

# Package: AddFonts (via r-universe)

May 13, 2026

**Title** Download and Register Fonts from GDPR-Compliant Providers For Use In R graphics

**Version** 0.2.0

**Description** Download and register fonts from GDPR-compliant providers for use in R graphics. Currently supports Bunny Fonts (<https://fonts.bunny.net/>). Architecture allows for easy additions of font providers.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE, packages = ``roxytypes"")

**RoxygenNote** 7.3.3

**Config/Needs/documentation** roxytypes

**SystemRequirements** woff2 (>= 1.0.2): woff2-tools (rpm) or woff2 (deb) or woff2 (arch)

**Imports** sysfonts, httr2, fs, rappdirs, cli, jsonlite, S7, glue, rlang, showtext

**Suggests** testthat (>= 3.3.0), usethis, withr

**Config/testthat/edition** 3

**LazyData** true

**Depends** R (>= 4.5.0)

**URL** <https://codeberg.org/gnoble/AddFonts>,  
<http://guillaume-noblet.com/AddFonts/>,  
<https://gnoble.r-universe.dev/AddFonts>

**BugReports** <https://codeberg.org/gnoble/AddFonts/issues>

**Collate** 'CacheMeta.R' 'CacheEntry.R' 'CacheEntryList.R'  
'FontProvider.R' 'add\_font.R' 'as\_CacheEntryList.R'  
'as\_FontProvider.R' 'as\_list.R' 'assert.R' 'cache.R'  
'cache\_ttf\_path.R' 'cache\_variant\_paths.R' 'conv\_fun.R'  
'download\_and\_cache.R' 'download\_variant\_generic.R'  
'download\_weights.R' 'preview\_font.R' 'register\_from\_cache.R'

'register\_from\_download.R' 'update\_download\_and\_cache.R'  
 'utils.R' 'woff2\_to\_ttf.R' 'zzz.R'

**Config/pak/sysreqs** cmake libfreetype6-dev make libpng-dev libuv1-dev libssl-dev zlib1g-dev

**Repository** <https://gnoblet.r-universe.dev>

**Date/Publication** 2026-02-12 12:32:47 UTC

**RemoteUrl** <https://codeberg.org/gnoblet/AddFonts>

**RemoteRef** HEAD

**RemoteSha** c02003811cbef5193d28687fa1697999e875a74b

## Contents

add_font . . . . .	3
as_CacheEntryList . . . . .	3
as_FontProvider . . . . .	4
as_list . . . . .	4
cache_clean . . . . .	5
cache_get . . . . .	5
cache_get_weights . . . . .	6
cache_read . . . . .	6
cache_remove . . . . .	7
cache_set . . . . .	8
cache_ttf_filename . . . . .	8
cache_ttf_path . . . . .	9
cache_variant_paths . . . . .	9
cache_write . . . . .	10
CacheEntry . . . . .	10
CacheEntryList . . . . .	11
CacheMeta . . . . .	11
conv_fun . . . . .	12
delete_files . . . . .	12
download_and_cache . . . . .	13
download_variant_generic . . . . .	14
download_weights . . . . .	15
FontProvider . . . . .	15
get_cache_dir . . . . .	16
get_provider_details . . . . .	16
preview_font . . . . .	17
register_from_cache . . . . .	18
register_from_download . . . . .	18
safe_id . . . . .	19
update_download_and_cache . . . . .	20
woff2_to_ttf . . . . .	21

**Index**

**22**

---

add_font	<i>Add a font to the local cache and register it for use</i>
----------	--

---

**Description**

Ensure a font is available locally: try the cache first, otherwise download/convert and register the font so it can be used by plotting devices. Returns (invisibly) the list of local file paths.

