

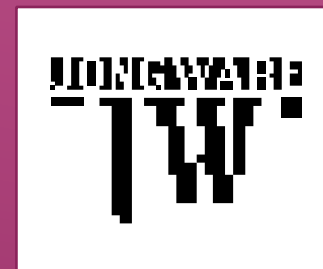


IndyFont 1.1

Create Pure OpenType Fonts
with InDesign CS4 / CS5 / CS6 / CC

A Jongware & Indiscripts Tool

U S E R M A N U A L



Overview



1. Description

IndyFont is a tool for Adobe InDesign® with which you can create your very own font in the familiar environment of InDesign itself, and use this new font straight away in your own documents.

With IndyFont you can create a new font from scratch, based on your very own character designs. The file it produces is a valid, cross-platform, OpenType font, which can be used in any software that supports OpenType fonts.

NOTE IndyFont is not a font editor in the traditional sense. You cannot edit an existing font other than the fonts you create yourself. (However, as we shall see, you can do great things *from* existing fonts!)

2. System requirements

- Mac OS X 10.6 or later,
or Windows XP / Vista / 7 / 8 (x86 or x64 editions).
- 800×600 pixel screen-resolution or greater.
- Adobe InDesign CS4, CS5, CS5.5, CS6, or CC.

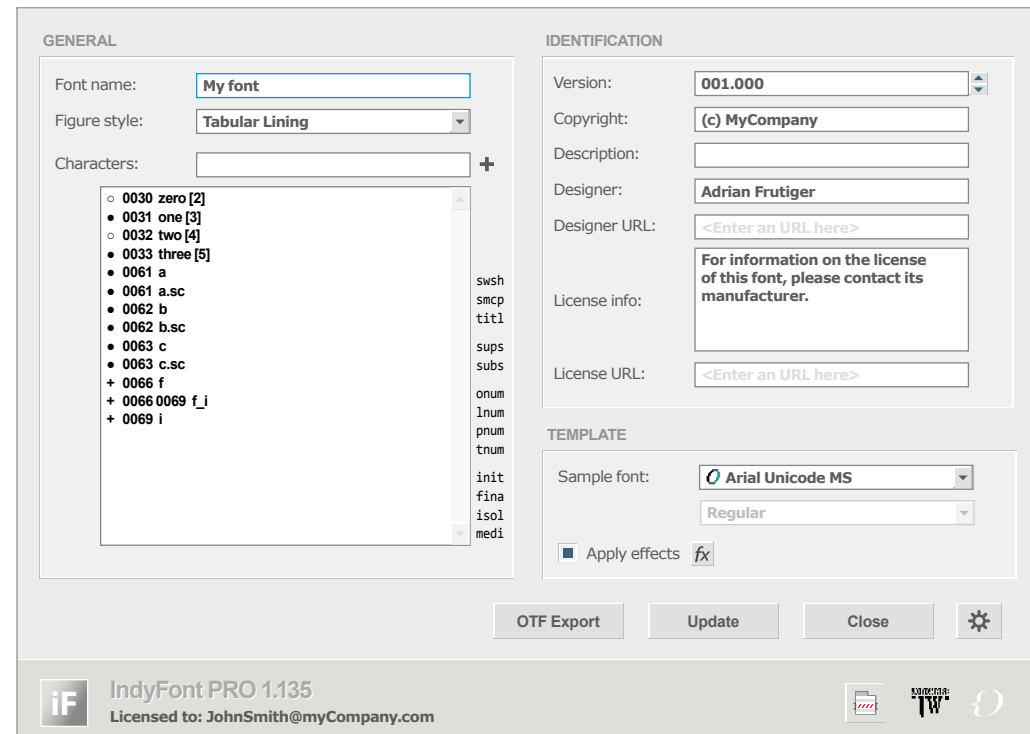


IndyFont both supports
CC 9.x and CC 2014

3. TRY vs. PRO version

You can download a free tryout version of IndyFont at:
<http://www.indiscripts.com/blog/public/scripts/IndyFontTry.zip>.
It offers most of the features of the PRO release, except it will only export a single character per font.

NOTE We strongly encourage you to install and test the TRY version before you purchase the PRO license of the product. Always make sure that your system meets the requirements.



The main dialog of IndyFont has a very similar look-and-feel in both Mac OS and Windows environments. The following languages are available (depending on your InDesign locale):

- English (default)
- French
- German

Getting started



1. Before you install

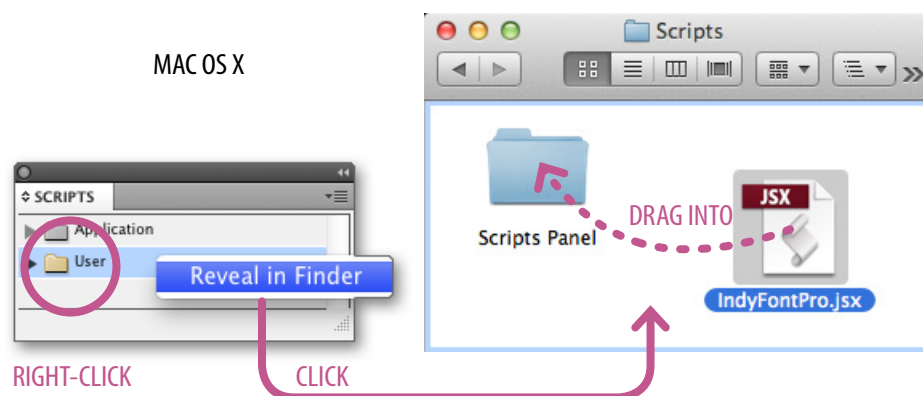
IndyFont resides in a single file: **IndyFontPro.jsx**. When you download the file from your private link, however, it is originally zipped.

The first step is to unzip the .zip file so you can place **IndyFontPro.jsx** at the desired location (see below).

As a precaution before you go on, save your working files and restart InDesign in a clean session.

2. Installing in Mac OS X

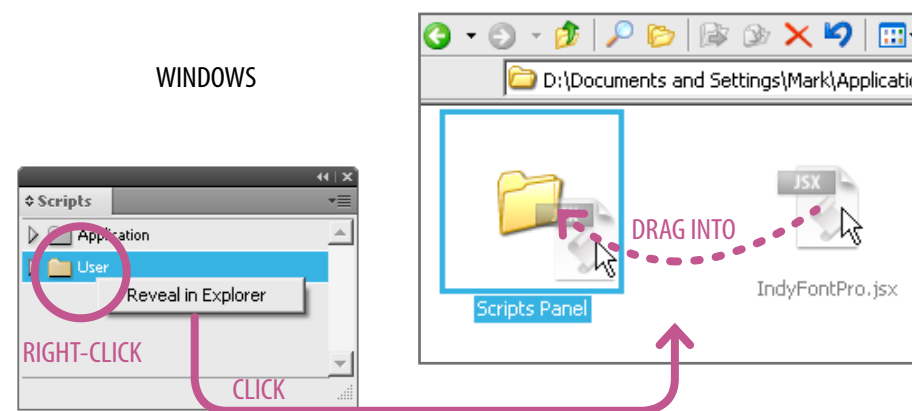
- 1) In InDesign, open the Scripts panel as follows:
 - CS4: Window ► Automation ► Scripts.
 - CS5, CS5.5, CS6, and CC: Window ► Utilities ► Scripts.



- 2) You see there two main folders: Application and User. Right-click the User folder and pick “Reveal in Finder.”
- 3) You should now see a Scripts Panel folder. Drag **IndyFontPro.jsx** into there. Congratulations, IndyFont is now installed!

3. Installing in Windows

- 1) In InDesign, open the Scripts panel as follows:
 - CS4: Window ► Automation ► Scripts.
 - CS5, CS5.5, CS6, and CC: Window ► Utilities ► Scripts.
- 2) You see there two main folders: Application and User. Right-click the User folder and pick “Reveal in Explorer.”
- 3) You should now see a Scripts Panel folder. Drag **IndyFontPro.jsx** into there. Congratulations, IndyFont is now installed!



Getting started



4. Installing an update

If you are notified that an update of the product is available, simply download the new package from your private link, then unzip and install the file **IndyFontPro.jsx** over the previous one, i.e. at the same location.

The new version is instantly functional, and your global settings are all preserved.

5. Running IndyFont from the Scripts panel

Once the installation is done, switch back to InDesign. You can run IndyFont from the Scripts panel as follows:

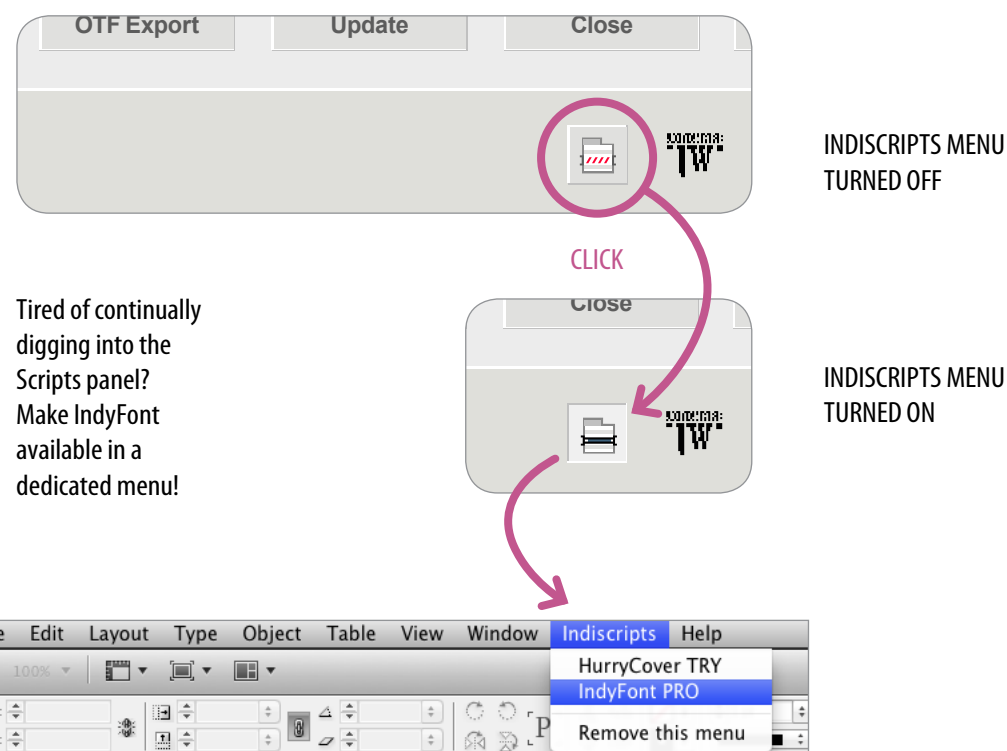
- 1) In InDesign, display the Scripts panel via:
 - Window ► Automation ► Scripts (CS4), or
 - Window ► Utilities ► Scripts (CS5, CS5.5, CS6, or CC).
- 2) Look for **IndyFontPro.jsx** in the User folder, then double-click on it.

6. Running IndyFont from the Indiscripts menu

To have IndyFont available in the InDesign's menu bar, simply click the Indiscripts-menu button as shown in the screenshot, below, then close the dialog box.

You can now run IndyFont going into:

Indiscripts ► IndyFont PRO

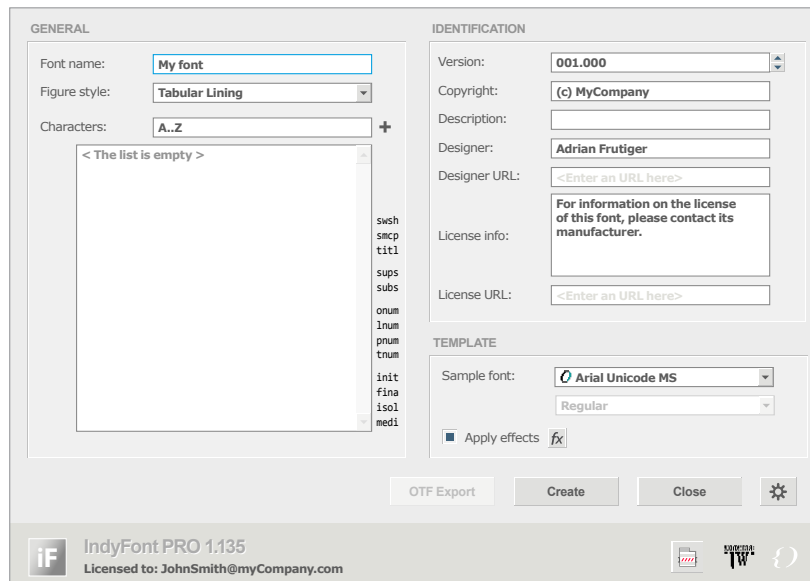


Getting started

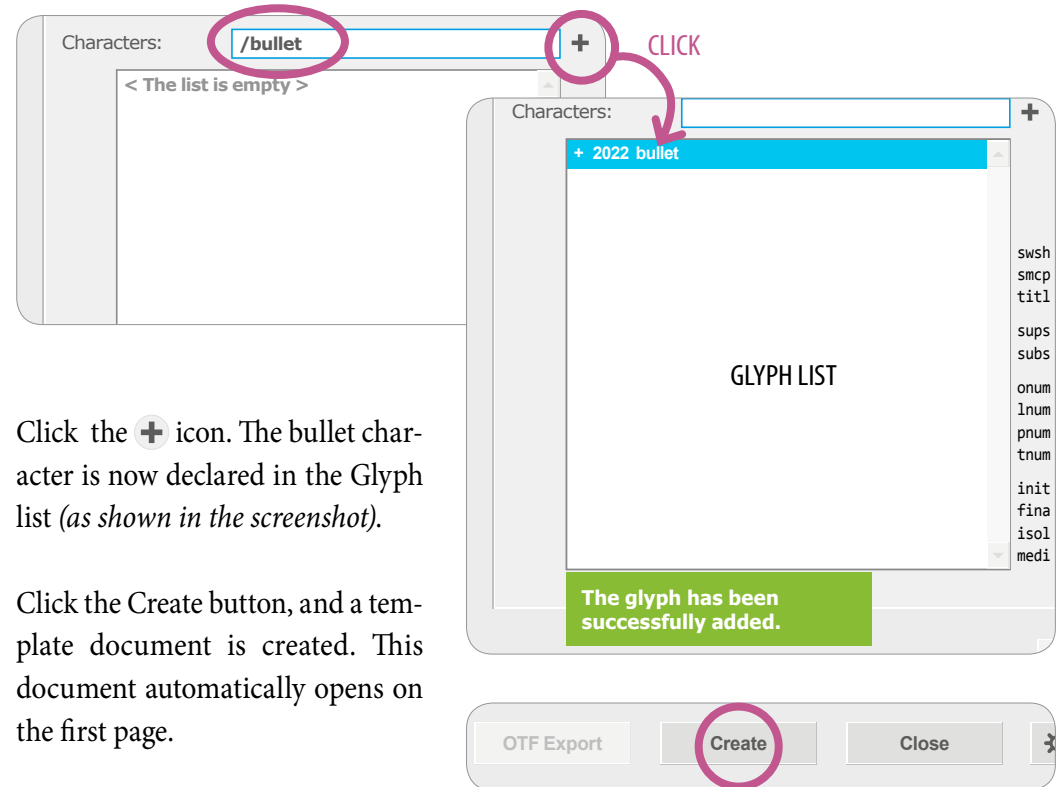
7. Create your first font

After installing the script, you are ready to create your first font!

- 1) Run IndyFont, and the main dialog pops up.

