Basic use cases

Jump to bottom

N-R-K edited this page on Jan 31 · 189 revisions

The **nnn** magic!

`nnn` unfolds on use. Some exquisite stuff you can do:

- ☑ Instantly load, sort, filter thousands of files
- ☑ Filter and type to navigate with automatic matching dir entry
- ☑ List input stream and pick entries to stdout or file as list
- ☑ Write a plugin in any language and *talk* to `nnn`
- ☑ `find` / `fd` / `grep` / `ripgrep` / `fzf` from `nnn` and list in `nnn`
- ☑ Never lose context - start where you quit
- ☑ Mount any cloud storage service in a few keypresses
- ☑ Select files from anywhere (not just a single dir)
- ☑ Unlimited bookmarks, plugins, cmds with custom hotkeys
- ☑ Edit and preview markdown, man page, HTML
- ☑ Open a file and auto-advance to the next
- ☑ Filter filtered entries, export list of visible files
- ☑ Configure the middle mouse click to do anything
- ☑ Multi-location (or subtree) fuzzy search and visit a file
- ☑ Load four dirs with custom settings at once
- ☑ Notifications on `cp`, `mv`, `rm` completion
- ☑ Auto-sync selection to system clipboard
- ☑ Access selection from another instance of `nnn`
- ☑ Open text files detached in another pane/tab/window
- ☑ Mount and modify archives
- ☑ Create files/dirs/duplicates with parents (like `mkdir -p`)
- ☑ Toggle hidden with `.`, visit `$HOME` with `~`, last dir with `-`
- ☑ Mark a frequently visited dir at runtime
- ☑ Sort by modification, access and inode change time
- ☑ Compile out/in features with make variables
- ☑ Watch matrix text fly or read fortune messages
- ☑ Configure in 5 minutes!

Add bookmarks

There are 2 ways (can be used together) to manage bookmarks.

Environment variable **NNN_BMS**

Set environment variable `NNN_BMS` as a string of `key_char:location` pairs separated by semicolons (`;`):

```
export NNN_BMS="d:$HOME/Documents;u:/home/user/Cam Uploads;D:$HOME/Downloads/"
```

The bookmarks are listed in the help and config screen (key `?`).

The select bookmark key `b` lists all the bookmark keys set in `NNN_BMS` in the bookmarks prompt.

Symlinked bookmarks

A symlinked bookmark to the current directory can be created with the `B` key (or manually under `~/.config/nnn/bookmarks`).

Pressing `Enter` at the bookmarks prompt takes to this directory. If `NNN_BMS` is not set, the select bookmark key directly opens it.

Press `-` to return to the original directory right after entering a symlinked bookmark.

Mark directory

To jump back and forth a specific directory during a session, you can *mark* the directory (key `,`). Visit it anytime using the Bookmark key followed by `,`. If a directory is marked, the Bookmark key lists `,` as well.

Configure cd on quit

Pick the appropriate file for your shell from [misc/quitcd] and add the contents to your shell's rc file. You'll need to spawn a new shell for the change to take effect. You should start `nnn` as `n` (or your preferred function name/alias).

By default, when `^G` is pressed, `nnn` writes the last working directory to

```
${XDG_CONFIG_HOME:-$HOME/.config}/nnn/.lastd
```

To *cd on quit* always, export the environment variable `NNN_TMPFILE` set to your preferred file path.

The extra key `^G` to *cd on quit* is to avoid writing to a file every time `nnn` quits and allows on-demand *cd on quit*.

