About $\text{pL\TeX}\,2\varepsilon$

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

Date: 2019/09/29

$p\text{L\TeX}$ is a Japanese $\text{L\TeX}$ format, which is adjusted/extended to be more suitable for writing Japanese documents. It requires $p\text{TeX}$\textsuperscript{1}, a $\text{TeX}$ engine with extensions for Japanese typesetting, which is designed for high-quality Japanese book publishing.\textsuperscript{2} Both of them were developed by ASCII Corporation (and its successor ASCII Media Works), so they are often referred to as “ASCII $p\text{TeX}$” and “ASCII $p\text{L\TeX}$” respectively.

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$\text{TeX}$ (around 2011), the default engine of $p\text{L\TeX}$ changed from original $p\text{TeX}$ to $\varepsilon$-$p\text{TeX}$ ($p\text{TeX}$ with $\varepsilon$-$\text{TeX}$ extension). Also, the original $p\text{L\TeX}$ itself is also frequently updated. On the other hand, $p\text{L\TeX}$ remained unchanged since 2006, which resulted in some incompatibility and limitations.

To follow these upstream changes, we (Japanese $\text{TeX}$ Development Community\textsuperscript{3}) decided to fork ASCII $p\text{L\TeX}$ and distribute the “community edition.” The development version is available from GitHub repository\textsuperscript{4}. The forked community edition is different from the original ASCII edition, so any bug reports and requests should be sent to Japanese $\text{TeX}$ Development Community, using GitHub Issue system.

This document (platex-en.pdf) is a brief explanation of the $p\text{L\TeX}\,2\varepsilon$ community edition. It is somewhat of a historical document now, since $p\text{L\TeX}\,2\varepsilon$ came into existence in 1995 (although the English translation has been done by Japanese $\text{TeX}$ Development Community since 2017).

\textsuperscript{1}The $p\text{TeX}$ website: https://asciidwango.github.io/ptex/ (in Japanese)

\textsuperscript{2}There is another old implementation of Japanese $\text{L\TeX}$ by NTT Electrical Communications Laboratories, named $\text{jL\TeX}$ (unavailable in $\text{TeX}$ Live). Also, MiKTeX has another program $\text{platex}$ for Polish, but it has nothing to do with our Japanese $p\text{L\TeX}$!

\textsuperscript{3}https://texjp.org

\textsuperscript{4}https://github.com/texjporg/platex
1 Introduction to this document

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

For Japanese typesetting, please refer to the documentation of pTEX (or “Japanese TEX”; the preliminary version of pTEX), [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 pLATEX 2ε. Also describes the standard classes and packages.

Section 3 The compatibility note for users of the old version of pLATEX 2ε or those of the original LATEX 2ε.

Appendix A Describes DOCSTRIP Options for this document.

Appendix B Description of ‘pldoc.tex’ (counterpart for ‘source2e.tex’ in LATEX 2ε).

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 pLATEX 2ε

The structure of pLATEX 2ε is similar to that of LATEX 2ε; it consists of 3 types of files: a format (platex.ltx), classes and packages.

2.1 About the Format

To make a format for pLATEX, process “platex.ltx” with INI mode of ε-TEX. A handy command ‘fmtutil-sys’ (or ‘fmtutil’) for this purpose is available in TEX Live. The following command generates platex.fmt.

fmtutil-sys --bfmt platex

Formerly both pTEX and ε-TEX can make the format file for pLATEX, however, it’s not true anymore because LATEX requires ε-TEX since 2017.
The content of `platex.ltx` is shown below. In the current version of p\LaTeX, first we simply load `latex.ltx` and modify/extend some definitions by loading `plcore.ltx`.

1 (\begin{verbatim}\begin{verbatim}
Temporarily disable \texttt{\textbackslash 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, \texttt{hyphen.cfg} provided by “Babel” package will be used.
4 \input {latex.ltx}

Load `plcore.ltx`.
5 \typeout{**************************\textbackslash C}
6 * \textbackslash C
7 * making \texttt{pLaTeX format}\textbackslash C
8 * \textbackslash C
9 **************************}
10 \makeatletter
11 \input {plcore.ltx}

Load font-related default settings, `pldefs.ltx`. If a file `pldefs.cfg` is found, then that file will be used instead.
12 \InputIfFileExists{pldefs.cfg}
13 \{\typeout{*************************************\textbackslash C}
14 * Local config file \texttt{pldefs.cfg} used\textbackslash C
15 *************************************}}%
16 \{\input{pldefs.ltx}}

In the previous version, we displayed p\LaTeX version on the terminal, so that it can be easily recognized during format creation; however \texttt{\everyjob} can contain any code other than showing a banner, so now disabled.
17 \%the\{\everyjob

Load `platex.cfg` if it exists at runtime.
18 \everyjob\expandafter{\%
19 \the\everyjob
20 \InputIfFileExists{platex.cfg}{\%
21 \typeout{**************************\textbackslash C}
22 * Loading `platex.cfg`\textbackslash C
23 **************************}}%
24 \input{platex.cfg}}{\%
25 }

Dump to the format file.
26 \let\dump\orgdump
27 \let\orgdump\@undefined
The file `plcore.ltx`, which provides modifications/extensions to make \LaTeX\ 2\epsilon, is a concatenation of stripped files below using \texttt{docstrip} program.

