About \texttt{pL\LaTeX} 2\(\varepsilon\)

Ken Nakano & Japanese \TeX Development Community

Date: 2022/03/06

\texttt{pL\LaTeX} is a Japanese \LaTeX format, which is adjusted/extended to be more suitable for writing Japanese documents. It requires \texttt{p\LaTeX}\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 \texttt{p\LaTeX}” and “ASCII \texttt{p\LaTeX}” respectively.

In 2010, ASCII \texttt{p\LaTeX} was incorporated into the world-wide \TeX distribution, \TeX Live. Since then, \texttt{p\LaTeX} has been maintained/improved/changed along with \TeX Live sources. In recent versions of \TeX Live and W32\TeX (around 2011), the default engine of \texttt{p\LaTeX} changed from original \texttt{p\LaTeX} to \(\varepsilon\)-\texttt{p\LaTeX} (\texttt{p\LaTeX} with \(\varepsilon\)-\TeX extension). Also, the original \texttt{L\LaTeX} itself is also frequently updated. On the other hand, \texttt{pL\LaTeX} 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 \texttt{pL\LaTeX} 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 (\texttt{platex-en.pdf}) is a brief explanation of the \texttt{pL\LaTeX} 2\(\varepsilon\) community edition. It is somewhat of a historical document now, since \texttt{pL\LaTeX} 2\(\varepsilon\) came into existence in 1995 (although the English translation has been done by Japanese \TeX Development Community since 2017).

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

This document briefly describes \texttt{pLATEX 2\epsilon}, but is not a manual of \texttt{pLATEX 2\epsilon}. For the basic functions of \texttt{pLATEX 2\epsilon}, 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 \texttt{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 2\epsilon}. Also describes the standard classes and packages.

Section 3 The compatibility note for users of the old version of \texttt{pLATEX 2\epsilon} or those of the original \texttt{LATEX 2\epsilon}.

Appendix A Describes DOCSTRIP Options for this document.

Appendix B Description of ‘pldoc.tex’ (counterpart for ‘source2e.tex’ in \texttt{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 \texttt{pLATEX 2\epsilon}

The structure of \texttt{pLATEX 2\epsilon} is similar to that of \texttt{LATEX 2\epsilon}; 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{\varepsilon-pLATEX}.\footnote{Formerly both \texttt{pTEX} and \texttt{\varepsilon-TEX} can make the format file for \texttt{pLATEX}, however, it’s not true anymore because \texttt{LATEX} requires \texttt{\varepsilon-TEx} since 2017.} A handy command ‘fmtutil-sys’ (or ‘fmtutil’) for this purpose is available in \texttt{TEX Live}. The following command generates \texttt{platex.fmt}.

\begin{verbatim}
fmtutil-sys --byfmt platex
\end{verbatim}

The content of \texttt{platex.ltx} is shown below. In the current version of \texttt{pLATEX}, first we simply load \texttt{latex.ltx} and modify/extend some definitions by loading \texttt{plcore.ltx}.

1 (*plcore)
Temporarily disable `\dump` at the end of `latex.ltx`.

```latex
\let\orgdump\dump
\let\dump\relax
```

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

```latex
\input latex.ltx
```

If `\typeout` is still undefined, the input of \LaTeX{} kernel should have failed; abort now.

```latex
\ifx\typeout\undefined
\errhelp{Please reinstall LaTeX, or check e-TeX availability.}\
\errmessage{Failed to load 'latex.ltx' properly}\
\expandafter\end
\fi
```

Load `plcore.ltx`.

```latex
\ifx\typeout\undefined
\errhelp{Please reinstall LaTeX, or check e-TeX availability.}\
\errmessage{Failed to load 'latex.ltx' properly}\
\expandafter\end
\fi
```

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

```latex
\InputIfFileExists{pldefs.cfg}{\typeout{*************************************^^J%*
Local config file pldefs.cfg used^^J%*************************************}}%
\ifx\code@after@pldefs\@undefined\else \code@after@pldefs \fi
```

In the previous version, we displayed pLaTeX 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.

```latex
%\the\everyjob
```

Load `platex.cfg` if it exists at runtime.

```latex
\everyjob\expandafter{%\the\everyjob
\IfFileExists{platex.cfg}{%\typeout{***************************^^J%*
Loading platex.cfg.^^J%***************************}%
\input{platex.cfg}}{}}%
```

Dump to the format file.

```latex
\let\dump\orgdump
\let\orgdump\@undefined
\makeatother
```
The file `plcore.ltx`, which provides modifications/extensions to make \LaTeX 2\epsilon, is a concatenation of stripped files below using DOCSTRIP program.

