

[For updates to this documentation, and improved versions of EOTFAST, visit <http://eotfast.com>]

Working With Fonts In Windows

EOTFAST is a utility written in C++ in accordance with the published Microsoft documentation for its GDI Fonts interface.

[http://msdn.microsoft.com/en-us/library/dd145147\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd145147(VS.85).aspx)

Please note that the Windows End-User-License-Agreement (EULA) commonly contains a subsection regarding “Font Components”.

It reads as follows:

Font Components.

While the software is running, you may use its fonts to display and print content.

You may only:

- embed fonts in content as permitted by the embedding restrictions in the fonts; and
- temporarily download them to a printer or other output device to print content.

Since EOTFAST is coded using Microsoft’s own API’s, EOTFAST carries through with any restrictions and limitations that are imposed by those APIs and they will be applied not only to the referenced “Font Components” but for all fonts no matter their origin.

So please be aware of the rights and restrictions regarding the Windows Font Components so as to meet your obligations. And please note that if you are working with fonts that are **not** Windows Font Components, such as fonts you yourself may have created, EOTFAST is not able to allow you to set, change or otherwise modify the embedding data - commonly called “the embedding bits” – as they exist in **any** font, Windows Font Component or not.

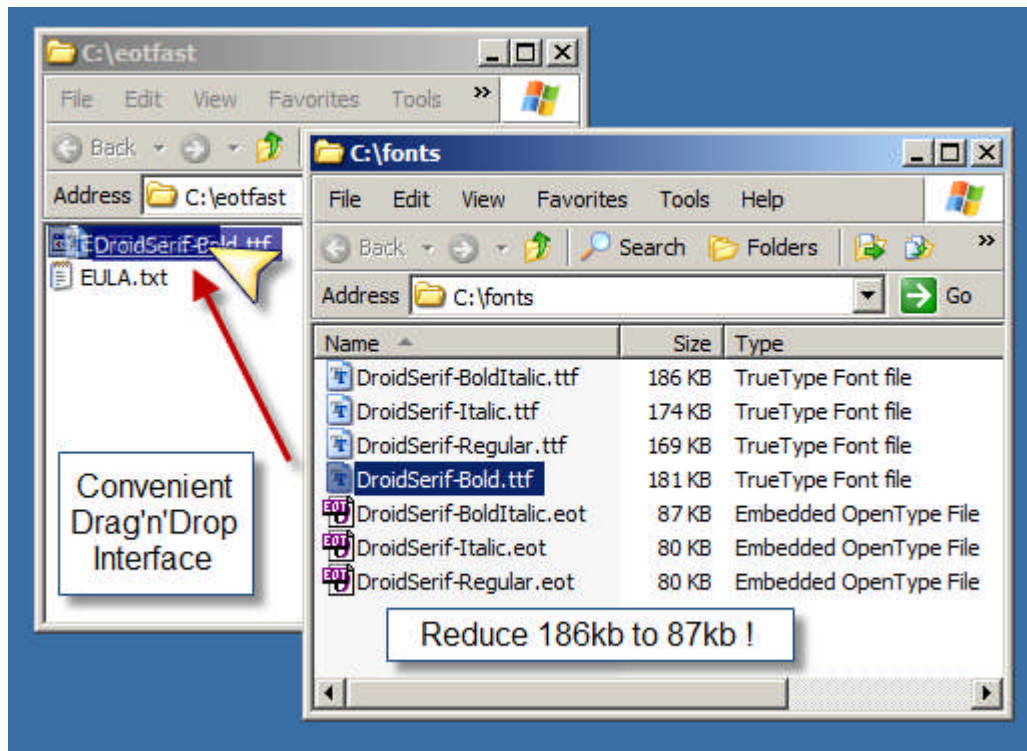
Using EOTFAST

1) Drag'n'Drop Operation

In Windows Explorer, drag and drop the TTF file right onto the EOTFAST-1.EXE file. The EOT file will be created in the same folder as the original TTF you dragged and dropped.

[Note: EOTFAST 2.0 will enable you to select multiple files for batch processing. You can sign up for product updates today at <http://eotfast.com>]

Example of Drag and Drop operation:



Command Line Operation

EOTFAST can also be used at the command line. For example, if both EOTFAST-1.EXE and the font file are located in the folder c:\truetype, open a command prompt window within that folder and specify the EOTFAST executable and then a space and then the TTF file.

```
C:\truetype>EOTFAST-1.EXE somefont.ttf
```

Hit enter and the EOT will be created.

