

# Package ‘hexfont’

May 8, 2026

**Type** Package

**Title** 'GNU Unifont' Hex Fonts

**Version** 1.0.0

**Description** Contains most of the hex font files from the 'GNU Unifont Project' <<https://unifoundry.com/unifont/>> compressed by 'xz'. 'GNU Unifont' is a duospaced bitmap font that attempts to cover all the official Unicode glyphs plus several of the artificial scripts in the '(Under-)ConScript Unicode Registry' <<https://www.kreativekorp.com/ucsur/>>. Provides a convenience function for loading in several of them at the same time as a 'bittermelon' bitmap font object for easy rendering of the glyphs in an 'R' terminal or graphics device.

**URL** <https://github.com/trevorld/hexfont>,  
<https://trevorldavis.com/R/hexfont/>

**BugReports** <https://github.com/trevorld/hexfont/issues>

**License** GPL (>= 2)

**Depends** R (>= 4.0.0)

**Imports** bittermelon (>= 1.1.2), tools, utils

**Suggests** knitr, oblicubes, rmarkdown, testthat, Unicode

**VignetteBuilder** knitr, rmarkdown

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Trevor L. Davis [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-6341-4639>>),  
GNU Unifont authors [cph]

**Maintainer** Trevor L. Davis <trevor.l.davis@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-03-11 23:20:02 UTC

## Contents

unifont . . . . .	2
unifont_combining . . . . .	3
unifont_version . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

unifont	<i>Load GNU Unifont font</i>
---------	------------------------------

---

## Description

The function `unifont()` loads in several GNU Unifont hex files as a single `bittermelon::bm_font()` object.

## Usage

```
unifont(
  upper = TRUE,
  jp = FALSE,
  csur = TRUE,
  sample = FALSE,
  ucp = NULL,
  cache = getOption("unifont.cache", NULL)
)
```

## Arguments

<code>upper</code>	Include glyphs above the Unicode Basic Multilingual plane.
<code>jp</code>	Use Japanese version of Chinese characters.
<code>csur</code>	Include (Under-)Conscript Unicode Registry glyphs.
<code>sample</code>	Add circle to "Combining" characters.
<code>ucp</code>	Character vector of Unicode Code Points: glyphs not in this vector won't be read in. If NULL (default) read every glyph in the font.
<code>cache</code>	If TRUE read a cached version of this font from <code>tools::R_user_dir("hexfont", "cache")</code> if it exists and if it does not exist than create a cached version of this font. If FALSE don't read or write a cached version of this font (even if it exists). If NULL read a cached version of this font if it exists but if it does not exist then don't create it. This argument is ignored if <code>ucp</code> is not NULL.

## Value

A `bittermelon::bm_font()` object. If `cache` is TRUE then as a side effect may create an `.rds` file in `tools::R_user_dir("hexfont", "cache")`.

**Examples**

```

# Much faster to load only the subset of GNU Unifont one needs
# Mandarin Chinese
if (require("bittermelon")) {
  s <- "\uff32\u5f88\u68d2\uff01"
  font <- unifont(ucp = str2ucp(s))
  bm <- as_bm_bitmap(s, font = font)
  print(bm, px = px_ascii)
}

# Emoji
if (require("bittermelon")) {
  s <- "\U0001f42d\U0001f432\U0001f435"
  font <- unifont(ucp = str2ucp(s))
  bm <- as_bm_bitmap(s, font = font)
  print(bm, px = px_ascii)
}

# Will take more than 5s on CRAN machines
# Compiling the entire font from the hex files takes a long time
system.time(font <- unifont(cache = FALSE))
prettyNum(length(font), big.mark = ",") # number of glyphs
# It is usually much faster to use a cached version of the font
if (file.exists(hexfont::unifont_cache_filename())) {
  system.time({font_from_cache <- unifont(cache = TRUE)})
}

```

---

unifont\_combining      *Get combining character code points*

---

**Description**

unifont\_combining() returns a character vector of the code points for all the "combining" characters in Unifont.

**Usage**

```
unifont_combining(upper = TRUE, csur = TRUE, unicode = FALSE)
```

**Arguments**

upper	Include glyphs above the Unicode Basic Multilingual plane.
csur	Include (Under-)Conscript Unicode Registry glyphs.
unicode	Include combining glyphs assigned by the Unicode Consortium (i.e. not ones in the Private Use Area like the CSUR ones). By default FALSE since <code>bittermelon::bm_compose()</code> can usually guess that a Unicode Consortium assigned glyph is a combining glyph by using <code>Unicode::u_char_property()</code> .

**Value**

A character vector of Unicode code points

**See Also**

Can be used with the `pua_combining` argument of `bittermelon::bm_compose()` and `bittermelon::as_bm_bitmap()`.

**Examples**

```
uc <- unifont_combining()
print(uc)

# Tengwar with combining glyphs
if (require("bittermelon")) {
  s <- "\ue004\u014\u04a\u005\u000\u040\u022\u04a\u003\u04e"
  font <- unifont(ucp = str2ucp(s))
  bml <- as_bm_list(s, font = font)
  to_raise <- which(names(bml) %in% c("U+E04A", "U+E04E"))
  bml[to_raise] <- bm_shift(bml[to_raise], top = 1L)
  bml <- bm_compose(bml, pua_combining = uc)
  bml <- bm_pad(bml, type = "trim", left = 1L, right = 0L)
  bm <- bm_call(bml, cbind)
  print(bm, px = px_ascii)
}
```

---

unifont_version	<i>GNU Unifont version number</i>
-----------------	-----------------------------------

---

**Description**

The function `unifont_version()` returns the GNU Unifont version number this package packed their hex files from.

**Usage**

```
unifont_version()
```

**Value**

The Unifont version number as a `numeric_version()` class.

**Examples**

```
unifont_version()
```

# Index

bittermelon::as\_bm\_bitmap(), 4  
bittermelon::bm\_compose(), 3, 4  
bittermelon::bm\_font(), 2  
  
numeric\_version(), 4  
  
Unicode::u\_char\_property(), 3  
unifont, 2  
unifont\_combining, 3  
unifont\_version, 4