

## Typing in Old English since 1967: A Brief History

Peter S. Baker, University of Virginia

### 1. The typewriter

Many readers of the *Old English Newsletter*, especially those who entered the academic profession in the 1970s or earlier, remember vividly their struggles to represent the Old English language with the technology that existed before the advent of the personal computer. In those dark days, scholars produced the final versions of their dissertations, articles and books using a “typewriter,” that is, a keyboard-operated device which produced images of letters by causing a piece of metal type to strike an inked ribbon, pressing it against a sheet of paper.

The typewriter was easy to use (though it lacked a mouse) and portable, and it offered one a satisfyingly material relationship with the paper output of one’s labor. But typewriters—at least those that most students and professors could afford—had one great limitation: a severely limited character set. The typewriter’s repertory of characters was in some ways like a printer’s font of types, but to keep the machine compact and affordable the selection of types had to be small. Most typewriters sold in the United States could produce text only in American English: to type in German, or even quote a price in pounds sterling, was quite beyond their capabilities. The characters used to write Old English (a language spoken by no living person) and modern Icelandic (a language spoken by fewer than 300,000) were of course not available on American or most European keyboards.

Fig. 1 shows a small section of a page from the first issue of “The Year’s Work in Old English Studies,” which appeared in *OEN* 2:1 (1968). (Publication of *OEN* had begun the year before under the editorship of Jess B. Besinger, Jr. and Fred C. Robinson.) Readers who were teaching in the 1970s or early 1980s will immediately

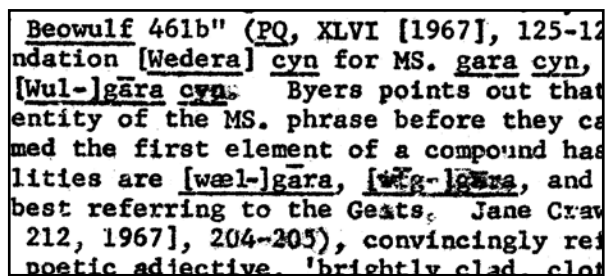


Fig. 1. *OEN* vol. 2 was printed on a mimeograph machine.

The figure illustrates several implications of both typewriting and mimeography. To represent Old English characters one had to improvise. To make æ the typist typed a, backed up the platen by about half a space, and typed e partly overlapping the a; this particular typist has closed up the extra half space this operation produced by skillfully using the backspace key to back up by fractional spaces. To produce þ one typically typed b on top of p, and to produce ð one might type d and draw a stroke across the ascender with a pen or type o and draw both the slanting ascender and the cross stroke. One could make a macron by rolling up the platen and typing an underline or hyphen over a vowel (the method followed here); or one could simply draw in a macron or other diacritic. Some Old English enthusiasts had their typewriters altered, replacing expendable bits of type with þ, ð and æ (see fig. 2). This approach produced



Fig. 2. An Olympia portable typewriter (ca. 1968) altered to type Old English.

legible copy, but not without cosmetic problems. It could be difficult for a typewriter shop to match a particular machine's style of type: in fig. 3 (a sample from the typewriter illustrated in fig. 2), þ is narrower and ð larger than the surrounding letters. It could also be difficult to align the new type precisely: here the bottoms of þ and æ are lighter than the tops because the types were soldered onto the typebars at a slight angle.

To correct typing errors on paper was difficult: one might use either an abrasive typewriter eraser or opaque white correction fluid. Both methods left behind visible traces (in fig. 3 a ghostly g is visible beneath the b of *bilegde*). To make a correction on a mimeograph stencil one used a special fluid that filled the openings that the metal type had cut in the wax coating. After the fluid had dried (the process could be speeded by blowing) one might type over the spot. In fig. 1 the typist has typed the last two letters of *cyn* and much of the following word ([*wīg-*]*gāra*) on correction fluid. The correction of stencils usually produced imperfect results: here one can still see traces of the earlier, erroneous text.

