

Harbor Type presents

Rocher

a solid font family

USER GUIDE

Contents

About 3

The family 4

How to use it 5

In Adobe InDesign 6

In Adobe Illustrator 9

In Microsoft Word 10

On the web 11

Character set 15

Language support 16

Specifications 17

About

Rocher was designed while looking for an answer to a simple question: what would a typeface look like if it was made of stone? It certainly would look solid, but did we have to add cracks and rubble so it would resemble rocks? We didn't think so.

We decided to tackle the problem a different way. We added corners where there usually aren't any and threw some unusual letterforms into the mix. The result is a typeface that feels like stone, but if you look closely there is nothing inherently stony about it. Unexpected corners provide just the right amount of roughness, while unusual letterforms give the text an informal aesthetic, traces of something naive and handmade.

A family was born when the sturdy letterforms were turned into a series of playful layers. With 9 fonts in total, Rocher can be mixed and matched to create unique layered compositions that add depth to the layout.

We designed Rocher to be used in logotypes, packaging, mobile apps and headlines. We are confident you will find another handful of scenarios where it can shine.



SASQUATCH
ZACH GALIFIANAKIS
Cerveza Weiss
IT'S ONLY ROCK'N'ROLL

The family

The Rocher family consists of 9 different styles. Some of them can be used on their own, like Regular, Inline and Jewel. Other styles only make sense when combined.

The styles are:

- Rocher Regular
- Rocher Inline
- Rocher Spine
- Rocher Bevel A
- Rocher Bevel B
- Rocher Jewel
- Rocher Outline A
- Rocher Outline B
- Rocher Extrude

Aenigmatite Regular

Bromargyrite Inline

Carlsbergite Spine

Djerfisherite Bevel A

Eðingtonite Bevel B

Felsöbányaite Jewel

Gwihabaite Outline A

Håleniusite Outline B

Ianbruceite Extrude

How to use it

The way you use Rocher varies from software to software. As a general rule, you will need a text frame for every layer and/or color you use.

If your software allows you to edit kerning, remember to always use the default option. It may be called Auto, Metrics or something like that. Never set the kerning mode to Optical as it will completely ignore the built-in metrics and, as a consequence, the layers will not align.

Keep in mind there is no easy way to achieve this in text processors like Microsoft Word, TextEdit or Notepad. You will need a graphics software (Adobe InDesign, Adobe Illustrator, CorelDRAW or similar) to use Rocher family to its fullest.



