Typesetting CJK and other exotic characters using LaTeX and XƎLaTeX

Anything goes (well, almost...)

Abstract
This paper tries to illustrate some of the particularities of typesetting CJK characters using several flavors of LaTeX. Special attention is given to Japanese. A short introduction is given about the nature of the character scripts and the special demands those alphabets put on character and font encodings. Typesetting Japanese using p(te)TeX, LaTeX, Lambda, and XƎLaTeX is discussed. Special discussion is given to XƎLaTeX, and the possibilities of including annotation markup and vertical typesetting in Japanese texts using XƎLaTeX. It will be shown that although typesetting vertical material is possible with XƎTeXv0.997, more development work will be needed in this area to create a dependable vertical typesetting system.

Keywords
XƎTeX, XƎLaTeX, CJK, unicode, horizontal and vertical typesetting, Japanese

This paper is the result of a question that was asked recently on the Dutch LaTeX mailing list. The question was whether and how it would be possible to typeset Japanese with LaTeX. In 2002, I did an internship in Japan, and remembered that I had to install pTeX, which was a patched version of (then) teTeX 2.2, and then some trickery was necessary to have the correct fonts show up in the final PostScript file. Thinking that life must have become easier in the meantime, I set out on a mission to see what the different flavours of LaTeX can do in scope of the CJK languages[1]. This paper is a (not very technical) summary of my experiences. The major focus is on Japanese, but examples in all CJK languages are provided. The paper will start with a bit of history to explain the origins of the Chinese character script, which provides insight into the considerable difficulties that (used to) exist in using CJK on computers. Then we'll discuss a bit about character and font encoding, which will be followed by a listing of possibilities of incorporating CJK texts into a LaTeX document. The paper will be concluded by some examples using XƎTeX which should be reproducible by anybody who has a recent version of TeXLive, Adobe Acrobat reader, and an internet browser.

History of Japanese characters

This short introduction follows that of [1]. Chinese characters originated in the Yangtze River region of China, between 2000 – 1500 BC. Starting as simple pictographs, the characters evolved to also express abstract concepts. Several pictographs could be combined into one character to express complex ideas, and provide different nuances in meaning. The well known square-formed characters (known in Japanese as kaisho, 楷書) developed around 200 AD. A more or less formalized system evolved where each character has a main part expressing the base meaning of the character (Japanese: 部首, 'radical'), adorned with other radicals to express pronunciation and nuance of meaning. It should come as no surprise that such a system can easily lead to a large number of different characters. Around 200 AD, there were an estimated 50.000 characters.

Chinese characters entered Japan between the third and fourth century AD, mainly by Chinese and Korean monks and scholars. In fact, the word kanji (漢字) literally means 'letters of the Han Dynasty' (206 BC – 220 AD). In Japan, kanji were initially only used to write Chinese texts, but over time kanji came to be used for Japanese texts as well. This lead to the development of different pronunciations for the same character. For example, the kanji 国 is pronounced 'kuni' in Japanese reading (kun yomi (訓読み), litt. 'reading for meaning'), and 'KOKU' in Chinese reading (on yomi (音読み), litt. 'reading for sound'). Usually, kanji appearing by themselves are read in kun yomi, and in combination kanji are read in on yomi: 母国 BOKOKU, 母 haha, 国 kuni.

The fundamental differences between Chinese and Japanese are that Japanese is an inflected, polysyllabic, non-tonal language, whereas Chinese is the opposite. Over time, it became necessary in order to write Japanese to come up with a way of expressing the inflected parts of verbs, for instance. From around the seventh century AD a system developed to express these inflected parts by standard kanji used phonetically: man'yōgana[2]. The man'yōgana eventually led to the kana (仮名, 'assumed names'), which are purely phonetic. The modern descendants of kana are hiragana and katakana. In modern Japanese, the non-inflectional part of a verb is written with kanji, and the inflected part is written in hiragana: 見る 'to see', 見ない 'to not see'. Katakana is used to write foreign words in Japanese transcription: ソースコード 'source code'.

In Chinese it is possible to distiguish between homophonic kanji by their tonality, but in Japanese that is not possible. As a result, many different kanji obtained an identical pronunciation. Over time in China the pronunciation of standard Chinese changed. The Japanese incorporated many of the 'newer' pronunciations of the existing kanji into their vocabulary, so that finally each character may have many different pronunciations (examples are 下, 'below', which has ten pronunciations, and 生 'life', with nine), and there are many characters sharing the same pronunciation. My electronic dictionary lists 323 kanji with pronunciation 'kō', and 267 under 'shō'. Because there are so many homophonic kanji, many television programs are subtitled.

Much debate rages over the total number of kanji. The famous Dai Kan-Wa Jiten (大漢和辞典)[3] 'Great Chinese - Japanese Dictionary'), published since 1955, contains a total of 49.964 kanji (although most of those differ only in their radicals). After 1945, the Japanese ministry of education tried to standardize a list of kanji and produced the 'tōyō kanji' list, litt. 'temporary use kanji', with 1850 kanji, 881 of which are known as 'kyōiku kanji' which are taught in the first six years of school. In 1981, the list was revised to become the 'jōyō kanji' list (general use kanji) with 1945 kanji, 996 of which are taught. Adherence to this list is not strictly enforced, and especially in scholarly works, literature and poetry many non-jōyō kanji can be found. For example, Kodansha's essential kanji dictionary [2] lists 1945 kanji, while the Compact Nelson [3] lists 3068 kanji, and my electronic dictionary lists 6355.

