The Font Wars

James Shimada
December 6, 2006

In 1984, two years after John Warnock and Chuck Geschke left Xerox PARC to form Adobe Corporation, the company introduced the PostScript page description language. By the late 1980s desktop publishing was in full swing and PostScript was a de facto standard in digital typography. While the PostScript language specification was open and anyone could license an interpreter to build a PostScript printer, Adobe held the secrets to the superior PostScript Type 1 font format, making Adobe the single source of licensing high-end fonts. While some companies were hard at work cracking the Type 1 format, Apple and Microsoft formed an alliance to create an alternative to PostScript technology. The resulting product was TrueType, which was incorporated into both Mac OS System 7 and Windows 3.1. The two competing font standards were responsible for intense rivalies, display incompatibilities, a proliferation of amateur fonts, and battles over the loyalty of the design community. These events collectively became known as the "Font Wars."

Adobe PostScript

The Adobe PostScript technology was at the core of the Font Wars. PostScript is a page description language that uses an interpreter to render the contents of a printed page on a printing device or computer screen. It is also a full-fledged programming language that describes a printed document, including fonts and formatting. All of the images on a printed page, including text, are specified by vector graphics – mathematical formulas consisting of straight lines and spline curves. These formulas are then converted to the dots needed for digital output in a process called rasterizing. A PostScript interpreter is able to rasterize on the fly, which allows for a great deal of flexibility in terms of scaling, rotating and other transformations suitable for the resolution of a particular output device. PostScript is also device-independent, so it can be expected to run on a variety of printers and output devices without modification. When Postscript was first released, in an effort to encourage support of the language, Adobe published the language specification and licensed the PostScript interpreter to printer manufacturers that wanted to build PostScript output devices.

The first Adobe Postscript manual was shipped to a potential customer in March 1984, but the seeds of a graphical description language had taken root in John Warnock's mind several years earlier. In the early 1970s, Warnock worked for the Evans and Sutherland Computer Corporation, which sold powerful custom graphics devices for real-time simulation, such as windshield graphics for maritime and flight training. Warnock led their Mountain View, California research office and was charged with developing a database system for their simulation machines. Because, for a real-time application, it was unfeasible to store a graphical representation of an environment as a series of bitmapped images, Evans and Sutherland employed a graphics model based on line segments and coordinate system transformations. By 1975, Warnock and company had completed the first version of the "Evans and Sutherland Design System," a programming language used to control the graphics model.

In 1978, Warnock joined Xerox PARC. At this time, Xerox was experimenting with a series of raster printers: the XGP, EARS and Dover. After the headaches of converting software from one printer to the next, it became clear to the researchers at PARC that application programs generating files destined for the printer should not be tied to the properties of a specific printer device. The company began working on a device-independent page description scheme called the "Press" format. Warnock later merged Press with his work on graphics modeling languages into a full-featured page description language he called "Interpress." With Interpress, a printed page was described, not as a giant bitmap image, but rather in terms of line segments, curves, and transformations. Warnock and Chuck Geschke tried, unsuccessfully, for two years to convince Xerox to make Interpress a commercial product. Xerox's lack of support for Interpress encouraged Geschke and Warnock to start their own company, which they named Adobe, after a creek that ran past Warnock's garden in Los Altos, California. (Cringley, 215)

PostScript was practically identical to Interpress, by comparison with other page description schemes available at the time, but it actually removed some features that Warnock had taken exception to during the development of Interpress. At PARC, a lot of work had gone into deliberately restricting Interpress in order to maintain page independence. Page independence is an important consideration for performance, because it allows a printer to process a page before processing all the preceding pages, making it possible to print page 457 in a 500-page document without having to process pages 1 through 456. Much to the chagrin of Butler Lampson, the PARC researcher who contributed this feature of Interpress, Warnock removed it from PostScript. Warnock favored more flexibility in the PostScript language, making page independence impossible. Because the PostScript language was dynamic, or "wild and free" as Lampson described it, a given page could not be printed until all the PostScript code on the previous pages had been interpreted. Interestingly, Adobe's PDF format – PostScript's eventual successor – would later tout page independence as an improvement over PostScript. (Lampson)

