Phonetic symbols in word processing and on the web

J.C. Wells
University College London
E-mail: j.wells@phon.ucl.ac.uk

ABSTRACT

Unicode provides a single coding system for all scripts used in printing the languages of the world, and includes the entire International Phonetic Alphabet. A standard Unicode-based phonetic font is now routinely bundled with the software supplied for new personal computers. Unlike the situation four years ago, most current browsers, word-processing packages, fonts and printers support Unicode. These welcome developments render obsolete the unstandardized and proprietary phonetic fonts hitherto in use. They are, however, poorly documented and have not been widely publicized. Several Unicode-based phonetic fonts are now available, and are listed and compared. The main problem outstanding is keyboarding: how does the user get the symbols into a document? A range of practical solutions are suggested.

1. INTRODUCTION

Since the last ICPhS four years ago, the availability of phonetic symbols for computers has been transformed. In 1999, we still mostly depended on special customized fonts, which were severely restricted in size and lacked standardization. Now, though, a standard Unicode-based phonetic font is routinely supplied with new PCs and is thus available not only to specialists but to the general public. All current browsers support it, as does the industry-standard word processor Word; authors of web pages can use it with confidence.

But very few people know about it. These welcome developments are poorly documented and have not been widely publicized.

2. PRE-UNICODE CODING SYSTEMS

There are three ways in which non-standard characters (and in particular, phonetic symbols) can be handled by computers.

i. ASCIIization. This is all that was available in the 1980s. Symbols missing from the basic ASCII set are replaced by ASCII surrogates. For example, the phonetic symbol string tʃʌŋk might be represented as tSVNk. One widely-used surrogate set is known as SAMPA (see www.phon.ucl.ac.uk/home/sampa). There are several others. ASCIIization is a makeshift solution, though it is very robust for applications such as e-mail, where only the 7-bit ASCII set is reliably transmitted.

ii. Custom one-byte fonts. These became available during the 1990s. For non-ASCII characters a special font (in our case, a phonetic font) is used. It must be selected when a special symbol is required, and deselected when the standard Latin alphabet is required. As far as phonetic fonts are concerned, various proprietary and free fonts are available, including those provided by the Summer Institute of Linguists (www.sil.org). This is a reasonably satisfactory solution, and is what many phoneticians still use. Its main disadvantage is that the special font has to be installed in every computer involved (unless for some specific document it is embedded into the file, as with .pdf files). Furthermore, the lack of standardization means that different fonts use different coding and different keyboard layouts, so that conversion from one to another is difficult or impractical.

Some of the rival codings are illustrated in fig. 1. While fonts agree on what should be mapped onto ASCII upper-case A, D and N, they wholly disagree in the cases of J, P and Q. Hence material encoded for one font appears as gibberish if viewed in another font.

code position (decimal)	65	68	78	74	80	81
ASCII	A	D	N	J	P	Q
SILDoulos IPA93	ɑ	ð	ŋ	ʲ	ɵ	æ
IPA-samd (UCL)	ɑ	ð	ŋ	ɲ	ʊ	ɒ
IPAKiel (Linguist's)	ɑ	ð	ŋ	ʝ	ø	ɖ
IPA Roman 1 (Atech)	ɑ	ð	ŋ	ʒ	ʏ	θ

Fig. 1. Symbols mapped onto selected ASCII characters

Conversely, the schwa (**ə**) is mapped onto ASCII 171 («) in SILDoulos IPA93 and IPAKiel, but onto 64 (@) in IPA-samd and onto 60 (<) in IPA Roman 1.

Away from phonetic symbols, we have all experienced the tiresomeness of receiving a message composed in Cyrillic, Greek, Hebrew or Thai, with a font using a character set that maps these characters onto code points 128-255 — or even a message from Eastern Europe which maps local accanted letters onto them — only to find that our mail reader presents them as gibberish.

iii. With **Unicode**, the multi-byte coding system to which the rest of this paper is devoted, these problems disappear. This is the appropriate system for the new millennium.

ISBN 1-876346-48-5

3. UNICODE

In Unicode there is a single coding system for all scripts of all languages. It includes phonetic symbols. This became possible with the transition from one-byte to multi-byte coding. "Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language." [1]

For a detailed account of Unicode, see [2], also [6]. Unicode characters are numbered hexadecimally and organized into blocks. First comes the basic ASCII set (Basic Latin), numbered 0020 to 007F (= decimal 32 to 1270). Next follows the Latin-1 Supplement, covering special characters required by the languages of Western Europe, numbered 00A0 to 00FF (dec. 161 to 255). This covers such characters as **à ê õ í Ä ñ ß þ**, and includes some which, like the basic Latin lower-case letters, also do duty as phonetic symbols: **æ ø ç ð**. Up to this point Unicode coincides with the one-byte Windows standard character set (codepage 1252) seen in the pre-Unicode fonts supplied with Windows 3.1 in Western Europe, the Americas and Australasia. Windows also has some extra characters in the range 0080-009F (dec. 128-159), which in Unicode are defined as control characters. Macintosh has its own, non-standard, character set, which is quite differently coded except for the basic ASCII range.

