license GNU GPL version 3 license MIT zsh v5.0.0 chat on gitter

# Zsh Navigation Tools

Set of tools like `n-history` – multi-word history searcher, `n-cd` – directory bookmark manager, `n-kill` – `htop` like kill utility, and more. Based on `n-list`, a tool generating selectable curses-based list of elements that has access to current `Zsh` session, i.e. has broad capabilities to work together with it. Feature highlights include incremental multi-word searching, approximate matching, ANSI coloring, themes, unique mode, horizontal scroll, grepping, advanced history management and various integrations with `Zsh`.

To use it, add `zsh-navigation-tools` to the plugins array in your zshrc file:

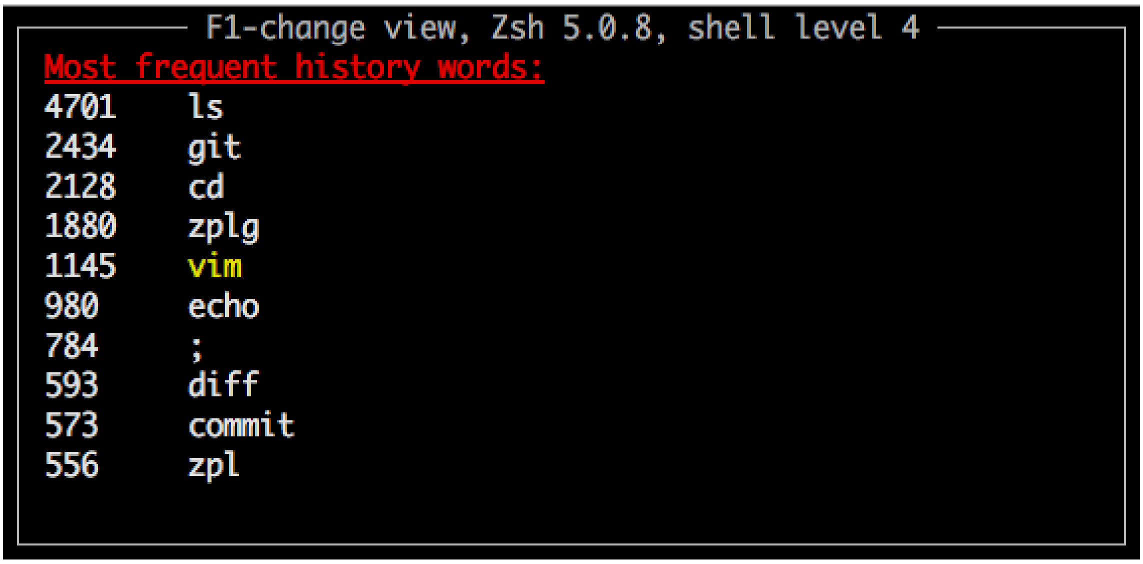
```
plugins=(... zsh-navigation-tools)
```

Also check out Zsh Command Architect and Zconvey

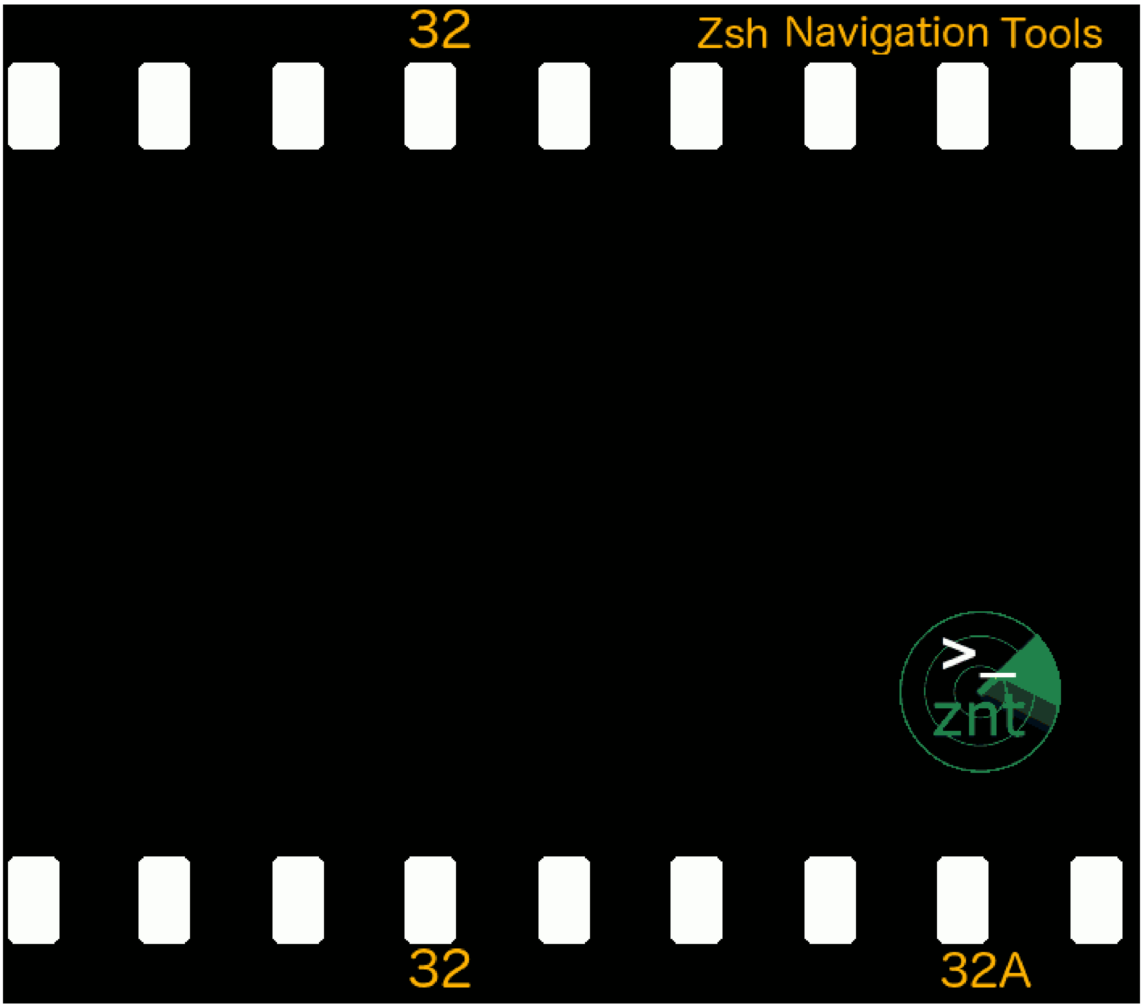
Videos:

- https://youtu.be/QwZ8IJEgXRE
- https://youtu.be/DN9QqssAYB8

Screenshots:



```
F1-change view, Zsh 5.0.8, shell level 4
Most frequent history words:
4701    ls
2434    git
2128    cd
1880    zplg
1145    vim
980     echo
784     ;
593     diff
573     commit
556     zpl
```



**History Widget**

To have `n-history` as the incremental searcher bound to `Ctrl-R` copy `znt-*` files into the `*/site-functions` dir (unless you do single file install) and add:

```
autoload znt-history-widget
zle -N znt-history-widget
bindkey "^R" znt-history-widget
```

to `.zshrc`. This is done automatically when using the installer, zgen, antigen or single file install. Two other widgets exist, `znt-cd-widget` and `znt-kill-widget`, they too can be assigned to key combinations ( `autoload` is done in `.zshrc` so no need of it):

```
zle -N znt-cd-widget
bindkey "^B" znt-cd-widget
zle -N znt-kill-widget
bindkey "^Y" znt-kill-widget
```

## Introduction

The tools are:

- `n-aliases` - browses aliases, relegates editing to `vared`
- `n-cd` - browses dirstack and bookmarked directories, allows to enter selected directory
- `n-functions` - browses functions, relegates editing to `zed` or `vared`
- `n-history` - browses history, allows to edit and run commands from it
- `n-kill` - browses processes list, allows to send signal to selected process
- `n-env` - browses environment, relegates editing to `vared`
- `n-options` - browses options, allows to toggle their state
- `n-panelize` - loads output of given command into the list for browsing

All tools support horizontal scroll with `<`, `>`, `{`, `}`, `h`, `l` or left and right cursors. Other keys are:

- `H`, `?` (from n-history) - run n-help
- `Ctrl-R` - start n-history, the incremental, multi-keyword history searcher (Zsh binding)
- `Ctrl-A` - rotate entered words (1+2+3 -> 3+1+2)
- `Ctrl-F` - fix mode (approximate matching)
- `Ctrl-L` - redraw of whole display
- `Ctrl-T` - browse themes (next theme)
- `Ctrl-G` - browse themes (previous theme)
- `Ctrl-U` - half page up
- `Ctrl-D` - half page down
- `Ctrl-P` - previous element (also done with vim's k)
- `Ctrl-N` - next element (also done with vim's j)
- `[`, `]` - jump directory bookmarks in n-cd and typical signals in n-kill
- `g`, `G` - beginning and end of the list
- `/` - show incremental search
- `F3` - show/hide incremental search
- `Esc` - exit incremental search, clearing filter
- `Ctrl-W` (in incremental search) - delete whole word

- `Ctrl-K` (in incremental search) - delete whole line
- `Ctrl-O` , `o` - enter uniq mode (no duplicate lines)
- `Ctrl-E` , `e` - edit private history (when in private history view)
- `F1` - (in n-history) - switch view
- `F2` , `Ctrl-X` , `Ctrl-/` - search predefined keywords (defined in config files)

## Configuration

`ZNT` has configuration files located in `~/.config/znt` . The files are:

```
n-aliases.conf
n-cd.conf
n-env.conf
n-functions.conf
n-history.conf
n-kill.conf
n-list.conf
n-options.conf
n-panelize.conf
```