Traditional and simplified Chinese

In China simplified characters have been used for many centuries, but those were not used in print widely. From 1949, the Communist government began assembling official lists of simplified kanji for everyday use in print. The current official list of simplified Chinese contains 2249 characters. This list is also the official list for Singapore and Malaysia. However, in the parts of China not influenced by communism (Taiwan, Hong Kong and Macau) the traditional characters have been in use continuously, of which no definitive number exist. As an illustration of simplified resp. traditional characters, consider the words 'China' and 'university': 中国 vs. 中國, 大学 vs. 大學.

Korea: hangul, chosŏn'gŭl and hanja

In Korea Chinese characters were also introduced very early on. The typical hangul alphabet (한글) was officially introduced in 1443 by King Sejong the Great. In its base form, hangul is a purely phonetic alphabet, with each symbol representing one sound. However, this system is complicated by the fact that several symbols can be assembled into one character to represent one syllabic block. For example, the word 'hangul' is composed of two syllabic blocks 한 + 글, which are each composed of three symbols: 한 + 글 = 'ㅎㅏㄴ' + 'ㄱㅡㄹ', 'h' 'a' 'n' + 'g' 'u' 'l'. There are 11.172 valid combinations in hangul. Apart from hangul, Chinese characters, known as hanja, are also still used on a small scale, for instance in proper names, official paperwork etc. In North-Korea the same hangul alphabet is used, although it is called choso˘n'gŭl (조선글), and no hanja are used.

Character encoding and font encoding

A computer can only handle information in the form of a stream of bits, and thus for a computer to handle characters, one needs a one-to-one mapping, mapping each character to a unique numerical representation. This numerical value is subsequently transformed into a sequence of bits in a prescribed manner. The number of characters that can be encoded depends on the number of bits that is used for the mapping (encoding). Traditional ASCII uses 7 bits for encoding, allowing a total of 128 possible characters, which is too limited to express even the simplest character lists in the CJK languages. For this reason, different encoding schemes were developed for CJK. For

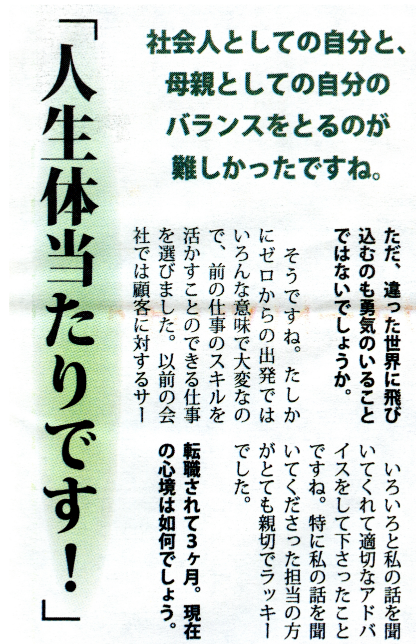
Japanese, EUC-JP, JIS and SJIS were developed. EUC (Extended Unix Code) is a multi-byte encoding; each character is encoded using either 1, 2 or 3 bytes, and each byte is capable of representing 94 characters (several bits per byte are required to distinguish whether the byte is part of a one-, two- or three-byte character, and hence not all bits are available to encode characters). Besides EUC-JP, there are EUC-CN (Chinese), EUC-KR (Korean) and EUC-TW (Taiwanese). EUC-JP is the norm for Unix(-like) operating systems.

JIS (Japanese Industrial Standards) consists of 6879 kanji and a specification for encoding into one or two bytes. A maximum of $94 \times 94 = 8836$ positions are available in JIS. A competing encoding is Shift-JIS (SJIS), originally developed by Microsoft (and others). SJIS differs from JIS in how the numerical values of characters are translated to one or two bytes. Because of the nature of SJIS, it is difficult to detect SJIS encoding automatically, often resulting in a messy screen, known in Japanese as 文字化け (mojibake), 'characters in disguise'. Individual vendors use the space not taken by the JIS character set to add their own characters to SJIS. For example, mobile phone operators use this space to encode emoticons, amongst other things. Microsoft uses their own extended SJIS in Windows (Codepage 932)[4].

Unicode[5] is an encoding standard with enough room to encode millions of different characters in one large set of assigned code points. Unicode provides three different ways to translate characters code points into a form capable of transmission: UTF-8 (Unicode Transformation Format 8) translates the code points into 8 bit units: a sequence of 1, 2, 3 or 4 bytes; UTF-16 translates into 16 bit units; and UTF-32 translates into 32-bit units. The total number of possible code points is 1114112 ($2^{20} + 2^{16}$).