**Usage**

```
add_font(
  name,
  provider = "bunny",
  family = NULL,
  regular.wt = 400,
  bold.wt = 700,
  subset = "latin"
)
```

**Arguments**

name	(character(1)) Name of the font as known to the provider.
provider	(character(1)) Provider id/name (default: "bunny").
family	(character   NULL) Optional family name to register the font under (default: NULL).
regular.wt	(numeric(1)) Regular weight to request (default: 400).
bold.wt	(numeric(1)) Bold weight to request (default: 700).
subset	(character(1)) Glyph subset to request (default: "latin").

**Value**

(list) Invisibly returns a list with paths for regular, italic, bold and bolditalic variants, or throws an error on failure.

---

as_CacheEntryList	<i>Read from list</i>
-------------------	-----------------------

---

**Description**

Read from list

**Usage**

```
as_CacheEntryList(1)
```

**Arguments**

1 (list) The list to convert to a CacheEntryList.

**Value**

(CacheEntryList) The CacheEntryList object created from the list.

---

as\_FontProvider      *Construct a FontProvider from a named list*

---

**Description**

Construct a FontProvider from a named list

**Usage**

as\_FontProvider(x)

**Arguments**

x (list) Named list (e.g. from JSON) with provider details.

**Value**

(FontProvider) The corresponding FontProvider object.

---

as\_list      *As list*

---

**Description**

As list

**Usage**

as\_list(x)

**Arguments**

x (CacheMeta | CacheEntry | CacheEntryList) The object to convert to a list.

**Value**

(list) The list representation of the CacheEntryList.

---

cache_clean	<i>Clean cache entries</i>
-------------	----------------------------

---

**Description**

Remove entries from the cache, optionally unlinking referenced files.

**Usage**

```
cache_clean(cache_dir = NULL, families = NULL, reset = FALSE, ...)
```

**Arguments**

cache_dir	(NULL   character(1)) Cache directory to use (default: NULL)
families	(character   NULL) Character vector of family names to remove, or NULL to clear the whole cache (default: NULL)
reset	(logical(1)) If TRUE, completely reset and clear the cache (default: FALSE).
...	(anyD) Additional arguments (currently unused).

**Value**

(character | NULL) Invisibly returns character vector of removed family names when deleting specific entries, or NULL when nothing changed. Remove files by default.

**See Also**

Other cache: [cache\\_get\(\)](#), [cache\\_get\\_weights\(\)](#), [cache\\_read\(\)](#), [cache\\_remove\(\)](#), [cache\\_write\(\)](#)

---

cache_get	<i>Get certain families from CacheEntryList</i>
-----------	---

---

**Description**

Get certain families from CacheEntryList

**Usage**

```
cache_get(x, families = NULL, quiet = TRUE)
```

**Arguments**

x	(CacheEntryList) The CacheEntryList object to query.
families	(character vector) The family names to retrieve.
quiet	(logical(1)) If TRUE, suppress informational messages (default: TRUE).

**Value**

(list) A list of CacheEntry objects matching the specified families. If no families

**See Also**

Other cache: [cache\\_clean\(\)](#), [cache\\_get\\_weights\(\)](#), [cache\\_read\(\)](#), [cache\\_remove\(\)](#), [cache\\_write\(\)](#)

---

cache_get_weights	<i>Check which weights are available in a cache entry</i>
-------------------	---

---

**Description**

Check which weights are available in a cache entry

**Usage**

```
cache_get_weights(entry, weights)
```

**Arguments**

entry	(CacheEntry) The CacheEntry object to check.
weights	(numeric) Vector of weights to check for availability.

**Value**

(lgl) Logical vector indicating which weights are cached).

**See Also**

Other cache: [cache\\_clean\(\)](#), [cache\\_get\(\)](#), [cache\\_read\(\)](#), [cache\\_remove\(\)](#), [cache\\_write\(\)](#)

---

cache_read	<i>Read cache entry from disk</i>
------------	-----------------------------------

---

**Description**

Read cache entry from disk

**Usage**

```
cache_read(cache_dir)
```

**Arguments**

cache_dir	(character   NULL) Cache directory to use. Use <a href="#">get_cache_dir()</a> to get the default cache directory.
-----------	--

**Value**

(CacheEntryList) The cache index as a if found and valid.

