>
    Reverse the order of files?

sort_unicode [bool]
    When sorting according to some string, should the unicode characters be compared, ins
    raw character values to save time?

sort [string] <oa>, <ob>, <oc>, <oe>, <om>, <on>, <ot>, <os>, <oz>
    Which sorting mechanism should be used? Choose one of atime, basename, ctime, extens
    type, size, random

    Note: You can reverse the order by typing an uppercase second letter in the key combi
    sort from Z to A.

status_bar_on_top [bool]
    Put the status bar at the top of the window?

hostname_in_titlebar [bool]
    Show hostname in titlebar?

tilde_in_titlebar [bool]
    Abbreviate $HOME with ~ in the titlebar (first line) of ranger?

unicode_ellipsis [bool]
    Use a unicode "..." character instead of "~" to mark cut-off filenames?

update_title [bool]
    Set a window title?

update_tmux_title [bool]
    Set the title to "ranger" in the tmux program?

use_preview_script [bool] <zv>
    Use the preview script defined in the setting preview_script?

vcs_aware [bool]
    Gather and display data about version control systems. Supported vcs: git, hg.

```
vcs_backend_git, vcs_backend_hg, vcs_backend_bzr, vcs_backend_svn [string]
    Sets the state for the version control backend. The possible values are:

       disabled   don't display any information.
       local      display only local state.
       enabled    display both, local and remote state. May be slow for hg and bzr.

viewmode [string]
    Sets the view mode, which can be miller to display the files in the traditional mille
    shows multiple levels of the hierarchy, or multipane to use multiple panes (one per t
    midnight-commander.

wrap_scroll [bool]
    Enable scroll wrapping - moving down while on the last item will wrap around to the t

xterm_alt_key [bool]
    Enable this if key combinations with the Alt Key don't work for you.  (Especially on
```

COMMANDS You can enter the commands in the console which is opened by
pressing ":".

```
You can always get a list of the currently existing commands by typing "2?" in ranger.  F
this is a list of the "public" commands including their parameters, excluding description

 alias [newcommand] [oldcommand]
 bulkrename
 cd [directory]
 chain command1[; command2[; command3...]]
 chmod octal_number
 cmap key command
 console [-pSTARTPOSITION] command
 copycmap key newkey [newkey2...]
 copymap key newkey [newkey2...]
 copypmap key newkey [newkey2...]
 copytmap key newkey [newkey2...]
 cunmap keys...
 default_linemode [path=regexp | tag=tags] linemodename
 delete
 echo [text]
 edit [filename]
 eval [-q] python_code
 filter [string]
 filter_inode_type [dfl]
 find pattern
 flat level
 grep pattern
```

```
help
jump_non [-FLAGS...]
linemode linemodename
load_copy_buffer
map key command
mark pattern
mark_tag [tags]
meta key value
mkdir dirname
open_with [application] [flags] [mode]
pmap key command
prompt_metadata [key1 [key2 [...]]]
punmap keys...
quit
quit!
quitall
quitall!
relink newpath
rename_append [-FLAGS...]
rename newname
save_copy_buffer
scout [-FLAGS...] pattern
search pattern
search_inc pattern
set option value
setintag tags option value
setlocal [path=<path>] option value
shell [-FLAGS...] command
source filename
terminal
tmap key command
touch filename
travel pattern
tunmap keys...
unmap keys...
unmark pattern
unmark_tag [tags]
```

There are additional commands which are directly translated to python functions, one for
ranger.core.actions.Actions class. They are not documented here, since they are mostly 1
to be typed in by a user. Read the source if you are interested in them.

These are the public commands including their descriptions:

alias [newcommand] [oldcommand]
  Copies the oldcommand as newcommand.

bulkrename
  This command opens a list of selected files in an external editor. After you edit and
  generate a shell script which does bulk renaming according to the changes you did in th

  This shell script is opened in an editor for you to review. After you close it, it wil

cd [directory]
  The cd command changes the directory. The command ":cd -" is equivalent to typing ``.

chain command1[; command2[; command3...]]
  Combines multiple commands into one, separated by semicolons.

chmod octal_number
  Sets the permissions of the selection to the octal number.

  The octal number is between 000 and 777. The digits specify the permissions for the use
  others. A 1 permits execution, a 2 permits writing, a 4 permits reading. Add those nu
  So a 7 permits everything.

  Key bindings in the form of [-+]<who><what> and <octal>= also exist. For example, +ar
  everyone, -ow forbids others to write and 777= allows everything.