Of course, if the font file is located in a different folder, the full path and file name will have to be specified.

Making EOT FILES

The following information is of most use to font designers who are looking to market their fonts for the web. But it's also for web designers and developers who, on occasion, might need to trouble-shoot an otherwise unexplainable problem with an EOT.

- Internet Explorer 6, 7, and 8 require that TrueType font files be converted to the EOT format for @font-face. There are a billion Windows installations in the world. And IE users account for, at a conservative estimate, 60+% of all web users. However, because there are thousands of freely available fonts and the web has always been largely a do-it-yourself medium, anyone looking to use web fonts may, at some point, run into a problem with an EOT file that, for one reason or another, just isn't working. This can happen with unconverted TTF and OTF fonts targeted at other browsers, too.

@Font-Face in Internet Explorer is twelve years old. There are language communities that have relied on it and it has withstood the test of time. If the @font-face font is not available, fallback font-families appear as they should and bolding and italic synthesized by IE will appear as expected. It works. @Font-Face in IE 6, 7, and 8 is largely interoperable with the CSS3 implementations to be found in Firefox 3.6, Safari 4, Chrome, and the upcoming IE9.

@Font-Face in Internet Explorer is TrueType-centric and a little different in spots from a conformant CSS3 implementation. Only TrueType fonts will convert to an EOT. That's a good thing or a bad thing depending upon who you ask. Whatever your view, it's a fact. And we urge font designers to provide converted and well-hinted TTF versions of their Postscript-flavored OTF CFF fonts for @font-face web linking.

What follows is a set of guidelines for font designers to bear in mind. As well as web designers and web developers who need to troubleshoot on those occasions when they can't quite figure why an EOT file isn't working.

The Specifics

In order for EOT files to work correctly in Internet Explorer, the original TTF font file has to meet certain requirements. Usually they do and you don't have to do anything.

[In EOTFAST 2.0 there will be automatic detection to ensure all requirements are met. Sign up for notification of EOTFAST 2.0 today, at <http://eotfast.com>]

But once in a while they don't work, and the original TTF font will need to be edited.

A font file is a little database made up of tables that store information for different aspects of the font. In order to make an EOT file that works right, some of the data entries inside the TTF have to be entered a certain way. Usually the font designer has already done so and it works fine without any changes, but sometimes adjustments are needed so that Internet Explorer accepts the EOT file and knows what to do with it.

The problem is usually only one out of three or four possible problems and easy to fix.

You can use a font editor or you can fix it with the free font-to-XML/XML-to-font converter [TTX](#).

Other Tools You May Need

If you will be working with @font-face web fonts and do not yet own a font editing tool, we suggest you download and install two **free** tools that will help you with any problems you may have.

They are:

Microsoft's Font Properties Extension Tool

<http://www.microsoft.com/typography/TrueTypeProperty21.msp>

This is free from Microsoft, the most important thing about the font Properties Extension is that it enables you to see what's entered in the name table of the font. Windows and Internet Explorer uses the entries in this table to identify the font when you write its name in the CSS.

TTX

<http://sourceforge.net/projects/fonttools/>

TTX is a free font-to-XML and XML-to-font converter that allows you make changes to the font data in any text editor. It's open source, and there's a good pre-prepared Windows installation package available.

Confirming A Successful Conversion From TTF to EOT

EOTFAST 1.0 comes with two tools designed to make it easy to confirm that your font has indeed converted successfully.

The EOT File Integrity Test Page

In the EOTFAST 1.0 package there is a HTML test page template EOT-File-Integrity-Test .htm which is designed to make it easy to confirm that a successful conversion has taken place.

After you convert the TTF to an EOT:

- 1) Make a copy of the Test Page and rename it to include the name of the font file
- 2) Open up the page and enter the filename and path of the EOT in the appropriate places.
(There are three places, delineated with % percent % signs)
- 3) Save changes and launch the page in IE and take a look to make sure Internet Explorer is displaying the font

[In EOTFAST 2.0 this test page will be automatically generated upon file conversion. Have we mentioned yet that you can sign up for notifications at <http://eotfast.com> ?]