Other encodings, unicode and han unification

For Chinese, Taiwanese, Hong Kong-ese, and Korean different encoding schemes were developed (Big5 for traditional Chinese, BG for mainland Chinese, KS for Korean). This makes it virtually impossible to typeset more than one of the CJK languages within the same file, because different characters would resolve to the same numerical presentation, and it would depend on the font encoding which character is actually displayed. Only unicode encodes all the characters separately, making it possible to use all kinds of alphabets indiscriminately in the same file. To reduce the number of CJK characters in unicode, the so-called 'han unification' is implemented, where several variants of the same character common to the CJK languages are mapped to the same unicode position. This leads to occasional protest, for instance when a character historically used for a proper name is designated as a variant of another character. The specific variant can then no longer be encoded separately in unicode, and cannot be typeset on a computer.



「人生体当たりです！」

社会人としての自分と、
母親としての自分の
バランスをとるのが
難しかったですね。

ただ、違った世界に飛び込むのも勇気のいることではないでしょうか。

そうですね。たしかにゼロからの出発ではいろんな意味で大変なので、前の仕事のスキルを活かすことのできる仕事を選びました。以前の会社では顧客に対するサー

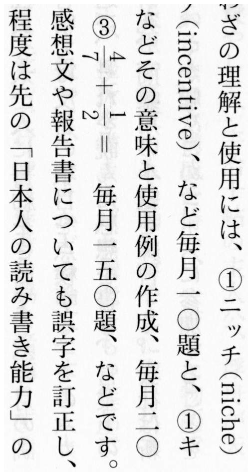
いろいろと私の話を聞いてくれて適切なアドバイスをして下さったことですね。特に私の話を聞いてくださった担当の方がとても親切でラッキーでした。

転職されて3ヶ月。現在の心境は如何でしょう。

Figure 1. An example showing the combined use of horizontal and vertical typesetting in a Japanese news paper article.

LaTEX and CJK

To typeset a text, the computer will read the input stream, and interprets a given sequence of bits as representing a certain character, based on the character encoding used. The corresponding character in the font set should then be displayed. If one has a font with the same font encoding as the input encoding, this implies a one-to-one mapping. If on the other hand a unicode font set is used with, say, SJIS encoding of the file, the SJIS characters from the input stream have to be translated to unicode values in order to display the correct character on the screen using the unicode font. Traditional LaTEX has several problems here because of built-in limitations: EUC or (S)JIS input has to be read, and a translation provided to typeset the cor-



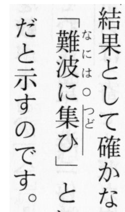
ざの理解と使用には、①ニッチ(niche)
(incentive)、など毎月一〇題と、①キ
などその意味と使用例の作成、毎月二〇
③ $\frac{4}{7}+\frac{1}{2}=$ 毎月一五〇題、などです。
感想文や報告書についても誤字を訂正し、
程度は先の「日本人の読み書き能力」の

Figure 2. An example showing various features of Japanese typesetting: Latin text is set sideways in vertical typesetting. The `equation' is also set sideways. The two punctuation marks and are set in burasage.

rect characters at the correct location, and the resulting DVI stream has to be correctly translated to a PS file (provided that an adequate PS font is available with glyphs for the kanji). Legacy TeX only allows for 256 character (1-byte) encodings. Several patches and packages have been developed over the years to circumvent these problems, as will be discussed later.

Specific typographic rules exist for CJK. In the case of Japanese, texts can be either read horizontally from left to right (LTR), or vertically from right to left (RTL). Almost all printed material in Japanese is set vertically: news papers, magazines, manga etc. Publications in the 'hard' sciences are usually set horizontally because of the presence of equations. Advertisements in printed matter, tabloids etc commonly feature both horizontal and vertical typesetting. Legacy TeX will only allow LTR typesetting. e-TeX allows LTR and RTL. XeTeX is the only flavor capable of setting texts vertically (although this depends on the specifics of the font used, as will be shown later). In figure 1 an illustration provided showing mixed use of horizontal and vertical typesetting in one news paper article.

A common misconception is that CJK languages do not have kerning. Although it is essentially true that all characters are thought of as being written on a square grid, there are characters that do not necessarily occupy a full character position (the punctuation marks 「,」,。 and 、for instance). Also, some characters are 'denser' than other characters and need a bit more room around them for legibility[6]. Most CJK fonts do not support kerning, but professional DTP software does. Another specific feature is that punctuation marks are allowed to protrude into the margins (burasage). See figure 2 for an illustration.



結果として確かな
「難波(なには)に集(つど)ひ」と、
だと示すのです。

Figure 3. An example of ruby, or furigana. The first two characters are glossed for pronunciation. The punctuation mark marks a grammatical rule. The last ruby denote a highly uncommon pronunciation.

Another feature of Japanese is 'ruby': hiragana characters printed above kanji (in horizontal texts) or to the right of kanji (in vertical texts) to indicate pronunciation[7] as illustrated in figure 3. Of course, ruby is most commonly encountered in publications for young readers, but you see it occasionally on name tags and official paperwork.