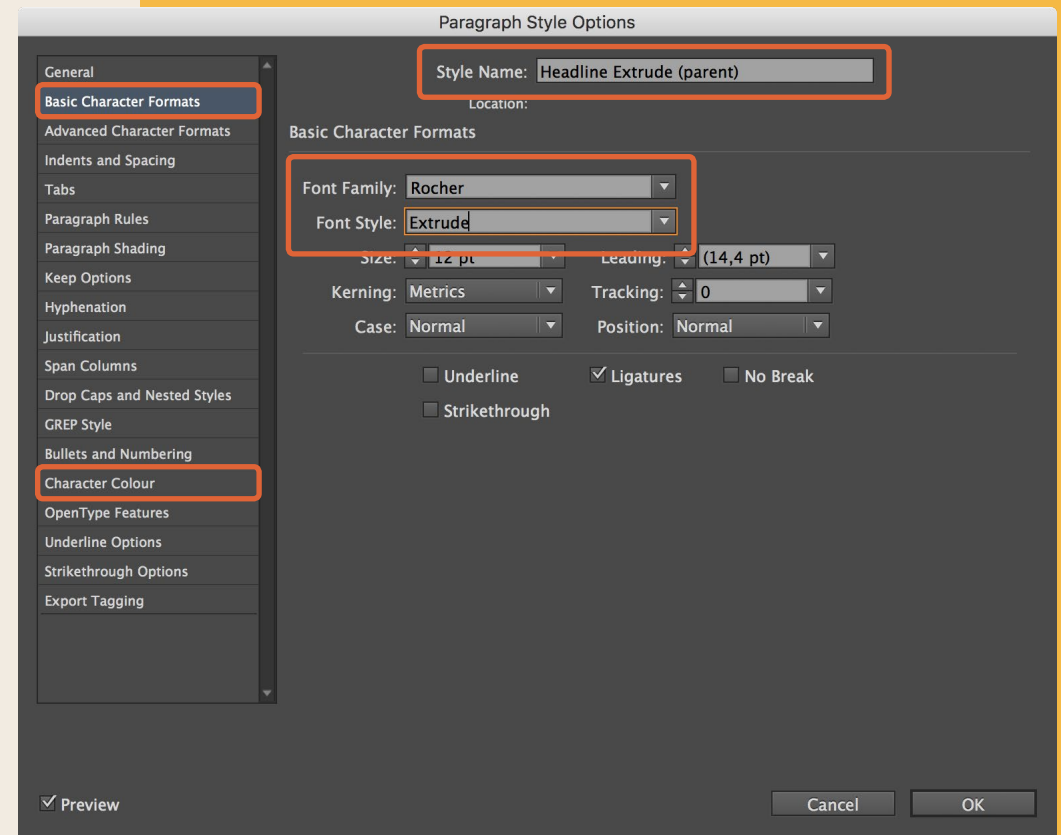
In Adobe InDesign

We highly recommend using paragraph and character styles for your layouts. It will take you a little longer to set them up, but it will be much easier for you to make changes later.

To learn more about paragraph styles, watch this [12-min video](#) and read the [official documentation](#).

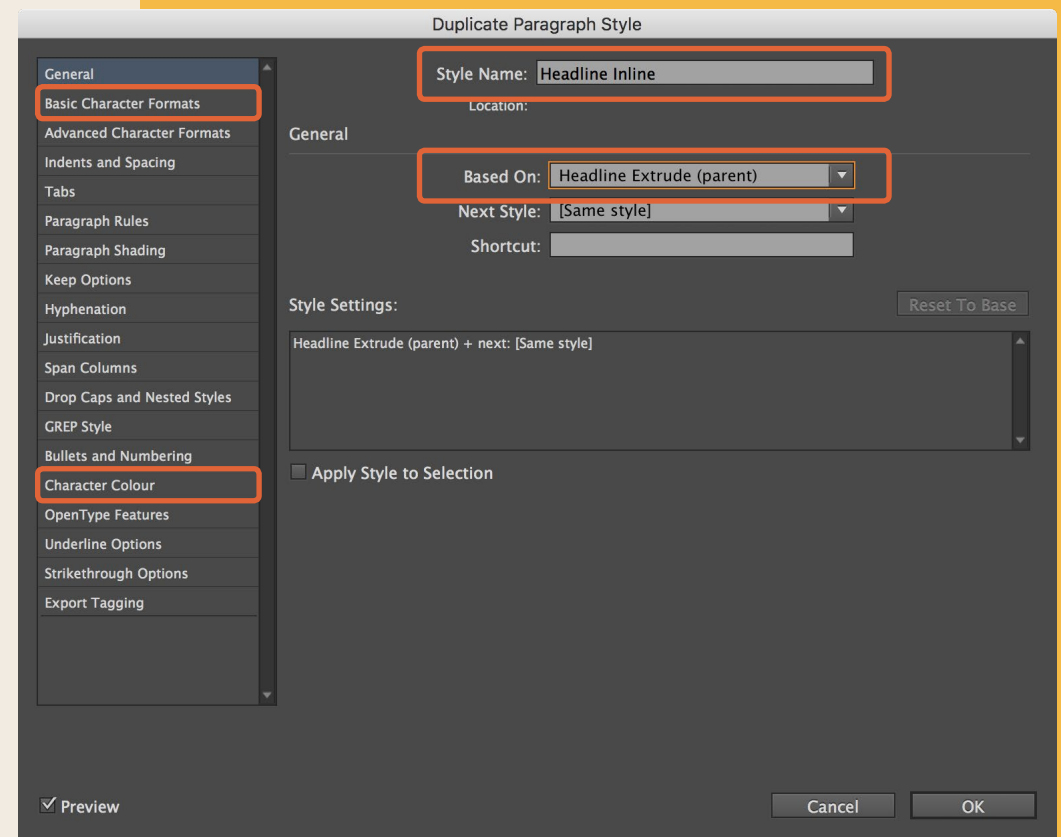
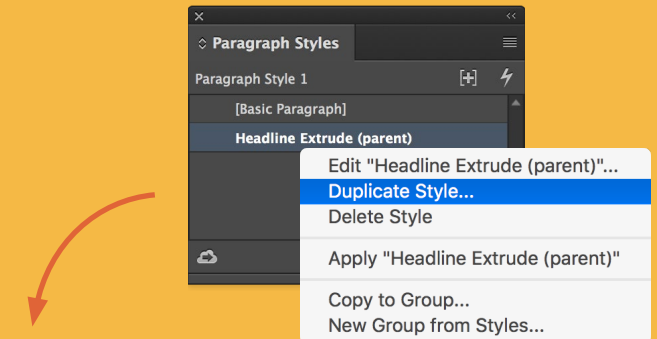
Setting up styles (recommended)