Personal Note – as I convert files and create test pages, they become a kind of reference library for me of the fonts I’ve worked with – with the browser being used as an instant “Font Viewer” for immediate inspection and review. I use an alternative viewer page for other browsers using “raw” OTFs and TTFs – to be published soon. These test “viewer” pages are much, much better than the Windows built-in font viewer and we’ll be introducing improved versions over the coming months and as EOTFAST evolves.

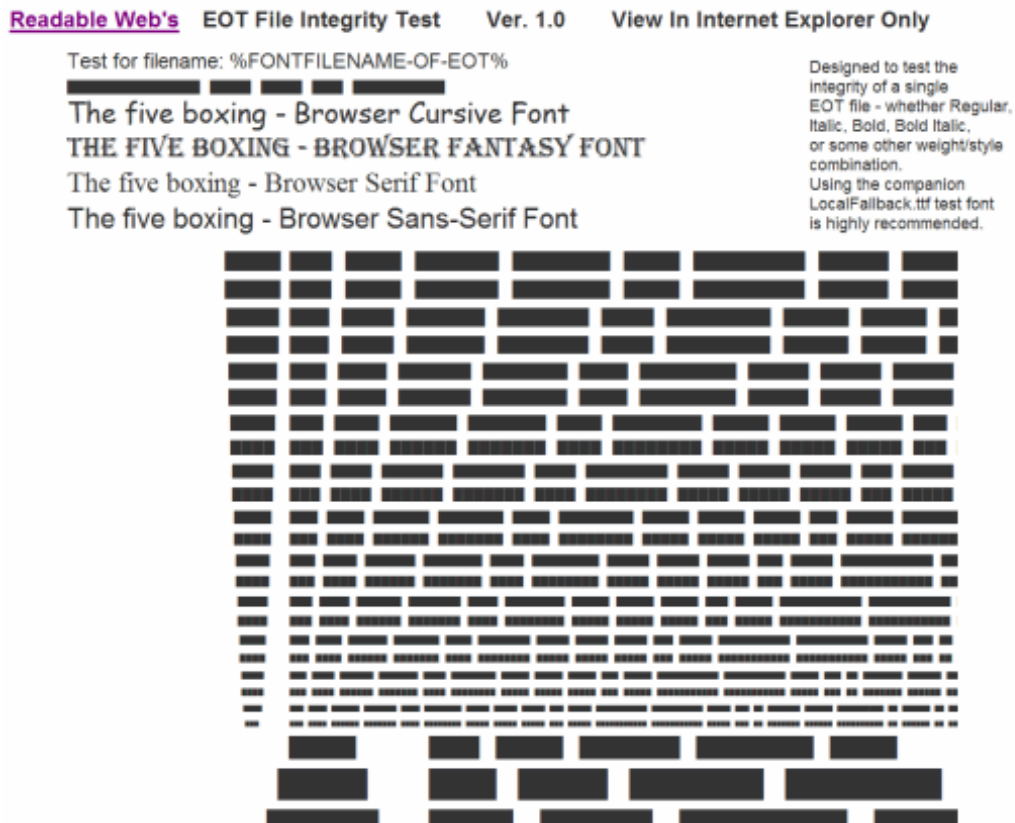
The LocalFallback Test Font – LocalFallback.ttf

Also in the package is a special test font designed to work with the File Integrity Test Page to make it instantly obvious if the EOT file is being successfully displayed by Internet Explorer.

When you begin working with @font-face, you’ll find that many fonts can look similar at first glance and it is very, very easy to get fooled into thinking that the external @font-face font is or isn’t being displayed. By installing LocalFallback.ttf in the operating system (just drag it into the Windows Font Folder) it’s easier to get instant visual confirmation that everything is OK with the CSS and that the EOT is being displayed by Internet Explorer.

For example, if the file you’ve converted has a problem or you didn’t enter the right file name and path, instead of seeing one of the browser’s “fallback” fonts and possibly coming away with the impression that you’re looking at “Franklin Gothic” instead of the fallback “Arial”, you’ll

instantly see there's a problem: the "box font" LocalFallback.ttf will reveal it instantly and looks as follows:



Oops. Lots of boxes. Pretty obvious the EOT font isn't loading.