`n-list.conf` contains main configuration variables:

```
# Should the list (text, borders) be drawn in bold
local bold=0

# Main color pair (foreground/background)
local colorpair="white/black"

# Should draw the border?
local border=1

# Combinations of colors to try out with Ctrl-T and Ctrl-G
# The last number is the bold option, 0 or 1
local -a themes
themes=( "white/black/1" "green/black/0" "green/black/1" "white/blue/0"
"white/blue/1"
         "magenta/black/0" "magenta/black/1" )
```

Read remaining configuration files to see what's in them. Nevertheless, configuration can be also set from `zshrc` . There are `5` standard `zshrc` configuration variables:

```
znt_history_active_text - underline or reverse - how should be active element
highlighted
znt_history_nlist_coloring_pattern - pattern that can be used to colorize elements
znt_history_nlist_coloring_color - color with which to colorize
znt_history_nlist_coloring_match_multiple - should multiple matches be colorized (0 or
1)
znt_history_keywords (array) - search keywords activated with `Ctrl-X`, `F2` or
`Ctrl-/`, e.g. ( "git" "vim" )
```

Above variables will work for `n-history` tool. For other tools, change `_history_` to e.g. `_cd_`, for the `n-cd` tool. The same works for all `8` tools.

Common configuration of the tools uses variables with `_list_` in them:

```
znt_list_bold - should draw text in bold (0 or 1)
znt_list_colorpair - main pair of colors to be used, e.g "green/black"
znt_list_border - should draw borders around windows (0 or 1)
znt_list_themes (array) - list of themes to try out with Ctrl-T, e.g. (
"white/black/1" "green/black/0" )
znt_list_instant_select - should pressing enter in search mode leave tool (0 or 1)
```

If you used `ZNT` before `v2.1.12`, remove old configuration files `~/.config/znt/*.conf` so that `ZNT` can update them to the latest versions that support integration with `Zshrc`. If you used installer then run it again (after the remove of configuration files).

## Programming

The function `n-list` is used as follows:

```
n-list {element1} [element2] ... [elementN]
```

This is all that is needed to be done to have the features like ANSI coloring, incremental multi-word search, unique mode, horizontal scroll, non-selectable elements (grepping is done outside `n-list`, see the tools for how it can be done). To set up non-selectable entries add their indices into array `NLIST_NONSELECTABLE_ELEMENTS`:

```
typeset -a NLIST_NONSELECTABLE_ELEMENTS
NLIST_NONSELECTABLE_ELEMENTS=( 1 )
```

Result is stored as `$reply[REPLY]` ( `$` isn't needed before `REPLY` because of arithmetic context inside `[]` ). The returned array might be different from input arguments as `n-list` can process them via incremental search or uniq mode. `$REPLY` is the index in that possibly processed array. If `$REPLY` equals `-1` it means that no selection have been made (user quitted via `q` key).

To set up entries that can be jumped to with `[`, `]` keys add their indices to `NLIST_HOP_INDEXES` array:

```
typeset -a NLIST_HOP_INDEXES
NLIST_HOP_INDEXES=( 1 10 )
```

`n-list` can automatically colorize entries according to a `Zsh` pattern. Following example will colorize all numbers with blue:

```
local NLIST_COLORING_PATTERN="[0-9]##"
local NLIST_COLORING_COLOR=$'\x1b[00;34m'
local NLIST_COLORING_END_COLOR=$'\x1b[0m'
local NLIST_COLORING_MATCH_MULTIPLE=1

n-list "This is a number 123" "This line too has a number: 456"
```

Blue is the default color, it doesn't have to be set. See `zshexpn` man page for more information on `Zsh` patterns. Briefly, comparing to regular expressions, `(#s)` is `^`, `(#e)` is `$`, `#` is `*`, `##` is `+`. Alternative will work when in parenthesis, i.e. `(a|b)`. BTW by using this method you can colorize output of the tools, via their config files (check out e.g. n-cd.conf, it is using this).

## Performance

`ZNT` are fastest with `Zsh` before `5.0.6` and starting from `5.2`

## A tip

Zsh plugins may look scary, as they seem to have some "architecture". In fact, what a plugin really is, is that:

1. It has its directory added to `fpath`
2. It has any first `*.plugin.zsh` file sourced

That's it. When one contributes to Oh-My-Zsh or creates a plugin for any plugin manager, they only need to account for this. The same with doing any non-typical Zsh Navigation Tools installation.

## More

- be aware of [this](this)

# Fixing tmux, screen and linux vt

If `TERM=screen-256color` (often a case for `tmux` and `screen` sessions) then `ncv` terminfo capability will have `2` nd bit set. This in general means that underline won't work. To fix this by creating your own `ncv=0` - equipped terminfo file, run:

```
{ infocmp -x screen-256color; printf '\t%s\n' 'ncv@,'; } > /tmp/t && tic -x /tmp/t
```

A file will be created in directory `~/.terminfo` and will be automatically used, `tmux` and `screen` will work. Similar is for Linux virtual terminal:

```
{ infocmp -x linux; printf '\t%s\n' 'ncv@,'; } > /tmp/t && tic -x /tmp/t
```

It will not display underline properly, but will instead highlight by a color, which is quite nice. The same will not work for FreeBSD's vt, `ZNT` will detect if that vt is used and will revert to highlighting elements via `reverse` mode.