  See also: man 1 chmod

cmap key command
  Binds keys for the console. Works like the "map" command.

console [-pN] command
  Opens the console with the command already typed in. The cursor is placed at N.

copycmap key newkey [newkey2 ...]
  See "copymap"

copymap key newkey [newkey2 ...]
  Copies the keybinding key to newkey in the "browser" context. This is a deep copy, so
  binding (or parts of it) later, the old one is not modified.

  To copy key bindings of the console, taskview, or pager use "copycmap", "copytmap" or "

copypmap key newkey [newkey2 ...]
  See "copymap"

copytmap key newkey [newkey2 ...]
  See "copymap"

cunmap [keys...]
  Removes key mappings of the console. Works like the "unmap" command.

default_linemode [path=regexp | tag=tags] linemodename
  Sets the default linemode. See linemode command.

  Examples:

  Set the global default linemode to "permissions":
   :default_linemode permissions

  Set the default linemode to "permissions" for all files tagged with "p" or "P":
   :default_linemode tag=pP permissions

  Set the default linemode for all files in ~/books/ to "metatitle":
   :default_linemode path=/home/.*?/books/.* metatitle

delete
  Destroy all files in the selection with a roundhouse kick. ranger will ask for a confi
  to delete multiple (marked) files or non-empty directories. This can be changed by mod
  "confirm_on_delete".

echo text
  Display the text in the statusbar.

edit [filename]
  Edit the current file or the file in the argument.

eval [-q] python_code
  Evaluates the python code. `fm' is a reference to the FM instance. To display text, u
  The result is displayed on the screen unless you use the "-q" option.

  Examples:
   :eval fm
   :eval len(fm.tabs)
   :eval p("Hello World!")

filter [string]
  Displays only the files which contain the string in their basename. Running this comma
  parameter will reset the filter.

  This command is based on the scout command and supports all of its options.

filter_inode_type [dfl]
  Displays only the files of specified inode type. To display only directories, use the '
  display only files, use the 'f' parameter. To display only links, use the 'l' paramete

combined. To remove this filter, use no parameter.

find pattern
  Search files in the current directory that contain the given (case-insensitive) string
  type. Once there is an unambiguous result, it will be run immediately. (Or entered, i

  This command is based on the scout command and supports all of its options.

flat level
  Flattens the directory view up to the specified level. Level -1 means infinite level. I
  view without flattened directory view. Level values -2 and less are invalid.

grep pattern
  Looks for a string in all marked files or directories.

help
  Provides a quick way to view ranger documentations.

jump_non [-flags...]
  Jumps to first non-directory if highlighted file is a directory and vice versa.

  Flags:
   -r   Jump in reverse order
   -w   Wrap around if reaching end of filelist

linemode linemodename
  Sets the linemode of all files in the current directory. The linemode may be:

    "filename": display each line as "<basename>...<size>"
    "fileinfo": display each line as "<basename>...<file(1) output>"
    "permissions": display each line as "<permissions> <owner> <group> <basename>"
    "metatitle": display metadata from .metadata.json files if
        available, fall back to the "filename" linemode if no
        metadata was found. See :meta command.

  The custom linemodes may be added by subclassing the LinemodeBase class. See the range
  for some examples.

load_copy_buffer
  Load the copy buffer from ~/.config/ranger/copy_buffer. This can be used to pass the l
  another ranger instance.

map key command
  Assign the key combination to the given command. Whenever you type the key/keys, the c
  executed. Additionally, if you use a quantifier when typing the key, like 5j, it will
  command as the attribute "self.quantifier".

The keys you bind with this command are accessible in the file browser only, not in the
pager. To bind keys there, use the commands "cmap", "tmap" or "pmap".

mark pattern
  Mark all files matching the regular expression pattern.

  This command is based on the scout command and supports all of its options.

mark_tag [tags]
  Mark all tags that are tagged with either of the given tags. When leaving out the tag
  files are marked.

meta key value
  Set the metadata of the currently highlighted file. Example:

    :meta title The Hitchhiker's Guide to the Galaxy
    :meta year 1979

  This metadata can be displayed by, for example, using the "metatitle" line mode by typi

mkdir dirname
  Creates a directory with the name dirname.

open_with [application] [flags] [mode]
  Open the selected files with the given application, unless it is omitted, in which case
  application is used. flags change the way the application is executed and are describe
  in this man page. The mode is a number that specifies which application to use. The l
  generated by the external file opener "rifle" and can be displayed when pressing "r" in

  Note that if you specify an application, the mode is ignored.

pmap key command
  Binds keys for the pager. Works like the "map" command.