What To Do When You Run Into A Problem

Here is a list, somewhat in the order of frequency at which problems with EOT files ordinarily occur:

- 1) **The Full Name (ID 4) entry in the name table must start with the string defined in the Family Name (ID 1) entry.**

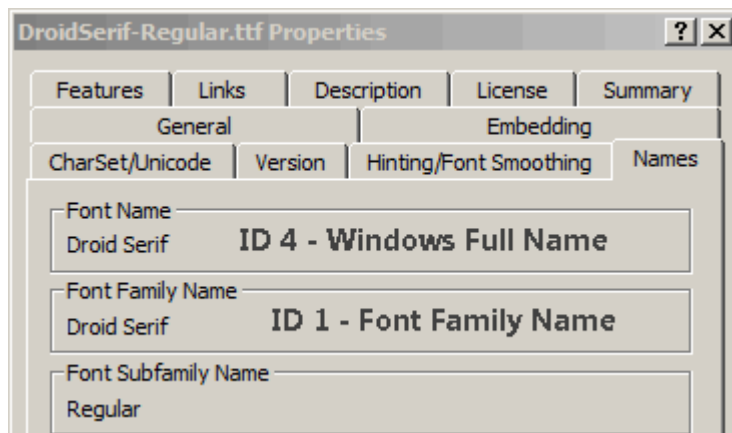
Here's what this means:

Inside the font there's a name table that has entries for the name of the font. But because of the history of font development – how computer fonts evolved – fonts have multiple names. Different names for different platforms and different formats. There are names that Windows uses to identify the font. There are names that Mac uses to identify the font. There are names

that particular applications use to identify the font. It's all very confusing but don't worry about it – here's what you need to do *if* you run into a problem with a TTF file that's misbehaving as an EOT:

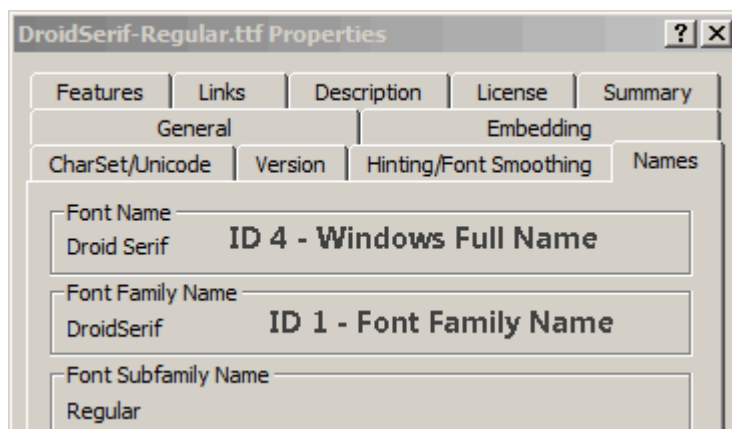
Once you've installed the **Font Properties Extension** and signed up for updates at <http://eotfast.com> and made a donation, (okay, okay, just install the **Font Properties Extension**) you can view the font names in the id tables of any font by right clicking and selecting Properties. One of the tabs will say "Names".

Like this:



Here, the Font Family Name (ID1) is "Droid Serif". That's the most important one because if the Font Name (ID4) doesn't begin with the string "Droid Serif" then a working EOT file can't be created. In this case, everything is OK.

However, if it looked like this:



Everything would not be OK. The Font Name "Droid Serif" does not begin with the string for the Font Family Name which is "DroidSerif". In order to make a working EOT file, this would have to be fixed. One or the other would have to be changed.

Fixing Names Using TTX

Fixing name problems like this is easy using the free TTX utility. After installing TTX, just drag and drop the TTF file onto the TTX icon and an XML text “dump” of the font will be created with a .ttx extension.

Open that up and search for “<names” to take you to the names table data.

There, you will see something like this:

```
<name>
  <namerecord nameID="0" platformID="1" platEncID="0" langID="0x0">
    Digitized data copyright © 2007, Google Corporation.
  </namerecord>
  <namerecord nameID="1" platformID="1" platEncID="0" langID="0x0">
    DroidSerif
  </namerecord>
  <namerecord nameID="2" platformID="1" platEncID="0" langID="0x0">
    Regular
  </namerecord>
  <namerecord nameID="3" platformID="1" platEncID="0" langID="0x0">
    Ascender - Droid Serif
  </namerecord>
  <namerecord nameID="4" platformID="1" platEncID="0" langID="0x0">
    Droid Serif
  </namerecord>
  <namerecord nameID="5" platformID="1" platEncID="0" langID="0x0">
    Version 1.00 build 112
  </namerecord>

  Etc.....
```

Just make the necessary changes to the names, save, and drag the .ttx file back onto the TTX icon to convert it back into a font file. And then you’re good to go.

