

The quick brown fox jumps over the lazy dog

Where are web fonts at?

Ben Weiner¹, 18 January 2010

¹ <http://readingtype.org.uk/about>

Intro

Who am I, and what qualifies me to talk tonight?

I'm a web developer and information designer. I studied in the Department of Typography and Graphic Communication at the University of Reading, but I was already sold on letters, printing and graphics in general before I started in 1996. Like a lot of people, I have been waiting for the chance to specify the fonts I liked within web pages for a long time, so when I heard that this might finally be possible I started to pay attention.

Tonight I'm going to take you on a short and hopefully interesting ramble through web fonts. The idea is to help you decide which of the various techniques either current or nearly current are worth your attention. Along the way we'll get some insights into what constitutes a font, wander about on the edge of free culture and the licensing jungle, and confront some of the pitfalls in current implementations.

If you've been tracking the web fonts question for any length of time you may well have come to the conclusion that it's a hopeless mess of incompatible confusion. By the end I think we'll be enthusiastic about what's coming up and the benefits that could result, even if we can't deny the mess.

Anyway. We should start by looking at what a 'web font' is, and then we can begin our tour.

What is a web font?

I guess the obvious definition is this: a web font is *anything recognisably a font that's transferred over the net before being used to render text in a page.*

When the first proposals for web fonts were emerging, people got quite excited. On their dedicated typography site, Microsoft said: 'This technology is changing the look of the Web, by empowering site designers to ensure their pages appear as they want them to.'² [SLIDE 2]

I don't know when that text was written but I would guess it could be any time from late 1997 (when Internet Explorer 4 was released) or thereabouts. So, it's 2010, twelve years have passed: who here's using web fonts?

And, do you feel *empowered*?

What is a font anyway?

So how many type nutters are in the room?

Very crudely, a font is *a computer file which contains the information needed to create images of letters*.

As a type nut I find that definition fails to convey some of the more intangible qualities of a font. But more importantly it fails to convey some of the essential ones too.

The essential qualities include not only the outlines themselves but the fact that each one relates aesthetically to all the others: the style and proportions are assumed to be complimentary so that the letters, however they're ordered, work well together. It's easy to overlook the fact that part of 'working well' is having the right space on all four sides of each character: top and bottom, left and right. [SLIDE 3]

The intangible qualities of a font include the discrimination of its designers, of course. Thousands of hours of creative work go into the production of typefaces, and drawing clear distinctions between what's governed by taste and what's just grunt-work is sometimes hard. For example, the spacing I just mentioned. It's not very interesting to spend hours adding or subtracting tiny bits of space from the left or right hand side of selected lower case letters, which might imply that this is grunt work, but the final decision is governed by visual judgement as much as by measurement.

There are a few other things about font production which it's easy to forget.

Firstly, there is the variety of character forms. Here are some examples.

- For latin type, there's a need for the many *diacritical marks* (that's accents to you and me). [SLIDE 4] We don't use them much in standard written English but lots of other languages do; you can go westwards to Wales or eastwards to France to get started, although in truth they don't make that much use of diacritics either.
- Then there are *ligatures*, which are two or more letters melded together for aesthetic reasons. [SLIDE 5]
- And there are the all the scripts that aren't Latin, which have their own special characteristics. [SLIDE 6] Arabic typefaces have a stand-alone version of each character just like latin, but they will also have *joining forms* for most letters in the alphabet. These differ depending on which side or sides of the letter are joined up to the letter next door.

Secondly, after the character outlines are correct and the spacing satisfactory, there's a need to trial the font as heavily as possible and, for the last twenty years at least, to provide *hinting instructions* that are used by the text rendering system built into operating systems.

And that is to ignore the fact that many fonts come in *families*. If you're a word processor user you may think families of fonts always have regular, bold, italic and bold italic. If you've got print design experience you will know that the desirable font families may have more than three times that many. [SLIDE 7]

My point is merely that this all adds up to a fair bit of work.

The history of web fonts

So we have these things called *fonts* which are aesthetic marvels and crafted with the care of a swiss watch, and we have the concept of *web fonts* as fonts we request over the network and use to render text in a browser window.

Why do we want to burden the network with these extra requests?

The massive majority of web pages today are making use of a remarkably small set of fonts. The first version of HTML didn't even include the [SLIDE 8] `` tag that would soon become common across the web. Browser software picked up a font file for displaying text from the selection available to the operating system. That font was most likely held in local storage on the computer's file system.

The original notion of a web site was really as a remote filing system; home pages provided lists of links that themselves yielded documents in the driest sense of the term. The typographic needs of the original users were not necessarily modest – physicists like those at CERN make ambitious demands on typography – but there was no need for the arsenal of typefaces used by professional designers.

But as soon as the web was a mass medium, it was a sure thing that the designers would move in. I was one: as an undergraduate tinkering with Netscape 2, I longed to be able to bring the skills I was learning in print design to this incredibly immediate medium.

Microsoft to the rescue

Designers brought with them a natural desire to be able to control *visual formatting*: the layout of the text and the font in which it was displayed. As far as layout was concerned, the answer was positioning through Cascading Stylesheets. That's a story in itself.³ But for type, the provision of flexibility was more difficult, because any system that allowed lots of different typefaces either required all users to have a large number of fonts or required also a way to give all users access to just the specific fonts for a given web page.

It was essentially the action of Microsoft that provided the first practical solution. A group of core web fonts was distributed with Microsoft's browser software, and by virtue of the reach of Microsoft, those fonts achieved a very high penetration. For a few years Microsoft even published the fonts on its typography site for free download, and these versions which are now much less well-featured than the ones included with current Windows are still legally available through third-party download sites.⁴ [SLIDE 9]

(Does anyone know the name of one of these fonts?)