The handling of fonts is one of the major challenges with a language designed to describe the appearance of printed material. Like other graphical objects in PostScript, fonts faces were defined in terms of lines and spline curves. Fonts represented in this manner are known as "outline fonts," and they offer significant advantages over bitmapped or "raster" fonts, which are essentially represented as a grid of pixels that are either on or off. One of the biggest advantages of outline fonts is that only one outline per character is needed to produce all the sizes of that character that one would ever need, rather than requiring a different and unique font for each size of the same typeface. Scaling a font allows it to be printed in a variety of sizes on a variety of output devices. But Fonts cannot simply be scaled linearly to a desired size because of the inherent approximations involved in digital output. These problems are especially apparent at low resolutions and small font sizes – with fewer dots available to draw letterforms, features such as stem weights, crossbar widths, and serif detail can become inconsistent or missed entirely. (Microsoft)

Because fonts required extra attention, a special language was used to describe a PostScript outline font, which was both a subset of and an extension to the PostScript language specification. Certain extensions addressed the scaling problem with "hints." The hints were additional instructions stored with the font outline that described certain guidelines the PostScript interpreter needed to maintain at different sizes, which could include deliberately distorting the font outline for the sake of appearance and legibility. With proper hinting, a font looked correct and maintained its personality at all different sizes, and was able to be output with high quality even at very small sizes or low resolution.

When PostScript was launched in 1984, the published language specification included the details necessary for third parties to create fonts that could work with the PostScript interpreter. This open font language specification was called the "Type 3" font format. While they made use of much of the PostScript language, Type 3 fonts contained no hinting mechanisms whatsoever. Adobe closely guarded its technology for font hinting by compressing and encrypting the fonts containing special hinting commands into what they called "Type 1" font format.

Anyone who wished to use the high-quality Type 1 fonts had to license them from Adobe. Anyone (outside of a select group of font foundries who struck licensing deals with Adobe) wishing to design their own PostScript fonts had to settle for the inferior Type 3 format.

The Rise of Desktop Publishing

Up until the 1960s, most publishers used 19th-century technology to perform typesetting. The Monotype was a mechanical typesetting system that used a key-punch machine to produce a roll of punched paper tape. This roll was then fed into a casting machine and, much like a scroll controls a player piano, the Monotype paper tape guided the machine to produce individual pieces of type from hot molten lead. In the 1960s, new machines based on photography began to replace the slow and expensive Monotype. These photo-typesetting machines also used a mechanical keyboard to enter text and punch paper, however, in this case the paper tape controlled a machine that exposed type onto light-sensitive paper using a strobe behind a rotating film strip containing typefaces. The exposed photo paper was then developed in a chemical processing machine to produce galleys.

With professional experience in both newspaper journalism and technology, Paul Brainerd, the founder of Aldus, was in a unique position to revolutionize the publishing process. He had already gained hands-on experience in phototypesetting during his graduate work in journalism at the University of Minnesota. In 1975, Brainerd worked for the *Minneapolis Star Tribune* and later for Atex, a company that sought to automate the process of newspaper production with computer technology. At Atex, Brainerd became involved with using software for prepress, and wrote specifications for a word processing program. Kodak eventually acquired Atex and closed the Redmond, Washington research and manufacturing plant at which Brainerd had been working. Unwilling to take a Kodak job that would have relocated him to Boston, Brainerd separated from Atex and started his own company. He named it Aldus after the 16th-century Venetian scholar and printer, Aldus Manutius, and he coined the term "desktop publishing" to describe his enterprise. (Wilma) In 1985, the same year that Apple launched the LaserWriter printer, Aldus developed PageMaker software.