prompt_metadata [keys ...]
  Prompt the user to input metadata with the "meta" command for multiple keys in a row.

punmap [keys ...]
  Removes key mappings of the pager. Works like the "unmap" command.

quit
  Closes the current tab, if there's only one tab. Otherwise quits if there are no tasks
  current directory will be bookmarked as ' so you can re-enter it by typing `` or '' the
  ranger.

quit!
  Like "quit", except will force quit even if tasks are in progress.

quitall
  Like "quit", except will quit even if multiple tabs are open.

quitall!
  Like "quitall", except will force quit even if tasks are in progress.

relink newpath
  Change the link destination of the current symlink file to <newpath>. First <tab> will
  link.

rename newname
  Rename the current file. If a file with that name already exists, the renaming will fa
  binding A for appending something to a file name.

rename_append [-flags...]
  Opens the console with ":rename <current file>" with the cursor positioned before the f

  Flags:
   -a   Position before all extensions
   -r   Remove everything before extensions

save_copy_buffer
  Save the copy buffer to ~/.config/ranger/copy_buffer. This can be used to pass the lis
  another ranger instance.

scout [-flags...] [--] pattern
  Swiss army knife command for searching, traveling and filtering files.

  Flags:
   -a   Automatically open a file on unambiguous match
   -e   Open the selected file when pressing enter
   -f   Filter files that match the current search pattern
   -g   Interpret pattern as a glob pattern
   -i   Ignore the letter case of the files
   -k   Keep the console open when changing a directory with the command
   -l   Letter skipping; e.g. allow "rdme" to match the file "readme"
   -m   Mark the matching files after pressing enter
   -M   Unmark the matching files after pressing enter
   -p   Permanent filter: hide non-matching files after pressing enter
   -r   Interpret pattern as a regular expression pattern
   -s   Smart case; like -i unless pattern contains upper case letters
   -t   Apply filter and search pattern as you type
   -v   Inverts the match

Multiple flags can be combined. For example, ":scout -gpt" would create a :filter-like
globbing.

search pattern
  Search files in the current directory that match the given (case insensitive) regular e

  This command is based on the scout command and supports all of its options.

search_inc pattern
  Search files in the current directory that match the given (case insensitive) regular e
  This command gets you to matching files as you type.

  This command is based on the scout command and supports all of its options.

set option value
  Assigns a new value to an option. Valid options are listed in the settings section. U
  get the current value of an option, though this doesn't work for functions and regular
  values are:

    setting type | example values
    ---------------+---------------------------
    bool | true, false
    integer | 1, 23, 1337
    string | foo, hello world
    list | 1,2,3,4
    none | none

setintag tags option value
  Assigns a new value to an option, but locally for the directories that are marked with
  this option only takes effect when visiting that directory.

  For example, to change the sorting order in your downloads directory, tag it with the v
  then use this command:

    setintag v sort ctime

setlocal [path=path] option value
  Assigns a new value to an option, but locally for the directory given by path. This mea
  only takes effect when visiting that directory. If no path is given, uses the current c

  path is a regular expression. This means that "path=~/dl" applies to all paths that st
  ~/dl2 and ~/dl/foo. To avoid this, use "path=~/dl$".

  path can be quoted with either single or double quotes to prevent unwanted splitting. p
  path="~/dl dl$"

```
shell [-flags] command
  Run a shell command.  flags are discussed in their own section.

source filename
  Reads commands from a file and executes them in the ranger console.

  This can be used to re-evaluate the rc.conf file after changing it:

   map X chain shell vim -p %confdir/rc.conf %rangerdir/config/rc.conf; source %confdir/r

terminal
  Spawns the x-terminal-emulator starting in the current directory.

tmap key command
  Binds keys for the taskview. Works like the "map" command.

touch filename
  Creates an empty file with the name filename, unless it already exists.

travel pattern
  Filters the current directory for files containing the letters in the string, possibly
  between.  The filter is applied as you type.  When only one directory is left, it is en
  is automatically reopened, allowing for fast travel.  To close the console, press ESC o

  This command is based on the scout command and supports all of its options.

tunmap [keys ...]
  Removes key mappings of the taskview. Works like the "unmap" command.

unmap [keys ...]
  Removes the given key mappings in the "browser" context.  To unmap key bindings in the
  pager use "cunmap", "tunmap" or "punmap".

unmark pattern
  Unmark all files matching a regular expression pattern.

  This command is based on the scout command and supports all of its options.

unmark_tag [tags]
  Unmark all tags that are tagged with either of the given tags.  When leaving out the ta
  files are unmarked.
```

FILES ranger reads several configuration files which are located in $HOME/.config/ranger or $XDG_CONFIG_HOME/ranger if $XDG_CONFIG_HOME is defined. You can use the –copy-config option to obtain the default configuration

files. The files contain further documentation.

    rc.conf, commands.py and colorschemes do not need to be copied fully as they will only be
    configuration files except if explicitly overridden. This may lead to some confusing situ
    when a key is being bound despite the corresponding line being removed from the user's co
    configuration file. This behavior may be disabled with an environment variable (see also:
    All other configuration files only read from one source; i.e. default OR user, not both.