But this group of half a dozen or so fonts was a drop in the ocean. Print design is a never-ending festival of typographic variety. [SLIDE 10] Graphic design on the web, even augmented with half a dozen good-quality fonts, was miserably second rate.

3 According to Wikipedia's article on CSS co-author Håkon Wium Lie, his thesis provides much background

4 Including SourceForge. Possibly the hardest way to get them: <http://corefonts.sourceforge.net/>

The first attempt at web fonts

As noted, the W3C web fonts effort formed part of the work on cascading stylesheets. As its early memo noted: 'W3C wants to ensure that graphical web clients have the necessary font resources to make rich presentations in many languages'. You can still read the W3C's first roundup of contenders from 1996.⁵ At that point nobody knew how to achieve such an aim.

A web fonts draft document dated 2 September 1999⁶ shows that amongst the authors were representatives of Adobe, Agfa and Bitstream: three font software vendors, but with very different positions in the software market. As noted on W3C's site, the outcome of this work was incorporated into CSS version 2, which introduces the notion of the font description as 'the bridge between an author's font specification and the font data'.⁷ [SLIDE 11] It's a nicely semantic system.⁸

The new technique was format-agnostic. The authors suggested that 'well-known formats' might be included; they named the vector formats PostScript Type 1, TrueType and OpenType.

Independent efforts at a suitable font format were made by Microsoft and Netscape in the late 1990s, as part of the 'browser wars' when these two companies slugged it out in the contest for market domination. They both rejected the idea of using any desktop font format, and it is likely that in both cases that was because they'd had conversations with font software vendors who said they'd never allow anyone to send a font file to a browser because it would immediately be copied by the recipient and passed on to family and friends. The respective web fonts formats were these:

Microsoft had Embedded OpenType (EOT), from IE 4.⁹ EOT did broadly do what it said on the tin, in that the files were very much OpenType files. OpenType is still in 2010 the prime font format. The inclusion of the word 'embedding' was curious: I don't believe that files are often 'embedded' into HTML (except that they can be base-64 encoded into URLs, and hence either into an HTML or a CSS file). But I think Microsoft wanted to create a perceived connection between the new technique and the existing ones for embedding fonts into PDF files and Word docs that were known to font software vendors.

Netscape Corporation had Portable Font Resource (PFR) from Communicator 4. PFR files were decoded by TrueDoc, a system that was invented by Bitstream. TrueDoc was intended to be licensed widely, with free plug-ins including one for Windows Internet Explorer 4 which was announced, but in the desktop market the system vanished quickly.¹⁰

5 Strictly special interest, and most links have rotted entirely away. <http://www.w3.org/Fonts/>

6 <http://www.w3.org/TR/1999/WD-font-19990902>

7 <http://www.w3.org/TR/1998/REC-CSS2-19980512/fonts.html#font-descriptions>

8 There is a lot more semantic stuff in the following section on font characteristics but it is a bit scary.

<http://www.w3.org/TR/1998/REC-CSS2-19980512/fonts.html#font-descriptors>

9 Until recently, and because they were part of a DRM system, EOT files could only be made using a Microsoft tool, 'WEFT', that was allegedly prone to fail to work. Following some work done by the Webkit project you can now get a command-line tool, ttf2eot, at <http://code.google.com/p/ttf2eot/>. The latter ignores the whole business of 'rootstrings' – intended by MS to tie the EOT to a single internet domain – effectively making the output files 'Compatible web type' or 'EOT Lite'. Ascender Corp also lets you make them via a web app.

10 Evidence of TrueDoc seems to be getting scarce but I found a PDF that is no longer available from Bitstream announcing the availability of the IE plugin. Nokia licensed TrueDoc for text display in their TV set-top boxes.

Failure.

It should be evident that these techniques both failed to take root. Netscape 4 in its entirety was a bit of a disaster area and plug-ins are perhaps not the best way to handle something as intimate as the text rendering in a browser. I do not know whether it was ever possible to obtain the TrueDoc font conversion software announced by Bitstream. Embedded OpenType is reported to have had some takeup in parts of the world where Windows had no fonts covering the local script, but I have no hard evidence of that.

What killed them?

To send a font file to another person's browser without breach of copyright, you need a licence. Contemporary font licences did all they could to rule out any such possibility; font software vendors have always been vulnerable to the fact that it's extremely easy to copy font files. So lack of support from font software vendors throttled the legitimate supply of fonts at source. Probably the involvement of Bitstream and Microsoft who are font software vendors in their own right did not encourage other foundries.

And the two solutions were entirely different so, unless TrueDoc had triumphed, each could have required a separately licensed font file. Remember, they were ideas cooked up at the height of the browser wars.

Web fonts in abeyance: cue the hacks

As far as the internet world in general was concerned, neither EOT nor PFR ever really happened. But designers of more or less ability now had the web by the scruff of its neck and they were not minded to let the niceties of font licensing stand in the way of typographic variety. So the stage was set for the decade of type hacks.

Stupid [SLIDE 12]

Firstly there were *stupid hacks*. Essentially pictures of text, an image used instead of a string of characters. These leave anyone who cannot see the picture with no notion that there was any attempt at an act of communication: sometimes there is some text in the 'alt' attribute of the image but this doesn't really match the mentality to which the technique appeals. I found a web site entirely constructed in this way only a few weeks ago, but it's getting quite rare.

I'm going to lump Flash-only sites into this category too, which is a bit unfair, but that's tough. I don't have any experience of using a screen reader to read a Flash web site, but I have tried in Lynx and I didn't get very far. And I also think that Flash driven sites are less common than they were, so let's hope this is the last time we have to mention them.

(Has anyone here heard of the *semantic web*?)

