Thge bxcoloremoji Package

Takayuki YATO (aka. “ZR”)

v1.0a [2024/11/18]

Abstract

The bxcoloremoji package lets users output color emojis in LaTeX documents. Compared to other packages with similar functionality, this package has the following merits:

- It supports all major LaTeX engines.
- Emojis can be entered as the characters themselves, as their Unicode code values, or as their short names.
- It works reasonably well in PDF strings when using hyperref.
- Emojis can be handled properly even in Japanese typesetting environments.

This package has been widely used among the Japanese LaTeX community, but there are already many emoji packages on CTAN and TeX Live. To avoid uploading a large amount of emoji image data that are essentially identical, the package was revised in version 1.0 so that the image output was delegated to the twmojis package. Therefore, this package now contains no image data.

Nevertheless, this package also supports the use of custom image sets (*custom families*) prepared by the user for output.

Contents

1 System Requirements

- TeX format : LaTeX
- TeX engine : Anything that supports the ε-TeX extension.
- Dependent packages :
  - etoolbox

- binhex (when expl3 is not available)
- bxghost (optionally)
- twemojis

Important note: This English documentation omits much of the information that is specific to Japanese typesetting and/or Japanese engines (i. e.(u)pLaTeX).

2 Package loading

For older (prior to 2018-04-01) pdfLaTeX, you need to enable `utf8` input encoding.

```
\usepackage[utf8]{inputenc}% for old systems
```

Then load the bxcoloremoji package.

```
\usepackage[⟨option⟩,…]{bxcoloremoji}
```

Note: For DVI output, you need to specify a global driver option.

```
\usepackage[dvipdfmx,a4paper]{article}% for dvipdfmx
```

2.1 Package Options

The available options are as follows:

- **Configuration parameters** : The configuration parameters listed in Section 3 (e. g.`size`, `scale`, etc.) can also be specified as package options.
- `names=`⟨*bool*⟩ : Whether to load the short name database.
  Note: If `false` is specified, only special short names (see Section 5) can be used.

The following are for advanced users:

- `nodvidriver` : Suppresses driver-dependent behavior. Specifically, the emoji image set type is fixed to `no-image` (emojis are not displayed).
- `resetdvidriver` : The negation of `nodvidriver`.
- `preload-names=`⟨*value*⟩ : Whether to load the short name database at once when loading the package.
  Note: Here `false` is a setting for the old TeX engine with low memory capacity.
  - `auto` (default) : If *any* of the following conditions are met then `true` is used (assuming there is enough memory), otherwise `false` is used.
    * The engine is either XeLaTeX or LuaLaTeX.
    * expl3 is enabled.
    * hyperref is loaded.
  - `true` : Loads all data when loading the package.
  - `false` : Loads on demand.
- `bbparam=`⟨*value*⟩ : Whether to specify the `bb` parameter for `\includegraphics` for emoji output.
  - `auto` (default) : Specifies `bb` only if the graphicx driver is dvipdfmx.
    Note: When using dvipdfmx, specifying `bb` will make the build faster.
  - `true`/`false` : Always/never specifies `bb`.
    Note: Some drivers prohibit the `bb` setting.

3 Configuration Parameters

Configuration parameters can be used in the following places:

- To specify them as package options.

  ```
  \usepackage[scale=2]{bxcoloremoji}
  ```

- To specify them as arguments to the `\coloremojisetup` command. The settings are changed on the spot and applied to subsequent emoji output commands.

  ```
  \coloremojisetup{scale=2}
  ```

- To specify as the first optional argument to an emoji output command. The setting is applied only to that emoji output.

  ```
  \coloremoji[scale=2]{⛄}
  ```

The available configuration parameters are as follows:

- **The type of emoji image set.** (default: `twemojis`[1])
  - `twemojis` : Delegates image output to the command of the twemojis package (the twemojis mode).
  - `family=`⟨*name*⟩ : Uses a custom family.
  - `no-image` : Uses fallback output only without using any emoji images.
- `size=`⟨*length*⟩ : The size of emojis. (default: 1 em)
- `scale=`⟨*real*⟩ : Changes the size of emojis by the given factor from `size`. (default: 1)

4 Usage

Note: For commands in this package, braces around mandatory arguments *cannot* be omitted.