**See Also**

Other cache: [cache\\_clean\(\)](#), [cache\\_get\(\)](#), [cache\\_get\\_weights\(\)](#), [cache\\_remove\(\)](#), [cache\\_write\(\)](#)

---

cache_remove	<i>Delete entry from cache</i>
--------------	--------------------------------

---

**Description**

Delete entry from cache

**Usage**

```
cache_remove(x, families = NULL, remove_files = TRUE, cache_dir = NULL)
```

**Arguments**

x	(CacheEntryList) The CacheEntryList object to modify.
families	(character   NULL) The font families to delete. If NULL, all entries are deleted.
remove_files	(logical(1)) If TRUE attempt to delete files referenced by removed entries (default: TRUE).
cache_dir	(character(1)   NULL) The cache directory to delete from. If NULL, the default cache directory is used.

**Value**

(CacheEntryList) The modified CacheEntryList with the specified entries removed.

**See Also**

Other cache: [cache\\_clean\(\)](#), [cache\\_get\(\)](#), [cache\\_get\\_weights\(\)](#), [cache\\_read\(\)](#), [cache\\_write\(\)](#)

---

cache_set	<i>Set cache entries</i>
-----------	--------------------------

---

**Description**

Set cache entries

**Usage**

```
cache_set(x, family, meta)
```

**Arguments**

x	(CacheEntryList) The CacheEntryList object to modify.
family	(character(1)) The font family to modify.
meta	(CacheMeta) The new metadata to set.

**Value**

(CacheEntryList) The modified CacheEntryList with the updated entry.

---

cache_ttf_filename	<i>Compose canonical filename for a cached TTF</i>
--------------------	--

---

**Description**

Compose canonical filename for a cached TTF

**Usage**

```
cache_ttf_filename(source, font_id, subset, weight, style)
```

**Arguments**

source	(character(1)) Provider source identifier.
font_id	(character(1)) Font id used to create a filesystem-safe filename.
subset	(character(1)) Glyph subset identifier.
weight	(integer(1)) Font weight.
style	(character(1)) Style string (e.g. "normal", "italic").

**Value**

(character(1)) Filename (not including the cache directory) for the cached TTF.

---

cache_ttf_path	<i>Compute canonical cache path for a TTF file</i>
----------------	--

---

**Description**

Compute canonical cache path for a TTF file

**Usage**

```
cache_ttf_path(source, font_id, subset, weight, style, cache_dir = NULL)
```

**Arguments**

source	(character(1)) Provider source identifier.
font_id	(character(1)) Font id used for filenames.
subset	(character(1)) Glyph subset identifier.
weight	(integer(1)) Font weight.
style	(character(1)) Style string (e.g. "normal", "italic").
cache_dir	(character   NULL) Cache directory to use (default: NULL)

**Value**

(character(1)) Path to the cached . ttf file.

---

cache_variant_paths	<i>Compute paths used for caching provider artifacts and any conversion intermediate files.</i>
---------------------	---

---

**Description**

Compute paths used for caching provider artifacts and any conversion intermediate files.

**Usage**

```
cache_variant_paths(provider, family, weight, style, subset, cache_dir = NULL)
```

**Arguments**

provider	(FontProvider) Provider object with source and optional conversion info.
family	(character(1)) Family identifier.
weight	(integer(1)) Font weight.
style	(character(1)) Style string (e.g. "normal", "italic").
subset	(character(1)) Glyph subset identifier.
cache_dir	(character   NULL) Cache directory to use (default: NULL)

**Value**

(list) A list with elements `to_convert` (path or NULL) and `ttf` (path).

