

# About p $\text{\LaTeX}$ 2 $\epsilon$

Ken Nakano & Japanese  $\text{\TeX}$  Development Community

Date: 2017/12/02

## *Attention:*

This document provides a brief description of p $\text{\LaTeX}$  2 $\epsilon$ , the Japanese extended version of  $\text{\LaTeX}$  2 $\epsilon$ . The original version of p $\text{\LaTeX}$  2 $\epsilon$  ('ASCII p $\text{\LaTeX}$  2 $\epsilon$ ') was developed by ASCII MEDIA WORKS<sup>1</sup> (formerly ASCII CORPORATION) during 1995 and 2006. The current version, which is now distributed in CTAN and  $\text{\TeX}$  Live, is a forked version called 'p $\text{\LaTeX}$  2 $\epsilon$  Community Edition.' It is now maintained by Japanese  $\text{\TeX}$  Development Community<sup>2</sup>.

p $\text{\LaTeX}$  is a Japanese  $\text{\LaTeX}$  format, which is adjusted/extended to be more suitable for writing Japanese documents. It requires p $\text{\TeX}$  ( $\text{\TeX}$  engine with extensions for Japanese typesetting; it is designed for high-quality Japanese book publishing, and the 'p' stands for 'publishing'<sup>3</sup>), and the development of both p $\text{\TeX}$  and p $\text{\LaTeX}$  was done by ASCII Corporation.

In 2010, ASCII p $\text{\TeX}$  was incorporated into the world-wide  $\text{\TeX}$  distribution ' $\text{\TeX}$  Live.' Since then, p $\text{\TeX}$  has been maintained/improved/changed along with  $\text{\TeX}$  Live sources. In recent versions of  $\text{\TeX}$  Live and W32 $\text{\TeX}$ , the default engine of p $\text{\LaTeX}$  changed from original p $\text{\TeX}$  to  $\epsilon$ -p $\text{\TeX}$  (p $\text{\TeX}$  with  $\epsilon$ -p $\text{\TeX}$  extension), and the original  $\text{\LaTeX}$  itself is also frequently updated. On the other hand, p $\text{\LaTeX}$  remained unchanged since 2006, which resulted in some incompatibility and limitations.

To follow these upstream changes, we (Japanese  $\text{\TeX}$  Development Community) decided to fork ASCII p $\text{\LaTeX}$  and distribute the 'community edition.' The development version is available from GitHub repository<sup>4</sup>. The forked community

---

<sup>1</sup><http://ascii.asciimw.jp/pb/ptex/>

<sup>2</sup><https://texjp.org>

<sup>3</sup>There is another old implementation of Japanese  $\text{\LaTeX}$ , called j $\text{\LaTeX}$  2 $\epsilon$  (but not included in  $\text{\TeX}$  Live). Also, MiK $\text{\TeX}$  has another program called 'platex,' but it has nothing to do with our Japanese p $\text{\LaTeX}$ !

<sup>4</sup><https://github.com/texjporg/platex>

edition is different from the original ASCII edition, so any bug reports and requests should be sent to Japanese T<sub>E</sub>X Development Community, using GitHub Issue system.

This document (platex-en.pdf) is a brief explanation of the pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> community edition. It is somewhat of a historical document now, since pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> came into existence in 1995 (although the English translation has been done by Japanese T<sub>E</sub>X Development Community since 2017). The detail of source codes are described separately in pldoc-en.pdf.

# 1 Introduction to this document

This document briefly describes p $\LaTeX$  2 $\epsilon$ , but is not a manual of p $\LaTeX$  2 $\epsilon$ . For the basic functions of p $\LaTeX$  2 $\epsilon$ , see [1] (in Japanese). For extensions of some commands for vertical writing (which were first described in [2] in Japanese), see `plex.ttx` section in `pldoc-en.pdf`.

For Japanese typesetting, please refer to the documentation of p $\TeX$  (or “Japanese  $\TeX$ ”; the preliminary version of p $\TeX$ ), [3] (in Japanese), [4] (in English) and [5] (in English).

