What flavour is it?

A gentle introduction to Albatross

Island of TEX

Version 0.5.0 – January 27, 2023

1 Introduction

> Albatross! Albatross!
> Albatross!
>
> Monty Python

Albatross is a command line tool for finding fonts that contain a given Unicode glyph. It relies on Fontconfig, a library for configuring and customizing font access. The tool is written in Kotlin and requires a Java virtual machine to run.

2 Requirements

> Two choc-ices please.
>
> Monty Python

Albatross has two hard requirements: a Java virtual machine (at least version 9, from any vendor) and the `fc-list` tool provided by the Fontconfig library, available in the system path. Linux and MacOS are known to have this library. For Windows, note that the TEX Live distribution contains Fontconfig tools. It is also highly recommended to use a terminal with Unicode support, as Albatross will try to render the given glyphs.

3 Basic use

> I haven't got choc-ices. I only got the albatross. Albatross!
>
> Monty Python

The tool is a typical command line application, so we need to invoke it by typing `albatross` in the terminal:

```
                __ __
.---.-.|  |  |--.---.-.|  |_.----.-----.-----.-----.
|  _  ||  |  _  |  _  ||   _|   _|  _  |__ --|__ --|
|___._||__|_____|___._||____|__| |_____|_____|_____|

Usage: albatross [OPTIONS] glyphs...

Options:
  -s, --show-styles                Show available font styles
  -d, --detailed                   Show a detailed font list
  -a, --ansi-level [n|a16|a256|tc]
                                   Set the default ANSI level
  -b, --border-style [0|1|2|3|4|5|6]
                                   Set the border style
  -o, --or                         Look for each glyph separately
  -t, --include-tex-fonts          Include fonts from the TeX tree
  -c, --clear-cache                Clear the font cache
  -V, --version                    Show the version and exit
  -h, --help                       Show this message and exit
```

Provided that Albatross is properly available in the underlying operating system, we will get the help message listing all the available options and the tool usage.

3.1 Glyphs

> What flavour is it?
>
> Monty Python

Albatross takes a list of glyphs, separated by spaces, as input. Three formats are supported by the command line tool:

– The glyph itself, e.g, ß (Eszett). Internally, the tool will convert it to the corresponding Unicode code point.

```
$ albatross ß
```

– The glyph as a Unicode code point in the hexadecimal notation, e.g, `0xDF`. The `0x` prefix is mandatory.

```
$ albatross 0xDF
```

Note that the tool takes the value as case insensitive, e.g, `0xDF` is equal to `0xdf` (or any case combination thereof).

– The glyph as a Unicode code point using the multiset union notation, e.g, `U+DF`. The `U+` prefix is mandatory.

```
$ albatross U+DF
```

Be mindful that this notation expects an uppercase `U`.

Formats can be used interchangeably.

It is worth noting that Albatross also provides proper grapheme support. According to this SO answer (reproduced verbatim):

> A *grapheme* is a sequence of one or more code points that are displayed as a single, graphical unit that a reader recognizes as a single element of the writing system. For example, both `a` and `ä` are graphemes, but they may consist of multiple code points (e.g. `ä` may be two code points, one for the base character `a` followed by one for the diaeresis; but there's also an alternative, legacy, single code point representing this grapheme). Some code points are never part of any grapheme (e.g. the zero-width non-joiner, or directional overrides).

Along these lines, in that same answer, we have a proper definition of a glyph is:

> A *glyph* is an image, usually stored in a font (which is a collection of glyphs), used to represent graphemes or parts thereof. Fonts may compose multiple glyphs into a single representation, for example, if the above `ä` is a single code point, a font may choose to render that as two separate, spatially overlaid glyphs.

When providing a glyph, Albatross will break it into multiple code points:

```
$ albatross ǔ
```

Since ỹ is composed of two code points, Albatross will query all fonts that have both `0x79` and `0x306`.