1. Create a new **Paragraph Style** by clicking the button that resembles a folded sheet of paper in the namesake panel. If you cannot find this panel, you can turn it on by accessing the menu **Window > Styles > Paragraph Styles**.
2. Double-click the newly created style to edit its settings. If this is the first paragraph style of the document, it is probably named **Paragraph Style 1**.
3. First of all, change the **Style Name** to something more meaningful. Let's say "Headline Extrude (parent)". We also added "parent" to the style name to indicate it is the main one. More on that later.



4. Click the **Basic Character Formats** tab on the left, set the font family to Rocher and the font style. For this example we'll be using Extrude.
5. Click the **Character Color** tab on the left and set the color for this layer. Close the window by clicking **OK**.
6. Back to the **Paragraph Style** panel, select the style you just created, right-click on it and click **Duplicate Style...**
7. On the window that opens, rename the style to something more meaningful, like "Headline Inline". Under the **General** settings, change the **Based On** to the first style you created (the parent).
8. Change the font style under **Basic Character Formats** and its color under **Character Color**.
9. Repeat steps 7–8 for each additional layer.

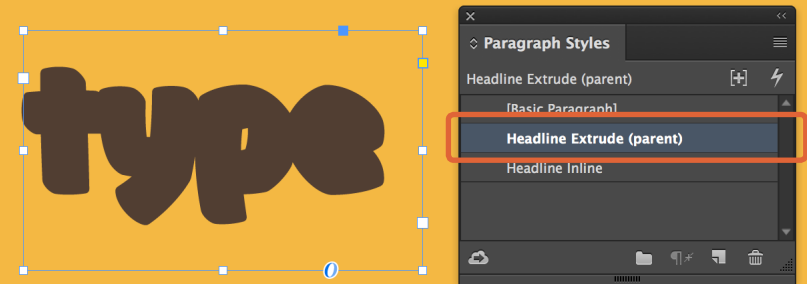
The styles are set. If you decide to change any of the attributes that are shared between them (size, leading, indents or anything other than the font style and color), do it only in the parent style. If you set the **Based On** property correctly, all styles will inherit the attributes of the parent, saving you some time and hassle.