This document consists of following parts:

**Section 1** This section; describes this document itself.

**Section 2** Brief explanation of extensions in p $\LaTeX$  2 $\epsilon$ . Also describes the standard classes and packages.

**Section 3** The compatibility note for users of the old version of p $\LaTeX$  2 $\epsilon$  or those of the original  $\LaTeX$  2 $\epsilon$ .

**Appendix A** Describes DOCSTRIP Options for this document.

**Appendix B** Description of ‘`pldoc.tex`’ (counterpart for ‘`source2e.tex`’ in  $\LaTeX$  2 $\epsilon$ ).

**Appendix C** Description of a shell script to process ‘`pldoc.tex`’, and a tiny perl program to check DOCSTRIP guards, etc.

## 2 About Functions of p $\LaTeX$ 2 $\epsilon$

The structure of p $\LaTeX$  2 $\epsilon$  is similar to that of  $\LaTeX$  2 $\epsilon$ ; it consists of 3 types of files: a format (`platex.ltx`), classes and packages.

### 2.1 About the Format

To make a format for p $\LaTeX$ , process “`platex.ltx`” with INI mode of  $\epsilon$ -p $\TeX$ .<sup>5</sup> A handy command ‘`fmtutil-sys`’ (or ‘`fmtutil`’) for this purpose is available in  $\TeX$  Live. The following command generates `platex.fmt`.

```
fmtutil-sys --byfmt platex
```

---

<sup>5</sup>Formerly both p $\TeX$  and  $\epsilon$ -p $\TeX$  can make the format file for p $\LaTeX$ , however, it’s not true anymore because  $\LaTeX$  requires  $\epsilon$ -p $\TeX$  since 2017.

The content of `platex.ltx` is shown below. In the current version of p<sup>L</sup>A<sup>T</sup>E<sub>X</sub>, first we simply load `latex.ltx` and modify/extend some definitions by loading `plcore.ltx`.

```

1 <(*plcore>

   Temporarily disable \dump at the end of latex.ltx.
2 \let\orgdump\dump
3 \let\dump\relax

   Load latex.ltx here. Within the standard installation of TEX Live, hyphen.cfg
   provided by “Babel” package will be used.
4 \input latex.ltx

   Load plcore.ltx.
5 \typeout{*****~J%
6         *~J%
7         * making pLaTeX format~J%
8         *~J%
9         *****}
10 \makeatletter
11 \input plcore.ltx

```

Display p<sup>L</sup>A<sup>T</sup>E<sub>X</sub> version on the terminal, so that it can be easily recognized during format creation.

```

12 \the\everyjob

   Load platex.cfg if it exists at runtime.
13 \everyjob\expandafter{%
14   \the\everyjob
15   \IfFileExists{platex.cfg}{%
16     \typeout{*****~J%
17             * Loading platex.cfg.~J%
18             *****}%
19     \input{platex.cfg}}{%
20 }

```

Dump to the format file.

```

21 \let\dump\orgdump
22 \let\orgdump\@undefined
23 \makeatother
24 \dump
25 %\endinput
26 </plcore>

```

The file `plcore.ltx`, which provides modifications/extensions to make p<sup>L</sup>A<sup>T</sup>E<sub>X</sub> 2<sub>ε</sub>, is a concatenation of stripped files below using DOCSTRIP program.

- `plvers.dtx` defines the format version of p<sup>L</sup>A<sup>T</sup>E<sub>X</sub> 2<sub>ε</sub>.

- `plfonts.dtx` extends NFSS2 for Japanese font selection.
- `plcore.dtx` defines other modifications to  $\text{\LaTeX} 2_{\epsilon}$ .

Moreover, default settings of pre-loaded fonts and typesetting parameters are done by loading `pldefs.ltx` inside `plcore.ltx`. This file `pldefs.ltx` is also stripped from `plfonts.dtx`.