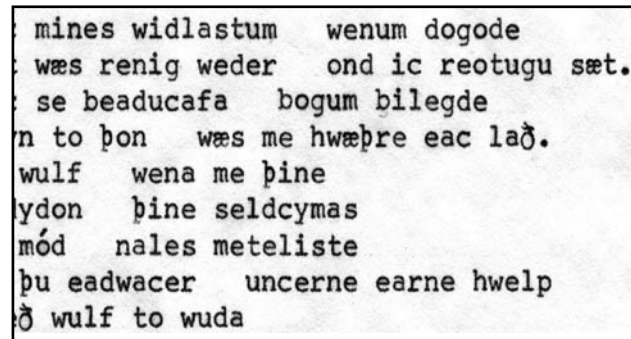


Fig. 3. A sample of type from the altered Olympia typewriter.

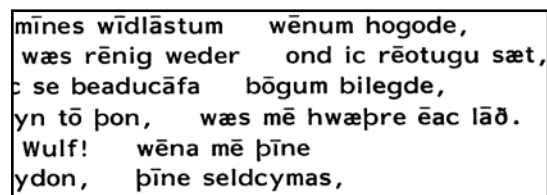
Beginning with vol. 3, *OEN* was edited at Ohio State University by Stanley J. Kahrl. It now was produced by offset printing, a process that could handle much larger press runs and generate higher quality output. Most copy, it appears, was now typed on an IBM Selectric typewriter, which used an interchangeable “type ball” rather than bits of type affixed to metal bars. The Selectric was a revolutionary typing machine. As an electric rather than a manual typewriter, and one with an exceptionally precise mechanism, it produced type that was remarkably even and consistent. While some Selectrics continued to use cloth ribbons, many, including the one on which *OEN* vol. 3 was typed, used a thin plastic ribbon with a carbon coating which produced an image that rivaled the quality of metal type for sharpness and blackness. Fig. 4, from “Year’s Work,” *OEN* 3:1 (1969), shows an example of this type; it also shows that some of the difficulties of typing Old English characters persisted.

The problem of Anglo-Saxon law in particular chapter of H. G. Richardson's and G. O. from Æðelbert to Magna Carta (Edinburgh, philosophy and Economics 20 [Edinburgh, also included, "Kent under Æðelbert." es on Anglo-Saxon royal figures and the re given here; the authors exclude one Christian interpolation, contradictory which have "not one svllable to suggest

Fig. 4. *OEN* vol. 3 appears to have been typed on an IBM Selectric and printed on an offset press.

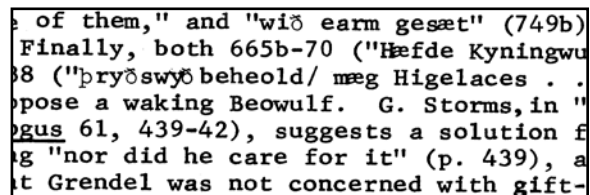
And yet the interchangeability of the Selectric type ball offered at least two decent solutions to the typist's dilemma. IBM being an enormous multinational corporation that sold its products even in tiny Iceland, one could order an Icelandic type ball and either swap it in when an Old English character was called for or put up with the inconvenience of an unfamiliar keyboard arrangement. Fig. 5, from *OEN* 16:2 (1983), illustrates this approach. Alternatively, one could have a custom type ball made up, as appears to have been done in fig. 6, from “Year’s Work,” *OEN* 12:1 (1978). The customized type ball could suffer the same kinds of alignment problems as other customized typewriters, but the interchangeability of type balls meant that one did not lose the ability to type the characters that had been replaced. If one needed, say, brackets, one could swap in an unaltered type ball; and of course interchangeability also improved a typist's ability to handle the modern languages in which Old English scholarship is published.

Late-model typewriters often used interchangeable “daisy-wheels” that offered the same advantages as IBM’s type ball. So-called “electronic” typewriters contained microprocessors and enough memory to store a line or more of text, easing correction and making possible such formatting tricks as right-justification (making the right margin even) and proportional spacing (giving each letter a different amount of space—see fig. 5). The typewriter had reached a remarkable level of sophistication by the time the computer revolution swept away nearly the entire industry.



mīnes wīdlāstum wēnum hogode,  
wæs rēnig weder ond ic rēotugu sæt,  
c se beaducāfa bōgum bilegde,  
yn tō þon, wæs mē hwæpre ēac lāð.  
Wulf! wēna mē þīne  
ydon, þīne seldcymas,

Fig. 5. From *OEN* vol. 16: An Icelandic typeball.



e of them," and "wið earm gesæt" (749b)  
Finally, both 665b-70 ("Hæfde Kyningwu  
88 ("þryðswyð beheold/ mæg Higelaces . .  
pose a waking Beowulf. G. Storms, in "  
agus 61, 439-42), suggests a solution f  
ig "nor did he care for it" (p. 439), a  
t Grendel was not concerned with gift-

Fig. 6. From *OEN* vol. 12 : An altered typeball.