Sync subshell **$PWD**

To configure `nnn` to cd to the `$PWD` of the exited `bash` subshell, add the following in `.bashrc`:

```
nnn_cd()
{
    if ! [ -z "$NNN_PIPE" ]; then
        printf "%s\0" "0c${PWD}" > "${NNN_PIPE}" !&
    fi
}

trap nnn_cd EXIT
```

Add a plugin with the following contents:

```
#!/usr/bin/env bash

# Description: Start an interactive bash shell.

export nnn="$1"

bash -i
```

You can also use the *run cmd as plugin* method:

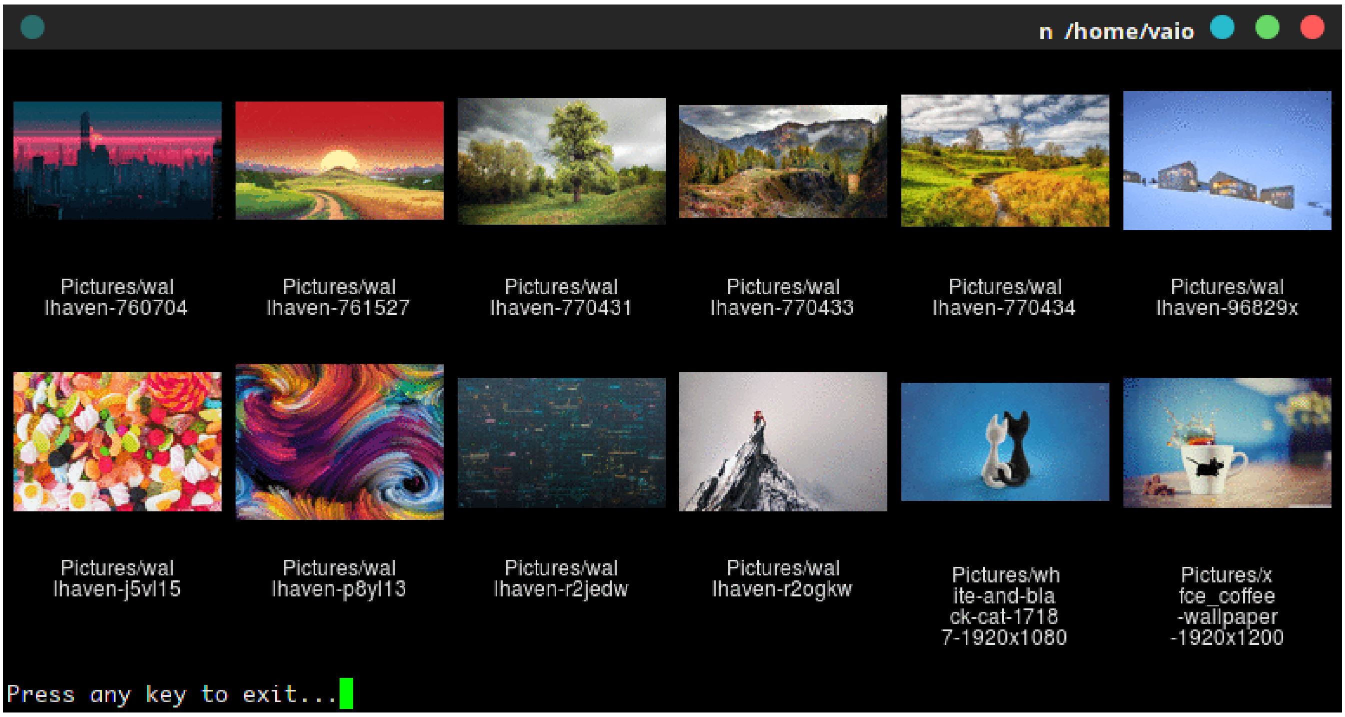
`s:!bash -i*`

Hot-plugged drives

Mount status external storage devices can be toggled in a REPL using the plugin nmount.

For auto-mounting external storage drives use udev rules or udisks wrappers.