Less stupid [SLIDE 13]

After the stupid hacks had generated a righteous howl from the accessibility advocates, who at that time were just getting into their stride, we had the *less-stupid hack*. This is Fahrner image replacement, which is pictures of text with the text still in place. It's labour-intensive and a great way to create cockups when the copy gets changed but the image does not, but that all adds to the fun.

Clever [SLIDE 14]

Still, time moves on, and eventually in about 2005 we get to the first of the *clever hacks*: sIFR¹¹. sIFR was the first mainstream technique that works along the grain of content management, which is a reflection of the fact that one feature of 'web 2.0' was that web *users* were becoming both *readers* and *writers*. This meant that the text on a page was increasingly likely to change unpredictably – and the change would take place long after the stylists and the web monkeys had finished building the page and gone home to tea.

sIFR doesn't leave the hacked text high and dry when its content changes. It works at the point when the page is shown in the user's browser. It uses a resizable Flash movie, containing a font, to typeset pre-flagged HTML elements. It takes care to do so in a way that preserves the original text for those who might not have the capability or the desire to see the excitingly distinctive font.

11 <http://www.mikeindustries.com/blog/sifr>

Cleverest [SLIDE 15]

The cleverest hacks also respond to the need to be accessible and to be compatible with a read-write web. But they do so without recourse to Flash; instead they take a more elemental approach, retrieving the outlines of the fonts on a web server and then drawing the outlines directly onto the browser window: techniques unthinkable a mere few browser versions ago.

Cufón¹²

Arriving at Cufón's web site, the visitor is redirected straight away to the 'Generator', a service that gift-wraps type ready for immediate use. As we will see later, this is a web font strategy that seems to be growing in popularity. Possibly that teaches us something about the kind of web developer likely to make use of such services.

The form that you use to interact with the Generator is interesting in its own right: full marks for *subsetting*, to reduce font file size; for having the option to *reduce the fidelity of the outlines*; for allowing *domain restrictions*; and for sticking in a tickbox that the user must tick to say that they *have the right* to 'Web Embed' the font files they've chosen.

The output is a rather natty Javascript file, which contains the font outlines encoded in JSON. Stick the URL for that file into script tag, add the Cufón javascript engine in another script tag, tell Cufón the elements you'd like it to replace, and off it goes.

typeface.js¹³

The server internals of typeface.js, which are described on the site, are Perl scripts making use of the Freetype library to derive character outlines from original font files. Like Cufón there's a service to do this for you – or you can grab the scripts (and a copy of the Freetype library if required) and run the whole thing yourself.

The outlines are encoded in JSON – the same tactic that Cufón adopts. Also similarly it is possible to sub-set the characters that are encoded in the interest of delivering typeface outlines to browsers more quickly.

When it comes to applying the outlines, the tactic is very different. Cufón wants to handle this according to its own ability, so the developer's job is to instruct Cufón in Javascript about which DOM elements should be attacked. typeface.js plays by the rules of CSS.

12 <http://cufon.shoqolate.com/>

13 <http://typeface.neocracy.org/>

So why use the cleverest hacks?

Well, while we can't get the fonts we want then we have a problem; these hacks are very clever, they're available here and now. And they don't require Flash.

The drawbacks of the cleverest hacks

You'll have to take it as read that I think these systems are both great. They certainly impress. My confession at this point is that never in the time since I first came across them have I been minded to try them out, so the trials I did for tonight represent my only experience of them: I don't know whether there are more problems that might turn up when using either system in the field.

I applaud the pragmatism that lies behind any project that seeks to get around the status quo. I mentioned the deadlock that emerged at the end of the 1990s: the technological legacy of this situation, as with so much in the domain of web page layout, is with us still.

But there are some drawbacks. Some are about implementation techniques and some are to do with the nature of typography.

Implementation

Firstly, implementation. Although browsers do still disagree about all kinds of things, there is agreement that the way to specify the fonts you want to see used in a page is through cascading stylesheets. Cufón introduces its own method, and that's a bit naughty.

Not all browsers can handle the two systems, but rather than sIFR's use of Flash, a plugin that draws protests from certain quarters, these systems both use techniques that are relatively recent in their introduction into browsers. Nevertheless, considering only desktop browsers, we find that typeface.js can claim support for Firefox from version 1.5, Safari from version 2, and Internet Explorer from version 6. To block either of the two clever hacks, you could of course switch off Javascript.

Typography

Secondly, typographical drawbacks. I mentioned the complexity of scripts other than the latin most people here habitually use. If you remember, there are scripts in which letters change their shape depending on their position in a word, like Arabic; and indeed there are latin typefaces in which the letters may change shape or join with others to form ligatures. How well do Cufón and typeface.js handle these challenges?

I ran Cufón and typeface.js against the GPL Arabeyes typefaces Kacst Poster and Kacst Art¹⁴, which are quite obviously distinct from the operating system Arabic font on my computer, to see what happened.

The result was very interesting. Cufón, sadly, can't be recommended. It renders disconnected characters, effectively gobbledigook. [SLIDE 16] But typeface.js passes my test. [SLIDE 17]

I then repeated the tests using a latin-only typeface with a tasty ligature joining two lower-case Fs and an I. Here, the glyph substitution was flunked entirely by Safari on every occasion. Using Firefox, I found that again Cufón could not achieve the substitution [SLIDE 18] whereas typeface.js could. The ligature is an OpenType substitution of a kind that only Firefox (and it appears Konqueror) currently support. [SLIDE 19]

I wonder whether typeface.js works better because it uses the CSS font-family property. Whatever, I hope this is a good first-round result for people to take away tonight. If you're intending only to run headlines in Latin script – which is probably what the manufacturers intended – then you're pretty much home and dry.

But...

One characteristic that all these techniques from the stupidest to the cleverest share is that none use 'real fonts'. That's the preserve of the final technique I want to discuss this evening, the one that has the official sanction of being a genuine, fully-fledged W3C recommendation.