The first block above the old limit of 00FF (dec. 255) is known as Latin Extended-A, 0100 to 017F (dec. 256-383). It is devoted to Latin letters with diacritics required for various languages of the rest of Europe, e.g. **ā ă ą ć ĉ ċ č**. These include three further characters that we know as IPA symbols: **ħ**, **ŋ** and **œ**. Next comes Latin Extended-B, 0180 to 024F (dec. 384-591), which includes various African and other letters, and also some non-IPA or ex-IPA (withdrawn or superseded) phonetic symbols: **ƀ ƈ ƕ ƙ ƞ ƥ ƪ ƫ ƨ ƨ**. This block is also where the current IPA click symbols are to be found: **ǀ ǁ ǂ ǃ**.

Only now do we reach the block entitled IPA Extensions, coded 0250 to 02AF (dec. 592-687). This comprises over eighty characters. (There is no space to show them all here, but you can see them on the web [3].) As well as currently approved symbols, this block also includes the now deprecated **ɪ** and **ɷ**, **ɻ** and **ɾ**, **ʃ** and **ʒ**, the former click symbols **ɨ ʖ ʗ** and **ʞ**, and also **ɖʃ** and the Sinologists' **ɪ ɻ** which as far as I know were never IPA-approved, although some scholars (e.g. [4], implicitly) have thought they were.

Provision for phonetic symbols continues with a block called Spacing Modifier Letters, 02B0 to 02FF (dec. 688-767). This comprises diacritics that occupy their own horizontal space, e.g. **ʲ ˠ ʰ**, and also stress marks and length marks. Diacritics that are attached above, below, or through another letter occupy the next block, Combining Diacritical Marks, 0300 to 036F (dec. 768-879). Here we find IPA diacritics such as are exemplified in **n̥ n̩ n̪ n̺ n̻**. Two diacritics present a problem of classification: are they Spacing or Combining? The no-audible-release diacritic is classified as Combining, although it normally occupies a space of its own: **t̚** (I have added a space character before it here, so that it does not overstrike the **t**). The rhoticity diacritic is classified as Spacing, although always attached to the letter with which it is associated: **ɑ˞**.

This still leaves a few IPA symbols in other blocks. The theta symbol **θ**, as might be expected, is in the Greek and Coptic block, 0370 to 03F. Perhaps less obviously, so are the beta **β** and chi **χ**. (For gamma and epsilon, though, Unicode rightly distinguishes between IPA **ɣ ɛ** and Greek **γ ε**.) The Unicode handbook, by the way, wrongly implies that lambda **λ** is also an IPA symbol.

This constitues a very satisfactory coverage of the IPA alphabet. Altogether, the latest version of Unicode (3.2) specifies 95,156 different glyphs, three-quarters of which are Chinese or Korean characters. (Among them are the symbols of the Chinese phonetic alphabet Bopomofo.) A preliminary announcement [5] relating to the forthcoming version 4 includes reference to a new block (1D00 to 1D7F) called Phonetic Extensions. I do not know what this will contain.

So much for the standard. What about its implementation? For Unicode to work in practice, the operating system and the applications must be Unicode-enabled, and an appropriate font or fonts must be installed. Printer drivers, too, must be Unicode-enabled; and the user must have a way to input characters that cannot be entered directly from the keyboard.

4. OPERATING SYSTEMS AND APPLICATIONS

"The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, … Microsoft, … and many others. Unicode is required by modern standards such as XML, Java … and is the official way to implement ISO/IEC 10646. It is supported in many operating systems, all modern browsers, and many other products. The emergence of the Unicode Standard, and the availability of tools supporting it, are among the most significant recent global software technology trends." [1]

Operating systems. From Windows 95 onwards, the Windows operating system has been in principle Unicode-compliant. So, from OS 8.5 onwards, has the Macintosh system.