Image, video, pdf



```
Pictures/wal      Pictures/wal      Pictures/wal      Pictures/wal      Pictures/wal      Pictures/wal
lhaven-760704     lhaven-761527     lhaven-770431     lhaven-770433     lhaven-770434     lhaven-96829x

Pictures/wal      Pictures/wal      Pictures/wal      Pictures/wal      Pictures/wh       Pictures/x
lhaven-j5vl15     lhaven-p8yl13     lhaven-r2jedw     lhaven-r2ogkw     ite-and-bla       fce_coffee
                                                                        ck-cat-1718       -wallpaper
                                                                        7-1920x1080       -1920x1200

Press any key to exit...
```

In addition to Live Previews, there is an independent plugin `imgview` to browse images/thumbnails of image, video or audio files in the terminal. Among other viewer utilities, it supports `sxiv` (opens its own window) which is particularly useful to browse images in a directory, set the wallpaper or copy image path to clipboard (instructions).

Browse, rename images

While the user can preview the hovered image in `preview-tabbed` and rename it, the following procedure is an alternative approach. It requires the `imgview` plugin (with `sxiv`) and `dmenu` (`zenity` can replace `dmenu`).

Save `sxiv` key-handler `~/.config/sxiv/exec/key-handler` with the following content:

```
#!/usr/bin/env sh

# ^X-^R: prompt for new image name (needs `dmenu`) and `mv`
# ^X-^C: copy the image path to X clipboard (needs `xclip`)

# Example zenity prompt:
#   name="$(zenity --entry --display=:0.0 --text "rename $file to")" 2> /dev/null

while read -r file
do
    case "$1"
        ")
                name="$(dmenu -p "rename $file to: ")" 2> /dev/null
                if [ -n "$name" ]; then
                    mv "

                ;;
        "C-c")
                printf "%s" "$file" | xclip -selection clipboard ;;
        *)
                printf "Sorry, I don't understand"
                exit 1
                ;;
    esac
done
```

Now hover on an image or directory and open it with the `imgsxiv` plugin. Browse images and press `^X` followed by `^R` to rename an image.

For file operations visit sxiv issue #228.

Read pdf as text

The plugin `pdfview` can convert and render PDF files as text in the terminal.

It has 2 methods to convert PDF to text. Please go through the script for the details.

Detached text

1. Set up a script (say `ewrap`) to open the editor in a new `tmux` split-pane or a new `xfce4-terminal` tab/window (you may want to change the terminal and the editor in the sample snippet below):

```
#!/usr/bin/env sh

if [ -n "$TMUX" ] ; then
    # tmux session running
    tmux split-window -h "vim \"$*\""
else
    # Remove option --tab for new window
    xfce4-terminal --tab -e "vim \"$*\""
fi
```

2. Make `ewrap` executable and drop it somewhere in your `$PATH` .
3. Set `$VISUAL` (or `$EDITOR` if you don't have `$VISUAL`) to `ewrap` :