- `plvers.dtx` defines the format version of \LaTeX\ 2\epsilon.
- `plfonts.dtx` extends NFSS for Japanese font selection.
- `plcore.dtx` defines other modifications to \LaTeX\ 2\epsilon.

Moreover, default settings of pre-loaded fonts and typesetting parameters are done by loading `pldefs.ltx` inside `platex.ltx`.\footnote{ASCII \LaTeX\ loaded `pldefs.ltx` inside `plcore.ltx`; however, \LaTeX\ community edition newer than 2018 loads `pldefs.ltx` inside `platex.ltx`.} This file `pldefs.ltx` is also stripped from `plfonts.dtx`.

**Attention:**

You can customize \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 “2019-10-01”) and the format name (“\LaTeX\ 2\epsilon”) of \LaTeX\ 2\epsilon are defined in `plvers.dtx`.

### 2.1.2 NFSS2 Commands

\LaTeX\ 2\epsilon uses NFSS2 as a font selection scheme, however, it supports only alphabetic fonts. \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 \LaTeX\ 2\epsilon.
2.1.3 Output Routine and Floats

`plcore.dtx` modifies and extends some `\LaTeX` 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 `\LaTeX` are based on those in original `\LaTeX`, with some Japanese localization.

`\LaTeX` 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`.

`\LaTeX` packages:

- `plext.sty`
  Useful macros and extensions for vertical writing; stripped from `plext.dtx`. 
3 Compatibility with Other Formats and Older Versions

Here we provide some information about the compatibility between current \LaTeX{} 2ε and older versions or original \LaTeX{} 2ε.

3.1 Compatibility with \LaTeX{} 2ε

\LaTeX{} 2ε is in most part upper compatible with \LaTeX{} 2ε, 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 \LaTeX{} 2ε and \LaTeX{} 2ε.

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

Some \LaTeX{} packages are known to be incompatible with \LaTeX{} 2ε. For those packages, \LaTeX{}-specific patches might be available. Please refer to the documentation of the \texttt{plautopatch} package (by Hironobu Yamashita).
3.2 Compatibility with \LaTeX{} 2.09

\plX{} has ‘\plX{} 2.09 compatibility mode”; use $\texttt{\documentstyle}$ 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’

\plX{} provides ‘platexrelease’ package, which is based on ‘latexrelease’ package (introduced in \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 \/orphan{DOCSTRIP Options}

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>plcore</td>
<td>Generates a fragment of format sources</td>
</tr>
<tr>
<td>pldoc</td>
<td>Generates ‘pldoc.tex’ for typesetting \plX{} sources</td>
</tr>
<tr>
<td>shprog</td>
<td>Generates a shell script to process ‘pldoc.tex’</td>
</tr>
<tr>
<td>plprog</td>
<td>Generates a tiny perl program to check DOCSTRIP guards nesting</td>
</tr>
<tr>
<td>Xins</td>
<td>Generates a DOCSTRIP batch file ‘Xins.ins’ for generating the above shell/perl scripts</td>
</tr>
</tbody>
</table>

B Documentation of \plX{} sources

The contents of ‘pldoc.tex’ for typesetting \plX{} 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 \plX{} sources is written in Japanese. If you need English version, first save $\texttt{\newif\ifJAPANESE}$ as \texttt{platex.cfg}, and process \texttt{pldoc.tex} (\plX{} Community Edition newer than July 2016 is required).
First, create `pldoc.dic`; it serves as a dictionary for ‘mendex’ (Japanese index processor\(^7\)), which is necessary for indexing control sequences containing Japanese characters (／西暦 and ／和暦).

