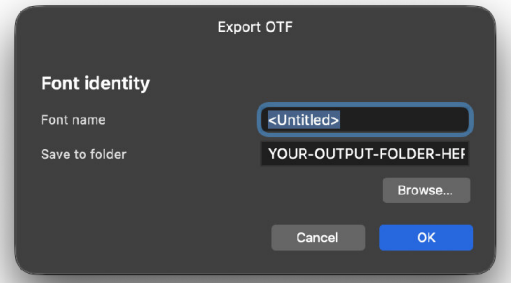
affinity-font-export

Reads the artboards built by `affinity-font-setup.js`, extracts the glyph outlines, and writes a working `.otf` font file to a folder you choose. Run this after you have drawn your glyphs — it is step two of the pair.

> ***Before first use:*** *open the script and set* `OUTPUT_FOLDER` *at the top to a real folder (it ships as* `YOUR-OUTPUT-FOLDER-HERE`*), and enable that folder under Affinity* ***Settings → File System Access*** *Without it, every write fails with* `PERMISSION_DENIED` *— including writes to the Desktop path most people will use. Set the constant once and every export works from then on.*



Installation

1. Download the zip file and unzip it.
2. Run the script in one of two ways: **a) Run directly** — Open the Script Editor panel (**Window > Scripting > Script Editor**), paste the contents of `affinity-font-export.js` into the editor, and press **Run**. **b) Save to Script Library first** — Follow step (a) above, then click **Save As...** to save the script to the Script Library. Open the Script Library panel (**Window > Scripting > Script Library**) and click the saved script once to run it.

Purpose

Converts a drawn font workspace into an installable font. The script walks every artboard on the first spread, turns each glyph's outlines into font coordinates (origin at the LSB marker, y-up from the baseline), builds the OpenType tables from scratch in pure script code — no external libraries — and saves `Desktop/<FontName>.otf`. A `.notdef` fallback glyph is always included as glyph zero.

Workflow (run order)

1. **Run `affinity-font-setup.js`** to create the workspace.
2. **Draw your glyphs** into the artboards (see the setup doc's drawing tips).
3. **Run this script** with the workspace document open:
   - Type the font name, check the save folder, and press OK.
   - Read the result alert: how many glyphs exported, where the file went, and which artboards were skipped and why.
4. **Install and test**: double-click the `.otf` (Font Book on Mac / right-click → Install on Windows), then type the characters in any app. Empty artboards are skipped, so a first test export with one drawn glyph is a sensible smoke test.

Dialog

- **Font name** — defaults to the document title (or `My Font`). Anything outside `A-Z a-z 0-9 _ -` becomes `_` in the file name.
- **Save to folder** — where the `.otf` is written, prefilled with the `OUTPUT_FOLDER` constant from the top of the script. Leaving the placeholder (or an empty field) aborts the export with a reminder. The folder must be allowed under Affinity Settings → File System Access. Windows paths work too (backslashes or forward slashes; trailing separators are trimmed) — the Windows path is code-reviewed but not run-tested here.
- **Browse...** — opens the operating system's file picker (the SDK has no folder picker: folders alone leave the picker's OK disabled, so pick any file inside the target folder and the script fills in its folder). If the picker refuses to open from inside the dialog, just type the folder — the manual field always works.

Cancelling the dialog exports nothing.

What a successful run looks like

```
Exported 1 glyphs to:
~/Desktop/a-font.otf
Skipped:
O (U+004F) (no artwork drawn)
G (U+0047) (no artwork drawn)
```

Skipped artboards are normal — they are simply left out of the font. Common skip reasons and what they mean:

- **(no artwork drawn)** — the artboard is empty. Draw something or ignore it.
- **(could not parse name)** — the artboard was renamed and no longer contains `(U+XXXX)`. Restore the suffix.
- **(missing required marker)** — an LSB, RSB, or baseline guide was deleted. Re-draw it or re-run setup for that glyph.

- **(RSB <= LSB)** — the sidebearing guides crossed. Drag the red lines back into order.

Requirements

- Affinity by Canva 3.3 (per script header).
- The workspace document open, with artboards named `Name (U+XXXX)` and intact LSB/RSB/baseline markers (i.e. a setup-script product).
- **Affinity Settings File System Access must allow the Desktop.** Without it, every write fails with `PERMISSION_DENIED` — including writes to the Desktop path the script discovers correctly. Enable Desktop there once and exports work from then on.

What this is (and is not) for

Same scope as the setup script: a **hobby/small-project font maker, not professional type software**. Each export is a single Regular weight with:

- No kerning pairs, no hinting, no OpenType layout features.
- Fixed vertical metrics (ascender 800, descender −200, line gap 200. and default OS/2 values.
- A basic Latin-friendly character map (BMP coverage) — ideal for **dingbats, logos, symbols, and icon fonts**, not for a text face.

Install the result, type with it, and if a glyph looks wrong, fix the drawing and re-export — the round trip takes seconds.

Known limitations

- Only artboards on the **first spread** are read.
- Glyph cells with no artwork are skipped silently into the skip list (by design — see above).
- Overlapping or self-intersecting outlines export literally; the script performs no outline cleanup or overlap removal.
- Vertical metrics and naming fields are fixed per export; there is no style/weight picker beyond the font name.
- Writing anywhere other than an allowed folder (e.g. a Desktop that is not enabled under File System Access) fails — see Requirements.

Changelog

1.2.0 - OS/2 `sCapHeight`/`sxHeight` now come from the document's actual cap-height/x-height marker lines (falling back to defaults only if a document has none), instead of always being hardcoded; standard glyph names (`A`-`Z`, `a`-`z`, digits, punctuation) now use their predefined CFF SIDs instead of bloating the custom string table; an unexpected error on one artboard is now skipped instead of aborting the whole export; settings moved into a `CONFIG` object

1.1.0 - "Save to folder" dialog field (prefilled with the new `OUTPUT_FOLDER` constant —

set it before first use); "Browse..." button fills the field from the OS file picker (pick any file in the target folder); write failures name the exact folder and point at File System Access; `File.create` migration (deprecation warning gone); Windows paths supported (both separators, trailing trimmed)

1.0.0 - Initial version