MILIEU GROTESQUE
TECHNICAL INFORMATION ON WEBFONTS

This is a basic technical guide (Version 2.1, March 2016) to help you integrating our webfonts. Should you have any additional questions, please don't hesitate to contact our tech team via support@milieugrotesque.com

1.0 Definitions and Formats

1.1 About Milieu Grotesque Web Fonts

Milieu Grotesque offers fonts in web-specific formats for use on your website. Our web font license covers use of a font for one or several pre-defined domains/URLs and a maximum number of page views. Milieu Grotesque does not charge any monthly or annual subscription fees.

1.2 WOFF and WOFF2 Formats

Milieu Grotesque offers fonts in the web-specific formats WOFF and WOFF2 (suffix .woff and .woff2) since these are supported by all major browsers: IE, Firefox, Chrome, Safari, Opera, all iOS browsers, Android Browser and Chrome for Android. Furthermore, WOFF is an open format – while it isn't controlled by any specific company, it is also restricted to use in web browsers. In other words: WOFF fonts won't work in e. g. InDesign or Microsoft Word.

A clear benefit of WOFF is its lossless compression. Compared to the original font, file sizes tend to be 50% smaller, enabling faster downloads and shorter page loading times. Due to growing mobile use and restricted bandwidth, this is an important feature.

Building on the premise of WOFF, the new WOFF2 format reduces file sizes even further: Still lossless, a WOFF2 file can be up to 30% smaller than a corresponding WOFF file. At the time of writing, the new format hasn't been adopted by all browsers yet, but we expect it to be supported by all major browsers soon. Check out http://caniuse.com/#search=woff2 for an up-to-date overview of supporting browsers. Our recommendation would be to start using WOFF2 together with WOFF as a fall-back option.

1.3 Alternative Formats

Over the years, several other formats like EOT (Embedded OpenType), TTF (TrueType) and SVG (Scalable Vector Graphics) were developed by different parties (e.g. EOT by Microsoft) for use in specific browsers. Due to WOFF's broad acceptance and support by all modern browsers (since 2012), these formats are no longer relevant or required.

Should you need one of these formats for a specific application, we will supply you with your format of choice upon request and for an additional fee. Just contact us with your project requirements.

2.0 Use with Cascading Style Sheets (CSS)
2.1 Using Web Fonts with CSS

You can use our web fonts with the CSS rule @font-face. To start, add your web fonts to your own web server. Please note: You will need a WOFF file for each font style.

Once this is done, add a @font-face rule in CSS to establish both the name of the typeface and the path on your server. For example:

```
@font-face {
 font-family: 'MaisonNeue';
 src: url('fonts/MaisonNeueWEB-Book.woff') format('woff');
 font-weight: normal;
 font-style: normal;
}
```

Afterwards, you can reference this font family in other CSS font stacks like any other font family, including fall-backs:

```
body {
  font-family: 'MaisonNeue', sans-serif;
}
```

If you want a browser to use WOFF2 whenever possible, but fall back to WOFF when and where this format isn't supported, the following example applies. Please keep in mind, though, that the order of specified alternatives is relevant since the browser will automatically select the first supported option.

```
@font-face {
 font-family: 'MaisonNeue';
 src:
 url('MaisonNeueWEB-Book.woff2') format('woff2'),
 url('MaisonNeueWEB-Book.woff') format('woff');
}
```

Naturally, there are more details to CSS rules and browser interpretations than we could cover in this document. One key aspect, however, is how a font's weight and style interacts with @font-face. The short answer: Don't give different font styles different font family names – use style linking instead. For example, in order to add italics to the version above, just add:

```
@font-face {
  font-family: 'MaisonNeue';
  src: url('fonts/MaisonNeueWEB-BookItalic.woff'
  format('woff');
  font-weight: normal;
  font-style: italic;
}
```

This ensures that CSS will not apply faux-italics to a font style that shouldn't ever be in italics – or maybe even already is. For further details and an extensive overview of this issue, we recommend the following article: http://smashingmagazine.com/2013/02/14/setting-weights-and-styles-at-font-face-declaration/

3.0 OpenType Features and Settings

3.1 OpenType Features

Web browsers are starting to support OpenType features like ligatures, contextual and stylistic alternates, tabular figures, etc. Such features can be enabled via CSS rules. But before we dive in, a quick word of warning: Features like kerning and ligatures can slow down rendering performance, so please test this first if you are targeting mobile devices.