Word processors. A Unicode-compliant operating system is of little use unless appropriate applications software is also available. For Windows, this began with Word 97, other parts of MS Office following gradually. However, provision for non-ASCII characters in the various parts of MS Office was for some years flaky and buggy — not to mention the shortcomings of printer drivers. Gradually, however, the bugs have been sorted out. By the time of writing (February 2003) pretty well everything in the current version of MS Office is working correctly. Many other popular Windows applications, however, — WordPerfect, the mailer Eudora and the web editor

ISBN 1-876346-48-5

Dreamweaver, for example — are still not Unicode-compliant.

Despite the compliance of the OS, very few Macintosh applications can handle Unicode. I have to say that this is one area where the Macintosh lags seriously behind, while Microsoft deserves our unaccustomed praise.

Browsers. For the World Wide Web, Unicode has been the preferred encoding since version 4 of HTML. Every Internet Explorer or Netscape browser from version 4 onwards has been able (in principle) to display web pages encoded in Unicode, given only that a font including the relevant characters was available. Windows browsers now generally work correctly. Macintosh browsers, however, ignore Unicode fonts, instead using Apple's WorldScript technology and proprietary encodings and attempting to map characters from these to the appropriate Unicode characters [6].

The popular web search engine Google is fully Unicode-compliant, though inputting non-keyboard characters requires cut-and-paste. A search for a string containing phonetic symbols can of course **only** be done using Unicode.

5. FONTS

From 1997 onwards, a number of Unicode fonts have routinely been bundled with Windows/Word. (This does not mean that the entire Unicode character set has been available: normally, Unicode fonts cover only a subset of the entire range.) In Western countries, they typically covered all the Latin-alphabet diacritic combinations required for European languages (including, for example, Latvian and Maltese), plus Greek (monotonic), and Cyrillic (including the non-Russian characters needed for e.g. Serbian and Ukrainian). Over the years more and more has been added, so that today one may also routinely receive fonts also covering Arabic, Hebrew, and Vietnamese. In Asian countries, naturally, Unicode fonts covering the local languages are supplied.

Importantly for us, Windows 98 and later comes with a font, **Lucida Sans Unicode,** that includes the entire International Phonetic Alphabet. This font, supplied to millions of users throughout the world, has for the first time ensured that a standard phonetic font is widely available for word processing and web browsing applications. It is, however, the **only** font covering the IPA that is made available in this way. It is a sans-serif font, with one or two idiosyncratic symbol shapes: see fig. 2.

That's not actually the part I was thinking of.
ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv

Fig. 2. Lucida Sans Unicode

Microsoft offers two other fonts that cover the IPA. They have been less widely distributed. One is **Arial Unicode MS**, which for a time was available for free download from the Microsoft website, but is now supplied only as part of Microsoft Office XP and Microsoft Publisher 2002. This enormous sans-serif font covers the entire character set of Unicode version 2, some forty thousand characters. The other is **MS Mincho**, supplied as part of the Office XP Japanese language pack. Although it covers most of the IPA Extensions block, this font lacks diacritics, length marks, stress marks, and various other symbols, as well as being remarkably ugly. It is not a contender for our attention.

In fig. 3. samples of various Unicode phonetic fonts known to me at the time of writing are displayed for comparison.

Lucida Sans Unicode:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Arial Unicode MS:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
SILDoulosUnicodeIPA:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Gentium:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Alphabetum:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Cardo:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Code 2000:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Junicode:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv
Thryomanes:	ðæts ˈnɒt ˈæktʃəli ðə ˈpɑːt aɪ wəz ˈθɪŋkɪŋ ɒv

Fig. 3. Unicode phonetic fonts

The remaining fonts to be discussed are not from Microsoft. They are all serif fonts. **SILDoulosUnicodeIPA**, still at the beta stage, is from SIL, who gave us the widely used single-byte phonetic fonts. It is described as usable for Linux and Macintosh as well as Windows systems, "to the extent the application allows" [7]. It is pioneering in that "it has diacritic placement built into the font. There is no longer a need for multiple versions of the same diacritic. Formation of contour-tone ligatures is also supported in this way". **Gentium** is the separate initiative of a member of the SIL International staff, and is designed to be "highly readable, reasonably compact, and visually attractive". [8]

The comprehensive **Thryomanes** is available from [9]. Two other fonts, **Cardo** [10] and **Junicode** [11], both still at the beta stage, currently lack certain symbols, notably the stress marks and the clicks; Junicode also lacks most diacritics. (In the specimens, apostrophes and colons replace the missing stress and length marks.)

These five fonts may be downloaded from websites free of charge. Two others are shareware and ask for a small contribution: **Alphabetum** [12] and **Code 2000** [13].