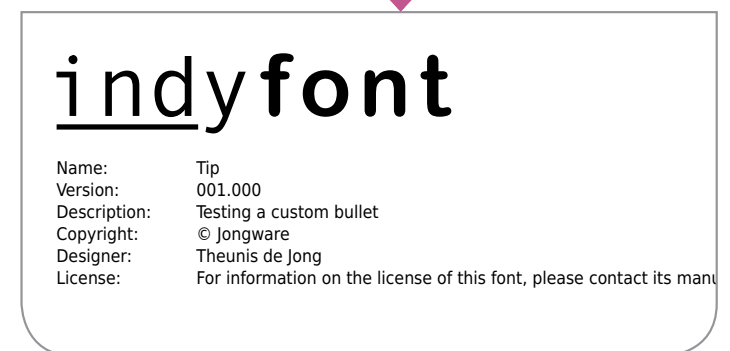


- 2) The default font name is “My Font”, and you can leave that like that for your first experiment, or change it to anything you like. In this example we will use the name “Tip”.
- 3) The only character in this font is going to be a custom bullet. Delete the default text A..Z and type /bullet in the Characters field.



- 4) Click the + icon. The bullet character is now declared in the Glyph list (as shown in the screenshot).
- 5) Click the Create button, and a template document is created. This document automatically opens on the first page.

NOTE You can see the information from the main dialog is repeated in here: the font name, and the default version number, copyright, designer and license strings. All of these could be changed later in the main dialog of IndyFont.



INFORMATION PAGE

Getting started

- 6) The second page contains a huge sample character at 1000 points. This is the scale at which you must work! Character outlines are stored inside a font at this size, and only when drawing an actual character on the screen (or anywhere else), it is scaled to the appropriate size.

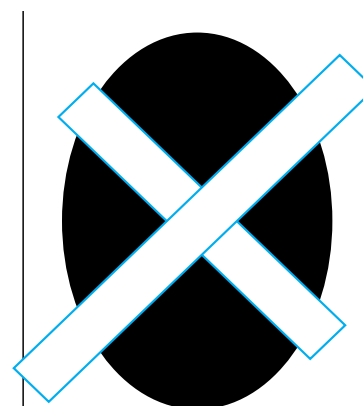
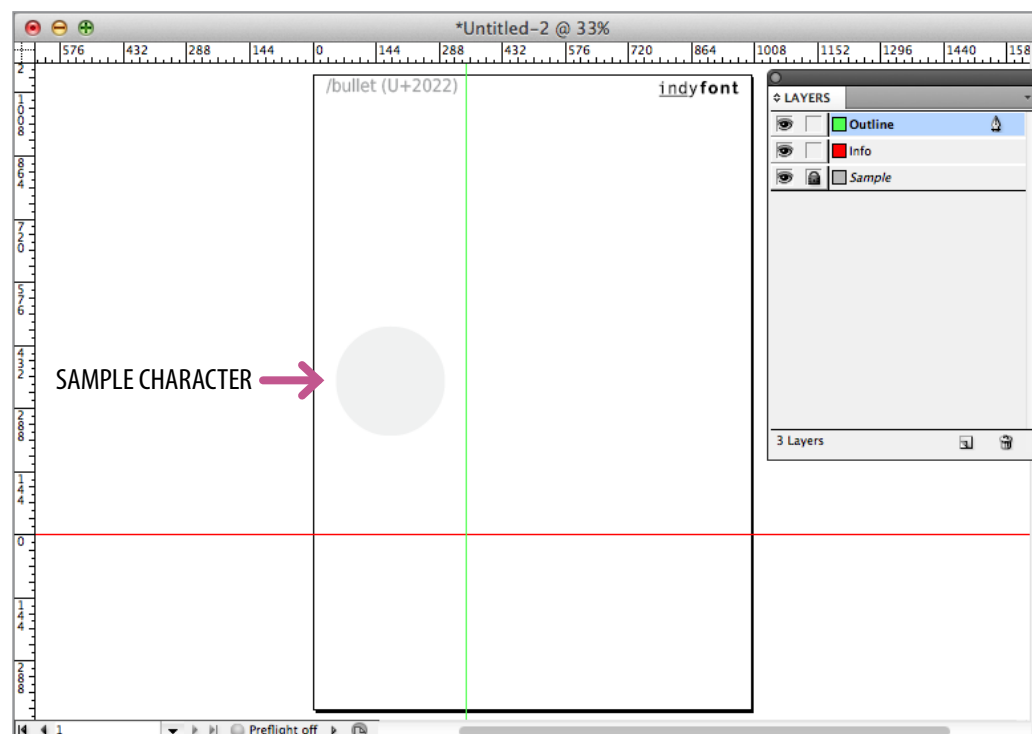
The light gray sample character is just that: a reminder of what character you should be drawing, and at what size. It's in a layer of its own ("Sample"), and this layer is locked, so you cannot accidentally select it.

- 7) Let's draw a bullet. Make sure that:
- you have the layer "Outline" selected;
 - you are drawing with no stroke; and
 - you are drawing with a [Black] fill.

NOTE Only [Black] and [Paper] filled vector drawings on the layer "Outline" are considered "something". Strokes (outlines) are ignored, and we recommend that you leave this at 0 points so you don't get a skewed view of how your glyph is going to look.

- 8) Draw something like this shape. Make sure it is at least as large as the example bullet. The blue outlines are to show the construction of this bullet only: two thin rectangles filled with a [Paper] color, on top of a simple oval filled with [Black].

You can combine basic shapes or use any of InDesign's other vector tools, such as the Pathfinder, to get exactly what you want, as long as you only use Fill, *not* Stroke.



Let's design a custom bullet on the layer "Outline". Remember that only [Black] and [Paper] filled shapes are "seen" by IndyFont.

Getting started



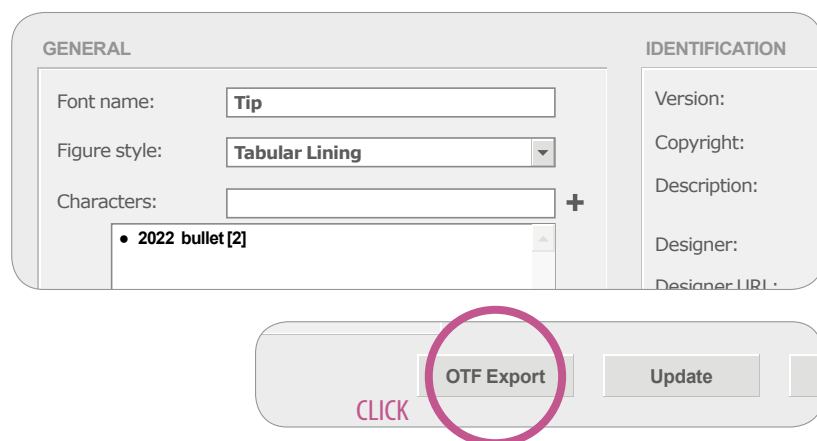
TIP You can disable the “Sample” layer’s visibility to get a better preview.

- 9) The vertical green guide indicates the width of your new character. Make sure there is a bit of white space at both the left and right side of your design. Drag the entire image to the right if it extends beyond the left side of the template, and drag the green guide to change the space on the right hand side.

If you are satisfied with your design, let’s make it into a font.

- 10) Export the OTF file:

- Run IndyFont again. It goes over each template page and checks if it can find anything usable.

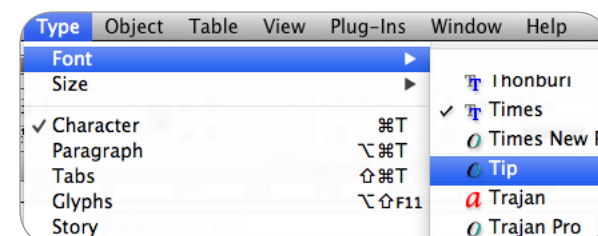


- If no errors are found, click “OTF Export”. IndyFont processes your character(s), and if all is well, it asks you to select a destination folder.

- You can save your font anywhere you want. For this example, save it into InDesign’s Fonts folder.

NOTE InDesign continuously scans your system to see if the font list was changed, and it usually picks up your new font right away. If it does not, switch to another application, wait a moment, and switch back to InDesign; that should be enough to trigger InDesign into re-building the font list.

- 11) Check if InDesign recognizes your new font:
- (a) in the Type Font menu;
 - (b) in a new Paragraph Style;
 - (c) or using the Glyphs panel.

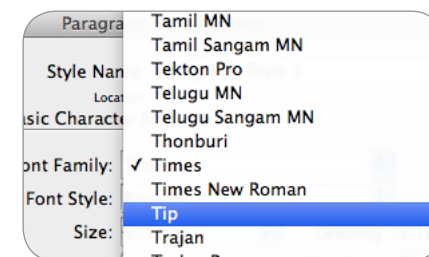


(a)

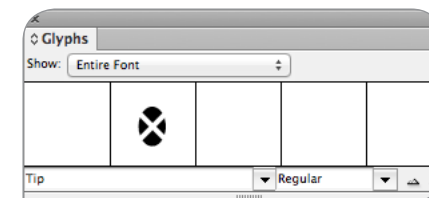
- 12) Create a new document, type some text, and insert a bullet (Option/Alt + 8) at the start of your paragraph.

TIP You can apply your new bullet automatically using “Bullets and Numbering”.

- 13) Select the bullet and apply the font “Tip”.



(b)



(c)

If you are not entirely satisfied with your design, go back to the font template, adjust it, then generate the font again. Second time around, IndyFont will warn you that the file already exists and you can overwrite it.

Manage your Character Set



1. Interaction between Glyph list and template pages

Once IndyFont has built a font template and as long as you have it active in InDesign, a close connection is established between pages and characters.

Apart from the Information page that introduces the font template, each valid template page in the document is intended to identify a distinct character, or glyph, in your font.

NOTE In most cases we use the terms “glyph” and “character” as loose equivalents. Unicode, however, defines a character as an abstract unit of text for which it provides a unique “code point” (a number). In that sense a character has no visual rendering in itself. By contrast, a glyph is the concrete, visible, specific shape of a character. Thus, you may have a single character rendered by multiple glyphs (referred to as “alternates”), as well as multiple characters mixed in a single glyph (referred to as a “ligature”). Therefore, strictly speaking, each canonical page in your font template does identify a *glyph*.

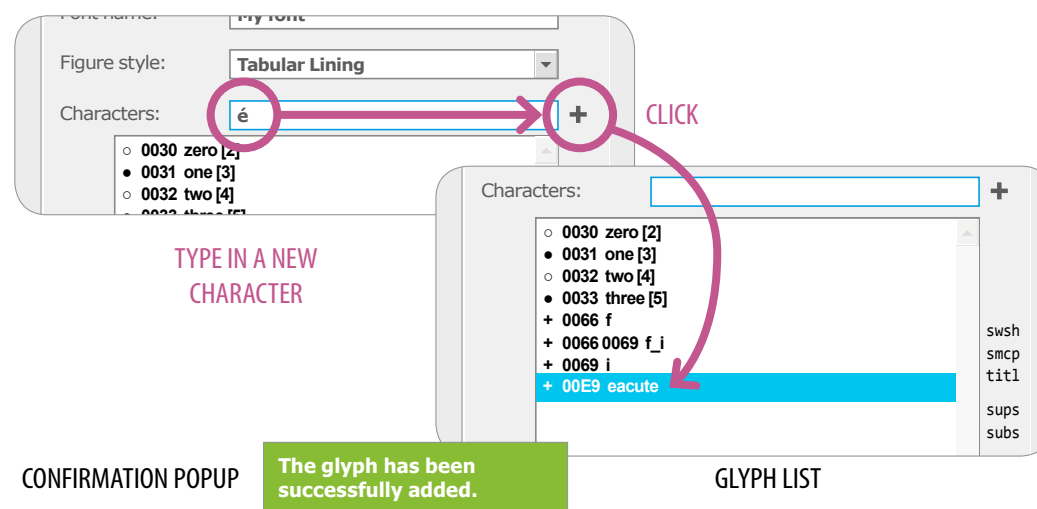
Whenever you run IndyFont, it scans the document and updates the Glyph list so that it matches your current character set.

Conversely, any insertion or deletion in the Glyph list is mirrored in the font template when you click the Update button.

2. Adding a new character

The easiest way to add a simple character into the Glyph list is to proceed as follows:

- 1) Run IndyFont and go into the Characters field. (To quickly set the focus on it and have any existing input already selected, you can click the “Characters” caption itself.)
- 2) Enter, or paste, the desired character—for example the letter é, as in the screenshot, below.
- 3) Click the **+** icon to the right of the edit box. A new item is then inserted in the Glyph list, e.g. +00E9 eacute. This formal code is based on both the Unicode value (U+00E9) and the standard glyph name (*eacute*) of the new character.



Manage your Character Set



- 4) Finally, if you want to have this new character immediately added to the template, click the Update button (or, in case no template exists yet, the Create button). A new template page is then added at the end of the document, with all elements required to identify and process the glyph.

3. Manually adding a template page

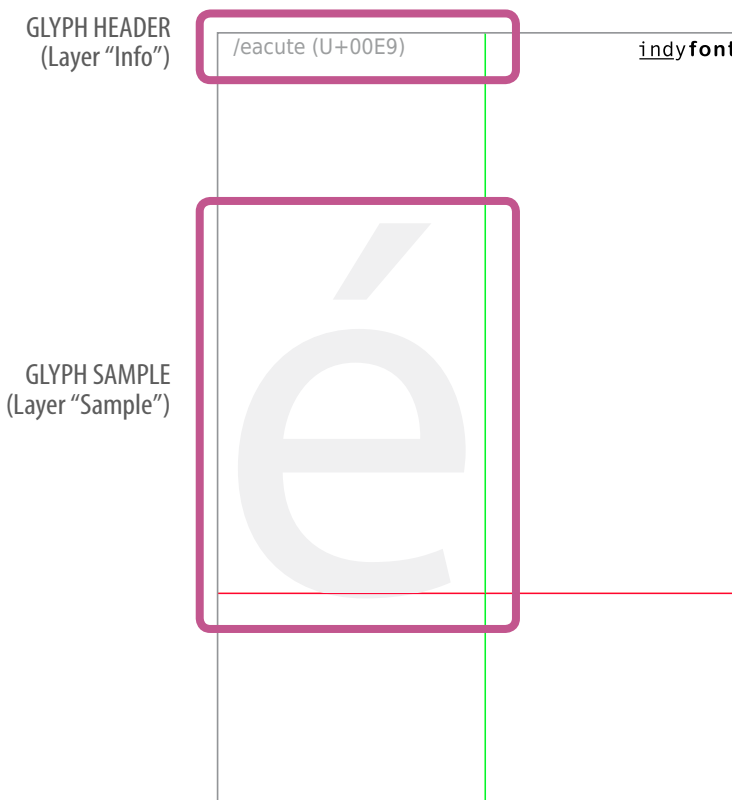
An alternative method to have a new character available is to manually create a template page, subject that you properly fill in the *Glyph header* frame (at the top left corner of the page).

- 1) Do one of the following:
 - Click the “Create new page” button at the bottom of the Pages panel;
 - Using the InDesign menu, go into: Layout ► Pages ► Add Page (*or* Insert pages...);
 - Alternatively, you can duplicate an existing glyph page; right-click its thumbnail in the Pages panel, then select “Duplicate Spread”.
- 2) In any case, make sure that the new page has the “Glyphs” master applied on it. (In the Pages panel, you should see the letter “G” on the page thumbnail.)
- 3) In the case you have *added*—not *duplicated*—a page, the Glyph header inherits the string “.notdef” from the master (also, the glyph sample shows a default question mark). In order to modify

this header, you have to unlock the “.notdef” frame from the “Glyphs” master spread. Then, going back to the template page, Cmd-Shift-Click the header to override the master object.

NOTE The Glyph header belongs to the “Info” layer, which for some reason might be locked. Make sure you have unlocked that layer too.

- 4) Go into the Glyph header text frame and type in the desired character; for example, the letter à. You can use as well a full IndyFont header format, such as `/agrave (U+00E0)`, but this is not a requirement. (Unless the leading slash is used, IndyFont will just interpret any character *as itself*.)
- 5) Run IndyFont. If your template page is successfully parsed, the new character should appear in the Glyph list. Otherwise, IndyFont will prompt an alert, typically “Page xx has no valid glyph header and will be ignored.”