Japanese fonts are usually available as Gothic and Mincho. Gothic is comparable to sans-serif, mincho is comparable to a serif font, and is most commonly used for printing. For manga etc, a 'hand-written' style is usual, while for poetry the 'cursive' style is commonly used. In cursive, the brush does not leave the paper when writing a character, yielding highly stylized and abstracted characters (in extreme cases denoted as 'grassy'). Please refer to the figures for some illustrations of the different styles of kanji.

漢字は難しい。時々全然読めない。

Figure 4. Mincho typeface (Kozuka Mincho Pro-VI)

漢字は難しい。時々全然読めない。

Figure 5. Gothic typeface (Kozuka Gothic Pro)

漢字は難しい。時々全然読めない。

Figure 6. Manga style hand-written typeface (YOzFontN04)

漢字は難しい。時々全然読めない。

Figure 7. Cursive typeface (HakusyuSeigyosyoKyo)

漢字は難しい。時々全然読めない。

Figure 8. Highly cursive, `grassy' typeface (HakusyuSousyuKyo)

国会議事堂前

Figure 9. Typeface `tensho', used on seals and stamps (HakusyuTensyoKyokan). Note that even for Japanese this type of writing is not always easy to read (国会議事堂前).

Integrating CJK characters into LaTeX documents

As it turns out, there are several ways of incorporating CJK into LaTeX documents, and all of these have their strong and weak points. This part of the paper is really what the original question was all about: "I want to include some Japanese text in my LaTeX document, how do I achieve this?". And as is to be expected, not all methods are equally applicable for given circumstances. I hope to give some insight into the various possibilities, and provide some guidance to when to use which method. The discussion of XƎTeX's capabilities will be put off until the next section.

pTeX, pteTeX, and pTeXlive

In Japan a patched version of TeX was developed called pTeX. pTeX is capable of reading and typesetting EUC-JP, (S)JIS and UTF-8 encoded files. Nowadays, inclusion of the patches is automated in the pteTeX and pTeXlive distributions[8]. Patching is required of TeX and dvips, although nowadays only patches for dvipdfmx are available. The user is required to install an adequate font for Japanese, like Cyberbit or Sazanami.

The pTeX distributions are the most complete way of typesetting Japanese with LaTeX, supporting burasage, ruby, and full vertical typesetting. The drawback is that only Japanese can be typeset. pteTeX is included in many linux distributions for the Japanese market, like VineLinux and TurboLinux, and pTeXLive is available as a small set of patches whose inclusion into a regular TeXLive distribution is automated.

Traditional LaTeX and the CJK package

Another option is to use the CJK package, available in TeXlive 2007. This package allows typesetting of Chinese, Japanese, Korean and Thai. The CJK package uses a pre-processor based on the Mule package of XEmacs (`cjk-enc.el`) to translate an input file in a given encoding (EUC-JP, Big5, BG, ...) to some canonical form, and then heavily uses translations from the input encoding to font encoding to typeset all the characters. Because the CJK package functions within legacy LaTeX, the font encoding (in NFSS) plays an important role. If only one of the CJK languages is used, the input file can be encoded in a relevant encoding (SJIS for Japanese, for instance), and pre-processed 'on the fly' using the `sjis(pdf)latex`, `bg5(pdf)latex`, ... scripts which are available in TeXlive 2007. If more than one CJK language is used in one file, the file should be encoded in UTF-8, preprocessed with (X)Emacs Mule, and then through LaTeX. Switching between the languages and the corresponding font families has to be done by the user using the corresponding commands in the input file. Because of the way the CJK package typesets the material, it is quite slow,

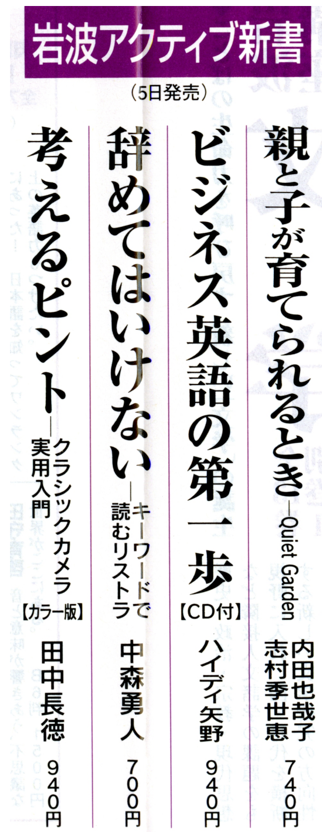


Figure 10. Example showing gothic and cursive kanji styles used together. This is from a leaflet advertising new books to be published. Notice that contrary to popular belief latin numerals can be used in vertical typesetting. Also notice that horizontal and vertical typesetting are often mixed.

and according to the manual, should only be used to typeset some CJK material in a given document, but is not very efficient to typeset large documents in the CJK languages. Personally, I consider the CJK package to be very useful (it does fully support ruby, for instance), but only relevant for the somewhat advanced user, mainly because of the selection of relevant font families (leading to all kinds of issues with .fd and .map files).