- `plvers.dtx` defines the format version of \LaTeX 2\epsilon.
- `plfonts.dtx` extends NFSS2 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`.

\section*{2.1.1 Version}

The version (like “2021-11-15”) and the format name (“\LaTeX 2\epsilon”) of \LaTeX 2\epsilon are defined in `plvers.dtx`.

\section*{2.1.2 NFSS2 Commands}

\LaTeX 2\epsilon uses NFSS2 as a font selection scheme, however, it supports only alphabetic fonts. `pl\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.

\section*{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{}2ε are based on those in original \LaTeX{}2ε, with some Japanese localization.

\LaTeX{}2ε classes:

• jarticle.cls, jbook.cls, jreport.cls
  Standard \textit{yoko-kumi} (horizontal writing) classes; stripped from jclasses.dtx.

• tarticle.cls, tbook.cls, treport.cls
  Standard \textit{tate-kumi} (vertical writing) classes; stripped from jclasses.dtx.

• jltxdoc.cls
  Class for typesetting Japanese .dtx file; stripped from jltxdoc.dtx.

\LaTeX{}2ε packages:

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

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

• pfltrace.sty
  \LaTeX{}2ε version of fltrace.sty (introduced in \LaTeX{}2ε 2014/05/01); stripped from plcore.dtx.

5
• oldpfont.sty
  Provides \LaTeX{} 2.09 font commands; stripped from \texttt{pl209.dtx}.

The packages “ascmac.sty” and “nidanfloat.sty”, which had been included in previous versions of \LaTeX{}, 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{} 2ε and older versions or original \LaTeX{} 2ε.

3.1 Compatibility with \LaTeX{} 2ε

\LaTeX{} 2ε is in most part upward 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{}. 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

\LaTeX{} 2ε has ‘\platex{} 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’

\LaTeX{} 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 \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>plocore</td>
<td>Generates a fragment of format sources</td>
</tr>
<tr>
<td>pldoc</td>
<td>Generates \texttt{pldoc.tex} for typesetting \LaTeX\ sources</td>
</tr>
<tr>
<td>shprog</td>
<td>Generates a shell script to process \texttt{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 \texttt{Xins.ins} for generating the above shell/perl scripts</td>
</tr>
</tbody>
</table>

B  Documentation of \LaTeX\ sources

The contents of \texttt{pldoc.tex} for typesetting \LaTeX\ sources is described here. Compared to individual processings, batch processing using \texttt{pldoc.tex} prints also changes and an index. The whole document will have about 200 pages.

By default, the description of \LaTeX\ sources is written in Japanese. If you need English version, first save\footnote{Developed by ASCII Corporation; the program ‘makeindex’ cannot handle Japanese characters properly, especially Kanji characters which should be sorted by its readings.} \begin{verbatim}
\newif\ifJAPANESE
\end{verbatim}
as \texttt{platex.cfg}, and process \texttt{pldoc.tex} (\LaTeX\ Community Edition newer than July 2016 is required).

First, create \texttt{pldoc.dic}; it serves as a dictionary for `mendex' (Japanese index processor\footnote{Developed by ASCII Corporation; the program ‘makeindex’ cannot handle Japanese characters properly, especially Kanji characters which should be sorted by its readings.}), which is necessary for indexing control sequences containing Japanese characters (\textbackslash 西暦 and \textbackslash 和暦).

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

We use \texttt{jitxdoc} class; we also require \texttt{plext} package, since \texttt{plext.dtx} contains several examples of partial vertical writing. Also, as of 2022 \texttt{doc} package v3.0 depends on \texttt{hyperref}, so we add a driver option \texttt{dvipdfmx} and load \texttt{pxjahyper} using \texttt{plautopatch} scheme. \texttt{plautopatch} 記由で \texttt{pxjahyper} パッケージも読み込みます。

\begin{verbatim}
\RequirePackage{plautopatch}
\documentclass[dvipdfmx,a4paper]{jitxdoc}
\AddToHook{env/macro/before}{\catcode\_=12\relax}
\AddToHook{env/macro/after}{\catcode\_=8\relax}
\end{verbatim}
Do not index some \TeX primitives, and some common plain \TeX commands.
\DoNotIndex{\def, \long, \edef, \xdef, \global}
\DoNotIndex{\if, \ifnum, \ifdim, \ifcat, \ifmmode, \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, \␣, \noexpand}
\DoNotIndex{\hsize, \vsize, \hskip, \vskip, \kern, \hfil, \hfill, \hss, \vss, \unskip}
\DoNotIndex{\m@ne, \z@, \z@skip, \␣, \tw@, \p@, \@plus}
\DoNotIndex{\newcommand, \renewcommand}