Canonical page generated for the letter é (U+00E9). For more on template page metrics and layout, see “Fashion your glyphs”.

Manage your Character Set

NOTE Any template page which does not have a valid Glyph header is seen as an “extra” page and will be simply ignored. Various factors may lead to such a situation. Some examples: the page has no header text frame at all; the frame is empty or cannot be parsed as having a valid syntax; the glyph has been already specified somewhere else; the specifier is not consistent—e.g. A (U+E123)— etc.

4. Removing characters

If you have template pages for characters, or glyphs, that you no longer need, you can easily remove the underlying page(s) using either the Pages panel, or IndyFont.

► Using the Pages panel:


- 1) Select the page(s).
- 2) Click the Trash, or select “Delete Spread” from the context menu.

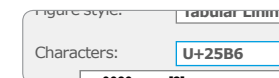
► Using IndyFont:

- 1) Run IndyFont and select the unwanted item(s) in the Glyph list. (Use Cmd/Ctrl Click to select multiple items.)
- 2) Press Delete or Backspace.
- 3) Click Update.

5. Specifying character codes and glyph names


Most characters can be entered on their own, or pasted, into the Characters field (see Section 2). Sometimes you may prefer to use a syntax that allows to either specify Unicode values or normalized glyph names.

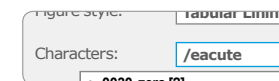
- 1) *Unicode values.* — Enter the character *code point* with a U+ prefix—for example, U+25B6—then click the  icon.



NOTE Here are the intervals of Unicode values that IndyFont supports:

U+0020 ▶ U+085F	U+10300 ▶ U+1034F	U+11180 ▶ U+111DF	U+1EE00 ▶ U+1EEFF
U+08A0 ▶ U+1AAAF	U+10380 ▶ U+103DF	U+11680 ▶ U+116CF	U+1F000 ▶ U+1F64F
U+1B00 ▶ U+1C7F	U+10400 ▶ U+104AF	U+12000 ▶ U+1247F	U+1F680 ▶ U+1F77F
U+1CC0 ▶ U+2FDF	U+10800 ▶ U+1085F	U+13000 ▶ U+1342F	U+20000 ▶ U+2A6DF
U+2FF0 ▶ U+A9DF	U+10900 ▶ U+1093F	U+16800 ▶ U+16A3F	U+2A700 ▶ U+2B81F
U+AA00 ▶ U+AB2F	U+10980 ▶ U+10A7F	U+16F00 ▶ U+16F9F	U+2F800 ▶ U+2FA1F
U+ABC0 ▶ U+D7FF	U+10B00 ▶ U+10B7F	U+1B000 ▶ U+1B0FF	U+E0000 ▶ U+E007F
U+E000 ▶ U+FFEF	U+10C00 ▶ U+10C4F	U+1D000 ▶ U+1D24F	U+E0100 ▶ U+E01EF
U+10000 ▶ U+101FF	U+10E60 ▶ U+10E7F	U+1D300 ▶ U+1D37F	U+F0000 ▶ U+FFFFD
U+10280 ▶ U+102DF	U+11000 ▶ U+1114F	U+1D400 ▶ U+1D7FF	U+100000 ▶ U+10FFFF

- 2) *Glyph names.* — Type the name of a glyph with a / (slash) prefix—for example, /eacute—then click the  icon.




NOTE IndyFont recognizes about 3,700 standard names. They all come from the Adobe Glyph List (AGL) which can be found at <http://sourceforge.net/adobe/aglfn/wiki/Home/>. In addition, you can specify custom glyph names—for example, /myLogo—which then are automatically assigned to special Unicode values in the “Private Use Area”.

Do not omit the leading slash (/) when you specify a glyph name. The string eacute would instead declare a set of characters (a, c, e, t, u).

Manage your Character Set




6. Adding a set of characters

You can easily enter a *set* of characters by just typing them in arbitrary order, then click the  icon:

aeiouy

NOTE If you found you made a mistake in entering the characters you want, don't worry; you can always remove any unwanted entry. See Section 4, above.

You can also copy text from an InDesign document or anywhere else and paste it into the Characters field, then click the  icon. Every single character will be added in the Glyph list.

NOTE IndyFont automatically discards double entries, because every glyph may only appear *once* in each font.

If your set needs to include Unicode values and/or glyph names, separate these elements with a space:

a /eacute U+2022

7. Adding a range of characters

To add an entire range of characters, use the following syntax: *first..last*, where *first* and *last* refer to any valid specifier. There are 2 periods between the first and last element.

Here are some examples of valid ranges:

a..z

0..9

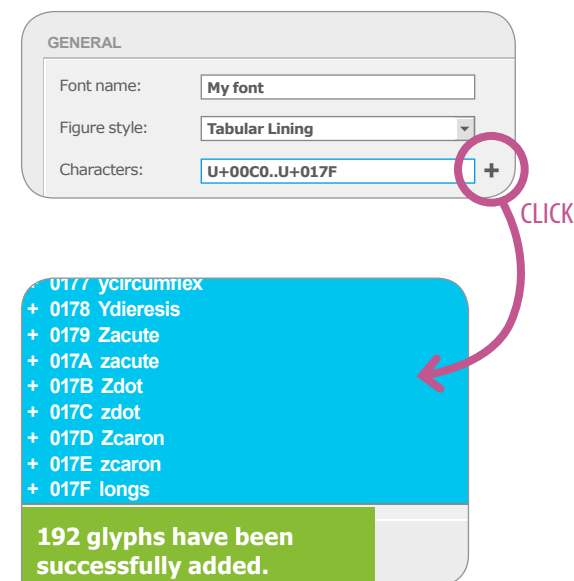
/eacute../ograve

U+00C0..U+017F

U+2020../bullet

Finally you can declare multiple ranges and/or elements in a single entry, using the space separator:

0..9 /underscore a..z A..Z



Adding a range of Unicode values in a single step.

8. Advanced syntax

► Alternates

Enter an element, followed by a period (.) and a suffix to add an OpenType-enabled variant of a glyph:

é.alt (or /eacute.alt)

This will also automatically add the character é on its own if not already present in the Glyph list.

Manage your Character Set



► Ligatures

To create a custom ligature, separate its characters with an underscore:

`P_h_é_n_o_m_!`

or

`P_h_eacute_n_o_m_exclam`

This will also automatically add the underlying characters, P, h, é, n, o, m, ! if not already present in the Glyph list.

As you can notice, *in the specific case of ligatures*, the leading slash is not required for glyph names. Alternatively, you can use Unicode values:

`U+221A_U+0032` (i.e. $\sqrt{2}$)

TIP The reserved symbols `_` (underscore), `.` (period), and `/` (slash) can be “escaped” with a `\` (backslash) prefix: `_ \. \/`

► Private Use Area

When you enter custom glyph names, such as `/myLogo`, IndyFont records them in a reserved zone, known as the “Private Use Area” (PUA). By default, an automatic Unicode value is assigned to any *private* character you declare. For this purpose, IndyFont uses the range `U+E000` ► `U+EFFF` (which is a subset of the full Unicode PUA). In most cases you won’t need to worry about that,

as IndyFont transparently manages these code points. However, you still can provide custom characters with *explicit* Unicode values, using the following syntax:

`/myLogo=U+E123`

9. Checking your characters

Various marks are used in the Glyph list so that you can check the current status of any character:

- ✚ The plus sign indicates items you have recently added in the Glyph list, but which have not yet a dedicated template page.
- An empty circle indicates other glyphs you have already declared in the template, but which have no valid outline.
- A filled circle indicates those glyphs which both have a valid template page *and* can be processed.

Each entry also indicates associated Unicode value(s), followed by the glyph name. Finally, all glyphs already present in the template have their page number in brackets.

For more on using alternates, ligatures, and OpenType-based features, see “Add typographic features.”

For more on Unicode and the Private Use Area, see “Additional notes.”

Characters:			
STATUS INDICATOR	UNICODE VALUE	GLYPH NAME	PAGE NUMBER (if any)
○	0030	zero	[2]
●	0031	one	[3]
○	0032	two	[4]
●	0033	three	[5]
✚	0066	f	
✚	0066 0069	f_i	
✚	0069	i	
✚	00E9	eacute	

Manage your Character Set

10. Declaring usual alternates for the selection

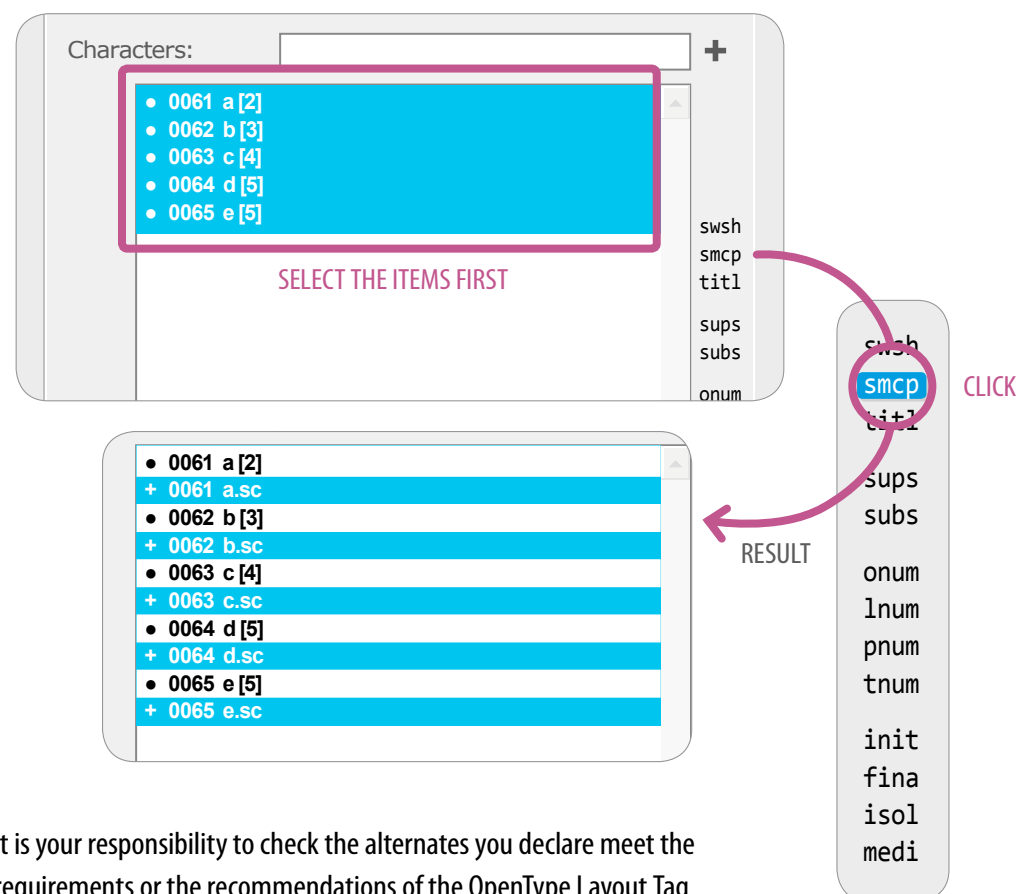
version 1.135 and higher

Suppose that you have a substantial amount of basic characters (e.g. the Latin lowercase letters) already created, and that you want to append “Small Caps” variants for all of them. Manually declaring those items through the Characters field would be tedious. Fortunately IndYFont now provides a set of special buttons that make it easy to apply usual OpenType features to the selected glyphs:

- 1) Select in the Glyph list the set or the range of items whose alternates have to be added. (Use Cmd/Ctrl or Shift Click to extend the current selection.)

- 2) Click the button having the desired OpenType feature tag:

- `swsh` → “Swash” (suffix: `.swash`),
- `smcp` → “Small Capitals” (suffix: `.sc`),
- `titl` → “Titling” (suffix: `.titling`),
- `sup` → “Superscript/Superior” (suffix: `.sup`),
- `sub` → “Subscript/Inferior” (suffix: `.sub`),
- `onum` → “Oldstyle Figures” (suffix: `.onum`),
- `lnum` → “Lining Figures” (suffix: `.lnum`),
- `pnum` → “Proportional Figures” (suffix: `.pnum`),
- `tnum` → “Tabular Figures” (suffix: `.tnum`),
- `init` → “Initial Forms” (suffix: `.init`),
- `fin` → “Terminal Forms” (suffix: `.final`),
- `isol` → “Isolated Forms” (suffix: `.isol`),
- `medi` → “Medial Forms” (suffix: `.medial`).



NOTE It is your responsibility to check the alternates you declare meet the requirements or the recommendations of the OpenType Layout Tag Registry: <http://microsoft.com/typography/otspec/featurelist.htm>

TIP Most tags can even be combined, as we shall see in the chapter “Add typographic features.”

Fashion your glyphs

On the distinction between “Character” and “Glyph”, see note page 7.

1. IndyFont layers

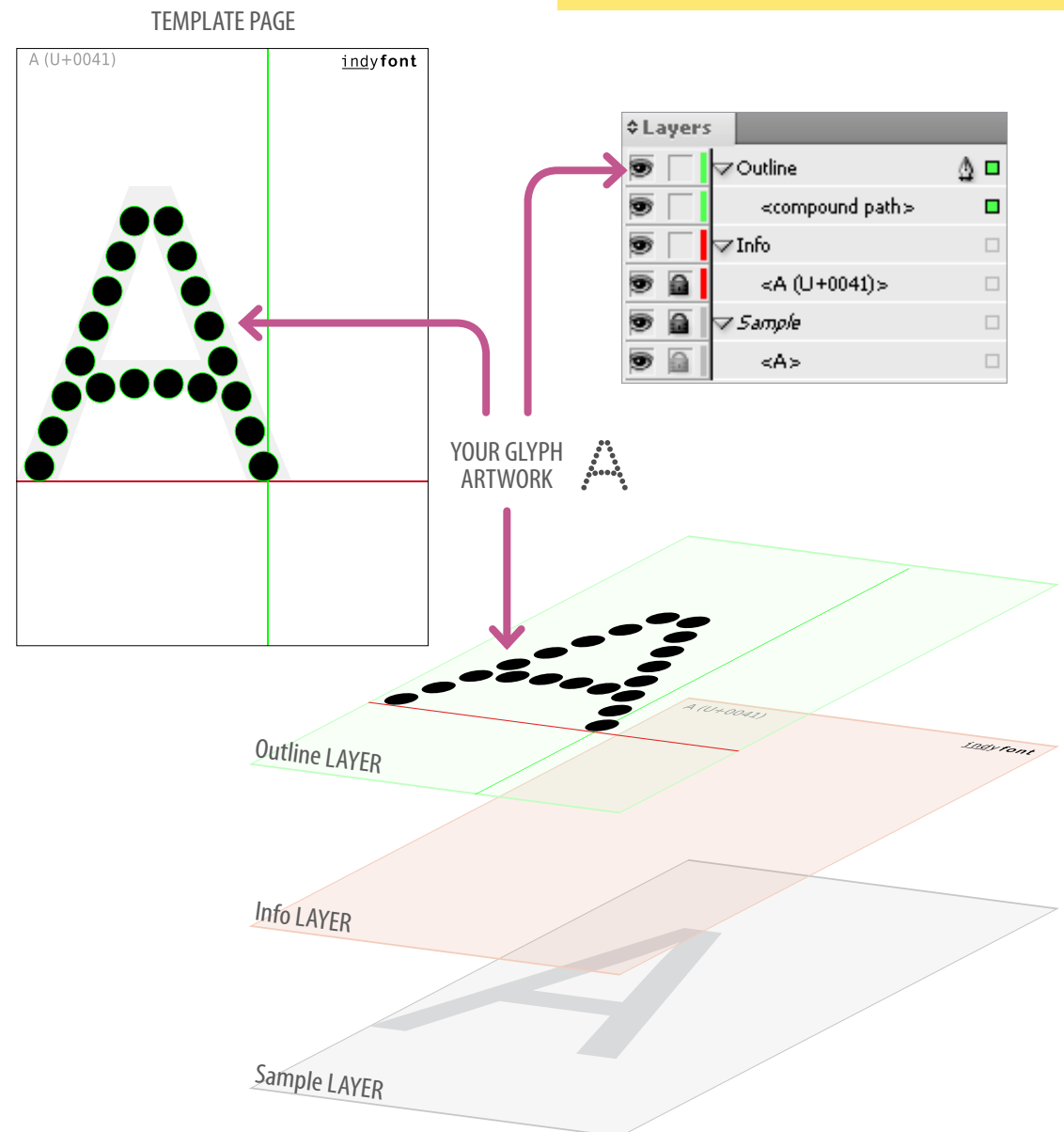
Any IndyFont template is based on three built-in layers, “Outline”, “Info” and “Sample”. When a new template is produced, both the layers “Info” and “Sample” are locked. Unless you need to manually edit a Glyph header (see “Manage your character set,” Section 3), or to tweak options in a Sample frame, we recommend you leave these layers in their original state.