*Attention:*

You can customize  $\text{\LaTeX} 2_{\epsilon}$  by tuning these settings. If you need to do that, copy/rename it as `pldefs.cfg` and edit it, instead of overwriting `pldefs.ltx` itself. If a file named `pldefs.cfg` is found at a format creation time, it will be read as a substitute of `pldefs.ltx`.

### 2.1.1 Version

The version (like “2017/10/28”) and the format name (“`pLaTeX2e`”) of  $\text{\LaTeX} 2_{\epsilon}$  are defined in `plvers.dtx`.

### 2.1.2 NFSS2 Commands

$\text{\LaTeX} 2_{\epsilon}$  uses NFSS2 as a font selection scheme, however, it supports only alphabetic fonts.  $\text{\LaTeX} 2_{\epsilon}$  extends NFSS2 to enable selection of Japanese fonts in a consistent manner with the original NFSS2.

Most of the interface commands are defined to be clever enough, so that it can automatically judge whether it is going to change alphabetic fonts or Japanese fonts. It works almost fine with most of the widely used classes and packages, without any modification.

For the detail of (the original) NFSS2, please refer to `fntguide.tex` in  $\text{\LaTeX} 2_{\epsilon}$ .

### 2.1.3 Output Routine and Floats

`plcore.dtx` modifies and extends some  $\text{\LaTeX} 2_{\epsilon}$  commands for Japanese processing.

- Preamble commands
- Page breaking
- Line breaking
- The order of float objects

- Crop marks (“tombow”)
- Footnote macros
- Cross-referencing
- Verbatim

## 2.2 Classes and Packages

Classes and packages bundled with p $\text{\LaTeX}$  2 $\epsilon$  are based on those in original  $\text{\LaTeX}$  2 $\epsilon$ , with some Japanese localization.

p $\text{\LaTeX}$  2 $\epsilon$  classes:

- jarticle.cls, jbook.cls, jreport.cls  
Standard *yoko-kumi* (horizontal writing) classes; stripped from `jclasses.dtx`.
- tarticle.cls, tbook.cls, treport.cls  
Standard *tate-kumi* (vertical writing) classes; stripped from `jclasses.dtx`.
- jltxdoc.cls  
Class for typesetting Japanese `.dtx` file; stripped from `jltxdoc.dtx`.

p $\text{\LaTeX}$  2 $\epsilon$  packages:

- plect.sty  
Useful macros and extensions for vertical writing; stripped from `plext.dtx`.
- ptrace.sty  
p $\text{\LaTeX}$  2 $\epsilon$  version of `tracefmt.sty`; the package `tracefmt.sty` overwrites p $\text{\LaTeX}$  2 $\epsilon$ -style NFSS2 commands, so `ptrace.sty` provides redefinitions to recover p $\text{\LaTeX}$  2 $\epsilon$  extensions. Stripped from `plfonts.dtx`.
- pfltrace.sty  
p $\text{\LaTeX}$  2 $\epsilon$  version of `fltrace.sty` (introduced in  $\text{\LaTeX}$  2 $\epsilon$  2014/05/01); stripped from `plcore.dtx`.
- oldpfont.sty  
Provides p $\text{\LaTeX}$  2.09 font commands; stripped from `p1209.dtx`.

- `ascmac.sty`, `tascmac.sty`

Create fancy boxes etc., within the scope of native  $\text{\LaTeX} 2_{\epsilon}$  picture commands (thus driver-independent).

- `nidanfloat.sty`

Enable bottom (b) placement option for double float in two column mode (*nidan-kumi*).

### 3 Compatibility with Other Formats and Older Versions

Here we provide some information about the compatibility between current  $\text{p}\text{\LaTeX} 2_{\epsilon}$  and older versions or original  $\text{\LaTeX} 2_{\epsilon}$ .

#### 3.1 Compatibility with $\text{\LaTeX} 2_{\epsilon}$

$\text{p}\text{\LaTeX} 2_{\epsilon}$  is in most part upper compatible with  $\text{\LaTeX} 2_{\epsilon}$ , but some parameters are adjusted to be suitable for Japanese. Therefore, you should not expect identical output, even though the same source can be processed on both  $\text{\LaTeX} 2_{\epsilon}$  and  $\text{p}\text{\LaTeX} 2_{\epsilon}$ .

