The Asian Journal of TEX, Volume 2, No. 1, April 2008
Article revision 2008/4/1

Development of Japanese TEX Environment

Nobuyuki Tsuchimura 土村 展之
The University of Tokyo
Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
tutimura@mist.i.u-tokyo.ac.jp

Yusuke Kuroki 黒木 裕介
kuroky@misojiro.t.u-tokyo.ac.jp

KEYWORDS source distribution, pTEX, UTF-8 encoding, Cygwin.

ABSTRACT This paper describes a Japanese TEX distribution 'ptetex' which facilitates the installation process. We have two projects: one is to develop a UNIX source distribution, and another is to provide a binary package for Cygwin.

We will also describe a new library, which helps Japanese TEX (pTEX), to handle character encoding operations even with UTF-8 encoding. As a result, it will be a breakthrough in typesetting UTF-8 encoded texts including CJK characters by pTEX.

1 Introduction

Many projects for multi-lingual extensions of TEX have had its ups and downs, but in Japan, pTEX [1] has been a de facto standard for Japanese extension of TEX over ten years. pTEX enables high-quality Japanese typesetting, horizontally and/or vertically, even for publishing purposes.

As each of the pTEX-related tools has been developed individually, the current situation is that files required for pTEX-based system are scattered over the Internet. Considering TEX-related tools alone, the required files are tangled and the number of them is large. In order to make the installation process easier, there are some distributions available. For example, in alphabet-using countries, teTEX [5] and MiKTEX [13] are released for UNIX source and Windows binary respectively, and recently TEX Live for both platforms; in Japan, W32TEX [6] for Windows binary. However, there have been no source distributions of Japanese TEX environment for UNIX.

In view of this situation, we have developed a Japanese TEX distribution for UNIX source, named 'ptetex.' It is gradually becoming common: for instance, it has been adopted as TEX system by some domestic Linux OSes and has been introduced in a book [9]. We consider that one of the reasons why ptetex has become so popular is that it has clear design goals, and that the goals are disclosed. Especially in Japan there are not so many TEX projects concerned with software development, even if they involve experts in typesetting and/or programming.

A distribution also plays a role as a base for discussion on what kind of improvement should be made. In fact, some projects have been launched under our environment. One is upTEX [14]. It is a further extension of pTEX to treat UTF-8 encoded files. However, pTEX is already so complicated that adding another extension might result in more complication. Another is ours. We are developing a library to separate character encoding operations from pTEX, which helps to simplify the original source. The library has a simple extension to handle UTF-8 encoded files so that they can be typeset by our products.

This paper is organized as follows. In the next section, we will give an introduction to TEX distribution. In Section 3, we will describe a Japanese TEX distribution. Section 4 introduces another project to distribute a binary package for Cygwin. Section 5 will discuss our attempt to create a library of character encoding operations.

2 TEX Distributions

TEX environment is application software consisting of such elements as TEX compiler, LATEX macros, some other extending macros, fonts, DVI drivers, etc. The environment can be constructed using free software. TEX has similarities in its construction to Linux, which also consists of free software. (See Table 1.)

It should be noted that there exist distributions for both TEX and Linux. A distribution is a cluster of software developed by many developers. Some of them may not be familiar with the whole current environment, and some may have stopped development. In order for various software and macros developed following various principles to work together well, the director's role is essential.

Many people used to think it clever to continue to use the environment they constructed, not updating it to a newer version, even if some bugs were found. This is because it required a lot of labor to construct an environment (due to low transmission rate, etc.) and because they thought the reproducibility of output was most important. However, in these days, it has become commonplace to use the newest version of OS and application software. Newer versions, of course, should preserve the quality and easy method of restoring the previous environment should be provided. If the conditions are met, distributions will encourage people to use the newest version.

Among TEX distributions which supply sources for UNIX, teTEX [5] is the most influential. TEX compilers and tools are fully compiled, and it includes necessary

TABLE 1. Comparison of TEX to Linux.

	TEX	Linux
Author of core	Donald E. Knuth	Linus Torvalds
Core	TEX compiler	Linux kernel
Compiler for core	Web2C (+ gcc)	gcc
Essential library	Kpathsea, LATEX	glibc
	macro, font	library
	BIBTEX, DVI-ware	application software
Directory structure	TDS (TEX Directory Structure)	FHS (Filesystem Hierarchy Standard)
Users group	TUG (TEX Users Group)	LUG (Linux Users Group)
Distribution	teTEX, MiKTEX, TEX Live, ...	Fedora Core, Debian, ...

and sufficient macros and fonts. It consists of free software and is adopted as their TEX environment by many Linux OSes, Mac OS, and Cygwin. It could have been a de facto standard TEX distribution. Yet, unfortunately, it was announced that the developer would no longer release a new version, the latest being version 3.0.