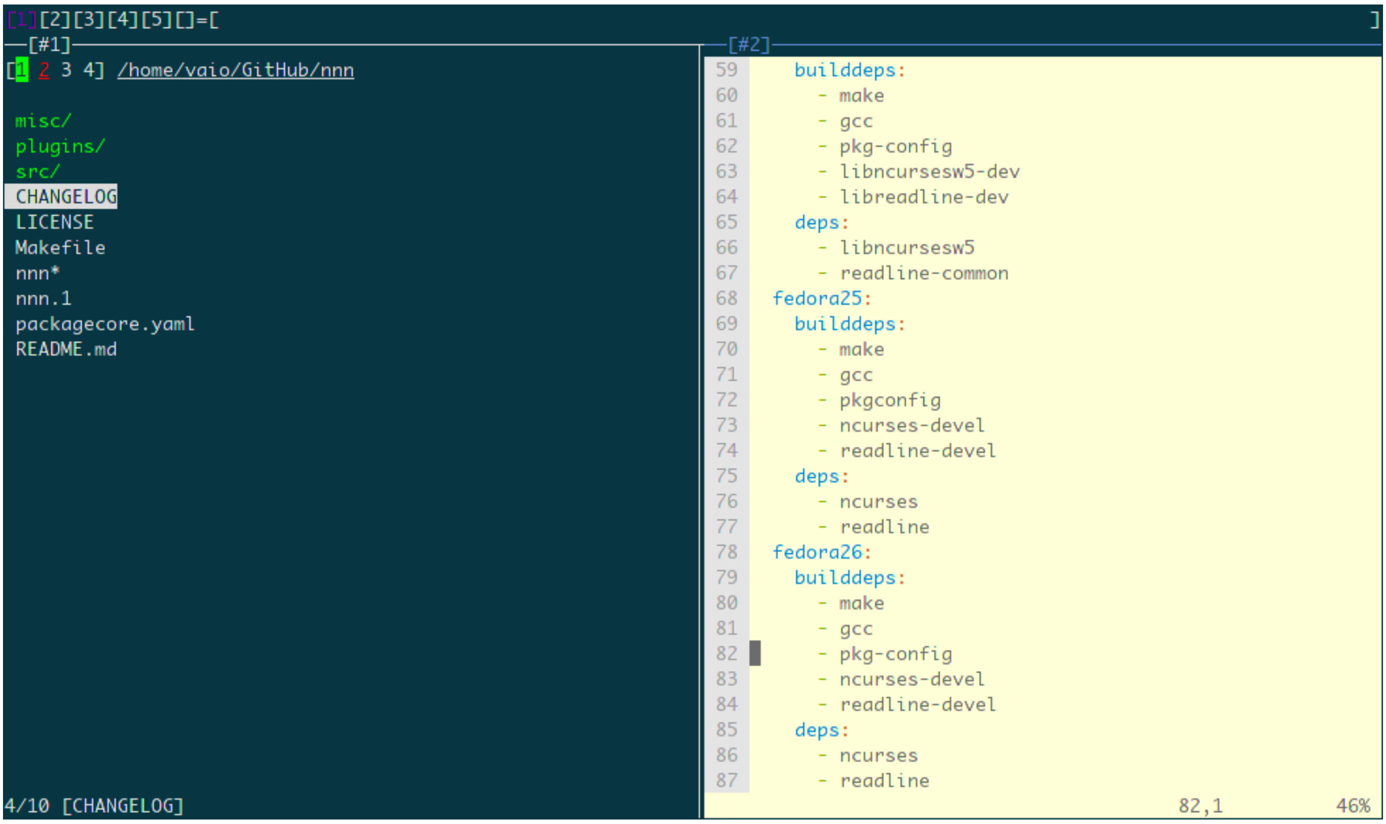
```
export VISUAL=ewrap
```

4. Start `nnn` with the program option `-e` .

Notes:

1. IMPORTANT: Use the program option `-E` with detached text editing enabled. `nnn` internal operations which need to wait for the editor (like batch rename, copy/move as) wouldn't work as usual because the control returns to `nnn` after the terminal emulator is spawned. To avoid this, set `$VISUAL` as above and `$EDITOR` to a CLI editor (like `vim`). The program option forces `$EDITOR` to be used for internal operations (overriding `$VISUAL` which takes precedence otherwise).
2. If you are already using `nuke` as opener modify it to use `ewrap` for text files. You can also enhance `ewrap` (re-use code from `nuke`) to handle text files by file type and use the right program (e.g. w3m for html, man for man pages). That way you don't have to change the opener.
3. If you are using a tiling window manager, the new terminal window with the editor would open beside your current window. In case of traditional window managers, the new window will be placed as per the window manager's placement configuration e.g. smart or centered on XFCE4.