Steve Jobs also played a major role in the rise of Desktop publishing. He was intent on bringing the PostScript printing technology to the personal computer, even though due to the complexity involved in interpreting the PostScript language, it may have been a more suitable printing technology for mainframes! (Cringley 216) Jobs knew the LaserWriter could set Apple apart from the competition by showing that one could produce higher quality printed material with a Macintosh than with any other personal computer available. The Hewlett Packard LaserJet had already been introduced in 1984, and it had set a new quality standard for PC printers far superior to that of the dot matrix printers on the market at that time. Documents produced by the HP LaserJet looked like they came from a typewriter. But Jobs believed he could do even better, and that if he could incorporate PostScript technology into the LaserWriter printer, Macintosh users would be empowered to create documents that approached the quality of a professional typesetter. (Carlton 111) In a grand display of confidence, Jobs invested $2.5 million dollars in Adobe stock, acquiring 15% of the company. (Cringley 220) Apple manufactured its LaserWriter with a built-in PostScript interpreter licensed from Adobe. Ironically, both the LaserWriter and the LaserJet were built on the Canon LBP-CX print engine, meaning that many of the components for feeding paper and fusing the image were actually interchangeable. It was Postscript that distinguished the LaserWriter from the LaserJet, which still had an archaic printer control language. Fitting PostScript into the LaserWriter provided great benefit, but also came at a significant cost. In order to support the PostScript interpreter, the LaserWriter needed extra processing power in the form of a 12Mhz Motorolla 68000 CPU, 512KB of RAM, and a 1MB frame buffer. At the time of its release, the LaserWriter was the most powerful computer in Apple's entire lineup, and garnered a hefty price tag of $7000.

Now the only thing Apple and Adobe needed was a great application, something compelling enough to convince a consumer to buy the whole system. A good fiend of Brainerd, computer-typesetting consultant John Seybold, set up a meeting between Apple and Aldus so they could discuss the opportunity of combining the LaserWriter printer with PageMaker. The system rolled out in 1985 and the winning combination of PostScript, Macintosh, LaserWriter and PageMaker sparked the desktop publishing revolution. Decent quality print material was easier than ever to produce, and at a reasonable price to the consumer. And although early versions of PageMaker lacked some of the features needed to emulate the craftsmanship of a true professional typesetter, the convenience, power, and speed of desktop publishing tools were perceived as revolutionary. Paul Andrews, a technology columnist for The Seattle Times, wrote in 1994 about the emergence of desktop publishing, "To anyone who remembered the tedious act of justifying columns in high school newspapers, of physically cutting and pasting headlines or awkward pieces of type onto glue-smudged page dummies, and of missing publication deadlines because the printer had four jobs ahead of theirs, desktop publishing was an occupational epiphany. It gave control to the creator of the document – control over the appearance, content, production and timing of the entire publishing process." (Andrews) It was a great time for everyone involved. Aldus and Adobe went from mere startups to wealthy enterprises, and Apple saved the Macintosh from lackluster sales and slipping market share. (Malone 387) The remarkable cooperation between these three companies created a win-win-win situation. The blissful alliance, however, would prove to be short-lived. Tensions began to mount around PostScript licensing fees, and then Microsoft entered the picture.

The Eruption of the Font Wars

Before long, PostScript was supported, not just on the LaserWriter, but also on high-end imagesetting devices, making it the de facto standard document format of desktop publishing, and giving Adobe a monopoly on the page description language implemented by output devices. Adobe built an entire library of fonts with outstanding type designs that were either licensed from original foundries or created by well-known designers on staff such as Carol Twombly and Robert Slimback. The demand for Type 1 fonts was high, and soon Adobe was making $100 million a year in printer software and fonts licensing.