NOTE IndyFont will not protest if you change the name of its native layers, provided that it can recognize them as its own creation. Never delete a native layer with the expectation of replacing it later. Your template will be broken. (“Sample” is the only optional layer—in case you don’t need samples anymore.)

You may create additional layers to your liking. However, the layer “Outline” is the unique place where you will set up artworks to be actually processed. During scanning and calculation, IndyFont only considers what is drawn on that layer.

2. Design space and units

IndyFont converts your vector drawing to an OpenType CFF (PostScript® flavor) font. The reason for this choice is that InDesign paths and PostScript-based types use the same outline format, cubic Bézier splines. The design grid is fixed to 1,000 units; that is, one em-space is 1,000 units wide. IndyFont uses InDesign’s points as its design unit.



Fashion your glyphs

The point coordinates of your design are rounded to whole points, including any curve control points. This means that there may be a small difference between your drawing and the generated font—but, typically, this should be on a scale of your font size divided by 1,000.

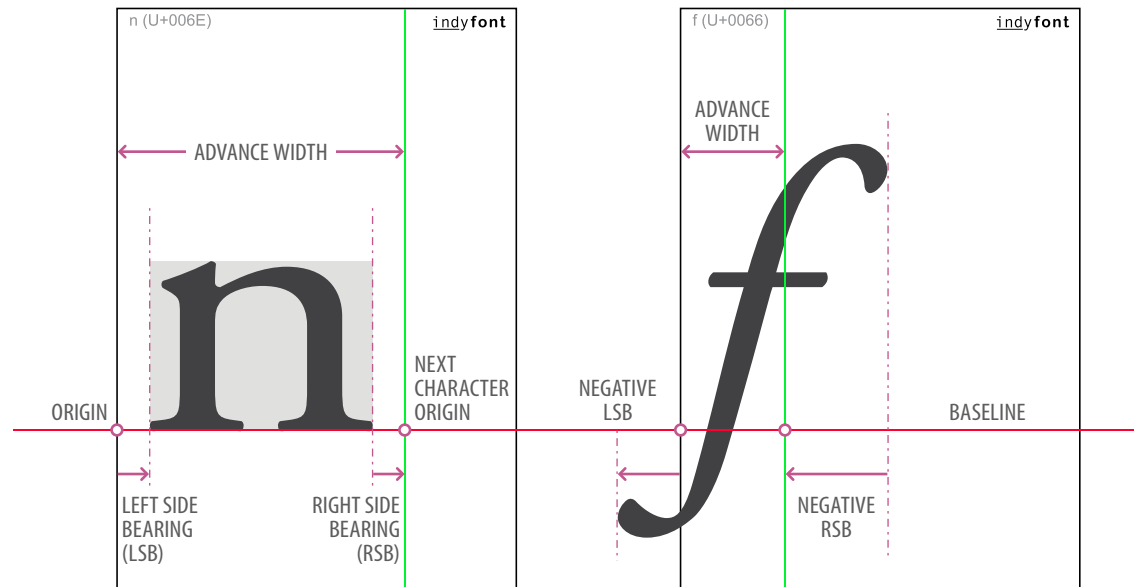
NOTE “Type 1 font programs generally use a 1000 to 1 scaling matrix for the definition of the relationship of character space units to user space units. [...] This allows character space coordinates to be expressed in integer values without significant loss of precision for most font designs.”—*Adobe Type 1 Font Format*, page 25.

However, if you want to ensure an exact match between your design and the final font:

- Choose View ► Grids & Guides ► Show Document Grid, and View ► Grids & Guides ► Snap to Document Grid.
- Make sure all of your points are snapped to this grid.

3. Glyph metrics

IndyFont defines the *origin* of a character as the intersection point between the left edge of the template page and the baseline (red horizontal guide). As you can see in the figure, the bounding box of the glyph is not aligned with the left of the page. This is because types need a little white space at their left and right sides (*side bearing*). If you put your image at the very left edge, your glyphs will run together when you type them **LIKETHIS**, and (usually) you do not want that.



- We call *left side bearing* (LSB) the horizontal distance from the origin to the left edge of the bounding box. IndyFont simply infers the LSB from the horizontal location of your artwork.
- Symmetrically, the *right side bearing* (RSB) refers to the horizontal distance from the right edge of the bounding box to the origin of the next character. In other words, this value represents the space you want to have at the right hand side of your character. In IndyFont, the RSB is indicated by a green vertical guide that you can adjust to your needs (see next page).
- Both the LSB, the bounding box width, and the RSB determine the *advance width* of the character. Visually, this is just the distance from the left edge to the green guide.

The italic *f*, above, illustrates the case of negative side bearings. Indeed, to avoid *ungraceful* letterspacing, we may need to make the advance width significantly lower than the actual width of the glyph bounding box. (Incidentally, note that IndyFont does not object to finding outlines beyond the page boundaries.)

- The red guide line at the bottom indicates the *baseline*. A character such as a capital “A” should be sitting on top of this, and a lowercase character such as a “g” should descend below it. A bullet typically floats a bit above it.

NOTE The baseline guide remains on the Glyphs master and is locked, meaning that it equally applies to every template page and is not supposed to be moved. Consider it a visual reference for your typeface.

4. Refining the advance width

Whenever IndyFont generates a template page, it seeks to position the green guide (RSB) at the best distance, with respect to the regular spacing of the sample character. If you are not satisfied with the result—or if IndyFont is wrong!—it is up to you to adjust the guide so that it fits the desired width.

- 1) Make sure your glyph is properly positioned relative to the left edge of the page (including the LSB shift).
- 2) Select and drag the green vertical guide to the desired x-location. A reworked bullet shape, for example, may require a wider space at the right side, as shown below.

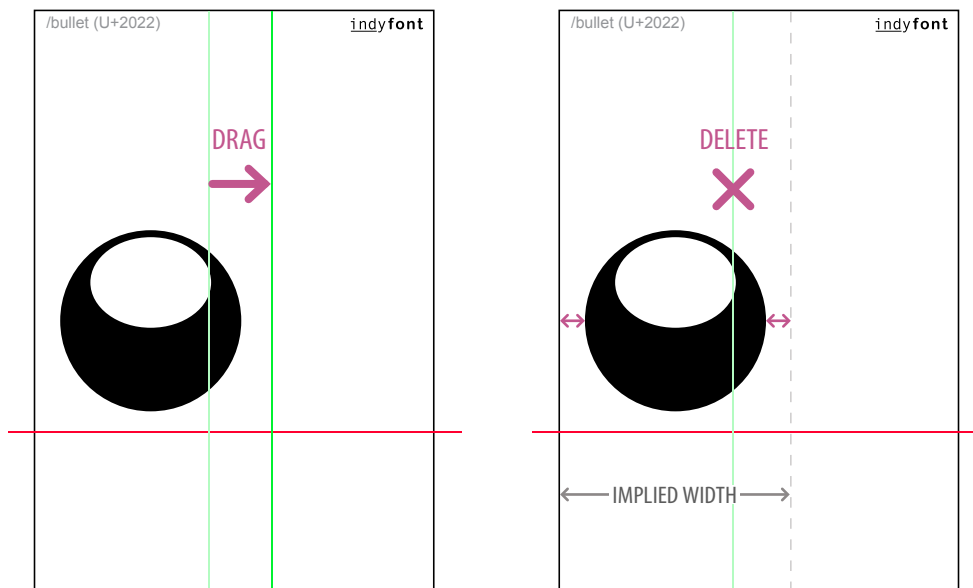
TIP If you remove the green guide, the space at the left side will be automatically mirrored at the right side. In other words IndyFont will assume $RSB = LSB$.

5. Drawing tools

While InDesign is in no way the prime tool for “vector artists” (!) it provides all the functionality required to make clean and pretty drawings if you are skillful enough.

Since type outlines are made up of points and curves, any of InDesign’s illustration feature intended to create, transform, or combine shapes—in the sense of “spline items”—can be used to elaborate the building blocks of your typeface.

Whatever method you use, the key rule is to end up with a set of vector artworks formed of fully closed paths.



Fashion your glyphs

You can either:

- Use shape tools to create rectangles, ovals, polygons, starbursts, etc.
- Use the Pen tool—or even the Pencil tool—to draw free-form shapes.
- Convert existing characters to outlines via Type ► Create outlines.
- Paste vector objects from Illustrator...

And so on!

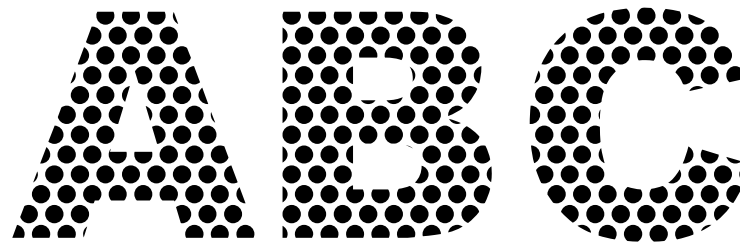
Then, of course, you can remix and fine-tune these spline items from both the Transform panel, the Align panel, the Pathfinder, etc.

6. About pathpoint complexity

There is an upper limit to the number of single points you can use in a single glyph. This runs into the tens of thousands of points, so it's not likely you run into it by accident. However, if your design contains extremely complex characters, and these show up as blanks in the Glyphs panel while others work as expected, you may have exceeded the total point limit. In that case you will have to simplify the design of the failing characters.

This is not as limiting as you may think; each of the characters below contain about 100 circles, and each circle in turn is defined by 4 “corner” points and an additional 8 Bézier curve control

points, which comes down to about 1,200 points per character. As you can see, that's no problem at all.

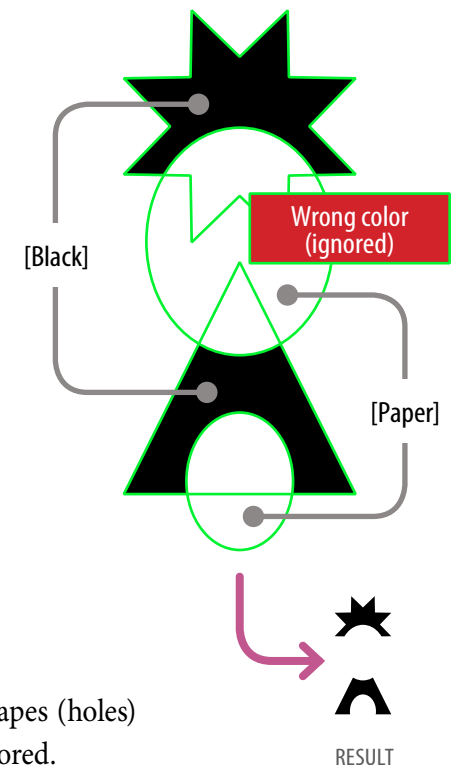


7. The “[Black] & [Paper]” rule

When IndyFont is in the process of scanning outlines to generate your typeface, it *only* inspects [Black] and [Paper] filled vector shapes present on the layer “Outline”.

- 1) Apply the regular [Black] swatch—not fake, mixed, or any custom black color—to all *positive* (inked) regions of the glyph.
- 2) Apply the regular [Paper] swatch—not fake, mixed, or any custom white color—to all *negative* (hole) regions that need to be subtracted.

IndyFont takes into consideration how shapes are stacked in your layout. Therefore, [Paper] filled shapes (holes) that do not overlap into any [Black] area will be ignored.

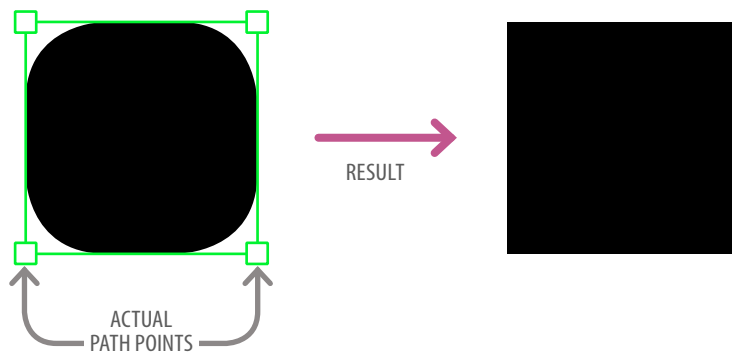


8. The “Path, not Stroke” rule

IndyFont disregards strokes. That is, it simply ignores *any* InDesign’s Stroke attribute, color, weight, type, corner options, etc., that might be applied to your shapes.

This rule has important consequences:

- Any straight line—whatever its weight, or type—is ignored. The reason for this is that the underlying path has no area.
- Dashed and dotted borders have no effect (as well as any stroke style).
- Rounded corners are ignored too. IndyFont only considers the *actual* path points of which the shape is made up:



TIP A popular trick can be used to convert rounded corners into *true* curved paths:

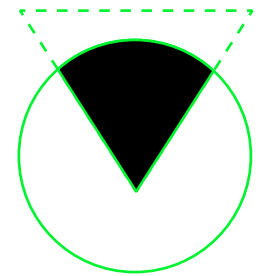
1. Select the object.
2. Choose Object ► Path ► Open Path.
3. Choose Object ► Path ► Close Path.

9. Combining shapes

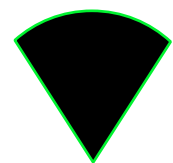
Each template page can contain as many building blocks as you need, and you do not have to group them before you export the font. IndyFont automatically detects all valid spline items (excluding anchored objects). It analyzes how they stack and/or nest, then produces outline data to be saved in the final font file.

IndyFont supports individual shapes, groups, as well as “nested shapes”—that is, single shapes *pasted into* a parent shape. Any shape can be based on either simple or compound paths. Use the Pathfinder to merge (add), subtract, intersect, or combine paths the way you want.

NOTE IndyFont may ignore complex nesting structures though. To improve quality and performance, we recommend you use finished paths rather than nested shapes. Also, as your font template grows up, you may notice that processing time increases with the number of individual objects and/or groups that are stacked in page templates. You can significantly reduce execution time by pre-converting those elements into single vector shapes.



In the figure, above, the [Black] filled triangle has been “pasted into” a [Paper] filled circle. Although IndyFont can deal with such nested structure, we recommend you use Pathfinder intersection instead:



10. Preventing overlapping issues

Before your vector shapes are converted into a font, IndyFont has to join all shapes into one, clean up stray points and loose ends, and manage overlapping areas.

Fashion your glyphs

An important rule in PostScript® font format is, *a single closed outline should not overlap itself*. Although InDesign does not have a problem drawing an object such as on the left, font drawing software interprets the overlapping part as a hole (right).



To avoid this, remove the overlapping region or split the single vector object into two.

11. Design guidelines (summary)

In summary, keep the following guidelines in mind as you design glyph shapes:

- Always have your final artwork on the “Outline” layer.
- Always use [Black] and [Paper] filled shapes.
- Use the rectangle tool to make filled shapes for straight lines, and remember that strokes are ignored as well as corner options.
- Close your paths.
- Use Pathfinder operations rather than nested shapes.
- Prevent individual shapes from self-overlapping.

