emo: emoji for all (LaTeX engines)

Robert Grimm

Version v0.1 (2023/03/16)

Abstract

Emo implements the `\emo{`⟨*emoji-name*⟩`}` command for including color emoji such as 🏝 or 🦜 in your documents independent of input encoding or LaTeX engine. The implementation uses the Noto color emoji font if the engine supports it and includes PDF graphics otherwise. The latter are derived from Noto's SVG sources, so the visual appearance is very similar. The source repository is at https://github.com/apparebit/emo.

Contents

1 Installation

Installation of the emo package is fairly straightforward, though it does involve a lot more files than usual.

1. Start by extracting this package's files from `emo.dtx` by running:

   ```
   $ pdflatex emo.dtx
   ```

2. Build the package documentation with indices by running:

```
$ source build.sh
```

3. If you want to reconfigure the supported emoji, see §3 below.
4. Put the following files somewhere LaTeX can find them. While your current project's directory will do, emo's installation potentially comprises thousands of files. Hence it may be preferrable to adhere to the TeX Directory Structure, with dedicated directories for the following files:
   (a) `emo.sty` with the package implementation;
   (b) `emo.def` with the emoji table;
   (c) `NotoSerifTC-Regular.otf` for `\lingchi`;
   (d) *all* PDF files in the `emo-graphics` directory.

When running on the LuaLaTeX engine, the emo package also uses the Noto color emoji and Linux Libertine fonts, with the latter used to render `\YHWH`. Neither file is included with emo's distribution, since both of them are distributed with major TeX distributions already. If they are not included with your LaTeX distribution, you can find them on CTAN.

2 Usage

As usual, you declare your document's dependency on emo with `\usepackage{emo}`. In addition to the unadorned form, emo takes up to two options:

extra Also define the `\lingchi` and `\YHWH` macros, which produce 凌遲 and יהוה, respectively, and are documented below.

index Create an emoji index tagged `emo` with the `.edx` extension for the raw index and the `.end` extension for the processed index. This option relies on the `index` package, generates the raw `.edx` file, but does not build or use the processed index.

`\emo` An `\emo{`⟨*emoji-name*⟩`}` invocation expands to the named emoji. For LuaLaTeX, it uses the Noto color emoji font. For all other engines, it uses PDF graphics. That way, `\emo{desert-island}` results in 🏝 and `\emo{parrot}` results in 🦜.

Since LaTeX tends to produce a lot of command line noise about underfull boxes and loaded fonts, it's a easy to miss meaningful warnings. For that reason, `\emo` expands to an attention-seeking error message upon undefined emoji names. For example, `\emo{boo}` produces `Bad \emo{boo}`.

2.1 Emoji Names

With some exceptions, emo's names for emoji are automatically derived from their Unicode names, with letters converted to lowercase, punctuation such as commas, colons, quotes, and parentheses stripped, and interword spaces replaced by dashes. Furthermore, instead of the rather verbose `dark-skin-tone`, `medium-dark-skin-tone`, etc modifiers, emo uses the more succinct `darkest`, `darker`, `medium`, `lighter`, and `lightest`.

For some emoji names, emo goes further by hard-coding shorter names. Those names are listed in Table 1.

Emo's `emo.def` contains the names and codepoints of all currently supported emoji. Its distribution also includes the `emoji-test.txt` file, which is part of Unicode TR-51 and contains the names and codepoints of all *potentially* supported emoji, i.e., all emoji. It further organizes emoji into groups and subgroups, with the current (sub)group being the one named on the closest line above the emoji that starts with `# (sub)group:`.

2.2 Extras

`\lingchi` `\YHWH` The `\lingchi` and `\YHWH` macros take no arguments and produce 凌遲 and יהוה, respectively. They are only available if emo is used with the `extra` option. The former renders the Chinese term for "death by a thousand cuts." While originally an execution method, the term applies to surprisingly many software systems as well. The latter produces the Tetragrammaton, the Hebrew name for God. Observant Jews never utter what's written, not even in their thoughts, substituting Adonai ("My Lord"), Elohim ("God"), or HaShem ("The Name") instead. In my mind, that nicely mirrors the very incomprehensibility of יהוה. Both macros preserve a subsequent space as space, no backslash needed.

3 Configuration

Emo's implementation is split over two files, `emo.sty` and `emo.def`. The former file defines the substance of the package, its options, its helper macros, and the user-visible `\emo`, `\lingchi`, and `\YHWH` macros. The latter file defines the table of supported emoji. For each such emoji, the table contains a command `\emo@emoji@`⟨*emoji-name*⟩ with the emoji's codepoints as value. `emo.sty` includes the table during loading and thereafter relies on it for validating emoji names and rendering emoji in LuaLaTeX. It may be possible to combine the two files into one, but that avoids at most a few disc reads and hence doesn't seem worth the trouble.