Set up the Index and Change History to use \part.
\ifJAPANESE
\IndexPrologue{索 引}
\markboth{索 引}{索 引}
\addcontentsline{toc}{part}{索 引}
\else
\IndexPrologue{Index}
\markboth{Index}{Index}
\addcontentsline{toc}{part}{Index}
\fi
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.
\ifJAPANESE
\GlossaryPrologue{変更履歴}
\markboth{変更履歴}{変更履歴}
\addcontentsline{toc}{part}{変更履歴}
\else
\GlossaryPrologue{Change History}
\markboth{Change History}{Change History}
\addcontentsline{toc}{part}{Change History}
\fi

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.

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

% Get the date and patch level from plvers.dtx
\makeatletter
\let\parindent\@empty
\begingroup
\def\ProvidesFile#1\pfmtversion#2#3\ppatch@level#4{\
  \date{#2}xdef\patchdate{#4}\endinput}
\input{plvers.dtx}
\endgroup

% Add the patch version if available.
\def\Xpatch{0}
\ifx\patchdate\Xpatch\else
  \ifnum\patchdate>0
    \textit{\textsc{The \LaTeXe\ Sources}}
    \author{Ken Nakano & Japanese \TeX\ Development Community}
  \fi
\fi

% Add the patch version if available.
\edef\@date{\@date\space Patch level\space patchdate}
\else
\edef\@date{\@date\space Pre-Release\space patchdate}
\fi

% Add the last update info, in case format date unchanged
% Note: \@ifl@t@r can be used only in preamble.
\def\lastupdate{0000/00/00}
\begingroup
\def\ProvidesFile#1[#2 #3]{%
  \def\@tempdate{#2}\endinput
  \@ifl@t@r{\@tempdate}{\lastupdate}{%
    \global\let\lastupdate@te\@tempdate
  }{}%}
\let\ProvidesClass\ProvidesFile
\let\ProvidesPackage\ProvidesFile
\input{plvers.dtx}
\input{plexpl3.dtx}
\input{plfonts.dtx}
\input{plcore.dtx}
\input{plext.dtx}
\input{pl209.dtx}
\input{kinsoku.dtx}
\input{jclasses.dtx}
\input{jltxdoc.cls}
\endgroup
\@ifl@t@r{\lastupdate@te}{\pfmtversion}{%
  \edef\@date{\@date\break (last updated: \lastupdate@te)}%
}{}
\makeatother

Here starts the document body.
\begin{document}
\pagenumbering{roman}
\maketitle
\renewcommand\maketitle{}
\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
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.
\hbadness=10000
\newcount\hbadness
\hfuzz=\maxdimen
\%\PrintChanges
\clearpage
\%
\begingroup
\def\endash{--}
\catcode’-\active
\def-\texttt{\futurelet\temp\indexdash}
\def\indexdash{\ifx\temp-\endash\fi}
\PrintIndex
\endgroup

Make sure that the index is not printed twice (\texttt{ltxdoc.cfg} might have a second command).
\let\PrintChanges\relax
\let\PrintIndex\relax
\end{document}

\section*{C Additional Utility Programs}

\subsection*{C.1 Shell Script \texttt{mkpldoc.sh}}

A shell script to process `pldoc.tex' and produce a fully indexed source code description. Run \texttt{sh mkpldoc.sh} to use it.

\subsubsection*{C.1.1 Content of \texttt{mkpldoc.sh}}

First, delete auxiliary files which might be created in the previous runs.
\begin{verbatim}
(*shprog*)
(ja) rm -f pldoc.toc pldoc.idx pldoc.glo
(en) rm -f pldoc-en.toc pldoc-en.idx pldoc-en.glo
\end{verbatim}
First run: empty the config file ltxdoc.cfg.

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

Now process pldoc.tex.

```bash
(ja) platex pldoc.tex
=en 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 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 \LaTeX\ 2\epsilon.

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.)