\begin{filecontents}{pldoc.dic}
੢ྐྵ ͍ͤΕ͖
࿨ྐྵ ΘΕ͖
\end{filecontents}

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

\documentclass{jltxdoc}
\usepackage{plext}
\listfiles

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

\DoNotIndex{\def,\long,\edef,\zdef,\gdef,\let,\global}
\DoNotIndex{\if,\ifnum,\ifdim,\ifcat,\ifmode,\ifmmode,%}
\DoNotIndex{\iftrue,\iffalse,\ifvoid,\ifix,\ifeof,\ifcase,\else,\or,\fi}
\DoNotIndex{\box,\copy,\setbox,\unvbox,\unbbox,\bbox,\%}
\DoNotIndex{\vbox,\vtop,\vcenter}
\DoNotIndex{\@empty,\immediate,\write}
\DoNotIndex{\group,\bgroup,\expandafter,\begingroup,\endgroup}
\DoNotIndex{\divide,\advance,\multiply,\count,\dimen}
\DoNotIndex{\relax,\space,\string}
\DoNotIndex{\catcode,\endinput}
\DoNotIndex{\jobname,\message,\read,\the,\m@ne,\noexpand}
\DoNotIndex{\hsize,\vsize,\hfil,\hfill,\hss,\vss,\unskip}
\DoNotIndex{\n@me,\z@,\z@skip,\@ne,\@tw@,\@plus}
\DoNotIndex{\dp,\wd,\ht,\setlength,\addtolength}
\DoNotIndex{\newcommand,\renewcommand}

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

\ifJAPANESE
\IndexPrologue{\part*{索 引}％}
\markboth{索 引}{索 引}％
\addcontentsline{toc}{part}{索 引}％
\else
\IndexPrologue{\part*{Index}％}
\fi

\footnote{Developed by ASCII Corporation; the program ‘makeindex’ cannot handle Japanese characters properly, especially Kanji characters which should be sorted by its readings.}
Modify the standard \changes command slightly, to better cope with this multiple file document.

Codelines are allowed to run over a bit without showing up as overfull.

Section numbers now reach eg 19.12 which need more space.
Produce a Change Log and (2 column) Index.

\RecordChanges
\CodelineIndex
\EnableCrossrefs
\setcounter{IndexColumns}{2}
\settowidth\MacroIndent{\ttfamily\scriptsize 000 \ }

Set the title, authors and the date for this document.
\title{The \LaTeXe\ Sources}
\author{Ken Nakano \& Japanese \TeX\ Development Community}

% Get the date and patch level from plvers.dtx
\makeatletter
\let\patchdate=\@empty
\begingroup
\def\ProvidesFile#1\pfmtversion#2\patch@level#4{%
  \date{#2}\xdef\patchdate{#4}\endinput}
\input{plvers.dtx}
\endgroup

% Add the patch version if available.
% number is assumed
\ifnum\patchdate>0
  \edef\@date{\@date Patch level\space\patchdate}
\else
  \edef\@date{\@date Pre-Release\patchdate}
\fi

% Add the last update info, in case format date unchanged
% Note: \@ifl@t@r can be used only in preamble.
\def\lastupd@te{0000/00/00}
\begingroup
\def\ProvidesFile#1[#2 #3]{%
  \def\@tempd@te{#2}\endinput
  \@ifl@t@r{\@tempd@te}{\lastupd@te}{%
    \global\let\lastupd@te\@tempd@te
  }{}}
\let\ProvidesClass\ProvidesFile
\let\ProvidesPackage\ProvidesFile
\input{plvers.dtx}
\input{plfonts.dtx}
\input{plcore.dtx}
\input{plext.dtx}
\input{pl209.dtx}
\input{kinsoku.dtx}
\input{jclasses.dtx}
\input{jltxdoc.cls}
\endgroup
Here starts the document body.

\begin{document}
\pagenumbering{roman}
\maketitle
\renewcommand\maketitle{}
\tableofcontents
\clearpage
\pagenumbering{arabic}

\DocInclude{plvers} % pLaTeX version
\DocInclude{plfonts} % NFSS2 commands
\DocInclude{plcore} % kernel commands
\DocInclude{plex} % external commands
\DocInclude{pl209} % 2.09 compatibility mode commands
\DocInclude{kinsoku} % kinsoku parameter
\DocInclude{jclasses} % Standard class
\DocInclude{jltxdoc} % dtx documents class

Stop here if ltxdoc.cfg says AtEndOfClass\{\OnlyDescription}.  
\StopEventually{end{document}}

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

\clearpage
\pagestyle{headings}
% Make TeX shut up.
\hbadness=10000
\newcount\hbadness
\hfuzz=\maxdimen

% PrintChanges
\clearpage
%
% \begingroup
% \def\endash{--}
% \catcode'\-=active

11
C Additional Utility Programs

C.1 Shell Script mkpldoc.sh

A shell script to process 'pldoc.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.

```
sh prog
rm -f pldoc.toc pldoc.idx pldoc.glo
```

First run: empty the config file ltxdoc.cfg.
```
echo "" > ltxdoc.cfg
```

Now process pldoc.tex.
```
platex pldoc.tex
```
```
platex -jobname=pldoc-en 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 \LaTeX\2e. 

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.)
```
mendex -s gind.ist -d pldoc.dic -o pldoc.ind pldoc.idx
```
```
mendex -s gind.ist -d pldoc.dic -o pldoc-en.ind pldoc-en.idx
```
```
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.