## 2. Early computer printing

The introduction of the personal computer in the late 1970s and early 1980s was in some ways a setback for Old English typography. The earliest personal computers offered a character set that, while much larger than that of a standard typewriter, was “hardwired” into the machine and thus not as flexible as a late-model typewriter. For example, the KayPro II computer, introduced in 1982, could display ASCII (the American Standard Code for Information Interchange), which included these characters:

```
!"#$%&'()*+,-./0123456789:;<=>?  
@ABCDEFGHIJKLMN.OPQRSTUVWXYZ[\]^_  
'`abcdefghijklmnopqrstuvwxyz{|}~
```

It could also display Greek—a fine thing for Greek speakers, but little help to Old English scholars. The IBM Personal computer (1981) and the Apple Macintosh (1984) also based their character sets on ASCII, and added a selection of characters (including æ) that enabled them to handle a number of western European languages—but those sold in the United States could not display Icelandic, and thus not Old English. Early adopters of computer technology often resorted to such workarounds as substituting \$ for þ and # for ð (one typically sent one’s publisher a key along with one’s typescript or disk, indicating the substitutions to be made); or they traveled to Iceland to purchase a computer. It was not until 1985 and the introduction of Microsoft Windows that þ and ð became available on an easily obtained computer platform (and Windows, in fact, was not widely used until 1990).

Early computer printers that produced typewriter-quality output (mostly daisy-wheel printers) were generally slow and too expensive for most professors and students. Individuals usually bought dot-matrix printers, which worked by firing pins in letter-like patterns against a ribbon instead of fully-formed pieces of type. Fig. 7 shows the difference in quality between a t produced by a Selectric and one produced by an ordinary dot-matrix printer of ca. 1989. This difference is a result of the different methods used by typewriters and dot-matrix printers to transfer images to paper. While the typewriter or daisy-wheel printer is an “analogue” device—the pieces of type bear the actual images of characters—the dot-matrix printer is a “raster device,” which works by arranging dots (“pixels”) in straight rows on a rectangular grid (see fig. 8). A computer’s display is a raster device; so are laser printers, inkjet printers, and the imagesetters and platesetters used in high-quality printing. From a reader’s point of view, the difference between the dot-matrix printer



Fig. 7. 1. Typed on a Selectric; 2. printed on a dot-matrix printer; 3. printed on an HP LaserJet II; 4. printed on a LaserJet 1200.

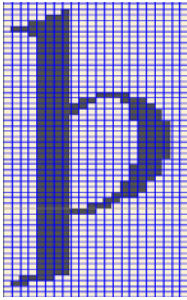


Fig. 8. A raster grid with the letter p.

of the 1980s and the modern platesetter is largely one of resolution. A dot-matrix printer may print at as little as 100 dpi (“dots per inch”), at which resolution characters look dotted or jagged. A platesetter, on the other hand, prints on a lithographic plate at such high resolutions (2400 dpi or greater) that even the most subtle curves appear smooth and well formed (open any recent issue of *Anglo-Saxon England* for an example).

Most modern printers are either laser or inkjet printers. In a laser printer, an image is drawn by a laser beam on a photo-sensitive drum and transferred to paper; an inkjet printer squirts fine jets of ink onto the paper’s surface. While laser printers had been around since the 1970s, the first one that a scholar might dream of owning was the Hewlett Packard LaserJet (1984), which cost around \$3,500. With a resolution of 300 dpi (see fig. 7, no. 3), the LaserJet produced attractive output which, while it did not match the quality of an IBM Selectric, at least was not a disgrace.

But the truly revolutionary printers, and the first truly useful ones for medievalists, were the LaserJet Plus and the Apple LaserWriter, both introduced in 1985. Take note of the year: the Apple Macintosh had appeared in 1984; Microsoft Windows in 1985. What these four products had in common was that the fonts they used were not burned into hardware, but rather loaded dynamically into memory.

