The Asian Journal of TEX, Volume 2, No. 1, April 2008
Article revision 2008/4/21

Handling Two-Byte Characters with DVIasm

Jin-Hwan Cho
Department of Mathematics, The University of Suwon
Republic of Korea
chofchof@ktug.or.kr

KEYWORDS DVIasm, TEX utility, DVI editing, Unicode, Chinese, Japanese, Korean characters, subfont scheme, Omega, pTEX, CJK-LATEX, koTEX.

ABSTRACT This paper is devoted to the second step of developing a new DVI editing utility, called DVIasm. DVIasm disassembles a DVI file into a human-readable text file the contents of which are easy to modify, and assembles the output back to the binary DVI format. The main theme of this paper is two-byte characters such as Chinese, Japanese, Korean, and Unicode characters. We discuss how DVIasm handles two-byte characters that are supposed to be typeset by 16-bit TEX extensions such as Omega and Japanese pTEX, or LATEX packages with subfont scheme such as CJK-LATEX and Korean koTEX.

1 Introduction

DVIasm is a TEX utility program that is designed for editing DeVice-Independent (DVI) files[1] directly. It consists of a single Python program, `dviasm.py`, in a human readable text format and it runs on any platform in which Python [13] is installed. All revisions of DVIasm can be downloaded at [3] that is controlled by the popular version control system, Subversion [14].

It is worth while to mention the other two DVI utilities, both of which were developed with a similar purpose. The first utility, `dvitype` [8], was written by Donald E. Knuth in 1982. It supported only a one-way conversion from the DVI format to a text format, and by this reason its practical application was restricted to validating and inspecting the contents of DVI files. On the other hand, `dt2dv` and `dv2dt` [11] written by Geoffrey Tobin supported a two-way conversion between the DVI format and the DVI Text Language (DTL) format. They also had a restricted application because DTL was too strict a format to be used by human for the purpose of editing DVI files.[2]

DVIasm also provides a two-way conversion and its text format is much simpler and more convenient than the DTL format by the following reasons.

1. The DVI file format was designed by David R. Fuchs in 1979. It is intended to be both compact and easily interpreted by a *machine* [8, §14].
2. For example, the four DTL commands from '`r1`' to '`r4`' are used to move the current reference point to the right. The only difference is that each command allows a different range of the right move. Therefore, users have to choose an appropriate command each time according to the amount of the right move. DVIasm, on the other hand, has only one command '`right`' for this purpose regardless of the amount of the right move.

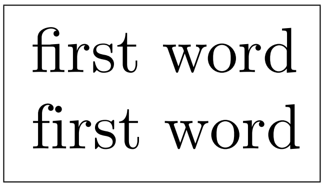
– DTL requires the full DVI structure consisting of the header and the body, but DVIasm does not require the header that will be generated automatically with the default values and the information collected from the body.
– Each page in DVIasm consists of simplified DVI commands [1, p. 213], but full DVI commands must be used in DTL. For instance, the four DVI commands from '`fnt1`' to '`fnt4`' have the same role changing the current font. The corresponding command in DVIasm is just '`fnt`'.
– DVIasm supports many popular units, `pt` (default), `bp`, `mm`, `cm`, `in`, as well as `sp` that is the only unit supported in DTL.

Moreover, the default encoding of DVIasm is UTF-8[3] in order to support two-byte characters in the input and the output. The output of both `dvitype` and the DTL package is encoded in ASCII.

DVIasm was first introduced in 2007 with a few examples handling various DVI specials [1]. After that the author noticed that DVIasm is not a convenient tool for people using Latin languages. The reason was in the two typographic features, kerning and ligature. We show in Code 1 two lines, the upper one with kerning and ligature and the lower one without them. One can find a kerning between 'w' and 'o' (`right: -0.277786pt`) and a ligature 'fi' (`\x0c`). Note that DVIasm and the DTL package do not

```
\nopagenumbers
first word \par
  % with kern/lig
f{}irst w{}ord
  % without kern/lig
\bye
```

```
fnt: cmr10 at 10pt
set: '\x0crst'
right: 3.333328pt
set: 'w'
right: -0.277786pt
set: 'ord'
```


first word
first word

CODE 1. A kerning between 'w' and 'o' and a ligature 'fi' (`\x0c`). The left code is the TEX source, and the right code is the output by DVIasm that corresponds to the upper line with kerning and ligature.

read font information from TEX Font Metric (TFM) files which contain the information of kerning and ligature. Then, do we have to insert kernings and ligatures manually? That is almost impossible! We have no choice but to generate a DVI output with the help of TEX.