3 Development of ptetex

In Japan, on the other hand, the situation was different. There were no distributions and it was difficult to create a well-customized Japanese TEX environment. To solve the problem, the first author, Nobuyuki TSUCHIMURA, has conducted a project of making a Japanese TEX distribution [15, 16]. In this section, we will first point out what the problem was, then introduce the aim and concept of the project, and lastly discuss some technical matters.

3.1 Japanese TEX Environment

For Japanese typeset, TEX is required to have an ability to typeset Japanese characters (up to 6000+ characters) horizontally and/or vertically (even in one page). In addition, as TEX, in a narrow definition, cannot handle multi-byte characters directly, some extension is necessary.

pTEX is a solution for this requirement, which is a 16-bit extension of TEX developed by ASCII Corporation, a Japanese publisher. pTEX, whose 'p' stands for publishing, has been a de facto standard extension of TEX in Japan for over ten years. pTEX is released in the form of a patch (difference) file for teTEX. Therefore, to install pTEX, first you have to obtain teTEX and then apply patches to it. Still, this is not sufficient for Japanese TEX environment.

pTEX's extension modifies TEX compiler and DVI format; peripheral tools also need to be modified. Some tools such as dvips have extensions developed by ASCII Corp., and other tools such as dvipdfm and xdvi by individual developers [2, 17]. As a result, patches are scattered on different sites and some settings (e.g., font configuration) are quite different from each other. The installation process is not automated because of the change of the directory structure, and users need to reinterpret the manuals according to their environment. The amount of labor and knowledge necessary for constructing a Japanese TEX environment is huge.

In spite of these conditions, there have been no distributions for Japanese environment. Even a TEX environment bundled with the OS is seldom coordinated sufficiently.

3.2 Aim of Development

We realized that we needed a Japanese TEX distribution corresponding to teTEX and started to develop ptetex from February 2004.

The aim of development consists of the following three goals. (1) A short-term goal: to enable ordinary users to build the Japanese TEX environment easily. This goal was fulfilled since we built an archive of required patches to teTEX and customized font settings for several tools. (2) A medium-term goal: to position ptetex as a standard TEX distribution which replaces 'teTEX + pTEX + many patches.' We are in this phase. (3) A long-term goal: to have upstream projects adopt our patches for Japanese characters. We will consider the goal accomplished when there is no reason to continue the ptetex project.

3.3 Guidelines for Development

In order to achieve the aim, we also have some guidelines for development. We follow the manners of open source software development by Raymond [11], and teTEX as a model. Particularly, we keep in mind the following two lessons [11]: "Release early. Release often. And listen to your customers", and "When you lose interest in a program, your last duty to it is to hand it off to a competent successor." We will enumerate the development policies below.

Concentrate on Japanese Materials We develop ptetex as patches to teTEX, i.e., we focus on Japanese materials, and avoid repairing teTEX. We collect and archive fonts and patches within a relatively small size (cf. Table 2). Also, we turn our efforts to source distribution, though we have released some packages as an example. We try to keep ptetex simple so that it is easy for binary packagers to access.

Keep Previous Versions Available We keep previous versions available on our site, to trace defects in our product; it is one of the roles which distribution should play as we have seen in Section 2. Some of the ingredients of the product are not version controlled appropriately, and therefore the older versions may not be available any more. ptetex also plays a role to keep their older versions.

Release Update and Security Fix Frequently Because teTEX has not made minor improvements and has not covered security holes after official release, we follow components' (official) updates as soon as possible. We also try to respond to users' bug reports and comments frequently, and to remove security bugs. Still, we can just follow Linux OSes' bug fix reports concerning security fix.

Prepare English Documents Some people, even though they do not understand Japanese, may need to have a Japanese TEX environment, such as producers of text-editors or distributors of international Linux trying to handle Japanese TEX. ptetex includes an English manual for installation. This is a necessary preparation for achieving our long-term goal.

Independent of Skills and Systems Early versions of ptetex often failed to compile, depending on the system. We have made use of information given by the users, and prepared some devices such as environmental checker. This improvement not only helps to reduce the amount of work, but also enables users to construct the same condition without dependence on special (UNIX) skills and environment on the user's side.