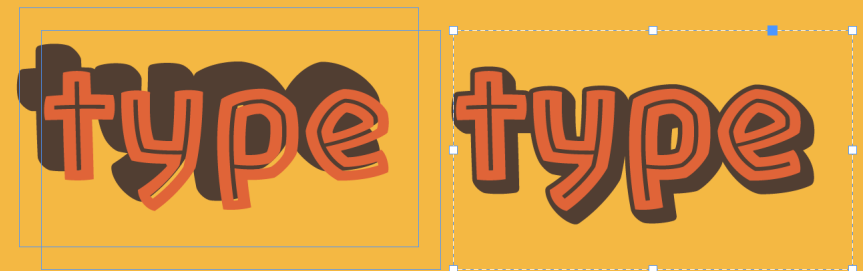
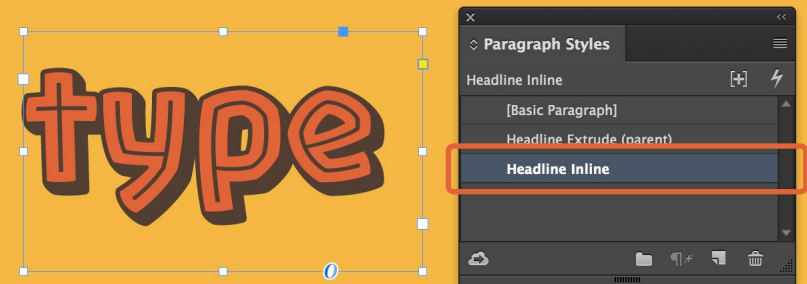
Actually using the fonts

You have set up your styles, now it's time to actually make some layouts. If you don't want to work with styles, keep in mind you'll need to manually change the fonts and color for each text frame you create.

1. Draw a text frame and type your content. For this frame, select the paragraph style that corresponds to the bottommost layer.
2. Duplicate the text frame with your content and select another of your paragraph styles. To duplicate exactly on top of each other, use the keyboard shortcuts **Copy** (⌘C or Ctrl+C) and **Paste in Place** (⌘⇧V or Ctrl+Alt+Shift+V).
3. Repeat step 2 for each additional layer. Pro-tip: it will save you some time if you work from the bottom up, as you won't need to rearrange objects later.
4. If you need to reorder the text frames, you can use the keyboard shortcuts **Send Backwards** (⌘[or Ctrl+[) and **Bring Forward** (⌘] or Ctrl+]). Also, the **Layers** panel is your friend.
5. After you have finished the layered composition, group the text frames so you don't accidentally move them individually.



Copy, then Paste in Place



Ungrouped

Grouped

In Adobe Illustrator

While Adobe Illustrator also supports paragraph styles, we have the feeling most people don't use them. If you do, follow the instructions in the previous section.

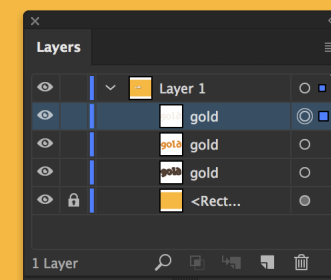
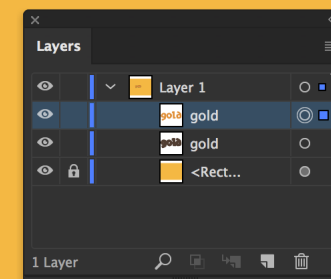
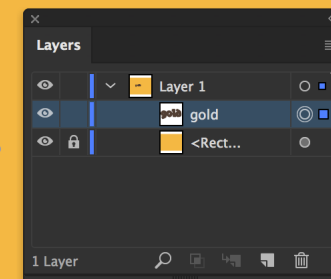
1. Type your text, set it to one of the Rocher styles and select a color. As we like to work with the bottommost object first, we'll set it to Rocher Extrude.
2. Duplicate the text object with the commands **Copy** (⌘C or Ctrl+C) and **Paste in Front** (⌘F or Ctrl+F). Select a different font style and color for this object.
3. Repeat step 2 for each additional layer.
4. If you need to reorder the text frames, you can use the keyboard shortcuts **Send Backwards** (⌘[or Ctrl+[) and **Bring Forward** (⌘] or Ctrl+]). Also, the **Layers** panel is your friend.
5. If you want to change colors or styles, you can cycle between objects with the commands **Next Object Below** (⌘[or Ctrl+Alt+[) and **Next Object Above** (⌘] or Ctrl+Alt+]). The **Layers** panel is now your best friend.
6. After you have finished the layered composition, group the text objects so you don't accidentally move them individually.