Meanwhile, personal computers had become more powerful, and the idea of moving the PostScript interpreter from the printer to the host machine became an exciting possibility. If the host machine could do the PostScript rasterization, and send a large bitmap to the printer, then the laser printers wouldn't need to be powerful computers themselves. This would also simplify desktop publishing software because it would allow the software to use one set of graphics routines to draw to the printer or the screen. As long as the Postscript interpreter remained in the printer, the user had to maintain one set of bitmap fonts for the screen and a separate set of Type 1 outline fonts for the printer. Moving the PostScript interpreter to the host machine would simplify things for the user, who would only need to maintain only one set of fonts. As less expensive laser printers were introduced, licensing the PostScript interpreter became a sticking point. The fact that PostScript was not resident on the host computer became a cost issue. (Tribble)

Adobe maintained an uncompromisingly technology-driven focus on its products, even at the expense of its relationships with key partners. At one point in 1985, Apple approached Adobe with some feature requests for the PostScript interpreter in order to improve the performance in printing Macintosh bitmap fonts. "They wanted to dump screens faster, and they wanted Apple-specific features added to the printer," Warnock explained to Robert Cringley. "Apple came to me and said, 'We want you to extend PostScript in a way that is proprietary to Apple.' I had to say no. What they asked would have destroyed the value of the PostScript standard in the long term." (Cringley 223) At that time, PostScript license fees from Apple accounted for a large portion of Adobe's revenue, and in this respect Warnock may have bitten the hand that fed him. In 1988, Microsoft tried to strike a deal with Adobe, asking for a PostScript interpreter for the screen that could be incorporated into the next version of Windows. The idea was that Windows users would be able to see the exact same fonts on-screen that they could print on a PostScript printer, a true WYSIWYG interface. The supposed benefit for Adobe would be a spike in their business selling fonts, since native PostScript support on Windows would dramatically increase their installed base. Warnock again said no, losing the opportunity for a strategic partnership with Microsoft.

In response to the demand for users to be able to see PostScript Type 1 fonts on screen, Adobe came up with a product called Display PostScript. As the name suggests, it was a system that interpreted PostScript to generate on-screen graphics. Display PostScript was actually a joint venture between Adobe and Steve Jobs' NeXT Computer in which the development was done by NeXT, but the product officially belonged to Adobe. NeXT used Display PostScript as a foundation for the NeXTStep windowing system, and even introduced a laser printer with no CPU since PostScript rasterization was handled entirely by the operating system. Meanwhile, Adobe could license Display PostScript to third parties at their discretion, and they did indeed attempt to license it to Microsoft and Apple. (Cringley 227) For Adobe, Display PostScript was a double-edged product. While moving the PostScript interpreter to the host computer had many advantages, it also held the potential of diminishing Adobe's business of selling

PostScript interpreters to printer manufacturers. Adobe proposed a steep licensing fee for Display PostScript, but this time it was Apple and Microsoft that declined: they had other ideas in mind.

Apple and Microsoft briefly put aside their already existing corporate rivalries, and formed a partnership to develop an alternative to PostScript. Apple had been developing an outline font technology, code-named Bass because the fonts were scale-able, (Kaasila) which would later be marketed as the product called TrueType. While PostScript was originally created to be printer language, TrueType was designed with the intention of using the host computer as the rasterizer, and was particularly well suited for displaying fonts on a computer screen. With the TrueType rasterizer built into a host computer's operating system, outline fonts could be rendered on-screen without any third-party software, and could be printed on either a PostScript or non-PostScript printer. The TrueType font format was compatible with, but not dependent upon the PostScript interpreter – fonts could either be rasterized before they were sent to the PostScript interpreter, or the TrueType rasterizer itself could be sent to the PostScript interpreter as a PostScript program. With TrueType, users would only have to maintain one set of outline fonts for both screen and printer, and desktop publishing applications would be able to truly embrace the WYSIWYG concept. As part of the partnership, Apple would allow Microsoft to use TrueType, and Microsoft would give Apple a PostScript-compatible interpreter that it had acquired from a developer named Cal Bauer. The coup d'etat to Adobe occurred at the Seybold Desktop Publishing Conference in San Francisco on September 20, 1989, when Apple and Microsoft announced their strategic alliance. Bill Gates declared that TrueType would be a new standard. John Warnock was distraught, and when it came time for him to speak at the conference, he publicly accused Microsoft and Apple of selling "snake oil." (Carlton 114) A few days later, Apple sold all of its Adobe shares for $89 million. While it was a profitable move for Apple, it was also a deliberate and distinctive separation from Adobe. The Font Wars had begun.