3.1 Update Configuration

Invoke `config/emo.py` to automatically reconfigures emo's supported emoji:

```
$ python3 config/emo.py ⟨selector⟩ ⟨selector⟩ ...
```

Each selector may be:

- The literal `ALL` (case-sensitive) for *all* emoji.
- Name of a group in `emoji-test.txt` lowercased and with spaces replaced by dashes and ampersand `&` replaced by an `and`; e.g., `travel-and-places`.
- Name of a group, a double colon `::`, and name of a subgroup, again lowercased and with spaces replaced by dashes and `&` by an `and`; e.g., `travel-and-places::place-geographic`.
- The name of an emoji; e.g., `desert-island`.

Table 1: Exceptional emoji names

Transformed Unicode Name	Emo Replacement Name
a-button-blood-type	a-button
ab-button-blood-type	ab-button
b-button-blood-type	b-button
o-button-blood-type	o-button
bust-in-silhouette	bust
busts-in-silhouette	busts
flag-european-union	eu
globe-showing-americas	globe-americas
globe-showing-asia-australia	globe-asia-australia
globe-showing-europe-africa	globe-africa-europe
hear-no-evil-monkey	hear-no-evil
index-pointing-at-the-viewer	index-pointing-at-viewer
index-pointing-at-the-viewer-darkest	index-pointing-at-viewer-darkest
index-pointing-at-the-viewer-darker	index-pointing-at-viewer-darker
index-pointing-at-the-viewer-medium	index-pointing-at-viewer-medium
index-pointing-at-the-viewer-lighter	index-pointing-at-viewer-lighter
index-pointing-at-the-viewer-lightest	index-pointing-at-viewer-lightest
keycap-*	keycap-star
keycap-#	keycap-hash
keycap-0	keycap-zero
keycap-1	keycap-one
keycap-2	keycap-two
keycap-3	keycap-three
keycap-4	keycap-four
keycap-5	keycap-five
keycap-6	keycap-six
keycap-7	keycap-seven
keycap-8	keycap-eight
keycap-9	keycap-nine
keycap-10	keycap-ten
magnifying-glass-tilted-left	loupe-left
magnifying-glass-tilted-right	loupe-right
palm-down-hand	palm-down
palm-down-hand-darkest	palm-down-darkest
palm-down-hand-darker	palm-down-darker
palm-down-hand-medium	palm-down-medium
palm-down-hand-lighter	palm-down-lighter
palm-down-hand-lightest	palm-down-lightest
palm-up-hand	palm-up
palm-up-hand-darkest	palm-up-darkest
palm-up-hand-darker	palm-up-darker
palm-up-hand-medium	palm-up-medium
palm-up-hand-lighter	palm-up-lighter
palm-up-hand-lightest	palm-up-lightest
rolling-on-the-floor-laughing	rofl
see-no-evil-monkey	see-no-evil
speak-no-evil-monkey	speak-no-evil

For data safety, `emo.py` does not overwrite PDF graphics and hence can only add emoji to the configuration. To remove emoji, simply remove their PDF graphics from `emo-graphics` and then run `emo.py`, which updates the emoji table accordingly.

`emo.py` effectively treats `emoji-test.txt` as registry of all emoji and the filenames of PDF graphics in `emo-graphics` as emo's current inventory. For all emoji named by selector arguments but not in the inventory, `emo.py` converts the SVG source graphic from the Noto color emoji sources to a PDF file and deletes the `/Page /Group` object from the the PDF again, since that object trips up `pdflatex`. And yeah, `emo.py` automatically downloads the Noto color emoji sources if necessary.

4 Implementation

Now that we understand how to configure emo, we are ready for exploring the implementation in detail. Let's get started:

```
⟨*package⟩
```

Except, the package implementation started near the top of the `emo.dtx` file, so that version number and date are more visible and declared only once. But that's also well before the documentation preamble and hence cannot be included in the annotated implementation. Nonetheless, we can simulate the lines:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{emo}[2023/03/16 v0.1 emo for all]
```

And no, I didn't repeat the version number and date. Check `emo.dtx`.

4.1 Package Options

Emo's extra and index options are simple flags. So, we declare a `\newif` for each option and, if the package use includes an option, it just toggles the conditional's state:

```
\newif\ifemo@extra\emo@extrafalse
\DeclareOption{extra}{\emo@extratrue}
\newif\ifemo@indexing\emo@indexingfalse
\DeclareOption{index}{\emo@indexingtrue}
\ProcessOptions\relax
```

4.2 Dependencies

The dependency on `inputenc` effectively declares this file's encoding to be UTF-8. The XeTeX and LuaTeX engines already expect files to be encoded that way and hence ignore the declaration. However, pdfTeX supports other (legacy) encodings and needs to be told.

```
\RequirePackage[utf8]{inputenc}
```

Depending on TeX engine, this package requires either `fontspec` or `graphicx` as the emoji-emitting backend. In turn, to tell the engines apart, it requires `iftex`.