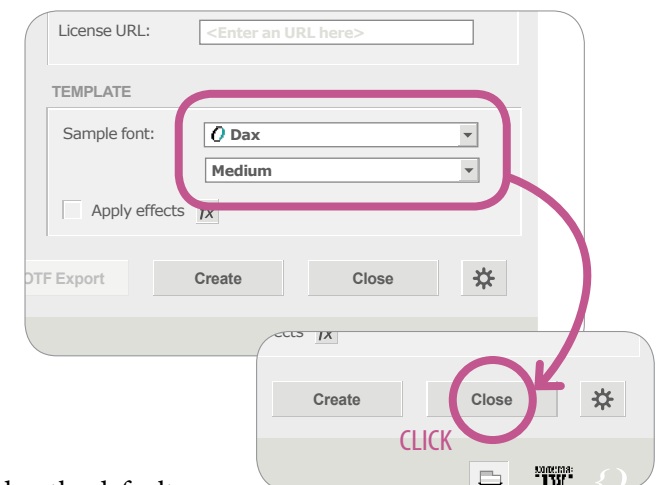
12. Changing the default Sample font

One of the most interesting features in IndyFont is font sampling. Any font available in InDesign can be used as a sample font for your template pages. When you create or update your document, light gray sample glyphs are automatically positioned in the background (on the layer “Sample”).

The very first time IndyFont starts up the Sample font is set to “Arial Regular”. If you want another font to be automatically selected as you run the plug-in, proceed as follows:

- 1) Close any InDesign document.
- 2) Run IndyFont.
- 3) In the TEMPLATE panel, click the Sample font dropdown list, then select the desired font family.
- 4) If needed, refine the font style from the second dropdown list.
- 5) Click the Close button.

Changing the default Sample font to “Dax Medium”...



The new Sample font is now saved as the default.

Fashion your glyphs

13. Changing the Sample font (for the current template)

You can also change the Sample font as you are creating, or updating a font template. In this case the new font is affected only to the current document.

- 1) In InDesign, activate your template document—if already created—, then run IndyFont.
- 2) In the **TEMPLATE** panel, click the Sample font dropdown list, then select the desired font family.
- 3) If needed, refine the font style (using the second dropdown list).
- 4) Click the Update (or Create) button.

IndyFont attempts to find a suitable sample for all characters declared in the Glyph list (see “Manage your character set”). If it cannot find some in your preferred sample font, it tries to find it in “Arial Unicode MS”. If it cannot find it there either, you will get a question mark “?” as sample.

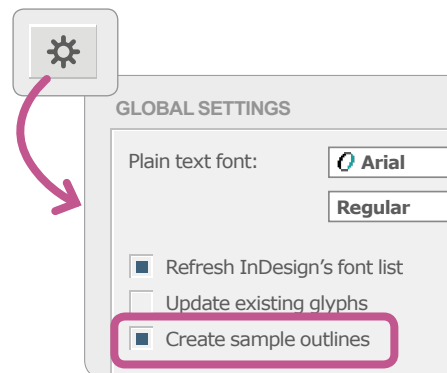
NOTE By default, as IndyFont applies a new Sample font to an existing font template (Update button), it only rewrites template pages that *do not have artwork yet* (so that existing glyphs are preserved). If you want to entirely re-sample the template, including glyphs you have already drawn, enable the option “Update existing glyphs” in the GLOBAL SETTINGS panel.

14. Generating sample outlines

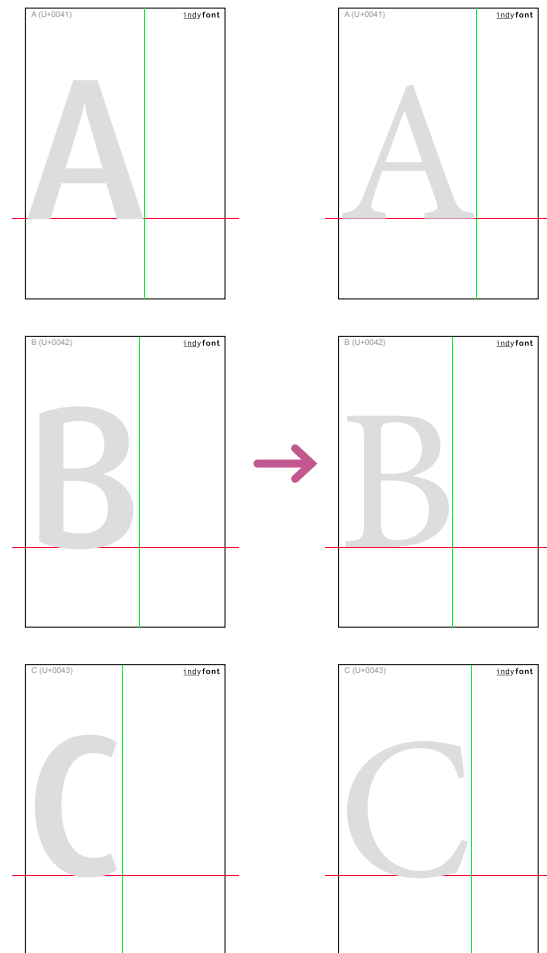
By default, sample glyphs are only placed on the layer “Sample”. These are visual placeholders to help you work on your own design.

In addition, IndyFont allows you to get already-generated sample outlines. This way you can use any sample glyph as a starting point, then transform and/or re-work it to your needs.

- 1) Run IndyFont.
- 2) Click the gear icon at the bottom right.
- 3) In the **GLOBAL SETTINGS** panel, turn on the option “Create sample outlines”, then click OK.



Changing “Dax Medium” into “Minion Pro Regular”.



Fashion your glyphs

4) Going back to the main dialog, click Update (or Create).

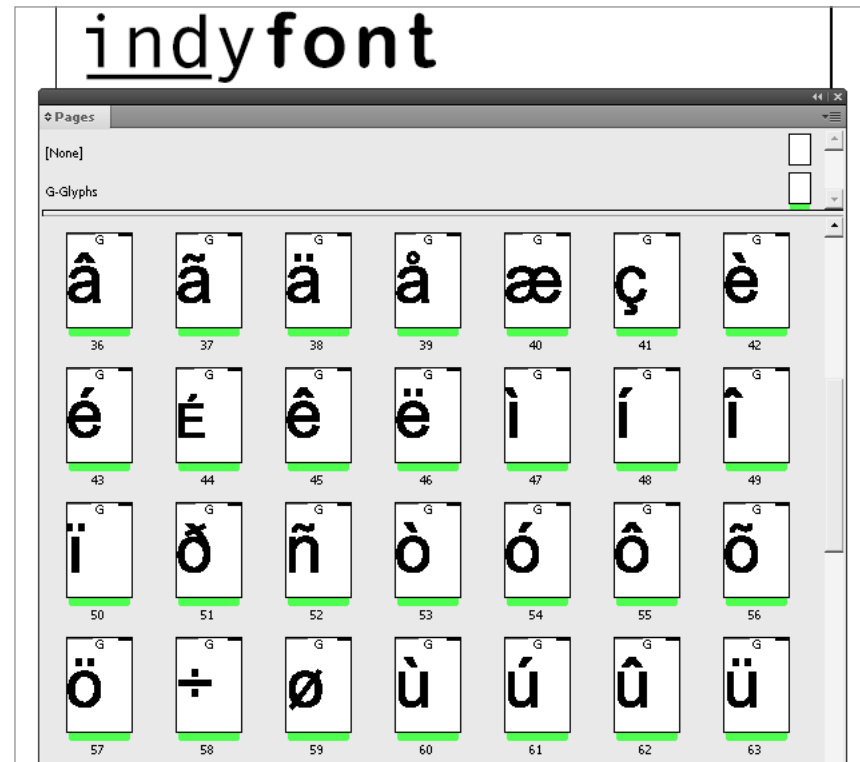
NOTE As the option “Create sample outlines” is a global preference, IndyFont will remember that you expect to have sample outlines on any font template being created/updated.

15. Organizing template pages

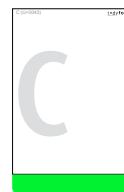
There is no particular requirement about how pages should be ordered in the font template. When you declare new glyphs, IndyFont always adds the corresponding template pages at the end of the document, but you can rearrange them to your liking. (Make sure that the Information page remains in first position though.)

NOTE Glyph order, in the final font, *will not reflect* template page order. IndyFont automatically sorts GIDs—glyph indices—by increasing Unicode values.

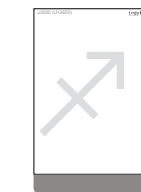
To easily navigate in your font template you can enable “Show Thumbnails” in the Pages panel’s contextual menu (Pages panel ► Panel options...) and choose an appropriate size. In the screenshot, below, we also have disabled “Show vertically” so the Pages panel looks like the regular Glyphs panel!



In InDesign CS5 and later, IndyFont applies green “color labels” to template pages for which a sample glyph has been properly created.



SAMPLE
FONT
IN USE



ARIAL
UNICODE
SUBSTITUTE



NO SAMPLE
GLYPH
AVAILABLE

A gray label may be used instead, indicating that the expected glyph has not been found in the Sample font and then has been rendered in Arial Unicode.

Finally, if the desired character is not found in Arial Unicode itself, IndyFont outputs a quotation mark, and applies a red label.

Fashion your glyphs

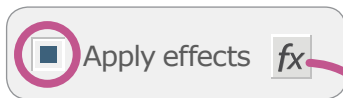
16. Applying special “effects” to your template

version 1.135 and higher

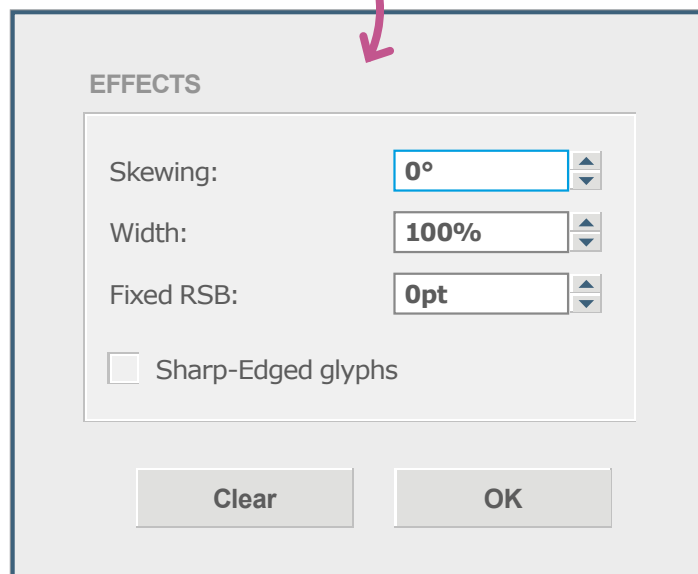
The TEMPLATE panel now provides a **fx** button which opens an EFFECTS window where you specify various transformations and settings to be applied to the glyphs (samples and/or outlines) from the template font.

NOTE Provided that the “Apply effects” checkbox is turned on, these *effects* are all processed targeting either new glyphs, or the whole font template if “Updated existing

APPLY THE EFFECTS

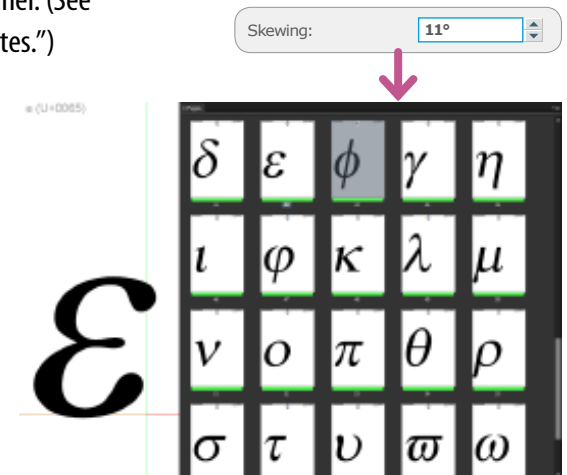


DEFINE THE EFFECTS

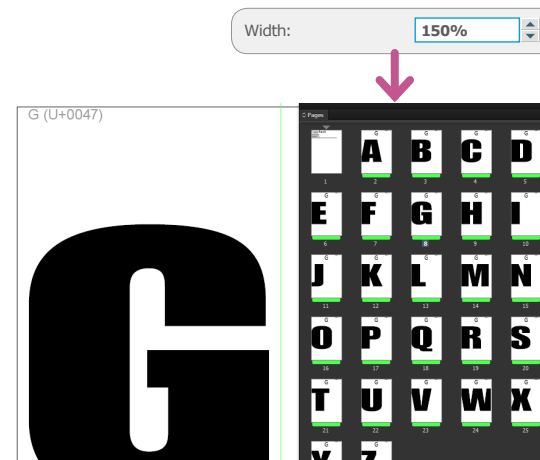


glyphs” is checked in the GLOBAL SETTINGS panel. (See the section “Global settings” in “Additional notes.”)

- *Skewing (“fake italics”)*
Enter a skew angle between -20 and 20° . An interesting usage of this feature is generating “faux italics” from a typeface that only offers roman glyphs. Skewing artworks around $10-12^\circ$ usually yields good results. (To remove this effect, simply reset the angle to 0° .)



- *Width (horizontal scaling)*
Going into the Width field, enter a value between 50 and 150%. The corresponding horizontal scaling transformation will be applied to the glyphs, so you can (roughly) sketch condensed or extended typefaces from any font. (To remove this effect, simply reset the value to 100%.)



11° skewing applied to Symbol (Regular). From there IndyFont can instantly produce and export a fresh “Symbol Italic” font!

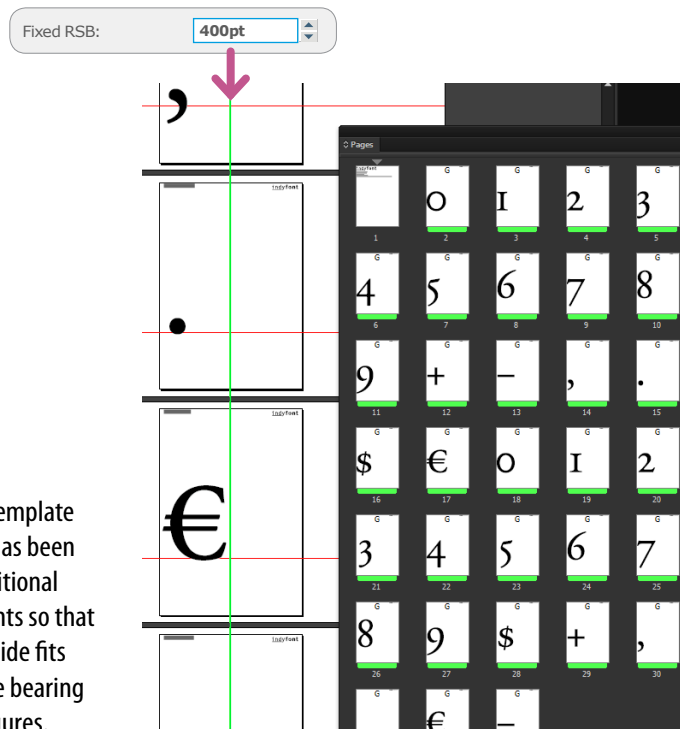
150% width applied to Impact (Regular).

Fashion your glyphs

► Fixed RSB (right side bearing)

The field “Fixed RSB” lets you control the x-location of the green vertical guide (which indicates the right side bearing). Just fill in a positive value—e.g. 400pt—and the targeted glyph template pages will have that fixed RSB guide, bypassing automatic positioning. This functionality is useful to create a set, or a subset, of “monospaced” glyphs. (To remove this effect, reset the value to 0pt.)

NOTE Fixed RSB can be filled in using any conventional unit (in, mm. ...). However, the value is automatically converted into integral points.