Copy, then Paste in Front



Copy, then Paste in Front



This is how this composition was assembled. Remember to group the objects so you don't move them individually.

In Microsoft Word

Unfortunately, Microsoft Word and other text processors do not provide an easy way to precisely stack text objects. This means you cannot achieve the multicolor effects shown in this document in this kind of software.

This doesn't mean you can't use the fonts in it. After you've installed the fonts in your system, they will be available in the font dropdown menu but you'll be limited to using single fonts in single colors.

But there is a silver lining. Some very nice folks are developing a color font format that you'll be able to use on the web and anywhere in operating systems, but it is still some years away from being implemented. When it does, we'll probably update Rocher to support this technology.

If you are unsure whether Rocher works on a specific software you use, send us an email at ahoy@harbortype.com and we'll be happy to investigate.



On the web

Using Rocher on the web is possible, but it involves some fiddling with HTML and CSS. If you don't know what they are, we recommend you to talk with your developer.

If you want to use the fonts plain and simple, just link them in the stylesheet and use them as any other font. This is described in the next page.

However, if you'd like to use layers, there are at least two different approaches: nesting them inside many `div` tags or using pseudo elements. Keep on reading this guide to learn how to use them.

Warning: as a result of their organic contours, the Rocher fonts are not hinted. This may lead to unreadable text in small sizes. Use them for headlines and in large sizes only (24px or more).

Format support by browser version

	<u>EOT</u>	<u>WOFF</u>	<u>WOFF2</u>
Chrome		6+	36+
Edge		12+	14+
Firefox		3.6+	35+
Internet Explorer	6+	9+	
Opera		11.10+	26+
Safari		5.1+	10+

Source: caniuse.com (links on headers)

Importing the fonts in the CSS

No matter what you do, you'll have to import the fonts in the website's stylesheet before you can use them. To do that, put the webfont files in the same folder as your stylesheet or in a subfolder (we like to put them in a subfolder called "fonts"). The webfonts have the extensions ".eot", ".woff" and ".woff2".

After the webfonts are in place, add the lines on the right to the top of your CSS file. Repeat these lines for every style you'll be using on your layout, changing the names for each style. After you've done that, the fonts are ready for use. For techniques on layering fonts, read the next pages.

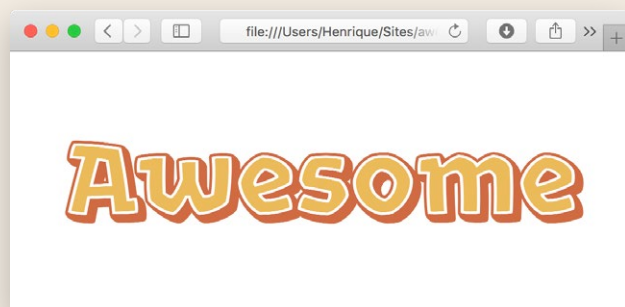
CSS

```
@font-face {  
    font-family: 'Rocher-Extrude';  
    src: url('./fonts/Rocher-Extrude.eot');  
    src: url('./fonts/Rocher-Extrude.eot?#iefix')  
    format('embedded-opentype'),  
        url('./fonts/Rocher-Extrude.woff2') format('woff2'),  
        url('./fonts/Rocher-Extrude.woff') format('woff');  
    font-weight: normal;  
    font-style: normal;  
}
```

Using DIVs to layer text

The most straightforward way to layer fonts on the web is to put your content inside a `div` and then repeat it for each layer you need. Study the code on the right and customize it to your needs. You can repeat the `div` element as many times as needed. Remember to apply the `-webkit-font-smoothing` property to the text element in order to minimize rendering issues.