Traditional LaTEX and the UCS package

Legacy LaTEX is perfectly capable of reading unicode input files with the UCS package. However, as discussed above, being capable of reading an input stream and subsequently putting the correct character in the output are two different things. If LaTEX is to typeset a Japanese unicode input, a translation still has to be made from the unicode input into some font encoding for NFSS to access the glyphs. For example, to properly typeset some Japanese text, \usepackage[utf8x]{inputenc} and \usepackage[C42,T1]{fontenc} are required to instruct LaTEX to read UTF-8 input, translate to C42 (NFSS SJIS) for the CJK characters, and use T1 for the non-CJK characters (the legacy CJK fonts in LaTEX do usually not include the 'latin' part of the font set, so a 'T1-capable' font like latin modern is used instead). If the options for fontenc are set correctly, TEXLive 2007 will run correctly (but note that if C40 is chosen instead of C42, the required kanji fonts are not included in TEXLive).

After some experimenting with UCS, I found that it will only typeset one of the CJK languages in a given document. I also found that line breaking is not performed, because legacy LaTEX uses whitespace or hyphenation for points where a line can be broken, neither of which are present in a CJK text. Ruby is not supported, although a package could probably be written. The UCS manuals and documentation are very sparse.

Omega, Lambda and dvipdfmx

My next attempt was to use Ω (Omega, an extended version of TEX) and Λ (Lambda, 'latex for omega') to typeset CJK material. Lambda can read unicode input and typeset LTR and RTL languages. Native Lambda does not support the CJK languages, so a patch (Omega-j) is needed[9]. For the CJK languages, the resulting DVI file can be converted to PDF with dvipdfmx, but only after some tweaking (see http://oku.edu.mie-u.ac.jp/~okumura/texfaq/japanese/ for details). Technically, Lambda and Omega have the possibilities of fully supporting Japanese, including vertical typesetting and burasage, although I was not capable of reproducing the vertical typesetting example I found in one of the manuals, and setting up the burasage requires manual tweaking of OTP files (which I consider to be too much hassle). Ruby is not supported, although a package could probably be written. But perhaps the biggest drawback is that Lambda and Omega are no longer being developed.

XƎTEX and XƎLaTEX: anything goes!

The most natural way of typesetting each and any character with LaTEX would be to use unicode encoding for the input, and a unicode encoded font for the output. This is what XƎTEX[10] is capable of doing. XƎTEX is an extension of TEX, and was originally written specifically for Mac OS. It is currently available on linux (TEXlive 2007, v0.996) and Windows (MikTEX, W32TEX, v0.997). XƎLaTEX is latex for XƎTEX. The two strong points of XƎTEX are that it is fully capable of handling unicode input, and it interacts directly with (unicode encoded) OpenType fonts installed on the computer. The direct interaction with the OpenType fonts installed on the system means that if one has a CJK-capable font available, typesetting CJK in XƎLaTEX becomes very easy: simply type the text into your favorite editor, run `xelatex` and behold the result. The direct interaction with the system fonts also implies that some of the finer details of typesetting are taken out of TEX, and are instead left to the peculiarities of the font in question and the font rendering software available on the system. While this may be unacceptable to the professional typesetter, it is a major improvement for the (advanced) LaTEX user, because there is no longer a need to deal with setting up all those .tfm, .fd and .map files that make font selection in LaTEX a hassle.

To use XƎTEX to typeset any type of character, one only needs:

1. An editor capable of reading and writing UTF-8 encoded files (for most recent linux distributions, UTF-8 is the default encoding, and gedit, XEmacs and other editors support it)
2. An OpenType font which has glyphs for the particular characters you want to appear in the output.

Since XƎTEX reads unicode directly and uses unicode encoded fonts, an input file to typeset Japanese in XƎLaTEX could be as simple as the following example:

```
\documentclass[]{article}
\usepackage{fontspec}
\begin{document}
\fontspec[Mapping=tex-text]{Kozuka Mincho Pro-
VI R}
Kozuka Mincho Pro-VI R: This is English. これは日本語で
す。
\fontspec[Mapping=tex-text]{Sazanami Mincho}
Sazanami Mincho: This is English. これは日本語です。
\end{document}
```

In this example, the `fontspec` package is used. This package provides a very simple interface to select a particular OpenType font for the document. It also provides options to use special font features. Note that this example shows that it is not necessary to load any other packages to typeset the Japanese characters if this input file is saved in UTF-8 encoding.

Prerequisites to use exotic characters in XeLaTeX

To typeset a particular text using ‘exotic’ characters, the first thing that is needed is an OpenType font with glyphs for the characters you wish to obtain in the output. For Japanese, a good option is the Adobe Acrobat Reader Japanese Language pack[11]. This will install two .otf fonts, Kozuka Gothic and Kozuka Mincho (on linux systems, the files are named KozGoPro-Medium.otf and KozMinProVI-Regular.otf). Another option is the Cyberbit font[12], or the Sazanami fonts[13].