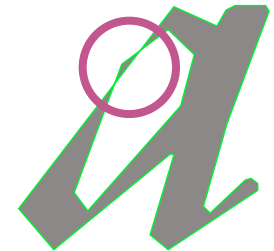


► Sharp-Edged glyphs (outlines only)

The option “Sharp-Edged glyphs” provides an experimental effect which consists in removing any curve from the artworks. Note that this option only alters outlines (so, make sure that “Create sample outlines” is checked in the GLOBAL SETTINGS panel). Here is a set of sharp-edged letters based on Adobe Caslon Pro Semibold Italic:

abcdefghijklmnopqrstuvwxyz

Note that some glyphs might then contain crossing paths that would benefit from being reworked before exporting the font.



Add typographic features



IndyFont can automatically generate a number of advanced OpenType features: swashes, small capitals, titling, number styles, superiors and inferiors, arbitrary alternates, ligatures...

You can only add OpenType features to regular glyphs that also exist in your font. You cannot, for example, add a Small Capital A to your font if you do not have a regular “a” as well. IndyFont automatically checks your font for missing glyphs, and adds these if necessary.

1. General alternate syntax

An *alternate* is a variant of some glyph with another design that a font user can manually select. All alternates appear in the InDesign Glyphs panel under their original character assignment, and are indicated by a small ► at the bottom right.

The general syntax to add an OpenType-enabled alternate is

base.tag

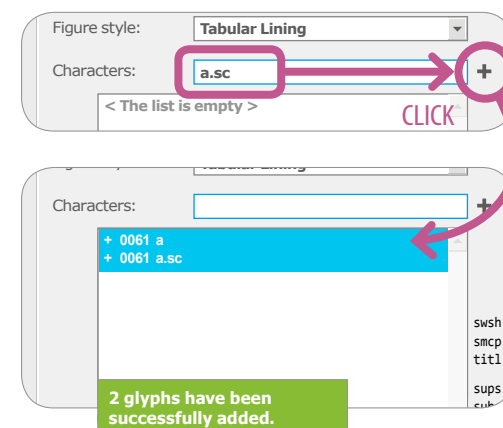
where *base* is the character itself or any valid specifier (glyph name, Unicode value, or ligature sequence as detailed in “Manage your character set”, pages 9-11) and *tag* is a code indicating the alternate, or feature name.

NOTE The *tag* element can only be formed of alphanumeric characters A-Z a-z 0-9.

To declare an alternate in IndyFont, simply enter the *base.tag* code in the Characters field, then click the + icon.

NOTE For more on feeding the Glyph list, see “Manage your character set”.

Use the tags described below to add standard features. All other feature names will be added as an arbitrary alternate glyph, and these will only be available in InDesign through the Glyphs panel.



Whenever you add an alternate, e.g. *a.sc*, the implied base glyph (*a*) is automatically inserted too, if not already present in the Glyph list.

2. Swashes (swsh)

This OpenType feature replaces default glyphs with corresponding swash variants. A swash is a graphic flourish, usually associated with an italic type, on a glyph.

To declare an automatic *swash* variant, add the suffix *.swsh* (or *.swash*), for example:

A.swsh ; /Eacute.swsh

If you enable “Swashes” from the OpenType ► Swash menu, you will see these characters replaced by their variant. Other characters without a swashed alternate will stay unchanged.

A É

A.swsh and
É.swsh in Adobe
Caslon Pro Italic.

Add typographic features

3. Small capitals (smcp)

This OpenType feature turns lowercase characters into small capitals. It only applies to scripts with case differences—i.e. bicameral scripts—, such as Latin, Greek, Cyrillic...

To declare OpenType small capitals, add either the suffix `.sc`, `.smcp`, or `.smallcap`. For example:

`a.sc` ; `/eacute.smcp`

You may also include small caps for non-letters, such as oldstyle figures (see Section 4, below). For this purpose, you can combine the tags `“sc”` and `“onum”` using the following syntax:

`2.onumSc` or `2.scOnum`

NOTE As a general rule, tags can be combined as you wish, subject that each added tag begins with an uppercase letter (`Swsh`, `Sc`, etc.). Thus, you could also mix smallcap and swash features this way: `a.swshSc`, or `a.scSwsh`.

4. Understanding figure styles

You can automatically add the codes for *oldstyle* and *proportional* digits to your font. While *lining* figures are all the same height and sitting on the baseline, *oldstyle* figures are the equivalent of lowercase letters: their height is based on lowercase text,

some digits hang out of the bottom of the baseline, and some digits are higher, as in ascending lowercase letters. These, for example, are the oldstyle numerals of Minion Pro:

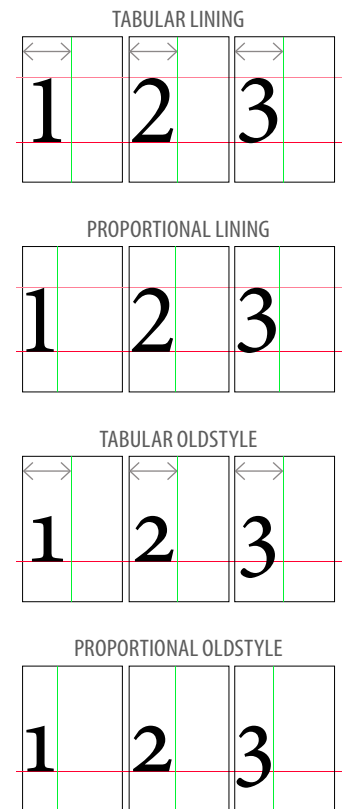
0 1 2 3 4 5 6 7 8 9 x p f

Usually all digits in a font have the same width, so they line up nicely when used in a table. Therefore, they are commonly called “tabular figures”. This is a disadvantage: you need to make sure the “1” is as wide as the other digits, and in some fonts that makes the space between two one’s too large. In “proportional” figures, the “1” has less space at its left and right hand side (some other digits may be narrower or wider than usual as well). Proportional figures look good in plain text, but of course they do not look very well in vertically aligned columns. A designer may therefore want to select proportional figures in body text for better readability, and only use tabular figures in tables.

In summary, there are four types of digit styles: tabular vs. proportional, and lining vs. oldstyle. This leads to four combinations:

- Tabular Lining (*usually the default*)
- Proportional Lining
- Tabular Oldstyle
- Proportional Oldstyle

One will be the “default” style for your font.

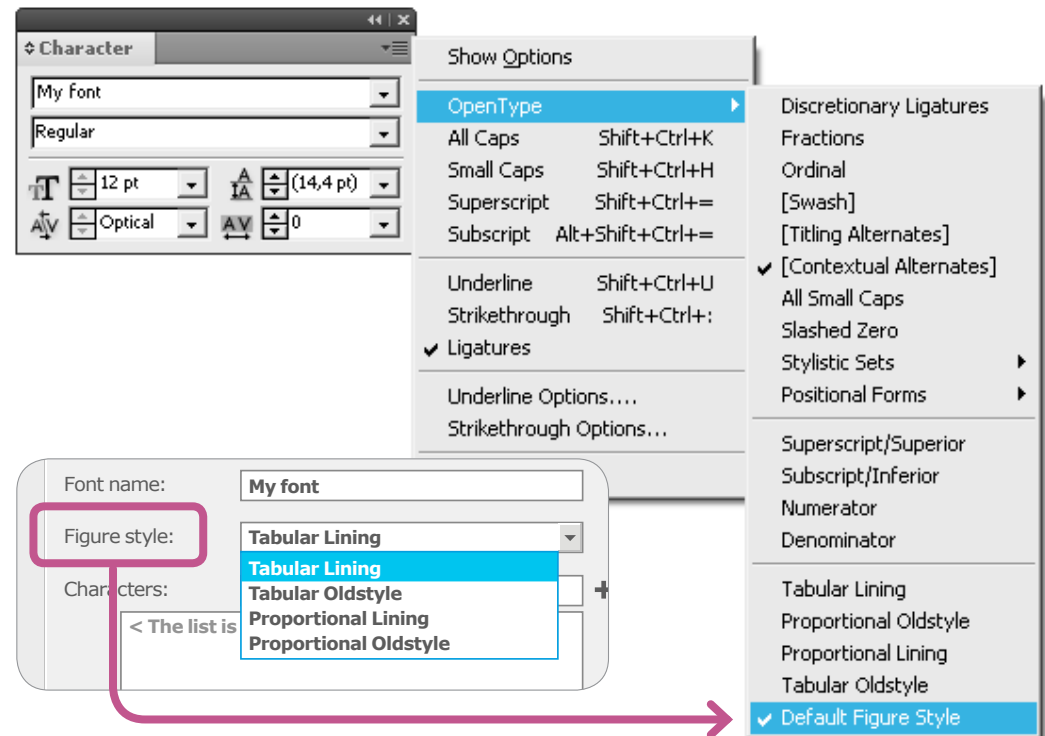


Add typographic features

5. Setting figure style and variants

To give your users a choice of figure style, do the following:

- 1) Run IndyFont.
- 2) Make sure the dropdown list “Figure Style” in the main dialog is set to the actual style of your *default* digits. That is, if your default style is e.g. “Proportional Oldstyle”, select it from the list.
- 3) Add regular digits **zero** to **nine** (i.e., the range 0..9) to the Glyph list. These are the default digits, and will be used if no special number style is selected.
- 4) To add *alternate* digits—i.e. glyphs that you want to make different from the default style—use the following tags:
 - .pnum (or .fitted) for *proportional* variants, e.g. 1.pnum or /one.fitted
 - .onum (or .oldstyle) for *oldstyle* variants, e.g. 2.onum or /two.oldstyle
 - .tnum (or .tab) for *tabular* variants, e.g. 3.tnum or /three.tab
 - .lnum (or .lining) for *lining* variants, e.g. 4.lnum or /four.lining

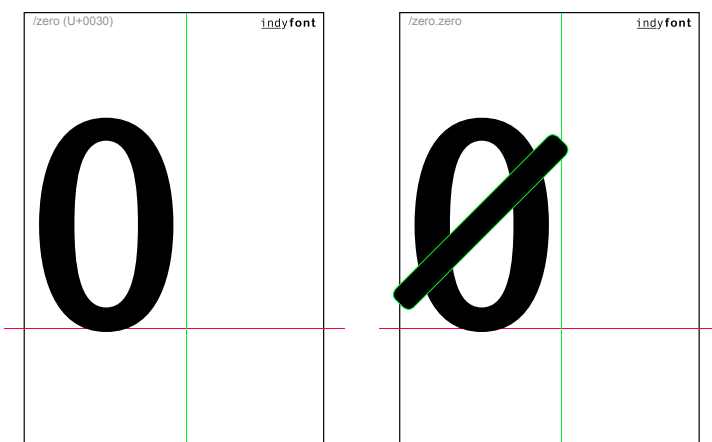


- or any relevant combination, using a leading uppercase letter for the second tag, e.g. 5.pnumOnum

The IndyFont “Figure style” option determines the default style for your digits.

NOTE A tag—or a combination of tags—is relevant if it actually leads to a variant relative to the default figure style. For example, /one.tnum does not make sense by itself if the default style is Tabular Lining. By contrast, both /one.pnum, /one.onum, and /one.pnumOnum would be relevant in that context.

Add typographic features



Drawing a slashed zero in its own template page.

6. Slashed zero (zero)

In some circumstances a Slashed zero is preferred over a regular zero; for instance, when it may not be mistaken for a capital “O”. You can draw your standard zero with a slash; but with this feature you can give your users a choice!

Add the suffix `.zero` (or `.slashed`) to the zero character, e.g.

`0.zero` ; `0.onumZero`

and then you can select “Slashed Zero” from the OpenType menu.

NOTE This feature must only be used for the zero character itself.

7. Superscript and Subscript (sups / subs)

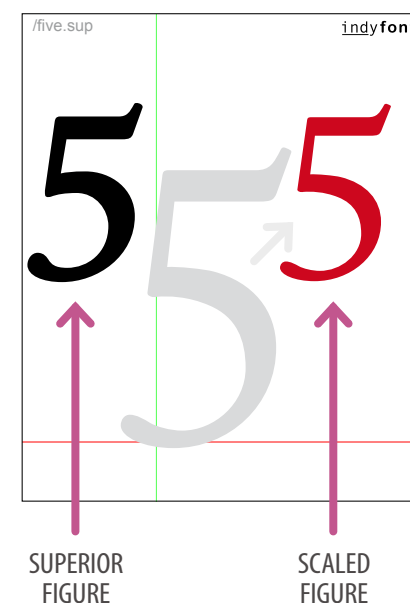
version 1.135 and higher

► *Superscript (superior glyphs)*

Superior figures, letters, or symbols used in footnotes or abbreviations (S^{tr}, n^o, M^{lle}) may need dedicated glyphs for reasons of legibility. Add the suffix `.sup` (or `.superior`) to declare these variants, e.g.

`5.sup` ; `$.superior`

NOTE The OpenType specification is quite vague about the distinction between the `sups` feature and other related tags (`ordn`, `numr`, `dnom`) which IndyFont does not implement. In addition, Unicode provides various superscript elements for digits (U+00B9, U+00B2...), letters or ordinal symbols (U+207F, U+00BA...), leading to proliferating ways of implementing superior forms.



► *Subscript (inferior glyphs)*

Add the suffix `.sub` (or `.inferior`) to declare subscript/inferior variants, e.g.

`a.sub` ; `/Euro.inferior`

Add typographic features

8. Titling (titl)

version 1.135 and higher

The *titling* feature allows to replace default glyphs with corresponding forms specially adjusted for viewing at larger sizes. Titling alternates are often defined for capitals but can apply to any other character. Add the suffix `.titl` (or `.titling`) to declare those glyphs, e.g.

A.titl ; €.titling

they also have interesting usages in cursive typefaces with connected glyphs.

► Initial forms (init)

Replace glyphs at the beginnings of words with alternate forms. Add the suffix `.init` to declare those variants, e.g.

É.init ; a.init

► Final forms (fina)

Replace glyphs at the ends of words with alternate forms. Add the suffix `.final` (or just `.fina`) to declare those variants, e.g.

s.final ; x.fina

► Isolated forms (isol)

Replace “the nominal form of glyphs with their isolated forms” (*OpenType Layout tag registry*.) Add the suffix `.isol` to declare those variants, e.g.

e.isol ; I.isol

► Medial forms (medi)

Replace glyphs “in the middles” of words with alternate forms. Add the suffix `.medial` (or just `.medi`) to declare those variants, e.g.

t.medial ; y.medi

9. Positional forms

version 1.135 and higher

(init / fina / isol / medi)

s.init



Positional forms work a bit like ligatures (Section 11), or swashes (Section 2), but they are triggered with respect to the *position* of the character within words. These features are required in various non-Latin sets—like Arabic—but

s.medial



s.final



POSITIONAL FORMS: DEACTIVATED

that is so messy

POSITIONAL FORMS: AUTOMATIC

that is so messy

Add typographic features

10. Arbitrary alternates

To create an arbitrary alternate—that is, a glyph variant that does not *implement* a standard OpenType feature—we recommend you use the `.alt` suffix:

`/bullet.alt`

will create a single alternate bullet only available, in InDesign, from the Glyphs panel.

You can create more than one alternate for each character:

`/bullet.alt1`

adds a second, and so on.

There is nothing special about the `.alt` suffix itself: the current version of IndyFont automatically creates an alternate form of the base character for all feature codes that it does not understand. However, because the list of supported feature codes may be expanded in future versions, it is safest to use the suffix `.alt` (with or without a following number) to guarantee that newer versions will create the same list of alternates.

11. Automatic ligatures

A *ligature* is a special glyph that should be used for a combination of two or more characters. In most fonts, the combination “f” and “i”, for example, would make the dot on top of the “i” (the “tittle”) clash into the top of the “f”. To prevent this, the type designer adds a ligature to the font. If ligatures are enabled, InDesign—as well as other OpenType aware software—will automatically replace all “fi” occurrences with this single glyph, “fi”.

f i ▶ fi ▶ fi

The “fi” ligature of
Adobe Caslon Pro.