Tips:

Usually all you have to do is add a space to one of the names or remove a hyphen from the name to get things working. (A lot of the time, the reason this happens is because the Postscript name for a font doesn’t allow spaces. And some fonts – especially OTF CFF fonts created less recently and then converted somewhere along the way into a TTF – will retain a “name soup” of Postscript/Windows names jumbled together. Font designers take note!)

Caution: Remember that web fonts are declared in the @font-face CSS as **Font-Families**. It might be a family with one member. (Easy.) It might be a family of four. (Easy still, once you know what to do.)

All the fonts in a family need to have exactly the same Font-Family Name.

If you are making changes to names, make sure the Family Name is the same across all the related fonts grouped together within the family. Make the accommodations necessary to create a working EOT file by changing the Full Name, **not** the Family Name unless you are prepared to change the Family name in *all* the font files grouped together in that family.

Of course, if it's a novelty font that's a family in and of itself, it will be a font-family with one member, and there's no problem. Just get the Family Name and Full Name to line up right and all will be well.

2) No name table entry should be larger than 5000 bytes

Finding name table entries larger than 5000 bytes is rare. And, actually, **most** installations of Internet Explorer **will** display EOT's that have this "problem". In other words sometimes it's not a problem. In fact, for *most* IE users it isn't a problem. But there's no way to tell where and for whom.

Here's what happened:

We all think of fonts as cute cuddly things, but the fact is, like any external resource, fonts can be used as an attack vector for malicious software:

- MS10-001: "**Vulnerability in the Embedded OpenType Font Engine could allow remote code execution**" <http://support.microsoft.com/kb/972270/>
- This security hole was patched around July 14, 2009:
<http://www.microsoft.com/technet/security/bulletin/MS09-029.msp>

But the patch caused an unwanted side effect:

- "**The wrong fonts appear in Web pages, in printed Word documents, or in printed PowerPoint presentations after you install security update 961371**"
<http://support.microsoft.com/kb/978909>

The first security patch caused EOT's with entries in the name table of over 5000 bytes to no longer work!

This was subsequently fixed six months later in a patch on January of 2010:

- <http://www.microsoft.com/technet/security/Bulletin/MS10-001.msp>

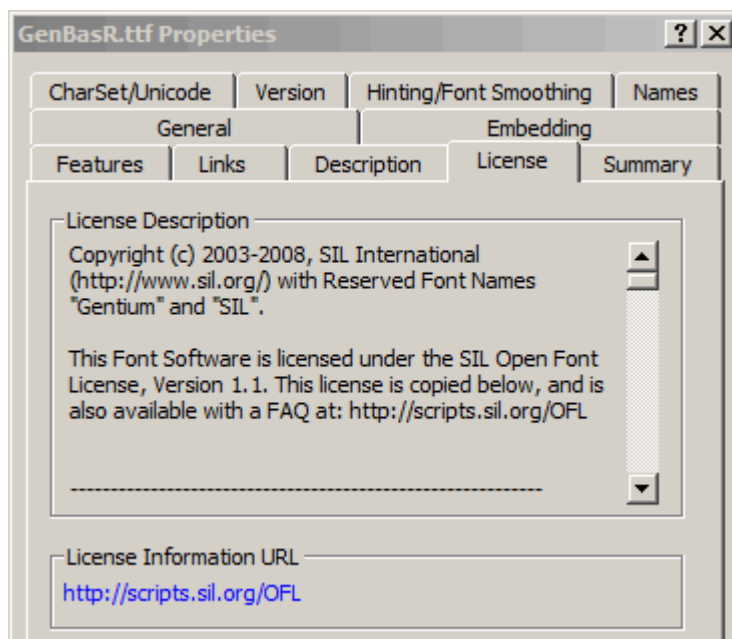
And so, those IE installations that received the first patch but, for whatever reason, not the second one six months later, remain broken and there is no way to detect one from the other. And with a billion Windows installations in the world, just a small percentage can add up to an awful lot of people.