Font linking with the @font-face at-rule

@font-face is the CSS at-rule that uses a font resource – that is, something other applications would recognise as a font file, rather than a flash movie or a chunk of javascript. The very same technique described so long ago in CCS2, the one that IE4 and Netscape 4 both attempted to implement.

The proposal is very straightforward.

Stick a font file – let's say an OpenType font file – onto a web server.

Then use the @font-face rule to associate this file's URL with a name and weight. Reference it exactly as you would a locally-loaded font file. Use the font-family property, a method which has been familiar to CSS authors for the last decade. Here's an @font-face rule defining a font family and associating a resource with it

[SLIDE 20]:

```
@font-face {
    font-family: 'CantarellRegular';
    src: url('Cantarell-Regular.eot');
    src: local('Cantarell Regular'), url('Cantarell-Regular.ttf') format('truetype');
}
```

Some points to dwell on:

- The font-family name can be whatever you want. Historically this was important because Windows does families of fonts that have regular, italic, bold and bold italic and it goes nuts if you add more than four members to your font-family
- Note the 'local' resource location included, in which you can specify the font's name as it would appear in application menus if you had the font installed locally. This provision allows you to save a little bandwidth by telling the browser to ask the OS to have a rummage for the font before starting a download. On a 10.6 Mac, should the file be floating around in your downloads folder, you'll be asked whether you want to use the downloaded file.¹⁵ [SLIDE 21]
- The *format* information is handy. User agents are supposed only to download the font format that they can use. We'll see the format issue crop up later.

Once your at-rule is in the bag, give the remotely-loaded font a workout simply by specifying it with a font-family property exactly as you'd expect [SLIDE 22]:

```
p {
    font-family: "CantarellRegular";
}
```

And you can always back this up with a web-safe font stack of course [SLIDE 23]:

```
p {
    font-family: "CantarellRegular", Arial, sans-serif;
}
```

I can still remember vividly my delight on getting this up and running with a copy of Safari. It was such a tiny, tiny difference in technological terms but it meant so much to know that finally the opportunity had arrived: use the right font for the job in hand, not simply a font that shares passing similarity.

Here are some real-world sites using @font-face:

- The NY Times 'skimmer'¹⁶ [SLIDE 24]
- A page design of mine, complete with a FOUT¹⁷ [SLIDE 25]

15 You do not have to specify a local font name. It's vulnerable to operating system differences and could be tedious to research. If the font is commercial few users will have it installed.

16 <http://www.nytimes.com/timeskimmer/>

17 <http://stbride.org/events/musicanddesign/programme.html>

The advantages of font linking

I'm going to jump up and down a bit here, because the thought of banishing hacky solutions to this problem makes me very happy. It boils down to the sheer straightforwardness of it. Using *actual fonts* when you want to send *character outline data* to a program that *understands how to use fonts already* seems like a good way to avoid duplication of effort. So, yay! [SLIDE 26]

Also, we're going to need a bit of enthusiasm to carry us through the gnarly stuff I will be covering shortly.

Above all of course, and particularly as we're at a meeting of London Web Standards, I must restate that this technique is legitimised by being exactly that: *a Standard*.

Some more things you need

We've looked at what you need at the front end: a light dusting of CSS and you're done. But is there an impact on the server itself?

Firstly, you do not need anything clever in the way of server scripting.

Secondly, you're going to be serving fonts, so you'll need some of those. Let's imagine that the supply of fonts is not a problem, because I want to talk about licensing considerations separately.

Then you might consider adding some *access restrictions* to the web server – typically, in an Apache virtual host configuration file, you would set up referer checking as arguments to the Deny/Allow access directives.¹⁸ Why? Well, you might not wish to allow other people to borrow your bandwidth to decorate their pages with fonts you chose for yours. Your licence for the font might require such a provision.

If you want to use fonts hosted on one domain in pages that are under another, you might also need to send a specific, currently non-standard, header out with your font files. The AJAX community has been working on a new technique that overcomes the security restrictions built into XMLHttpRequest. This standard has been co-opted by Firefox's web fonts people because they want to be able to build the same kind of trust around remotely-loaded fonts as there are around javascript. I think we can expect this to become the norm because maliciously-crafted font files are capable of causing trouble, even though the payload in a font is less apparently dangerous than Javascript.¹⁹

The technique I'm referring to is known as *Cross-Origin Resource Sharing*. And the header you need to add is 'Access-Control-Allow-Origin' [SLIDE 27]. As far as I know, it's only Firefox that has implemented the technique for web fonts to date.^{20 21}

With your @font-face font linking CSS, access control headers in place and your server loaded up with suitably licensed fonts, you're ready to go.

Well, you are really ready to go, at least in theory, on the following browsers: Chrome, Firefox, Konqueror, Opera, Safari and of course Internet Explorer.

18 <http://httpd.apache.org/docs/1.3/misc/FAQ-G.html#image-theft>

19 Indeed Google have recently identified such a vulnerability in the Windows t2embed software library, allowing an attacker to 'take complete control of an affected system' through the decompression of a sneaky EOT file. <http://blogs.zdnet.com/security/?p=5207>, <http://www.microsoft.com/technet/security/bulletin/MS10-001.msp>

20 http://openfontlibrary.org/wiki/Web_Font_linking_and_Cross-Origin_Resource_Sharing, <http://hacks.mozilla.org/2009/07/cross-site-xmlhttprequest-with-cors/>

21 This is not DRM. It is a way for the browser to use information from the web page and the server to decide whether the resource is safe to fetch.

More good things about web fonts to make us happy

Firstly, to reiterate, this is it: the real deal. Actual font files. We waited almost decade for this simple, intelligent and obvious solution to come along! Celebrate this remarkable breakthrough if you celebrate nothing else tonight.

