

Package: AddFonts (via r-universe)

October 7, 2026

Title Download and Register Fonts from GDPR-Compliant Providers for Use in R Graphics

Version 1.1.0

Description Download and register fonts from GDPR-compliant providers for use in R graphics. Supports weight-based providers (e.g. Bunny Fonts, <<https://fonts.bunny.net/>>) and file-based providers (e.g. Bye Bye Binary, <<https://gitlab.com/bye-bye-binary>>). Architecture allows for easy additions of font providers.

License GPL (>= 3)

Encoding UTF-8

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

Config/Needs/documentation roxytypes

SystemRequirements woff2 (>= 1.0.2, optional): needed only for weight-based providers that serve woff2 (e.g. Bunny Fonts).
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 (>= 4.1.0)

URL <https://codeberg.org/gnoble/AddFonts>,
<https://gnoble.r-universe.dev/AddFonts>

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

Collate 'CacheMeta.R' 'CacheEntry.R' 'CacheEntryList.R'
'FontProvider.R' 'FontProviderFile.R' 'FontProviderWeight.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_and_cache_file.R' 'download_variant_file.R'
'download_variant_generic.R' 'download_weights.R'
'first_use_message.R' 'helpers.R' 'preview_font.R'
'provider_registry.R' 'register_from_cache.R'
'update_download_and_cache.R' 'utils.R' 'woff2_to_ttf.R'
'zzz.R'
```

Config/roxygen2/version 8.0.0

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

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

Date/Publication 2026-08-08 06:31:49 UTC

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

RemoteRef HEAD

RemoteSha 88e2f4bc37fe88a4864418bc52da462c743c4c8f

Contents

.add_font_direct_url	3
.add_font_file	4
.add_font_local	4
.add_font_weight	5
.fetch_url_to_cache	6
.persist_cache_entry	6
.validate_variants	7
add_font	8
as_CacheEntryList	9
as_FontProvider	9
as_list	10
cache_clean	10
cache_file_path	11
cache_get	12
cache_get_variants	12
cache_get_weights	13
cache_read	14
cache_remove	14
cache_set	15
cache_ttf_filename	16
cache_ttf_path	16
cache_variant_paths	17
cache_write	17
CacheEntry	18
CacheEntryList	19
CacheMeta	19
conv_fun	20
copy_and_cache_local	20
copy_variant_to_cache	21
delete_files	21

<code>download_and_cache</code>	22
<code>download_and_cache_file</code>	23
<code>download_and_cache_url</code>	24
<code>download_variant_file</code>	24
<code>download_variant_generic</code>	25
<code>download_weights</code>	26
<code>FontProvider</code>	26
<code>FontProviderDirectURL</code>	27
<code>FontProviderFile</code>	27
<code>FontProviderLocal</code>	28
<code>FontProviderWeight</code>	28
<code>get_cache_dir</code>	29
<code>get_provider_details</code>	30
<code>list_providers</code>	30
<code>maybe_show_first_use</code>	31
<code>preview_font</code>	31
<code>register_from_cache</code>	32
<code>register_provider</code>	33
<code>safe_id</code>	33
<code>unregister_provider</code>	34
<code>update_download_and_cache</code>	34
<code>woff2_to_ttf</code>	35

Index 36

`.add_font_direct_url` *Route `add_font()` for provider = "url" (direct download)*

Description

Handles the cache-check → download → register cycle when the user supplies provider = "url".
Uses source key "url" in the cache index.

Usage

`.add_font_direct_url(name, family_name, variants, cache_dir)`

Arguments

<code>name</code>	(character(1)) Font name (used as the family component of cache filenames).
<code>family_name</code>	(character(1)) Family name to register the font under.
<code>variants</code>	(list) Named list mapping symbolic variant keys to full download URLs.
<code>cache_dir</code>	(character(1)) Cache directory path.

Value

(list) Invisibly, a named list of local file paths for all registered variants.

<code>.add_font_file</code>	<i>Route <code>add_font()</code> for a file-based provider</i>
-----------------------------	--

Description

Handles the cache-check → download → register cycle for `FontProviderFile` providers using symbolic variant keys.

Usage