### 3. Soft fonts

All computer fonts are software: they store instructions which the processor in a computer or printer must execute to produce images of letters on screen or paper. In early computers and printers, fonts were stored in read-only memory (ROM) chips which could not be altered. But as the cost of both disk storage and random-access memory (RAM, the computer’s working memory) fell, it became practical to store fonts on disk and load them into memory as needed. With these “soft fonts,” the computer–printer combination finally became as flexible as an IBM Selectric, with analogous advantages for Old English scholars: if the fonts packaged with one’s computer or printer could not display or print Old English satisfactorily, one could, at least in theory, acquire fonts that would do the job.

Early soft fonts for personal computers were “bitmap” fonts: they contained a “map” of each character composed of binary digits, or “bits,” specifying which pixels in a raster device to turn on. A single font file pertained to a single resolution and style: one file for “12 point Times,” one for “10 point Times bold,” one for “8 point Times italic.” The LaserJet Plus and its successor, the LaserJet II, both used this kind of font. But in 1984 a new company called Adobe Systems introduced a page description language called PostScript, which Apple licensed to drive its LaserWriter. PostScript used a new font format for printers, called Type 1, which stored each character as a series of points on a very fine geometric grid (see fig. 9). The coordinates of these points, when plugged into a mathematical formula, defined the outline of a letter which could be scaled to any size. The PostScript interpreter in the LaserWriter “rasterized” this outline, filling it up with pixels, and then printed the result. PostScript was popular among graphic designers and publishers, but because of Adobe’s high licensing fees it did not immediately catch on with the general public. To compete with PostScript, Apple Computer in 1991 introduced a new font format called TrueType. Interestingly, one effect of Apple’s new offering was to promote the popularity of PostScript by forcing Adobe’s prices down. But TrueType was destined to become the dominant font format on both Macintosh and Windows computers,

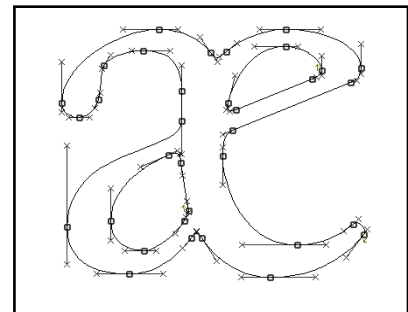


Fig. 9. An outline from a PostScript Type 1 font. The outline must pass through the points marked o; the points marked x define the curves.

largely because it was designed to work precisely as well on the screen as it did on a printer. This was no small trick.

A rasterizer works by first scaling a character's outline to the current size, superimposing it on a raster grid, and then turning on any pixel whose center falls within the outline. This method works well on a modern 1200-dpi laser printer but is problematic on a low-resolution device. The difficulty is that the lines that make up the outline rarely fall in optimal locations on the grid. The result, especially on a screen

<b>1. plain</b>	The quick brown fox jumps over the lazy dog
<b>2. instructed</b>	The quick brown fox jumps over the lazy dog
<b>3. anti-aliased</b>	The quick brown fox jumps over the lazy dog

Fig. 10. Three samples of TrueType text from a computer's display.

display, is that the quality of type ranges from ugly to unendurable (see fig. 10, line 1). Adobe's Type 1 fonts, which were intended mainly for use in printers, included a way for type designers to "hint" their fonts—that is, to mark significant points as a way of helping the rasterizer to fit the character's outline to the grid. This system worked well on the 300-dpi LaserWriter but was not good enough for a 72-dpi screen display. The TrueType format, by contrast, included a sophisticated system of "instructions"—in essence a programming language—that designers could use to fit outlines to the raster grid, normalize stems and curves, and even delete features that could not be rendered well at low resolution. Fig. 11 shows the TrueType outline of **ð** from a single font as instructed for two different devices: for the 72-dpi screen (left) the instructions have deleted the little flags at the ends of the cross-bar and made all strokes equal to the width of a pixel; for the 600-dpi printer version the very same instructions have altered the original character-shape only slightly, nudging the outline onto the gridlines at key points.

Fig. 10, line 2, shows how instructions improve the quality of type on screen. Line 3 adds another enhancement, available on most modern systems: the letters are "anti-aliased" by shading the edges of diagonal and curved strokes with pixels in shades of gray. This method smooths jagged lines and makes the resolution of a device look higher than it actually is.

