

# Using the FIM (Fbi IMproved) Universal Image Viewer

A scriptable and highly configurable, yet minimalistic Image Viewer for X, the Linux Framebuffer, and ASCII Art, for Command Line Users

Michele Martone

Open Media Devroom at FOSDEM  
4 February 2023

## Before FIM



*Young Hare*

Albrecht Dürer, 1502 (from Wikipedia)

- ▶ I was a user of Gerd Hoffmann's `fbi` (Linux framebuffer imageviewer)
- ▶ I wanted VIM-styled *arrow keys* `h` `j` `k` `l`

## FIM: *Fbi IMproved*




*Wolperdinger*

Rainer Zenz's edit on Albrecht Dürer's Young Hare (from Wikipedia)


fim began as a fork of fbi, inspired by


- ▶ the VIM text editor
- ▶ the Mutt mail user agent
- ▶ shell languages


## a Unix tool

 one task: viewing images

>\_ command line-oriented

 configuration files `/etc/fimrc`, `$HOME/.fimrc`

 regexes, stdin, stdout, scripting

 highly interoperable

*there's more than one way to do it (like in Perl and C++)*

**Warning: a viewer is *not* an editor**

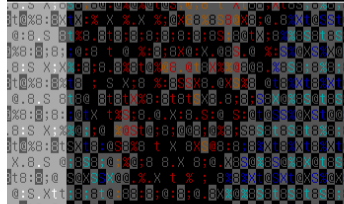
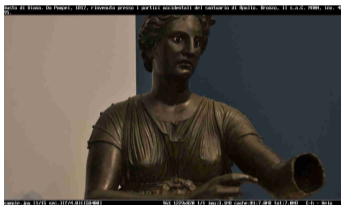
not here at least

## respects slower hardware and unusual setups

- ▶ image caching (spare repeated load)
- ▶ background prefetch (anticipate load)
- ▶ minimal user interface
- ▶ all video modes are optional:
  - ▶ X11 (`libSDL`)
  - ▶ Linux Framebuffer
  - ▶ ASCII Art (`libcaca/aalib`)
- ▶ friendly to ssh and screen

# an universal image viewer

- ▶ consistent look and feel
- ▶ several graphical modes
- ▶ `-o fb` / `-o sdl` vs `-o ca` vs `-o aa`:



## basic invocation >\_



```
$ fim file.jpg detect type by magic number
```



```
$ fim dir/ consider graphics-like filenames
```



```
$ fim -R dir/ recurse
```



```
$ fim -B dir/ background load
```

## keyboard-oriented interactive use

The scaling and “move around” keys you expect:

`+`, `-`, `Left`, `n`, `p`, `r`, `PageDown`, ...



*bind* `keys` with `actions` in the FIM language

- ▶ `+` `magnify`
- ▶ `-` `reduce`
- ▶ `Left` `pan_left`
- ▶ `n` `next`
- ▶ `p` `prev`
- ▶ `r` `rotate90`
- ▶ `Del` `list 'pop';reload`
- ▶ `PageDown` `if(_display_console==0){next;}else{scd;}`

commands "live" in the `:` command line mode

- ▶ `:` activates command line mode
- ▶ `Tab` triggers autocompletion
- ▶ same syntax as configuration files and scripts

## FIM's language elements

- ▶ `commands`
- ▶ `aliases` (customisable)
- ▶ `variables` (customisable)
- ▶ `if` and `while` blocks
- ▶ shortcut expressions and statements

## how do I use `vim`

- ▶ daily
  - ▶ interactive usage
  - ▶ organize pictures collections
- ▶ occasionally
  - ▶ use special functionality
  - ▶ use internal command line
- ▶ rarely
  - ▶ change configuration file
  - ▶ create shell aliases/functions with it

this talk's focus ↴

## showcase of FIM features

- ▶ not a tutorial
- ▶ not exhaustive
- ▶ for `language-specific topics`, see my other FOSDEM talk

simulate `$KeySym` press at startup

▶ `$ fim . -k $KeySym` *keysyms listed in man fimrc*

## interactive keys with `-k`

- ▶ `$ fim . -k r` *rotate once at startup*
- ▶ `$ fim . -k Del` *remove first image from list*
- ▶ `$ fim . -k C-h` *get help message*

key combos with `-K`

- ▶ `$ fim . -K rA` *rotate and autoscale*



prepend key press by number to iterate 

digits are special:

▶ `$ fim . -K 2r` *rotate two times*

repeat last key action by appending `.` (`recording 'repeat_last'`)

magnify, then magnify again, one more time:

▶ `$ fim . -K ++`

▶ `$ fim . -K +.`

▶ `$ fim . -K +1.`

magnify, then magnify again, *three more* times:

▶ `$ fim . -K +3.`

▶ `$ fim . -K ++++`

## don't overdo

◻ repeats *last key's* action only

▶ `vim . -K Z+n3.`

vs

▶ `vim . -K Z+nZ+nZ+nZ+n`

▶ For complex actions, customize `~/.vimrc`:

```
1 bind 'S' 'sleep 1;magnify;next;';
```

and

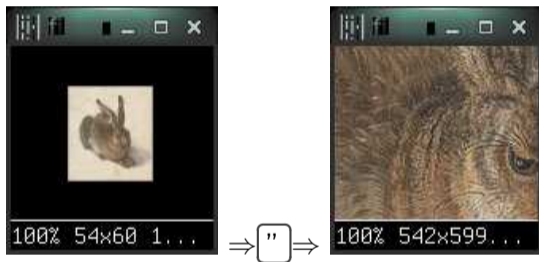
▶ `vim . -K 4S`

▶ `vim . -K S3.`

## browse *same-named* variants of a file

```
$ fim ~/lr/ --load-shadow ~/hr/
```

if viewing `~/lr/a.jpg`, pressing `"` replaces image with `~/hr/a.jpg`



```
1 fim -- -N -osdl=128 lr/ --load-shadow hr/
```

browse "thumbnails", load same-named hi-res equivalent.

## jump between last two views with `'` (apostrophe)

- ▶ retain positioning and scale



```
1 fim -q -c'_want_wm_caption_status=" "' -N -osdl=400:200 duerer_1502_hare.jpg  
   zenz_2005_wolperdinger.jpg -= "-K+kkn+kk"
```

- ▶ use with *shortlisting*:

`Enter` selects an image, ``` (backtick) limits list selection

## conversion pipelines

- ▶ on certain file extensions (**SVG**, DIA, XCF, ...)
- ▶ invoke an external converter (**inkscape**, ...)
- ▶ **-X** to inhibit

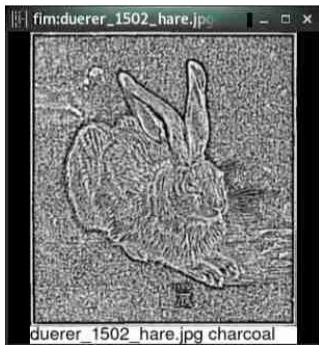


```
1 fim -q -osdl=240 /usr/share/inkscape/branding/tux.svg
```

## custom conversion pipelines

- ▶ filtering each image before viewing
- ▶ remember `find -exec 'cmd {}';' ?`

```
fim --pread-cmd="convert {} -charcoal 5 -pointsize 36 label:'{} charcoal' -append ppm:-" *
```



## custom actions using shell commands

```
: ! "identify" i:_filename
```

```
1 fim -N -q duerer_1502_hare.jpg -osdl=800:40 -K':! "identify" i:_filename' -K '' -=
```



```
fim:duerer_1502_hare.jpg100%[1/1]
duerer_1502_hare.jpg JPEG 542x599 542x599+0+0 8-bit sRGB 95117B 0.000u 0:00.000
|_ | C-h - Help
```



really, can't I edit images?

Warning: danger zone (you're discouraged from doing this) 

```
:alias "remove_exif_tags" "!'exiftool' '-all=' '--exif:DateTimeOriginal'  
i:_filename" # strip from EXIF tags
```


## load from images list file

```
$ cat objectors.list
```

```
1 daniel.jpg  
2 edward.jpg  
3 chelsea.jpg
```

```
$ fim -L objectors.list *
```

```
# for custom separators, use --read-from-stdin-elds
```

description file with `i:_comment` variable for each image 

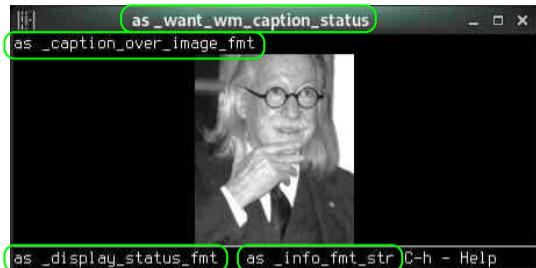
```
$ cat intellectuals.desc
```

```
1 ivan.jpg          1926-2002
2 joseph.jpg        1923-2008
3 noam.jpg           1928
```

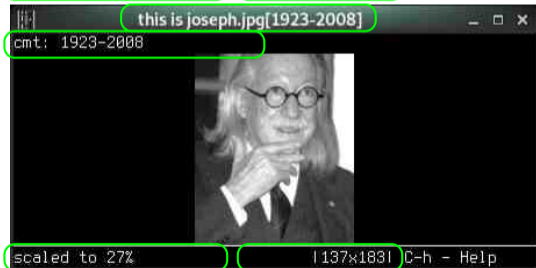
```
$ fim -D intellectuals.desc -L intellectuals.list
```

```
# for custom separators, use -S
```