3.4 Technical Matters

We will show some technical matters of ptetex in this subsection.

Structure A distribution archive includes Japanese patches and a shell-script to expand them. The archive was originally associated with teTEX to build a system. Conditions required for building it is the same as those for teTEX. Likewise, Ghostscript extra should be installed.

License Our new scripts are distributed under the modified BSD license as pTEX. The licenses of included files are defined by each developer. Our products as a whole can be redistributed; however, a small number of materials are not allowed to be modified, for example, CMaps by Adobe.

Installation Process The installation process of ptetex is as follows: (1) download three files, shown in Table 2, into the same directory, (2) execute commands as Figure 1, then (3) modify the environmental variable `PATH`.

TABLE 2. Required files for installation.

Name	Size
`tetex-src-3.0.tar.gz`	13 MB
`tetex-texmf-3.0po.tar.gz`	88 MB
`ptetex3-`⟨*version number*⟩`.tar.gz`	5 MB

(Extract file)
```
$ gzip -cd ptetex3-⟨version number⟩.tar.gz | tar xvf -
$ cd ptetex3-⟨version number⟩
```
(Compile and test by user privilege)
```
$ make
```
(Install by root privilege)
```
$ su
# make install
```

FIGURE 1. Executing commands for installation.

Achievement Recently, ptetex has been increasingly used in Japan. There have been around ten thousand direct accesses for download in the last one-year period. Some domestic Linux OSes (e.g., Plamo, Momonga, Vine) have adopted ptetex. Moreover, some volunteers are distributing compiled binaries for Mac OS X. We also have another project to provide a binary package for Cygwin. We will describe the project in Section 4.

4 Binary Packaging for Cygwin

The second author, Yusuke KUROKI, has another project for providing a binary package of ptetex for Cygwin [7, 8]. In this section, we will describe the project briefly.

Cygwin [4] is a Linux-like environment for Windows. It provides Linux-compatible system calls and a number of Linux tools. If Cygwin repository does not have software which a user wants to use, he/she needs to compile it. However, not all application software can be compiled successfully because Cygwin is not a complete Linux system.

Unfortunately, early versions of ptetex failed to compile. At first, we tried to compile ptetex as it was so that we could feedback the cause of failure to source development. After several trials, we succeeded in compiling it in Cygwin. We believe this is a big advance for it has turned out that ptetex can be compiled even on a 'poor' Linux system. We did not only check whether it can be compiled, but also we have managed to provide a binary package of ptetex. In the remainder of this section, we will introduce three major advantages of this project.

Alternative Binary for Windows Broadly speaking, our binary package can be an alternative for W32TEX, a Japanese TEX binary distribution for (native) Windows. From a risk management perspective, it is important to have an alternative, because W32TEX had been the only choice.

Time Saving On Linux, it takes just around ten minutes to compile and install ptetex in a PC (e.g., Pentium 4 [3.0 GHz] and 1 GB RAM). However, on Cygwin, we need more than two hours to do that in the same PC due to the overhead of Cygwin and the slow file system of Windows. By using our binary product (through Cygwin official net-installer), the process may finish in around 20 minutes, depending on transmission rate and machine speed.

Attention to Ghostscript Once we fix a system, we should consider Ghostscript, though it is outside the scope of the ptetex source distribution. Cygwin repository has a compiled package of Ghostscript, but it cannot handle Japanese texts, especially in vertical direction. So, we also prepare a well-customized compiled package of Ghostscript by using gs-cjk products [18].

5 Refinement of pTEX

After we had concentrated on collecting existing materials and adjusting them for a long time, we started to find out how to refine pTEX. We believe it will provide a new way to typeset CJK characters, and a ground for discussion as to what is necessary for CJK typesetting.

5.1 A New Library: ptexenc

For the Japanese language, three different character encodings—EUC-JP, ISO-2022-JP, and Shift_JIS (for short, EUC, JIS, and SJIS, respectively)—have been used, depending on OS; nowadays UTF-8 encoding is becoming popular. pTEX originally had three executable files for each of three traditional character encodings. Some improvements were made so that one executable file could support three encodings, but they were not sufficient. To solve this issue, we have created a new library, named 'ptexenc,' which makes the encoding functions clear. We will illustrate what we did below.

First, we made an extension with which encodings of internal format, file I/O, and terminal output can be specified individually. (In the original pTEX, all the encodings are interlocked.)

Second, we separated an encoding conversion routine from pTEX and organized it into a library. This library is also useful for Japanese tools such as jBIBTEX, mendex (customized version of makeindex for Japanese), etc.