```
.add_font_file(provider_obj, name, family_name, variants, cache_dir)
```

Arguments

<code>provider_obj</code>	(<code>FontProviderFile</code>) File-based provider object.
<code>name</code>	(<code>character(1)</code>) Font name at the provider (used as {family} in the URL template).
<code>family_name</code>	(<code>character(1)</code>) Family name to register the font under.
<code>variants</code>	(<code>list</code>) Named list mapping symbolic variant keys to filename stems.
<code>cache_dir</code>	(<code>character(1)</code>) Cache directory path.

Value

(`list`) Invisibly, a named list of local file paths for all registered variants.

<code>.add_font_local</code>	<i>Route <code>add_font()</code> for provider = "file" (local copies)</i>
------------------------------	---

Description

Handles the cache-check → copy → register cycle when the user supplies `provider = "file"`. Uses source key "file" in the cache index.

Usage

```
.add_font_local(name, family_name, variants, cache_dir)
```

Arguments

<code>name</code>	(<code>character(1)</code>) Font name (used as the family component of cache filenames).
<code>family_name</code>	(<code>character(1)</code>) Family name to register the font under.
<code>variants</code>	(<code>list</code>) Named list mapping symbolic variant keys to absolute local file paths.
<code>cache_dir</code>	(<code>character(1)</code>) Cache directory path.

Value

(list) Invisibly, a named list of local file paths for all registered variants.

.add_font_weight	<i>Route add_font() for a weight-based provider</i>
------------------	---

Description

Handles the full cache-check → optional partial update-download-register cycle for FontProviderWeight providers.

Usage

```
.add_font_weight(  
  provider_obj,  
  name,  
  family_name,  
  regular.wt,  
  bold.wt,  
  subset,  
  cache_dir  
)
```

Arguments

provider_obj	(FontProviderWeight) Weight-based provider object.
name	(character(1)) Font name at the provider.
family_name	(character(1)) Family name to register the font under.
regular.wt	(numeric(1)) Regular weight to request.
bold.wt	(numeric(1)) Bold weight to request.
subset	(character(1)) Glyph subset to request.
cache_dir	(character(1)) Cache directory path.

Value

(list) Invisibly, a named list of local file paths for all registered variants.

`.fetch_url_to_cache` *Download a URL to a local file via httr2*

Description

Issues an HTTP GET for `url`, writing the response body to `local_path`. On failure (httr2 error or missing output file) warns (unless `quiet`) and returns `NULL`.

Usage

```
.fetch_url_to_cache(url, local_path, family, variant, quiet)
```

Arguments

<code>url</code>	(<code>character(1)</code>) Full URL to fetch.
<code>local_path</code>	(<code>character(1)</code>) Destination path for the downloaded file.
<code>family</code>	(<code>character(1)</code>) Font family name — used in warning messages only.
<code>variant</code>	(<code>character(1)</code>) Variant key — used in warning messages only.
<code>quiet</code>	(<code>logical(1)</code>) Suppress warnings and messages.

Value

(`character` | `NULL`) `local_path` on success, or `NULL` on failure.

`.persist_cache_entry` *Build a CacheEntry, persist the cache index, and return the entry*

Description

Constructs a [CacheMeta\(\)](#) and a [CacheEntry\(\)](#), upserts the entry into the on-disk cache index via [cache_read_safe\(\)](#), [cache_set\(\)](#), and [cache_write\(\)](#), then returns the new entry. This is the shared persist tail used by all download/copy orchestrators.

Usage

```
.persist_cache_entry(
  source,
  family_name,
  files_entry,
  cache_dir,
  failed_keys = character(0)
)
```

Arguments

<code>source</code>	(character(1)) Provider source identifier (e.g. "bunny", "file", "url").
<code>family_name</code>	(character(1)) Family identifier to register the entry under.
<code>files_entry</code>	(list) Named list of variant-key to local file path mappings.
<code>cache_dir</code>	(character(1)) Path to the cache directory.
<code>failed_keys</code>	(character(0+)) A character vector of keys that were requested but failed to download. Empty if all requested keys were successfully downloaded. (default: character(0))

Value

(CacheEntry) The newly created cache entry.

<code>.validate_variants</code>	<i>Validate a variants list</i>
---------------------------------	---------------------------------

Description

Checks that `variants` is a non-NULL named list whose names are a subset of the recognised symbolic keys and contains at least "regular". Aborts with an informative error on the first violation found.

Usage