When a list of glyphs is provided, the tool will take a conjunctive approach and look for fonts that contain all elements in such list (default behaviour). Use the `-o` flag (or `--or` for the long flag) to look for each glyph separately. For instance:

- Look for fonts that contain both `a` and `b`:

```
$ albatross a b
```

- Look for fonts that contain `a` and fonts that contain `b`, separately:

```
$ albatross --or a b
```

3.2 Output

> It's a bird, innit? It's a bloody sea bird… it's not any bloody flavour. Albatross!
>
> Monty Python

Albatross prints the results as a table. The default behaviour is to just display the font names, e.g,

```
                    Unicode code point DF mapping to ß
┌──────────────────────────────────────────────────────────────────────────┐
│ Font name                                                                │
├──────────────────────────────────────────────────────────────────────────┤
│ 3270Medium Nerd Font                                                     │
├──────────────────────────────────────────────────────────────────────────┤
│ 3270Medium Nerd Font Mono                                                │
├──────────────────────────────────────────────────────────────────────────┤
...
├──────────────────────────────────────────────────────────────────────────┤
│ Zilla Slab,Zilla Slab Medium                                             │
├──────────────────────────────────────────────────────────────────────────┤
│ Zilla Slab,Zilla Slab SemiBold                                           │
└──────────────────────────────────────────────────────────────────────────┘
```

There is a `-s` option (or `--show-styles` for the long option) that includes the styles available for each font, e.g,

Unicode code point DF mapping to ß

Font name	Available styles
3270Medium Nerd Font	Medium
3270Medium Nerd Font Mono	Medium
...	
Zilla Slab,Zilla Slab Medium	Medium Italic, Italic, Medium, Regular
Zilla Slab,Zilla Slab SemiBold	SemiBold, Regular, SemiBold Italic, Italic

For even more details, including the font type and paths, there is the `-d` option (or `--detailed` for the long option), e.g,

Unicode code point DF, font details, mapping to ß

Name	3270Medium Nerd Font
Type	OpenType Font
Files	/home/paulo/.local/share/fonts/NerdFonts/3270-Medium Nerd Font Complete.otf

Unicode code point DF, font details, mapping to ß

Name	3270Medium Nerd Font Mono
Type	OpenType Font
Files	/home/paulo/.local/share/fonts/NerdFonts/3270-Medium Nerd Font Complete Mono.otf

...

Unicode code point DF, font details, mapping to ß

Name	Zilla Slab,Zilla Slab Medium
Type	OpenType Font
Files	/usr/share/fonts/mozilla-zilla-slab/ZillaSlab-MediumItalic.otf /usr/share/fonts/mozilla-zilla-slab/ZillaSlab-Medium.otf

Unicode code point DF, font details, mapping to ß

Name	Zilla Slab,Zilla Slab SemiBold
Type	OpenType Font
Files	/usr/share/fonts/mozilla-zilla-slab/ZillaSlab-SemiBold.otf /usr/share/fonts/mozilla-zilla-slab/ZillaSlab-SemiBoldItalic.…

For more verbosity, `--detailed` can be combined with `--show-styles` to include all font details, e.g,

Unicode code point DF, font details, mapping to ß

Name	3270Medium Nerd Font
Type	OpenType Font
Files	/home/paulo/.local/share/fonts/NerdFonts/3270-Medium Nerd Font Complete.otf
Styles	Medium

Unicode code point DF, font details, mapping to ß

Name	3270Medium Nerd Font Mono
Type	OpenType Font
Files	/home/paulo/.local/share/fonts/NerdFonts/3270-Medium Nerd Font Complete Mono.otf
Styles	Medium

...

Unicode code point DF, font details, mapping to ß

Name	Zilla Slab,Zilla Slab Medium
Type	OpenType Font
Files	/usr/share/fonts/mozilla-zilla-slab/ZillaSlab-MediumItalic.otf /usr/share/fonts/mozilla-zilla-slab/ZillaSlab-Medium.otf
Styles	Medium Italic, Italic, Medium, Regular

Unicode code point DF, font details, mapping to ß

Name	Zilla Slab,Zilla Slab SemiBold
Type	OpenType Font
Files	/usr/share/fonts/mozilla-zilla-slab/ZillaSlab-SemiBold.otf /usr/share/fonts/mozilla-zilla-slab/ZillaSlab-SemiBoldItalic.…
Styles	SemiBold, Regular, SemiBold Italic, Italic

Since the results can span several rows (the more common the glyph, the more fonts will contain it), we strongly recommend using a pipeline and pass the results to a terminal pager, e.g, the `less` utility:

```
$ albatross ß | less
```

3.3 Coloured output

> Do you get wafers with it?
>
> Monty Python

Albatross supports terminal colours by including the `-a` option (or `--ansi-level` for the long option) followed by the corresponding colour level. The following levels are available:

Level	Description
`n`	No colours at all (default)
`a16`	ANSI 16 colours
`a256`	ANSI 256 colours
`tc`	Support for true colours

Note that colours might mess the output when passed to a terminal pager. You might need to either adjust the terminal pager accordingly (e.g, `less -r` makes control characters to be displayed) or disable colours altogether (which is the default behaviour).

