About pLATEX 2ε

Ken Nakano & Japanese TeX Development Community

Date: 2020/09/28

pLATEX is a Japanese LATEX format, which is adjusted/extended to be more suitable for writing Japanese documents. It requires pTEX\textsuperscript{1}, a 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 pTEX” and “ASCII pLATEX” respectively.

In 2010, ASCII pTEX was incorporated into the world-wide TeX distribution, TeX Live. Since then, pTEX has been maintained/improved/changed along with TeX Live sources. In recent versions of TeX Live and W32TeX (around 2011), the default engine of pLATEX changed from original pTEX to ε-pTEX (pTEX with ε-TeX extension). Also, the original pTEX itself is also frequently updated. On the other hand, pLATEX remained unchanged since 2006, which resulted in some incompatibility and limitations.

To follow these upstream changes, we (Japanese TeX Development Community\textsuperscript{3}) decided to fork ASCII pLATEX 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 TeX Development Community, using GitHub Issue system.

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 It is somewhat of a historical document now, since pLATEX 2ε came into existence in 1995 (although the English translation has been done by Japanese TeX Development Community since 2017).

\textsuperscript{1}The pTEX website: https://asciidwango.github.io/ptex/ (in Japanese)
\textsuperscript{2}There is another old implementation of Japanese LATEX by NTT Electrical Communications Laboratories, named pTEX (unavailable in TeX Live). Also, MiKTeX has another program platex for Polish, but it has nothing to do with our Japanese pTEX!
\textsuperscript{3}https://texjp.org
\textsuperscript{4}https://github.com/texjporg/platex
1 Introduction to this document

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

For Japanese typesetting, please refer to the documentation of \texttt{pTEX} (or “Japanese \TeX”; the preliminary version of \texttt{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 \texttt{pLATEX 2e}. Also describes the standard classes and packages.
- **Section 3** The compatibility note for users of the old version of \texttt{pLATEX 2e} or those of the original \texttt{TEX}.
- **Appendix A** Describes \texttt{docstrip} Options for this document.
- **Appendix B** Description of ‘pldoc.tex’ (counterpart for ‘source2e.tex’ in \texttt{TEX} 2e).
- **Appendix C** Description of a shell script to process ‘pldoc.tex’, and a tiny perl program to check \texttt{docstrip} guards, etc.

2 About Functions of \texttt{pLATEX 2e}

The structure of \texttt{pLATEX 2e} is similar to that of \texttt{TEX} 2e; it consists of 3 types of files: a format (\texttt{platex.ltx}), classes and packages.

### 2.1 About the Format

To make a format for \texttt{pLATEX}, process “\texttt{platex.ltx}” with INI mode of \texttt{e-TEX}.ootnote{Formerly both \texttt{pTEX} and \texttt{e-TEX} can make the format file for \texttt{pLATEX}, however, it’s not true anymore because \texttt{TEX} requires \texttt{e-TEX} since 2017.} A handy command ‘\texttt{fmtutil-sys}’ (or ‘\texttt{fmtutil}’) for this purpose is available in \texttt{TEX} Live. The following command generates \texttt{latex.fmt}.

```
fmtutil-sys --byfmt platex
```
The content of `platex.ltx` is shown below. In the current version of \LaTeX, first we simply load `latex.ltx` and modify/extend some definitions by loading `plcore.ltx`.

1 \texttt{(\ast plcore)}

Temporarily disable `\dump` at the end of `latex.ltx`.

2 \texttt{\let\orgdump\dump}

3 \texttt{\let\dump\relax}

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

4 \texttt{\input latex.ltx}

Load `plcore.ltx`.

5 \texttt{\typeout{**************************\^^J%
   \*\^^J%
   * making pLaTeX format\^^J%
   \*\^^J%
   **************************}}

6 \texttt{\makeatletter}

7 \texttt{\input plcore.ltx}

Load font-related default settings, \texttt{pldefs.ltx}. If a file `pldefs.cfg` is found, then that file will be used instead. Some code may be executed after loading.

8 \texttt{\InputIfFileExists{pldefs.cfg}{\typeout{*************************************\^^J%
   \* Local config file pldefs.cfg used\^^J%
   \*************************************}}%}

9 \texttt{\input{pldefs.ltx}}

10 \texttt{\ifx\code@after@pldefs\@undefined\else \code@after@pldefs \fi}

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

11 \texttt{\%the\everyjob}

Load \texttt{platex.cfg} if it exists at runtime.