Although it *embeds* Unicode characters, a ligature in itself does not have any assigned Unicode value. Instead, to “recognize” the text when you search in a PDF, for example, Acrobat relies on the glyph name.

A ligature glyph needs a special name format to be recognized: the names of its individual characters, separated by the underscore “_”. Thus, the “fi” ligature is internally named `f_i`.

To create a ligature in IndyFont, proceed as follows:

For more on using advanced syntax for both ligatures and alternates, see “Manage your character set”, Section 8.

Add typographic features

- 1) Enter the full name of your ligature in the Characters field, and insert an underscore “_” between each element, then click the  icon:

I_n_d_y_F_o_n_t

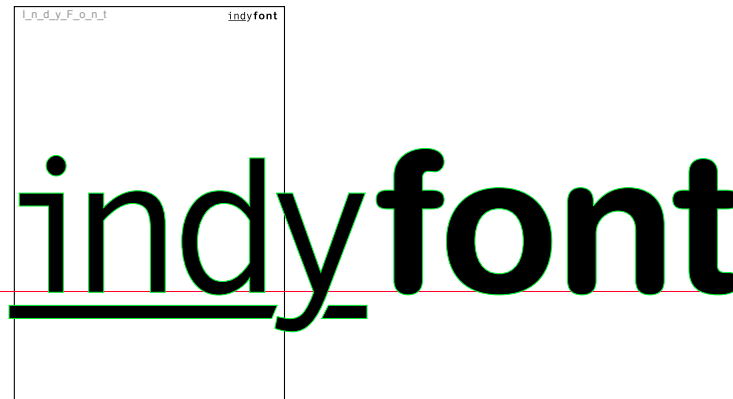
NOTE You can only add ligatures for glyphs that also appear in your font on their own. IndyFont knows the individual glyphs are needed as well, so it automatically will create templates for each of them. In this example, templates will be created for “I”, “F”, “d”, “n”, “o”, “t”, and “y”, as well as a single “I_n_d_y_f_o_n_t” page.

- 2) Add designs for the individual glyphs, so you can see what you are typing before the ligature itself is applied.
- 3) Draw your custom ligature on the page where its full name appears.
- 4) Finally, re-run IndyFont and click the OTF Export button to generate the font. The result:

indyfont ▶ indyfont

The I_n_d_y_F_o_n_t ligature in our freshly generated font.

You can see the same “n” gets used twice at the left, but the ligature on the right is a single design and so you can change each any way you like.



Designing wide ligatures will sometimes lead you to draw beyond the page boundaries. This is not a problem for IndyFont.

You don’t have to create an entire font if you only need a custom ligature of your company name. You can draw a simple circle or rectangle for the individual characters, then use InDesign’s GREP Styles to locate the company name and have it automatically change the font to yours.

NOTE Unicode defines “precomposed” ligature codes for several common sequences, such as “fi” and “ffi”. These are *not* automatically used by InDesign with the “Ligatures” option. Also, if you copy – in InDesign, or from an exported PDF – or export your text, they are not automatically translated back to the original separate characters. It is advised to use only the underscored “f_i” format to create these ligatures.

Add typographic features



12. Space width

version 1.133 and higher

In version 1.133 and later IndyFont provides a new field, Space width, that allows you to manually edit the width of the regular space character—U+0020—in em units. Here you can enter an integer value between 1 and 750 (the default is 230).

NOTE The space width is automatically registered as the *advanceWidth* of the space character in the related OpenType font file tables (that is, HTMX and CFF glyph list).

A screenshot of the IndyFont software interface. The window is divided into two main sections: 'GENERAL' on the left and 'IDENTIFICATION' on the right. In the 'GENERAL' section, there are four fields: 'Font name:' with the text 'My font', 'Figure style:' with a dropdown menu showing 'Tabular Lining', 'Space width:' with a numeric input field containing '230', and 'Characters:' with an empty text box and a '+' button. The 'Space width:' field is highlighted with a red rectangular border. Below the 'Characters:' field, there is a text box containing '< The list is empty >'. The 'IDENTIFICATION' section on the right contains four fields: 'Version:', 'Copyright:', 'Description:', and 'Designer:'. Below these, there is a 'Designer URL:' field.

This feature is especially useful when you design a monospaced typeface. In such case you will set up the same *advance width* (vertical green guide) for your whole font template. Then make sure that the space width matches the *x*-coordinate of the guide.

Build your font

1. Setting the font name

Enter the full name of your font in the Font name field (e.g. “My font Pro Semibold Condensed”). This is the font family and style as it will appear in InDesign’s Type ► Font menu, as well as in any other font related field.

If you want to make a regular font (that is, not bold, italic, condensed or anything), you do not have to add “Regular” or “Normal” to the name; a “Regular” style is the default.

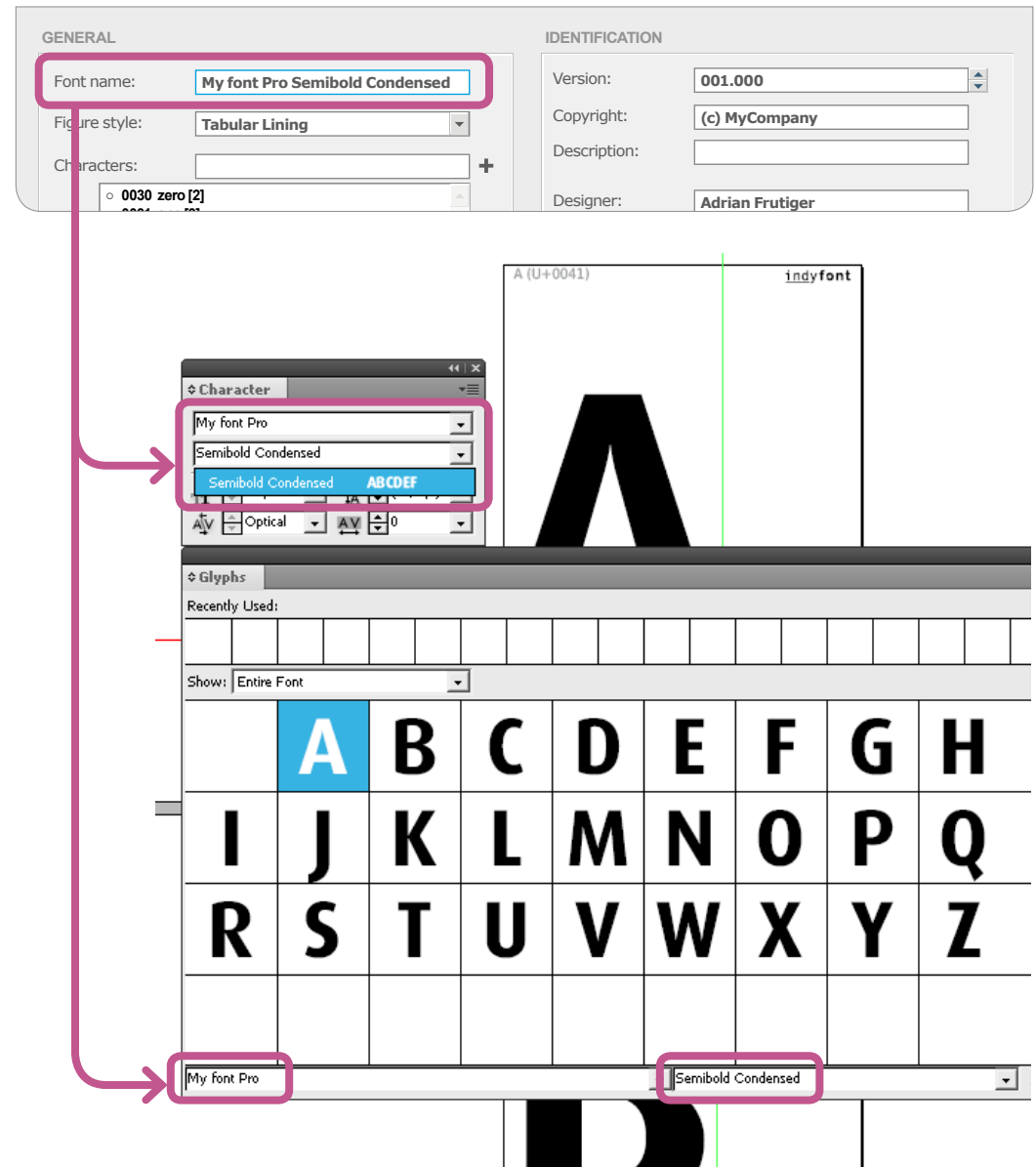
For an italic, bold, or bold italic style, add this to the name. For example, “Tip Italic” will create a font *family* named “Tip”, and the *style* of this particular font will be “Italic”.

You can also use more advanced style names, such as “SuperFont Bold Extended Oblique”, as described in the next section.

2. Style name guidelines

InDesign sorts fonts by their font family name, and inside each family, the font styles by weight, width, and slope (slant).

IndyFont automatically splits a given full font name into the font family name and the font style. For example, a font name “Really Light Condensed Italic” is split into a family “Really” and a style “Light Condensed Italic”.



Build your font

Words that are considered to be part of a style name are:

- “Light”, “Ultralight”, “Extralight”, “Thin”,
- “Regular”, “Normal”,
- “Bold”, “Demibold”, “Semibold”, “Medium”, “Heavy”, “Black”, “Extrablack”, “Ultrablack”,
- “Condensed”, “Narrow”, “Extended”, “Expanded”,
- “Italic”, “Slant”, “Slanted”, “Oblique”.

Any of these words may appear, with any case (uppercase, lowercase, or mixed), separated by a space or hyphen, and in any order. IndyFont then attempts to make sense of your entry and applies some automatic rules to help you manage your family in a consistent way.

► Weight

The following keywords define the weight for your font:

WEIGHT NAME	WEIGHT CLASS
Ultralight (<i>Thin</i>)	100
Extralight	200
Light	300
<empty string> (<i>Normal, Regular</i>)	400
Medium (<i>Book</i>)	500
Semibold (<i>Demibold</i>)	600
Bold	700
Extrabold (<i>Heavy</i>)	800
Black (<i>Extrablack, Ultrablack</i>)	900

► Width

The following keywords define the width for your font:

WIDTH NAME	WIDTH CLASS
UltraCondensed	1
ExtraCondensed	2
Condensed (<i>Narrow</i>)	3
SemiCondensed (<i>DemiCondensed</i>)	4
<empty string> (<i>Normal</i>)	5
SemiExtended (<i>SemiExpanded</i>)	6
Extended (<i>Expanded</i>)	7
ExtraExtended (<i>ExtraExpanded</i>)	8
UltraExtended (<i>UltraExpanded</i>)	9

► Slope

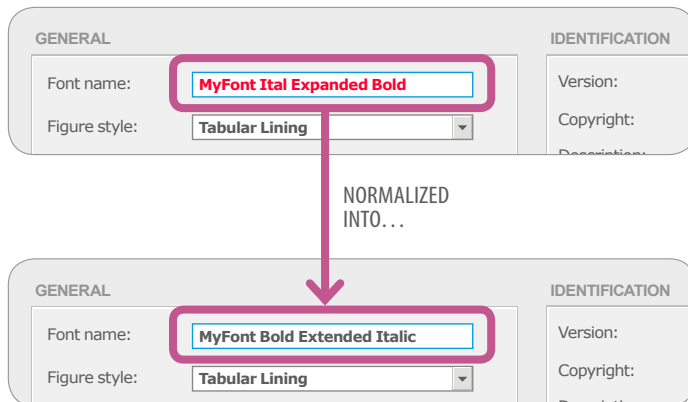
The following keywords define the slope for your font:

SLOPE NAME	SLOPE TYPE
Italic (<i>Ital</i>)	ITAL
<empty string> (<i>Normal, Regular</i>)	<i>none</i>
Oblique (<i>Slant, Slanted</i>)	OBL.

Gray words, in the three tables above, are always converted into the related Weight, or Width, or Slope name supported by IndyFont. Also, any combination you enter in the Font name field is re-arranged in “official” Weight-Width-Slope order.

NOTE IndyFont’s current version does not allow to bypass these naming rules, which you may consider quite restrictive. On the other hand, ►

Build your font



this prevents a number of inconsistencies and ensures that large font families will be properly addressed (especially in InDesign).

3. Additional identification

At the right hand side of the main dialog you can enter further information about your font. Some of this information is shown in InDesign if you choose “Find Font” and press the “More Info” button; more is shown in general font management software, such as FontBook (on the Mac) and Explorer (on Windows).

- 1) In the Version field you enter a regular version number. According to the OpenType specifications, this should be in the format “3 digits major version, 3 digits minor version”.

NOTE IndyFont does not automatically increment the version number. If necessary, you have to do so yourself.

- 2) In the Copyright field, you can put any copyright statement you like. The default string is taken from your user license name, but you may change it to anything you like.
- 3) The Description field is to include a short description of the font, possibly with some hints for its usage, a short history, or any special features.
- 4) Designer says “IndyFont” by default but of course you can put your own name in there!
- 5) In Designer URL you can insert a hyperlink (complete with “http://” prefix) where someone might find out more about the designer.
- 6) License Info should contain a short text describing how the font might be used legally. Microsoft recommends that “this field should be written in plain language, not legalese”.
- 7) An additional License URL can be specified where more information can be found on the license.

The IDENTIFICATION panel allows you to add further detail about your font.

The screenshot shows the 'IDENTIFICATION' panel of a font management application. It contains several text input fields and a dropdown menu. The 'Version' field has '002.003'. The 'Copyright' field has '(c) Jongware and all.'. The 'Description' field has 'Latin typeface with full diacritics'. The 'Designer' field has 'Theunis de Jong'. The 'Designer URL' field has 'http://www.jongware.com'. The 'License info' field has a text area with the text 'For information on the license of this font, please contact its manufacturer.'. The 'License URL' field has 'http://www.indiscripts.com'.

These strings are inserted as plain text on the first page in your new font document. None of these are required for your font to operate; you may leave them at their defaults, or even delete them.

Build your font

NOTE Identification strings all need to be compatible between Mac OS X and Microsoft Windows, and this means you cannot insert any and all characters. More on this topic can be found online in the OpenType Specifications by Microsoft: www.microsoft.com/typography/otspec/name.htm

4. Before you export

Two main conditions must be met in order to have the “OTF Export” button enabled:

- The Glyph list must be in sync with the current state of your template document—no pending entries prefixed with a [+] mark, etc. For more on this subject, see “Manage your character set”.
- IndyFont needs to have detected *at least one exportable glyph* among the template pages.

In both cases you may have to update the template (IndyFont ► Update button) before you can export.

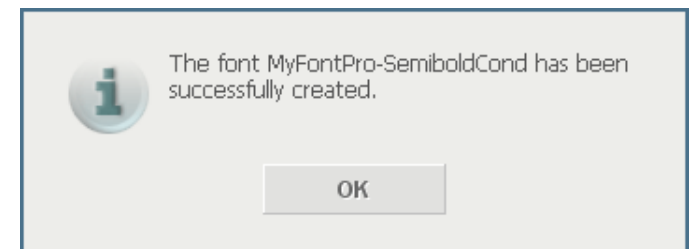
If IndyFont cannot find any artwork at all, this means that you did not draw anything on the “Outline” layer, or you did not use the [Black] swatch in any object, or you have no filled objects on any of your pages... Go back to “Fashion your glyphs” (in particular section 11, Design guidelines) and check your template.

5. Exporting to OTF

- 1) Run IndyFont.
- 2) Make sure the final font name and the IDENTIFICATION fields are properly set—as described in the previous pages.
- 3) Click the “OTF Export” button. IndyFont processes the glyph outlines, then invites you to...
- 4) Select the destination folder for your font file. You can save your font anywhere you want. However, that alone will not automatically make it available in InDesign.
 - If you save a font into the InDesign Fonts folder, it will only appear inside InDesign itself.
 - You can also save it into the Fonts folder in your local Adobe Common Files folder (this will enable the font for all of your Adobe software).
 - Lastly, you can install it into your system, so you can use it with all software that supports OpenType fonts.