As you might have guessed by now:

[In EOTFAST 2.0 name table entries of over 5000 bytes will be auto-detected and you will be prompted as to whether you wish to continue creating the EOT file or not. You know what to do.]

For now, the Font Properties extension allows you to visually check for this if you want.

When a particularly large entry in the name table occurs it's usually the license info. For example, the quite excellent open-source font Gentium from SIL includes the entire SIL license in the name table instead of just a link to the license:



As you can see from the scrollbar, the license goes on and on.

Checking for entries over the 5000kb limit and fixing it or not is up to you. And certainly, the size of the pool of users it can possibly affect will fade over time.

As with naming, TTX will do the job if you choose to fix.

A little news story about it, here: <http://blogs.zdnet.com/security/?p=5207>

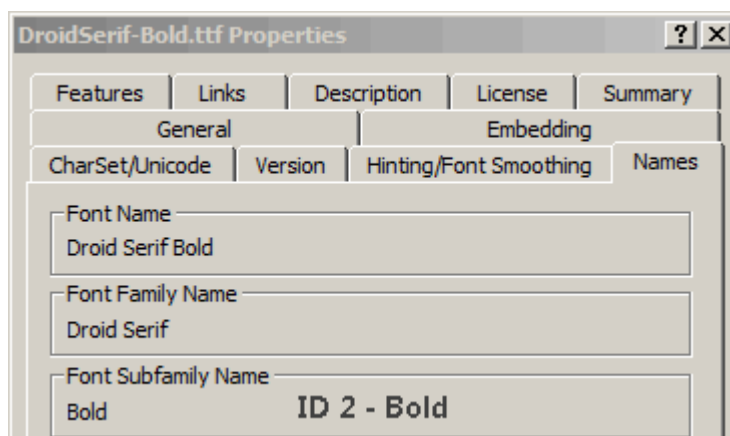
Note For Font Designers: On the web, page load time and therefore file sizes are a big concern. We urge you to keep extraneous information inside the font file to a minimum. Usually a simple link will do. We urge you to bear this in mind when offering fonts for use with @font-face.

- 3) If you are declaring the font as part of a larger Font-Family, the same style and weight as you are declaring for font-weight and font-style in the CSS needs to be defined inside the font, since IE 6, 7, and 8 will read that font data – and possibly override the CSS weight/style settings.

Once again, this isn't usually a problem. (But there are tons of free fonts floating around on the web and anything can happen.) And it's only a *potential* problem when you're declaring the font as part of a larger family with more weights and styles than Regular – like Regular *and* Bold, *and* Italic, etc.

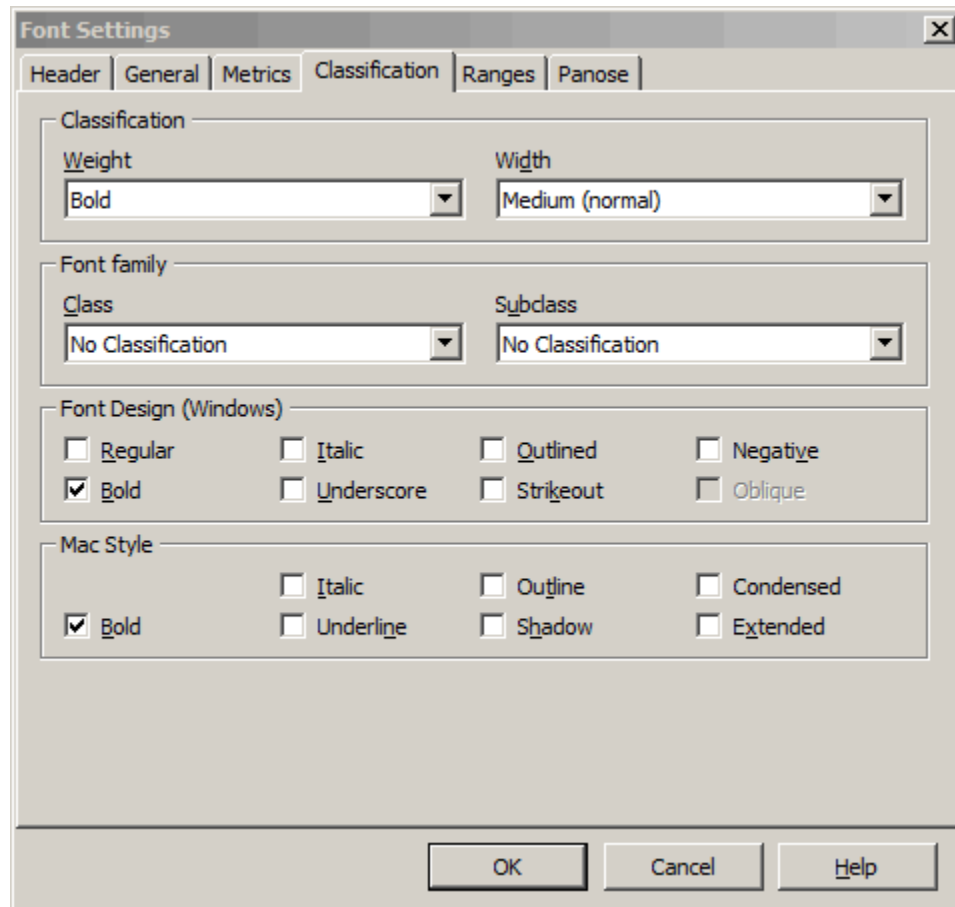
IE 6, 7, and 8 won't rely on the CSS @font-face declarations alone to decide which members of a font-family are Regular, or Bold, or Italic, it also **looks inside the font** to see how the font declares itself, and the settings inside the font need to match up with the CSS, that's all.