12 \texttt{\everyjob\expandafter{%}

13 \texttt{\the\everyjob}

14 \texttt{\IfFileExists{platex.cfg}{%}

15 \texttt{\typeout{**************************\^^J%
   \* Loading platex.cfg.\^^J%
   \**************************\^^J%

16 \texttt{\input{platex.cfg}}%}

17 \texttt{\fi}}

Dump to the format file.

18 \texttt{\let\dump\orgdump}
The file \texttt{plcore.ltx}, which provides modifications/extensions to make \LaTeX{} 2\epsilon, is a concatenation of stripped files below using \texttt{docstrip} program.

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

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

\textit{Attention:}

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

\subsection{Version}

The version (like “2020-10-01”) and the format name (“\LaTeX{}2e”) of \LaTeX{} 2\epsilon are defined in \texttt{plvers.dtx}.

\subsection{NFSS2 Commands}

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

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) NFSS 2, please refer to \texttt{fntguide.tex} in \LaTeX{} 2\epsilon.
2.1.3 Output Routine and Floats
plcore.dtx modifies and extends some \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 \LaTeX\,2\epsilon\ are based on those in original \LaTeX\,2\epsilon, with some Japanese localization.

\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.

\LaTeX\,2\epsilon\ packages:

- plex.sty
  Useful macros and extensions for vertical writing; stripped from plex.dtx.
• ptrace.sty

\LaTeX{} 2\varepsilon version of \texttt{tracefnt.sty}; the package \texttt{tracefnt.sty} overwrites \LaTeX{} 2\varepsilon-style NFSS2 commands, so \texttt{ptrace.sty} provides redefinitions to recover \LaTeX{} 2\varepsilon extensions. Stripped from \texttt{plfonts.dtx}.

• pltrace.sty

\LaTeX{} 2\varepsilon version of \texttt{ftrace.sty} (introduced in \LaTeX{} 2\varepsilon 2014/05/01); stripped from \texttt{plcore.dtx}.

• oldpfont.sty

Provides \LaTeX{} 2.09 font commands; stripped from \texttt{p1209.dtx}.

The packages “asmac.sty” and “nidanfloat.sty”, which had been included in previous versions of \LaTeX{} 2\varepsilon, is now distributed as a separate bundle.

3 Compatibility with Other Formats and Older Versions

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

3.1 Compatibility with \LaTeX{} 2\varepsilon

\LaTeX{} 2\varepsilon is in most part upward compatible with \LaTeX{} 2\varepsilon, 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{} 2\varepsilon and \LaTeX{} 2\varepsilon.

We hope that most classes and packages meant for \LaTeX{} 2\varepsilon works also for \LaTeX{} 2\varepsilon without any modification. However for example, if a class or a package redefines a command which is already modified by \LaTeX{} 2\varepsilon, it might cause an error at the worst case. We cannot tell whether a class or a package works fine with \LaTeX{} 2\varepsilon 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{} 2\varepsilon. 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 pLATEX 2.09

pLATEX 2ε has ‘pLATEX 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’

pLATEX 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 DOCSTRIp Options

By processing 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 pLATEX 2ε 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 pLATEX 2ε sources

The contents of ‘pldoc.tex’ for typesetting pLATEX 2ε 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 pLATEX 2ε sources is written in Japanese. If you need English version, first save