We hope that most classes and packages meant for  $\text{\LaTeX} 2_{\epsilon}$  works also for  $\text{p}\text{\LaTeX} 2_{\epsilon}$  without any modification. However for example, if a class or a package redefines a command which is already modified by  $\text{p}\text{\LaTeX} 2_{\epsilon}$ , it might cause an error at the worst case. We cannot tell whether a class or a package works fine with  $\text{p}\text{\LaTeX} 2_{\epsilon}$  beforehand; the easiest way is to try to use it. If it fails, please refer to the log file or a package manual.

#### 3.2 Compatibility with $\text{p}\text{\LaTeX} 2.09$

$\text{p}\text{\LaTeX} 2_{\epsilon}$  has ‘ $\text{p}\text{\LaTeX} 2.09$  compatibility mode’; use `\documentstyle` to enter it, but the support might be limited. Note that the 2.09 compatibility mode is provided solely to allow you to process very old documents, which were written for a very old system.

#### 3.3 Support for Package ‘`latexrelease`’

$\text{p}\text{\LaTeX}$  provides ‘`latexrelease`’ package, which is based on ‘`latexrelease`’ package (introduced in  $\text{\LaTeX}$  <2015/01/01>). It may be used to ensure stability where

needed, by emulating the specified format date without regenerating the format file. For more detail, please refer to its documentation.

## A DOCSTRIP Options

By processing `platex.dtx` with DOCSTRIP program, different files can be generated. Here are the DOCSTRIP options for this document:

<i>Option</i>	<i>Function</i>
<code>plcore</code>	Generates a fragment of format sources
<code>pldoc</code>	Generates ‘ <code>pldoc.tex</code> ’ for typesetting pL <sup>A</sup> T <sub>ε</sub> X sources
<code>shprog</code>	Generates a shell script to process ‘ <code>pldoc.tex</code> ’
<code>plprog</code>	Generates a tiny perl program to check DOCSTRIP guards nesting
<code>Xins</code>	Generates a DOCSTRIP batch file ‘ <code>Xins.ins</code> ’ for generating the above shell/perl scripts

## B Documentation of pL<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> sources

The contents of ‘`pldoc.tex`’ for typesetting pL<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> sources is described here. Compared to individual processings, batch processing using ‘`pldoc.tex`’ prints also changes and an index. The whole document will have about 200 pages.

By default, the description of pL<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> sources is written in Japanese. If you need English version, first save

```
\newif\ifJAPANESE
```

as `platex.cfg`, and process `pldoc.tex` (pL<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> Community Edition newer than July 2016 is required).

First, create `pldoc.dic`; it serves as a dictionary for ‘`mendex`’ (Japanese index processor<sup>6</sup>), which is necessary for indexing control sequences containing Japanese characters (`\西暦` and `\和暦`).

```
27 < *pldoc>
28 \begin{filecontents}{pldoc.dic}
29 西暦      せイレキ
30 和暦      わレキ
31 \end{filecontents}
```

---

<sup>6</sup>Developed by ASCII Corporation; the program ‘`makeindex`’ cannot handle Japanese characters properly, especially Kanji characters which should be sorted by its readings.



We use `jltxdoc` class; we also require `plext` package, since `plext.dtx` contains several examples of partial vertical writing.

```
32 \documentclass{jltxdoc}
33 \usepackage{plext}
34 \listfiles
35
```