---

<code>cache_write</code>	<i>Write CacheEntryList to disk as JSON</i>
--------------------------	---

---

**Description**

Write CacheEntryList to disk as JSON

**Usage**

```
cache_write(x, cache_dir = NULL, quiet = TRUE)
```

**Arguments**

<code>x</code>	(CacheEntryList) The CacheEntryList object to write to disk.
<code>cache_dir</code>	(character(1)) The cache directory to write to (default: NULL).
<code>quiet</code>	(logical(1)) Whether to suppress output messages (default: TRUE).

**Value**

(NULL) Invisibly returns NULL.

**See Also**

Other cache: [cache\\_clean\(\)](#), [cache\\_get\(\)](#), [cache\\_get\\_weights\(\)](#), [cache\\_read\(\)](#), [cache\\_remove\(\)](#)

---

<code>CacheEntry</code>	<i>S7-backed cache entry (CacheEntry)</i>
-------------------------	---

---

**Description**

S7-backed cache entry (CacheEntry)

**Usage**

```
CacheEntry(family = character(0), meta = CacheMeta())
```

**Arguments**

<code>family</code>	(character(1)) Family name for this cache entry (safe identifier containing only letters, digits, and hyphens).
<code>meta</code>	(CacheMeta) A CacheMeta object describing the cached files and origin for this family.

**Value**

(S7\_object) A validated S7 CacheEntry object.

---

CacheEntryList	<i>S7 list of cache entries (CacheEntryList)</i>
----------------	--

---

**Description**

S7 list of cache entries (CacheEntryList)

**Usage**

```
CacheEntryList(entries = list())
```

**Arguments**

entries (list(1+)) A non-empty list of CacheEntry objects.

**Value**

(S7\_object) A validated S7 CacheEntryList object.

---

CacheMeta	<i>S7-backed cache metadata (CacheMeta)</i>
-----------	---

---

**Description**

S7-backed cache metadata (CacheMeta)

**Usage**

```
CacheMeta(family_id = character(0), source = character(0), files = list())
```

**Arguments**

family_id	(character(1)) A non-empty safe identifier for the font family. Allowed characters: letters, digits and hyphen (no spaces, slashes, dots, colons, tildes, etc.).
source	(character(1)) Name of the provider or source that produced the cached font files.
files	(list(1+)) A non-empty named list of file paths. Names are weight identifiers (e.g., "400" for normal weight 400, "400italic" for italic weight 400). Each element is a character(1) path to the local font files.

**Details**

The added property is a getter-only field that returns the current timestamp as a character string when accessed. It cannot be set during construction.

Weights are not explicitly stored - they are derived from the names of the files list. Registration functions are responsible for selecting appropriate weights for regular/bold variants.

**Value**

(S7\_object) A validated S7 CacheMeta object.

---

conv_fun	<i>Resolve a conversion name to the converter function used by the package (currently only woff2_to_ttf).</i>
----------	---

---

**Description**

Resolve a conversion name to the converter function used by the package (currently only woff2\_to\_ttf).

**Usage**

```
conv_fun(conversion)
```

**Arguments**

conversion (character(1)) Name of the conversion to resolve.

**Value**

(function) The conversion function if known; otherwise the helper aborts.

---

delete_files	<i>Delete files</i>
--------------	---------------------

---

**Description**

Attempt to delete a set of files (character vector or list of paths).

**Usage**

```
delete_files(entries, quiet = c("full", "success", "fail", "none"))
```

**Arguments**

entries (character) Character vector of file paths to remove.

quiet (full | "success" | "fail" | "none") If "full", capture all output and suppress console messages; if "success", only show success messages; if "fail", only show error messages; if "none", show all messages (default: "none").

**Value**

(list) A list with the following elements:

- `deleted: character()` — paths successfully deleted
- `failed: character()` — paths that existed but could not be deleted
- `not_found: character()` — paths that were not found on disk

---

`download_and_cache`      *Download font variants and add to cache*

---

**Description**

Downloads font files for requested weights, creates a cache entry, and writes to cache. Does NOT register the font - caller should use `register_from_cache()` for that.