```bash
(ja) mendex -s gind.ist -d pldoc.dic -o pldoc.ind pldoc.idx
=en mendex -s gind.ist -d pldoc.dic -o pldoc-en.ind pldoc-en.idx
(ja) mendex -f -s gglo.ist -o pldoc.gls pldoc.glo
=en mendex -f -s gglo.ist -o pldoc-en.gls pldoc-en.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.

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

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

```bash
echo "" > ltxdoc.cfg
(ja) platex pldoc.tex
=en platex -jobname=pldoc-en pldoc.tex
```

# EOT

## DOCSTRIP

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
<<plprog
push(@dst,"DUMMY"); push(@dst,"000");
push(@env,"DUMMY"); push(@env,"000");
while (<>) {
```

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

13
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
\let\catcode\'=#12 \edef\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
References


[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>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995/05/08</td>
<td>v1.0</td>
<td>first edition</td>
</tr>
<tr>
<td>1995/08/25</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 $TEXMF/tex/latex2e/base directory.</td>
</tr>
<tr>
<td>1997/01/25</td>
<td>v1.0d</td>
<td>Add to filecontents environment for pdl.doc. Rename plpatch.ltx to plpatch.ltx.</td>
</tr>
<tr>
<td>2016/01/27</td>
<td>v1.0e</td>
<td>Add -e test before rm command. Updated descriptions of \return@\if@twocolumn\TeX\else\LaTeX\fi files.</td>
</tr>
<tr>
<td>2016/02/16</td>
<td>v1.0f</td>
<td>Add a description of platexrelease.</td>
</tr>
<tr>
<td>2016/04/12</td>
<td>v1.0g</td>
<td>Update document.</td>
</tr>
<tr>
<td>2016/05/07</td>
<td>v1.0h</td>
<td>Save \return@\if@twocolumn\TeX\else\LaTeX\fi banner.</td>
</tr>
<tr>
<td>2016/05/08</td>
<td>v1.0i</td>
<td>Exclude plpatch.ltx from the document.</td>
</tr>
<tr>
<td>2016/05/12</td>
<td>v1.0j</td>
<td>Undefine temporary command \orgdump in the end.</td>
</tr>
<tr>
<td>2016/05/20</td>
<td>v1.0k</td>
<td>Add description of `pfltrace'.</td>
</tr>
<tr>
<td>2016/05/21</td>
<td>v1.0l</td>
<td>Print also changes.</td>
</tr>
<tr>
<td>2016/06/19</td>
<td>v1.0m</td>
<td>Get the patch level from plvers.dtx.</td>
</tr>
<tr>
<td>2016/08/26</td>
<td>v1.0n</td>
<td>Moved loading platex.cfg from plcire.ltx to platex.ltx.</td>
</tr>
<tr>
<td>2016/09/14</td>
<td>v1.0o</td>
<td>Improved banner saving method.</td>
</tr>
<tr>
<td>2017/09/24</td>
<td>v1.0p</td>
<td>Allow negative patch level for pre-release.</td>
</tr>
<tr>
<td>2017/11/11</td>
<td>v1.0q</td>
<td>Moved banner saving code from platex.ltx to plcire.ltx.</td>
</tr>
<tr>
<td>2017/11/29</td>
<td>v1.0r</td>
<td>New English documentation added.</td>
</tr>
<tr>
<td>2017/12/02</td>
<td>v1.0s</td>
<td>English references added.</td>
</tr>
<tr>
<td>2017/12/05</td>
<td>v1.0t</td>
<td>Moved loading default settings from plcire.ltx to platex.ltx.</td>
</tr>
<tr>
<td>2018/02/07</td>
<td>v1.0u</td>
<td>Moved asmac 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>2018/09/22</td>
<td>v1.0x</td>
<td>Show last update info on pdl.doc.</td>
</tr>
<tr>
<td>2020/03/24</td>
<td>v1.1</td>
<td>Update document.</td>
</tr>
<tr>
<td>2020/09/26</td>
<td>v1.1a</td>
<td>Add plexpl3.dtx.</td>
</tr>
<tr>
<td>2020/09/28</td>
<td>v1.1b</td>
<td>Add hook after loading defs.</td>
</tr>
<tr>
<td>2021/02/25</td>
<td>v1.1c</td>
<td>Check for latex.ltx status.</td>
</tr>
<tr>
<td>2021/03/14</td>
<td>v1.1d</td>
<td>Print expl3 commands correctly.</td>
</tr>
<tr>
<td>2022/03/06</td>
<td>v1.1e</td>
<td>Adapt to new ltxdoc.cls.</td>
</tr>
</tbody>
</table>