4.1 Basic Emoji Commands

- `\coloremojisetup{`⟨*parameters*⟩`}` : Changes the parameter settings (see Section 3).

  Note: Here ⟨*parameters*⟩ represents a list of the form "⟨*key*⟩`=`⟨*value*⟩`,`…". The same applies below.

- `\coloremoji[`⟨*parameters*⟩`]{`⟨*string*⟩`}` : Outputs the argument string as color emojis. However, if it cannot be output as emojis because the target image is missing, it will fall back to normal text output.

  ```
  \coloremoji{🧨3🙃⛄}%Output:🧨3🙃⛄
  ```

- `\coloremojicode[`⟨*parameters*⟩`]{`⟨*code value list*⟩`}` : Outputs color emojis for the characters input as Unicode code values or short names defined in the JoyPixels emoji-toolkit library.[2] A code value is represented by the hexadecimal notation, and a short name is represented by a string of the format "`:`⟨*short-name*⟩`:`". When multiple characters are entered, each character must be delimited with a space.

  ```
  \coloremojicode{:sushi: 23 20E3 1F643 :snowman:}%Output:🍣#⃣🙃⛄
  ```

[1] The default value has been changed to `twemojis` since version 1.0. In addition, `twitter`, `hires` and `lowres`, which were deprecated since version 0.12, have been *abolished*.

[2] The emoji-toolkit library: https://github.com/joypixels/emoji-toolkit

Note: Hereafter, this input method will be called *code value list*.

- `\coloremojiucs[`⟨*parameters*⟩`]{`⟨*code value list*⟩`}` : An alias for `\coloremojicode` in the old version.

  Note: For other commands and environments with the name `coloremojicode…`, those that existed before version 0.4 have the alias `coloremojiucs…` similarly.

4.2 The counter output command for outputting keycap emojis

The following commands are provided to output an integer value between 0 and 10 as the keycap emoji (0️⃣–🔟) that corresponds to the value.
Note: If the input value is out of range, a fallback output will be used.

- `\coloremojikeycapof[`⟨*parameters*⟩`]{`⟨*number*⟩`}` : Outputs the keycap emoji corresponding to the given number.

  ```
  \coloremojikeycapof{8}%Output:8️⃣
  ```

- `\coloremojikeycap[`⟨*parameters*⟩`]{`⟨*counter name*⟩`}` : Outputs the keycap emoji corresponding to the current value of the given counter.

- `\pagenumbering{coloremojikeycap}` : Changes the page number format to keycap emoji.

  Note: `\pagenumbering{coloremojikeycap}` is employed in this document.

4.3 Features similar to the pifont package

The following commands are provided, which are the emoji versions of the pifont package features (such as the `\dingfill` command and the `dingautolist` environment).
Note: The commands and environments listed here cannot have parameter options.

1️⃣ `\coloremojifill{`⟨*string*⟩`}` : A filler command (like `\dotfill`) that fills a line by arranging multiple outputs of `\coloremoji{`⟨*string*⟩`}`.

2️⃣ `\coloremojiline{`⟨*string*⟩`}` : Outputs a decorative border using emojis. That is, it outputs a separate line containing only the output of `\coloremojifill{`⟨*string*⟩`}` (but with some space on both ends).

3️⃣ `\begin{coloremojilist}{`⟨*string*⟩`}`…`\end{coloremojilist}` : Outputs a bulleted list with the output of `\coloremoji{`⟨*string*⟩`}` as item label.

4️⃣ `\begin{coloremojiautolist}{`⟨*string*⟩`}`…`\end{coloremojiautolist}` : This is also an environment that outputs a bulleted list with emojis as item label, but the argument must be one of the emojis in one of the "emoji orders". It determines the label according to the emoji order starting from that character.

For example, in the list created by `\begin{coloremojiautolist}{`♠️`}` the first label is "♠️", and the subsequent labels go "♥️", "♦️", and "♣️".

Note: The list you see now is generated by `\begin{coloremojiautolist}{`1️⃣`}`.

5️⃣ For the commands and environments mentioned above, there are also versions that use a code value list as an argument.

- ✏️ `\coloremojicodefill{`⟨*code value list*⟩`}`
- ✏️ `\coloremojicodeline{`⟨*code value list*⟩`}`
- ✏️ `\begin{coloremojicodelist}{`⟨*code value list*⟩`}`

Table 1: The list of special short names.