use `i:_comment` in captions *expandos*



```
1 _caption_over_image=2;  
2 _info_fmt_str="as _info_fmt_str";  
3 _display_status_fmt="as _display_status_fmt";  
4 _want_wm_caption_status="as _want_wm_caption_status";  
5 _caption_over_image_fmt="as _caption_over_image_fmt";
```



```
1 _caption_over_image=2;  
2 _info_fmt_str="|%wx%h|";  
3 _display_status_fmt="scaled to %p%";  
4 _want_wm_caption_status="this is %N %k";  
5 _caption_over_image_fmt="%?_comment?cmt: %:_comment:?"
```

inherit custom variables' values into `i:`

```
$ cat programmers.desc
```

```
1 #!fim:category=businessman  
2 bill.jpg  
3 #!fim:category=activist  
4 aaron.jpg      RIP  
5 asbesto.jpg    https://museo.freaknet.org/  
6 bram.jpg       https://iccf.nl/  
7 richard.jpg    https://fsf.org/
```

## shebang (`#!fim:`) modifiers and `@variable` expansion

```
$ cat activists.desc
```

```
1 #!fim:~=she
2 #!fim:activity=programs useful tools
3 alexandra.jpg    @activity
4 greta.jpg        blasts leaders over greenwashing
5 #!fim:~=he
6 bram.jpg         @activity
```

## jump to file with matching description

- ▶ `:` goto `'/green/'`
- ▶ `/green` search line, a shortcut for the above
- ▶ `?green` reverse direction search line



```
1 fim -c'_want_wm_caption_status="%k"' -N -Dactivists.desc -Lactivists.list -osdl=400:200 -K/green -K ''
```

jump to the next file with a different "i:category" or "i:\_comment"

```
: goto '+category|_comment+'
```

```
$ cat programmers.desc
```

```
1  #!fim:category=businessman  
2  bill.jpg  
3  #!fim:category=activist  
4  aaron.jpg          RIP  
5  asbesto.jpg  
6  bram.jpg  
7  richard.jpg
```



## jump to first file in *next other* directory

1. `$ fim A/1.jpg A/2.jpg B/3.jpg`
2. `:` `goto '+/S'` *jump from A/1.jpg to B/3.jpg*

if many specs, jump to first one satisfied

Ⓜ

```
goto '+category|_comment+' '+/S' ...
```

*see help goto for more*

# temporarily shorten files list ✂



```
limit 'category' 'activist'
```



(images: Wikipedia)

## *limit* on other properties

- ▶ `:` `limit '~='` *files with same (duplicate) basename*
- ▶ `:` `limit '~d'` *all files from the same day*
- ▶ `:` `limit '~z'` *all files with the same size*
- ▶ ...

Mutt-inspired

## summary

FIM's base idea:

- ▶ identify your workflows and write a few `aliases`
- ▶ `bind` your favourite aliases to `keys`
- ▶ customize `~/.fimrc` with those
- ▶ only memorize what you use daily
- ▶ in doubt, consult `:` `help`

## the end

- ▶ on the web: <http://www.nongnu.org/fbi-improved>
- ▶ in the distros:
  - ▶ `$ sudo apt-get install fim` *Debian & family*
  - ▶ check out your distro, be it NixOS, Gentoo, arch, ...
- ▶ see `man fim` for options and `man fimrc` for language
- ▶ enjoy fim! (feedback welcome!)

### Reminder

For more of *fim's language*, check out my other FOSDEM'23 talk.

## extra: amateur files recovery

```
$ fim --offset 2:9 dump.bin seek image signature in bytes range (offset)
```

```
$ fim -C _seek_magic=GIF89a dump.bin try load only after specific signature
```

```
1 00000000: 6f66 6673 6574 0a47 4946 3839 610f 000f  offset.GIF89a...
2 00000010: 00b3 0e00 ffea 0045 4545 0000 00ff ce00  ....EEE.....
3 00000020: ffc9 00ff b400 fe9d 00ff fe93 fffd 13ff  ....
4 00000030: ffff ffff c733 3333 ffff ebf5 e500 0000  ....333.....
5 00000040: 0000 0000 21f9 0401 0000 0e00 2c00 0000  ....!.....,...
6 00000050: 000f 000f 0000 045b d049 196a 9dd8 55a6  ....[.I.j..U.
7 00000060: ce19 1716 7008 029c 8311 4c41 2708 27f0  ....p.....LA'. '
8 00000070: a6ab 7698 710e 6a0a 0e9f 3f82 2a70 03c6  ..v.q.j...?.*p..
9 00000080: 8285 ca21 c704 1092 01d3 a209 583c 2b03  ...!.....X<+.
10 00000090: d96b bb15 ae02 5941 623c 1610 769a c1a0  .k....YAb<..v...
11 000000a0: c135 d358 6a82 5cfe 1605 0af8 c230 c3b2  .5.Xj.\.....0..
12 000000b0: d426 1100 3b                .&..;
```

p.s. this is not how you do steganography ;-)