```
\RequirePackage{iftex}
\ifluatex
\RequirePackage{fontspec}
\else
\RequirePackage{graphicx}
\fi
```

Emo requires `xcolor` for formatting highly visible error messages within the text. Always including another package that is only used when there are errors is not ideal. But when I tried calling `\RequirePackage` for `xcolor` from inside the error macro, it didn't work. Alternatively, I could make in-text errors optional.

```
\RequirePackage{xcolor}
```

Finally, emo's options also have dependencies, with extra requiring the `xspace` package and index requiring the `index` package:

```
\ifemo@extra
\RequirePackage{xspace}
\fi
\ifemo@indexing
\RequirePackage{index}
\fi
```

4.3 The Emoji Table

For each emoji with a PDF graphic in the `emo-graphics` directory, a macro named `\emo@emoji@`⟨*emoji-name*⟩ expands to its Unicode sequence. With over 3,000 distinct emoji in Unicode 15, emo relies on a Python script for populating the graphics directory and writing the table to the `emo.def` file. Since the package code does not change after installation but the emoji table may very well change, they are kept separate for now. Alternatively, we could use DocStrip to assemble the package file from three parts, the code from the previous sections, then the contents of the emoji table in `emo.def`, and then all subsequent code.

```
\input{emo.def}
```

4.4 Internal Macros

`emo@error@fg` `emo@error@bg` `emo@error` Define two colors and a function that uses the two colors for formatting an attention-grabbing error message. If you use an invalid emoji name and overlook the warning in the console, you *will* notice the error messsage in the document thusly formatted.

```
\definecolor{emo@error@fg}{rgb}{1,1,1}
\definecolor{emo@error@bg}{rgb}{.6824,.0863,.0863}
```

```
\def\emo@error#1{%
    \colorbox{emo@error@bg}{%
        \textcolor{emo@error@fg}{%
            \textsf{Bad} \texttt{\textbackslash emo\{#1\}}%
        }%
    }%
}
```

`emo@ifdef` Validate the emoji name given as first argument. The macro expands to the second argument if the name is valid and an error message otherwise. Its implementation relies on the `emo@emoji` table.

```
\def\emo@ifdef#1#2{%
    \ifcsname emo@emoji@#1\endcsname#2\else%
       \PackageWarning{emo}{Unknown emoji name in `\string\emo{#1}'}%
        \emo@error{#1}%
    \fi%
}
```

`emo@index` If indexing is enabled, record the use of an emoji. Otherwise, do nothing.

```
\ifemo@indexing
\newindex{emo}{edx}{end}{Emoji Index}
\def\emo@index#1{\index[emo]{#1}}
\else
\def\emo@index#1{}
\fi
```

4.5 User Macros

`emo` Emit the named color emoji. Both the font-based version for LuaTeX and the graphics-based fallback validate the emoji name and then invoke the `\emo@index` macro. But they differ in how they actually display the emoji. The LuaTeX version turns the emoji name into its Unicode sequence and wraps that in a group that also uses the previously declared Noto color emoji font. The fallback version instead includes a suitably sized PDF graphic.

```
\ifluatex
\newfontface\emo@font[Renderer=Harfbuzz]{NotoColorEmoji.ttf}
\newcommand\emo[1]{%
    \emo@ifdef{#1}{%
        \emo@index{#1}%
        {\emo@font\csname emo@emoji@#1\endcsname}%
    }%
}
\else
\newcommand\emo[1]{%
    \emo@ifdef{#1}{%
        \emo@index{#1}%
     \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/#1}}%
```

```
    }%
}
\fi
```

lingchi YHWH

The definitions for the optional \lingchi and \YHWH macros follow from that of \emo, except that (a) there are no arguments to validate and hence no equivalent to \emo@ifdef; (b) Hebrew is written right-to-left and hence \YHWH requires a \textdir TRT; (c) subsequent space should be preserved and hence both macros end with \xspace. While it would be nice to use Unicode inside the groups for the LuaTeX macros, doing so breaks the package documentation. So \char it is.

```
\ifemo@extra
\ifluatex
\newfontface\emo@chinese{NotoSerifTC-Regular.otf}
\newfontface\emo@hebrew{LinLibertine_R.otf}
\newcommand\lingchi{%
    \emo@index{lingchi}%
    \begingroup\emo@chinese \char"51CC\char"9072\endgroup%
    \xspace}
\newcommand\YHWH{%
    \emo@index{YHWH}%
  \begingroup\textdir TRT\emo@hebrew \char"5D9\char"5D4\char"5D5\char"5D4\endgroup%
    \xspace}
\else
\newcommand\lingchi{%
    \emo@index{lingchi}%
  \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/lingchi}}%
    \xspace}
\newcommand\YHWH{%
    \emo@index{YHWH}%
  \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/YHWH}}%
    \xspace}
\fi
\fi
```

Et voilà. That's it!

⟨/package⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.