To install extra fonts on a (recent) linux system, simply put the (.otf, .ttf) file in $HOME/.fonts and run `fc-cache -fv` to update the system font database. To use a given font in your document, you need to know the name of the font to enter in the \fontspec{} command. A list of available fonts on your linux system can be obtained by running `fc-list`, which will give a list of font names and capabilities. For example, running ‘`fc-list | grep Koz`’ yields:

- Kozuka Mincho ProVI, 小 塚 明 朝 Pro-VI,Kozuka Mincho ProVI R,小塚明朝 ProVI R:style=R,Regular
- Kozuka Gothic Pro, 小 塚 ゴ シッ ク Pro,Kozuka Gothic Pro M,小塚ゴシック Pro M:style=M,Regular

while ‘`fc-list | grep Cyber`’ results in

- Bitstream Cyberbit:style=Roman

The fonts in the Acrobat Reader Japanese Language Pack are not by default stored in a systemwide font directory. I have copied the fonts to my $HOME/.fonts directory for use in this paper. Recent installations of OpenOffice provide a.o. the Baekmuk fonts (for Korean) and the AR (Arphic) family of Chinese fonts. Free OpenType fonts are available for many languages, and an internet search will reveal candidate fonts for your language of choice. Unicode supports many contemporary languages, as well as other scripts, like medieval variant alphabets, Ancient Greek Linear-B, and cuneiform etc. All these can be set with XƎLaTEX if one has an adequate font available. Especially useful fonts are Code2000 and Code2001, which have glyphs for many character sets, for example cuneiform using Code2001: . For languages not endorsed in the unicode standard, a ‘private range’ is left available in unicode for individual use; candidates for the private range are for instance Klingon ( ) and Elvish (Tengwar): , set with Code 2000). Note that line breaking etc is not (yet) properly defined for these languages, hence underfull and overfull hboxes result.

To enter Japanese or Chinese into a computer requires a special input method. This kind of software will not be decribed here in detail. For Windows, the IME (Input Method Editor) allows switching between latin and Japanese input. On linux, canna and SCIM or UIM do the same. As a subsitute, go to `http://babel.altavista.com/` and type in some words, then have it translated to the character types you like to try (Japanese, Simplified or Traditional Chinese, Korean, as long as your font is capable of displaying the characters). Copypaste the result in your editor, save as UTF-8, run `xelatex`, and enjoy your PDF output.

Specific support for Japanese typography

In XƎTEX v0.996 there is no specific support for the Japanese language. Most notably, support for ruby is lacking. With a simple patch, the existing package `nruby.sty` allows simple ruby support. Burasage is not (yet) supported. Babel support is not (yet) fully available for the CJK languages. Bibtex seems to work with UTF-8 input files with CJK characters (the reference list for this paper is made with bibtex).

For XƎTEX v0.997 (available through `svn`) the package `zhspacing.sty` is under development to bring specific support for inter-character spacing in CJK, line breaks in CJK texts, spacing between CJK and non-CJK text, and support for CJK characters as elements in mathematical equations (superscript, subscript etc).

Support for vertical typesetting

XƎTEX does not support vertical typesetting per se, but it is still possible to typeset material vertically. To achieve this, one has to have an OpenType font with the 'vrt2' property. Glyphs in such a font can be rotated. If one puts some CJK text with rotated glyphs inside a \rotatebox, vertical typesetting is obtained. It should be noted that vertical typesetting is not very stable at this moment. Several manuals give examples of vertical typesetting (e.g. the fontspec and zhspacing documentation), but Your Mileage May Vary depending on your system. In my case, Texlive 2007 with XƎTEX v0.996 yielded incorrect typesetting in vertical mode.

I was advised to upgrade to v0.997, because of better support for vertical typesetting. After some trial and error, I was able to typeset some material vertically, as illustrated in figure 11, which was set with the following source:

```
\begin{figure}
\begin{center}
\rotatebox{-90}{
\begin{minipage}{0.35\textwidth}

\fontspec[Script=CJK,RawFeature=vertical]{Kozuka Mincho Pro-VI}
三一 \ruby{坂上是則}{さかのうえのこれのり}
\\[2\baselineskip]

\fontspec[Script=CJK,RawFeature=vertical]
{YOzFontKA04}
朝ぼらけ

\\[0.5\baselineskip]
\ruby{有明}{ありあけ}の月と　見るまでに

\\[0.5\baselineskip]
\ruby{吉野}{よしの}の里に　ふれる\ruby{白雪}{しらゆき}

\end{minipage}
}
\end{center}
\end{figure}
```

To obtain the result of figure 11, the text is set in a minipage with rotated glyphs. The entire minipage is put inside a \rotatebox. The ruby is provided by the `nruby` package. The line spacing is not optimal with ruby, so some extra space is added manually here. Without the minipage, incorrect line breaking would occur. XƎTEX has some rudimentary support built in for line breaking in CJK languages using the \XeTeXlinebreaklocale="ja" command, but to get an acceptable result the package `zhspacing` should be used. Also, the combination of a rotatebox and minipage implies that sectioning commands are not available, and if more than one page of text has to be set, a simple overfull box will result, instead of the text being set on the next page. At the time of writing, a discussion was going on as to the best solution to this problem.

三一 坂上是則（さかのうえのこれのり）

朝ぼらけ

有明（ありあけ）の月と　見るまでに

吉野（よしの）の里に　ふれる白雪（しらゆき）

Figure 11. From the `Hyakunin Isshu', the `Hundred poems by the famous poets' [4], Poem 31, by Sakanoue no Korenori: *At the firsrt light, it is not really the late moon shining, that casts its light on Yoshino, but the whiteness of the snow.* Set with YOzFontKA04.