3.4 Table styles

> Course you don't get bloody wafers with it. Albatross!
>
> Monty Python

Albatross provides 7 table styles. Include the `-b` option (or `--border-style` for the long option) followed by the corresponding border style. The following styles are available:

- Style 0 (basically no borders):

```
Name: 3270Medium Nerd Font
Styles: Medium
...
```

- Style 1:

```
+------------------------------+----------------------------------------+
| Font name                    | Available styles                       |
+------------------------------+----------------------------------------+
| 3270Medium Nerd Font         | Medium                                 |
+------------------------------+----------------------------------------+
...
```

- Style 2:

```
╭──────────────────────────────────────────────────────────────────────────────╮
│ Font name                                                                    │
├──────────────────────────────────────────────────────────────────────────────┤
│ 3270Medium Nerd Font                                                         │
├──────────────────────────────────────────────────────────────────────────────┤
...
```

– Style 3:

```
  Font name                      Available styles

  3270Medium Nerd Font           Medium

...
```

– Style 4:

```
╔══════════════════════════════╦═══════════════════════════════════════════════╗
║ Font name                    ║ Available styles                              ║
╠══════════════════════════════╬═══════════════════════════════════════════════╣
║ 3270Medium Nerd Font         ║ Medium                                        ║
╚══════════════════════════════╩═══════════════════════════════════════════════╝
...
```

– Style 5:

```
┌──────────────────────────────┬───────────────────────────────────────────────┐
│ Font name                    │ Available styles                              │
├──────────────────────────────┼───────────────────────────────────────────────┤
│ 3270Medium Nerd Font         │ Medium                                        │
├──────────────────────────────┼───────────────────────────────────────────────┤
...
```

– Style 6:

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃ Font name                    ┃ Available styles                              ┃
┣━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╋━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┫
┃ 3270Medium Nerd Font         ┃ Medium                                        ┃
┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┻━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛
...
```

These are the keys to be used as option values:

Key	Description
0	Style 0 (no borders)
1	Style 1 (ASCII)
2	Style 2 (rounded corners)
3	Style 3 (blank)
4	Style 4 (double lines)
5	Style 5 (square corners, default)
6	Style 6 (heavy lines)

3.5 TEX fonts

How much is it?

Monty Python

Albatross can include fonts from the TEX tree in the query via the `-t` option (or `--include-tex-fonts` for the long option). Please note that the font indexing might take a while for the first time. For subsequent calls, Albatross will use a cache instead. To clear this cache, use the `-c` option (or `--clear-cache` for the long option).

License

Ninepence.

Monty Python

Albatross is licensed under the New BSD License. Please note that the New BSD License has been verified as a GPL-compatible free software license by the Free Software Foundation, and has been vetted as an open source license by the Open Source Initiative.

Changelog

I'll have two please.

Monty Python

0.5.0 (current)

Added

- Experimental font look up in the current TEX tree has been added.
- The new border style (none, `-b0`) hides the border of the tables entirely.
- Support for graphemes has been added. Note that, when searching for multiple glyphs using the `--or` modifier (disjunctive behavior), grapheme elements always will rely on conjunctive behavior.

Changed

- Table for font names (default view) has no internal borders now.
- Bump required Java version to 9. This drops support for Java 8.

Fixed

- The conjunctive behavior introduce in 0.3.0 has not been working as intended. Now it works as documented.

0.4.0 (2021-11-22)

- Included support for the Unicode code point using the `U+` multiset union notation, which behaves exactly the same as the `0x` counterpart. So ß, `0xDF` and `U+DF` denote the same Unicode entity.

0.3.0 (2021-01-13)

Changed

- Conjunctive behavior is now default. Previously, `albatross a b` would have looked for fonts for `a` and separately for fonts for `b`. As we see more use cases for looking for fonts that contain `a` as well as `b`, we changed the default and left the previous behavior as `albatross --or a b`.

0.2.0 (2020-12-09)

Added

- Inclusion of a man page.

Fixed

- Windows paths were incorrectly parsed, causing font names and styles to be displayed incorrectly.

0.1.0 (2020-12-07)

- Initial release.

The team

> Gannet on a stick.
>
> —Monty Python

Albatross is brought to you by the Island of TEX. If you want to support TEX development by a donation, the best way to do this is donating to the TEX Users Group.

No albatrosses were harmed during the making of this user manual.