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

### Perl Script `dstcheck.pl`

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

Usage:

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

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

```perl
if (/^%<\*(\[^>]+)>/) { # check conditions
    push(@dst,$1);
    push(@dst,$.);
} elsif (/^%</\([^>]+)>/) {
    $linenum = pop(@dst);
    $conditions = pop(@dst);
    if ($1 ne $conditions) {
        if ($conditions eq "DUMMY") {
            print "ARGV: '</$1>' (l.$.) is not started.
        }
        push(@dst,"DUMMY");
    } else {
        print "ARGV: '<*$conditions>' (l.$linenum) is ended by '<*$1>' (l.$.)
    }
}
```
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.
\input docstrip
\keepsilent
{\catcode'#=12 \gdef\MetaPrefix{## }}
\declarepreamble\thispre
\endpreamble
\usepreamble\thispre
\declarepostamble\thispost
\endpostamble
\usepostamble\thispost
\generate{
\file{dstcheck.pl}{\from{platex.dtx}{plprog}}
\file{mkpldoc.sh}{\from{platex.dtx}{shprog,ja}}
\file{mkpldoc-en.sh}{\from{platex.dtx}{shprog,en}}
}
\endbatchfile
⟨/Xins⟩
References

[1] 中野 賢『日本語 $\LaTeX$ 2e 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 $\LaTeX$』アスキー出版局, 1994


[10] 河野 真治『入門 Perl』アスキー出版局, 1994
## Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995/05/08</td>
<td>v1.0</td>
<td>first edition</td>
</tr>
<tr>
<td>1995/05/25</td>
<td>v1.0a</td>
<td>Added 'Compatibility', 'Usage of docstrip' and 'References'</td>
</tr>
<tr>
<td>1996/02/01</td>
<td>v1.0b</td>
<td>Adjusted for the latest docstrip (omake-sh.ins and omake-pl.ins)</td>
</tr>
<tr>
<td>1997/01/23</td>
<td>v1.0c</td>
<td>Adjusted for the latest docstrip.</td>
</tr>
<tr>
<td>1997/01/25</td>
<td>v1.0c</td>
<td>Add to filecontents environment for pldoc.dic.</td>
</tr>
<tr>
<td>1997/01/25</td>
<td>v1.0c</td>
<td>Rename pltpatch.ltx to plpatch.ltx.</td>
</tr>
<tr>
<td>2016/01/27</td>
<td>v1.0d</td>
<td>Add -e test before rm command</td>
</tr>
<tr>
<td>2016/02/16</td>
<td>v1.0e</td>
<td>Add a description of platexrelease</td>
</tr>
<tr>
<td>2016/04/12</td>
<td>v1.0f</td>
<td>Update document</td>
</tr>
<tr>
<td>2016/05/07</td>
<td>v1.0g</td>
<td>Save IPX banner</td>
</tr>
<tr>
<td>2016/05/08</td>
<td>v1.0h</td>
<td>Exclude pltpatch.ltx from the document</td>
</tr>
<tr>
<td>2016/05/12</td>
<td>v1.0i</td>
<td>Undefine temporary command \orgdump in the end.</td>
</tr>
<tr>
<td>2016/05/20</td>
<td>v1.0j</td>
<td>Add description of \pstrace</td>
</tr>
<tr>
<td>2016/05/21</td>
<td>v1.0k</td>
<td>Print also changes</td>
</tr>
<tr>
<td>2016/06/19</td>
<td>v1.0l</td>
<td>Get the patch level from plvers.dtx</td>
</tr>
<tr>
<td>2016/08/26</td>
<td>v1.0m</td>
<td>Moved loading platex.cfg from pcore.ltx to platex.ltx</td>
</tr>
<tr>
<td>2016/09/14</td>
<td>v1.0n</td>
<td>Improved banner saving method</td>
</tr>
<tr>
<td>2017/09/24</td>
<td>v1.0o</td>
<td>Allow negative patch level for pre-release</td>
</tr>
<tr>
<td>2017/01/25</td>
<td>v1.0c</td>
<td>Add to filecontents environment for pldoc.dic.</td>
</tr>
<tr>
<td>2019/09/29</td>
<td>v1.0z</td>
<td>Fix typos in document</td>
</tr>
</tbody>
</table>

17