However, the situation would not be all bad for Chinese, Japanese, and Korean (CJK) people, because their languages very seldom use kerning and ligature. We will focus on two-byte characters, CJK and Unicode characters. There are four kinds of TEX engines which can typeset two-byte characters, Omega [10], Japanese pTEX [12], XƎTEX [6], and ordinary LATEX packages with subfont scheme such as CJK-LATEX [9] and Korean ko.TEX [7]. In the following sections we discuss how DVIasm handles two-byte characters in the (extended) DVI files generated by these TEX engines except for XƎTEX. We require DVIasm to show the two-byte characters contained in the DVI file into a human-readable text format, and conversely, to generate a DVI file from a text file containing two-byte characters.

3. "UTF-8 is a variable-length character encoding for Unicode. It is able to represent any character in the Unicode standard, yet the initial encoding of byte codes and character assignments for UTF-8 is backwards compatible with ASCII." [15, UTF-8].

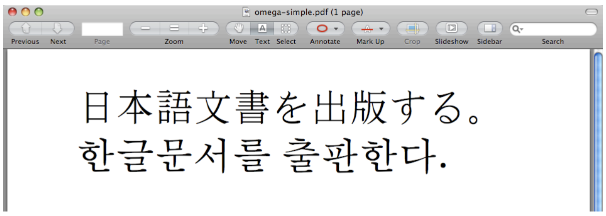
2 Omega and DVIasm

Chinese, Japanese, and Korean languages are characterized by multibyte characters covering more than 60% of two-byte Unicode, UCS-2.[4] Omega is a 16-bit extension of TEX having UCS-2 as its internal representation. In principle, Omega is free from the limitations in handling two-byte characters. Notice that the first version of DVIasm already supported the Omega DVI format, because the default encoding of DVIasm works very well with the internal encoding of Omega.

We provide in Code 2 a DVIasm code containing Japanese and Korean characters, and its output generated by DVIPDFMx [4]. The following commands were run to get the result in the right. Here `omega-simple.dump` is the text file containing the code.

```
$ python dviasm.py omega-simple.dump > omega-simple.dvi
$ dvipdfmx omega-simple
```

```
[page 1]
fnt: cyberb at 40pt
push:
set: ’日本語文書を出版する。’
pop:
down: 50pt
set: ’한글문서를 출판한다.’
```



日本語文書を出版する。
한글문서를 출판한다.

CODE 2. A DVIasm code containing Japanese and Korean characters (left) and its output generated by DVIPDFMx (right).

According to the font selection command ‘`fnt`’ in the second line of the DVIasm code, DVIPDFMx will find the Omega Font Metric (OFM) file, `cyberb.ofm`, the character information of which comes from Bitstream Cyberbit.[5] See [2] for more information about typesetting CJK characters with Omega.

3 pTEX and DVIasm

The most widely used TEX engine in Japan is none other than pTEX, a 16-bit extension of TEX localized to Japanese language. It can handle multi-byte characters natively as Omega. Furthermore, it can typeset vertically as well as horizontally.

```
\nopagenumbers
日本語\TeX を出版分野で
本格的に応用する。
\bye
```

日本語 TEX を出版分野で本格的に応用する。

CODE 3. A pTEX source (left) and its output (right).