Secondly, because this is Real Type, not cleverly repositioned graphics where real type should be, the questions of text layout, character shaping, and even, dare I say, *advanced* typography are firmly back with the browser and not part of a third-party javascript library however devilishly clever. That's good news. Browsers historically have followed the example of the Web's Founding Fathers: there shall be bugger-all in the way of typography. Browsers don't hyphenate. They're barely capable of justified setting. You might think Donald Knuth²² had never lived. But that is going to change, and there's a great demo out there already, implemented in a branch of what used to be known as Firefox 3.7.²³

Using the typesetting capabilities of the browser is especially important when you consider that the formatting requirements of English text are so very basic compared with many of the world's scripts, and you think about the number of people who use these scripts. From a quick survey of Pakistani online news sites in Urdu, it seems that the majority of these output all their copy as bitmap image files. It'll be a struggle to syndicate those. The BBC's Urdu site is real text, but Safari on my Mac and Konqueror on a Kubuntu VM both struggled to render it with fonts that may or may not be legible to Urdu readers. Internet Explorer 8 did much better. [SLIDE 28]

If you believe that access to good-quality information can transform people's lives then good-quality web fonts – carrying the complex information to enable typesetting within them – are an essential aid.

And thirdly, this whole thing – and the interest it is generating in the web design community – seems like a great opportunity for font software vendors. They have a big new market.

22 <http://www-cs-staff.stanford.edu/~uno/>

23 <http://hacks.mozilla.org/2009/10/font-control-for-designers/>

Cue the ideological problems: licensing

Oh, but wait: the foundries still aren't playing ball! They certainly managed to annoy Mark Pilgrim, who put his argument, and his sentiments, very eloquently.²⁴ [SLIDE 29]

Mark saw the foundries' position as supreme dogs-in-the-manger, totally set on denying access to their goods because they were afraid somebody might rip them off.

You may remember that it was Microsoft and Netscape who introduced the first, incompatible web fonts implementations back in the days of the browser wars. It has long been Microsoft's stated aim never to hurt the foundries by providing a mechanism that might allow regular desktop operating system fonts to be used for web fonts. Hence Microsoft's implementation of web fonts has no support for OpenType, TrueType or PostScript.²⁵

For their part the foundries have to my knowledge made few if any sales of EOT fonts in the last decade. So they viewed Safari's support for OpenType and TrueType without enthusiasm: it looked like a way that people could get around their licensing with little or no effort, and their lack of experience in generating revenue from this market left them with no business intelligence.

To date, I believe *no long-established foundry* is prepared to admit to licensing commercial paid-for fonts in OpenType or TrueType format for use as web fonts. I do have some contradictory evidence, though this in itself changes nothing. [SLIDE 30]

24 <http://diveintomark.org/archives/2009/04/21/fuck-the-foundries>

25 Internet Explorer only supports Embedded OpenType, a proprietary, patent-encumbered, closed font format that tries to achieve 'digital rights management' – an objective that doesn't fit the requirements of a W3C standard.

Then the technical problems...

Leaving aside the licensing question, I have to report also that not everything about the software support for web fonts is great. Here are some issues.

Bandwidth ...

Adding web fonts can significantly change the bandwidth profile of a site. I'm going to make a claim that I cannot illustrate, but I guess the average file size of a latin font as about 35K. And there is a file for each variant: regular, italic, bold, bold italic; and once the typographers come in there will be condensed, expanded, super-light, ultra-bold, small caps, and so on. You can count on it.

I said 'average size': that's the average size of an average print designer's latin-alphabet font file. OS font files like Microsoft's Arial Unicode or Bitstream's Deja Vu are over a megabyte in size and the whole file is needed before any text can be rendered. If the license allows it, you can subset the font. If not, you might want to think again.

Browsers differ in how they handle delays in downloading font files, leading to 'flashes of unstyled text' or 'unfonted content'. **[SLIDE 31]**

Opinions are divided about how the missing font file affects page rendering. Firefox will display a fallback font while it waits for the file to appear, while Safari simply shows no characters at all until the font's been transferred. The Firefox approach can be a little disruptive because if the character widths in the fallback font are very different – which is extremely likely – the text will be totally reset when the desired font is loaded and all the line breaks will change. It's a bit like trying to pull out the tablecloth without disturbing the place settings. Safari's approach just makes the user drum their fingers on the table, which is a known browser usage trope.

... and family ties

Fonts come in families, and the members of the family provide features like different weight or 'boldness' and different slope or 'italicisation'. It's very easy to get web font-supporting browsers to do the right thing when you specify a font family with a single member; there is little opportunity for things to go wrong. But with a regular and an italic, for example, it's a bit more tricky.

This is how a family of four fonts should be specified with `@font-face` at-rules²⁶ [SLIDE 32]:

```
@font-face {
  font-family: "DejaVu Serif";
  src: url("/fonts/DejaVuSerif.ttf") format("TrueType");
  font-weight: 400;
  font-style: normal;
}
@font-face {
  font-family: "DejaVu Serif";
  src: url("/fonts/DejaVuSerif-Italic.ttf") format("TrueType");
  font-weight: 400;
  font-style: italic;
}
@font-face {
  font-family: "DejaVu Serif";
  src: url("/fonts/DejaVuSerif-Bold.ttf") format("TrueType");
  font-weight: 700;
  font-style: normal;
}
@font-face {
  font-family: "DejaVu Serif";
  src: url("/fonts/DejaVuSerif-BoldItalic.ttf") format("TrueType");
  font-weight: 700;
  font-style: italic;
}
```

It would be great to report that this system was understood and working well. I am seeing many instances where Chrome and Safari do one thing, usually in agreement with Firefox, but Konqueror and Opera each go their own way. Most frequently they seem to be confused about whether it's a bold, italic or regular family member they're supposed to be using.