For example, if you are going to be using the font as the Bold member of a family, make sure the Sub-Family is defined as Bold in the name table.



Further – make sure the Subfamily name matches up with the **Classifications**. This will, unfortunately require a font editor such the free **Font Forge** or **TypeTool** (retail) or **FontCreator** (retail).

You can see the font's **Classification** settings in the following screen grab from FontCreator:



At the very least – as a matter of good practice in the interest of avoiding confusion - the classifications should match up with the Font Sub-Family name.

Note: This is a bit advanced and more for type designers looking to market for the web but if, for some reason, you wished to combine different-looking fonts to make up a custom "font-family", you would almost certainly have to change the style and weight data to make it happen cross-browser to include Internet Explorer 6, 7, and 8.

In a conformant CSS3 implementation you don't have to do that - you just have to declare the fonts as members of the same family and set the font-styles/font-weights in the CSS @font-face declarations. But as long as IE6, 7, and 8 have a user base, there's the chance you might have to go into the font data.

4) The “Embedding Bits” (OS/2 fsType) Are Set To “No-Embedding.”

The OpenType specification includes data that is designed for font makers to express their desired licensing policies for “embedding” into documents. These are commonly called the “embedding bits”. Firefox, Safari, Chrome, and Opera **ignore** these bits when deciding whether to load a font file. They are not a part of the CSS3 specification. The Windows Font API, however, takes these bits into account when allowing or not allowing a font to be embedded in a document – such as a PDF or a PowerPoint presentation - and this behavior is also found in Internet Explorer. If the embedding bits are set to “No-Embedding” within the EOT, Internet Explorer will not load it.

Should the need arise, the embedding bits in a TTF can be viewed and manipulated within limits using the free Microsoft **Font Properties** tool:

<http://www.microsoft.com/typography/property/fpedit.htm>

However, the tool will not change No Embedding to something less restrictive. For that you must turn to a font-editing program.

5) The Font Family Name Can't be Longer Than 31 Chars Including Spaces

It's **extremely** rare for a font name to be longer than 31 characters. If you're having a problem, this probably won't be it.

Once again, if necessary, the font name can be changed with TTX.

6) The Font Must Be TrueType flavored, not CFF.

OpenType fonts come in two varieties. Adobe's OpenType CFF and Microsoft's TrueType.

Only TrueType (.ttf) files will successfully convert to EOT files.

Note: Simply renaming an .otf file with the extension .ttf will not work!

That's it! As the use of @Font-Face continues to grow on the web, we hope you find EOTFAST a highly useful tool. And please subscribe via RSS to **Readable Web** at <http://readableweb.com> and check back at <http://eotfast.com> for more information, tips, and tutorials as time goes by and typography on the web continues to evolve.