luaharfbuzz Documentation

Module `harfbuzzsubset`

Lua bindings to the Harfbuzz subset API.

- Wiki
- Source on Github
- API Coverage Status

This module provides bindings to the HarfBuzz `hb-subset` API. It is designed to be used together with the main harfbuzz module.

Typical usage:

```
local hb = require('harfbuzz')
local hb_subset  = require("harfbuzzsubset")

local input = hb_subset.SubsetInput.new()

local wght = hb.Tag.new("wght")
local wdth = hb.Tag.new("wdth")
local face = hb.Face.new("myfont.ttf")

input:pin_axis_location(face, wght, 100)
input:pin_axis_location(face, wdth, 100)

input:keep_everything()

local new_face = hb_subset.subset(face, input)

local blob = new_face:blob()
local fh, err = io.open("out.ttf", "wb")
if not fh then error(err) end
fh:write(blob:get_data())
fh:close()
```

Info:

- **Copyright**: 2025
- **License**: MIT
- **Author**: Patrick Gundlach <gundlach@speedata.de>

Functions

subset (face, input)	Wraps `hb_subset_or_fail`.
version ()	Wraps `hb_version`.

Class SubsetInput

SubsetInput.new ()	Wraps `hb_subset_input_create_or_fail`.
SubsetInput:unicode_set ()	Wraps `hb_subset_input_unicode_set`.
SubsetInput:pin_axis_location (face, tag, value)	Wraps `hb_subset_input_pin_axis_location`.

SubsetInput:keep_everything ()	Wraps `hb_subset_input_keep_everything`.
SubsetInput:__gc ()	Wraps `hb_subset_input_destroy`.

subset (face, input)

Wraps `hb_subset_or_fail`.

Performs the actual subset operation, producing a new `Face` that only contains the glyphs and data configured on this input.

Parameters:

- face source `Face` object to subset.
- input `SubsetInput` object to subset based on.

Returns:

new `Face` object representing the subset, or `nil` on failure.

version ()

Wraps `hb_version`.

Lua wrapper for `hb_subset_input_t` type.

Objects of this type control which parts of a `Face` are kept when creating a subset.

Instances are usually created via SubsetInput.new.

SubsetInput.new ()

Wraps `hb_subset_input_create_or_fail`.

Initializes a new `hb_subset_input_t`. The returned object can be used to configure which glyphs and features are kept when subsetting a face.

Returns:

`SubsetInput` object, or `nil` on failure.

SubsetInput:unicode_set ()

Wraps `hb_subset_input_unicode_set`.

Returns the Unicode set associated with this subset input. The returned harfbuzz.Set can be modified (e.g. via add) to control which Unicode codepoints are included in the subset.

Returns:

`Set` object representing the Unicode set, or `nil` on failure.

SubsetInput:pin_axis_location (face, tag, value)

Wraps `hb_subset_input_pin_axis_location`.

Pins a variation axis at a specific value for subsetting.

Parameters:

- face `Face` object whose design space is being subset.
- tag `Tag` object representing the axis tag (e.g. `"wght"`, `"wdth"`).
- value numeric axis value to pin.

Returns:

`true` on success, `false` on failure.

SubsetInput:keep_everything ()

Wraps `hb_subset_input_keep_everything`.

Configures the input so that everything in the original face is kept. This is useful as a starting point before applying more specific filters.

SubsetInput:__gc ()

Wraps `hb_subset_input_destroy`.

Destructor for subset inputs. Normally this is invoked automatically by the garbage collector; applications rarely need to call it directly.