```
.validate_variants(variants)
```

Arguments

<code>variants</code>	(list) Named list of symbolic variant keys to font-specific values (filename stems, absolute paths, or URLs depending on the caller).
-----------------------	---

Value

(NULL) Returns `invisible(NULL)` on success; called for its side-effect.

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,  
  variants = 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) FontProvider) Provider id/name (default: "bunny"), or a Font-Provider object constructed with FontProviderWeight() or FontProviderFile() (bypasses the registry lookup).
family	(character NULL) Optional family name to register the font under (default: NULL).
variants	(list NULL) For file-based providers only. Named list mapping symbolic variant keys ("regular", "italic", "bold", "bolditalic") to filename stems served by the provider (without extension). Must include at least "regular". Ignored for weight-based providers (default: NULL).
regular.wt	(numeric(1)) For weight-based providers. Regular weight to request (default: 400).
bold.wt	(numeric(1)) For weight-based providers. Bold weight to request (default: 700).
subset	(character(1)) For weight-based providers. Glyph subset to request (default: "latin").

Details

For weight-based providers (e.g. Bunny Fonts), supply regular.wt,bold.wt, and subset. For file-based providers (e.g. Bye Bye Binary), supply variants instead.

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(l)

## S7 method for class <list>
as_CacheEntryList(l)
```

Arguments

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

Value

(CacheEntryList) The CacheEntryList object created from the list.

as_FontProvider	<i>Construct a FontProvider subclass from a named list</i>
-----------------	--

Description

Reads the type field ("weight" or "file") and constructs the appropriate subclass. Missing type defaults to "weight" for backward compatibility with existing providers.json entries.

Usage

```
as_FontProvider(x)

## S7 method for class <list>
as_FontProvider(x)
```

Arguments

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

Value

(FontProviderWeight | FontProviderFile) The corresponding provider object.

as_list

As list

Description

As list

Usage

```
as_list(x)
```

```
## S7 method for class <AddFonts::CacheMeta>
```

```
as_list(x)
```

```
## S7 method for class <AddFonts::CacheEntry>
```

```
as_list(x)
```

```
## S7 method for class <AddFonts::CacheEntryList>
```

```
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

Clean cache entries

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).

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_variants\(\)](#), [cache_get_weights\(\)](#), [cache_read\(\)](#), [cache_remove\(\)](#), [cache_write\(\)](#)

cache_file_path	<i>Compute canonical cache path for a file-based (symbolic-variant) font file</i>
-----------------	---

Description

Used by file-based providers (e.g. Bye Bye Binary) where each variant is identified by a symbolic key ("regular", "italic", "bold", "bolditalic") rather than a numeric weight.

Usage

cache_file_path(source, family, variant, file_ext, cache_dir = NULL)

Arguments

- source (character(1)) Provider source identifier.
- family (character(1)) Family name (will be made filesystem-safe via safe_id()).
- variant (character(1)) Symbolic variant key: one of "regular", "italic", "bold", "bolditalic".
- file_ext (character(1)) File extension of the cached font (e.g. "ttf", "otf").
- cache_dir (character | NULL) Cache directory. Defaults to get_cache_dir() when NULL.

Value

(character(1)) Full path to the locally cached font file.

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

Description

Get certain families from CacheEntryList

Usage

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

```
## S7 method for class <AddFonts::CacheEntryList>
cache_get(x, families = NULL, source = NULL, quiet = TRUE)
```

Arguments

x	(CacheEntryList) The CacheEntryList object to query.
families	(character vector) The family names to retrieve.
source	(character(1) NULL) If provided, look up by exact compound "{source}::{family}" key (fast). If NULL, scan all entries and match on family name alone (default: NULL).
quiet	(logical(1)) If TRUE, suppress informational messages (default: TRUE).

Value

(list) A list of CacheEntry objects matching the specified families, or NULL.

See Also

Other cache: [cache_clean\(\)](#), [cache_get_variants\(\)](#), [cache_get_weights\(\)](#), [cache_read\(\)](#), [cache_remove\(\)](#), [cache_write\(\)](#)

cache_get_variants	<i>Check which symbolic variant keys are present in a CacheEntry</i>
--------------------	--

Description

Used for file-based providers whose CacheMeta@files uses the key "regular", "italic", "bold", "bolditalic" instead of numeric weight strings.

Usage

```
cache_get_variants(entry, variants)
```

```
## S7 method for class <AddFonts::CacheEntry>
cache_get_variants(entry, variants)
```

Arguments

entry (CacheEntry) The cache entry to inspect.
 variants (character) Character vector of symbolic variant names to check.

Value

(lgl) Named logical vector indicating which variants are cached.

See Also

Other cache: [cache_clean\(\)](#), [cache_get\(\)](#), [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)

## S7 method for class <AddFonts::CacheEntry>
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_get_variants\(\)](#), [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)

## S7 method for class <character>
cache_read(cache_dir)
```

Arguments

cache_dir (character(1)) Cache directory path. Must not be NULL. Use cache_read_safe() for a NULL-tolerant variant that returns an empty index on error.