Even before the Seybold convention, however, Adobe was already starting to feel pressure from other sources and its stronghold on printing was beginning to be threatened. Bitstream, one of the oldest pure digital type foundries, founded in Cambridge Massachusetts in 1981, was at that time on the verge of cracking the encrypted Type 1 font format. Several developers were already selling PostScript clones to low-end printer manufacturers. In 1990, after years of guarded secrecy around the Type 1 font format, Adobe finally published the Type 1 specifications, allowing anyone to build and sell the superior Type 1 fonts. Adobe switched directions from trying to license Display Postscript, and announced a product called Adobe Type Manager (ATM). As another system that could interpret PostScript code to generate on-screen displays, ATM only worked for type, not arbitrary vector graphics like Display PostScript. The key was that ATM was to be sold directly to consumers so, without any assistance from Apple, Type 1 fonts would be available on Apple computer screens.

Adobe announced ATM before the product existed. At the time of the announcement, the Mac OS System 7 was not expected to be released for another year and a half. The ATM announcement could be interpreted as an offensive move to try and dissuade Apple from going the separate route with different display technology. If ATM could be expected to enjoy a good deal of market acceptance, then Apple wouldn't need to go through the trouble of developing an alternative font technology. (Cringley 227) On the other hand, the announcement of ATM could be interpreted as a defensive move: if Apple was going part ways with Adobe and pursue its alternative font technology, then at least by selling directly to consumers there would still be some chance that Type 1 fonts could make their way to Mac computer screens. Either way, Adobe was on the hook for delivering ATM. The TrueType font format became an integral part of Mac OS System 7 in 1991, and in Windows 3.1 in 1992.

Extensions of Type 1 and TrueType

Throughout the 1990s, Microsoft gradually improved its support for TrueType in Windows. Many of the enhancements were geared toward improving the stability and performance of the buggy TrueType rasterizer introduced in Windows 3.1, but a notable TrueType enhancement came with Windows 95: font smoothing. Font smoothing is an anti-aliasing technique that involves black, white, and three intermediate shades of gray to smooth the edges of bitmap shapes, improving the appearance on-screen while minimizing the blurriness that's often found in anti-aliased text.

Meanwhile, Apple and Adobe were much more ambitious in their plans to take type display technologies to the next level. When Steve Jobs came back to Apple from NeXT he brought his respect for John Warnock and his support for PostScript back with him. As Apple acquired NeXT, they also got the Display PostScript technology that NeXT had developed, and supported PostScript fonts natively in the Mac OS. Driven by both "intense collaboration and intense rivalry," Apple and Adobe continued to develop increasingly sophisticated type display technologies. (Tribble)

Apple's TrueType GX fonts, later known as Apple Advanced Typography, had advanced typographical features based on QuickDraw GX, an extensive graphics library that became a core part of the Mac OS system. There were many typographical enhancements in TrueType GX, including automatic ligatures, optical alignment, and "variations." TrueType GX featured a strong distinction between semantic and stylistic qualities of glyphs and, as a result, automatic common ligature substitutions could be made that didn't affect the underlying character representations in the text. A ligature is two or more letters that are joined into one character, such as *fl*. A common technique in typesetting software had been to replace the two characters that needed to be joined with a single, special ligature character. This technique had created problems because it caused a change in semantics that prevented spell checking and text searching tools from recognizing the ligature as two distinct characters. With TrueType GX, extended information that facilitated automatic ligatures could be stored in the font itself, without affecting the underlying semantics. Another advancement in TrueType GX was the automatic optical alignment of text edges. Different letterforms, even when precisely lined up on an edge, sometimes presented the optical illusion that they were misaligned, due to the individual shapes. Automatic alignment took this into account and provided edges that appeared flush to the human eye. Outline font "variations" were another key advancement in TrueType GX. Variations were a form of parameterized font outlines, an idea proposed in 1982 by renowned computer scientist and digital typography pioneer, Donald Knuth. (Knuth "Meta-Font") Variations allowed for a single font specification to adapt to parameters along a design axis, such as "weight" or "stem width." Using variations along the weight axis, for instance, a font developer could parameterize a font outline to create a continuous range of weight instances going from extra light, to light, to bold, to extra bold.