**Usage**

```
download_and_cache(
  provider,
  name,
  font_id,
  family_name,
  regular.wt = 400,
  bold.wt = 700,
  subset = "latin",
  cache_dir = NULL
)
```

**Arguments**

<code>provider</code>	(FontProvider) Provider object used for downloads.
<code>name</code>	(character(1)) Font name at the provider.
<code>font_id</code>	(character(1)) Filesystem-safe font id.
<code>family_name</code>	(character(1)) Family name for the font.
<code>regular.wt</code>	(integer(1)) Regular weight to fetch (default: 400)
<code>bold.wt</code>	(integer(1)) Bold weight to fetch (default: 700)
<code>subset</code>	(character(1)) Glyph subset to request (default: "latin")
<code>cache_dir</code>	(character   NULL) Cache directory to use (default: NULL)

**Value**

(CacheEntry | NULL) Cache entry with downloaded fonts, or NULL on failure.

---

download\_variant\_generic

*Download and (if needed) convert a provider artifact to a local TTF file for a given family/weight/style and return the local path.*

---

## Description

Download and (if needed) convert a provider artifact to a local TTF file for a given family/weight/style and return the local path.

## Usage

```
download_variant_generic(  
    provider,  
    family,  
    weight,  
    style,  
    subset = "latin",  
    cache_dir = NULL,  
    quiet = FALSE  
)
```

## Arguments

provider	(FontProvider) Provider object with url_template and source.
family	(character(1)) Family identifier.
weight	(integer(1)) Font weight to fetch (100-900).
style	(character(1)) Style (e.g. "normal", "italic").
subset	(character(1)) Glyph subset to request (default: "latin")
cache_dir	(character   NULL) Cache directory to use (default: NULL)
quiet	(logical(1)) Suppress warnings/messages (default: FALSE)

## Value

(character | NULL) Path to the local .ttf file on success, or NULL on failure.

---

download_weights	<i>Download font files for specified weights</i>
------------------	--

---

**Description**

Downloads normal and italic variants for each weight and returns a named list with weight-based keys.

**Usage**

```
download_weights(provider, name, weights, subset, cache_dir, quiet = TRUE)
```

**Arguments**

provider	(FontProvider) Provider object used for downloads.
name	(character(1)) Font name at the provider.
weights	(numeric) Vector of weights to download.
subset	(character(1)) Glyph subset to request.
cache_dir	(character(1)) Cache directory to use.
quiet	(logical(1)) Whether to suppress download messages (default: TRUE).

**Value**

(list) Named list where names are weight identifiers (e.g., "400", "700italic") and values are file paths.

---

FontProvider	<i>Font provider specification (FontProvider)</i>
--------------	---

---

**Description**

Font provider specification (FontProvider)

**Usage**

```
FontProvider(
  source = character(0),
  url_template = character(0),
  conversion = NULL,
  conversion_ext = NULL,
  aliases = list()
)
```

**Arguments**

source	(character(1)) Provider id/name (e.g. "bunny").
url_template	(character(1)) URL template used to construct download URLs.
conversion	(character(1)   NULL) Optional conversion function name (as string) or NULL.
conversion_ext	(character(1)   NULL) Original extension handled by the provider (e.g. "woff2").
aliases	(list   NULL) Optional list of alias names to match (e.g. "fonts.bunny.net").

**Value**

(FontProviders) S7 class representing a font provider specification.

---

get_cache_dir	<i>Get the package cache directory for fonts</i>
---------------	--

---

**Description**

Determine a platform-appropriate cache directory for AddFonts and ensure it exists, creating it when necessary.

**Usage**

```
get_cache_dir()
```

**Value**

(character(1)) Absolute path to the cache directory.

---

get_provider_details	<i>Get provider details from internal data</i>
----------------------	--

---

**Description**

Load and return a FontProvider object for the specified provider. The providers data is stored in the package's internal sysdata.rda.