Value

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

See Also

Other cache: [cache_clean\(\)](#), [cache_get\(\)](#), [cache_get_variants\(\)](#), [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,
  source = NULL,
  remove_files = TRUE,
  cache_dir = NULL
)

## S7 method for class <AddFonts::CacheEntryList>
```

```
cache_remove(  
  x,  
  families = NULL,  
  source = 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.
- source (character(1) | NULL) If provided, remove only the entry for "{source}::{family}". If NULL, remove all entries whose family name matches, regardless of source.
- 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_variants\(\)](#), [cache_get_weights\(\)](#), [cache_read\(\)](#), [cache_write\(\)](#)

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

Description

Set cache entries

Usage

```
cache_set(x, family, meta)  
  
## S7 method for class <AddFonts::CacheEntryList>  
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).

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

Description

Write CacheEntryList to disk as JSON

Usage

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

## S7 method for class <AddFonts::CacheEntryList>
cache_write(x, cache_dir = NULL, quiet = TRUE)
```

Arguments

`x` (CacheEntryList) The CacheEntryList object to write to disk.

`cache_dir` (character(1)) The cache directory to write to (default: NULL).

`quiet` (logical(1)) Whether to suppress output messages (default: TRUE).

Value

(NULL) Invisibly returns NULL.

See Also

Other cache: [cache_clean\(\)](#), [cache_get\(\)](#), [cache_get_variants\(\)](#), [cache_get_weights\(\)](#), [cache_read\(\)](#), [cache_remove\(\)](#)

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

Description

S7-backed cache entry (CacheEntry)

Usage

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

Arguments

`family` (character(1)) Family display/registration name for this cache entry - the exact string passed to `sysfonts::font_add()` and later used by callers as the family in plotting code. Any non-empty string, including spaces (e.g. "Roboto Condensed"). Not restricted to filesystem-safe characters: cache filenames are sanitised separately via `safe_id()`.

`meta` (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(source, files, key_scheme = NULL, failed_keys = character(0))
```

Arguments

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 must follow the scheme declared by key_scheme. (default: NULL)
key_scheme	(character(1)) Key scheme used in files: "weight" for numeric weight keys (e.g. "400", "700italic") or "symbolic" for variant keys ("regular", "bold", etc.).
failed_keys	(character(0+)) A character vector of keys that were requested but failed to download. Empty if all requested keys were successfully downloaded. (default: character(0))

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.

copy_and_cache_local	<i>Copy local font files to cache and create a cache entry</i>
----------------------	--

Description

Copies each local font file into the cache directory, creates a [CacheEntry\(\)](#) with symbolic keys and source "file", writes the entry to the cache, and returns it.

Usage

```
copy_and_cache_local(name, family_name, variants, cache_dir = NULL)
```

Arguments

name	(character(1)) Font name (used as the family component of cache filenames).
family_name	(character(1)) Family name under which to register the font.
variants	(list) Named list mapping symbolic variant keys to absolute file paths. Names must be a subset of c("regular", "italic", "bold", "bolditalic"). At minimum, "regular" must be present.
cache_dir	(character NULL) Cache directory to use (default: NULL).

Value

(CacheEntry | NULL) Cache entry with copied variants, or NULL if the regular variant could not be copied.

copy_variant_to_cache	<i>Copy one local font file into the cache</i>
-----------------------	--

Description

Copies a local font file into the AddFonts cache directory, naming it according to the standard "file-{family}-{variant}.{ext}" convention.

Usage

```
copy_variant_to_cache(  
  src_path,  
  family,  
  variant,  
  cache_dir = NULL,  
  quiet = FALSE  
)
```

Arguments

- src_path (character(1)) Absolute path to the source font file.
- family (character(1)) Family identifier used in the cache filename.
- variant (character(1)) Symbolic variant key: one of "regular", "italic", "bold", "bolditalic".
- cache_dir (character | NULL) Cache directory. Defaults to get_cache_dir() when NULL.
- quiet (logical(1)) Suppress warnings/messages (default: FALSE).

Value

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

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 = FALSE)
```

Arguments

entries	(character) Character vector of file paths to remove.
quiet	(logical(1)) If TRUE, suppress all console messages. If FALSE, show success and failure messages (default: FALSE).