Third, we gave UTF-8 support for ptexenc. It was not so difficult to accomplish the last step, but it will have a great effect because more OSes (e.g., Mac OS X and Fedora Core) have used UTF-8 as the default encoding to realize a multilingual setting. Many Japanese users have waited for pTEX to handle UTF-8.

It should be noted that, even if an input file is encoded in UTF-8, pTEX with ptexenc handles it in EUC-JP or Shift_JIS encodings internally. The characters which cannot be converted to EUC or SJIS are translated to `^^ab` format, and are typeset as they are typeset by TEX. Therefore, the characters outside the EUC or SJIS range can be transformed to ASCII transcription by using inputenc macro with utf8 or utf8x option and appropriate dfu lists. It is because pTEX has virtual fonts to output Chinese and Korean characters [12] that the library open a way to typeset UTF-8 encoded texts, including CJK characters, directly using pTEX + ptexenc. In the next subsection, we will show an example of what pTEX + ptexenc can output.

5.2 What Can We Typeset?

We would like to show an example (Figure 2) which includes Japanese and Korean characters in one document. The example sentences are taken from the Omega-CJK Project [3].

The corresponding source to Figure 2 is shown in Figure 3. In line 1, `jsarticle` is the Japanese standard document class by Okumura [10]. Lines 2–7 are for ASCII transcription. Especially, lines 4–7 are dfu lists. The dfu lists are introduced by the inputenc macro; they declare the relations between Unicode point and TEX-solvable explanation. We enumerated the dfu lists of hangul as `otf-hangul.dfu` (Figure 4) in line 4. Lines 5–7 are temporary dfu lists for Sinographs outside the EUC or SJIS range. To obtain a vertical writing version, uncomment line 10 (and, precisely speaking, change ASCII punctuation marks to multi-byte ones). Lines 11–12 are Japanese texts and lines 14–15 are Korean.

The commands we used are shown in Figure 5. The option `-kanji=utf8` is the extension of ptexenc. Dvipdfmx [2] converts DVI to PDF.

This library can be used to typeset Japanese-based multi-lingual articles since pTEX is a high-quality typesetting system for the Japanese language. Besides, it will be a useful starting point of discussion as to what rules are required for each language. For example, Japanese typesetting prefers to have a slight space between an ASCII character and a Japanese character, but Korean typesetting inhibits such spacing because a space should be inserted between words in Korean orthography. We realize the rule by pTEX's primitive in line 13 of Figure 3.

TEX はスタンフォード大学のクヌース教授によって開発された組版システムであり、組版の美しさと強力なマクロ機能を特徴としている。

TEX은 스탠포드 大學의 크누스 教授에 의해 開發된 組版 시스템으로, 組版의 美와 强力한 매크로 機能이 特徵이다.

(a) Horizontal writing.

TEX はスタンフォード大学のクヌース教授によって開発された組版システムであり、組版の美しさと強力なマクロ機能を特徴としている。

TEX은 스탠포드 大學의 크누스 教授에 의해 開發된 組版 시스템으로、組版의 美와 强力한 매크로 機能이 特徵이다。

(b) Vertical writing. (Note: we changed punctuation marks in the Korean text manually.)

FIGURE 2. Example of a Japanese-Korean bilingual text. (The text says *"TEX is a typesetting system developed by Prof. Knuth from Stanford University; it features beauty of typeset and powerful macro functions."*)

```
\documentclass{jsarticle}
\usepackage[utf8]{inputenc}
\usepackage[multi]{otf}
\input{otf-hangul.dfu}% dfu lists of hangul (our original)
\DeclareUnicodeCharacter{5F3A}{\UTF{5F3A}}% 强
\DeclareUnicodeCharacter{654E}{\UTF{654E}}% 敎
\DeclareUnicodeCharacter{5FB5}{\UTF{5FB5}}% 徵
\begin{document}
\fbox{\vbox{\hsize=21zw
%\tate\adjustbaseline% for vertical writing
{\TeX}はスタンフォード大学のクヌース教授によって開発された組版システムであり、
組版の美しさと強力なマクロ機能を特徴としている。\par\bigskip
{\noautoxspacing% to inhibit a slight space between ASCII and Korean
{\TeX}은 스탠포드 大學의 크누스 敎授에 의해 開發된 組版 시스템으로,
組版의 美와 强力한 매크로 機能이 特徵이다.\par
}}}
\end{document}
```

FIGURE 3. The corresponding source to Figure 2 (`example.tex`).

```
\DeclareUnicodeCharacter{3130}{\UTFK{3130}}
⋮
\DeclareUnicodeCharacter{318F}{\UTFK{318F}}
\DeclareUnicodeCharacter{AC00}{\UTFK{AC00}}
⋮
\DeclareUnicodeCharacter{D7AF}{\UTFK{D7AF}}
```

FIGURE 4. The dfu lists of hangul (`otf-hangul.dfu`).

```
$ platex -kanji=utf8 example.tex
$ dvipdfmx example.dvi
```

FIGURE 5. The commands to typeset Figure 2.