There is, however, a problem with this approach: the text will be repeated three times in the code. While this doesn't seem like a problem, visitors who use text-to-speech technologies will listen to the text three times. Also, your webpage will have repeated text in search engines results. This is why we actually recommend you to use the pseudo elements for layering text.



On the web

HTML

```
<div id="layer1">
  <h1>Awesome</h1>
</div>
<div id="layer2">
  <h1>Awesome</h1>
</div>
<div id="layer3">
  <h1>Awesome</h1>
</div>
```

CSS

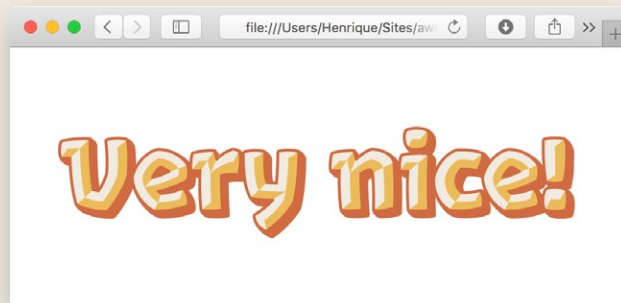
```
#layer1, #layer2, #layer3 {
  display: block;
  position: absolute;
}
h1 {
  font-size: 90px;
  -webkit-font-smoothing: antialiased;
}
#layer1 h1 {
  font-family: 'Rocher-Extrude';
  color: rgb(209,107,67);
}
#layer2 h1 {
  font-family: 'Rocher-Regular';
  color: rgb(236,187,90);
}
#layer3 h1 {
  font-family: 'Rocher-OutlineA';
  color: rgb(255,255,255);
}
```

Using pseudo elements to layer text

Our preferred method for using layer fonts on the web is to employ the `::before` and `::after` pseudo elements.

In a nutshell, we create a heading element inside a `div` and then replicate its content through the `text=` attribute. We then use CSS to create two pseudo elements (`::before` and `::after`) with the same text but positioned behind and on top of the original heading. Remember to apply the `-webkit-font-smoothing` property to the text element in order to minimize rendering issues.

We like this approach better because it degrades gracefully. Screen readers will only read the text once and, if the user's browser doesn't support pseudo elements, it will still render the heading in its base formatting (a single layer and color, but still readable). Unfortunately, you cannot combine more than 3 styles using this method. If you need to do so, you are stuck with the `div` approach for now.



HTML

```
<div class="heading-container">  
  <h1 text="Very nice!">Very nice!</h1>  
</div>
```

CSS

```
h1 {  
  font: 90px 'Rocher-Regular';  
  color: rgb(236,187,90);  
  -webkit-font-smoothing: antialiased;  
}  
h1::before {  
  content: attr(text);  
  font-family: 'Rocher-Extrude';  
  color: rgb(209,107,67);  
  z-index: -1;  
}  
h1::after {  
  content: attr(text);  
  font-family: 'Rocher-BevelB';  
  color: rgb(242,235,225);  
}  
h1::before, h1::after {  
  position: absolute;  
  top: 0;  
  left: 0;  
}  
.heading-container {  
  position: relative;  
  display: block;  
}
```