Some more examples of XƎLaTEX capabilities

Here follows some material in Hindi, Chinese, Vietnamese, and Japanese.

Hindi, using Raghindi font (raghu.ttf)

अमरीकी में मंदी की आशंका को देखते हुए अमरीकी केंद्रीय बैंक फेडरल रिज़र्व ने ब्याज दरों में आधे फीसदी की कटौती की है. फेडरल रिज़र्व ने दो दिनों की बैठक के बाद ब्याज दरों को 3.5 फीसदी से घटाकर तीन फीसदी कर दिया है. पिछले सप्ताह ही केंद्रीय बैंक ने दुनिया के शेयर बाज़ारों को संभालने के लिए ब्याज दरों में कटौती की घोषणा की थी. माना जा रहा है कि इस कदम से अमरीकी अर्थव्यवस्था को मंदी की संभावना से उबरने में

मदद मिलेगी. इसके पहले 2007 के अंतिम तीन महीनों में अमरीकी अर्थव्यवस्था के आँकड़े जारी किए गए थे जिसमें अर्थव्यवस्था की मंदी के संकेत मिले थे. अक्तूबर और दिसंबर के बीच अमरीकी अर्थव्यवस्था की विकास की दर में 0.6 फ़ीसदी की गिरावट आई. विशेष क़दम विशेषज्ञों का कहना है कि फ़ेडरल रिज़र्व ने ये विशेष क़दम इसलिए उठाया है ताकि जल्द ही फिर कटौती की कोई ज़रूरत न रहे.

Chinese, AR PL ShanHeiSun Uni font

美国联邦储备局9天内第二次削减利率，试图避免美国经济进入衰退。美国央行经过两天的会议后将利率从3.5%降到3% 。上周美国联邦储备局削减利率，那次降息成为25年以来削减幅度最大的一次。美联邦储备局试图通过大幅度降息平息全球股市震荡。美国联邦储备局希望降息能够减少信贷紧缩和房屋市场衰退对美国经济的冲击。美国央行的联邦公开市场委员会(FOMC)表示，金融市场仍然受到很大压力，商业和家庭信贷会进一步紧缩 ，另外最新的数据显示房屋市场进一步收缩，劳务市场出现疲软。股市上扬降息半个百分点已经超过了一些经济学家的预测，所以降息会鼓舞金融市场。

Vietnamese, Bitstream Cyberbit font

Ngân Hàng Trung Ương Hoa Kỳ đã cắt giảm lãi xuất chính nửa phần trăm để xuống còn là 3% trong một nỗ lực nhằm vực dậy nền kinh tế của nước Mỹ mà hiện đang rơi vào một cơn đình đốn. Tổng thống Bush nói rằng "nền kinh tế của Hoa Kỳ đang phải giáp mặt với các khó khăn ngắn hạn, và ông vững tin vào triển vọng xán lạn về lâu về dài". Đây là lần thứ nhì trong vòng tám ngày, Ngân Hàng Trung Ương đã cắt lãi xuất và cũng là lần đầu tiên từ 25 năm nay, Ngân Hàng mới áp dụng một biện pháp nhanh gọn đến như thế.

Korean, Baekmuk Gulim font

‘단군 이래 최대 소송’으로 일컬어지는 삼성자동차 채권 환수 소송에서 삼성측이 약 3조1500억원을 물어내라는 판결이 내려졌다.서울중앙지법 민사합의21부(김재복 부장판사)는 31일 삼성자동차 채권단인 서울보증보험 등 14개 금융기관이 이건희 회장과 삼성그룹의 28개 계열사를 상대로 낸 약 5조원의 약정금 청구 소송에서 원고 일부 승소 판결을 내렸다.재 판부는 “채권단의 주장에 상당부분 일리가 있다”며 “삼성측은 채권단과 약정한 2조 4500억원을 모두 갚아야 한다”고 판결했다. 재판부는 “채권단이 맡고 있는 주식을 삼성측이 팔아서 1조 6338억여원까지 만들고 나머지 부족한 부분은 이건희 회장이 개인적으로 갖고 있는 삼성생명 주식을 팔아서 2조 4500억원을 채우라”고 판결했다. 재판부는 그러나 연체이자는 1조 6338억여원에 해당하는 이자만 지급하면 된다고 판결했다. 이자가 2001년부터 계산되기 때문에 약 7000억원에 달한다. 따라서 삼성은 원금과 이자를 합쳐 약 3조 1500억원을 채권단에 지급해야 할 것으로 보인다.