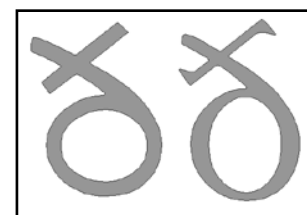


Fig. 11. TrueType outlines as instructed for a screen display (left) and a 600-dpi printer (right).

#### 4. The rise and fall of Junius

The inclusion of **þ**, **ð** and **æ** in the Microsoft Windows character set was a major step forward for Old English scholars. Unfortunately, Macintosh and MS-DOS users still had no way to type Old English, and Windows users who needed letters with macrons or dots were still out of luck. But their long struggle with the typewriter had made Old English scholars resourceful. Through the mid-1980s, articles in *OEN* shared tips for displaying and printing Old English:

**1983:** Donald K. Fry on the basics of word processing and the daisy-wheel he had gotten from a friend in Reykjavík; Milton McC. Gatch on how to get a modified daisy-wheel (17:1, pp. 24-26, 27).

**1984:** William Schipper on the Xerox X-9700 Page Printer, an early laser printer (17:2, pp. 24-30).

**1985:** Susan G. Fein on editing and printing Old English with the WordStar word processing program and a daisy-wheel printer; Greg Waite on the VAX "minicomputer"; Marilyn Deegan and David Denison on a program called Vuwriter (18:2, pp. 36-37, 38-39, 40-43).

**1986:** Robert Boenig on printing Old English on the IBM Quietwriter (19:2, pp. 32-35).

1987: Katherine O'Brien O'Keeffe and Sheryl E. Perkins on WordPerfect and an Epson dot matrix printer; Constance B. Hieatt on the IBM Quietwriter (20:2, pp. 28-30; 21:1, p. 32).

Many of us, indeed, came up with innovative, if sometimes *ad hoc*, solutions to the Old English printing problem during this period. For a time I hooked up a Selectric to my KayPro II, later used a highly customizable print program called FancyFont that coaxed amazingly good printing out of an ordinary dot-matrix printer, and for informal purposes wrote a rough-and-ready program that drove the printer in text mode but shifted it into graphics mode to print Old English characters. During this period specialized Old English fonts for the Macintosh began to appear. By 1985 Patrick Conner had created three bitmap fonts named Exeter, York and Codex (the latter a symbol font for drawing diagrams of manuscript collations); in 1987 he released these via the Compuserve network. Other Macintosh fonts included Gordon Gallacher's Ælfric (an outline font, released in 1991 or earlier), Richard Monaghan's Nero (bitmap, 1993), and Catherine Ball's Times Old English (the classic Times outline font with Icelandic characters moved into accessible locations, 1995 or earlier).

In late 1992, I decided on a whim to design a font based on the Old English type in my copy of George Hickes's *Linguarum Vett. Septentrionalium ... Thesaurus* (Oxford, 1703-05), and to make it an outline font rather than a bitmap font. I named the font "Junius" after Franciscus Junius, who had commissioned the original typeface. In June 1993 I released Junius and several minor fonts (now largely forgotten) on the ANSAXNET server as "The Old English Font Pack for Windows," and that fall I published an announcement of the release in *OEN* (27:1), including a tongue-in-cheek prediction of a revival of the "Saxon" types preferred by early scholars. Once I had begun a career in font design I found it difficult to stop. Junius soon acquired italic, bold and bold italic styles, and also a Macintosh version. Because Junius, with its insular letter-forms, was not terribly practical, it was soon followed by "Junius Modern," based on the same original design but in a modern style, and "Junius Standard," with standard Windows and Macintosh character sets (see fig. 12). I had taken over typesetting "Year's Work" for *OEN* in 1989; by 1995 the "Junius" family of fonts was mature enough to use for this task, and it made its debut as the official type of "Year's Work" that fall.

The Junius family, and especially Junius Modern, quickly became popular. I was gratified, of course, but it soon dawned on me that in making Junius Modern I had actually done a Very Bad Thing. A computer text is a stream of numbers, each one of which represents a letter: for example, on most computers 32 is a space,

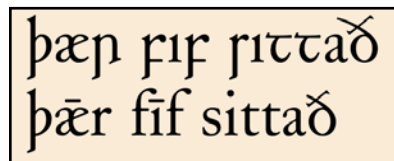


Fig. 12. Junius and Junius Modern.