Another options is to combine and use `nnn` with a multiplexer like `dvtm` :



```
[1][2][3][4][5][]=[                                                              ]
─[#1]──────────────────────────────────────────┬─[#2]─────────────────────────────
[1 2 3 4] /home/vaio/GitHub/nnn                    builddeps:
                                                     - make
 misc/                                               - gcc
 plugins/                                            - pkg-config
 src/                                                - libncursesw5-dev
 CHANGELOG                                           - libreadline-dev
 LICENSE                                           deps:
 Makefile                                            - libncursesw5
 nnn*                                                - readline-common
 nnn.1                                           fedora25:
 packagecore.yaml                                  builddeps:
 README.md                                           - make
                                                     - gcc
                                                     - pkgconfig
                                                     - ncurses-devel
                                                     - readline-devel
                                                   deps:
                                                     - ncurses
                                                     - readline
                                                 fedora26:
                                                   builddeps:
                                                     - make
                                                     - gcc
                                                     - pkg-config
                                                     - ncurses-devel
                                                     - readline-devel
                                                   deps:
                                                     - ncurses
                                                     - readline
4/10 [CHANGELOG]                                                   82,1          46%
```

Run commands

There are several ways to run commands from `nnn`:

1. Launch a shell within the current directory. This is your regular shell.
2. Use the prompt key to show the native command prompt and enter your command. The commands are invoked using the `$SHELL` so you can use commands, aliases (in `~/.zshenv` for zsh), environment variables, pipes, redirections.
3. You can also assign keys to arbitrary non-background cli commands (non-shell-interpreted) you use frequently and invoke like plugins. (instructions).

Shell depth indicator

If you use `!` to spawn a shell in the current directory and your shell is bash or zsh, it could be nice to add:

```
[ -n "$NNNLVL" ] && PS1="N$NNNLVL $PS1"
```

To your shell's rc. This will have your prompt indicate that you are within a shell that will return you to `nnn` when you are done.

This together with cd on quit becomes a powerful combination. In addition, the quitcd scripts check `$NNNLVL` to guard against nesting.

Launch applications

GUI applications can be launched using a drop-down menu using plugin `launch`. `nnn` picks it up from the plugin location. If `launch` is not found, `nnn` shows a regular prompt for the application executable name.

To configure `launch` as an independent app launcher add a keybind to open launch in a terminal e.g.,

```
xfce4-terminal -e "${XDG_CONFIG_HOME:-$HOME/.config}/nnn/plugins/launch
```

GUI applications can also be launched from the prompt. Use `&` to launch in background so `nnn` isn't blocked.

Note:

- You can also launch an instance of your favorite terminal in the current directory (*open terminal here...*) this way.

Open as root

If you are a sudoer, open a new instance of `nnn` from the built-in prompt with `sudo nnn`.

You can also have a short and sweet (but powerful) alias (which also works at the `nnn` prompt):

```
alias N='sudo -E nnn -dH'
```

File picker

To use `nnn` as a file picker and redirect the output to other programs, use option `-p`. The option takes an output file as argument (use `-` to print the list to stdout).

Ways to pick files:

- to pick a single file and quit, hover on it and press `Enter`
- to pick selection and quit, select file(s) and press `Enter` or `q`

Notes:

1. pressing `Enter` on a directory enters it; when trying to pick a directory, e.g. to pipe to `cd`, use single file selection
2. to discard selection and quit, press `^G`

Example picker mode use cases:

```
// list some files
ls -l $(nnn -p -)
-rw-rw-r-- 1 vaio vaio 14722 Jul 10 07:24 /home/vaio/GitHub/nnn/CHANGELOG
-rw-rw-r-- 1 vaio vaio  1472 Jul  7 21:49 /home/vaio/GitHub/nnn/LICENSE
-rw-rw-r-- 1 vaio vaio  1697 Jul 10 07:24 /home/vaio/GitHub/nnn/Makefile
```

```
// Attach some files and send a mail
neomutt -a $(nnn -p -) ...
```

```
// append some files to MOC playlist
```

```
mocp -a $(nnn -p -)
```

Remote mounts

`nnn` integrates with `sshfs` and `rclone` to support remote mounts. The mount points (for remotes as well as archives) are created within:

```
${XDG_CONFIG_HOME:-$HOME/.config}/nnn/mounts
```

Upon successful unmount, the mount points are removed.