And guess what: Internet Explorer doesn't follow it at all! The Redmond interpretation of `@font-face` really seems to represent a determined effort not to understand. What happens is that IE only understands these two properties [SLIDE 33]:

```
font-family: "DejaVu Serif";
src: url("/fonts/DejaVuSerif-BoldItalic.ttf");
```

So when you mark out bold and italic with `font-weight` and `font-style`, what IE sees is repeated re-defining of the `font-family` to map to a different URL. Commenting out the properties IE ignores, you can see that the last at-rule countermands the first. In fact it countermands all the other rules for the same `font-family` [SLIDE 34]:

```
@font-face {
  font-family: "DejaVu Serif";
  src: url("/fonts/DejaVuSerif.ttf");
/* font-weight: 400; */
/* font-style: normal; */
}
. . .
@font-face {
```

26 There are other properties, but they will add little until browser/OS typography improves

```
font-family: "DejaVu Serif";  
src: url("/fonts/DejaVuSerif-BoldItalic.ttf");  
/* font-weight: 700; */  
/* font-style: italic; */  
}
```

You note that I've left out the format hint because when IE sees one of those it rejects the at-rule altogether.

As you can imagine this means that if you wish to use webfonts and include IE, you need a totally different and rather complex set of @font-face rules which depart significantly from the specification. You need a different font-family for every single member of the typeface family.

What's being done to sort things out?

The teething problems are attracting a fair amount of attention. While the browsers have yet to catch up with the bug reports, bands of mercenaries are coming up with more or less inspired work-arounds.

Fixing the small problems

For instance [this text needs checking] the flash of unstyled text can be mitigated by adjusting the order in which fonts in the @font-face rule are listed, or by giving the DOM a bit of a kick. See Paul Irish's blog post²⁷ for some help.

Fonts – in common with other resources – can be compressed on the server, which may make them a bit leaner. They can also be sub-setted, but that is only technically allowed if your licence permits it. It also looks a bit odd when somebody comes to leave a comment on your blog that includes a character you filtered out of the subset. **SLIDE 35**

The problems of ensuring that the correct family members are selected seem more tricky. I think this is going to come down to concerted bug-squashing. It looks as if Firefox knows what it's doing; it looks as if other browser vendors are struggling more than they should with concepts like 'bold' and 'italic'. Fingers crossed.

27 <http://paulirish.com/2009/fighting-the-font-face-fout/>

Fixing the bigger problems

Right now there's no established business model for selling web fonts, or more precisely *licensing* web fonts. A bunch of start-ups and not-so start-ups are slowly rushing into this space. Meanwhile the users – us – cannot always have the fonts we would like because we cannot license them. I'll return to licensing in a tick.

Then there is the font format itself.

One of the great things about the @font-face proposal was its agnosticism about the format of the font files that could be used. This meant that X11 systems with whacky bitmap fonts could be accommodated in the same specification that allowed PostScript and TrueType. The unstated assumption was that whichever formats were used would already exist and be in use. This is a sensible approach: the problem of font file formats had already been addressed elsewhere and so there was no need to address it in the context of web fonts.

But as I have already said that's not the way font software vendors wanted it: they got terribly excited about the possibility that regular desktop fonts would be travelling from one person's computer (the server) to another person's (the browser). And it was essentially this fact that led to the proposal running aground when it was first written.

And then one significant browser vendor – Apple – simply picked up the proposed standard and implemented it in the spirit it was written.

It makes font software vendors exceptionally unhappy that anyone should create a mechanism that might, possibly, allow people another way in which to breach the terms of their licence. It makes them so unhappy that they refuse to consider offering such licences to people who *want to pay* to host fonts and serve them to browsers. But then also remember that Apple is the operating system vendor of choice for designers, that Safari is available on Windows, and also that Apple is a font software customer and reseller in its own right. So things were cooking nicely once Apple had made their move.

In this market the defenders of the old regime have largely carried the day. That's because in contrast to the music market fonts are effectively pieces of machinery – let's remember they're referred to as software – whereas music, for instance, is data. Nobody tries to insert music into the workings of another program and have it do something, whereas those little letter shapes are made to line up in rows, be read by people, and generally work for their living. For serious applications, like body text, people need ones that work and are not missing bits. Crap fonts are not expressive creative works. They're just crap.

I'm labouring the point here, but in essence the majority of 'free' fonts do not compete against commercially-produced fonts, and that is simply because of the amount of labour and expertise that goes into the production of a font. It's *worth* paying for a font, and that's why even the permissively-licensed operating systems have all acquired system fonts that were produced by people who got paid, not by people donating their time.

So to reach a point where there is a chance for licences to be a possibility, the simple and correct idea that an existing font format would be the best suited has been ditched. A web fonts working group has been set up by the W3C to agree on the recommended web-only font formats.²⁸

28 <http://www.w3.org/2009/08/WebFonts/charter.html>

WOFF, a leading contender for the web-only font format

The Web Open Font Format, WOFF, was proposed initially by Erik van Blokland and Tel Leming,²⁹ who are denizens of the type and tech world. Their reputation probably gave the proposal more traction amongst foundries than a roomful of software engineers could ever have done.

The proposal was then worked on by Jonathan Kew and John Daggett at Mozilla. Their stature as browser software engineers has made the techie people happy. All in all, it's been a bundle of fun for everyone.

But what about our friends at Microsoft? The news from them is very positive. Officially, they're prepared to commit to WOFF as a standard format when they believe that the foundries support it too. This guarded approach can I think be taken as a sign that they will indeed be supporting a common format for web fonts before hell freezes over. And that feels like progress.³⁰

The Web Fonts Working Group is widely expected to support WOFF.

What will WOFF do that other formats won't?