**Usage**

```
get_provider_details(provider)
```

**Arguments**

provider	(character(1)) Provider id/name (e.g. "bunny").
----------	---

**Value**

(FontProvider) A validated FontProvider object.

---

`preview_font`*Preview a font by ensuring it's installed and drawing a sample string*

---

## Description

Ensure the requested font is installed via `add_font()` and draw a brief sample using `showtext` for proper font rendering.

## Usage

```
preview_font(  
  name,  
  provider = "bunny",  
  family = NULL,  
  sample = "The quick brown fox jumps over the lazy dog",  
  size = 28,  
  subset = "latin",  
  regular.wt = 400,  
  bold.wt = 700  
)
```

## Arguments

<code>name</code>	(character(1)) Font name as used by the provider (e.g. "oswald").
<code>provider</code>	(character(1)) Provider name to use (default: "bunny")
<code>family</code>	(character   NULL) Optional family name to register the font under (default: NULL)
<code>sample</code>	(character(1)) Sample text to display (default: "The quick brown fox jumps over the lazy dog")
<code>size</code>	(numeric(1)) Font size in points for the preview (default: 28)
<code>subset</code>	(character(1)) Glyph subset to request (default: "latin")
<code>regular.wt</code>	(integer(1)) Regular weight to display (default: 400)
<code>bold.wt</code>	(integer(1)) Bold weight to display (default: 700)

## Value

(list) Invisibly returns the list of paths produced by `add_font()`.

---

register_from_cache	<i>Validate a cache entry and register the font with sysfonts if the required files exist. Returns the prepared files list or NULL when registration cannot proceed.</i>
---------------------	--

---

### Description

This is the **ONLY** function that calls `sysfonts::font_add()`. It does not print success messages - callers should handle user feedback.

### Usage

```
register_from_cache(entry, regular.wt = 400, bold.wt = 700)
```

### Arguments

entry	(CacheEntry) Cache entry object with family and metadata.
regular.wt	(numeric(1)) Regular weight to use for regular and italic variants (default: 400).
bold.wt	(numeric(1)) Bold weight to use for bold and bolditalic variants (default: 700).

### Value

(list | NULL) Prepared files list (with regular, italic, bold, bolditalic) or NULL.

---

register_from_download	<i>Download necessary variants for a font, write a cache entry and register the font with sysfonts. Returns the prepared files list on success, or NULL if a regular font could not be obtained.</i>
------------------------	--

---

### Description

This function downloads fonts, creates a cache entry, and then calls `register_from_cache()` to perform the actual registration with sysfonts.

### Usage

```
register_from_download(
  provider,
  name,
  font_id,
  family_name,
  regular.wt = 400,
  bold.wt = 700,
```

```

    subset = "latin",
    cache_dir = NULL
)

```

### Arguments

provider	(FontProvider) Provider object used for downloads.
name	(character(1)) Font name at the provider.
font_id	(character(1)) Filesystem-safe font id.
family_name	(character(1)) Family name to register the font under.
regular.wt	(integer(1)) Regular weight to fetch (default: 400)
bold.wt	(integer(1)) Bold weight to fetch (default: 700)
subset	(character(1)) Glyph subset to request (default: "latin")
cache_dir	(character   NULL) Cache directory to use (default: NULL)

### Value

(list | NULL) List of local file paths for variants, or NULL on failure.

---

safe_id	<i>Create a filesystem-safe id from a name</i>
---------	--

---

### Description

Replace disallowed characters with - and convert to lower-case so the resulting id is safe to use in filenames.

### Usage

```
safe_id(name)
```

### Arguments

name	(character(1)) Input name to sanitise.
------	--

### Value

(character(1)) Sanitised identifier.

---

`update_download_and_cache`*Download missing weights and update an existing cache entry*

---

### Description

Downloads missing font weights and updates the cache entry. Does NOT register the font - use `register_from_cache()` after this.