The key to mount a remote folder is c (for *c*onnect).

Note: The `mounts` directory can be a symbolic link too. The target directory should have permissions to mount.

sshfs

Example primary setup: connect to SSH sever from desktop.

To connect to and mount remote shares using `sshfs`, `nnn` requires the ssh configuration file `~/.ssh/config` to have the host entries. sshfs reads this file.

Example host entry for a Termux environment on Android device:

```
Host *
    ServerAliveInterval 300
    ServerAliveCountMax 2

Host phone
    HostName 192.168.43.1
    User u0_a117
    Port 8022
    Compression no
    # Ciphers chacha20-poly1305@openssh.com
    # ProxyJump jumphost
```

Once you try to connect using `sshfs`, `nnn` will prompt for the name of the host (`phone` in the example above). Type the exact name (as it appears in this file) optionally followed by the path in the remote host you want to mount.

Examples:

```
phone         <----- Mount your remote user's $HOME folder in `phone`
phone:/tmp    <----- Mount the remote `tmp` folder in `phone`
```

Host `phone` will be mounted at `${XDG_CONFIG_HOME:-$HOME/.config}/nnn/mounts/phone`.

If you need to pass options to the `sshfs` command, you can do so:

```
export NNN_SSHFS='sshfs -o reconnect,idmap=user,cache_timeout=3600'
```

Options must be preceded by `sshfs` and comma-separated without any space between them.

rclone

The remote needs to be configured and authenticated beforehand (one-time activity). The name used to configure is all `nnn` needs to connect and mount (as well as unmount) the remote service locally.

If you need to pass options, use the dedicated environment variable for `rclone`:

```
export NNN_RCLONE='rclone mount --read-only --no-checksum'
```

A maximum of 5 flags can be passed.

Tip: To list all the remotes in the rclone config file: `rclone listremotes`

Unmount remotes

The unmount option (which also unmounts mounted archives BTW) works in 2 ways:

1. hover on the mount point, press `u` (for *u*nmount). This works for both remotes and archives.
2. if the current entry is not a mount point, `nnn` prompts for the remote host name

Notes:

1. `nnn` opens the mount point in a smart context after successful mounts.
2. More information on sshfs.

Synced quick notes

Open your note file OR directory (if your editor supports opening a dir like `vim`) in your favourite editor like a plugin (see plugin docs for more info):

```
export NNN_PLUG=n:-!vim /home/vaio/Dropbox/Public/Docs/Notes/note*
```

To keep the file synced across systems you may want to specify a file synced by a cloud storage service or stored in a network share.

Drag and drop

Use the plugin `dragdrop`.

Duplicate file

To duplicate the current entry:

- press `^R`
- keep the name unchanged in the prompt and press `Enter`
- in the next prompt ("copy name"), provide the preferred name (and optionally the path to the duplicate file) for the duplicate and press `Enter`

Note:

- To cancel the rename or duplicate operations, press Enter in both prompts without changing the file name.

Create batch links

To create symbolic or hard links in batch:

- select the file(s) you want to link
- go to the target directory
- press `n`
- press `s` for symbolic link or `h` for hard link
- enter the link prefix (or `@` for no prefix)

The reasons only prefix is allowed (and not full name):

- avoid prompting for link names one by one for a selection
- having the same prefix groups the newly created links together

Disable bold fonts

Most modern terminal emulators have the option to disable bold fonts.

Top ⬆
Wiki Home 📓
Project Home 🐴

▸ Pages 12

1. Release checklist
2. Verify sources
3. make options
4. Static compilation
5. Optimization tips
6. Compile for Pi
7. Cygwin instructions
8. Compile with musl libc
9. Cross-compile for Android
10. Debugging nnn

Clone this wiki locally

```
https://github.com/jarun/nnn.wiki.git
```