6. INPUTTING

There remains the question: how can the user input Unicode characters not represented on the keyboard? There is no "input locale" [14] available for inputting phonetic symbols. It is true that there are one or two third-party keyboard software packages available, but I find them awkward and unreliable. Nevertheless, there are various keyboarding techniques available that require no special software. The discussion that follows is restricted at first to the creation of documents including phonetic symbols using Word 97 or later under Windows 98 or later.

First, obviously, it is necessary to ensure that a Unicode

 ISBN 1-876346-48-5

phonetic font (e.g. Lucida Sans Unicode) has been installed. It must next be selected using the Font box or the Format menu. After that, there are various ways of proceeding. Choose between the following.

1. Do Insert | Symbol. Find the symbol you want in the drop-down box that appears. Double-click it.
2. Use the program Character Map, which can be launched from Start | Programs | Accessories | System Tools | Character Map. If necessary, re-select the font (e.g. Lucida Sans Unicode). Find the symbol you want, then Select, Copy. Paste it into your document.
3. Copy and paste the symbol you want from a document or web page that contains it. There are web pages [15, 16] designed with this in mind.
4. Create a Word shortcut (macro) for frequently required characters.. You can use one of two methods: Shortcut Key or AutoCorrect. With the first you assign a keystroke for the character, e.g. Alt-@ for ə. With the second, you assign a string that must begin and end with a non-alphanumeric character, e.g. |@|. (This method is described in detail in [17].) Whichever you choose, once you have set up the shortcut you can store it as part of the document template.
5. In Word 2002, it is reportedly possible to enter characters by Unicode hexadecimal number in the Insert Symbol box, and to toggle between a displayed character and its number [18]. (In earlier versions of Word, Alt+number, with a leading zero, from the numeric keypad, can be used only to enter the basic range 032-0255.)

For other applications, one of these methods may work. But generally the easiest approach is to create a Word document first. Then one can copy and paste the character or string desired from this document into the other application. For example, this is the way to input a phonetic character into a Google search string (see above).

To include Unicode phonetic characters when editing a web page, there are two possible methods:

- Create the page as a Word document, as above, then Save As HTML/Save As A Web Page.
- Alternatively, write direct HTML/XHTML, using decimal or hexadecimal numeric character references.

The first will cause Word to convert non-ASCII characters into the Unicode encoding known as UTF-8, which can be interpreted by a browser. For the second, you write the Unicode number of each special symbol between `&#` and `;` (decimal), or between `&#x` and `;` (hex). For example, to include the velar nasal symbol ŋ, which has the Unicode number 014B (dec. 331), write `&#331;` or `&#x014B;`.

For the transcription of the English word *thing*, θɪŋ, write `&#952;&#618;&#331;` or, alternatively, `&#x03B8;&#x026A;&#x014B;`. Unicode numbers are available in the manual [2] and its web version [3]; for phonetic symbols, see particularly [16].

It is best to declare the encoding in the head section of the (X)HTML file. Put
`<meta http-equiv="content-type" content="text/html; charset=utf-8" />`.

Apart from the shortage of Macintosh software, the problem of phonetic symbols for word processing and the web is essentially now solved.

REFERENCES

[1] "What is Unicode?", World Wide Web page http://www.unicode.org/standard/WhatIsUnicode.htm, 2003

[2] The Unicode Consortium, *The Unicode Standard, Version 3.0*, Reading MA: Addison-Wesley, 2000

[3] http://www.unicode.org/charts/PDF/U0250.pdf, 2002

[4] P. Kratochvíl, *The Chinese language today*, London: Hutchinson, 1968, pp. 24, 28-29.

[5] http://www.unicode.org/Public/4.0-Update/Blocks-4.0.0d1a.txt, 2002

[6] http://www.alanwood.net/unicode/

[7] http://www.sil.org/computing/catalog/ipa_unicode.html

[8] http://www.sil.org/~gaultney/gentium/

[9] http://www.io.com/~hmiller/lang/

[10] http://scholarsfonts.net/cardofnt.html

[11] http://www.engl.virginia.edu/OE/junicode/junicode.html

[12] http://user.dtcc.edu/~berlin/font/unicode.htm

[13] http://home.att.net/~jameskass/

[14] http://www.microsoft.com/globaldev/DrIntl/faqs/Locales.mspx

[15] http://www.phon.ucl.ac.uk/home/wells/phoneticsymbols.htm

[16] http://www.phon.ucl.ac.uk/home/wells/ipa-unicode.htm

[17] http://www.phon.ucl.ac.uk/home/wells/Eureka.doc

[18] http://www.alanwood.net/unicode/utilities_editors.html#word2002

[19] http://www.unicode.org/charts/