Do not index some  $\TeX$  primitives, and some common plain  $\TeX$  commands.

```
36 \DoNotIndex{\def,\long,\edef,\xdef,\gdef,\let,\global}
37 \DoNotIndex{\if,\ifnum,\ifdim,\ifcat,\ifmmode,\ifvmode,\ifhmode,%
38             \iftrue,\iffalse,\ifvoid,\ifx,\ifeof,\ifcase,\else,\or,\fi}
39 \DoNotIndex{\box,\copy,\setbox,\unvbox,\unhbox,\hbox,%
40             \vbox,\vtop,\vcenter}
41 \DoNotIndex{\@empty,\immediate,\write}
42 \DoNotIndex{\egroup,\bgroup,\expandafter,\begingroup,\endgroup}
43 \DoNotIndex{\divide,\advance,\multiply,\count,\dimen}
44 \DoNotIndex{\relax,\space,\string}
45 \DoNotIndex{\csname,\endcsname,\@spaces,\openin,\openout,%
46             \closein,\closeout}
47 \DoNotIndex{\catcode,\endinput}
48 \DoNotIndex{\jobname,\message,\read,\the,\m@ne,\noexpand}
49 \DoNotIndex{\hsize,\vsize,\hskip,\vskip,\kern,\hfil,\hfill,\hss,\vss,\unskip}
50 \DoNotIndex{\m@ne,\z@,\z@skip,\@ne,\tw@,\p@,\@minus,\@plus}
51 \DoNotIndex{\dp,\wd,\ht,\setlength,\addtolength}
52 \DoNotIndex{\newcommand,\renewcommand}
53
```

Set up the Index and Change History to use `\part`.

```
54 \ifJAPANESE
55 \IndexPrologue{\part*{索引}}%
56             \markboth{索引}{索引}%
57             \addcontentsline{toc}{part}{索引}%
58 イタリアック体の数字は、その項目が説明されているページを示しています。
59 下線の引かれた数字は、定義されているページを示しています。
60 その他の数字は、その項目が使われているページを示しています。}
61 \else
62 \IndexPrologue{\part*{Index}}%
63             \markboth{Index}{Index}%
64             \addcontentsline{toc}{part}{Index}%
65 The italic numbers denote the pages where the corresponding entry
66 is described, numbers underlined point to the definition,
67 all others indicate the places where it is used.}
68 \fi
69 %
70 \ifJAPANESE
71 \GlossaryPrologue{\part*{変更履歴}}%
72             \markboth{変更履歴}{変更履歴}%
73             \addcontentsline{toc}{part}{変更履歴}}
74 \else
```

```

75 \GlossaryPrologue{\part*{Change History}%
76             \markboth{Change History}{Change History}%
77             \addcontentsline{toc}{part}{Change History}}
78 \fi
79

```

Modify the standard `\changes` command slightly, to better cope with this multiple file document.

```

80 \makeatletter
81 \def\changes@#1#2#3{%
82   \let\protect\@unexpandable@protect
83   \edef\@tempa{\noexpand\glossary{#2\space\currentfile\space#1\levelchar
84             \ifx\saved@macroname\@empty
85             \space\actualchar\generalname
86             \else
87             \expandafter\@gobble
88             \saved@macroname\actualchar
89             \string\verb\quotechar*%
90             \verbatimchar\saved@macroname
91             \verbatimchar
92             \fi
93             : \levelchar #3}}%
94   \@tempa\endgroup\@esphack}
95 \makeatother

```

Produce a Change Log and (2 column) Index.

```

96 \RecordChanges
97 \CodelineIndex
98 \EnableCrossrefs
99 \setcounter{IndexColumns}{2}
100 \settowidth\MacroIndent{\ttfamily\scriptsize 000\ }

```

Here starts the document body.

```

101 \begin{document}
102   \title{The \pLaTeXe\ Sources}
103   \author{Ken Nakano \& Japanese \TeX\ Development Community}
104
105 % This command will be used to input the patch file
106 % if that file exists.
107 \newcommand{\includeltpatch}{%
108   \def\currentfile{plpatch.ltx}
109   \part{plpatch}
110   {\let\ttfamily\relax
111     \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
112   Things we did wrong\ldots
113   \IndexInput{plpatch.ltx}}
114
115 % Get the date and patch level from plvers.dtx
116 \makeatletter
117 \let\patchdate=\@empty