Language support

Abenaki	Chavacano	Galician	Jèrriais	Māori	Potawatomi	Sorbian (Lower Sorbian)	Venetian
Afaan Oromo	Chichewa	Ganda	Kaingang	Marquesan	Q'eqchi'	Sorbian (Upper Sorbian)	Vepsian
Afar	Chickasaw	Genoese	Kala Lagaw Ya	Megleno-Romanian	Quechua	Sotho (Northern)	Volapük
Afrikaans	Cimbrian	German	Kapampangan (Latin)	Meriam Mir	Rarotongan	Sotho (Southern)	Vöro
Albanian	Cofán	Gikuyu	Kaqchikel	Mirandese	Romanian	Spanish	Wallisian
Alsatian	Corsican	Gooniyandi	Karakalpak (Latin)	Mohawk	Romansh	Sranan	Walloon
Amis	Creek	Guadeloupean Creole	Karelian (Latin)	Moldovan	Rotokas	Sundanese (Latin)	Waray-Waray
Anuta	Crimean Tatar (Latin)	Guaraní	Kashubian	Montagnais	Sami (Inari)	Swahili	Warlpiri
Aragonese	Croatian	Gwich'in	Kikongo	Montenegrin	Sami (Lule)	Swazi	Wayuu
Aranese	Czech	Haitian Creole	Kinyarwanda	Murrinh-Patha	Sami (Northern)	Swedish	Welsh
Aromanian	Danish	Hän	Kiribati	Nagamese Creole	Sami (Southern)	Tagalog	Wik-Mungkan
Arrernte	Dawan	Hawaiian	Kirundi	Ndebele	Samoan	Tahitian	Wiradjuri
Arvanitic (Latin)	Delaware	Hiligaynon	Klingon	Neapolitan	Sango	Tetum	Wolof
Asturian	Dholuo	Hopi	Ladin	Ngiyambaa	Saramaccan	Tok Pisin	Xavante
Atayal	Drehu	Hotçak (Latin)	Latin	Niuean	Sardinian	Tokelauan	Xhosa
Aymara	Dutch	Hungarian	Latino sine Flexione	Noongar	Scottish Gaelic	Tongan	Yapese
Bashkir (Latin)	English	Icelandic	Latvian	Norwegian	Serbian (Latin)	Tshiluba	Yindjibarndi
Basque	Estonian	Ido	Lithuanian	Novial	Seri	Tsonga	Zapotec
Bemba	Faroese	Ilocano	Lojban	Occidental	Seychellois Creole	Tswana	Zulu
Bikol	Fijian	Indonesian	Lombard	Occitan	Shawnee	Tumbuka	
Bislama	Filipino	Interglossa	Low Saxon	Oshiwambo	Shona	Turkish	
Bosnian	Finnish	Interlingua	Luxembourgish	Ossetian (Latin)	Sicilian	Turkmen (Latin)	
Breton	Folkspraak	Irish	Maasai	Palauan	Silesian	Tuvaluan	
Cape Verdean Creole	French	Istro-Romanian	Makhuwa	Papiamentu	Slovak	Tzotzil	
Catalan	Frisian	Italian	Malay	Piedmontese	Slovenian	Uzbek (Latin)	
Cebuano	Friulian	Jamaican	Maltese	Polish	Slovio (Latin)		
Chamorro	Gagauz (Latin)	Javanese (Latin)	Manx	Portuguese	Somali		

Specifications

Each font of the family contains 421 glyphs and support over 200 Latin-based languages. The current version is 1.000.

Rocher was designed by Henrique Beier between 2016 and 2017. It was released in April 2017.

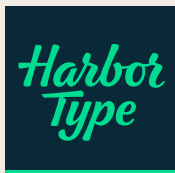
Available formats

Desktop fonts: CFF-flavoured OpenType (OTF)

Webfonts: WOFF, WOFF2 and EOT

Other formats are available upon request.





Harbor Type is an independent type foundry based in the city of Porto Alegre, Brazil. We work with everything type, from retail and custom typefaces to letterings and logotypes. The quality of our work has been recognized by institutions like Tipos Latinos Biennial and Typographica. Harbor Type is run by Henrique Beier, graphic designer by major but type designer by heart.

www.harbortype.com

ahoy@harbortype.com