4. “UCS-2 (2-byte Universal Character Set) is an obsolete character encoding which is a predecessor to UTF-16.” [15, UCS-2]

5. “Bitstream Cyberbit is a commercial Unicode font designed by Bitstream Inc. It is freeware for non-commercial uses. It was historically one of the first widely available fonts with support for a large proportion of the Unicode repertoire.” [15, Cyberbit]

In the DVI output of pTEX two-byte characters are encoded in ISO-2022-JP (simply JIS). DVIasm converts these characters into UCS-2 using the Python internal library,[6] and then into UTF-8 for the text output. For example, the DVI output of the pTEX source in Code 3 is disassembled into the text output in Code 4 by the following command.

```
$ python dviasm.py --ptex ptex-sample.dvi > ptex-sample.dump
```

The encoding of two-byte characters in a DVI file will be changed from UCS-2 to ISO-2022-JP provided the '`--ptex`' option is given. Switching the encoding of the text output from UTF-8 to Shift-JIS is performed by the option '`--encoding=sjis`'.

```
[preamble]
id: 2
numerator: 25400000
denominator: 473628672
magnification: 1000
comment: ' TeX output 2008...'

[postamble]
maxv: 667.202545pt
maxh: 469.754990pt
maxs: 3
pages: 1

[font definitions]
fntdef: cmr10 at 10pt
fntdef: min10 at 10pt

[page 1 0 0 0 0 0 0 0 0 0]
push:
  down: -14pt
pop:
down: 643.202545pt
push:
  down: -633.202545pt
  push:
    right: 20pt
    fnt: min10 at 10pt
    set: '日本語'
    w: 2.405533pt
    fnt: cmr10 at 10pt
    set: 'T'
    push:
      right: -1.667023pt
      down: 2.152771pt
      set: 'E'
    pop:
    right: 3.888550pt
    set: 'X'
    w0:
    fnt: min10 at 10pt
    set: 'を出版分野で本格的に応用する。'
  pop:
pop:
down: 24pt
```

CODE 4. DVIasm disassembles the DVI output of the pTEX source in Code 3.

Assembling a DVIasm code is also possible by the following command.

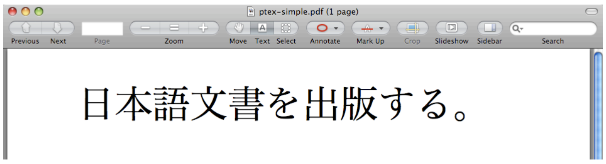
```
$ python dviasm.py --ptex ptex-sample.dump > ptex-sample.dvi
```

Here, the option '`--ptex`' indicates that two-byte characters in DVI must be encoded in ISO-2022-JP. We provide in Code 5 a DVIasm code for pTEX and its output by DVIPDFMx.

pTEX has a new DVI command '`dir`' to support vertical typesetting. One can get a rotated result if '`dir: 1`' is inserted after the first line of the DVIasm code in Code 5. However, two-byte characters should not be rotated in the Japanese vertical typesetting. This purpose can be achieved by changing the font '`min10`' to the corresponding font '`tmin10`' for vertical typesetting.

6. The escape sequence, '`ESC$B`', is attached in front of each two-byte character to switch to JIS X 0208-1983. Then the string is converted to Unicode by the Python method `decode('iso2022-jp')`. See [15, ISO-2022-JP] for more information.

```
[page 1]
fnt: min10 at 40pt
set: '日本語文書を出版する。'
```



日本語文書を出版する。

CODE 5. A DVIasm code for pTEX (left) and its output by DVIPDFMx (right)

4 DVIasm and subfont scheme

Subfont scheme splits a set of large number of characters into groups of 256 characters or less, the number of characters that a TFM file can hold. The main advantage of subfont scheme is to be able to use 8-bit TEX engines such as pdfTEX directly. CJK-LATEX and Korean koTEX are representatives of the most popular and powerful LATEX package supporting subfont scheme.

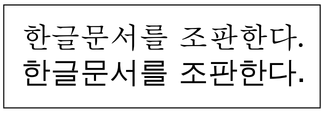
DVIasm does not use Subfont Definition Files (SFD) in checking whether the given font is a subfont or not. Instead it assumes that all subfonts are inherited from one SFD called 'Unicode.sfd'. Then, the first byte of each two-byte charcater is used for selecting an 8-bit font and the second byte is used for the position in the selected font. For example, the UCS-2 code of the character '한' is 0xd55c. If 'outbtm' is the subfont base name, the character will be assigned to the 0x5c entry of the font 'outbtmd5'.