\newif\iffJAPANESE

as platex.cfg, and process pldoc.tex (pLATEX 2ε 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 `jltxdoc` class; we also require `plext` package, since `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,\iffalse,\ifnum,\ifdim,\ifcat,\ifmode,\ifmmode,\ifmode,\ifvmode,\ifhmode,\iftrue,\iffalse,\ifvoid,\ifx,\ifeof,\ifcase,\else,\or,\fi}
\DoNotIndex{\box,\copy,\setbox,\unvbox,\unhbox,\hbox,\vbox,\vtop,\vcenter}
\DoNotIndex{\@empty,\immediate,\write}
\DoNotIndex{\egroup,\bgroup,\expandafter,\begingroup,\endgroup}
\DoNotIndex{\divide,\advance,\multiply,\count,\dimen}
\DoNotIndex{\relax,\space,\string}
\DoNotIndex{\csname,\endcsname,\@spaces,\openin,\openout,\closein,\closeout}
\DoNotIndex{\catcode,\endinput}
\DoNotIndex{\jobname,\message,\read,\the,\m@ne,\noexpand}
\DoNotIndex{\hsize,\vsize,\hskip,\vskip,\kern,\hspace,\vspace,\hspace,\vspace}
\DoNotIndex{\make@size,\z@,\z@skip,\z@me,\z@s,\p@,\p@,\\@mm,\@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}}%
```

\(^7\)Developed by ASCII Corporation; the program ‘makeindex’ cannot handle Japanese characters properly, especially Kanji characters which should be sorted by its readings.
The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

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.

111 \RecordChanges
112 \CodelineIndex
113 \EnableCrossrefs
114 \setcounter{IndexColumns}{2}
115 \settowidth\MacroIndent{\ttfamily\scriptsize 000 \ }

Set the title, authors and the date for this document.

116 \title{The \LaTeXe\ Sources}
117 \author{Ken Nakano \& Japanese \TeX\ Development Community}

119 \setcounter{IndexColumns}{2}
120 \makeatletter
121 \let\patchdate=\@empty
122 \begingroup
123 \def\ProvidesFile#1\pfmtversion#2\pacth@level#4{\%
124 \date(#2)\def\patchdate(#4)\endinput}
125 \input{plvers.dtx}
126 \endgroup
128 \% Get the date and patch level from plvers.dtx
129 \makeatletter
130 \let\patchdate=\@empty
131 \begingroup
133 \def\ProvidesFile#1[#2 #3]{\%
134 \date(#2)\def\patchdate(#3)\endinput}
135 \input{plvers.dtx}
137 \input{plexpl3.dtx}
138 \input{plfonts.dtx}
139 \input{plcore.dtx}
140 \input{plext.dtx}
141 \input{pl209.dtx}
142 \input{kinsoku.dtx}
143 \input{jclasses.dtx}
144 \input{jltxdoc.cls}
Here starts the document body.

\begin{document}
\pagenumbering{roman}
\maketitle
\renewcommand{\maketitle}{\null}
\tableofcontents
\clearpage
\pagenumbering{arabic}
\\DocInclude{plvers} % pLaTeX version
\\DocInclude{plexpl3} % additions to expl3
\\DocInclude{plfonts} % NFSS2 commands
\\DocInclude{plcore} % kernel commands
\\DocInclude{plext} % 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 \texttt{ltxdoc.cfg} says \texttt{\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.
\% Make TeX shut up.
\hbadness=10000
\newcount\hbadness
\hbadness=\maxdimen
\% 
\PrintChanges
\clearpage
\%
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.

```bash
rm -f pldoc.toc pldoc.idx pldoc.glo
```

First run: empty the config file `ltxdoc.cfg`.

```bash
echo "" > ltxdoc.cfg
```

Now process `pldoc.tex`.

```bash
platex pldoc.tex
```

Make the Change log and Glossary (Change History) using mendex. ‘Mendex’ is a Japanese index processor, which is mostly upward 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 `ETX2ε`.

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.)
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.

```bash
223 echo "\includeonly{}" > ltxdoc.cfg
224 (ja)platex pldoc.tex
225 (en)platex -jobname=pldoc-en pldoc.tex
```

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

```bash
226 echo "" > ltxdoc.cfg
227 (ja)platex pldoc.tex
228 (en)platex -jobname=pldoc-en pldoc.tex
229 # EOT
230 (/shprog)
```

### C.2 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.