NOTE On InDesign CS5 and newer, you can create a folder called “Document Fonts” in the folder where you save your regular documents. That way, the font is available for those documents *only*.

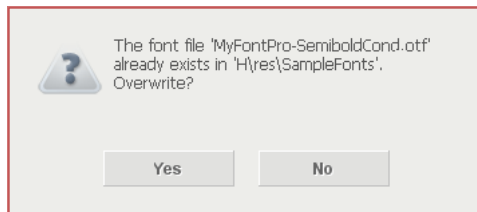
Typical confirmation message for a newly generated font file.



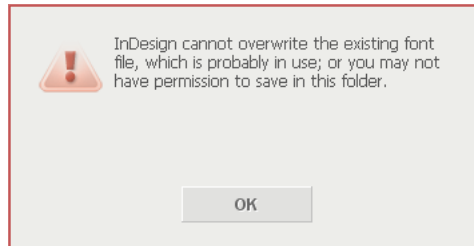
6. Updating the font file

InDesign does not mind if you change a font behind its back. Even if it doesn't "take" the change immediately, it is enough to temporarily switch out of InDesign and back again.

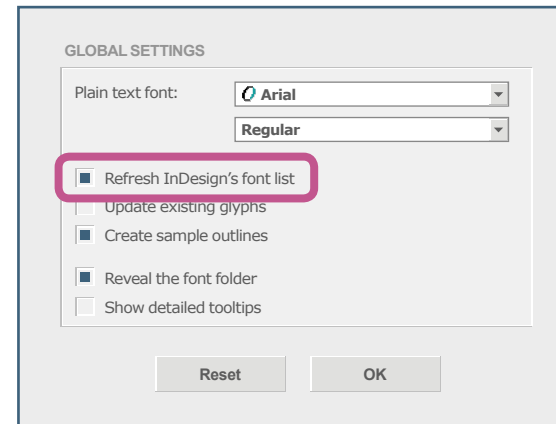
However, it is possible your operating system (Windows or Mac OS X) sees it as a problem, because it thinks the font file is "in use" and you get an error while trying to save your font file. IndyFont cannot solve this problem for you.



IndyFont's alert and error messages related to updating issues.



Sometimes it's enough to wait a moment and try again; sometimes, you need to make sure the font is not "in use" by closing the document in which you are using it. And sometimes, you have to exit InDesign and re-start it. If every reasonable action fails, change the name of your font in the main dialog of IndyFont, so a totally new font file will be created.






In addition, the GLOBAL SETTINGS panel provides an option that automatically checks and refreshes the InDesign font list whenever IndyFont starts up or creates a font file.

1. An introduction to Unicode

The Unicode Standard is the universal character encoding scheme for written characters and text. It defines a consistent way of encoding multilingual text that enables the exchange of text data internationally and creates the foundation for global software.

The Unicode Standard, Version 3.0

► Characters and “code points”

You’d be very surprised if you change the *font* of your text and you get a different *text*. Only its *appearance* should change, not its *content*. It is the responsibility of the font designer to ensure that the letter “A” in his font looks, well, like a letter “A” — even if it looks like  or  or . It should definitely *not* look like K or S or ã.

That is why each character has a unique number assigned to it—a Unicode *code point*. The font may change, but this number stays the same, *always*.

The Unicode Consortium is an international non-profit organization that states which character is assigned what code point (and some related details, such as “should there be a code for an uppercase version of the German character ‘ß?’”). OpenType fonts follow the Unicode recommendations for character encoding, so as long as you use software that internally uses Unicode,

you can comfortably change between your font and another one without having to worry that something *in your text* may change as well.

The Unicode Standard defines lots of characters—and new ones are added constantly, in reaction to changing views on font design (an uppercase variant of the German “ß” was included in 2008 with the code U+1E9E), what constitutes an alphabet (the Takri alphabet, a member of the Brahmic script family, was added in 2012), and determining what a separate character “is” to warrant a code point of its own (emojis were added in Unicode version 6.0 in 2011). It may only be a matter of time before the Klingon alphabet gets officially added (the Unicode Consortium rejected this in 2001), as well as a separate code point for the “Why not Zoidberg” emoticon: (V)(°,°)(V)

The common convention of writing a Unicode code point is “U+”, followed by the code point number in *hexadecimal*.

NOTE It’s a fair guess the consortium members are all programmers. In hexadecimal, you do not count up to 10 before adding a digit, you count all the way up to 16. The “single digits” 10 to 15 are represented by the characters A to F, so A has a value of 10, B is 11, and the value of 12 is $16 \times 1 + 2$, or 18. . . . A good thing is you do not actually need to know this. All you need to remember is that U+0123 stands for *some* Unicode character (this would be a “g”), and so does U+221E (the symbol “∞”).

If you want to add some math symbol or a Greek or Cyrillic character to your font, you do not need to look up the number, as IndyFont knows the Unicode for a lot of characters and is happy to look it up for you. For the “unnamed” ones you have to use its Unicode value.

NOTE IndyFont recognizes about 3,700 standard names that all come from the Adobe Glyph List:
<http://sourceforge.net/adobe/aglfn/wiki/Home/>
All other characters should be entered as Unicode values.

► *Unicode ranges*

Unicode values do not run uninterrupted from the very lowest up to the newest additions. The entire range of code points is divided into code *ranges*, where each block of codes contains a set of characters that belong together (more or less).

There are lots of codes with a “reserved” status; the invisible code that toggles from left-to-right to right-to-left text setting, for example, is a Unicode element as well. The allocated space for several alphabets have been deliberately over-estimated to allow for johnny-come-lately additions. Besides, the Unicode consortium likes to see their range numbers start on round numbers (round, at least, in their preferred hexadecimal notation).

Using a reserved or invalid Unicode in your font is a bad idea in general and may even make it unusable, so IndyFont verifies

all codes in your font against a list of known ranges, and it will warn you if it finds something wrong.

NOTE All Unicode values—i.e. code points—that IndyFont supports are given in “Manage your character set”, Section 5, *Specifying character codes and glyph names*.

► *The “Private Use Area”*

What of characters that do not have a Unicode? You may want to draw a character-plus-accent for which there is no code (yet), such as “V̂”, or you added a custom ligature, or perhaps you want to have a single character in the shape of your company logo. Unicode even allows for this: the code range between U+E000 and U+F8FF is designated as “Private Range”, and you can put anything you like under one of these available 6,400 “free” codes. A drawback, of course, is that these character codes are valid *for your font only*—changing the font will definitely *not* show “the same character but in a different font”, whatever other font you select.

NOTE To prevent collisions with standard glyph names regularly registered in the Private Use Area, IndyFont only addresses a subset of the Unicode PUA for your own glyph names, that is, the range U+E000 ► U+FFFF. See “Manage your character set”, Section 5, *Specifying character codes and glyph names*.

It is tempting to put all of your designs, whatever they represent, into this Private Range. It is also tempting to go alphabetically

and memorize “A is for alpha, B is for Beta, P is for Pi” for a single font of which you are the only user. However, as soon as you do more than making a single font—perhaps you want a matching Italic or Bold version—you will find the extra work in making sure the right character goes under the right code pays off.

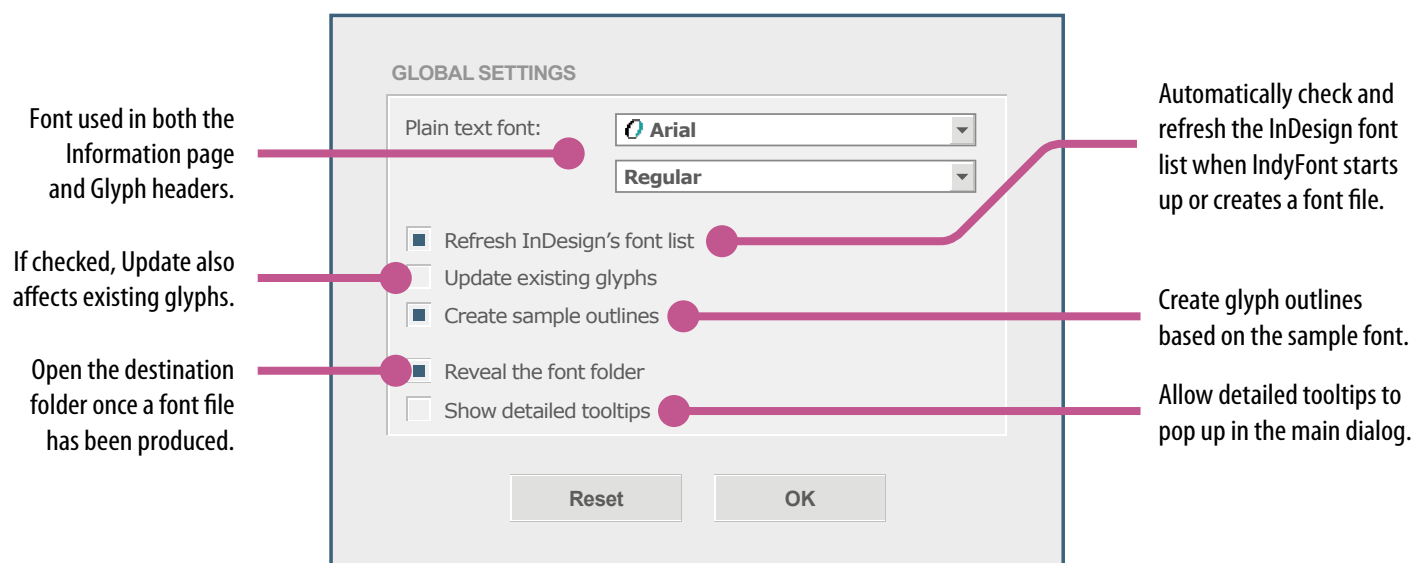
NOTE All is well as long as you are using your font inside InDesign itself, but as soon as, for example, you export your file to an e-pub, “P is for pi” no longer holds and you will find a regular “P” end up in your e-book. If you assign your pi character the correct code, it is going to be π in *everything* you export, from a lowly plain text file up to a PDF.

2. Global settings

IndyFont provides some additional options which you can set from the GLOBAL SETTINGS panel. Being in the main dialog, click the gear icon button to get access to this panel.

All these options are “global”, that is, they affect the behavior of IndyFont itself, not only the current font template.

- Click the OK button to register your settings.
- Click the Reset button to reset the native IndyFont preferences.



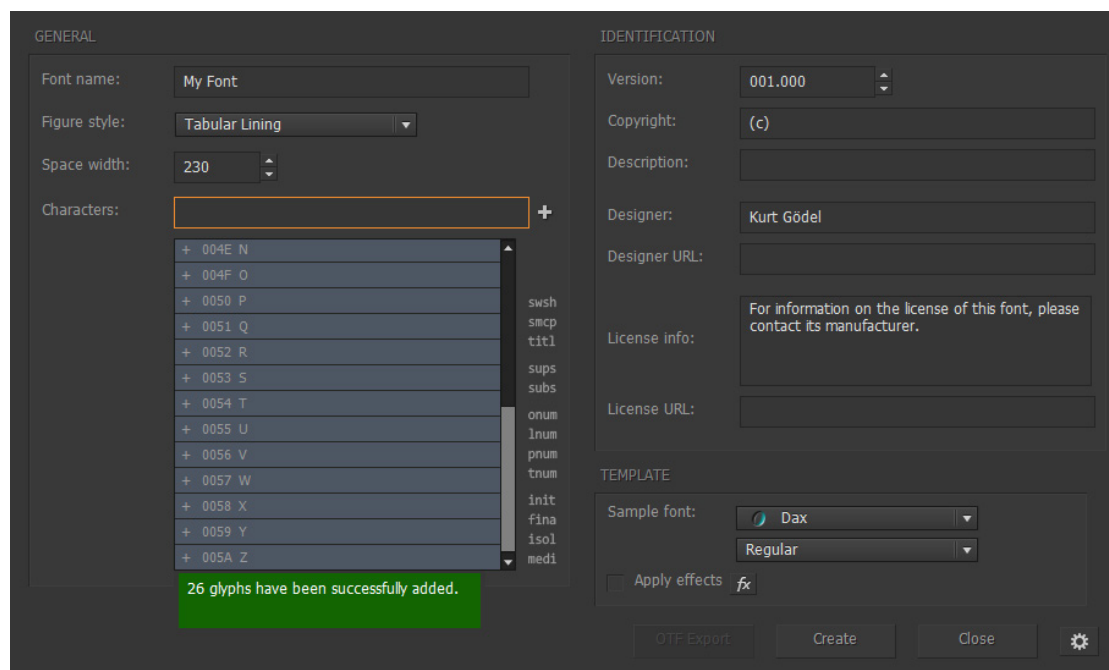
3. InDesign CC compatibility

IndyFont v. 1.127 and later is compatible with InDesign CC. Various issues or bugs were gradually fixed in this regard:

- 1) Version 1.127—released on September 12, 2013—has provided a preliminary patch. A number of problems were remaining though.
- 2) Version 1.129—released on November 9, 2013—has mainly fixed a bug specific to Mac OS X 10.9 (Mavericks).
- 3) Version 1.133—released on February 9, 2014—has improved the compatibility at many levels: support of the new Color Theme, enhanced readability of the dialog controls and popups, etc. The unexpected behavior of the Enter key on Mac OS X was fixed too. Also, InDesign CC 9.2 x32 shouldn't crash anymore on Win64 platforms.
- 4) Version 1.135—released on November 21, 2014—has minor issues fixed and supports InDesign CC 10.x (2014).

NOTE IndyFont, of course, remains backwards compatible with InDesign CS4, CS5, and CS6 (Mac OS X and Windows).

As Adobe is constantly updating InDesign CC, you may still experience issues that we cannot anticipate despite the many tests we perform. Please, feel free to report any new bug at support@indiscripts.com.



IndyFont's main dialog rendered in InDesign CC 10.1 (2014) with dark theme selected from Preferences ► Interface ► Appearance.



IndyFont

1.1

www.indiscripts.com

A plug-in for Adobe® InDesign® based on Adobe® ExtendScript and ScriptUI. Created by Theunis de Jong. Developed by Theunis de Jong and Marc Autret. Designed by Dominique Chiron and Marc Autret. User Interface available in English, French, and German. German localization provided by Birgit Stolte.

"We would like thank our beta-testers, in no special order, Joel Cherney, John Hawkinson, Peter Kahrel, Dominique Chiron, Marijan Tompa and Peter Spier. They got press-ganged into testing the very first incarnations of IndyFont, but all of them were very willing to do so, and they all urged us to go ahead and make it into a full commercial version. Without their continuous support, IndyFont would probably have been left as an interesting-yet-unfinished idea, somewhere in a dusty corner of my hard disk." — Jongware

Main Product Page: <http://www.indiscripts.com/category/projects/IndyFont>
Tryout version: <http://www.indiscripts.com/blog/public/scripts/IndyFontTry.zip>
Technical Support: support@indiscripts.com

Purchasing IndyFont: <http://www.indiscripts.com/store/IDFT>
End User License Agreement: <http://www.indiscripts.com/pages/eula>
Terms and Conditions of Sale: <http://www.indiscripts.com/pages/cgv>
Copyright Notice: <http://www.indiscripts.com/pages/copyright>
Jongware's website: <http://www.jongware.com>

This manual, as well as the software documented in it, is released under license and may be used or copied only in accordance with the terms of that license. The content of this document is subject to change without notice. Every effort has been made to ensure that the information in this document is accurate. However, Indiscripts assumes no responsibility or liability for any error that may appear in this document. InDesign, the InDesign logos, are trademarks of Adobe Systems Incorporated.

© Jongware, 2012 — © Jongware & Indiscripts, 2013-2014.
All rights reserved. Made in France.