65 an A, and 254 a þ. The assignment of numbers to characters is called "encoding." As long as you keep all your texts to yourself, you can use as eccentric an encoding scheme as you like; but as soon as you decide to send your file to a colleague or publisher, your encodings had better agree. To facilitate the exchange of files, the International Organization for Standardization (ISO) and other standards organizations had by the 1990s devised a number of encoding standards. Versions of Microsoft

Windows sold in the United States and western Europe, for example, used a slightly altered version of ISO 8859-1, also called "Latin-1"; versions of Windows sold in other countries used other standards (for example, ISO 8859-2 for eastern Europe).

What I had done in creating Junius Modern was analogous to replacing bits of type in a typewriter. I didn't understand at the time that the implications of swapping characters in a computer font are much greater: I had unwittingly created a deviant version of ISO Latin-1 for Windows, and a quite different deviant version of the Macintosh character set. In doing so I had guaranteed that a Windows user and a Mac user would be unable to exchange files, though both had used my font, and neither of them would be able to exchange files with anyone who had not installed Junius Modern. To help sort out the confusion I

had caused, I started in the late 1990s to work on another font, which was intended to conform to the latest standard encoding scheme. Version 0.1 of this font, released in September 1998 under the name “Junicode,” contained 498 characters.

## 5. Unicode, Junicode and MUFI

Until the mid-1990s, most computer texts were made up of eight-bit numbers called bytes. As a byte can represent any of  $2^8$  or 256 values, the maximum number of characters that can be represented by a one-byte encoding scheme is 256. In practice, standard encodings such as Latin-1 reserve a number of encoding slots for non-printing control codes; and many more slots are taken by punctuation, digits, currency symbols, basic mathematical symbols and various useful squiggles. A 256-character encoding scheme is therefore not as lavish as one might guess. Latin-1 has slots for just 56 upper- and lower-case pairs, plus ß and ŷ, for which it provides no upper-case equivalents—114 alphabetic characters in all.

Users of Latin-1 who wish to type in languages not supported by that encoding scheme (such as Greek, Polish, Russian or Old English) or with characters from the International Phonetic Alphabet (IPA), must switch to a different encoding scheme—if one exists to do the job. Desktop computers of the 1980s and early 1990s provided ways to switch between encoding schemes, but doing so was cumbersome, and mixing languages in a standards-compliant way could be a problem; so users needing to mix languages or use IPA symbols usually adopted non-standard solutions. The result was confusion for scholars and publishers who needed to share files.

In the late 1980s a consortium of technology companies began work on a new encoding scheme, called Unicode, intended to unify all existing encoding schemes. This new standard lifted the 256-slot limit that had constrained previous encoding schemes: in a Unicode-enabled computer system, one might effortlessly combine Latin, Greek and Cyrillic alphabets with text in Japanese and Chinese. By the late 1990s the Macintosh, Windows and Linux operating systems all incorporated at least rudimentary support for Unicode; as of 2007 all major operating systems use Unicode internally, and all major text-processing programs recognize it.

The name “Junicode” stands for “Junius Unicode.” (I once meant to change the name, since I thought it was ugly, but it is now too late.) The current version (0.6.13) contains all of the Latin characters I have been able to identify in Unicode, plus IPA, Runic, Greek, and many useful symbols—1929 characters in all (see fig. 13). It has been used for most modern and medieval European languages, transliterations of Sanskrit and medieval Arabic, and several modern African languages. Until early 2006, Junicode was free but informally licensed; it is now an Open Source project, meaning that the source code from which it is built is available to all, and all are free to reuse the font or any part of it in other Open Source projects.

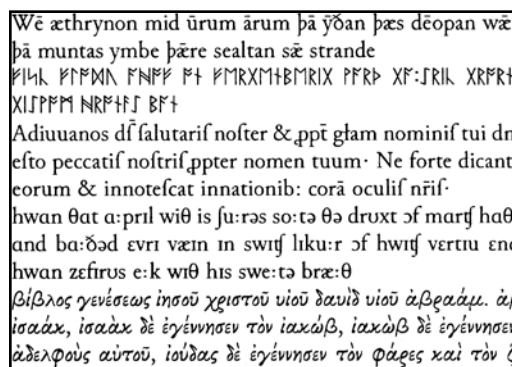


Fig. 13. Some scripts supported by Junicode.