```

```

118 \begingroup
119   \def\ProvidesFile#1\pfmtversion#2#3\ppatch@level#4{%
120     \date{#2}\xdef\patchdate{#4}\endinput}
121   \input{plvers.dtx}
122 \global\let\X@date=\@date
123
124 % Add the patch version if available.
125   \long\def\Xdef#1#2#3\def#4#5{%
126     \xdef\X@date{#2}%
127     \xdef\patchdate{#5}%
128     \endinput}%
129   \InputIfFileExists{plpatch.ltx}
130     {\let\def\Xdef}{\global\let\include\patch\relax}
131 \endgroup
132
133 \ifx\@date\X@date
134   \def\Xpatch{0}
135   \ifx\patchdate\Xpatch\else
136     % number is assumed
137     \ifnum\patchdate>0
138       \edef\@date{\@date\space Patch level\space\patchdate}
139     \else
140       \edef\@date{\@date\space Pre-Release\patchdate}
141     \fi\fi
142 \else
143   \@warning{plpatch.ltx does not match plvers.dtx!}
144   \let\include\patch\relax
145 \fi
146 \makeatother
147
148 \pagenumbering{roman}
149 \maketitle
150 \renewcommand\maketitle{}
151 \tableofcontents
152 \clearpage
153 \pagenumbering{arabic}
154
155 \DocInclude{plvers}    % pLaTeX version
156
157 \DocInclude{plfonts}  % NFSS2 commands
158
159 \DocInclude{plcore}   % kernel commands
160
161 \DocInclude{plext}    % external commands
162
163 \DocInclude{pl209}    % 2.09 compatibility mode commands
164
165 \DocInclude{kinsoku}  % kinsoku parameter
166
167 \DocInclude{jclasses} % Standard class

```

```

168
169 \DocInclude{jltxdoc} % dtx documents class
170
171 %\includeltpatch      % patch file (comment out May 8, 2016)
172
    Stop here if ltxdoc.cfg says \AtEndOfClass{\OnlyDescription}.
173 \StopEventually{\end{document}}
174

    Print Change History and Index. Please refer to Appendix C.1 for processing of
    Change History and Index.

175 \clearpage
176 \pagestyle{headings}
177 % Make TeX shut up.
178 \hbadness=10000
179 \newcount\hbadness
180 \hfuzz=\maxdimen
181 %
182 \PrintChanges
183 \clearpage
184 %
185 \begingroup
186   \def\endash{--}
187   \catcode'\- \active
188   \def-{\futurelet\temp\indexdash}
189   \def\indexdash{\ifx\temp-\endash\fi}
190
191   \PrintIndex
192 \endgroup

    Make sure that the index is not printed twice (ltxdoc.cfg might have a second
    command).

193 \let\PrintChanges\relax
194 \let\PrintIndex\relax
195 \end{document}
196 \pdoc)

```

## C Additional Utility Programs

### C.1 Shell Script `mkpldoc.sh`

A shell script to process ‘`pdoc.tex`’ and produce a fully indexed source code description. Run `sh mkpldoc.sh` to use it.

### C.1.1 Content of mkpldoc.sh

First, delete auxiliary files which might be created in the previous runs.

```
197 <shprog>
198 for f in pldoc.toc pldoc.idx pldoc.glo ; do
199 if [ -e $f ]; then rm $f; fi
200 done
```

First run: empty the config file ltxdoc.cfg.

```
201 echo "" > ltxdoc.cfg
```

Now process pldoc.tex.

```
202 platex pldoc.tex
```

Make the Change log and Glossary (Change History) using mendex. ‘Mendex’ is a Japanese index processor, which is mostly upper compatible with ‘makeindex’ and automatically handles readings of Kanji words.

Option `-s` employs a style file for formatting. Here we use `gind.ist` and `gglo.ist` from L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

Option `-o` specifies output index file name.