Japanese, with vertical typesetting

歴史的寒波、影響は1億人超に温首相も動く　２００８年０１月３０日１９時５５分　中国の中南部を襲った歴史的な寒波で中国政府は３０日、影響を被った人が1億人を超えたことを明らかにした。旧正月（2月7日=春節）の帰省ラッシュを直撃した災害に政府は危機感を強め、温家宝（ウェン・チアパオ）首相を湖南、広東両省に急派した。石炭が運べず、火力発電所が広範囲で停止。日系企業にも影響が出ている。胡錦濤（フー・チンタオ）主席は２９日、輸送と発電の復旧に全力を挙げるよう指示。温首相は同日朝、被害が大きい湖南省の長沙駅で帰省客に「復旧にそれほど時間はかからない。家で年越しできる」と拡声機で励ました。　３０日は広州市の農産物市場で販売員に「値上がりしていませんか」と質問した。白菜は数日で7割上昇、羊肉も2日で5割値上がりしている。　その直前に訪ねた広州駅周辺では８０万人が足止め。当局は出稼ぎ労働者２６００万人に、「帰省せず、広東で年越しを」と呼び掛ける一方、「足止め客に宿舎と食事を確保する」と強調。上海市、福建省なども同様の状況だ。　上海では３０日まで6日間連続で雪が降った。空の便は国際、国内線で計１０００便以上が欠航、遅延。長距離列車も連日５０~６０本が運休か遅延している。主な高速道も閉鎖、上海は一時「陸の孤島」状態になった。　上海駅前では２９日午後、暖かい構内に入ろうとする客と警官がもみ合いに。店員、孫海蓉さん（１６）は「上海のアパートを引き払ったので列車に乗れないと泊まる場所もない」と話した。　日系企業が集中する江蘇省無錫市などでは、発電用の石炭不足による電力供給カットで工場の生産に支障が出た。広州でも、部品が届かず、日系乗用車メーカー３社に影響が出ている。

Concluding remarks

There are several ways of incorporating CJK material in a LaTEX document. The most complete support for typesetting Japanese, including all bells and whistles for that language, are found in the pTEXLive distribution, which specifically supports Japanese. To include a 'small amount' of CJK material in a LaTEX document, there is the CJK package to do all three CJK languages, with all bells and whistles. The UCS package provides a theoretical possibility of incorporating CJK by enabling LaTEX to read unicode input, but restrictions apply (only one CJK language per document), and without extra definitions of line breaking etc., application is limited and troublesome. One could use Lambda and Omega, but the versions supplied in TEXLive 2007 are not CJK capable and need to be patched. Most importantly, Ω is no longer being developed further, making this an unattractive option.

The most user-friendly option I found is XƎTEX and the associated macro-package XƎLaTEX in combination with fontspec. As long as one has an OpenType font available on the system with the appropriate glyphs, character sets can be used indiscriminately in one document, as long as glyphs are available in the font. The current version of XƎTEX on TEXLive 2007 is v0.996, and not all bells and whistles work equally well on all systems, as I found out. CJK typesetting is improved in v0.997, although some issues remain for vertical typesetting.

References

[1] K.G. Henshall. A guide to remembering Japanese characters. Charles E. Tuttle, Co., thirteenth edition, 1998.

[2] Kodansha's essential kanji dictionary. Kodansha international, 1991, 2002.

[3] The compact Nelson Japanese - English character dictionary. Charles E. Tuttle, Co., 1999, 2004.

[4] 出井光哉. プラス英訳百人一首. 風塵社, 1991.

Fonts used in this document

Various fonts are used throughout this article. All these fonts are freely available on the internet. To enable the reader to experiment, these are the font names and where they can be found. I downloaded all these fonts and installed them in $HOME/.fonts, and used 'fc-cache -fv' to make them available in XeLaTeX.

- The main font throughout is 'Kozuka Mincho Pro-VI R' (Acrobat Japanese Language Package)
- The 'YOzFont' fonts are available at http://yozvox.web.infoseek.co.jp/446F776E6C6F6164.html
- The 'Hakusyu' fonts are available at http://www.linkclub.or.jp/~ma3ki/lc-hp/font.html
- Code2000 and Code2001 are available at http://www.code2000.net
- The 'Raghindi' font used for Hindi can be found at http://tdil.mit.gov.in/download/Raghu.htm
- For Korean, the Baekmuk Gulim font is part of OpenOffice; for Chinese, the AR PL ShanHeiSun Uni is part of OpenOffice.

For the reader wanting to experiment with the possibilities of XƎLaTEXand CJK, some simple input files are available on the NTG website (go to http://www.ntg.nl/maps/36/xetex/).

Footnotes

1. Chinese, Japanese, Korean
2. Named after the man'yōshū, "Collection of Ten Thousand Leaves", a collection of poems written between 600 and 759 AD in standardized, phonetic kanji
3. http://www.taishukan.co.jp/kanji/daikanwa.html
4. I don't know whether this applies to XP / Vista
5. http://www.unicode.org/
6. See for an example http:// www.lukew.com/ ff/ entry.asp?111
7. Ruby is also known as furigana, 'guiding kana', and kanbun, 'kana letters'
8. http://www.nn.iij4u.or.jp/tutimura/
9. See http:// zoonek.free.fr/ LaTeX/ Omega-Japanese/ doc.html for an example
10. http://scripts.sil.org/xetex
11. http://www.adobe.com/products/acrobat/acrrasianfontpack.html
12. http://http.netscape.com.edgesuite.net/pub/communicator/extras/fonts/windows/
13. http://sourceforge.jp/projects/efont/files/

Wilfred van Rooijen
wvanrooijen@yahoo.com