### Usage

```
update_download_and_cache(  
    entry,  
    provider,  
    name,  
    family_name,  
    missing_weights,  
    subset = "latin",  
    cache_dir = NULL,  
    cel = NULL  
)
```

### Arguments

<code>entry</code>	(CacheEntry) Existing cache entry to update.
<code>provider</code>	(FontProvider) Provider object used for downloads.
<code>name</code>	(character(1)) Font name at the provider.
<code>family_name</code>	(character(1)) Family name for the font.
<code>missing_weights</code>	(numeric) Vector of weights to download and add to cache.
<code>subset</code>	(character(1)) Glyph subset to request (default: "latin").
<code>cache_dir</code>	(character   NULL) Cache directory to use.
<code>cel</code>	(CacheEntryList) Current cache entry list to update.

### Value

(CacheEntry | NULL) Updated cache entry with new weights (NOT registered), or NULL on failure.

---

woff2_to_ttf	<i>Convert a .woff2 font to .ttf using the system 'woff2_decompress' tool</i>
--------------	---

---

**Description**

Internal helper. Uses the system `woff2_decompress` tool to convert a `.woff2` file into a `.ttf` file. Documented with roxytypes annotations for clarity.

**Usage**

```
woff2_to_ttf(font_file, overwrite = FALSE, remove_old = TRUE, quiet = "fail")
```

**Arguments**

<code>font_file</code>	( <code>character(1)</code> ) Path to the <code>.woff2</code> file to convert.
<code>overwrite</code>	( <code>logical(1)</code> ) If <code>TRUE</code> , overwrite an existing <code>.ttf</code> conversion.
<code>remove_old</code>	( <code>logical(1)</code> ) If <code>TRUE</code> , remove the original <code>.woff2</code> file after conversion try.
<code>quiet</code>	( <code>full</code>   <code>"success"</code>   <code>"fail"</code>   <code>"none"</code> ) If <code>"full"</code> , capture all output and suppress console messages; if <code>"success"</code> , only show success messages; if <code>"fail"</code> , only show error messages; if <code>"none"</code> , show all messages (default: <code>"none"</code> ).

**Value**

(`character(1)`) Invisibly returns the path to the `.ttf` file on success.

# Index

## \* **cache**

- cache\_clean, [5](#)
- cache\_get, [5](#)
- cache\_get\_weights, [6](#)
- cache\_read, [6](#)
- cache\_remove, [7](#)
- cache\_write, [10](#)

add\_font, [3](#)

as\_CacheEntryList, [3](#)

as\_FontProvider, [4](#)

as\_list, [4](#)

cache\_clean, [5](#), [6](#), [7](#), [10](#)

cache\_get, [5](#), [5-7](#), [10](#)

cache\_get\_weights, [5](#), [6](#), [6](#), [7](#), [10](#)

cache\_read, [5](#), [6](#), [6](#), [7](#), [10](#)

cache\_remove, [5](#), [6](#), [7](#), [7](#), [10](#)

cache\_set, [8](#)

cache\_ttf\_filename, [8](#)

cache\_ttf\_path, [9](#)

cache\_variant\_paths, [9](#)

cache\_write, [5-7](#), [10](#)

CacheEntry, [10](#)

CacheEntryList, [11](#)

CacheMeta, [11](#)

conv\_fun, [12](#)

delete\_files, [12](#)

download\_and\_cache, [13](#)

download\_variant\_generic, [14](#)

download\_weights, [15](#)

FontProvider, [15](#)

get\_cache\_dir, [16](#)

get\_cache\_dir(), [6](#)

get\_provider\_details, [16](#)

preview\_font, [17](#)

register\_from\_cache, [18](#)

register\_from\_download, [18](#)

safe\_id, [19](#)

update\_download\_and\_cache, [20](#)

woff2\_to\_ttf, [21](#)