Option `-f` forces to output Kanji characters even non-existent in dictionaries. (Makeindex does not have this option.)

```
203 mendex -s gind.ist -d pldoc.dic -o pldoc.ind pldoc.idx
204 mendex -f -s gglo.ist -o pldoc.gls pldoc.glo
```

Second run: append `\includeonly{}` to ltxdoc.cfg to speed up things. This run is needed only to get changes and index listed in .toc file.

```
205 echo "\includeonly{" > ltxdoc.cfg
206 platex pldoc.tex
```

Third and final run: restore the cfg file to put everything together.

```
207 echo "" > ltxdoc.cfg
208 platex pldoc.tex
209 # EOT
210 </shprog>
```

### C.2 Perl Script dstcheck.pl

Here we provide a perl script which helps checking the nested DOCSTRIP guards.

Usage:

```
perl dstcheck.pl <file-name>
```

The description of this script itself is available only in Japanese.

```
211 <plprog>
212 ##
```

```

213 ## DOCSTRIP 文書内の環境や条件の入れ子を調べる perl スクリプト
214 ##

215 push(@dst,"DUMMY"); push(@dst,"000");
216 push(@env,"DUMMY"); push(@env,"000");

217 while (<>) {

218   if (/^%<*(\[>]+\>)/) { # check conditions
219     push(@dst,$1);
220     push(@dst,$.);
221   } elsif (/^%<\/([>]+\>)/) {
222     $linenum = pop(@dst);
223     $conditions = pop(@dst);
224     if ($1 ne $conditions) {
225       if ($conditions eq "DUMMY") {
226         print "ARGV: '</$1>' (l.$.) is not started.\n";
227         push(@dst,"DUMMY");
228         push(@dst,"000");
229       } else {
230         print "ARGV: '<*$conditions>' (l.$linenum) is ended ";
231         print "by '<*$1>' (l.$.)\n";
232       }
233     }
234   }

235   if (/^% *\\begin\\{verbatim\\}/) { # check environments
236     while(<>) {
237       last if (/^% *\\end\\{verbatim\\}/);
238     }
239   } elsif (/^% *\\begin\\{([~}]+)\\}\\{(.*)\\}/) {
240     push(@env,$1);
241     push(@env,$.);
242   } elsif (/^% *\\begin\\{([~}]+)\\}/) {
243     push(@env,$1);
244     push(@env,$.);
245   } elsif (/^% *\\end\\{([~}]+)\\}/) {
246     $linenum = pop(@env);
247     $environment = pop(@env);
248     if ($1 ne $environment) {
249       if ($environment eq "DUMMY") {
250         print "ARGV: '\\end{$1}' (l.$.) is not started.\n";
251         push(@env,"DUMMY");
252         push(@env,"000");
253       } else {
254         print "ARGV: '\\begin{$environment}' (l.$linenum) is ended ";
255         print "by '\\end{$1}' (l.$.)\n";
256       }
257     }
258   }
259 }

```

```

260 $linenum = pop(@dst);
261 $conditions = pop(@dst);
262 while ($conditions ne "DUMMY") {
263     print "$ARGV: '<*$conditions>' (1.$linenum) is not ended.\n";
264     $linenum = pop(@dst);
265     $conditions = pop(@dst);
266 }

267 $linenum = pop(@env);
268 $environment = pop(@env);
269 while ($environment ne "DUMMY") {
270     print "$ARGV: '\\begin{$environment}' (1.$linenum) is not ended.\n";
271     $linenum = pop(@env);
272     $environment = pop(@env);
273 }
274 exit;
275 </plprog>

```

### C.3 DOCSTRIP Batch file

Here we introduce a DOCSTRIP batch file ‘Xins.ins,’ which generates the scripts described in Appendix C.1 and C.2.

```

276 < *Xins>
277 \input docstrip
278 \keepsilent

279 {\catcode'#=12 \gdef\MetaPrefix{## }}

280 \declarepreamble\thispre
281 \endpreamble
282 \usepreamble\thispre

283 \declarepostamble\thispost
284 \endpostamble
285 \usepostamble\thispost

286 \generate{
287     \file{dstcheck.pl}{\from{platex.dtx}{plprog}}
288     \file{mkpldoc.sh}{\from{platex.dtx}{shprog}}
289 }
290 \endbatchfile
291 </Xins>