```bash
231 (plprog)
232 #
233 ## DOCSTRIP 文書内の環境や条件の入れ子を調べる perl スクリプト
234 ##
235 push(@dst,"DUMMY"); push(@dst,"000");
236 push(@env,"DUMMY"); push(@env,"000");
237 while (<>) {
238   if (/^%<\*(\[^>]+)>/) { # check conditions
239     push(@dst,$1);
240     push(@dst,$.);
241   } elsif (/^%</(\[^>]+)>/) {
242     $linenum = pop(@dst);
243     $conditions = pop(@dst);
244     if ($1 ne $conditions) {
245       if ($conditions eq "DUMMY") {
246         print "$ARGV: '</$1>' (l.$.) is not started.
";
247         push(@dst,"DUMMY");
248         push(@dst,"000");
249       } else {
250         print "$ARGV: '<*$conditions>' (l.$linenum) is ended ";
251       }
252     }
253   }
254 }
```

13
print "by '<*$1>' (l.$.)\n";  
}
}

if (/^% *\begin\{verbatim\}/) { # check environments
    while<> {
        last if (/^% *\end\{verbatim\}/);
    }
    elsif (/^% *\begin\{([^{}]+)\}\{(.*)\}/) {
        push(@env,$1);
        push(@env,$.);
    } elsif (/^% *\begin\{([^{}]+)\}/) {
        push(@env,$1);
        push(@env,$.);
    } elsif (/^% *\end\{([^{}]+)\}/) {
        $linenum = pop(@env);
        $environment = pop(@env);
        if ($1 ne $environment) {
            if ($environment eq "DUMMY") {
                print "$ARGV: '\end{$1}' (l.$.) is not started.\n";
                push(@env,"DUMMY");
                push(@env,"000");
            } else {
                print "$ARGV: '\begin{$environment}' (l.$.) is not ended.\n";
            }
        }
    }
    $linenum = pop(@dst);
    $conditions = pop(@dst);
    while ($conditions ne "DUMMY") {
        $linenum = pop(@dst);
        print "$ARGV: '<*$conditions>' (l.$.) is not ended.\n";
        $linenum = pop(@dst);
        $conditions = pop(@dst);
    }
else {
    $linenum = pop(@env);
    $environment = pop(@env);
    while ($environment ne "DUMMY") {
        print "$ARGV: '\begin{$environment}' (l.$.) is not ended.\n"
        $linenum = pop(@env);
        $environment = pop(@env);
    }
}
exit;

⟨/plprog⟩
Here we introduce a DOCTRIP 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{} 2ε 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 \TeX{}』 アスキー出版局, 1994


[8] Laslie Lamport. “\emph{\LaTeX{}}: 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 \LaTeX{}, アスキー, 1990)


[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>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. Don't copy gind.ist and gglo.ist from STEXMLF/tex/latex2e/base directory.</td>
</tr>
<tr>
<td>1997/01/25</td>
<td>v1.0c</td>
<td>Added to filecontents environment for pdedoc.dic.</td>
</tr>
<tr>
<td>1997/01/29</td>
<td>v1.0c</td>
<td>Rename pltexpatch.ltx to pltexpatch.ltx</td>
</tr>
<tr>
<td>2016/01/27</td>
<td>v1.0d</td>
<td>Add -e test before rm command . Updated descriptions of pLATEX 2e files</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 I\TeX \text{\textunderscore} banner</td>
</tr>
<tr>
<td>2016/05/08</td>
<td>v1.0h</td>
<td>Exclude pltexpatch.ltx from the document</td>
</tr>
<tr>
<td>2016/05/12</td>
<td>v1.0i</td>
<td>Undefine temporary command \texttt{\textbackslash{}orgdump} in the end.</td>
</tr>
<tr>
<td>2016/05/20</td>
<td>v1.0j</td>
<td>Add description of ‘plttrace’</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 pltvers.dtx</td>
</tr>
<tr>
<td>2016/08/26</td>
<td>v1.0m</td>
<td>Moved loading platex.cfg from plcore.ltx to platex.ltx</td>
</tr>
<tr>
<td>2016/09/14</td>
<td>v1.0n</td>
<td>Added compatibility, 'Usage of DOCSTRIP' and 'References'.</td>
</tr>
<tr>
<td>2017/01/23</td>
<td>v1.0c</td>
<td>Moved banner saving code from platex.ltx to plcore.ltx</td>
</tr>
<tr>
<td>2017/12/02</td>
<td>v1.0r</td>
<td>Added a description of platexrelease</td>
</tr>
<tr>
<td>2017/12/05</td>
<td>v1.0s</td>
<td>Moved loading default settings from plcore.ltx to platex.ltx</td>
</tr>
<tr>
<td>2018/02/07</td>
<td>v1.0t</td>
<td>Moved asmac package to separate bundle</td>
</tr>
<tr>
<td>2018/02/18</td>
<td>v1.0u</td>
<td>Moved nidanfloat package to separate bundle</td>
</tr>
<tr>
<td>2018/04/06</td>
<td>v1.0v</td>
<td>Sync with the latest source2e.tex</td>
</tr>
<tr>
<td>2018/04/08</td>
<td>v1.0w</td>
<td>Stop showing banner during format generation for safety</td>
</tr>
<tr>
<td>2018/09/03</td>
<td>v1.0x</td>
<td>Mention platexcheat (Japanese only)</td>
</tr>
<tr>
<td>2018/09/22</td>
<td>v1.0y</td>
<td>Mention platautopatch.</td>
</tr>
<tr>
<td>2018/09/29</td>
<td>v1.0z</td>
<td>Show last update info on pdedoc.pdf</td>
</tr>
<tr>
<td>2019/09/29</td>
<td>v1.0a</td>
<td>Add hook after loading defs</td>
</tr>
<tr>
<td>2020/03/24</td>
<td>v1.1</td>
<td>Update document</td>
</tr>
<tr>
<td>2020/06/26</td>
<td>v1.1a</td>
<td>Add hook after loading defs</td>
</tr>
<tr>
<td>2020/09/28</td>
<td>v1.1b</td>
<td>Add hook after loading defs</td>
</tr>
</tbody>
</table>

17