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	<i>Download font variants and add to cache</i>
--------------------	--

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,
  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.
family_name	(character(1)) Family name for the font.
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

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

download_and_cache_file	<i>Download all variants of a file-based font and add to cache</i>
-------------------------	--

Description

Downloads each named variant from a `FontProviderFile` provider, creates a `CacheEntry()` with symbolic keys ("regular", "italic", "bold", "bolditalic"), writes the entry to the cache, and returns it.

Usage

```
download_and_cache_file(  
    provider,  
    name,  
    family_name,  
    variants,  
    cache_dir = NULL  
)
```

Arguments

provider	(FontProviderFile) A file-based provider object.
name	(character(1)) Font name as known to the provider (used in URL template as {family}).
family_name	(character(1)) Family name under which to register the font.
variants	(list) Named list mapping symbolic variant keys to filename stems. Names must be a subset of c("regular", "italic", "bold", "bolditalic"). At minimum, "regular" must be present. Values are filename stems without extension (e.g. list(regular = "Alpaga-Regular", bold = "Alpaga-Bold")).
cache_dir	(character NULL) Cache directory to use (default: NULL).

Value

(CacheEntry | NULL) Cache entry with downloaded variants, or NULL if the regular variant could not be downloaded.

download_and_cache_url

Download direct-URL font variants and add to cache

Description

Downloads each variant from a direct URL, creates a `CacheEntry()` with symbolic keys and source "url", writes the entry to the cache, and returns it.

Usage

```
download_and_cache_url(name, family_name, variants, cache_dir = NULL)
```

Arguments

name	(character(1)) Font name (used as the family component of cache filenames).
family_name	(character(1)) Family name under which to register the font.
variants	(list) Named list mapping symbolic variant keys to full URLs. Names must be a subset of c("regular", "italic", "bold", "bolditalic"). At minimum, "regular" must be present.
cache_dir	(character NULL) Cache directory to use (default: NULL).

Value

(CacheEntry | NULL) Cache entry with downloaded variants, or NULL if the regular variant could not be downloaded.

download_variant_file *Download one font file from a file-based provider*

Description

Downloads a single font variant directly from a `FontProviderFile` using its `base_url` template. No conversion is performed — the file is stored as received.

Usage

```
download_variant_file(
  provider,
  family,
  filename,
  variant,
  cache_dir = NULL,
  quiet = FALSE
)
```

Arguments

provider	(FontProviderFile) A file-based provider object.
family	(character(1)) Family identifier used in the URL template and cache filename.
filename	(character(1)) Filename stem (without extension) for the specific variant (e.g. "Alpaga-Regular"). Substituted into the {filename} placeholder of provider@base_url.
variant	(character(1)) Symbolic key for this variant: one of "regular", "italic", "bold", "bolditalic".
cache_dir	(character NULL) Cache directory to use (default: NULL).
quiet	(logical(1)) Suppress warnings/messages (default: FALSE).

Value

(character | NULL) Path to the locally cached font file on success, 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>Base font provider class (FontProvider)</i>
--------------	--

Description

Abstract base class shared by all provider types. Do not construct this directly. Use FontProviderWeight() or FontProviderFile() instead.

Usage

```
FontProvider(  
  source = character(0),  
  aliases = list(),  
  first_use_message = character(0),  
  first_use_url = character(0)  
)
```

Arguments

source (character(1)) Provider id/name (e.g. "bunny", "bbb").
aliases (list) Optional list of alias strings to recognise the provider by (e.g. list("fonts.bunny.net")).
first_use_message (character(1) | NULL) Optional message displayed once per R session the first time this provider is used (e.g. a licensing notice). NULL means no message.
first_use_url (character(1) | NULL) Optional URL shown alongside first_use_message. NULL means no URL.

Value

(FontProvider) S7 base class. Use a subclass constructor in practice.

FontProviderDirectURL *Sentinel provider for direct-URL font downloads (provider = "url")*

Description

Returned internally when add_font(provider = "url") is used. Carries no extra properties — its type alone signals the direct-URL dispatch path.