```

## References

- [1] 中野 賢『日本語 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> ブック』アスキー, 1996.
- [2] インプレス・ラボ監修, アスキー書籍編集部編『縦組対応 パーソナル日本語 T<sub>E</sub>X』アスキー出版局, 1994
- [3] アスキー出版技術部責任編集『日本語 T<sub>E</sub>X テクニカルブック I』アスキー, 1990.
- [4] Haruhiko Okumura, pT<sub>E</sub>X and Japanese Typesetting The Asian Journal of T<sub>E</sub>X, Volume 2, No. 1, 2008.  
(<http://ajt.ktug.org/2008/0201okumura.pdf>)
- [5] Hisato Hamano, Vertical Typesetting with T<sub>E</sub>X. TUGboat issue 11:3, 1990.  
(<https://tug.org/TUGboat/tb11-3/tb29hamano.pdf>)
- [6] Donald E. Knuth. “*The T<sub>E</sub>Xbook*”. Addison-Wesley, 1984. (邦訳：斎藤信男監修, 鷺谷好輝訳, T<sub>E</sub>X ブック 改訂新版, アスキー出版局, 1989)
- [7] Laslie Lamport. “*L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*”. Addison-Wesley, second edition, 1994.
- [8] Laslie Lamport. “*L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*”. Addison-Wesley, 1986. (邦訳：倉沢良一監修, 大野俊治・小暮博通・藤浦はる美訳, 文書処理システム L<sup>A</sup>T<sub>E</sub>X, アスキー, 1990)
- [9] Michel Goossens, Frank Mittelbach, Alexander Samarin. “*The L<sup>A</sup>T<sub>E</sub>X Companion*”. Addison-Wesley, 1994.
- [10] 河野 真治『入門 Perl』アスキー出版局, 1994



## Change History

1995/05/08 v1.0		2016/05/07 v1.0g	
first edition . . . . .	3	Save L <sup>A</sup> T <sub>E</sub> X banner . . . . .	4
1995/08/25 v1.0a		2016/05/08 v1.0h	
Added ‘Compatibility’, ‘Usage of		Exclude <b>plpatch.ltx</b> from the	
DOCSTRIP’ and ‘References’ . . .	3	document . . . . .	10
1996/02/01 v1.0b		2016/05/12 v1.0i	
Adjusted for the latest DOCSTRIP		Undefine temporary command	
( <b>omake-sh.ins</b> and		<b>\orgdump</b> in the end. . . . .	4
<b>omake-pl.ins</b> . . . . .	15	2016/05/20 v1.0j	
1997/01/23 v1.0c		Add description of ‘pfltrace’ . . . . .	6
Adjusted for the latest DOCSTRIP. .	15	2016/05/21 v1.0k	
Don’t copy gind.ist and gglo.ist		Print also changes. . . . .	1
from		2016/06/19 v1.0l	
\$TEXMF/tex/latex2e/base		Get the patch level from	
directory. . . . .	13	<b>plvers.dtx</b> . . . . .	10
1997/01/25 v1.0c		2016/08/26 v1.0m	
Add to filecontents environment		Moved loading <b>platex.cfg</b> from	
for pldoc.dic. . . . .	8	<b>plcore.ltx</b> to <b>platex.ltx</b> . . .	4
1997/01/29 v1.0c		2016/09/14 v1.0n	
Rename <b>pltpatch.ltx</b> to		Improved banner saving method . .	4
<b>plpatch.ltx</b> . . . . .	10	2017/09/24 v1.0o	
2016/01/27 v1.0d		Allow negative patch level for	
Add -e test before rm command . .	13	pre-release . . . . .	10
Updated descriptions of pL <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub>		2017/11/11 v1.0p	
files . . . . .	6	Moved banner saving code from	
2016/02/16 v1.0e		<b>platex.ltx</b> to <b>plcore.ltx</b> . . .	4
Add a description of platexrelease	7	2017/12/02 v1.0r	
2016/04/12 v1.0f		English references added . . . . .	3
Update document. . . . .	1		