



## Font management

Font management

[amusewiki.org](https://amusewiki.org)



# Contents

<b>fontspec.json</b>	<b>5</b>
<b>Custom files for chapter/section headings</b>	<b>7</b>
<b>Languages (version 2.534 2022-02-12)</b>	<b>8</b>
Example for Japanese . . . . .	8

Then add this stanza to `/var/lib/amusewiki/fontspec.json`

```
[
  {
    "desc" : "Source Han Serif (Japanese)",
    "languages" : [
      "ja"
    ],
    "name" : "Source Han Serif",
    "type" : "serif"
  },
  {
    "desc" : "Source Han Sans (Japanese)",
    "languages" : [
      "ja"
    ],
    "name" : "Source Han Sans",
    "type" : "sans"
  },
  // other fonts
]
```

Beware trailing at the end or missing commas between the definitions. You can validate the JSON with, for example, `jq`

```
jq < /var/lib/amusewiki/fontspec.json
```

For Japanese you may want to enable the option in the admin called: *Allow Unicode URIs for categories (author, topics)*. Recommended for non-latin scripts.

A similar setup should be used for sites mixing languages using different scripts. This way you can assign the languages of each fonts.

Restart amusewiki:

```
amusewiki restart
```

# Languages (version 2.534 2022-02-12)

Especially when working with languages which don't use the Latin script, it becomes necessary to configure which fonts should be attached to a given languages. You can do so adding a key to each font family description in the `fontspec.json` file. The key is, unsurprisingly, called `languages` and it needs to be set to an array of language ISO codes.

Example:

```
{
  "desc" : "Source Han Serif (Simplified Chinese)",
  "languages" : [
    "zh"
  ],
  "name" : "Source Han Serif SC",
  "type" : "serif"
},
... // more fonts here
```

Here we added a font definition for Chinese. We can have more of them, but this means that for Chinese we will always use a font defined for that language.

If the font you explicitly chose to use has not the language definition for the language in use, but other font families have it, the first font family with the language set will be used (order of the definitions matters).

## Example for Japanese

Install the Source Han font, available from <https://packages.amusewiki>

```
apt install fonts-source-han-sans-ja fonts-source-han-
serif-ja
```

# fontspec.json

The global font definition file is called `fontspec.json` and resides in the application directory (`/var/lib/amusewiki` on Debian or wherever you installed it). This file can be overridden per site if `fontspec.json` is found in the repo's `site_files` directory, for example `/var/lib/amusewiki/repo/amw/site_files/fontspec.json`.

The default file is generated by the script `muse-create-font-file.pl` which comes with the `Text::Amuse::Compile` module (amusewiki's dependency). By default it prints the content to the standard output, or write it to a file if it's passed as argument.

```
muse-create-font-file.pl fontspec.json
```

The content looks so:

```
[
  {
    "bold" : "/usr/share/fonts/truetype/cmu/cmunbx.ttf",
    "bolditalic" : "/usr/share/fonts/truetype/cmu/cmunbi.ttf",
    "desc" : "CMU Serif",
    "italic" : "/usr/share/fonts/truetype/cmu/cmunti.ttf",
    "name" : "CMU Serif",
    "regular" : "/usr/share/fonts/truetype/cmu/cmunrm.ttf",
    "type" : "serif"
  },
  # more entries
]
```

`muse-create-font-file.pl` uses a default list of font and search them using `fc-list`.

The mandatory parameters are:

**name** The font name as seen by the `fontconfig`. You can get the list of installed fonts using `fc-list`. It's very important that the name actually exists and `fontconfig` sees it, otherwise when trying

to compile a PDF using this font, you will get a failure. Try e.g.  
`fc-list "Linux Libertine 0"`

**desc** This is the description you will see on the Book Builder (/bookbuilder) or in the Custom Formats interface (/settings/formats). It's informative only, so it can be an arbitrary string.

**type** Valid values are `serif`, `sans`, `mono` and should be clear what this means.

To embed the fonts in the EPUBs, you also need the file location, and for this there are the four parameters (all of them must be present):

**regular** The regular font path

**italic** The italic font path

**bold** The bold font path

**bolditalic** The bold italic font path

## Custom files for chapter/section headings

In the bookbuilder and in the format definition, there is a checkbox: "Use sans fonts for section titles". If you want to use a serif font for that as well, you can cheat, edit the `fontspec.json` file and change its `type` from `serif` to `sans`. Sans fonts are allowed as main font body.