DVIasm provides the option '--subfont' to receive the list of subfont base names from user. Some popular subfont base names such as Un fonts are already recorded inside of 'dviasm.py'. The argument of '--subfont' is a string of comma separated font names. If it is not given, DVIasm uses only the internal subfont list.

The following command interprets the DVIasm code in Code 6 and generate a DVI file. Here, two subfont base names 'outbtm' and 'outgtm' are given explicitly.

```
$ python dviasm.py --subfont=outbtm,outgtm, \
                 kotex-sample.dump > kotex-sample.dvi
```

```
[page 1]                      down: 12pt
push:                         push:
  fnt: outbtm at 10pt           fnt: outgtm at 10pt
  set: '한글문서를'               set: '한글문서를'
  right: 3.333328pt             right: 3.333328pt
  set: '조판한다'                 set: '조판한다'
pop:                          pop:
```



한글문서를 조판한다.
한글문서를 조판한다.

CODE 6. A DVIasm code for koTEX (left) and its output by DVIPDFMx (right).

5 Conclusion

DVIasm is an auxiliary program for TEX and its various extensions. The main purpose of this program is changing small part or adding some decorations in the DVI file generated by TEX engines. In the second stage of development, DVIasm now supports two important 16-bit TEX engines, Omega and pTEX. It also supports subfont scheme within the limit of UCS-2 based subfonts. CJK-LATEX and ko.TEX are the representatives of this kind of subfont scheme provided the UTF-8 encoding is used.

However, it is still not much convenient to use DVIasm in editing DVI files because of two typographic features, kerning and ligature. To support these features, DVIasm needs to handle font metric information directly from the Python source. Moreover, communication with the kpathsea library is also required. These topics will be the main theme of the final stage of the development of DVIasm. I hope DVIasm works well with XETEX and LuaTEX at that time.

References

1. Jin-Hwan CHO, *Hacking DVI files: Birth of DVIasm*, The PracTEX Journal (2007), no. 1, and TUGboat **28** (2007), no. 2, 210–217. http://www.tug.org/TUGboat/Articles/tb28-2/tb89cho.pdf
2. Jin-Hwan CHO and Haruhiko OKUMURA, *Typesetting CJK languages with Omega*, TeX, XML, and Digital Typography, Lecture Notes in Computer Science, vol. 3130, pp. 139–148, 2004.
3. Jin-Hwan CHO, *The DVIasm Python Script*. http://svn.ktug.or.kr/viewvc/dviasm/?root=ChoF
4. Jin-Hwan CHO, Matthias FRANZ, and Shunsaku HIRATA, *The DVIPDFMx Project*. http://project.ktug.or.kr/dvipdfmx/
5. Hans HAGEN, Hartmut HENKEL, and Taco HOEKWATER, *LuaTEX homepage*. http://www.luatex.org
6. Jonathan KEW, *The XeTeX typesetting system*. http://scripts.sil.org/xetex
7. Dohyun KIM, Kangsoo KIM, and Koanghi UN, *Korean TEX (ko.TEX)*. http://project.ktug.or.kr/ko.TeX
8. Donald E. KNUTH, *The DVItype processor* (Version 3.6, December 1995). CTAN:systems/knuth/texware/dvitype.web
9. Werner LEMBERG, *CJK extensions for LATEX*. http://cjk.ffii.org
10. John PLAICE and Yannis HARALAMBOUS, *The Omega Typesetting and Document Processing System*. http://omega.enstb.org
11. Geoffrey TOBIN, *The DTL Package* (Version 0.6.1, March 1995). CTAN:dviware/dtl/
12. ASCII Corporation, *ASCII Nihongo TEX (Publishing TEX)*. http://www.ascii.co.jp/pb/ptex/
13. Python Software Foundation, *Python Programming Language*. http://www.python.org
14. CollabNet, *Subversion*. http://subversion.tigris.org
15. *Wikipedia, the free encyclopedia*, http://wikipedia.org