Usage

FontProviderDirectURL()

FontProviderFile *File-based font provider (FontProviderFile)*

Description

Provider that downloads font files directly by filename from a base URL, with no weight/style/subset parameterisation. Covers Git-hosted collections like Bye Bye Binary where each variant has a fixed filename.

Usage

```
FontProviderFile(
  source,
  base_url,
  file_ext = "ttf",
  aliases = list(),
  first_use_message = NULL,
  first_use_url = NULL
)
```

Arguments

source	(character(1)) Provider id/name (e.g. "bbb").
base_url	(character(1)) Glue-style URL template. Must contain {family} and {filename} placeholders (e.g. "https://gitlab.com/bye-bye-binary/{family}/-/raw/main/ttf/{filename}"; {family} is substituted as a lower-cased, hyphenated slug; {filename} is substituted verbatim (it is an exact, case-sensitive server-side stem, not a display name).
file_ext	(character(1)) Extension of the font files served by this provider (default: "ttf"). Must be "ttf" or "otf" (no conversion is performed).
aliases	(list) Optional alias strings (inherited from FontProvider()).
first_use_message	(character(1) NULL) Inherited from FontProvider().
first_use_url	(character(1) NULL) Inherited from FontProvider().

Value

(FontProviderFile) A validated S7 FontProviderFile object.

FontProviderLocal	<i>Sentinel provider for local font files (provider = "file")</i>
-------------------	---

Description

Returned internally when add_font(provider = "file") is used. Carries no extra properties — its type alone signals the local-copy dispatch path.

Usage

FontProviderLocal()

FontProviderWeight	<i>Weight-based font provider (FontProviderWeight)</i>
--------------------	--

Description

Provider that resolves font variants by numeric weight and style via a glue-style URL template. This covers APIs like Bunny Fonts.

Usage

```
FontProviderWeight(  
  source,  
  url_template,  
  conversion = NULL,  
  conversion_ext = NULL,  
  aliases = list(),  
  first_use_message = NULL,  
  first_use_url = NULL  
)
```

Arguments

- source (character(1)) Provider id/name (e.g. "bunny").
- url_template (character(1)) Glue-style URL template. Must contain {family}. Typically also uses {subset}, {weight} (integer), and {style}. {family} is substituted as a lower-cased, hyphenated slug (e.g. "Roboto Condensed" becomes "roboto-condensed"); the other placeholders are substituted verbatim.
- conversion (character(1) | NULL) Name of a conversion function to apply after download (e.g. "woff2_to_ttf"), or NULL if the provider serves TTF directly.
- conversion_ext (character(1) | NULL) File extension of the downloaded artifact before conversion (e.g. "woff2"), or NULL.
- aliases (list) Optional alias strings (inherited from [FontProvider\(\)](#)).
- first_use_message (character(1) | NULL) Inherited from [FontProvider\(\)](#).
- first_use_url (character(1) | NULL) Inherited from [FontProvider\(\)](#).

Value

(FontProviderWeight) A validated S7 FontProviderWeight object.

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.

list_providers	<i>List all available font providers</i>
----------------	--

Description

Returns a named list of all FontProvider objects: built-in providers first, then any providers registered in the current session. Session providers with the same source name as a built-in take precedence.

Usage

```
list_providers()
```

Value

(list) Named list of FontProvider objects keyed by their source name.

maybe_show_first_use	<i>Show a provider's first-use message, at most once per session</i>
----------------------	--

Description

If `provider@first_use_message` is non-NULL and no message has been shown for this provider's source in the current R session, emits the message via `cli::cli_inform()` and records that it has been shown. Subsequent calls for the same provider are silently ignored.

Usage

```
maybe_show_first_use(provider)
```

Arguments

<code>provider</code>	(FontProvider) The provider object whose message should (possibly) be displayed.
-----------------------	--

Value

(invisible(NULL)) Called for its side-effect only.