Because WOFF obfuscates itself as a non-desktop-font format, it is by far less threatening to font software vendors. That is good news because there are already some more adventurous foundries who have realised, albeit after a decade, that this web font thing might be a good idea. To the extend of actually coming out in support of WOFF. Notably FontShop, who publish some very tasty fonts including Meta by the well-known designer Erik Spiekermann. In October 2009 they said [SLIDE 36]:

'Erik Spiekermann ... and the FontShops endorse the woff specification, with default same-origin loading restrictions, as a Web font format. FontFont expects to license fonts for Web use in this format'.³¹

WOFF files are compressed, but not encrypted. Compression is a good way to save bandwidth; encryption is a good way to make clever people want to waste time breaking codes.

And the new format has 'green-washed' itself as a friendly format by adopting an XML outer shell that has clearly-defined spaces where a little life-story of the font can be written. Third-party tools like web browsers could expose this information to curious punters: it might include the designer's name, the licence, and a URL from which the font could be bought. This is, I think, a good way to crumble open the closed world of the font file so I give it a big thumbs-up.

I must mention EOT Lite, (or CWT - 'Compatible web type') because that may well also be in the running. It's only on the table because old versions of Internet Explorer have a habit of hanging around like a bad smell. EOT Lite could be a W3C-compatible fudge that would at least partially bring these old browsers some benefit. But nobody likes it except Ascender.³²

However, this is the future not the present. We move on.

29 <http://lists.w3.org/Archives/Public/www-font/2009JulSep/0361.html>

30 Email from Simon Daniels at MS Typography to BW, January 2010

31 <http://www.edenspiekermann.com/woff/> (you'll need Firefox 3.6+ to see the full beauty of their web-fonted press release)

32 <http://www.ascendercorp.com/info/web-fonts/>

So where can you get good-quality web fonts *now*?

I claimed earlier that the only good fonts were ones you paid for. And then I said you couldn't pay for the ones you wanted to use.

Well, I was lying again. Here are some commercial services which are just starting to break through:

- Typotheque³³ is a font software vendor which has chosen to offer hosted web fonts as a bundle along with conventional print licences for its entire library.
- Typekit, which launched last Autumn, has chipped away at foundry resistance to OpenType by offering a font-hosting service that I presume gives the participating foundries the opportunity to dip their toes in the water without the need to make their own investment in notionally pirate-proof font-serving infrastructure. So far, Typekit has the support of commercial foundries like FontShop³⁴ and Typofonderie Porchez³⁵.

You won't find Ascender, who are Microsoft's font publishing agents, amongst the clients of Typekit.³⁶ Likewise you won't find Linotype talking about its web fonts policy in public.

And actually there are loads of very good fonts out there which you may not directly have paid for. You might have unwittingly subsidised them somehow in a very broad sense but essentially they will cost you nothing. Some of these fonts are freeware and some of them are free software.

To find such fonts and make them famous I suggest you pay a visit to the Font Squirrel. I've mentioned this creature already tonight but not really explained what he's about. Essentially Font Squirrel is a place to get hand-picked fonts that cost nothing at the point of use. I suspect that I care more than Font Squirrel does about the reasons each of these fonts is zero-cost, but more importantly Font Squirrel cares a lot more than I do about the quality of each of the fonts he has on display. You can save yourself time and trouble by going to fontquirrel.com and having a look around.

I must also blow the Open Font Library's trumpet. OFLB is a place where people who are working on permissively-licensed fonts can publish them. We demand that the fonts in the library are all appropriately licensed and we recommend SIL's Open Font License³⁷. All such fonts are explicitly available for use as web fonts, of course.

Incidentally I've been involved with a re-launch of this site for well over a year now and it hasn't happened yet. But I'm patient and I am confident that it will soon be live. When it is, you will be able to use the OFLB as a font-hosting service, as long as we remember to add the Common-origin Resource Sharing header of course.

And there are lots of other lists of good permissively-licensed fonts if you're not afraid to google and to test.

33 <http://www.typotheque.com/webfonts>

34 <http://blog.typekit.com/2009/11/18/fontfont-library-now-on-typekit/>

35 <http://blog.typekit.com/2009/12/21/porchez-typofonderie-and-delve-fonts-join-typekit/>

36 But some of their best recent work is available, at least in part. They designed the Droid font family for Google's Android platform. These are licensed under Apache, and you can unravel them from the Android SDK and host them yourself if you so desire.

37 <http://scripts.sil.org/OFL>

How do I get started?

I think I have succeeded so far in turning the small issue of 'web fonts' into a big, scary pileup. You won't want to even think about mucking about with web fonts now, because it's such a mess.

Overall, my advice is to take it easy, and expect things to degrade but not always gracefully. You can try an online generator such as Font Squirrel's to see how the syntax works and ease yourself in.³⁸

Headings

Firstly, fiddling with heading styles is low-risk, so perhaps start there. In particular you might be able to work with a single font family member, which means you won't have to accommodate Internet Explorer's failure to create font-families with more than one member.

Body copy

Secondly, displaying the body copy of a web page with web fonts is still considered very cutting edge. There are a couple of obvious reasons why.

For one thing, there's the risk of a big time lag in the download of the fonts. This leads to the famed flash of unstyled text I mentioned earlier. To recap, some thoughts on mitigating this are in the following articles:

- Mark Boulton's 2009 24ways article on web fonts and font stacks³⁹
- Paul Irish's blog post on FOUT, the flash of unstyled text, and how to prevent it⁴⁰

Also, the latin core fonts published by Microsoft have a large body and a big x-height. In other words they take up a greater amount of the space available to each letter than fonts designed for print. The idea is to make the parts of the letter that make them distinctive nice and big so that even the most determined rasterizer can't produce illegible bitmaps. That technical need has now gone, but you will soon discover that when you degrade or fall back to core fonts they will tend to look bigger – and take up more lines – than the fonts you are pulling in through @font-face. **SLIDE 37** The difference in size between the web font and the chosen core fonts has to be appreciated pretty early in the design process or it will bite you on the arse. The worst candidates are Georgia and Verdana.