`+`	U+200D		(ZWJ)
`/1`	U+1F3FB		(light skin tone modifier)
`/2`	U+1F3FC		(medium-light skin tone modifier)
`/3`	U+1F3FD		(medium skin tone modifier)
`/4`	U+1F3FE		(medium-dark skin tone modifier)
`/5`	U+1F3FF		(dark skin tone modifier)
`!/red`	U+1F9B0		(“`+ !/red`”forms a hair style)
`!/curly`	U+1F9B1		(“`+ !/curly`”forms a hair style)
`!/bald`	U+1F9B2		(“`+ !/bald`”forms a hair style)
`!/white`	U+1F9B3		(“`+ !/white`”forms a hair style)
`!black`	U+2B1B		(“`+ !black`”forms a color indicator)
`!white`	U+2B1C		(“`+ !white`”forms a color indicator)
`!red`	U+1F7E5		(“`+ !red`”forms a color indicator)
`!blue`	U+1F7E6		(“`+ !blue`”forms a color indicator)
`!orange`	U+1F7E7		(“`+ !orange`”forms a color indicator)
`!yellow`	U+1F7E8		(“`+ !yellow`”forms a color indicator)
`!green`	U+1F7E9		(“`+ !green`”forms a color indicator)
`!purple`	U+1F7EA		(“`+ !purple`”forms a color indicator)
`!brown`	U+1F7EB		(“`+ !brown`”forms a color indicator)
`!female`	U+2640		(“`+ !female`”forms a gender indicator)
`!male`	U+2642		(“`+ !male`”forms a gender indicator)
`!flag`	U+1F3F4		(the base of tag sequence for flags)
`!<`	U+2B05		(“`+ !<`”forms a direction indicator)
`!>`	U+27A1		(“`+ !>`”forms a direction indicator)
`!A`–`!Z`	U+1F1E6–1F1FF		(flag sequence components)
`@`	U+E007F		(the tag sequence terminator)
`@0`–`@9`	U+E0030–E0039		(tag sequence components)
`@a`–`@z`	U+E0061–E007A		(tag sequence components)

✏️ `\begin{coloremojicodeautolist}{`⟨*code value list*⟩`}`

Currently, the following “emoji orders” are defined.

- ♈→♉→♊→♋→♌→♍→♎→♏→♐→♑→♒→♓
- ♠→♥→♦→♣
- 🕐→🕑→🕒→🕓→🕔→🕕→🕖→🕗→🕘→🕙→🕚→🕛
- 0️⃣→1️⃣→2️⃣→3️⃣→4️⃣→5️⃣→6️⃣→7️⃣→8️⃣→9️⃣→🔟

5 “Short Names” for Emojis

For the short names used in code value lists, those declared in the emoji-toolkit library of JoyPixels (formerly known as EmojiOne) are available. In addition, the *special short names* listed in Table 1 can also be used.

Note: For special short names, the `:…:` surrounding the name can be omitted.

Note: For short names other than special short names, the current behavior is that the `:…:` can be omitted if it cannot be interpreted as a hexadecimal notation of an integer, but this may change in the future.

Examples :

```
\coloremojicode{:man: + :woman: + :girl: + :girl:}%Output:👨‍👩‍👧‍👧
\coloremojicode{!flag @g @b @w @l @s @}%Output:🏴
\coloremojicode{1F647 + !male}%Output:🙇‍♂️
```

6 Using emojis in PDF strings

You can use commands for outputting emojis in the document information string (hereafter called *PDF string*) when using the hyperref package. For example, when you include `\coloremoji` in the argument of `\section`, it will be output as an emoji (image) on the page, and will be displayed as text in the PDF bookmark.

However, this is subject to the premise that Unicode characters in PDF strings are properly processed. Namely, the following condition must be satisfied.

Note: Nothing is required for TeX Live 2022 or later (unless you are using (u)pLaTeX).

- The "PDF encoding" of the hyperref package is Unicode.

  This is the default in recent versions of hyperref (version 7.00g and later). In older versions, you need to add the `unicode` option when loading hyperref.

  ```
  \usepackage[unicode]{hyperref}
  ```