<code>preview_font</code>	<i>Preview a font by ensuring it's installed and drawing a sample string</i>
---------------------------	--

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,
  variants = NULL,
  sample = "The quick brown fox jumps over the lazy dog",
  size = 28,
  subset = "latin",
  regular.wt = 400,
  bold.wt = 700
)
```

Arguments

name	(character(1)) Font name as used by the provider (e.g. "oswald").
provider	(character(1)) Provider name to use (default: "bunny")
family	(character NULL) Optional family name to register the font under (default: NULL)
variants	(list NULL) For file-based providers (provider = "file", provider = "url", or a FontProviderFile object). Named list of symbolic variant keys to file paths / URLs / filename stems. Passed through to add_font() . Ignored for weight-based providers (default: NULL).
sample	(character(1)) Sample text to display (default: "The quick brown fox jumps over the lazy dog")
size	(numeric(1)) Font size in points for the preview (default: 28)
subset	(character(1)) Glyph subset to request (default: "latin")
regular.wt	(integer(1)) Regular weight to display (default: 400)
bold.wt	(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_provider	<i>Register a font provider for the current session</i>
-------------------	---

Description

Adds a `FontProvider` object to the session-level registry so it can be referenced by name in `add_font()`. The registry is cleared when the R session ends.

Usage

```
register_provider(provider, overwrite = FALSE)
```

Arguments

provider	(FontProvider) A validated <code>FontProvider</code> object to register.
overwrite	(logical(1)) If TRUE, silently overwrite an existing provider with the same source name. If FALSE (default), error instead.

Value

(NULL) Called for its side-effect; returns NULL invisibly.

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

Description

Replace disallowed characters with `-` and convert to lower-case, collapsing runs of `-` and trimming leading/trailing `-`, so the resulting id is safe to use in filenames and as a URL path segment.

Usage

```
safe_id(name)
```

Arguments

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

Value

(character(1)) Sanitised identifier.

unregister_provider	<i>Remove a font provider from the session registry</i>
---------------------	---

Description

Dispatches on a source name (character) or a FontProvider object.

Usage

```
unregister_provider(x)
```

Arguments

x	(character(1) FontProvider) Source name of the provider to remove, or the FontProvider object itself.
---	---

Value

(NULL) Called for its side-effect; returns NULL invisibly.

update_download_and_cache	<i>Download missing weights and update an existing cache entry</i>
---------------------------	--

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

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

Usage

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

Arguments

font_file	(character(1)) Path to the .woff2 file to convert.
overwrite	(logical(1)) If TRUE, overwrite an existing .ttf conversion.
remove_old	(logical(1)) If TRUE, remove the original .woff2 file after conversion try.
quiet	(logical(1)) If TRUE, suppress the success message. Errors always abort regardless of this setting (default: FALSE).

Value

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

Index

* **cache**

- cache_clean, 10
- cache_get, 12
- cache_get_variants, 12
- cache_get_weights, 13
- cache_read, 14
- cache_remove, 14
- cache_write, 17
- .add_font_direct_url, 3
- .add_font_file, 4
- .add_font_local, 4
- .add_font_weight, 5
- .fetch_url_to_cache, 6
- .persist_cache_entry, 6
- .validate_variants, 7
- add_font, 8
- add_font(), 32, 33
- as_CacheEntryList, 9
- as_FontProvider, 9
- as_list, 10
- cache_clean, 10
- cache_clean(), 12–15, 18
- cache_file_path, 11
- cache_get, 12
- cache_get(), 11, 13–15, 18
- cache_get_variants, 12
- cache_get_variants(), 11–15, 18
- cache_get_weights, 13
- cache_get_weights(), 11–15, 18
- cache_read, 14
- cache_read(), 11–13, 15, 18
- cache_remove, 14
- cache_remove(), 11–14, 18
- cache_set, 15
- cache_set(), 6
- cache_ttf_filename, 16
- cache_ttf_path, 16
- cache_variant_paths, 17
- cache_write, 17
- cache_write(), 6, 11–15
- CacheEntry, 18
- CacheEntry(), 6, 20, 23, 24
- CacheEntryList, 19
- CacheMeta, 19
- CacheMeta(), 6
- conv_fun, 20
- copy_and_cache_local, 20
- copy_variant_to_cache, 21
- delete_files, 21
- download_and_cache, 22
- download_and_cache_file, 23
- download_and_cache_url, 24
- download_variant_file, 24
- download_variant_generic, 25
- download_weights, 26
- FontProvider, 26
- FontProvider(), 28, 29
- FontProviderDirectURL, 27
- FontProviderFile, 27
- FontProviderLocal, 28
- FontProviderWeight, 28
- get_cache_dir, 29
- get_provider_details, 30
- list_providers, 30
- maybe_show_first_use, 31
- preview_font, 31
- register_from_cache, 32
- register_provider, 33
- safe_id, 33
- unregister_provider, 34

update_download_and_cache, [34](#)

woff2_to_ttf, [35](#)