    When starting ranger with the --clean option, it will not access or create any of these f

CONFIGURATION rc.conf Contains a list of commands which are executed on
startup. Mostly key bindings and settings are defined here.

    commands.py
              A python module that defines commands which can be used in ranger's console by
              rc.conf file. Note that you can define commands in the same manner within plug

    commands_full.py
              This file is copied by --copy-config=commands_full and serves as a reference fo
              is entirely ignored by ranger.

    rifle.conf
              This is the configuration file for the built-in file launcher called "rifle".

    scope.sh This is a script that handles file previews. When the options use_preview_scri
              are set, the program specified in the option preview_script is run and its outp
              determines rangers reaction.

    colorschemes/
              Colorschemes can be placed here.

    plugins/ Plugins can be placed here.

STORAGE bookmarks This file contains a list of bookmarks. The syntax is
/^(.):(.*)$/. The first character is the bookmark key and the rest after the colon
is the path to the file. In ranger, bookmarks can be set by typing m, accessed
by typing ' and deleted by typing um.

    copy_buffer
              When running the command :save_copy_buffer, the paths of all currently copied f
              file. You can later run :load_copy_buffer to copy the same files again, pass t
              instance or process them in a script.

    history   Contains a list of commands that have been previously typed in.

    tagged   Contains a list of tagged files. The syntax is /^(.:)?(.*)$/ where the first le
              name of the tag and the rest after the optional colon is the path to the file.
              be set by pressing t and removed with T. To assign a named tag, type "<tagname

ENVIRONMENT These environment variables have an effect on ranger:

RANGER_LEVEL
        ranger sets this environment variable to "1" or increments it if it already exist
        can determine whether they were spawned from ranger by checking for this variable

RANGER_LOAD_DEFAULT_RC
        If this variable is set to FALSE, ranger will not load the default rc.conf. This
        copied the whole rc.conf to ~/.config/ranger/ and don't need the default one at a

VISUAL Defines the editor to be used for the "E" key. Falls back to EDITOR if undefined

EDITOR Defines the editor to be used for the "E" key if VISUAL is undefined or empty. I

SHELL   Defines the shell that ranger is going to use with the :shell command and the "S'
        "/bin/sh".

TERMCMD Defines the terminal emulator command that ranger is going to use with the :term:
        "t" run flag. Defaults to "xterm".

XDG_CONFIG_HOME
        Specifies the directory for configuration files. Defaults to $HOME/.config.

PYTHONOPTIMIZE
        This variable determines the optimize level of python.

        Using PYTHONOPTIMIZE=1 (like python -O) will make python discard assertion statem
        efficiency at the cost of losing some debug info.

        Using PYTHONOPTIMIZE=2 (like python -OO) will additionally discard any docstrings
        disable the <F1> key on commands.

W3MIMGDISPLAY_PATH
        By changing this variable, you can change the path of the executable file for ima
        default, it is set to /usr/lib/w3m/w3mimgdisplay.

EXAMPLES There are various examples on how to extend ranger with plugins or combine ranger with other programs. These can be found in the /usr/share/doc/ranger/examples/ directory, or the doc/ranger/ that is provided along with the source code.

LICENSE GNU General Public License 3 or (at your option) any later version.

LINKS Download: https://ranger.github.io/ranger-stable.tar.gz The project page: https://ranger.github.io/ The mailing list: https://savannah.nongnu.org/mail/?group=ranger IRC channel: #ranger on freenode.net

    ranger is maintained with the git version control system. To fetch a fresh copy, run:

```
git clone git://git.savannah.nongnu.org/ranger.git
```

SEE ALSO rifle(1)

BUGS Report bugs here: https://github.com/ranger/ranger/issues

```
Please include as much relevant information as possible.  For the most diagnostic output,
"PYTHONOPTIMIZE= ranger --debug"
```

2018-02-05 ranger-1.9.0 RANGER(1)