When Unicode was new, it had little support for Old or Middle English. Thanks in no small part to the efforts of Michael Everson, a major contributor to the standard, Unicode now includes support for many minor and archaic scripts, including Ogham, Gothic and Runic. It has all the characters ordinarily used to set Old and Middle English—even the elusive yogh (see fig. 14). Yet there are many characters of interest to medievalists that are so specialized that they are not in the standard. Unicode sets aside a block of encoding slots as a “Private Use Area,” where font designers can encode arbitrary characters without fear



At the application level, support is still spotty. Microsoft Word 2003 for Windows provides some useful capabilities. Word for the Mac reportedly does not, but Mellel, a word processor for the Mac, does. High-end desktop publishing applications such as Adobe InDesign provide pretty extensive, though incomplete, support. A free program called XeTeX, an extension of the venerable TeX typesetting system, provides excellent support, but it is non-interactive and therefore difficult for some users to learn.

Here are some advanced typographical features likely to be useful to Anglo-Saxonists (it is a very small subset of the large number of OpenType features available):

1. **Ligatures.** Most typesetting systems can automatically substitute ligatures like **fi** for combinations like **f+i**. Some Adobe fonts also support a large number of “historical ligatures” of the kind one finds in books printed up through the eighteenth century.
2. **Substitution of precomposed letter+diacritic combinations.** Unicode contains a large number of precomposed characters; but actually using these can be tedious. It is far better, and easier once one gets the hang of it, to type a letter followed by one or more diacritics and let the system worry about substituting the precomposed character if it is available.
3. **Diacritic positioning.** Explained above: if a precomposed character is not available, the system should be able to position diacritics correctly anyway.
4. **Language sensitivity.** Letter shapes may differ depending on language. For example, the ogonek on a Polish **e** is positioned differently from the hook on an **e** in a Latin or Old English text; and yet one uses the same Unicode character for both. Further, both **þ** and **ð** look different in Icelandic from the way they do in many editions of Old English texts (most contemporary publishers use the Icelandic forms for Old English, since they are easy to get). Modern systems allow one to specify the language of any stretch of text: the system should automatically choose the correct letter-form for the language.
5. **Stylistic alternates and sets.** OpenType allows one to change the style of text by substituting characters on the fly while the underlying text remains unchanged. An example in Junicon is that one can automatically substitute insular for modern letter-forms to reproduce the look of the original “Junius” font. One can also substitute long s (**f**) for **s** without changing the underlying text.
6. **Swash letters.** Medieval manuscripts often add flourishes to letters, and some editors, especially of Middle English texts, reproduce these, since it is often uncertain whether they are significant. Junicon has an array of letters with flourishes (requested by Middle English editors), and these can be displayed via the OpenType “swash” feature without changing the underlying text.

These features are supported by Junicon (see fig. 16) and a number of other fonts. Perhaps the most important feature, diacritic positioning, is supported by MS Word (but not by InDesign); but the only application I know of that supports all of them is XeTeX. AAT and OpenType promise to make available important new typographical resources for Old English scholars and their publishers; but as seems always to be the case with computers, technological nirvana is somewhere beyond the horizon.

<p><b>Ligatures.</b> first flat difference difficult afflict.  <b>Diacritics.</b> æt is fōt  <b>Language Sensitivity.</b> Um haustit sendi Mørðr Valgæ  narr myndi vera einn heimi, en lið alt myndi vera ni  heyverkum. Riðu þeir Gizurr Hviti ok Geirr Goði aus  spurðu þat, ok austr yfir sanda til Hofs.  <b>Stylistic Sets.</b> ƿe æthþýnon mid urum arum þa ýðan ƿ  gerapon eac þa muntar ýmbe þære fealtan fæ ftranbe  dum hræggle and gerunðfullum pinðum þær gericobor  þære fægereftan þeobe.  <b>Swash letters.</b> breedt ook of egg</p>
--

Fig. 16. Some advanced typographical features in Junicon



Fig. 17. Some free fonts of interest to medievalists. These are all available without charge and licensed under generous terms (i.e. unaccompanied by onerous restrictions or threats of prosecution). Cardo and Junicode implement the MUF1 recommendation in whole or in part. Charis SIL and Doulos SIL have especially strong IPA support. The DejaVu fonts, Gentium and Junicode have matching italic faces; DejaVu and Junicode also have bold and bold italic faces.