3.2 Font-Feature-Settings

This property gives you low-level control over OpenType font features. Enable this option by using official OpenType feature tags like ‘kern’, ‘liga’, ‘tnum’, etc. See https://www.microsoft.com/typography/otspec/featurelist.htm for a complete list of features. Any feature tags that aren’t actually present in the font are simply ignored. Although specifications recommend using ‘font-variant‘ and its related sub-properties, ‘font-feature-settings’ is supported by a much wider range of browsers. Time for another example: To enable kerning, common ligatures and tabular number spacing for a paragraph, try the following:

```
p {
   font-feature-settings: 'kern' 1, 'liga' 1, 'tnum' 1;
}
```

The simple rule: To enable a particular feature, add a 1 to the tag – or a 0 to disable it. If you would like to explore this further, check out this experimental online tool – it even works for locally installed fonts: http://clagnut.com/sandbox/css3/. In line with all new CSS features, don‘t forget to use vendor prefixes like -moz-font-feature-settings.

3.3 Text-Rendering

For a while, people were using ‘text-rendering’ to enable kerning or even ligatures. While text-rendering had its uses, there are two severe drawbacks to this method: First up, with Internet Explorer and Edge two major browsers don’t support this rule at all. Secondly, this rule doesn’t really control OpenType features – rather, it broadly indicates to the browser what needs to be optimised when rendering text, i. e. whether to make trade-offs in terms of rendering speed, legibility or geometric precision.

```
p {
   text-rendering: optimizeLegibility;
}
```

In this well-known example, the rule tells a browser to disregard rendering speed and geometric precision. Some browsers will then enable kerning and ligatures, but others might not – or even experience issues rendering the text at all (Android). To cut a long story short: If you want kerning, enable it via ‘font-feature-settings’ instead.

4.0 Blocking and Protecting Font Files
4.1 Block Unauthorised Use

When other websites link directly to your hosted web font files (otherwise known as hotlinking), they use your fonts (without a licence) and your bandwidth, thus adding to your site's traffic costs. Our license agreement stipulates that you have to take reasonable measures against hotlinking and direct access to font files. To prevent unauthorised use, for example, you could enable 'origin and referrer checking' on your web server. This simple measure blocks any request for a font file that doesn't originate from a page on your own site. The technique itself is simple, yet implementation depends on your particular server technology. The following examples refer to a basic Apache HTTP Server.

Let's say your web fonts are stored in a folder called /fonts/ and its content is accessible to the outside world, e.g. via www.example.com/fonts/my-font.woff.

To prevent misuse, you can simply add a file called .htaccess to this folder, containing the following lines:

```
SetEnvIf Referer example\.com localreferer
<FilesMatch \.(woff)$>
Order deny,allow
Deny from all
Allow from env=localreferer
</FilesMatch>
```

The technique itself is covered in the official Apache documentation, see http://httpd.apache.org/docs/2.2/rewrite/access.html. Naturally, you will need to replace 'example.com' with the name of your host and the '\' before any '.'

Now, requests for any files ending in .woff originating from other sites than your own (or without referrer information) will trigger an HTTP error 403 (forbidden). If you get an HTTP error 500 (internal server error) or if you're still able to download the WOFF file, your server probably doesn't allow this kind of configuration settings adjustment. In this case, please refer to your server documentation or contact your server administrator.