Adobe Multiple Masters was another technology that extended the Type 1 format to parameterize a font outline. Using Multiple Masters, two font outlines were supplied for the opposite ends of a design axis. The font would be interpolated across the axis and the designer could modify any of the in-between instances as needed. A handful of axes were available to parameterize the outlines, but Multiple Masters fonts would generally support one or two, such as weight, width, or optical size. The reason Multiple Masters fonts typically only made use of a small number of axes was that each axis required two font outlines for each extreme, and therefore the number of font outlines the developer needed to produce increased quadratically with the number of axes. Multiple Masters fonts never became very popular and by 1997, there were only about 36 such fonts on the market, half of which were produced by Adobe. It often made more sense for a font developer to release a font set as multiple Type 1 packages. An example of this is the HTF Didot font by Hoefler & Frere-Jones, which is available in six designs: light, medium, bold, light italic, medium italic and bold italic. Apple's extension to TrueType was not met with much commercial success either.

Years earlier, when Donald Knuth developed the METAFONT language he already understood some challenges with the acceptance of parametric outline fonts like Type 1 Multiple Masters or TrueType GX. Knuth's METAFONT took parametric outlines to the extreme. His Computer Modern typeface was built around more than 60 design axes. During the years he was devoted to digital typography he came to understand a key difference in the thought process type designers and programmers seemed to employ when confronted with a problem. Type designers traditionally worked with a fixed set of specifications to solve a specific design problem. Computer programmers, on the other hand, were unique in their ability to understand parameters, where the solution to a problem could adapt to different sets of input. (Knuth "Lessons") With outline fonts, a typeface is actually a functioning computer program, rather than the artifact of a particular design solution. Parametric outline fonts may be a natural extension to the outline font as a program, but Knuth's lessons learned from METAFONT highlight a gap between program-oriented thinking and design-oriented thinking. This gap undoubtedly contributed to the weak acceptance of Type 1 Multiple Masters and TrueType GX at a time when the convergence between these two cultures was relatively new.

Loyalties of the Design Community

Users went from having to purchase Type 1 fonts directly from Adobe to having the ability to purchase fonts in several different formats and from various sources. Major font foundries and professional designers, however, remained loyal to Type 1 fonts. The perception in the design community was that TrueType was an inferior technology.

Ironically, TrueType offered advantages in its hinting features that would have seemed to attract high-end font designers wanting fine-grained control over the appearance of their fonts. TrueType allowed complete control over rasterization, right down to pixel-by-pixel control on the bitmap grid, supporting instructions to move any part of the glyph's outline as much as needed. This permitted the glyph to completely change shape at different sizes, for the ultimate in design flexibility. TrueType put the hinting intelligence in the fonts themselves rather than in the rasterizer. This gave the font developer very fine-grained control over the final appearance of the font, but it also meant that the bulk of the hinting calculations took place during font development rather than at runtime. To take full advantage of TrueType's hinting power, a substantial amount of programming was required. The expertise required to properly program the hints made it difficult to produce high-quality fonts and the TrueType font market became flooded with amateurish fonts.