6 Concluding Remarks

So far, we have discussed the Japanese TEX environment and our distribution. Today, the requirements for a typeset system that deals with CJK characters are just beginning to be discussed among relevant parties in Japan and Korea. It would be better if TEX developers around the globe make developments with Sinograph cultures in mind, compared to the present situation in which Japanese or other Chinese character-using cultures design their specific environment by themselves. To support this, we hope to write additional papers, preferably in English, which describes the requirements for a Sinographic typeset system.

Acknowledgments

We would like to thank Ken NAKANO and Hideaki TOGASHI at ASCII Corporation for their development of pTEX and their management of pTEX developers' mailing list. We would like to acknowledge Haruhiko OKUMURA, who always takes care of us, Akira KAKUTO who pays attention to tools for UNIX, and Takuji TANAKA and ZR who taught us a lot about Unicode and TEX families. Special thanks go to Myung-Chul CHO, Kangsoo KIM, Haruhiko OKUMURA, and Takeo TATSUMI from discussion with whom we were able to obtain the idea of Section 5.

We would like to express our gratitude here to very many TEX developers for their continuing support and for their tireless efforts.

References

1. ASCII Corporation, アスキー日本語 TEX (pTEX). http://www.ascii.co.jp/pb/ptex/
2. J.-H. CHO and S. HIRATA, The DVIPDFMx project. http://project.ktug.or.kr/dvipdfmx/
3. J.-H. CHO and and H. OKUMURA, *Typesetting CJK languages with Omega*, TEX, XML, and Digital Typography, Lecture Notes in Computer Science, vol. 3130, Springer, 2004, pp. 139–148.

4. Cygwin information and installation. http://cygwin.com/
5. T. ESSER, The teTEX homepage. http://www.tug.org/tetex/
6. A. KAKUTO, W32TEX. http://www.fsci.fuk.kindai.ac.jp/~kakuto/win32-ptex/
7. Y. KUROKI, Japanese TEX environment for Cygwin. http://www.misojiro.t.u-tokyo.ac.jp/~kuroky/tex/index.en.html
8. Y. KUROKI, *Japanese TEX environment for Cygwin*, Computer Software **25** (2008), 39–46. (In Japanese, except abstract.) http://www.jstage.jst.go.jp/article/jssst/25/2/25_2_39/_article
9. 奥村晴彦『[改訂第 4 版] LATEX2ε 美文書作成入門』(2007, 技術評論社)
10. 奥村晴彦, pLATEX2ε 新ドキュメントクラス. http://oku.edu.mie-u.ac.jp/~okumura/jsclasses/
11. E. S. RAYMOND, *The cathedral and the bazaar*, 2000. http://www.catb.org/~esr/writings/cathedral-bazaar/ (Japanese translation by H. YAMAGATA is available at http://cruel.org/freeware/cathedral.html.)
12. 齋藤修三郎, LATEX2ε 的. http://psitau.at.infoseek.co.jp/
13. C. SCHENK, MiKTEX project page. http://miktex.org/
14. T. TANAKA, upTEX, upLATEX—unicode version of pTEX, pLATEX. http://homepage3.nifty.com/ttk/comp/tex/uptex_en.html
15. N. TSUCHIMURA, ptetex—Japanese patch collection for teTEX. http://tutimura.ath.cx/~nob/tex/ptetex.en.html
16. N. TSUCHIMURA, *Development of a Japanese TEX distribution 'ptetex3'*, Computer Software **24** (2007), 40–50. (In Japanese, except abstract.) http://www.jstage.jst.go.jp/article/jssst/24/4/24_4_40/_article
17. M. TSUCHIYA, N. TSUCHIMURA, and T. UCHIYAMA, Xdvik-jp cleanup project. http://xdvi.sourceforge.jp/index.en.html
18. T. YAMADA, Data storage for gs-cjk project. http://www.aihara.co.jp/~taiji/gyve/