Incidentally this size discrepancy is one of the factors that might encourage web designers to sketch with CSS rather than Photoshop. I once saw a feathered pig so it might happen.

Finally, there is the risk of creating client expectations that cannot be met. If you show your client a site sketch that uses a font of your own choosing, build a site that uses web fonts, and then expect that his or her antique copy of Firefox 3.0 will display them then you're setting yourself up for a fall. Equally it may be best to forget about supporting IE.

38 <http://www.fontsquirrel.com/blog/2009/11/what-to-expect-from-our-font-face-generator>

39 <http://24ways.org/2009/designing-for-the-switch>

40 <http://paulirish.com/2009/fighting-the-font-face-fout/>

Some use cases

Here are some suggested scenarios in which a web font might sneak in.

Personal sites: blogs, portfolios etc where you are the client and you can dip your toe in the water.

Type nutters and bleeding-edge enthusiasts, who forgo all compromise. Sometimes, that's just the right attitude.

Corporate sites. This is not really to be recommended, but someone's got to be the first to persuade their clients to use the custom brand font that cost so much to commission and that is compulsory on all printed material from billboards to business cards. Haven't seen it done; look forward to it.

You can get a really good 'light touch' intro from Jonathan Snook's article in last Christmas' 24ways.⁴¹ And Nice Web Type has a good article that works through all the twists and turns of implementation. I recommend you read it.⁴²

41 <http://24ways.org/2009/spruce-it-up>

42 <http://nicewebtype.com/notes/2009/10/30/how-to-use-css-font-face/>

Erm, how do WOFF, EOT and OpenType fit together?

I thought I would go over this ground again as I find it confusing. You probably will too when I have finished, but let's look at the question of multiple font formats together.

Microsoft has been supporting `@font-face` for a decade using Embedded OpenType, a format that is proprietary and which it cannot share. This suited it well back in the day, but now it's out on a limb.

Other browsers cannot use EOT, but they can take the much simpler route of supporting OpenType and TrueType.

Meanwhile a deliberate feature of `@font-face`'s `src` property is that it can accept multiple values. I showed a rule earlier which had both a `url` and a `local` value. There can be multiple `url` values and conforming user agents – browsers which have been written to support the spec – are supposed to select the first one that they think they can use. So according to the spec bunging in an EOT file and an OpenType is absolutely fine; you can add a WOFF too.

Font Squirrel lets you generate EOT and WOFF files and the relevant CSS cascade to include both. Here's a sample triple-source `@font-face` rule from Font Squirrel. Note that Font Squirrel has made a distinct font-family for each member of the 'real' family, thus bringing IE into the fold at the expense of misinterpreting the W3C spec.⁴³ **[SLIDE 38]**

```
@font-face {
    font-family: 'CantarellRegular';
    src: url('Cantarell-Regular.eot');
    src: local('Cantarell Regular'), local('Cantarell-Regular'), url('Cantarell-Regular.woff')
    format('woff'), url('Cantarell-Regular.ttf') format('truetype');
}
@font-face {
    font-family: 'CantarellBold';
    src: url('Cantarell-Bold.eot');
    src: local('Cantarell Bold'), local('Cantarell-Bold'), url('Cantarell-Bold.woff')
    format('woff'), url('Cantarell-Bold.ttf') format('truetype');
}
```

Remember that unless your license allows it you are not allowed to re-encode a font in a new format. That's for the licensor I'm afraid.

43 Internet Explorer has real problems parsing the `src` values and can generate extraneous requests to non-existent URIs. This is summarised at <http://readableweb.com/mo-bulletproofer-font-face-css-syntax/>

So what can we look forward to?

One thing widely predicted is a rash of poorly-chosen fonts that look dreadful on screen and have missing characters. I think not.

I hope that before too long the various combinations of browser and operating system can be made to agree about how they interpret the @font-face rule. If there are ambiguities in the specification itself around that problem then they need to be identified; I have not seen any so far. And Internet Explorer 9 could be the first from Microsoft that complies fully.

(I went looking for the whispered rumour I remembered but could not find it.)

Then we need WOFF to be approved.

I am resigned to the idea that we use a lightly-obfuscated font format to encourage commercial font software vendors and to allow Microsoft to ditch EOT once and for all. In return font vendors – whether offering fonts commercially or as free software – can add lots of information within the font file itself, which does address a real weakness in the conventional font formats like OpenType.

Then we have the maturing of the font hosting services: Typekit, Typotheque, Kernest⁴⁴, and of course the Open Font Library. Right now the best thing about these services is that they offer ways to get your hands on @font-face without having to understand the infrastructure or even @font-face. In time they may prove to have other virtues, such as having the financial muscle to commission new designs, proprietary or not, from able designers. My hope is for new vigour in the type design world, especially away from boring old latin type.

Another brilliant thing I anticipate is the death of image-substitution in all its forms.

My final thought on the topic is this. A 1998 FAQ about how to display Telugu script on web pages asked: 'Is font embedding a panacea? Does this make scripts like telugu first class citizens on the net?'⁴⁵ Telugu is a language of the Indian subcontinent, but despite being the third most widely spoken in India, at 74 million speakers, its script⁴⁶ isn't included in operating systems even now. With @font-face and unicode-compliant font formats such as WOFF or OpenType, it is possible to overcome this limitation quite easily. Let's get on with it. [SLIDE 39]

44 <http://kernest.com/>

45 <http://bhaavana.tripod.com/embedded/faq.html#60>

46 http://en.wikipedia.org/wiki/Telugu_language#Writing_system