In fact, many of the first available TrueType fonts were actually PostScript Type 1 fonts converted by automatic means to the TrueType format. Since the hinting information did not translate directly across the two formats, these fonts had hinting instructions that were inferior to their Type 1 copies. To make matters worse, fonts converted automatically from Type 1 to TrueType suffered because of different formulas used to define curves in the glyph outline – Type 1 used cubic bézier curves while TrueType used quadratic bézier curves. Although the quadratic curve is technically a subset of the cubic curve, the conversions were still imperfect due to small rounding errors. As a result, the first TrueType fonts to hit the market were low quality imitations of existing PostScript fonts; hence, the perception that TrueType was an inferior technology. (Phinney) With this negative perception of TrueType fonts within the design community, the high-quality tools that professional designers needed to actually harness the power of TrueType's hinting were very slow to reach the market. It wasn't until 1997 with the release of FontLab 3.0, that there were retail versions of font editing software with native support for TrueType's hinting and quadratic bézier curves.

TrueType and Type 1 highlight an issue that comes up repeatedly in technology: the most powerful tools often require the most technical expertise to master. TrueType was analogous to a low-level machine language, in contrast to Type 1, which was like a declarative interpreted language. Type 1 hints are fairly easy to understand – the font designer declares hints for certain qualities of a glyph (such as vertical and horizontal guides, overshoots, stem snaps, equal counters and shallow curves), and these specify general constraints to the PostScript interpreter. The interpreter then uses special techniques for filling in the outline at rasterization time. Type 1 hints do not specify exactly what pixels will be turned on at the bitmap grid level, rather they tell the interpreter what features ought to be controlled, and the interpreter uses its own intelligence to decide how to do it.

Microsoft Windows 3.1 had some flaws in its initial TrueType rasterizer that not only contributed to the perception of TrueType as an inferior technology, but also helped steer designers away from using Windows as a platform for typography and graphic arts. In addition, Microsoft didn't show enough interest in typographical tradition for designers to have confidence in Windows. When Microsoft standardized on TrueType in Windows 3.1, they needed a built-in sans-serif font. Microsoft opted to go with Arial rather than Helvetica, in part because it was cheaper and, in part, because they didn't feel that their users would know the difference. (Simonson) Arial had first appeared as a direct substitute for Helvetica in a PostScript clone in the late 1980s. Since it was drawn to match the proportions and weight of Helvetica, Arial could be automatically substituted on a PostScript output device when Helvetica was specified. At first glance, Arial looked just like Helvetica, but there were in fact many differences between the fonts. Details, such as the fact that Helvetica has a tail in its "a" and Arial does not, that may have seemed trivial or arbitrary to a software company like Microsoft, were important and noticeable to the design community. Helvetica had been the standard, ubiquitous corporate font since the 1970s. When the PostScript interpreter came out in 1984, it included four basic built-in fonts and Helvetica was one of them. Demonstrating respect for the integrity of type design, Adobe had licensed all four fonts from the original foundries. When Microsoft chose Arial to showcase in their software, designers interpreted this to mean that Microsoft didn't care about design details or typographical tradition. To the astute observer of type, it was as if Microsoft needed a corporate spokesman and hired the stunt double of a somewhat boring TV star. Apple, on the other hand, chose Helvetica when they standardized on TrueType in System 7.0, and paid an appropriate licensing fee to the original font foundry, Linotype. The design community remained loyal to Apple. The perception of Windows as a business computer and the Mac as an artist's computer lasts to this day.

OpenType and the End of the Font Wars

By the mid-1990s, Microsoft and Adobe had strong incentives to make peace. Windows was clearly dominating the personal computer market and Type 1 fonts couldn't be supported on-screen without installing ATM software. Without a share in the Windows market, Adobe faced a great obstacle to selling Type 1 fonts to a broad range of users. Microsoft was trying to gain credibility in the design community. (Mendelson) The result of these converging interests was the OpenType initiative, which allowed TrueType and Type 1 formats to become interchangeable.

Announced in 1997, OpenType acted as an abstraction for both TrueType and Type 1 fonts. It was a TrueType-like wrapper for each font format, which meant that OpenType fonts could be supported cross-platform and applications no longer need to distinguish which type of font was actually in the OpenType wrapper. With the release of Windows 2000 Professional, OpenType was fully supported in Windows, and by 2002 Adobe finished converting its entire font library to the OpenType format. With Apple's 10.4 release of OS X, the Mac, which had supported both TrueType and Type 1 for years, embraced OpenType as well. In addition to compatibility, OpenType incorporated many of the same advanced typographical features that had never made it to the mainstream with Apple's TrueType GX and Adobe's Multiple Masters, such as automatic ligatures, true small caps, parametric outlines, and even niceties like a "Qu" where the tail of the "Q" extends below the "u".

The OpenType initiative marked the official end of the Font Wars, but in truth, the differences between font formats that had once been widely disparate, had already started to blur as a result of the industry's drive toward compatibility and standards. A TrueType rasterizer had become a standard component of a PostScript interpreter, and all the major operating systems supported all the major font formats. Adobe licensed, at no charge, the source code used for its own OpenType font development, which allowed third-party font editing applications such as FontLab and FontMaster to add support for OpenType with relative ease.

Compatibility problems will always exist as long as different organizations produce software on which data needs to be shared, but the issues of the present day have become much more specific and advanced. It's easy to take for granted how something as fundamental to computing as displaying type can be accomplished so seamlessly across platforms. Considering the business and technical issues involved in type display and the chaos of the Font Wars, this compatibility is nothing less than a marvel.

Bibliography

Adobe Systems Incorporated. "Introduction to OpenType." adobe.com. November, 2006. <http://www.adobe.com/type/opentype>

Adobe Systems Incorporated. Adobe Type 1 Font Format. New Jersey: Addison-Wesley, 1990

Andrews, Paul. "A New Page: Mr. Desktop Publishing Embarks on His Post Computer Existence." The Seattle Times. July 17, 1994

Carlton, Jim. Apple: The Inside Story of Intrigue, Egomania, and Business Blunders. New York: Random House, 1997

Cringely, Robert X. Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date. New York: Harper Business, 1996

Kaasila, Sampo. Interview with Laurence Penney. truetype-typography.com. October 1996

Knuth, Donald. "The Concept of a Meta-Font." Digital Typography. Stanford: CLSI Publications, 1999 [Originally published in Visible Language 16 (1982) 3-27]

Knuth, Donald. "Lessons Learned from METAFONT." Digital Typography. Stanford: CLSI Publications, 1999 [Originally published in Visible Language 19 (1985) 35-53]

Lampson, Butler. "Xerox PARC, Workstations, and Distributed Computing." CSE P590A. University of Washington, Seattle, October 17, 2006

Malone, Michael. Infinite Loop. New York: Random House, 1999

Mendelson, Edward. "OpenType Ushers in New Era of Typography." creativepro.com. June 20, 2000. November, 2006. <http://www.creativepro.com/story/feature/6503.html>

Microsoft Corporation. "TrueType Hinting." Microsoft Typography. June 1997. November 2006 <http://www.microsoft.com/typography/TrueTypeHintingIntro.mspx>

Phinney, Thomas. "TrueType and Type 1: What's the Difference?" The FontSite.com. October 1, 1997. November, 2006. <http://www.fontsite.com/Pages/Features/T1vsTTa.html>

Simonson, Mark. "The Scourge of Arial." Mark Simonson Studio. February 2001. November, 2006. <http://www.ms-studio.com/articles.html>

Thompson, R. Scott. Quartz 2D Graphics for Mac OS X® Developers. New Jersey: Addison-Wesley, 2006

Tribble, Bud. "Macintosh software, from the original Mac to OS-X, and leveraging open source." CSE P590A. University of Washington, Seattle, November 29, 2006

Wilma, David. "Paul Brainerd" HistoryLink.org Encyclopedia of Washington State History. February 22, 2006. November 2006 <http://www.historylink.org/essays/output.cfm?file_id=7657>

Colophon

This Adobe PDF document was prepared using Microsoft Word on an Apple computer with Mac OS X 10.4. The Monotype Baskerville font used was originally designed by John Baskerville in 1757. At the time, the stark contrast between thick and thin elements of his letterforms was considered extreme and dismissed as amateurish.